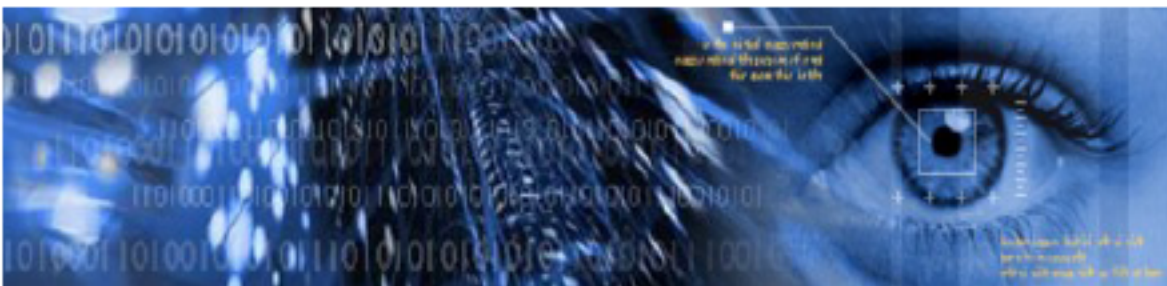


Analyse and use perception to amplify cognitive complexities

SECURITY ANALYSIS AND DATA VISUALIZATION



HIGHLIGHTS...

- Multivariate Data analysis
- Protocol Agnostic Packet Reconstruction
- Statistical Protocol Identification (SPID)
- Visual Mapping and Human Computer Interaction (HCI)
- Effects of Deep Packet Inspection (DPI)
- Data Protection and Data Leak Prevention (DLP)

...Includes over 120 Illustrations and 38 Case Studies

Security Analysis And Data Visualization

Olu Akindeinde

October 16, 2009

Copyright © 2009 Olu Akindeinde

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in Appendix B entitled "GNU Free Documentation License".

Security Analysis and Data Visualization

PREFACE

This is not a book on information security assessment methodologies, neither is it on penetration testing techniques: it is however, about a subtle combination of the core elements of both, but this time employed in the process of interactive security analysis and statistical graphics. Why a book on security analysis and visualization? Well, over the last couple of years, the analysis and visualization concepts have slowly become important components of overall enterprise security assessment albeit devoid of proper acknowledgment. In reality though, both are relatively uncharted territories even in the field of computing. Notwithstanding this surrounding conspiracy of obscurity, this book explores the subject matter in great depth. *Security Analysis and Data Visualization* primarily concerns itself with the use of perception to recognize and amplify cognitive complexities. This book therefore, sets out to present an interactive and practical approach to these abstractions.

In modern business, to effectively compete and excel, organizations must increasingly raise service delivery and operational levels whilst expanding reach. Today's global economic climate creates considerable obstacles to managing the enterprise. The proliferation of bandwidth and telecommuting via remote access, as well as worldwide emerging markets, are in essence driving the development of a virtual workforce which is widely distributed across a rapidly rising number of remote sites. A network is therefore required to provide the personnel at these locations access to the applications, systems, and data that often reside within the head quarter's data center. As a result of employee productivity's dependence on network security and health, statistical network analysis and interactive visualization becomes a pivotal component of the entire organizational security process.

The team responsible for maintaining the network security is constantly living with the day-to-day analysis of security data, a fact hitherto lost on most enterprises. Data leakages, log monitoring, fault analysis, packet captures, alerts and even binary files take time and effort to analyze using text-based and non-intuitive tools - and once the analysis is complete, the picture does not always reflect the root cause and the result isn't necessarily clear, or even timely for that matter and in the analysis of enterprise security, time is always of the essence.

Endeavours towards security data retrieval and meaningful exploration are usually a major concern for most security analysts. The amount and variety of the information contained in the raw data is sometimes an impediment to developing concise conclusions and obtaining a high level view of the real events hidden behind audit trails. However, the crystallization of

security events in an analytical and visual representation will lead to a higher quality security analysis than the one obtained from the text-based reports and even from more traditional graphical approaches such as pie or bar charts. Perhaps, a subtle but often overlooked advantage of the statistical representation of security data is that while it is unnatural for human beings to remember patterns expressed in a static file, it is fairly straight forward to remember spatial objects such as pictures and maps or visual diagrams based on event data.

To tackle today's information security landscape therefore, there is a need for new methods of analysis. Pattern recognition and trending through detailed data exploration and visualization can be used to profile threats on the data link through to the application layer of the enterprise architecture.

Security Analysis and Data Visualization offers perspectives into what is happening in and around the network of an enterprise. Information pertaining to traffic flows, protocols, and even individual data packets can induce the team responsible for the network security to keep it operating securely and at peak performance. As a result, the net benefit viz-a-viz information reduction compared with the unmanageable number of events on the audit trails provides us with an aggregate form of handling when the analyst intends to do a meaningful iteration on various security data elements.

On a final note, this book is heavily tilted in favour of the use of free and open source tools (both on Microsoft Windows and Linux platforms). Part of the rationale for this is to bring the analyst up to speed with the concepts and techniques of security analysis and visualization without having a recourse to proprietary tools and applications. I think in my humble estimation, it bridges the knowledge gap quicker whilst bringing the core subject matter to the fore.

HOW THIS BOOK IS ORGANIZED

This book consists primarily of four distinct but complementary parts modeled on the concept of **Capture -> Process -> Visualize -> Govern**.

Part 1: Capture

Chapter 1 - Security Data Acquisition: chronicles the challenges facing security analysts in gaining access to information so they can conduct security analysis and respond accord-

ingly. In view of this, the chapter begins with an appraisal of the techniques of collecting and managing network security data set. It further examines data source formats as well as methods of extracting and preserving security data for analysis.

Chapter 2 - Security Data Integration: delves into the multivariate analysis and methodology for integrating enterprise security data by examining packet inspection and traffic flow. Furthermore, we introduce the concept of security data parsing and advance analysis of unstructured data, finishing off by taking an in-depth look at the tools and applications that will aid in completing the initial stage of security analysis.

Part 2: Process

Chapter 3 - Security Data Carving: examines active and passive security data carving and data mining viz-a-viz statistical protocol Identification (SPID) and port independent protocol identification (PIPI) algorithms - using advance techniques of packet regeneration and reassembly for statistical analysis.

Chapter 4 - Security Data Exploration: forages into interactive off-line security data reconstruction and exploration techniques. This is done by way of free form concrete and contextual reassembly of raw network security data.

Part 3: Visualize

Chapter 5 - Security Visual Mapping: investigates the concept of visualization, the processes involved as well as the various elements of security data dissemination and mapping from generic free form graphs and charts to scatter plots, link graphs and parallel coordinates. The chapter further discusses when and how to choose the right graph for any specific data visualization scenario.

Chapter 6 - Security Visualization Techniques: The main goal of this chapter is to aid the analyst in conducting decision support and communicating information clearly and effectively through graphical means. In order to convey ideas effectively, we need to marry both aesthetic form and functionality so as to provide insights into a rather sparse and complex data set by communicating its key aspects in a more intuitive way. With this in mind, the chapter starts by investigating the techniques of time-based visualization

of security data set. It then examines some of the components of interactive visual mapping of data variables and attributes already abstracted. Lastly, it examines the tools and applications available for interactive data visualization and statistical graphics.

Part 4: Govern

Chapter 7 - Security Data Management: We plumb the discipline that embodies a convergence of governance and data management surrounding the handling of security data in an organization. The concepts of deep packet inspection, privacy, data loss prevention and data protection will be highlighted

Finally, The ultimate aim of this book is not only to get security analysts interested in this field of endeavour but be stimulated enough to go forthwith and infix security data analysis and statistical visualization in their daily itinerary. The book gives all the necessary information and tools at the same time illustrating test and use-case scenarios that should assist in applying the concepts to security analysis difficulties.

AUDIENCE

This book is primarily written for security administrators, network and security analysts as well as auditors and digital forensic investigators. If you want to learn security analysis and data visualization, then this book is for you. I bring to the fore new ways of analyzing security data: from the analysis of perimeter bottlenecks, insider threats, intrusion detection, compliance monitoring through to fraud detection and forensics.

I assume that the reader has a basic understanding of networking concepts and is conversant with the TCP/IP model. I also assume that the reader is fairly familiar with the Linux environment especially basic CLI operations as well as basic installation and execution of Linux binary applications (rpm anyone!). However, you don't have to be an expert in IT security or programming. I try as much as possible to provide clear and comprehensive explanations of the concepts you need to understand and the procedures you need to perform so as to assist you in your day-to-day security analysis experience.

DEDICATION

This book is dedicated to my late brother and grandmother.

They were my friends and my confidants.

*They were loved by everyone who knew them,
and they were described as angels by family and friends.*

They were my precious angels

Though I can't see them anymore,

I thank God for the blessing of His gifts to me.

ACKNOWLEDGMENTS

"The power of conception outweighs the power of birth. For something to be born, it has to be conceived."

This book owes much to the people involved in the collation and review process, without whose support it would never have seen the light of day. A further special note of thanks goes to all the staff of **Inverse** and **Digital Encode** whose contributions throughout the whole process, from inception of the initial idea to final publication, have been invaluable. In particular I wish to thank **Wale Obadare**, **Sina Owolabi** and **Ola Amudipe** - brilliant minds, artists and scientists creating the future - for their insights and excellent contributions to this book. Many thanks also to everyone who assisted me in the review process.

Finally, I say thank you to my glorious mother and best friend **Mrs T.A. Akindeinde** - a perfect convergence of grace and substance, a paragon of virtues without whom I may never have been able to write this book. Many women do noble things, but mum, you surpass them all. Thank you once again.

Olu Akindeinde
Lagos, Nigeria
October 2009

ABOUT THE AUTHOR

Olu has 9 years experience working in the IT and information security arena, but has spent the better part of the last few years exploring the security issues faced by Electronic Funds Transfer (EFT) and Financial Transaction Systems (FTS). He has presented the outcome of his research work at several conferences; including the Information Security Society of Africa (ISS), the forum of the Committee of Chief Inspectors of Banks in Nigeria, the apex bank - Central Bank of Nigeria (CBN) as well as 8 of the top 10 financial institutions in Nigeria.

In his professional life, Seyi, as he is otherwise called, sits on the board of two companies. In addition to being the CTO, he holds a vital position as the Senior Security Analyst at Digital Encode Ltd an information security advisory and assurance company, not only performing various technical security assessments and digital forensics but also providing technical consulting in the field of security design and strategic technology reviews for top notch local clients. He has over the years developed an in-depth knowledge of security modeling which has hitherto improved his ability to initiate, perform and deliver world class enterprise security services that add veritable value to the corporate goals and objectives of organizations.

Olu is the author of the Open Source Security Assessment Report (OSSAR) - a model framework for reporting and presenting enterprise security assessment findings. He is a speaker on matters bordering on information security, and has presented technical papers on a wide range of IT security and risk management topics for a number of high profile financial service providers at different retreats and forums. Furthermore, he has delivered several information security and ethical hacking training courses to delegates from diverse industries including finance, manufacturing, oil and gas, telecoms as well as State and Federal Government Agencies. He has administered security analysis and penetration testing courses to representatives of the National Assembly, Defense Intelligence Agency (DIA) and Office of the National Security Agency (NSA) through the annual Hacker Counterintelligence Program (HACOP) where he's been involved as a resident trainer and technical consultant for the last couple of years.

As a foremost exponent and consummate advocate of open source software, he championed the use of Linux and open source software in a few of the local institutions of higher learning. He subsequently led a team to deploy these solutions in such schools as LASU (2005) and Fountain University (2008). Olu has used different variants of the Linux OS primarily as his

platform of choice for the last 10 years. He is also the founder of Inverse Information Systems Ltd an open source professional services company providing open source business solutions and Linux consulting services. Having forged distinct alliances with industry technology leaders, the company currently boasts of some of the biggest open source infrastructure deployments in the country with clients mainly in the Financial, Pension Funds, Insurance, Diversified Services and Service Provider sectors of the economy. Olu instituted a model for the delivery of structured Linux training through the Applied Linux Institute, now a wholly owned division of Inverse. He has delivered countless of such trainings to many delegates and organizations.

Finally, he holds a Bachelor of Science (BSc) Degree in Civil Engineering from the University of Lagos, Nigeria. In his spare time he loves to drive in fast cars and enjoys playing Flight Simulator and Pro Evolution Soccer (PES) on PS3. He considers himself an amateur student of Business and will like to pursue a Doctorate program in Economics down the line. He also harbors the dream of flying a helicopter in his lifetime.

Contents

I	CAPTURE	1
1	Security Data Acquisition	3
1.1	Sources of Security Data	4
1.2	TCP/IP Reference Model	5
1.2.1	Network Interface Layer	5
1.2.2	Internet Layer	6
1.2.3	Transport Layer	7
1.2.4	Application Layer	7
1.3	Security Data Capture	7
1.4	Network Interface Layer	8
1.4.1	Switch Port Mirroring	8
1.4.2	Man-in-the-middle	9
1.4.3	Packet Redirection through ARP Poisoning	10
1.4.3.1	Ettercap	10
1.4.3.2	Case Study 1: ARP Poisoning with Ettercap	12
1.4.4	Network Packet Sniffing	14
1.4.4.1	Tcpdump	15
1.4.4.2	Case Study 2: Packet Capture with Tcpdump	16
1.4.4.3	Wireshark	17
1.4.4.4	Case Study 3: Packet Capture with Wireshark	20
1.4.4.5	Extracting and Preserving Security Data	21

1.5	Internet and Transport Layers	23
1.5.1	Ports	23
1.5.2	Socket	24
1.5.3	Port Scanning	25
1.5.3.1	Unicornsca n	26
1.5.3.2	Case Study 4: Basic Scan with Unicornsca n	27
1.5.3.3	Case Study 5: Advance Scan with Unicornsca n	28
1.5.3.4	Nmap	30
1.5.3.5	Case Study 6: Basic Nmap Scan	32
1.5.3.6	Case Study 7: Nmap Target and Output Specifications	34
1.6	Application Layer	35
1.6.1	Vulnerability Assessment	36
1.6.1.1	Nessus	36
1.6.1.2	Case Study 8: Vulnerability Assessment with Nessus	38
1.7	Summary	39
2	Security Data Analysis	41
2.1	Protocol and Packet Analysis	41
2.1.1	Packet Capture Libraries	42
2.1.2	Packet Analysis with Wireshark	44
2.1.2.1	Case Study 9: Investigating Spyware Infection	44
2.1.2.2	Case Study 10: Malware Behaviour Analysis	45
2.2	Traffic Flow Analysis	48
2.2.1	Audit Record Generation and Utilization System	48
2.2.2	Security Analysis with Argus	49
2.2.2.1	Case Study 11: Basic Analysis with Argus	50
2.2.2.2	Case study 12: Argus Data Manipulation	51
2.2.3	Argus and Netflow	54
2.2.3.1	Reading Netflow Data	55
2.2.3.2	Processing Netflow Data	56

2.2.3.3	Situation Awareness	58
2.2.3.4	Argus CSV Output	59
2.3	Parsing Security Data	60
2.3.1	Parsing Nmap Output	60
2.3.1.1	Case Study 13: Nmap Output Parsing 1 - Unix Shell	61
2.3.1.2	Case Study 14: Nmap Output Parsing 2 - nmapxmlparser.py	62
2.3.2	Parsing Nessus Output	63
2.3.2.1	Case Study 15: Nessus Output Parsing - tissynbe.py	64
2.4	Advance Security Analysis	68
2.4.1	Picalo	68
2.4.1.1	Usage Considerations	69
2.4.1.2	Data Analysis with Picalo	70
2.4.1.3	Case Study 16: Working with tables in Picalo	72
2.4.2	Reporting	77
2.4.2.1	Logi Report	78
2.4.2.2	Reporting with Logi Report	78
2.4.2.3	Case Study 17: Report Generation with Logi Report Studio	79
2.5	Summary	82
II	PROCESS	83
3	Security Data Carving	85
3.1	Application Layer Packet Classification	85
3.1.1	Layer 7 Monitoring	86
3.2	Statistical Protocol ID	86
3.2.1	SPID Algorithm Overview	87
3.2.2	SPID Algorithm Requirements	88
3.2.3	SPID Data Flow	89
3.2.4	Protocol Models	90
3.2.5	Comparison of Protocol Models	91

3.2.6	SPID Results	92
3.2.7	SPID PoC Implementation	92
3.2.7.1	Case Study 18: SPID Algorithm PoC	93
3.2.8	SPID Algorithm Tuning	96
3.3	Passive Network Analysis	96
3.3.1	Tcpxtract	97
3.3.1.1	Case study 19: Extract Data with Tcpxtract	97
3.3.2	Chaosreader	99
3.3.2.1	Case Study 20: Using Chaosreader	101
3.3.3	NetworkMiner	103
3.3.3.1	Features	104
3.3.3.2	Case Study 21: Processing with NetworkMiner	105
3.3.3.3	Case study 22: TFTPgrab	108
3.4	Summary	109
4	Security Data Exploration	111
4.1	Signature Analysis	111
4.1.1	Port	112
4.1.2	String Match	112
4.1.3	Numerical Properties	112
4.1.4	Behavior and Heuristics	114
4.2	Application Classification	115
4.3	NetWitness Investigator	115
4.3.1	Primer	115
4.3.2	Features	116
4.3.3	Concepts	118
4.3.4	Collection Navigation	119
4.3.4.1	Navigation View	119
4.3.4.2	Navigation Toolbar	119
4.3.4.3	Session List View	120

4.3.4.4	Content View	121
4.4	Investigator Operations	121
4.4.1	Case Study 23: Basic Session Reconstruction	122
4.4.2	Case Study 24: Malware Investigation	123
4.4.3	Case Study 25: More Session Reconstruction	128
4.4.4	Case Study 26: Searching Content	130
4.4.5	Case Study 27: Packet Capture	132
4.4.6	Investigator Options	132
4.5	Summary	133
III	VISUALIZE	135
5	Security Visual Mapping	137
5.1	Visual Perception	138
5.1.1	Memory Limitations	139
5.1.2	Envisioning Information	140
5.2	Information Visualization	142
5.2.1	Visualization Pipeline	142
5.2.2	Visualization Objectives	143
5.2.3	Visualization Benefits	143
5.2.4	Visualization Process	144
5.2.4.1	Data Tables	144
5.2.4.2	Visual Structures	147
5.2.4.3	Views	147
5.2.5	Feedback Loops	148
5.3	Visualization vs Graphing	148
5.4	Visual Data Representation	149
5.4.1	Frequency Charts	151
5.4.1.1	Pie Charts	151
5.4.1.2	Bar Chart	152

5.4.1.3	Pareto Chart	154
5.4.2	Trend Charts	154
5.4.2.1	Line Graphs	155
5.4.2.2	Run Chart	155
5.4.2.3	Control Charts	156
5.4.3	Distribution Charts	157
5.4.3.1	Histograms	157
5.4.3.2	Frequency Polygon	158
5.4.4	Association Charts	158
5.4.4.1	Scatter Plots	159
5.4.4.2	Parallel Coordinates	160
5.4.5	Tree Graphs	163
5.4.6	Link Graphs	164
5.5	Limitations of Visualization	165
5.6	Summary	166
6	Security Visualization Techniques	167
6.1	Evolution	168
6.2	Tools	169
6.3	AfterGlow	169
6.3.1	Parsers in Afterglow	170
6.3.2	Graph Representation	170
6.3.3	Configuration File	170
6.3.4	Case Study 28: Visualization with Afterglow and Graphviz	172
6.3.5	Functions in Afterglow	176
6.4	TNV	178
6.4.1	Case Study 29: Visualization with TNV	179
6.5	Rumint	180
6.5.1	Case Study 30: Parallel Coordinate Plot	182
6.5.2	Case Study 31: Combined Imagery Analysis	183

6.6	EtherApe	183
6.6.1	Case Study 32: Etherape Visualization	183
6.6.2	EtherApe protocols	186
6.6.3	Additional Options	187
6.6.4	Filters in EtherApe	188
6.7	NetGrok	189
6.7.1	Case Study 33: The Drill on NetGrok	189
6.7.1.1	Edge Table	190
6.7.1.2	Graph View	190
6.7.1.3	TreeMap View	191
6.7.2	NetGrok Advantages	194
6.8	RadialNet	194
6.8.1	Case Study 34: Visualization with RadialNet	194
6.9	InetVis	198
6.9.1	InetVis Plotting Scheme	199
6.9.2	Case Study 35: Scatterplot Visualization	200
6.10	GGobi	201
6.10.1	Case Study 36: Interactive GGobi	202
6.11	DAVIX	206
6.11.1	DAVIX Tools	207
6.12	Summary	209
IV	GOVERN	211
7	Security Data Management	213
7.1	Threat Evolution	214
7.2	Shallow Packet Inspection	215
7.3	Deep Packet Inspection	216
7.3.1	DPI Analogy	218
7.3.2	Header Vs Payload	218

7.3.3	DPI Uses	220
7.3.4	DPI Concerns	220
7.3.5	Privacy	221
7.3.6	Net Neutrality	223
7.4	DPI Evasion Techniques	224
7.4.1	Case Study 37: DPI Evasion with OpenSSH	224
7.4.2	Case Study 38: DPI Evasion with Putty	227
7.5	Data Leak Prevention (DLP)	229
7.5.1	DLP Operating Levels	230
7.5.2	Data Leakage Problem	231
7.5.3	Business Case for DLP	232
7.5.4	DLP Workings	233
7.5.5	Overview of DLP Components	234
7.5.6	DLP Phases	235
7.5.7	Extending DLP Solution	236
7.5.8	DLP Use Case Summary	237
7.6	Encryption	238
7.6.1	Data Types	240
7.6.2	Encryption Tools	240
7.7	Data protection	241
7.7.1	Continuous Data Protection	241
7.7.1.1	CDP vs Near Continuous	243
7.7.2	Data Protection Act	243
7.7.3	Formulating a Data Protection Policy	245
7.7.3.1	Organizational Data Protection Policy	245
7.7.3.2	Data protection policy and People	245
7.8	Summary	246
	Appendix A: Glossary	247
	Appendix B: GNU Free Documentation License	261

Appendix C: General Public License version 3 (GPLv3)

271

Part I

CAPTURE

Chapter 1

Security Data Acquisition

As enterprise networks continue to grow geometrically, the challenge for security analysts is effectively that of gaining access to required data so they can conduct security analysis in a timely manner. Ironically, at the same time as this growth, organizations are striving for efficiency by centralizing and consolidating IT. The lack of local IT presence at remote sites complicates the execution of network and security analysis.

Realizing the benefits of security analysis for increasingly complex, widespread enterprises requires a process that accounts for the following key factors:

- **Availability** – The process must include robust security analysis applications and tools that can be deployed throughout the enterprise, but have the ability to be controlled remotely. The time, cost and impacts associated with dispatching technical personnel to identify and resolve anomalies are quite prohibitive and downright unacceptable.
- **Collection** – The employed process must possess the scalability to aggregate network information from various locations, regardless of direct or third-party delivery of requisite information.
- **Depth** – The process must provide insight into activities across all seven layers of the network. That means it must include not only traffic analysis, but also the packet capture and decoding capabilities for network and security packet analysis.
- **Visibility** – The process employed must automatically discover and account for network updates so that accurate analytics can be maintained without requiring technical staff to

be sent to remote sites. It must display data through graphs, charts, reports, and network diagrams, so that identification and problem resolution can quickly and conclusively be achieved anywhere in the enterprise.

When armed with these information, organizations can see network security issues and application degradations before they become issues. Assurance and audit tasks involve frequency analysis of the events registered in the security and application logs. The traditional approach to the information contained there is mainly based on text-format representation of the data. Bandwidth allocation and capacity management can be tuned a lot more effectively. Security, availability, quality of service (QoS) metrics and the mean time to repair MTTR problems can be subsequently improved when they arise through accurate detection and rapid isolation.

1.1 Sources of Security Data

It goes without saying that data is required for analysis. Before we can start any form of discussion on security analysis or data visualization, we need to source out security data. It is therefore, pertinent for us to start by taking a look at data categories because all security data cannot be viewed and analyzed in the same manner. There is data resident on all layers of the TCP/IP stack and knowing what form of security data can be extracted on what layer will help us narrow down the analysis type that should take place on that layer. *(It is very important to know that we dwell heavily on the structure of TCP/IP reference model in relation to security analysis throughout this book)*

The individual vulnerabilities of hosts on a network can be combined by an attacker to gain access that would not be possible if the hosts were not interconnected. Most tools available tend to report vulnerabilities in isolation and in the context of individual hosts in a network. Security analysis, however, extends this by searching for strings and series of interrelated issues, distributed among the various hosts on the network. Previous efforts did not take into account a realistic representation of network connectivity. These models were enough to exhibit the usefulness of the model checking approach but would not be sufficient to analyze real-world network security issues. This section presents network connectivity at the different levels of the TCP/IP stack appropriate for use in a model checker. With this improvement, it is possible to represent networks including common network security devices such as switches, routers, packet filters, and firewalls.

1.2 TCP/IP Reference Model

While an in-depth discussion of the TCP/IP reference model is not the focus of this book, it is worthwhile taking a brief look at it at least from a security perspective. TCP/IP model consists of four layers that represent a functional division of the tasks required for communications on a network¹. It provides end-to-end connectivity by specifying how data should be formatted, addressed, transmitted, routed and received at the destination. See Figure 1.1

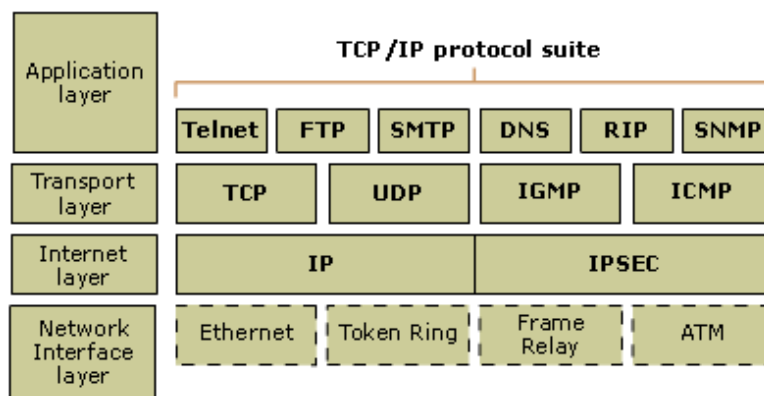


Figure 1.1:

As mentioned earlier, the TCP/IP reference model consists of four layers: *the network interface layer, the Internet layer, the transport layer, and the application layer.*

In the following sections, we describe the functions of each layer in more detail especially as it relates to data gathering, starting with the network interface layer and working our way up to the application layer.

1.2.1 Network Interface Layer

The network interface layer is the lowest layer in the TCP/IP reference model. This layer consists of the protocols that the computer uses to send data to the other devices that are attached to the network. The functions of this layer include:

¹In the TCP/IP model the physical layer is not covered because the data link layer is considered the point at which the interface occurs between the TCP/IP stack and the underlying networking hardware.

- Definition of how to use the network to transmit a *frame*, which is the data unit passed across the physical connection.
- Transmission of data between the computer and the physical network.
- Sending data back and forth between two devices on the same network. To deliver data on the local network, the network layer protocols use the physical addresses of the nodes on the network. A physical address is hard coded or “burned” in the network adapter card of a computer or other device.

The Ethernet switch is a device that operates at this layer. Unlike higher level protocols, the network interface layer protocols must understand the details of the underlying physical network, such as the packet structure, maximum frame size, and the physical addressing scheme in use.

1.2.2 Internet Layer

In the TCP/IP reference model, the layer above the network interface layer is known as the Internet layer. This is the layer tasked with message routing through the Internet or inter networks. The *router* is responsible for routing messages between networks. It is a device with two interfaces which accepts network packets from one interface and passes those *packets* to a different network via the second interface. Sometimes they are referred to as *gateways*.

The internetwork layer protocols provide a *packet* network service. *Packets* consists of a header, data, and a trailer. The header contains information such as the source address, the destination address and security labels. Trailers contain a checksum value, which is used to ensure integrity of the packet across the internetwork. The communicating entities—which can be computers, operating systems, programs, processes, or people—that use the *packet* must specify the destination address (using control information) and the data for each message to be transmitted. The Internet layer protocols package the message in a packet and send it off.

A *packet* does not support any concept of a session or connection. The entire communication is connectionless. If there is any need for any connection oriented communication, the protocols in the transport layer will have that responsibility. If the receiving end detects a transmission error, it simply ignores (or drops) the packet without notifying the receiving higher-layer entity.

1.2.3 Transport Layer

The protocol layer just above the Internet layer is the host-to-host transport layer. It is responsible for providing end-to-end data integrity and provides a highly reliable communication service for entities that want to carry out an extended two-way conversations.

In addition to the usual transmit and receive functions, the transport layer uses open and close commands to initiate and terminate the connection. This layer accepts input as a stream of characters and it outputs returns information also as a stream.

The service implements the virtual circuit connection concept. A connection is the state of the transport layer between the time that an open command is accepted by the receiving computer and the time that the close command is issued by either computer.

1.2.4 Application Layer

The topmost layer in the TCP/IP reference model is the application layer. This layer provides functions for users and their programs. It provides the services that user applications use to communicate over the network, and it is the layer in which user-access processes reside.

This layer includes all applications protocols that use the transport protocols to deliver data. Other functions such as compression, decompression, encryption and decryption also take place at the application layer.

The application layer is also responsible for managing the connections or sessions between cooperating applications. In the TCP/IP protocol hierarchy, sessions are not identifiable as a separate layer, and these functions are performed by the transport layer. For applications to be able to exchange information, they must agree about how the information is represented.

1.3 Security Data Capture

Having touched on the features of each of the respective layers in the foregoing section, how do we capture security data from them? We begin our discussion with the lowest layer.

1.4 Network Interface Layer

For any security data to be captured at the network interface layer, we need to be able to intercept and display TCP/IP and other packets being transmitted or received over the network. In the sections that follow, we will be discussing Tcpdump and Wireshark - the two most versatile packet capture tools available for the Linux operating system. It is important to be aware that capturing packets is a bit of a double edged sword. The experiences can range from "it simply works" to "strange problems". To avoid any unpleasant surprises, the following gives you a guide through the capture process².

- **Setup the machine's configuration to allow capture:** Make sure that you have sufficient privileges to capture packets, e.g. root or Administrator privileges. Also make sure the capture computer's time and time zone settings are correct, so that the time stamps captured are meaningful
- **Capture traffic to and from localhost:** Capturing localhost traffic is the easiest way to successfully capture traffic. The traffic to and from localhost is obviously available independent of network topology.
- **Capture traffic destined for other machines on the network:** Make sure the capture is done from a location in the network where all relevant traffic will pass through by selecting the right place in your network topology in order to get the required network traffic.

The following are two illustrations of how networks can be setup to capture packets that are destined for other machines

1.4.1 Switch Port Mirroring

Most Ethernet switches (usually called "managed switches") have a monitor mode. This monitor mode can dedicate a port to connect your capturing device. It's sometimes called

²Ensure that you are allowed to capture packets from the network you are working on. The security policy might prevent you from capturing on the network you're using. Obtain permission if need be.

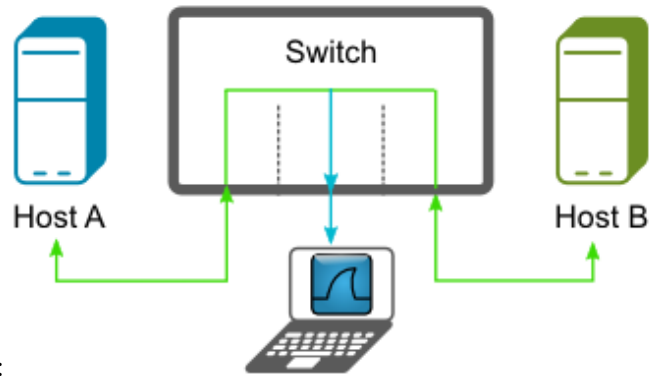


Figure 1.2:

port mirroring, port monitoring, or Switched Port Analyzer or 'SPAN' (Cisco). As illustrated in Figure 1.2 Using the switch management, you can select both the monitoring port and assign a specific port you wish to monitor. Actual procedures, however, vary between different switch models.

1.4.2 Man-in-the-middle

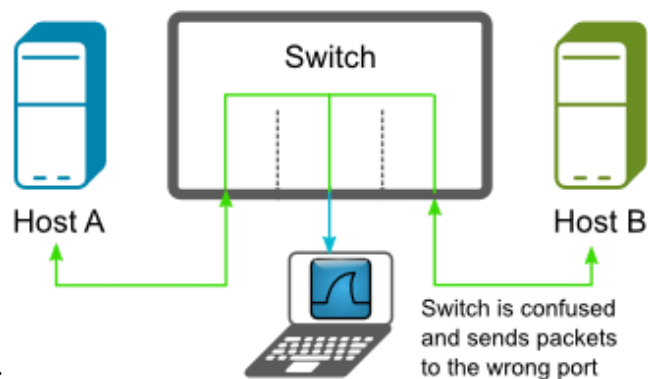


Figure 1.3:

The diagram (Figure 1.3) shows a typical man-in-the-middle attack. To capture packets going between two computers on a switched network, you can employ an ARP poisoning technique called Man-in-the-middle MITM attack. This type of attack will make the two computers believe that your MAC address is the MAC address of the other machine. This will in turn have the effect of making the switch route all of their traffic to your computer where it can then be sniffed and then passed through to its eventual destination in a very transparent manner. This type of attack can cause disruptions on some switches and networks so it should be used with due care³.

Whilst I will not discuss the set up process for port mirroring (as this is dependent on vendor, make and model), I will, however, walk through the process of setting up a Man-in-the-middle attack using ARP poisoning to redirect traffic through our capture device.

1.4.3 Packet Redirection through ARP Poisoning

There is nothing extraordinary about ARP Poisoning. The ARP protocol is a layer 3 protocol used to translate IP addresses to physical network card addresses or MAC addresses. When a device tries to access a network resource, it first sends requests to other devices asking for the associated MAC address to IP that it wants to communicate with. The caller will keep the corresponding IP to MAC association in its cache, the ARP cache, to speed up new connections to the same IP address.

The attack comes when a machine requests the other ones to locate the MAC address associated with an IP address. The pirate (in our case the capture machine) will answer to the caller with fake packets saying that the IP address is associated to its own MAC address and in this way, will "short-cut" the real IP - MAC association answer coming from another host. This attack is referred to as ARP poisoning or ARP spoofing and is possible only if the capture machine and the victims are on the same broadcast domain which is defined on the host by an IP address and a subnet mask as shown in Figure 1.4. Enter Ettercap.

1.4.3.1 Ettercap⁴

Ettercap is exactly the tool we need for the task at hand. It is a suite for man in the middle attacks on switched networks. It features the sniffing of live connections, content filtering on

³Please do not try this on any network other than the one you have authority over.

⁴[HTTP://ettercap.sourceforge.net](http://ettercap.sourceforge.net)

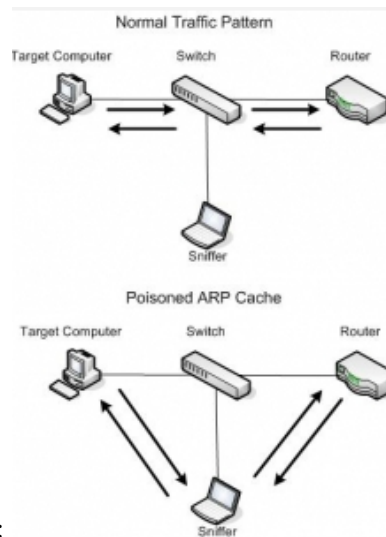


Figure 1.4:

the fly and a lot of other interesting tricks. It supports active and passive dissection of many protocols (even ciphered ones) and includes quite a number of features for network and host analysis.

We will use the illustration below where an ARP poisoning attack is carried out between the machine with IP 192.168.1.2 (client) and another with IP 192.168.1.1 (server) on a local network. After the ARP poisoning attack, The capture machine becomes the MITM. The following should be noted about the machine behaviour of Ettercap:

- Every time Ettercap starts, it disables IP forwarding in the Linux kernel and begins to forward packets itself.
- It can cause a significant reduction in network throughput between the hosts because of the machine's processing time.
- Ettercap needs system level (root) privileges to open the network sockets. It should be run in a directory with the right permissions because it has to write to log files,.

Ettercap Installation

To install ettercap is quite straight forward with *yum* on Fedora Core 10 by running the following command (without the #⁵) as root

```
# yum -y install ettercap6
```

and that's it. Ettercap is installed and can be launched from the command console by typing the following also as root

1.4.3.2 Case Study 1: ARP Poisoning with Ettercap

```
# ettercap -G &
```

Note: The “&” symbol is used to launch any unix application or command in the background.

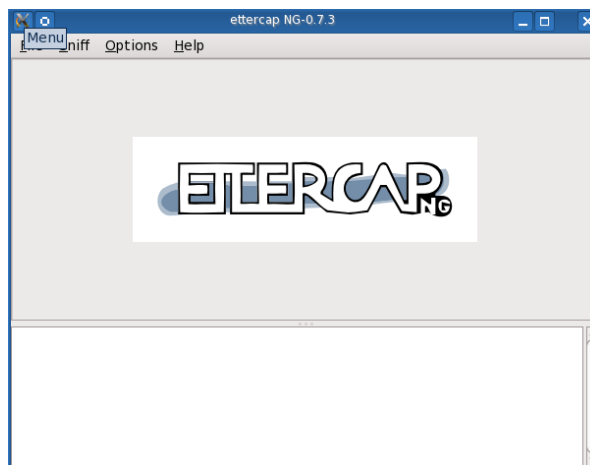


Figure 1.5:

⁵This just signifies that you are working as the user root.

⁶Check `yum -help` for additional options

You are immediately shown the opening GUI screen Figure 1.5 on page 12 where there are four tabs. *File, Sniff, Options and Help*. Follow the following steps to initiate your MITM Attack using ARP Poisoning. I assume your Ethernet interface is eth0.

- Select the **sniff mode -> Unified Sniffing**

Select your network device (eth0)

- Select **Host -> Scan for hosts**

The network range scanned will be determined by the IP settings of the interface you have just chosen in the previous step

- Click on **Hosts-> Hosts list**

To monitor the MAC & IP addresses of the hosts inside your subnet.

Choose the machines to poison. We can choose only to ARP poison the windows machine 192.168.1.2 and the gateway 192.168.1.1.

- Highlight the IP 192.168.1.1 and click on the "**target 1**" button.
- Highlight the IP 192.168.1.2 and click on the "**target 2**" button.

If you don't select any machine as target, all the machines within the subnet will be ARP poisoned Figure 1.6

Start the ARP poisoning

- **Mitm -> Arp poisoning**

Start the sniffer

- **Start -> Start sniffing**

And that's it. Now all network packets will be re-routed through the capture machine acting as Man-in-the-middle. This can be verified with one of the numerous ettercap plugins.

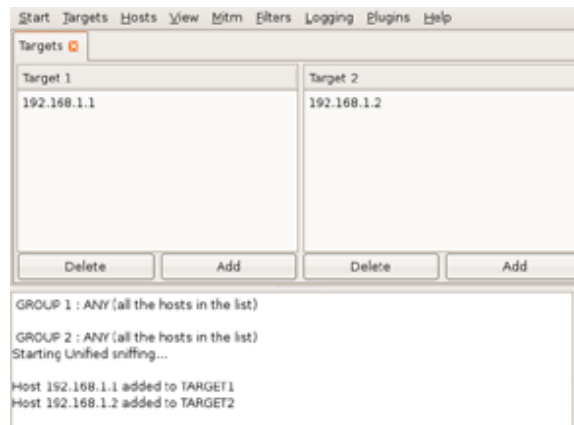


Figure 1.6:

Ettercap Plugins

We will use here the Ettercap plugin called `chk_poison` to test for the success of the ARP poisoning attack. To start the plugin, you need to activate the `chk_poison` plugin in the Ettercap graphical interface.

- **Plugins -> Manage the plugins**
- Double click on **chk_poison**

If the ARP Poisoning is successful, you will be informed in the second pane

Ettercap is by no means the only application used to accomplish an ARP Poisoning. There are indeed other applications⁷ but it is by far the most effective and intuitive tool that I have used.

1.4.4 Network Packet Sniffing

We have only set the stage for the real packet capture. In this section we will be taking a critical look at the packet capture methodology as well as the tools available to carry it out. I have to quickly point out that whilst ettercap has in-built capability to sniff packets across a switched segment, it does not compare with other tools like `Tcpdump` and `Wireshark` in this regard. It's potency is inherent in being able to perform poisoning attacks.

⁷<http://www.monkey.org/~dugsong/dsniff/>

In its simplest form, a packet sniffer captures all of the packets of data that pass through a given network interface. Essentially, the packet sniffer would only capture packets that were intended for the machine in question. However, if placed into promiscuous mode, the packet sniffer is also capable of capturing ALL packets traversing the network regardless of destination. Packet sniffers work by capturing "binary" data passing through the network, most if not all decent sniffers "decode" and try to present this data in a human readable format. The analysis that takes place by and large vary, some are straight forward, just breaking down the *packet* information while others are a bit more complex giving more detailed information about what it sees on the packet.

1.4.4.1 Tcpcap⁸

Tcpcap is a simple command line application that is used to dump traffic on a network. It is available on most Unix-like operating systems and uses the *libpcap* library to capture packets (packet capture library is discussed in Chapter 2). There is a port of Tcpcap called Windump that runs on the Windows operating system but needless to say, I have never used it. I won't recommend the use of the Windows operating system for any sort of packet capture even though it can be used for off-line analysis.

Now that our machine is the man-in-the-middle, we need to set up Tcpcap to capture the packets as they traverse the network and save the packet capture to a file. Again our capture interface is *eth0*.

Tcpcap Installation

To install tcpcap:

```
# yum -y install tcpcap
```

This will install tcpcap with associated dependencies from the yum repos

```
# tcpcap -nni eth0 -vv -w capture.pcap9
```

⁸<http://www.tcpcap.org>

⁹check the man page of tcpcap for options

After leaving the command to run for a few seconds, it can be terminated by using [ctrl-c]. To get an initial view of the pcap file, it is passed as an argument to the *capinfos* command thus;

```
# capinfos capture.pcap
```

which yields the following output

```
File name: capture.pcap
File type: Wireshark/tcpdump/... - libpcap
File encapsulation: Ethernet
Number of packets: 31
File size: 2973 bytes
Data size: 6589 bytes
Capture duration: 3.160192 seconds
Start time: Mon Sep 14 23:06:50 2009
End time: Mon Sep 14 23:06:53 2009
Data rate: 2085.00 bytes/s
Data rate: 16680.00 bits/s
Average packet size: 212.55 bytes
```

For single 3 second file download, it comes in at approximately 4.0Kb and contains 31 packets. If we had captured for about an hour, we'll have had a lot more. It is quite tedious to go through every single packet if it is not important. What if we are not interested in knowing the payload in the packet but the connection summary such as number of packets sent by one host to another or number of bytes transferred in the connection or yet duration of the connection? Lets sift through some low level static analysis.

1.4.4.2 Case Study 2: Packet Capture with Tcpcap

Machine 1 - 192.168.1.2

Machine 2 - 4.2.2.2

Machine 1 queries the DNS server on Machine 2

To get the number of packets sent by Machine 1 to Machine 2

```
# tcpdump -ttttnr capture.pcap ip src 10.0.1.1 | wc -l
reading from file capture.pcap, link-type EN10MB (Ethernet)
16
```

To get the number of packets sent by Machine 2 to Machine 1

```
# tcpdump -ttttnr http-download.pcap ip src 4.2.2.2 | wc -l
reading from file capture.pcap, link-type EN10MB (Ethernet)
27
```

Now what if we want to know the number of bytes sent by Machine 1 to Machine 2 and vice-versa? This would be quite exhaustive if we have to view those packets and count. Herein lies the limitations of this low level type of static packet analysis.

Lastly, this I find very useful in determining the noisiest machine on corporate networks:

```
# tcpdump -tnn -c 20000 -i eth0 | awk -F "." '{print $1"."$2"."$3"."$4}' \
| sort | uniq -c | sort -nr | awk ' $1 > 100 '
```

The beauty of the Unix CLI.¹⁰

1.4.4.3 Wireshark¹¹

Wireshark is the world's foremost network protocol analyzer, and is the de facto standard across many industries and educational institutions. It is very similar to tcpdump, but has a GUI front-end, and many more information sorting and filtering options. It also allows the user to see all traffic being passed over ethernet networks by putting the network interface into promiscuous mode¹².

¹⁰Note that the command above should be on a continuous line

¹¹<http://www.wireshark.org>

¹²A mode of operation in which every data packet transmitted can be received and read by a network adapter

Wireshark is a tool that greatly lends itself to deep understanding of packet analysis by allowing you to 'see' the packets and protocols in action and by 'fiddling' around with them – observing the sequence of messages exchanged between two entities, digging down into the details of packet operation, and causing protocols to perform certain actions and then observing these actions and their consequences.

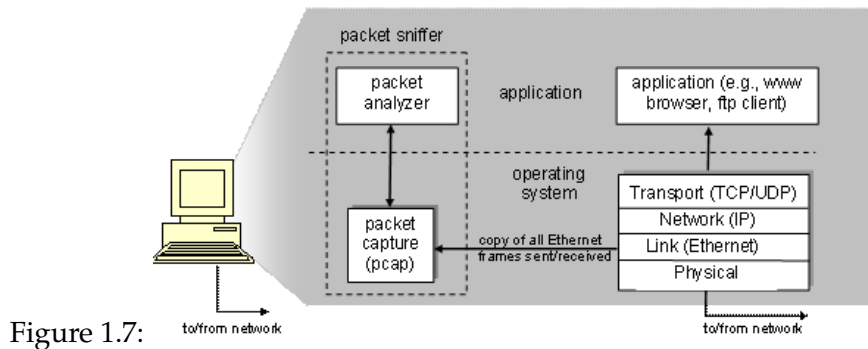


Figure 1.7:

Figure 1.7 shows the structure of a packet sniffer. The protocols and applications are represented on the right. The packet sniffer, shown within the dashed rectangle is made up of two parts. The packet capture library *pcap* receives a copy of every link-layer frame that is sent from or received by the capture computer. Messages exchanged by higher layer protocols such as TCP, UDP or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as the ethernet cable. The physical media is assumed to be ethernet, and so all upper layer protocols are eventually encapsulated within the ethernet frame. Capturing all link-layer frames thus gives all messages sent or received by all protocols and executing applications.

Wireshark Installation

Installing wireshark is fast and easy from the yum repositories. Simply type the following as root

```
# yum -y install wireshark-gnome
```

This should sort out your dependencies. If you simply use `yum -y install wireshark` on Fedora Core, you will only have text mode Tshark and not the Wireshark GUI.

To launch it simply type:

```
# wireshark &
```

The Wireshark GUI in Figure 1.8 will be displayed even though no data will be initially displayed in any of the three panes.

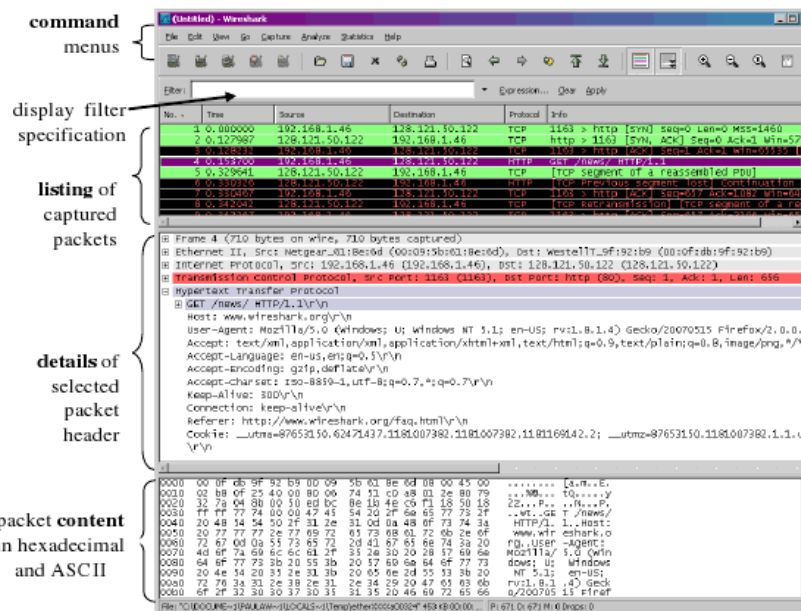


Figure 1.8:

The Wireshark interface has five noteworthy components:

Command menus are standard pull down menus located at the top of the window. Of interest obviously are the **File** and **Capture** menus. The **File** menu gives us the ability to save captured packet or open a file containing a previously captured packet. The **Capture** menu allows us to us to initiate packet capture.

Packet-listing window displays a one-line summary for each packet captured, including the packet serial number, the time of packet capture, the source and destination address, the protocol type, and protocol-specific information contained within the packet. The packet listing can be sorted in any order by clicking on a column name. The protocol type field lists the highest layer protocol that sent or received this packet.

Packet-header details window provides details about the packet selected or highlighted in the packet listing window. These include details about the Ethernet frame and IP datagram that contains this packet. The number of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the right-pointing or down-pointing arrow-head to the left of the Ethernet frame or IP datagram line in the packet details window. The details over which transport layer protocol (TCP or UDP) the packet was carried over is also be displayed, which can similarly be expanded or minimized.

Packet-contents window displays the entire contents of the frame captured, in both hexadecimal and ASCII formats.

Towards the top of the Wireshark graphical user interface, is the **packet display filter field**, This can be used as a display filter to narrow down packet analysis of a specific type.

1.4.4.4 Case Study 3: Packet Capture with Wireshark

Capturing packets with Wireshark is simple, we assume that our ethernet interface is *eth0*:

- Select the **Capture** pull-down menu then select Options. This will cause the **Wireshark: Capture Options** window to be displayed, as shown in Figure(1.9)
- Use majority of the default values in this window. The network interface is eth0 (default) . After selecting the network interface, pick two other options - **updating the list of packet in real time** and **automatic scrolling of live capture** (marked by the red square in Figure1.9).
- Click **Start**. Packet capture will now begin - all packets being sent/received from on the network will now be captured. You can allow it run for a couple of minutes before clicking Stop.

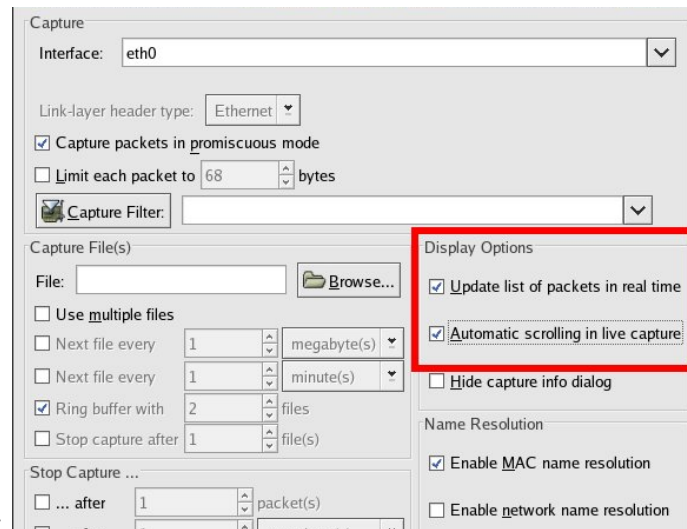


Figure 1.9:

1.4.4.5 Extracting and Preserving Security Data

Having captured network data, it is highly unlikely as analysts, that we will want to start our online analysis there and then. In this case, we need some way to extract and preserve the data. Wireshark provides us with the tools necessary to accomplish these.

□ File -> Save As

This gives us the option of extracting the captured data in any one of the several packet capture formats available within Wireshark. You can make a practice of saving your captures in libpcap, tcpdump (*.pcap, *.cap, *.dmp) formats. The list is quite exhaustive and you can consult the Wireshark manual for more.

□ File -> Export

I use the export functionality of Wireshark quite a bit in daily packet captures. Figure 1.10 shows that packet captures can be preserved in any one of several formats available, *text*, *postscript*, *PSML - XML* and *PDML - XML* file formats. By far, the best of them (and my view is highly subjective), is the CSV file format. In fact this is where I think Wireshark excels.

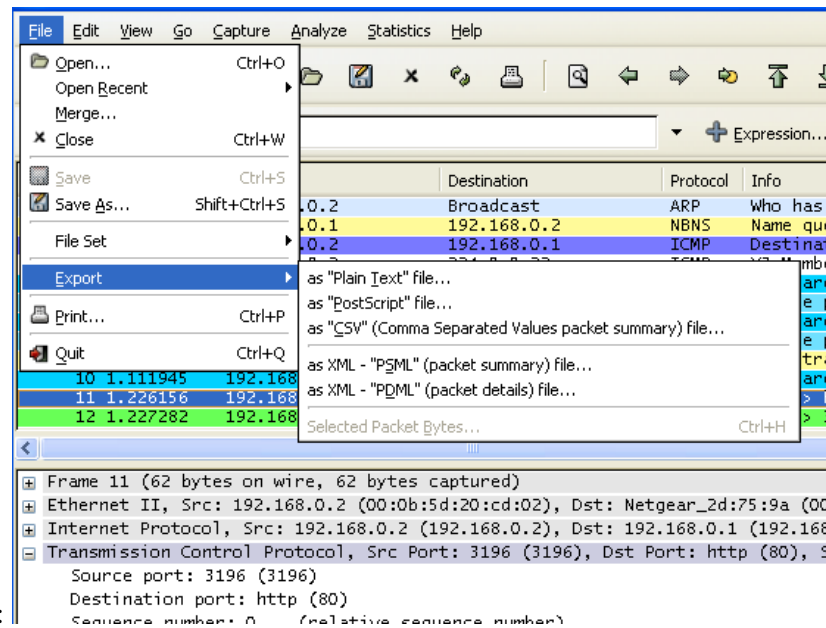


Figure 1.10:

So now we have our capture files. We will see in subsequent chapters of this book how these packet captures can be used for detailed off-line security analysis and visualization.

Another option which may come in handy on non GUI boxes is the *dumppcap* utility that comes with Wireshark. To capture packets with the *dumppcap* utility run the following

```
# dumppcap -i eth0 -a duration:300 -w capture.pcap
```

This will capture packets for a duration of 300 seconds (5 minutes) and write output to *capture.pcap*¹³

We will conduct a more examination of Wireshark in the next chapter: Protocol Analysis

¹³check the man page of dumppcap for more options

1.5 Internet and Transport Layers

From the foregoing discussion in Section 1.2, we observe that capturing security data on higher layers such as Internet and Transport will be quite different from the Network Interface Layer. We have gradually moved from low level packet captures now to higher level data and information gathering techniques used in security assessment and penetration testing.

We are combining these two layers as it will aid us in performing our security analysis in more depth rather than deal with each on its own merit. However, we do not need any type of elaborate setup on these layers as we did the last. We will take a rather different approach here as we can be a bit more flexible in our data gathering approach.

We are not going to delve into the gory details of the structure of the TCP, UDP, IP or ICMP packets neither are we interested in such concepts as IP addressing scheme, the TCP sliding window and the 3-way handshake - mind you, I assume you are conversant with these concepts fairly well, but if not you can easily pick up a networking book to get up to speed¹⁴. Having said that, I will briefly touch on two fundamental concepts that will enable us understand the process of Internet information gathering: *ports* and *sockets*

1.5.1 Ports

Simply put, a port is a number, a 16 bit number. It is essentially a channel of communication for computers in networks(1.11). Therefore, it means that there are up to 65,535 possible ports on every single machine on the network. These ports are however logical and not physical. In addition, they operate based on connection requests (Figure 1.11) and replies (Figure 1.12) and a single port can exist in one of three states - *open*, *closed* or *filtered*.

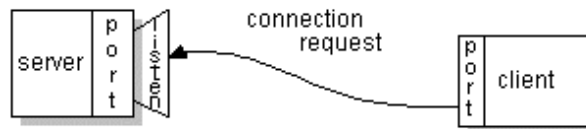


Figure 1.11:

¹⁴A good book which comes highly recommended is TCP/IP Illustrated Volume 1 by W. Richard Stevens

Because port numbers can range from 0 to 65535, they have been further divided into three main categories:

- 0 to 1023 are called *well known ports*, meaning they are reserved for special services like SSH (22), TELNET (23), SMTP (25), HTTP (80), POP3 (110), IMAP (143), etc.
- 1024 to 49151 are known as *registered ports*, meaning they are registered for specific protocols by software companies.
- 49152 to 65536 are *dynamic and/or private ports*, meaning that anyone can use these as required. I like to refer to them as '*ephemeral ports*'

Port numbers are managed by the IANA (Internet Assigned Numbers Authority)¹⁵.

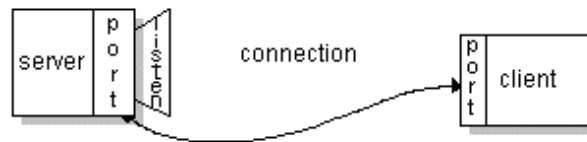


Figure 1.12:

1.5.2 Socket

In its simplest form, a socket is one end-point of a two-way communication link between two programs running on the network. Because it is an end-to-end connection between a client and a server, it identifies four variables: the source and destination IP address and the source and destination port number (Figure 1.13). The client and the server must also agree on a protocol to use for communication over the socket. In fact, if you have ever used a browser, you have communicated via a socket.

`http://www.google.com:80` is an example of a socket connection. You normally don't need to add the `:80` because web servers already are configured to listen on this port.

¹⁵<http://www.iana.org>

Sockets come primarily in two flavors: active and passive. An active socket is connected to another remote active socket via an open data connection. Closing the connection destroys both active sockets at each end point. A passive socket, however, is not connected, but rather sits and waits for an incoming connection, which will then spawn a new active socket.

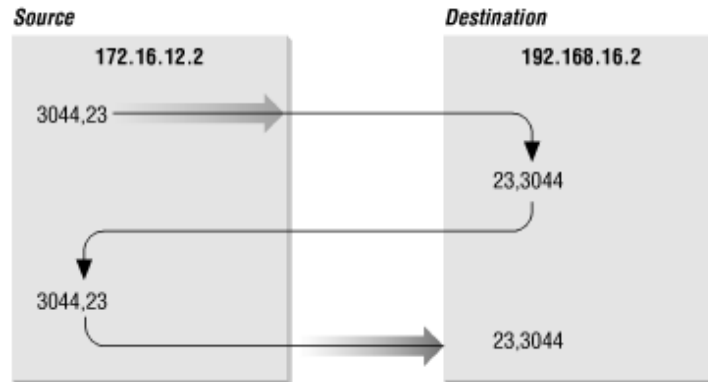


Figure 1.13:

Though there is a close relationship between a port and a socket, they are different. The relationship is that of many-to-one. Each port can have a single passive socket, waiting for incoming connections and multiple active sockets each corresponding to an open connection on the port.

The question is, how then do we actively acquire security related data? Enter port scanning...

1.5.3 Port Scanning

Port scanning is the method for discovering exploitable communication channels and a technique used to determine what ports or protocol abstraction of a host are listening for in-bound connections. The idea is to probe and identify as many listeners as possible, and keep track of the ones that are responsive to a particular need. Scanning machines typically involves a measure of brute force attempts. We send dozens of packets for various protocols, and we deduce which services are listening from the responses or lack of.

Over time, a number of tools and techniques have been developed for surveying the protocols and ports on which a target machine is listening. They all offer different features

and benefits. We will be discussing two highly rated free and open source tools and techniques and then we attempt to do some comparative analysis:¹⁶ *Unicornscan* and *Nmap*¹⁷

1.5.3.1 Unicornscan¹⁸

According to its website, Unicornscan is a new information gathering and correlation engine built for and by members of the security research and testing communities. It was designed to provide an engine that is Scalable, Accurate, Flexible, and Efficient. It is intended to provide a researcher a superior interface for introducing a stimulus into and measuring a response from a TCP/IP enabled device or network. Some of its benefits amongst others include:

- Asynchronous stateless TCP scanning with all variations of TCP Flags.
- Active and Passive remote OS, application, and component identification by analyzing responses.
- PCAP file logging and filtering
- Relational database output

As you can see, I have deliberately highlighted the features that have come in handy in a lot of the security analysis and assessment consulting engagements I have been involved in. Off-line analysis will be greatly enhanced because it can dump scan results in *pcap* file format as well as output to an RDBMS - I haven't tested its RDBMS feature yet, partly because it uses postgresql as its DBMS, but its fairly trivial to output to a MySQL DB once you have the *pcap* file (This will be dealt with in the next chapter).

Unicornscan Installation

As usual installation is simple. I simply downloaded the rpm file from the website and ran (as root)

```
# rpm -ivh unicornscan-0.4.7-4.fc8.i386.rpm
```

¹⁶These two I use regularly, however check here <http://sectools.org> for more tools

¹⁷nmap is pretty much industry standard now.

¹⁸<http://www.unicornscan.org>

You might need to install the postgresql libraries to solve dependency issues. That can be done with yum:

```
# yum -y install postgresql-libs
```

And there you go. It's installed.

Usage

Unicornscan must be run as root. Shortly after execution, Unicornscan will change effective UID to nobody inside of a chrooted environment. This is done to limit inherent risks in taking arbitrary user input off of the wire.

1.5.3.2 Case Study 4: Basic Scan with Unicornscan

```
# unicornscan inverse.com.ng
TCP open      ftp[ 21]      from 76.74.146.218  ttl 40
TCP open      ssh[ 22]      from 76.74.146.218  ttl 40
TCP open      smtp[ 25]     from 76.74.146.218  ttl 40
TCP open      http[ 80]     from 76.74.146.218  ttl 40
TCP open      pop3[ 110]    from 76.74.146.218  ttl 40
TCP open      imap[ 143]    from 76.74.146.218  ttl 40
TCP open      https[ 443]   from 76.74.146.218  ttl 40
TCP open      smtps[ 465]   from 76.74.146.218  ttl 40
TCP open      imaps[ 993]   from 76.74.146.218  ttl 40
TCP open      pop3s[ 995]   from 76.74.146.218  ttl 40
```

In this example, all we specified is the name of the server to scan. The hostname was resolved to the address of 76.74.146.218. A TCP SYN (-mTS, which is the default scan mode) scan was sent to that IP on the Unicornscan Quick Ports as defined in the `/etc/unicornscan/unicorn.conf` file. Ports that respond with a SYN/ACK return as open. So here we can see that the *ftp*, *ssh*, *http*, *pop3*, *imap*, *https*, *smtps*, *imaps* and *pop3s* ports all return as open.

1.5.3.3 Case Study 5: Advance Scan with Unicornscan

This next illustration shows the output of an advance unicorn scan that outputs directly to a pcap file *scan.pcap* (represented with the *-w* option).

```
# unicornscan -i eth0 -r1000 -I -v -w scan.pcap -mT
www.yahoo.com/29:80,44319
TCP open 87.248.113.11:80 ttl 49
TCP open 87.248.113.10:80 ttl 50
TCP open 87.248.113.9:80 ttl 49
TCP open 87.248.113.8:80 ttl 49
TCP open 87.248.113.15:80 ttl 114
TCP open 87.248.113.14:80 ttl 114
TCP open 87.248.113.13:80 ttl 49
TCP open 87.248.113.12:80 ttl 49
TCP open 87.248.113.12:443 ttl 49
TCP open 87.248.113.13:443 ttl 49
TCP open 87.248.113.14:443 ttl 114
TCP open 87.248.113.15:443 ttl 114
TCP open 87.248.113.8:443 ttl 49
TCP open 87.248.113.9:443 ttl 49
TCP open 87.248.113.10:443 ttl 50
TCP open 87.248.113.11:443 ttl 49
listener statistics 18 packets received 0 packets dropped and 0 interface drops
TCP open      http[ 80]      from 87.248.113.8  ttl 49
TCP open      https[ 443]     from 87.248.113.8  ttl 49
TCP open      http[ 80]      from 87.248.113.9  ttl 49
TCP open      https[ 443]     from 87.248.113.9  ttl 49
TCP open      http[ 80]      from 87.248.113.10 ttl 50
TCP open      https[ 443]     from 87.248.113.10 ttl 50
TCP open      http[ 80]      from 87.248.113.11 ttl 49
TCP open      https[ 443]     from 87.248.113.11 ttl 49
TCP open      http[ 80]      from 87.248.113.12 ttl 49
TCP open      https[ 443]     from 87.248.113.12 ttl 49
TCP open      http[ 80]      from 87.248.113.13 ttl 49
```

¹⁹Check the man page for options used

```
TCP open      https[ 443]      from 87.248.113.13  ttl 49
TCP open      http[ 80]       from 87.248.113.14  ttl 114
TCP open      https[ 443]     from 87.248.113.14  ttl 114
TCP open      http[ 80]       from 87.248.113.15  ttl 114
TCP open      https[ 443]     from 87.248.113.15  ttl 114
```

The result of this scan is written to the *scan.pcap* file in addition to stdout (terminal). You can view the *scan.pcap* file with Wireshark.

This command resolves the name *www.yahoo.com* to 87.248.113.11 and sends 1000 packets per second (pps) to the IP range of 87.248.113.14/29 (87.248.113.8-16) on TCP ports 80 and 443. Figure 1.14 is the view of the scan in Wireshark GUI

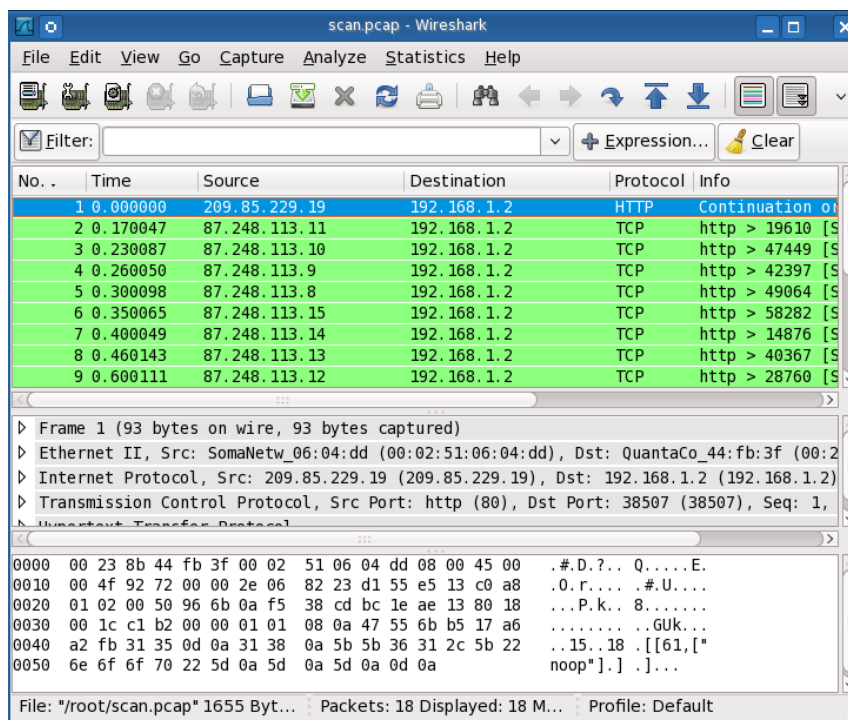


Figure 1.14:

1.5.3.4 Nmap²⁰

No one describes nmap better than its developer “Fyodor”. Nmap (“Network Mapper”) is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.

The output from Nmap is a list of scanned targets, with supplemental information on each depending on the options used. Key among that information is the “interesting ports table”. That table lists the port number and protocol, service name, and state. The state is either *open*, *filtered*, *closed*, or *unfiltered*.

Open.

means that an application on the target machine is listening for connections/packets on that port.

Filtered.

means that a firewall, filter, or other network obstacle is blocking the port so that Nmap cannot tell whether it is open or closed.

Closed.

ports have no application listening on them, though they could open up at any time.

Ports are classified as unfiltered when they are responsive to Nmap’s probes, but Nmap cannot determine whether they are open or closed. Nmap reports the state combinations **open | filtered** and **closed | filtered** when it cannot determine which of the two states describe a port. The port table may also include software version details when version detection has been requested. When an IP protocol scan is requested, Nmap provides information on supported IP protocols rather than listening ports.

²⁰<http://nmap.org>

In addition to the interesting ports table, Nmap can provide further information on targets, including reverse DNS names, operating system guesses, device types, and MAC addresses.

That pretty much sums it up.

Nmap Installation

At the time of writing this book, nmap v5.00 was just released and i decided to take it for a spin. Whilst your operating system may have shipped with an earlier version of nmap, i suggest you download²¹ or install from the site the new one.

```
# rpm -Uvh http://nmap.org/dist/nmap-5.00-1.i386.rpm
# rpm -Uvh http://nmap.org/dist/ncat-5.00-1.i386.rpm
# rpm -Uvh http://nmap.org/dist/zenmap-5.00-1.noarch.rpm
```

As you may have observed, we have installed two more applications - *ncat* and *zenmap*. A brief explanation is in order here.

Ncat acts like the unix *cat* command. It does to network sockets what *cat* does to UNIX files - redirection. In other words, it is used mainly for socket redirection and stands for network (n) concatenation (cat). Ncat is a nifty utility that reads and write data across a network from the UNIX command line. Furthermore, it uses both TCP and UDP for communication and is designed to be a reliable back-end tool to instantly provide network connectivity to other applications and users. Ncat not only works with IPv4 and IPv6 but it provides the user with a virtually limitless number of potential uses.

Zenmap is a GUI frontend and results viewer for Nmap. Need I say more.

Even as this book is predicated on security assessment, we will not go into the detailed workings of *ncat* and *zenmap*. If you need more information however, you can consult the excellent book on nmap²² written by none other than its developer. It doesn't get any better than that does it?

²¹<http://nmap.org>

²²<http://nmap.org/book/>

Usage

A typical Nmap scan is shown in Case Study 6. The only Nmap arguments used in this example are *-A*, to enable OS and version detection, script scanning, and traceroute; *-T4* for faster execution;

1.5.3.5 Case Study 6: Basic Nmap Scan

```
# nmap -A -T4 inverse.com.ng
Starting Nmap 5.00 ( http://nmap.org ) at 2009-09-16 15:08 WAT
Interesting ports on venus.ultracrest.com (76.74.146.218):
Not shown: 988 filtered ports
PORT STATE SERVICE VERSION
20/tcp closed ftp-data
21/tcp open  ftp PureFTPd
22/tcp closed  ssh
53/tcp open  domain ISC BIND 9.2.4
80/tcp open  http Apache httpd 2.0.63 ((Unix)
mod_ssl/2.0.63 OpenSSL/0.9.7a mod_auth_passthrough/2.1 mod_bwlimited/1.4
FrontPage/5.0.2.2635 PHP/5.2.5)
|_ html-title: Welcome To Inverse Information System Ltd.
| robots.txt: has 13 disallowed entries
| /administrator/ /cache/ /components/ /editor/ /help/
| /images/ /includes/ /language/ /mambots/ /media/ /modules/
|_ /templates/ /installation/
110/tcp open  pop3 Courier pop3d
|_ pop3-capabilities: USER STLS IMPLEMENTATION(Courier Mail Server)
UIDL PIPELINING LOGIN-DELAY(10) TOP
143/tcp open  imap Courier Imapd (released 2008)
|_ imap-capabilities: THREAD=ORDEREDSUBJECT QUOTA STARTTLS
THREAD=REFERENCES UIDPLUS ACL2=UNION SORT ACL
IMAP4rev1 IDLE NAMESPACE CHILDREN
443/tcp open  http Apache httpd 2.0.63 ((Unix) mod_ssl/2.0.63 OpenSSL/0.9.7a
mod_auth_passthrough/2.1 mod_bwlimited/1.4 FrontPage/5.0.2.2635 PHP/5.2.5)
|_ html-title: cPanel&reg;
465/tcp open  ssl/smtp Exim smtpd 4.69
```

CHAPTER 1. SECURITY DATA ACQUISITION 1.5. INTERNET AND TRANSPORT LAYERS

```
|_ sslv2: server still supports SSLv2
| smtp-commands: EHLO venus.ultracrest.com Hello example.org [41.184.2.109],
SIZE 52428800, PIPELINING, AUTH PLAIN LOGIN, HELP
|_ HELP Commands supported:
993/tcp open ssl/imap Courier Imapd (released 2008)
|_ sslv2: server still supports SSLv2
|_ imap-capabilities: THREAD=ORDEREDSUBJECT QUOTA AUTH=PLAIN
THREAD=REFERENCES UIDPLUS ACL2=UNION SORT ACL
IMAP4rev1 IDLE NAMESPACE CHILDREN
995/tcp open ssl/pop3 Courier pop3d
|_ sslv2: server still supports SSLv2
|_ pop3-capabilities: USER IMPLEMENTATION(Courier Mail Server)
UIDL PIPELINING LOGIN-DELAY(10) TOP
8443/tcp open ssl/http Apache httpd 2.0.46 ((Red Hat)
mod_ssl/2.0.46 OpenSSL/0.9.7a)
| html-title: VZPP Plesk - Log in to Plesk
|_ Requested resource was
https://inverse.com.ng:8443/vz/cp/panel/plesk/frameset
|_ sslv2: server still supports SSLv2
Aggressive OS guesses: Linux 2.6.9 (93%),
Linux 2.6.9 - 2.6.24 (93%), Infoblox NIOS Release 4.1r2-5-22263 (92%),
Linux 2.6.9 (CentOS 4.4) (92%), Linux 2.6.11 (90%),
Linux 2.6.18 (CentOS 5.1, x86) (89%), FreeBSD 6.2-RELEASE (89%),
Blue Coat Director (Linux 2.6.10) (89%), HP Brocade 4100 switch; or
Actiontec MI-424-WR, Linksys WRVS4400N, or
Netgear WNR834B wireless broadband router (88%),
AVM FRITZ!Box FON WLAN 7170 WAP (88%)
No exact OS matches for host (test conditions non-ideal).
TRACEROUTE (using port 22/tcp)
HOP RTT ADDRESS
1 0.50 192.168.1.1
2 ...
3 130.04 10.1.0.10
4 139.76 34-33.rv.ipnxtelcoms.com (62.173.34.33)
5 140.00 34-253.rv.ipnxtelcoms.com (62.173.34.253)
6 140.14 41.191.108.33
```


1.5. INTERNET AND TRANSPORT LAYERS CHAPTER 1. SECURITY DATA AQUISITION

```
7 149.97 41.191.108.9
8 359.86 41.191.108.6
9 ...
10 no response
11 249.66 xe-0.level3.londen03.uk.bb.gin.ntt.net (129.250.8.138)
12 345.69 ae-34-52.ebr2.London1.Level3.net (4.69.139.97)
13 370.65 ae-42-42.ebr1.NewYork1.Level3.net (4.69.137.70)
14 360.64 ae-3-3.ebr4.Washington1.Level3.net (4.69.132.93)
15 350.07 ae-64-64.csw1.Washington1.Level3.net (4.69.134.178)
16 350.02 ae-91-91.ebr1.Washington1.Level3.net (4.69.134.141)
17 533.70 ae-2.ebr3.Atlanta2.Level3.net (4.69.132.85)
18 369.88 ae-62-60.ebr2.Atlanta2.Level3.net (4.69.138.3)
19 380.12 ae-2.ebr2.Miami1.Level3.net (4.69.140.141)
20 369.92 ae-25-52.car1.Miami1.Level3.net (4.69.138.98)
21 389.51 PEER-1-NETW.car1.Miami1.Level3.net (4.79.96.210)
22 ... 25 no response
26 429.83 76.74.149.252
27 1679.11 76.74.146.2
28 389.74 venus.ultracrest.com (76.74.146.218)
OS and Service detection performed. Please report any incorrect
results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 299.02 seconds
```

This tells nmap to run an aggressive scan (-T4), version detection, OS fingerprinting, script scanning and traceroute (-A) on our domain inverse.com.ng or it's equivalent IP address. This is all well and good for isolated scans. In the real world, you will be scanning a good number of servers or IP addresses simultaneously, perhaps an entire DMZ of an organization, or even the external facing infrastructure. Running nmap scans and sending the output to *stdout* isn't the best way to achieve this as it does not lend itself to our subject matter - offline analysis.

1.5.3.6 Case Study 7: Nmap Target and Output Specifications

Nmap supports multiple target specifications. If you maintain a set number of IP addresses that you scan, you can input all the IPs in a text file, one per line and call it up the file with *-iL*

<input_file> option. If on the other hand you just want to scan an entire subnet or range of IPs, you can simply specify those on the command line thus *10.0.0.1-255* or its CIDR equivalent, *10.0.0.0/24*. Nmap knows to scan the subnet. You can also input your IPs manually on the command line such as *192.168.1.4*, *192.168.1.25*, *abc.com* etc. Nmap can pass hostnames, ip addresses and networks. Your mileage may differ.

Any security tool is as effective as the output it generates. Complex tests and algorithms are of little value if they aren't presented in an organized and comprehensible fashion. In this regard, Nmap supports multiple output file format. It supports the normal, XML and greppable file formats and these can be represented as *-oN*, *-oX* and *-oG <filename>* options respectively. In addition you can use the *-oA <filename>* to output to the three formats all at once. Of interest to us though is the XML output format which gives us the most flexibility.

So let's assume that I want to scan some IP addresses in a particular DMZ and also want the output in the three Nmap file formats whilst detecting service version and OS fingerprints and traceroute info without host discovery. First, I input those IPs in a text file, say *dmz.txt*, one per line, then I proceed thus:

```
# nmap -v -PN -T4 -A -iL ./dmz.txt -oA dmz
```

This is the time to take a little break, depending on the number of IPs, it could take a while. In reality, when I am on a pen test engagement, I run this over the course of the night and go take a nap. At the end of the scan three files should be in current working directory - *dmz.gnmap*, *dmz.nmap* and *dmz.xml*.

So we have another set of security data to work with. The Nmap file formats generated will be used as input source for analysis in a later case study. See 6.8.1.

1.6 Application Layer

Now we are at the topmost layer where we need to gather security data relevant to applications being run on the network. We need to be able to source data pertaining to application level vulnerabilities.

1.6.1 Vulnerability Assessment

Vulnerability assessment (VA) is the process of identifying, quantifying, and prioritizing (or ranking) the vulnerabilities inherent in systems within the network. This includes but not limited to unpatched and outdated software, insecure software configurations and weak access controls. VA involves another scanning technique, but quite unlike what obtains on the Internet and transport layers, application layer scans go beyond just banner grabbing by exploring in greater depth loopholes inherent in software implementations as well as application level misconfiguration.

1.6.1.1 Nessus²³

I seriously doubt that there is any security analyst or auditor out there who has not made use of Nessus at one point or the other, given that it has occupied the first position for a couple of years now²⁴. It is described as the world-leader in active scanners, featuring high speed discovery, configuration auditing, asset profiling, sensitive data discovery and vulnerability analysis of security posture. Due to its client/server architecture, It can be setup in a distributed combination throughout an entire enterprise, inside DMZs, and across physically separate networks.

It allows the following types of security audits:

- credentialed and un-credentialed port scanning
- network based vulnerability scanning
- credentialed based patch audits for Windows and most UNIX platforms
- credentialed configuration auditing of most Windows and UNIX platforms
- robust and comprehensive credentialed security testing of 3rd party applications such as iTunes, JAVA, Skype and Firefox
- custom and embedded web application vulnerability testing
- SQL database configuration auditing

²³<http://nessus.org>

²⁴<http://www.sectools.org>

- software enumeration on Unix and Windows
- testing anti-virus installs for out-of date signatures and configuration errors

Nessus can also be used for ad-hoc scanning, daily scans, and quick-response audits. The developers however, closed the source code in 2005 and removed the free "registered feed" version in 2008. A limited "Home Feed" is still available, though it is only licensed for home network use. Nevertheless, it is constantly updated, with more than 20,000 plugins.

Nessus Installation

At the time of writing, Tenable just released Nessus v4.0. So we hop on to nessus.org to download Nessus server. We install as root:

```
# rpm -ivh Nessus-4.0.0-fc10.i386.rpm
Preparing...          ##### [100%]
1:Nessus             ##### [100%]
Fetching the newest plugins from nessus.org...
```

It should install under `/opt/nessus` but you can verify by typing the following also as root

```
# rpm -ql Nessus
/etc/rc.d/init.d/nessusd
/opt/nessus
/opt/nessus/bin
/opt/nessus/bin/nasl
/opt/nessus/bin/nessus
.
.
.
/opt/nessus/var/nessus/tmp
/opt/nessus/var/nessus/users
```

Usage

Before you can launch Nessus, you need to create a nessus user. You do this by typing

```
# /opt/nessus/sbin/nessus-adduser
```

Create a user (*abc*) and give it a password (*12345*). Then launch

```
# /opt/nessus/sbin/nessusd -D &  
Processing the Nessus plugins... [#####]
```

I am running it in daemon mode (-D). Wait for it to process all the plugins.

To connect to the nessus daemon, you have two options. Either download the GUI Nessus client (aptly called NessusClient) from the nessus download site, or use the command line client that comes pre-installed with the Nessus server. I typically use the command line client as it's less overhead and can be launched from an *ssh* terminal. Again your mileage may differ.

Take note that the client can be installed on another machine. All you need do is allow connections to the Nessus server on port 1241 which is the default Nessus server port. You can even make multiple client connections to different Nessus servers that are situated differently. I have to admit the application is quite compelling.

1.6.1.2 Case Study 8: Vulnerability Assessment with Nessus

To scan an entire network comprising switches, routers, firewalls, servers, workstations, Windows, Unix, Linux etc...and save the output in its native *nbe*²⁵ format, I simply create a text file (say, *my_scan.txt*) with all my IP addresses or subnets in it and launch the client thus²⁶:

```
# /opt/nessus/bin/nessus -q -x -T 'nbe' localhost 1241 abc 12345 \  
./my_scan.txt ./my_scan.nbe27
```

²⁵nbe is nessus proprietary output file format

²⁶In this case the client and server are on the same machine - localhost

²⁷check the man page of nessus client for options

The `-T 'nbe'` is what tells nessus client to save in nbe format. Other formats include `.html`, `.txt` and `.nessus`.

`abc/12345` are the username/password combination to Nessus server

`my_scan.nbe` is the output data (the object of importance!)

If for some reason or the other, you need quick access to the html report generated by nessus, you can simply convert the `nbe` file to `html` thus:

```
# /opt/nessus/bin/nessus -i my_scan.nbe -o my_scan.html
```

Bingo! you can view `my_scan.html` with your browser.

Note: As an alternative to entering the absolute path to your Nessus client, Linux users can create soft (sym) links to a path within their environment variable.

This can be done with the `ln` command thus:

```
# echo $PATH
```

Use this to obtain path statement, then;

```
# ln -s /opt/nessus/bin/nessus /usr/local/bin
```

To create the soft link. All you simply have to do now to start up the Nessus client is to type `nessus` on the command line thus

```
# nessus &
```

1.7 Summary

Security Analysis is the process of analyzing security data and we have thus far examined data capture and information gathering procedures. Furthermore, the chapter presented a model for security data capture using the TCP/IP reference model. We distilled the principles and concept of security analysis and discussed the different methods of capturing and preserving data at the different layers of the model. We also foraged a bit into enterprise security assessment techniques and appraised the tools of the trade in capturing data at each of the four layers of the model. In the next chapter we will investigate packet inspection and traffic flow analysis concepts.

Chapter 2

Security Data Analysis

Security data analysis is a process of transforming data with the purpose of drawing out useful information, suggesting conclusions, and supporting decision making. This chapter presents the assumptions, principles, and techniques necessary to gain insight into raw data via statistical analysis of security data. As this chapter heavily relies on raw data, it is therefore necessary to distinguish between the often used terms - protocol and packet analysis.

2.1 Protocol and Packet Analysis

What is the difference? You may ask. Perhaps six of one, half a dozen of another. Performing a google search on either uncovers the use of these terms interchangeably by most sites including sites purportedly belonging to experts in the field. Protocol analysis is quite different from packet analysis, and in fact, far less complete in analysis of network and security data. Let me expantiate.

It is appropriate to consider protocol analysis as a subset of packet analysis. Firstly, protocol analyzers interrogate packet headers to determine the protocol that is being used for communication, like HTTP, then further ensures that the rules of the protocols are strictly adhered to, a somewhat complicated analysis, but suffice to know that this happens purely at the network interface layer.

However, there are circumstances when the protocol is absolutely right and correct, but network performance is still degraded. That's when we need to get to deeper layers of analysis, or true packet analysis. Packet headers, which contain the information about the protocol, aren't

the only sources of information for network analysis. Packet payloads also contain critical information regarding the workings of the network, and when payload analysis is included with protocol analysis we more or less have packet analysis - the complete package. Packet analyzers are capable of addressing more complex network issues be it a network or application specific issue. So for our analysis we will employ packet rather than protocol analysis.

2.1.1 Packet Capture Libraries

Packet capture (*pcap*) simply means to grab a copy of packets off the wire before they are processed by the operating system. Packets arrive at the network card, it verifies the checksum, extracts link layer data and triggers an interrupt. The interrupt then calls the corresponding kernel driver for packet capturing. The flow diagram is shown in Figure 2.1

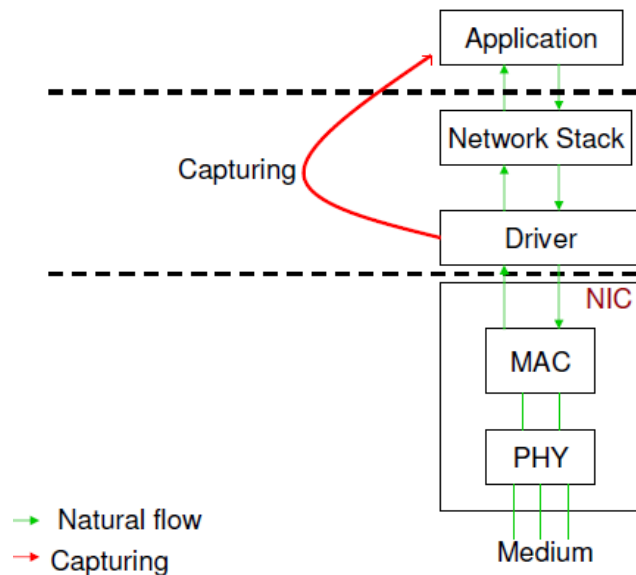


Figure 2.1:

A major function of many security tools is to capture network traffic and then either reassemble or extract information from the packets flowing across the network. What makes this possible is the packet capture library. This section provides a quick and practical introduction to packet capture using the commonly available *libpcap* library on wired and wireless networks, and is intended to accelerate and simplify the process of creating a packet-capturing tool.

Libpcap is an open source library that provides a high level interface to network packet capture. *Libpcap* is an open source C-language library for capturing network packets. *Libpcap* is available for a number of different platforms, including most Unix and Unix-like platforms (such as Linux and BSD), as well as for Windows (*Winpcap*). See Figure 2.2

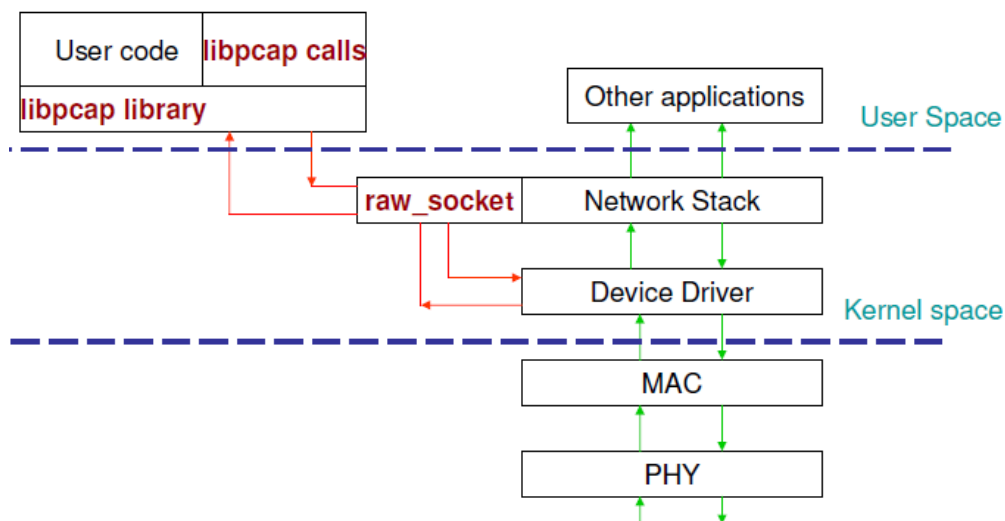


Figure 2.2:

Although *libpcap* is primarily a packet-capturing tool, it also can create and manipulate packets from saved files, which can then be used in the wide variety of tools that support the *libpcap* format.

Libpcap hides much of the complexity inherent in network packet capture. Packet capture is possible using native network functionality on most platforms; however, the interfaces and semantics required for capturing packets are not as direct. *Libpcap* and *WinPcap* also support saving captured packets to a file, and reading files containing saved packets; applications can be written. A capture file saved in the format that *libpcap* and *WinPcap* use can be read by applications that understand that format.

Libpcap and *WinPcap* provide the packet-capture and filtering engines of many open source and commercial network tools, including protocol analyzers (packet sniffers), network monitors, network intrusion detection systems, traffic-generators and network-testers.

The implementers of the *pcap* API wrote it for use in C and C++, so other languages such as Java, .NET languages, and scripting languages generally use a wrapper.

2.1.2 Packet Analysis with Wireshark

We have discussed Wireshark in the previous chapter. So in this section we will go straight into the two case studies. In the first we look at sample trace files of a node on the network infected with spyware and in the second we discuss malware analysis.

2.1.2.1 Case Study 9: Investigating Spyware Infection

In this case study, a user has called complaining that his machine drops its network connection three minutes or so after booting up. When this happens the host also maxes out its processor utilization. The problem is just isolated to this one node on the network. On investigating, we realize that the problem does indeed exist. We fire up Wireshark and get down to business. You can download the trace file here¹

Examining the packet

In examining the trace file we can see that there are quite a number of issues, but we will focus our attention on one client machine we are investigating. It's IP address is 172.16.1.10. The first thing we notice at the beginning of the capture is that an external IP address keeps trying to make a connection to this machine on a certain high port 26452. It however doesn't know what to do with this traffic since it is not expecting it, therefore it continually sends RST packets effectively ending the communication. This definitely should not be happening, however it continues for long spells until packet #70 where we see the client requesting a copy of a file called *analiz.exe* from a rogue IP address via TFTP Figure 2.3. The host starts downloading this file, and ironically enough, that is when the host starts experiencing problems.

70	73.212438	172.16.1.10	68.164.173.62	TFTP	Read Request, File: analiz.exe\000, Transfer type: octet\000
71	74.222177	172.16.1.10	68.164.173.62	TFTP	Read Request, File: analiz.exe\000, Transfer type: octet\000
72	75.587003	68.164.173.62	172.16.1.10	TFTP	Data Packet, Block: 1
73	75.587135	172.16.1.10	68.164.173.62	TFTP	Acknowledgement, Block: 1
74	76.518261	68.164.173.62	172.16.1.10	TFTP	Data Packet, Block: 1
75	76.518450	172.16.1.10	68.164.173.62	TFTP	Acknowledgement, Block: 1
76	77.518675	172.16.1.10	68.164.173.62	TFTP	Acknowledgement, Block: 1
77	77.736979	68.164.173.62	172.16.1.10	TFTP	Data Packet, Block: 2
78	77.737164	172.16.1.10	68.164.173.62	TFTP	Acknowledgement, Block: 2

Figure 2.3:

¹<http://inverse.com.ng/trace/capture1.pcap>

Final Analysis

After analyzing the physical host the process *analiz.exe* was found to be running. On terminating the process and removing the file via an automated spyware removal tool, the machine started functioning well again. What really happened was that after the host boots up it tries to make a connection to the rogue IP address associated with this spyware. After some time the client then downloads an updated copy of this spyware via TFTP and run it again on the system.

This brings up a number of good points to remember about security. The first one of these is the most obvious and that is educating users on spyware and how to avoid it. However, another key point to take note of is that this infection did a lot of its dirty work through TFTP which is a UDP based protocol. A lot of times when configuring firewall filtering rules most admins completely ignore UDP packets and only focus on TCP. As can be seen, we cannot afford to neglect UDP protocols as a lot of spyware make use this weakness to completely subvert firewall filtering rules.²

2.1.2.2 Case Study 10: Malware Behaviour Analysis

For this case study, we assume that we have been asked to analyze an executable called *bot.exe* that a number of network users have received in their email. We execute this file in a controlled environment (probably in a Virtual Machine) and capture with Wireshark for about 60 seconds. The resulting packet capture is available here³

Loading the packet capture, we see that in the first 16 packets, quite a bit is revealed. In packet #14 we notice a DNS query to a strange domain name immediately followed by a response from 84.244.1.30. A quick WHOIS search reveals that this IP is in Amsterdam. The next line really starts to clue us in. We notice the localhost making a call to 84.244.1.30 with a destination port of 5050. A google search for “port 5050” reveals a UDP reference (irrelevant, as the capture indicates TCP); the fact that Yahoo! Messenger uses TCP 5050; but most of all, at the ISS site⁴ we learn that TCP 5050 is used by eggdrop, indicated as “the most popular bot.”

Now for the good stuff. Right-click on “packet #16” and choose Follow TCP Stream. See Figure 2.4

²Thanks to Chris Sanders

³<http://inverse.com.ng/trace/capture2.pcap>

⁴http://www.iss.net/security_center/advice/Exploits/Ports/5050/default.htm

```

Stream Content
NICK [P00|GBR|64180]
USER XP-2015 * 0 :ZOMBIE1
:CandC.local 001 [P00|GBR|64180] :Welcome to the CandC server [P00|GBR|64180]!XP-2015@192.168.1.1
:CandC.local 002 [P00|GBR|64180] :Your host is CandC.local
:CandC.local 003 [P00|GBR|64180] :This server was created May 6, 2006
:CandC.local 004 [P00|GBR|64180] CandC.local CandC script
:CandC.local 005 [P00|GBR|64180] CMDS=KNOCK,MAP,DCCALLOW,USERIP SAFELIST HCN MAXCHANNELS=10 CHANLIMIT=#:10
MAXLIST=b:60,e:60,I:60 NICKLEN=30 CHANNELLEN=32 TOPICLEN=307 KICKLEN=307 AWAYLEN=307 MAXTARGETS=20 WALLCHOPS :
this server
:CandC.local 005 [P00|GBR|64180] WATCH=128 SILENCE=15 MODES=12 CHANTYPES=# PREFIX=(qao)~6@#
CHANMODES=beI,kfL,lj,psmtirRc0AQKVGcuzNSMTG NETWORK=home CASEMAPPING=ascii EXTBAN=-,cqr ELIST=MNUCT STATUSME
INVEX :are supported by this server
:CandC.local 251 [P00|GBR|64180] :There are 1 users and 2 invisible on 1 servers
:CandC.local 252 [P00|GBR|64180] 1 :operator(s) online
:CandC.local 254 [P00|GBR|64180] 1 :channels formed
:CandC.local 255 [P00|GBR|64180] :I have 2 clients and 0 servers
:CandC.local 265 [P00|GBR|64180] :Current Local Users: 2 Max: 2
:CandC.local 266 [P00|GBR|64180] :Current Global Users: 2 Max: 2
:CandC.local 422 [P00|GBR|64180] :MOTD File is missing
:[P00|GBR|64180] MODE [P00|GBR|64180] :+iw
MODE [P00|GBR|64180] +B
JOIN #reptile
:[P00|GBR|64180]!XP-2015@192.168.1.1 JOIN :#reptile
:CandC.local 332 [P00|GBR|64180] #reptile :.version
:CandC.local 333 [P00|GBR|64180] #reptile :commander 1160011515

```

Figure 2.4:

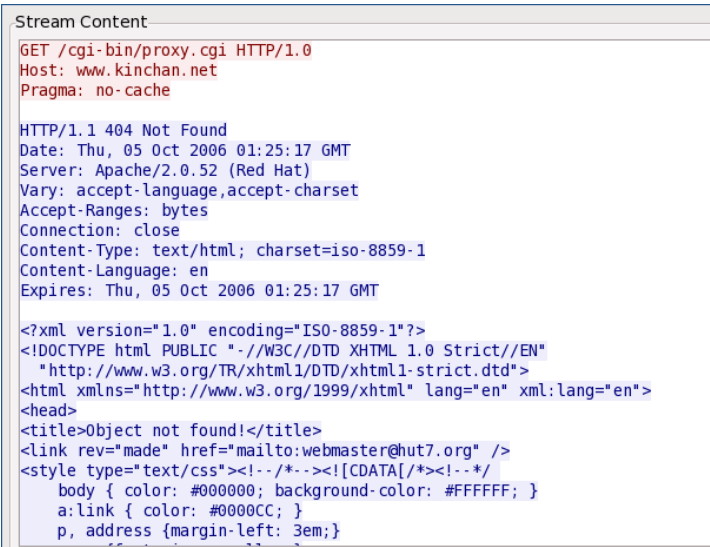
Now our virtual guest is now a bot and talking to a command and control server (CandC). But we still don't know exactly what bot we really have. Hit *Clear* on your Filter toolbar after you're done.

Further Analysis

We have seen quite a lot in just 16 packets, but what else is there? As you scroll through the packet capture, you'll start to notice some HTTP GET requests especially at packet #42, as well as more interesting DNS requests. Packet #95 really grabs my attention: a DNS request to *www.kinchan.net* – not good news. Packet #111 gives it all away. Right-click that packet and Follow TCP Stream again (see Figure 2.5), and you'll immediately see: if you Google <http://www.kinchan.net/cgi-bin/proxy.cgi> you'll get a fairly immediate hit on *Worm/SdBot.24A3@net*. Further research at the Sophos site quickly reveals that an alias for our little friend is *W32/Sdbot*. Well, we have been infected with a variant of *Sdbot*, one of many variants running amok on the Internet.

Creating firewall rules with Wireshark

Amongst the plethora of functionality Wireshark includes is the ability to create firewall rules from a packet capture. Following up with the *Sdbot* packet capture, we highlight packet #17, then choose Firewall ACL Rules from the **Analyze** menu.



```

Stream Content
GET /cgi-bin/proxy.cgi HTTP/1.0
Host: www.kinchan.net
Pragma: no-cache

HTTP/1.1 404 Not Found
Date: Thu, 05 Oct 2006 01:25:17 GMT
Server: Apache/2.0.52 (Red Hat)
Vary: accept-language,accept-charset
Accept-Ranges: bytes
Connection: close
Content-Type: text/html; charset=iso-8859-1
Content-Language: en
Expires: Thu, 05 Oct 2006 01:25:17 GMT

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
<title>Object not found!</title>
<link rev="made" href="mailto:webmaster@hut7.org" />
<style type="text/css"><!--/*--><![CDATA[/*><!--*/
  body { color: #000000; background-color: #FFFFFF; }
  a:link { color: #0000CC; }
  p, address {margin-left: 3em;}

```

Figure 2.5:

Choose Cisco IOS (extended) and you'll see Figure 2.6

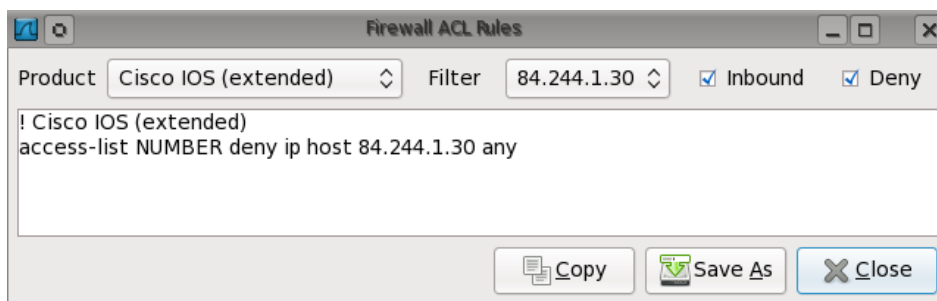


Figure 2.6:

There are a number of other options with the 'Product' setting including IPFilter, IPFW, IPTables, and even a Windows Firewall option. Entering this command on your cisco router will immediately apply the filter.⁵

⁵Thanks to Toolsmith

2.2 Traffic Flow Analysis

We continue our data analysis process by taking a cursory look at traffic flow and network activity audit process. Traffic analysis fills a huge niche in many enterprise security systems by supporting network audit. Traffic Analysis is completely different from other types of security measures, such as firewalling (mandatory access control) or IPS, where a system can generate a notification or alarm if something of interest happens. The goal of traffic analysis, is to provide accountability for network use.

If done well, traffic analysis can enable a large number of mechanisms, situational awareness, security, network audit, traffic accounting and even billing to name but a few. Audit is effective when you have good audit data generation, collection, distribution, processing and management.

2.2.1 Audit Record Generation and Utilization System⁶

According to its website, the Argus Project is focused on developing network activity audit strategies and prototype technology to support network operations, performance and security management. Furthermore, it is a data network transaction auditing tool that categorizes network packets which match the boolean expression into a protocol-specific network transaction model. Argus reports on the transactions that it discovers, as they occur.

The Argus sensor is designed to process packets (either capture files or live packet data) and generate detailed status reports of the 'flows' that it detects in the packet stream. The flow reports that Argus generates capture much of the semantics of every flow, but with a great deal of data reduction, so you can store, process, inspect or analyze large amounts of network data in a short period of time. Argus provides reachability, availability, connectivity, duration, rate, load, good-put, loss, jitter, retransmission, and delay for all network flows, and captures most attributes that are available from the packet contents, such as L2 addresses, tunnel identifiers (*MPLS*, *GRE*, *ESP*, etc...), protocol IDs, SAP's, hop-count, options, L4 transport identification (*RTP*, *RTCP* detection) and host flow control indications

For many sites, Argus is used to establish network activity audits that are then used to supplement traditional IDS based network security. These sites use contemporary IDS/IPS technology like snort to generate events and alarms, and then use the Argus network audit data to provide context for those alarms to decide if the alarms are real problems. In many DIY efforts, snort and argus run on the same high performance device. The audit data that Argus

⁶<http://www.qosient.com/argus/>

generates is great for network forensics, non-repudiation, network asset and service inventory, behavioral baselining of server and client relationships, detecting very slow scans, and supporting zero day events. The network transaction audit data that Argus generates has also been used for a wide range of other tasks including Network Billing and Accounting, Operations Management and Performance Analysis.

Argus can be considered an implementation of the technology described in the IETF IPFIX⁷ Working Group. Argus pre-dates IPFIX, and the project has actively contributed to the IPFIX effort, however, Argus technology should be considered a superset of the IPFIX architecture, providing *proof of concept* implementations for most aspects of the IPFIX applicability statement. Argus technology can read and process Cisco Netflow data, and many sites develop audits using a mixture of Argus and Netflow records.

2.2.2 Security Analysis with Argus

The best way to get started using argus, is to get the argus and client software, compile it on and play around with analyzing a few packet streams to see how it basically works, and what it looks like. Usually, those first steps will get you thinking on to how to use argus to solve network problems. Argus is supplied in source code format.

Argus Installation

Hop over to <http://www.qosient.com/argus/downloads.htm> to download the argus server and client applications. These need Flex and Bison as dependencies. Follow these steps to install (as root):

Dependencies

```
# yum -y install flex bison
```

Argus

```
# tar -xvzf argus-3.0.2.tar.gz
# cd argus-3.0.2
# ./configure && make && make install
```

⁷<http://www.ietf.org/html.charters/ipfix-charter.html>

Argus Client

```
# tar -xzvf argus-clients-3.0.2.beta.12.tar.gz
# cd argus-clients-3.0.2.beta.12
# ./configure && make && make install
# cp argus-3.0.2/support/Config/argus.conf /etc/
# chmod 600 /etc/argus.conf
```

With any luck, we have argus and argus client installed. We move on to the next stage.

2.2.2.1 Case Study 11: Basic Analysis with Argus

Argus processes data packets and generates summary network flow. If you have packets, and want to know something about what's going on, argus is a great way of looking at aspects of the data that you can't readily get from generic packet analyzers. Number of hosts talking, which host is communicating with which, how often is one address sending all the traffic? Argus is designed to generate network flow status information that can answer these and a lot more questions that may arise.

We begin by processing one of our previous packet captures, say *capture1.pcap*.

```
# argus -r capture1.pcap -w capture1.argus
```

This tells argus to *read* (-r) a pcap file and *write* (-w) to an output file for processing

```
# ra -r capture1.argus
```

This simply reads argus (ra) data from the output file (-r)

```
03:28:54.315142 e tcp 63.144.115.50.adrep <?> 172.16.1.10.micros 2 268 RST
03:28:56.067008 e tcp 172.16.1.10.sms-xf <?> 216.155.193.156.nntp 5 350 FIN
03:28:56.968593 e tcp 172.16.1.10.midnig ?> 80.67.66.62.http 1 54 RST
03:28:56.968669 e tcp 172.16.1.10.pxc-nt ?> 80.67.66.62.http 1 54 RST
03:28:59.817527 e s tcp 68.45.134.187.dnox -> 172.16.1.10.26452 6 348 RST
03:29:10.552606 e s tcp 67.38.252.160.rnmap -> 172.16.1.10.26452 6 348 RST
03:29:31.777136 e s tcp 68.45.134.187.ipfltb -> 172.16.1.10.26452 6 348 RST
```

```
03:29:50.645987 e tcp 68.164.173.62.4731 -> 172.16.1.10.epmap 6 352 FIN
.
.
03:31:23.691496 e s tcp 68.45.134.187.4475 -> 172.16.1.10.26452 6 348 RST
03:31:24.322077 e udp 172.16.1.10.pwdis <-> 68.164.173.62.tftp 10 2508 CON
03:31:29.694215 e udp 172.16.1.10.pwdis <-> 68.164.173.62.tftp 9 1950 CON
.
.
03:31:45.823607 e udp 172.16.1.10.pwdis -> 68.164.173.62.tftp 1 46 INT
03:31:45.910834 e tcp 68.45.134.187.4528 -> 172.16.1.10.26452 2 116 RST
```

Observe the bi-directional TFTP UDP established connections from the IP 172.16.1.10. Compare the output of *argus* with that of *Tcpdump* or *Wireshark*. You should see that the outputs are completely different. Instead of seeing packets on an individual level, what you will see are flow status records.

2.2.2.2 Case study 12: Argus Data Manipulation

Network Flow is an altogether a different kettle of fish not least because it provides network traffic summarization by accounting for certain attributes in the network session. Case study 11 provided us with simple analysis of a *pcap* with *Argus*. Here we will look at a more involving *Argus* analysis.

Several tools are available in *Argus* to assist in analyzing security data. All the commands are of the *ra* variety according to table below The main tools are *ra* and *racluster*.

Argus Commands	Description
ra	Reads and filters argus data
rabin	Reads and splits argus data
racluster	Aggregates argus data
racount	Performs arbitrary argus data count
radium	Works as a multiplexer allowing multiple clients concurrent access to data
ragraph	Displays argus data graphically
ragrep	Regular expression search of argus data
rasort	Sort argus output by any field
rasplit	Split argus output into several files
rastrip	Strip different parameters from argus output files
ratop	Live argus data for situation awareness

In order to proceed we need to understand the function of *racluster*. It reads argus data from an argus-data source, and merges - clusters the records based on the flow key criteria specified either on the command line, or in the *racluster* configuration file, and outputs a valid argus-stream. The *racluster* tool is primarily used for data mining, data management as well as report generation. This means that we can merge the records based on the flow key.

To generate the statistical report using start time, source address, destination address and protocol as flow key. Notice that -m proto is specified on command line below but we use -s to print the field that we want

We go back to our packet capture *capture1.pcap* (*capture1.argus*) file. this time we need to drill down on the rouge IP. We start thus:

```
# racluster -L0 -V -m stime saddr daddr proto -s stime saddr dir \
  daddr proto trans pkts bytes -r capture1.argus
```

The option "-L0" is passed to Argus so that we can have it label the column fields.

```
StartTime      SrcAddr      Dir  DstAddr      Proto Trans TotPktsTotBytes
03:28:54.315142 63.144.115.50 <-> 172.16.1.10   tcp  1    2    268
03:28:56.067008 172.16.1.10  <-> 216.155.193.156 tcp  1    5    350
03:28:56.968593 172.16.1.10  ->  80.67.66.62   tcp  1    2    108
03:28:59.817527 68.45.134.187 <-> 172.16.1.10   tcp  5   26   1508
```

```

03:29:10.552606 67.38.252.160 <-> 172.16.1.10 tcp 4 24 1392
03:29:50.645987 68.164.173.62 <-> 172.16.1.10 tcp 9 46 6224
03:30:07.527580 172.16.1.10 <-> 68.164.173.62 udp 18 258 68730
03:30:15.537150 172.16.1.10 -> 207.46.196.46 tcp 2 12 648
03:30:25.641777 172.16.1.10 <-> 172.16.0.254 udp 3 6 783
03:30:25.701839 172.16.1.10 <-> 69.64.34.124 tcp 3 48 8429
03:30:26.457069 172.16.1.10 <-> 216.127.33.119 tcp 7 47 13052
03:30:29.640286 68.164.194.35 <-> 172.16.1.10 tcp 3 30 3778
03:30:40.393489 80.167.183.40 <-> 172.16.1.10 tcp 2 6 348
03:31:30.388619 163.32.78.60 <-> 172.16.1.10 tcp 1 2 128

```

Looking at this output, this line jumps out

```
03:30:07.527580 172.16.1.10 <-> 68.164.173.62 udp 18 258 68730
```

This means that within this period, the total packets transmitted between our host 172.16.1.10 to the IP 68.164.173.63 is 258 and total size of 68,730 bytes. Secondly the transmission occurs over UDP and it is bi-directional as <-> signifies. Investigating the IP reveals that is the rogue. The above command can be piped to *rasort* which is capable of sorting the output in any format needed.

```

# racluster -V -m saddr daddr -w - -r capture1.argus - ip | \
  rasort -V -L0 -m bytes -s dur saddr dir daddr bytes pkts
Dur          SrcAddr      Dir  DstAddr      TotBytes TotPkts
98.296028    172.16.1.10 <-> 68.164.173.62 68730 258
32.231670    172.16.1.10 -> 216.127.33.119 13052 47
13.090138    172.16.1.10 -> 69.64.34.124 8429 48
34.163174    68.164.173.62 -> 172.16.1.10 6224 46
0.845039     68.164.194.35 -> 172.16.1.10 3778 30
166.093338   68.45.134.187 -> 172.16.1.10 1508 26
128.429535   67.38.252.160 -> 172.16.1.10 1392 24
1.214282     172.16.1.10 <-> 172.16.0.254 783 6
15.530030    172.16.1.10 ?> 207.46.196.46 648 12
0.120749     172.16.1.10 <?> 216.155.193.156 350 5
5.791988     80.167.183.40 -> 172.16.1.10 348 6

```

0.000046	63.144.115.50	<?>	172.16.1.10	268	2
0.000049	163.32.78.60	->	172.16.1.10	128	2
0.000076	172.16.1.10	?>	80.67.66.62	108	2

What the command is doing is to order the output to the specified object aggregation of source and destination address and then sorting the output in descending order of total bytes transmitted. I have included the duration so as to investigate total time period of packet flow. As you must have noticed, the extra option `"-w -"` is to send output to `stdout`. From here we can immediately notice again that within 98 seconds, the IP 172.16.1.10 and 68.164.173.62 exchanged a total of 68,730 bytes and 258 packets. This is a source of concern.

Another indication of the spyware on the network is a change in protocol distribution. In order to deduce this however, you need to have a baseline of normal protocol distribution on the network for comparative analysis. Once this is done you can then compare with this;

```
# racluster -L0 -V -m proto -r capture1.argus -s proto packets bytes
Proto TotBytes
udp      69513
tcp      36233
```

This shows a high number of UDP traffic. Knowing fully well that the spyware communicated with TFTP which is a UDP protocol, our level of suspicion increases. Of course if this is deemed normal, then we overlook it. You can combine in many more ways for further analysis.

All in all, there is a lot more complex aggregations that can be accomplished with the *ra* suite but it is generally beyond the scope of this book. However you can visit the Argus website form more information on usage.

2.2.3 Argus and Netflow⁸

Cisco Netflow data has become the de-facto standard for flow data reporting. Understandable, as many Cisco routers can generate this data. Netflow v5 is the predominant version because of the number of legacy routers in the market. The latest version, Netflow v9, is growing in interest because of the addition fields that the format supports. With the basic completion of

⁸<http://www.qosient.com/argus/argusnetflow.htm>

the IETF's IPFIX WG original charter, there is absolutely no doubt that Netflow V9 will be the flow data that Cisco switches and routers will generate for quite some time.

All argus-client programs can currently read native Netflow data, versions 1-8, and work has started on supporting Netflow v9. Argus data is a superset of Netflow's data schema and so there is no data loss in the conversion. Argus data that is derived from Netflow data does have a specific designation, so that any argus data processing system can realize that the data originated from a Netflow data source.

2.2.3.1 Reading Netflow Data

All argus client programs can read Cisco Netflow data, versions 1-8, and convert them to argus data streams. This enables you to filter, sort, enhance, print, graph, aggregate, label, geolocate, analyze, store, archive Netflow data along with your argus data. The data can be read from the network, which is the preferred method, or can be read from files that use the flow-tools format.

```
# ra -r capture.netflow
```

When reading from the network, argus clients are normally expecting Argus records, so we have to tell the *ra** program that the data source and format are Netflow, what port, and optionally, what interface to listen on. This is currently done using the `"-C [host:]port"` option.

```
# ra -C 1234
```

If the machine's *ra* is running on has multiple interfaces, you may need to provide the IP address of the interface you want to listen on. This address should be the same as that used by the Netflow exporter.

```
# ra -C 192.168.1.190:1234
```

While all *ra* programs can read Netflow data, if you are going to collect Netflow persistently, the preferred method is to use *radium* to collect and redistribute the data. Radium can collect from up to 256 Netflow and Argus data sources simultaneously, and provides you with a single point of access to all your flow data. Radium supports distributing the output stream to as many as 256 client programs. Some can act as IDS/IPS applications, others can build

near real-time displays and some can manage the flows as an archive, which can be a huge performance bottleneck.

All argus records contain a "Source ID", which allows us to discriminate flow data from multiple sources, for aggregation, storage, graphing, etc. The source ID used for Netflow v 1-8 data is the IP address of the transmitter.

2.2.3.2 Processing Netflow Data

There are lots of differences between argus data and netflow data: protocol support, encapsulation reporting, time precision, size of records, style and type of metrics covered. These differences are getting smaller with Netflow v9, but the biggest difference with regard to processing of Netflow data, is the directional data model.

Argus is a bi-directional flow monitor. Argus will track both sides of a network conversation when possible, and report the metrics for the complete conversation in the same flow record. The bi-directional monitor approach enables argus to provide Availability, Connectivity, Fault, Performance and Round Trip metrics. Netflow, on the other hand, is a uni-directional flow monitor, reporting only on the status and state of each half of each conversation, independently. This is a huge difference, not only in amount of data needed to report the stats (two records per transaction vs one) but also in the kind of information that the sensor can report on. There are benefits from an implementation perspective (performance) to reporting only half-duplex flow statistics, but argus sensors works great in asymmetric routing environments, where it only sees one half of the connection. In these situations, argus works just like Netflow.

Argus-client aggregating programs like *racluster*, *ratop*, *rasqlinsert* and *rabins*, have the ability to stitch uni-directional flow records into bi-directional flow records. In its default mode, *racluster* will perform **RACLUSTER_AUTO_CORRECTION**, which takes a flow record, and generates both uni-directional keys for cache hits and merging.

For *ra*

```
# ra -L0 -r capture.netflow
StartTime Flgs Proto SrcAddr Sport Dir DstAddr Dport TotPkts TotBytes State
03:28:54.315142 e tcp 63.144.115.50.adrep <?> 172.16.1.10.micros 2 268 RST
03:28:56.067008 e tcp 172.16.1.10.sms-xf <?> 216.155.193.156.nntp 5 350 FIN
03:28:56.968593 e tcp 172.16.1.10.midnig ?> 80.67.66.62.http 1 54 RST
```

```

.
.
03:30:07.527580 e udp 172.16.1.10.pwdis <-> 68.164.173.62.tftp 9 1980 CON
03:30:12.232043 e tcp 68.164.173.62.etebac -> 172.16.1.10.epmap 6 1768 FIN
03:30:13.051705 e udp 172.16.1.10.pwdis <-> 68.164.173.62.tftp 12 3112 CON
.
.
03:31:45.823607 e udp 172.16.1.10.pwdis -> 68.164.173.62.tftp 1 46 INT
03:31:45.910834 e tcp 68.45.134.187.4528 -> 172.16.1.10.26452 2 116 RST

```

Then for *racluster*

```

# racuster -L0 -r capture.netflow
StartTime Flgs Proto SrcAddr Sport Dir DstAddr Dport TotPkts TotBytes State
03:28:54.315142 e tcp 63.144.115.50.adrep <?> 172.16.1.10.micros 2 268 RST
03:28:56.067008 e tcp 172.16.1.10.sms-xf <?> 216.155.193.156.nntp 5 350 FIN
03:28:56.968593 e tcp 172.16.1.10.midnig ?> 80.67.66.62.http 1 54 RST
.
.
.
03:30:07.527580 e udp 172.16.1.10.pwdis <-> 68.164.173.62.tftp 9 1980 CON
03:30:12.232043 e tcp 68.164.173.62.etebac -> 172.16.1.10.epmap 6 1768 FIN
03:30:13.051705 e udp 172.16.1.10.pwdis <-> 68.164.173.62.tftp 12 3112 CON
.
.
03:31:45.823607 e udp 172.16.1.10.pwdis -> 68.164.173.62.tftp 1 46 INT
03:31:45.910834 e tcp 68.45.134.187.4528 -> 172.16.1.10.26452 2 116 RST

```

When uni-directional flows are merged together, *racluster* will create some of the metrics that *argus* would have generated, such as duration statistics, and some TCP state indications. And now filters like "con" (flows that were connected) and aggregations oriented around Availability (*racluster -A*) work.

When establishing an archive of *argus* data, most sites will process their files with *racluster* early in the archive establishment, but it is optional. When the data is derived from Netflow Data, the use of *racluster* is compelling and should be considered a MUST.

2.2.3.3 Situation Awareness

There are a number of definitions for situation awareness as it relates to the various subject matter⁹ but the definition that is apt for our discussion here is taken from Wikipedia. Situation awareness, or SA, is the perception of environmental elements within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future. It is also a field of study concerned with perception of the environment critical to decision-makers in complex, dynamic areas from aviation, air traffic control, power plant operations, military command and control — to more ordinary but nevertheless complex tasks such as driving an automobile or motorcycle.¹⁰ SA is simply *knowing what is going on so you can figure out what to do* (Adam, 1993)

So how does this relate to our subject matter? It is simply a matter of asking one fundamental question. *How much do we really know about our network?*

We need to know the current state of our network, network services being probed, protocols flying around, bandwidth utilization and consumption. This for us, is where the concept of situation awareness comes in. As security analysts, we are concerned with network situation awareness. Argus pretty much comes in handy here too. There are a few argus client tools that can be used for near real time network situation awareness. One of such tools is you guessed it another *ra-* tool called *ratop*

Ratop works just like the UNIX *top* process viewer command, it connects to the *argus* monitor and displays network flow data in near real time view, it also offers *vi*-like feature, where you can use */* to search for flows, and *:* as command mode to perform various actions such as network flow record filtering, sorting, flow record field re-ordering, and even extract flow record based on certain time span in real time. To run *ratop*, the *argus* monitor must be running first;

```
# argus -mAJZRU 128 -P 561
-P <portnum> Specifies the <portnum> for remote client connection
# ratop -S localhost:561
```

See Figure 2.7

To quit *ratop*, just type *[esc]:q* and you will disconnect from *argus* monitor (similar to quitting the *vi* text editor). *Ratop* is quite very useful when comes to real time network monitoring,

⁹<http://www.raes-hfg.com/crm/reports/sa-defns.pdf>

¹⁰http://en.wikipedia.org/wiki/Situation_awareness

```

ratop -s 127.0.0.1:551
Rank      StartTime      Flgs Proto      SrcAddr Sport  Dir      DstAddr Dport  TotPkts  TotBytes State
1  11:07:46.196269 e      tcp      192.168.1.2.59713 -> 216.92.197.167.http 32 8377 FIN
2  11:07:58.291135 e      tcp      192.168.1.2.59716 -> 216.92.197.167.http 32 8377 FIN
3  11:07:47.266002 e      tcp      192.168.1.2.59714 -> 216.92.197.167.http 26 6330 FIN
4  11:07:59.329462 e      tcp      192.168.1.2.59717 -> 216.92.197.167.http 26 6330 FIN
5  11:08:29.062522 e      tcp      192.168.1.2.43312 -> 69.63.178.118.http 15 3093 FIN
6  11:08:29.834968 e      tcp      192.168.1.2.43313 -> 69.63.178.118.http 14 3040 FIN
7  11:08:34.256291 e      tcp      192.168.1.2.51296 -> 67.202.94.94.http 10 2901 FIN
8  11:07:54.414207 e      tcp      192.168.1.2.51290 -> 67.202.94.94.http 9 2835 FIN
9  11:08:14.256109 e      tcp      192.168.1.2.51293 -> 67.202.94.94.http 9 2835 FIN
10 11:08:14.749359 e      tcp      192.168.1.2.41520 <?> 69.63.187.16.http 5 3430 CON
11 11:07:37.001987 e      arp      192.168.1.1      who 192.168.1.2 4 204 CON
12 11:07:47.818516 e      udp      192.168.1.2.mdns -> 224.0.0.251.mdns 3 464 INT
13 11:07:43.410501 e      udp      192.168.1.2.54069 <-> 62.173.32.89.domain 2 216 CON
14 11:07:43.582464 e      udp      192.168.1.2.48251 <-> 62.173.32.89.domain 2 229 CON
15 11:08:28.495162 e      udp      192.168.1.2.47845 <-> 62.173.32.89.domain 2 214 CON
16 11:08:28.662404 e      udp      192.168.1.2.47845 <-> 62.173.32.89.domain 2 192 CON
17 11:08:28.812235 e      udp      192.168.1.2.35151 <-> 62.173.32.89.domain 2 192 CON
18 11:08:28.932290 e      udp      192.168.1.2.52106 <-> 62.173.32.89.domain 2 338 CON

```

Figure 2.7:

while it doesn't offer you insightful information, it gives quick view of the layer 2 and 3 network conversation. Other features such as sorting can be toggled on with ':s', or filtering with ':f'.

2.2.3.4 Argus CSV Output

Just as with Wireshark, Argus can also output its data in CSV file format. Comma-Separated Values (CSV) file format is widely used and it can be easily parsed by lot of analysis and graphing tools. Here's the simple way to generate CSV data which includes source and destination address, transfer, packets and bytes from our capture1.pcap file.

```
# argus -w - -r capture1.pcap | ra -nnr - -c ',' -s saddr \
daddr trans pkts bytes
```

This will write the following to stdout

```
63.144.115.50,172.16.1.10,1,2,268
172.16.1.10,216.155.193.156,1,5,350
172.16.1.10,80.67.66.62,1,1,54
172.16.1.10,80.67.66.62,1,1,54
68.45.134.187,172.16.1.10,1,6,348
.
.
```

```
.  
68.164.173.62,172.16.1.10,1,17,3986  
172.16.1.10,68.164.173.62,1,170,45738  
.  
.  
.  
68.45.134.187,172.16.1.10,1,2,116
```

This can easily be redirected to a file thus;

```
# argus -w - -r capture1.pcap | ra -nnr - -c ',' -s saddr \  
daddr dport trans pkts bytes > capture1.csv
```

There you go. You should see a file called *capture1.csv* in your current working directory.

One of the major advantages of argus is that it provides a wide range of useful flow metrics so that you can actually generate a complete set of data for graphing purpose. Another argus *ra-* tool, *ragraph* is used for graphing traffic flow.

2.3 Parsing Security Data

So far we have only discussed the analysis of packet captures. Essentially we have only examined the network interface layer. In this section we go straight into the analysis of higher layer security output. We will examine the analysis of our network and transport layer scans as well as the application layer security assessment. Without further ado we go straight.

Security parsing in simple terms implies data conversion. It is the process of syntactic analysis of data to determine and build a recognized data structure that is useful in decision making. To parse data, it is first read and then interpreted and a new output file is created. The parsing process is capable of editing and manipulating certain data items by making calculations, moving things around and so on.

2.3.1 Parsing Nmap Output

Recall that other than single isolated Nmap scans where you probably want the result displayed on *stdout*, you will be scanning entire subnets or IP ranges. For that, it won't be prudent to have the results on *stdout*. You will more than likely be saving the results in one or

all of the three Nmap output file formats - *nmap*, *gnmap* and *xml* file formats. This has a lot of advantages as these file formats lend themselves to manipulation through analysis.

2.3.1.1 Case Study 13: Nmap Output Parsing 1 - Unix Shell

Our entry point is of course, the Unix command shell. We can use various Unix tools to manipulate the output files. So let's assume we have a lot of saved scan Nmap files in our working directory, we can analyze thus:

```
# find . -name *.gnmap -exec grep 'Status: Up' {} \;  
Host: 172.18.100.10 () Status: Up  
Host: 172.18.100.11 () Status: Up  
Host: 172.18.100.25 () Status: Up  
Host: 172.18.100.197 () Status: Up  
Host: 172.18.100.220 () Status: Up
```

This command will find all Nmap greppable output files and grep for lines with "Status: Up": We can further improve the above command with *awk* to print out only the IP addresses that appeared online:

```
# find . -name *.gnmap -exec awk '/Status:\ Up/ {print $2}' {} \;  
172.18.100.10  
172.18.100.11  
172.18.100.25  
172.18.100.197  
172.18.100.220
```

If we still need to know what file the matching results came from, we can do the following:

```
# find . -name *.gnmap -exec awk '/Status:\ Up/ {print $2}' {} \; -print  
172.18.100.10  
172.18.100.11  
172.18.100.25  
172.18.100.197  
172.18.100.220  
./scan1.gnmap
```

So much more that you can do with *awk* but we'll stop here. Unix Power Tools¹¹ is your friend here.

2.3.1.2 Case Study 14: Nmap Output Parsing 2 - `nmapxmlparser.py`

I discovered this little nifty utility¹² a while back. I have used it throughout various pen test engagements and thought it would be nice to give it a go here. `nmapxmlparser.py` is a Python script that parses Nmap's XML output (.xml files) and converts them into CSV file format. If a host has multiple ports open, it expands the results into multiple lines — one port per line into the following fields:

```
Hostname
Port
OS
OS Class
OS Match
Method
Lastboot
Hops
RTT
```

The script then splits up the port information into the following subfields:

```
Port
Protocol
State
Status
Service
Service Description
Reason
Version
```

Together, the aforementioned fields make up each result. The script has been tested using an installation of Python 2.5.2.

To use the script is as simple as typing:

¹¹<http://isbn.nu/0596003307>

¹²<http://tssci-security.com/upload/nmapxmlparser.py>

```
# python nmapxmlparser.py <scan.xml> <output.csv>
```

Let' go back to our original simple scan and save our output

```
# nmap -A -oX scan1.xml -T4 inverse.com.ng
```

This will generate an output XML file called *scan1.xml* (this will come in handy a bit later). So to convert this to CSV file format;

```
# python nmapxmlparser.py scan1.xml scan2.csv
Writing data to scan2.csv...
```

This will generate the CSV file if it doesn't exist. If it does, it appends to it. And that's it. Open it and view. You should see an output like Figure 2.8

A	B	C	D	E	F	G	H
hostname	addr	status	portid	protocol	state	service	servicedesc
venus.ultracrest.com	76.74.146.218	up					
venus.ultracrest.com	76.74.146.218	up	20	tcp	closed	ftp-data	
venus.ultracrest.com	76.74.146.218	up	21	tcp	open	ftp	PureFTPd
venus.ultracrest.com	76.74.146.218	up	22	tcp	closed	ssh	
venus.ultracrest.com	76.74.146.218	up	53	tcp	open	domain	ISC BIND 9.2.4
venus.ultracrest.com	76.74.146.218	up	80	tcp	open	http	Apache httpd 2.0.63
venus.ultracrest.com	76.74.146.218	up	110	tcp	open	pop3	Courier pop3d
venus.ultracrest.com	76.74.146.218	up	143	tcp	open	imap	Courier Imapd
venus.ultracrest.com	76.74.146.218	up	443	tcp	open	http	Apache httpd 2.0.63
venus.ultracrest.com	76.74.146.218	up	465	tcp	open	smtp	Exim smtpd 4.69 ssl venu
venus.ultracrest.com	76.74.146.218	up	993	tcp	open	imap	Courier Imapd ssl
venus.ultracrest.com	76.74.146.218	up	995	tcp	open	pop3	Courier pop3d ssl
venus.ultracrest.com	76.74.146.218	up	8443	tcp	open	http	Apache httpd 2.0.46 ssl

Figure 2.8:

The beauty of this script is that it can convert any number of XML files and output as just one CSV file thus:

```
# python nmapxmlparser.py scan1.xml scan2.xml scan3.csv
Writing data to scan3.csv...
Appending data to scan3.csv...
```

2.3.2 Parsing Nessus Output

In this section, we will discuss parsing Nessus *.nbe* proprietary output file format.

2.3.2.1 Case Study 15: Nessus Output Parsing - tissynbe.py

We are back to our friends at Top Secret/Secure Computing Information¹³ for the latest installment of the Nessus output parser¹⁴. This is a decent tool, not only because it formats and sanitizes output data but because it inserts and interacts with a database - it comes with its database schema *nessusdb.sql*. It also outputs to a CSV file for good measure and further splits up descriptions from solutions as best as possible.

Installation

The script can be downloaded here http://www.tssci-security.com/upload/tissynbe_py/tissynbe.py. Being a python script, there is no need to install, just execute. Make sure you have python installed. This was tested with Python 2.5.2. Note also that you need to install the MySQL Python library.

```
# wget -c http://www.tssci-security.com/upload/tissynbe_py/tissynbe.py
# wget -c http://www.tssci-security.com/upload/nessusdb.sql
# yum -y install MySQL-python
```

Usage

We execute thus:

```
# python tissnbe.py -h
Usage: tissynbe.py [options] args
tissynbe.py -d database -f results.nbe
tissynbe.py -d database -o output.csv
tissynbe.py -d database -o output.csv --order scriptid --sort desc
tissynbe.py -d database -o output.csv --count
tissynbe.py -f results.nbe -o output.csv
tissynbe.py -f results.nbe -d database -o output.csv
Options:
```

¹³<http://www.tssci-security.com>

¹⁴http://www.tssci-security.com/projects/tissynbe_py/

```

-d DATABASE, --database=DATABASE query results from specified MySQL database
-f INFILE, --file=INFILE input nbe file to parse
-o OUTFILE, --output-file=OUTFILE output to CSV file
-r RISK, --risk=RISK minimum risk criticality to query
--count output results by count
--order=ORDER order database query by column
--sort=SORT sort results descending
-h, --help show this help message and exit

```

As can be seen, there are all sorts of options that can be passed to the output file but the simplest usage is to convert our nessus *nbe* output file to CSV file format.

```
# python tissynbe.py -f nscan1.nbe -o nscan1.csv
```

Opening up the *nscan1.csv* file we have the following as shown in Figure 2.9

A	B	C	D	E	
Network	Host	Service	PluginID	Severity	Risk Description
196.200.65	196.200.65.249	snmp (161/tcp)	35296	1	This plugin reports the protocol version of the r
196.200.65	196.200.65.249	general/tcp	12053	1	
196.200.65	196.200.65.249	telnet (23/tcp)	22964	1	
196.200.65	196.200.65.249	http (80/tcp)	22964	1	
196.200.65	196.200.65.249	Http-mgmt (280/tcp)	22964	1	
196.200.65	196.200.65.249	telnet (23/tcp)	10281	1	A telnet server is listening on the remote port Tf
196.200.65	196.200.65.255	general/tcp	12053	1	
196.200.65	196.200.65.249	snmp (161/udp)	10800	1	The System Information of the remote host can
196.200.65	196.200.65.129	snmp (161/tcp)	35296	1	This plugin reports the protocol version of the r
196.200.65	196.200.65.129	general/tcp	12053	1	
196.200.65	196.200.65.129	http (80/tcp)	22964	1	
196.200.65	196.200.65.151	general/tcp	12053	1	
196.200.65	196.200.65.145	general/tcp	12053	1	
196.200.65	196.200.65.241	general/tcp	12053	1	
196.200.65	196.200.65.129	telnet (23/tcp)	22964	1	
196.200.65	196.200.65.129	http-mgmt (280/tcp)	22964	1	
196.200.65	196.200.65.66	snmp (161/tcp)	35296	1	This plugin reports the protocol version of the r

Figure 2.9: 196.200.65 196.200.65.66 snmp (161/tcp) 35296 1 This plugin reports the protocol version of the r

Even though we can analyze a CSV file with the in-built functionalities of most spreadsheets, we may still need the flexibility that a relational database server provides, perhaps as it relates with storage and being able to serve an external reporting tool. To this end, *tissynbe.py* ships with its MySQL schema file which we will just load directly into a MySQL server instance. First install MySQL server and client application and follow the steps highlighted below;


```
# yum -y install mysql mysql-server
```

Since MySQL installs without a password for the root account, you may want to immediately password protect it thus;

```
# mysqladmin -u root password <NEWPASSWORD>
```

You may want to load the *nessusdb.sql* file into MySQL at this moment. I don't recommend using the root account for administering your database. I suggest you create another account and grant this account the necessary permissions thus;

```
# mysql -u root -p < nessusdb.sql
Enter your password.
```

This creates the database *nessus* with two tables - *results* and *timestamps*

Create a new user

```
# mysql -u root -p
Enter your password
mysql> CREATE USER <user> IDENTIFIED BY PASSWORD '<password>';
```

Grant Privileges

```
mysql> GRANT ALL ON nessus TO '<user>'@'localhost';
```

And that's it for MySQL setup.

Now you need to open up the *tissynbe.py* file in a text editor and search for these three lines

```
DB_HOST    = 'hostname'
DB_UNAME   = 'username'
DB_PASSWD  = 'password'
```

Change *DB_HOST* to your hostname, typically localhost, *DB_UNAME* to the user account you created and *DB_PASSWD* to the password you assigned, then save.

To populate the MySQL database with our *nessus nbe* file, execute the following;

```
# python tissynbe.py -d nessus -f nscan1.nbe
Processing nscan1.nbe...
Executing SQL INSERT...
tissynbe.py:195: Warning: Out of range value adjusted for
column 'scriptid' at row 1
VALUES (%s, %s, %s, %s, %s, %s, %s)"""', (small_results))
tissynbe.py:195: Warning: Out of range value adjusted for
column 'scriptid' at row 9
VALUES (%s, %s, %s, %s, %s, %s, %s)"""', (small_results))
tissynbe.py:195: Warning: Out of range value adjusted for
column 'scriptid' at row 17
VALUES (%s, %s, %s, %s, %s, %s, %s)"""', (small_results))
.
.
.
.
tissynbe.py:195: Warning: Out of range value adjusted for
column 'scriptid' at row 96
VALUES (%s, %s, %s, %s, %s, %s, %s)"""', (small_results))
Number of rows inserted: 202 results
Number of rows inserted: 36 timestamps
```

Now log into mysql to verify by querying the nessus database.

```
# mysql -u <user> -p nessus
Enter your password
mysql> select * from results;
```

You will see a lot of data output on *stdout*. The database is now be populated with our nessus result.

Note:

You may want to also convert the Nmap CSV file into an SQL file and load into MySQL. I use a certain service to do this. Navigate to <http://www.sqldbu.com/eng/sections/tips/mysqlimport.html> and load up your

Nmap CSV file in the form provided, leave all other settings as they are but make sure you highlight the MySQL version you have, then click '**Start**'. It will immediately generate a text (.txt) file of the same name as the CSV file. Load the text file into MySQL as shown above. Sometimes, you may need to tweak the generated SQL a little bit, but it should be straight forward. You may also take a look at ¹⁵ as well. Picalo2.4.1 which we'll look at in the next section also provides a convenient method of importing data from text files (CSV, TSV, etc.) and other sources into databases.

2.4 Advance Security Analysis

We present in this section the principles and techniques necessary to gain insight into security data via advance mode security data analysis.

Security analysis is a process of gathering, modeling, and transforming security data with the goal of highlighting useful information, suggesting conclusions, and supporting decision making. It is an approach or framework that employs various techniques to maximize insight into a security dataset, uncover underlying security data structure, extract important security data variables and detect security anomalies. Security analysis is not identical to security visualization even though the two terms are often used interchangeably. Whilst security visualization is a collection of graphical techniques based on one data characterization aspect, security analysis encompasses a larger superset that reveals its underlying structure and model. We look at security visualization in later chapters.

2.4.1 Picalo¹⁶

Permit me to introduce this piece of software. I came across this application sometime ago while I was looking for alternatives to Microsoft Excel and Open office Calc's in-built analysis functionalities. I was looking for something that would be flexible enough to analyze a CSV file format as well as connect to and pull information from a datastore like a MySQL database for detailed analysis. There and then I came across this open source project called Picalo.

Now let's get down to brass tacks. Picalo is a data analysis application, with a focus on fraud detection and data retrieved from corporate databases. It serves also as the foundation for an automated fraud detection system. However, it is an open framework that could actually

¹⁵impsql

¹⁶<http://www.picalo.org>

be used for many different types of data analysis: network logs, scientific data, any type of database-oriented data and data mining.

So you may ask why not just use Microsoft Excel or Open Office Calc? Conan Albrecht the author answers thus:

Microsoft Excel (or, insert your favorite spreadsheet here) has become a powerful, mature application for number analysis. Spreadsheets are widely known and used, and they are visual in their analysis. However, spreadsheets are best suited for ad-hoc analysis rather than formal database-oriented analysis. For example, Excel is an excellent choice for tracking investments or calculating a home mortgage schedule. It is less suitable for querying, stratifying, summarizing, joining, matching and trending routines Picalo specializes in.

Picalo is meant to work with data retrieved from large databases; Excel is meant to work primarily with small sets of numbers in free-form. While Excel can only handle about 65,000 records, Picalo can handle millions upon millions of records (limited only by available memory in your computer). Picalo is meant for the professional. It's startup cost is higher, but it provides economies of scale not possible with ad-hoc programs like Excel.

2.4.1.1 Usage Considerations

Let's examine some best practice methods of using Picalo. As your needs may vary to ascertain degree, I will attempt to highlight how best to achieve optimal results with Picalo. As already mentioned, Picalo can be used to data mine and analyze huge databases for security, fraud and irregularities. These databases are often very large and require a significant amount of processing. Data modification is not something I embark upon very often, so I relate with them purely as a read-only data source. The architecture I typically use is shown in Figure 2.10. This diagram shows Picalo being used to query corporate data sources to populate a data warehouse, where Picalo is then used for analysis. Other applications are also used to present and query data. The process is as follows:

- The first step is to create a data warehouse.

You should set up a server, which is simply a computer with a database like PostgreSQL, MySQL, SQL Server, or some other production database. In particular, PostgreSQL and MySQL are free; MySQL is quite easy. However, any database works.

Based upon the types of analysis you want to do, design a database that will support your needs.

- Using ODBC or Picalo's data import wizard, transfer the appropriate data from the corporate server to your data warehouse. ODBC is set up in the Windows control panel. Once the data is queried into a Picalo table, Picalo's Upload Table feature can then be used to upload the data into the data warehouse.
- Using ODBC or Picalo's data import wizard, transfer the appropriate data from the corporate server to your data warehouse. ODBC is set up in the Windows control panel. Once you query the data into a Picalo table, you can use Picalo's Upload Table feature to upload the data into the data warehouse.

Note from the process above that Picalo's native file format, *.pco*, is not used directly. Use Picalo's native format for ad hoc data storage and temporary holding. Normally, data warehouses using MySQL, or PostgreSQL hold your data. Realize that Picalo's primary function is analysis. It is most powerful when paired with a database that specializes in data storage, quick access, and powerful queries.

2.4.1.2 Data Analysis with Picalo

We start by explaining the primary data structure in Picalo: the table. Using Picalo tables is typically the first port of call to easily learn its usage because all functions and routines work on tables of data. Tables are very similar to Excel or Calc spreadsheets – they are two dimensional grids made up of columns and rows.

Picalo tables are two dimensional grids of data. The data can be of any type, and all cells in the table do not have to be the same. All data should be kept in tables because almost all functions and routines in Picalo expect tables for input and usually produce tables for output. In fact, most times data will be imported into Picalo from Excel, Calc, CSV file, or similar data source because Picalo is an primary analysis application rather than a data entry application. There are however, some important differences between Picalo tables and spreadsheets. These are described as follows:

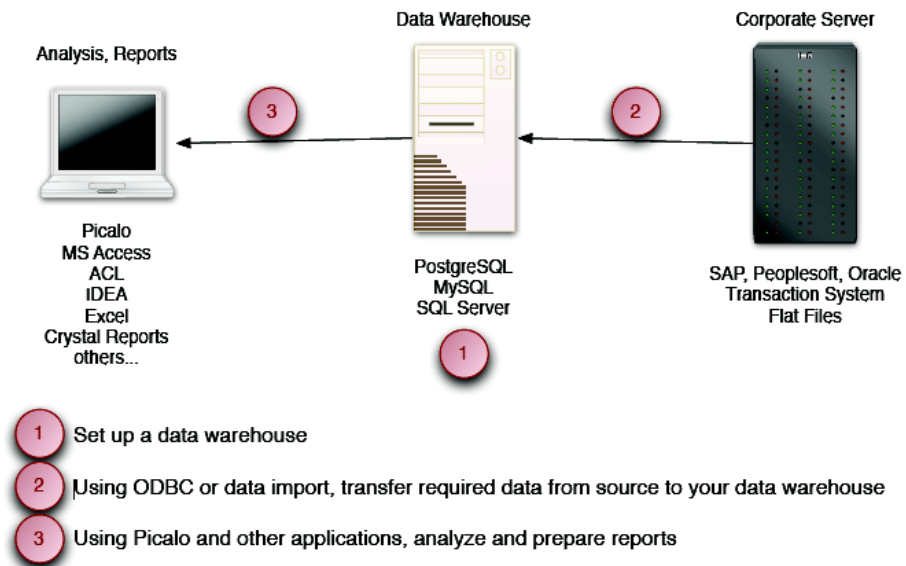


Figure 2.10:

- Each row, also known as a *record* in a Picalo table usually represent a real world item, such as a person or a transaction. Each record is an individual entry. If you want to hold information about 20 information assets, your table will have 20 records.
- Columns also referred to as a *field* in Picalo tables are always named and usually hold the same type of data. If your table holds information about computers, the first column might be the category, the second column might be type, and so forth. The column names are not records in the table like they would be in a spreadsheet.
- Picalo tables have only as many columns and rows as you have data for. In contrast, spreadsheets always have 65,000 rows and A-IV columns. If you want to add another record to a Picalo table, you first have to append a new row to the end of it.
- Picalo tables normally hold large amounts of data with limited calculations. Spreadsheets specialize in calculations between cells, and they are often used for tasks like accounting. In contrast, calculations in Picalo are normally done on entire columns rather than individuals cells.

Each Picalo table can theoretically hold up to 2.1 billion rows and 2.1 billion columns, but most

users will most likely not reach this limit. The primary limitation on table size in Picalo is the memory available. The realistic limit on number of rows and columns depends on the size the data in each cell and the ratio of columns to rows¹⁷. Practically, you should expect to hold at least hundreds of thousands of records with a relatively modern workstation. Tables are always named in Picalo. When you load or create a table, Picalo will ask you to enter a name that it can be referenced by in dialogs, routines, and other operations. This name is usually similar to the filename the data was loaded from, but you can set it to anything.

In case study 13 we'll start interacting with Picalo tables, but before then we need to obtain and install it. Picalo is an application that runs on most platforms including Windows, Linux, Debian and Mac OS X. As usual our platform of choice (where we can help it) is Linux. The installation on Windows is quite easy and straightforward.

Installation

We visit <http://www.picalo.org/download.spy> to download the rpm version of Picalo. At the time of writing, Picalo's version number is 4.37.

```
# wget -c http://www.picalo.org/download/current-picalo-rpm.tar.gz
# tar -xzvf current-picalo-rpm.tar.gz
# rpm -ivh picalo*.rpm
```

Note With this version there are two pieces of apps present *picalo-4.37-1.noarch.rpm* and *picalo-core-4.37-1.noarch.rpm*

Picalo is a python based application and it installs in the */usr/share/picalo* directory on the Linux filesystem

2.4.1.3 Case Study 16: Working with tables in Picalo

Before we start, lets go back to our *nscan1.csv* file. This is the file we will be working with.

You can launch the Picalo executable thus;

¹⁷If you need to work with huge tables (in the hundreds of millions to billions of records range), you should store your records in a production database. Picalo's Database module can iterate through records individually rather than load entire tables into memory at once.

```
# /usr/share/picalo/Picalo.pyw &
```

It prompts you to create a folder anywhere within the filesystem (where you have permissions) to store your data. You are immediately presented with a splash screen followed by a work area.¹⁸

To load a Picalo table, follow this process;

1. Click **File -> Import -> Select...** (At this point navigate to the location of the *nscan1.csv* file on your hard disk. You can leave the 'New Table Name' option as it is)
2. Click **Next -> Click Delimited Text File radio button -> Click Next**
3. Leave these settings as they are -> **Click Next**
4. Enter *Packet* into '**Field Name**' and Leave '**Field Type**' as *String* -> **Click Finish**

It should load the *nscan1.csv* file now. It might take about a minute or so. Eventually you are presented with the data import shown in Figure 2.11

	Network	Host	Service	PluginID	Severity	
0	196.200.65	196.200.65.129	snmp (161/udp)	10264	2	The community name of the remote SNMP server can be guessed. I
1	196.200.65	196.200.65.129	http-mgmt (280/tcp)	10815	2	The remote web server is prone to cross-site scripting attacks. The
2	196.200.65	196.200.65.129	snmp (161/tcp)	35296	1	This plugin reports the protocol version of the remote SNMP agent.
3	196.200.65	196.200.65.129	general/tcp	12053	1	
4	196.200.65	196.200.65.129	http (80/tcp)	22964	1	
5	196.200.65	196.200.65.129	telnet (23/tcp)	22964	1	
6	196.200.65	196.200.65.129	http-mgmt (280/tcp)	22964	1	
7	196.200.65	196.200.65.129	telnet (23/tcp)	10281	1	A telnet server is listening on the remote port. The remote host is n
8	196.200.65	196.200.65.129	snmp (161/udp)	10800	1	The System Information of the remote host can be obtained via SN
9	196.200.65	196.200.65.129	snmp (161/udp)	10551	1	The list of network interfaces cards of the remote host can be obta
10	196.200.65	196.200.65.129	snmp (161/udp)	34022	1	The list of IP routes on the remote host can be obtained via SNMP. I
11	196.200.65	196.200.65.129	general/udp	10287	1	
12	196.200.65	196.200.65.129	http-mgmt (280/tcp)	34850	1	The remote web server seems to transmit credentials in clear text.
13	196.200.65	196.200.65.129	http-mgmt (280/tcp)	10107	1	A web server is running on the remote host. This plugin attempts to
14	196.200.65	196.200.65.129	general/tcp	11936	1	
15	196.200.65	196.200.65.129	http-mgmt (280/tcp)	24260	1	Some information about the remote HTTP configuration can be extr
16	196.200.65	196.200.65.129	general/tcp	19506	1	
17	196.200.65	196.200.65.129	general/tcp	10919	1	
18	196.200.65	196.200.65.132	general/tcp	12053	1	
19	196.200.65	196.200.65.132	general/icmp	10114	1	It is possible to determine the exact time set on the remote host.
20	196.200.65	196.200.65.132	general/udp	10287	1	

Figure 2.11:

¹⁸Please refer to Picalo manuals for more.

Descriptives

Viewing table descriptives is an important first step that you should always complete when analyzing a table. Descriptives show basic statistics, control totals, and record counts. You should use these types of descriptives to ensure that your data imported correctly, were typed to the correct type without problems, and no records were missed.

1. Double-click a table in the left-side project browser to view table data.
2. Select **Analyze -> Descriptives...**
3. A new table shows containing the descriptives for the table.

Most of the descriptives are self-explanatory. The *NumEmpty*, *NumNone*, and *NumZero* are subtly different. The *NumEmpty* descriptive tells you how many cells are the empty string (""). The *NumNone* descriptive tells you how many cells are set to the None type, regardless of the column type (since any cell anywhere in a table can be set to None). The *NumZero* descriptive tells you how many cells are set to 0 (for numerical typed columns). Further down, you'll also notice that it gives useful statistics about the severity - *min*, *max*, *median*, *mean*, *variance* and *standard deviation*.

Sorting

The Picalo GUI supports sorting a table by up to three fields and sorting by any number of fields. The sort method requires that all fields be sorted ascending or descending. Here we sort first by Severity, then by Service, then by Host. This will give you a general overview of the services with the most vulnerabilities.

1. Double-click a table in the left-side project browser to view table data.
2. Select **File -> Sort...**
3. Select the fields to sort by in the dialog that comes up.

The table should be sorted accordingly.

Database Connectivity

Connecting Picalo to a datastore say, MySQL or any other ODBC data source is the best way to perform data interaction. This gives the database the flexibility of storing data and allows Picalo do what it was primarily designed to do which is data analysis. Picalo provides access to relational databases through the Database module. It comes with the drivers for three types of connections: ODBC (Windows only), SQLite, Postgresql, Oracle and MySQL. The ODBC driver connects to any database you can set up an ODBC connection for.

Since we already have a MySQL database setup, we simply connect to it with Picalo. I make the assumption that we are running both Picalo and MySQL on the same host.

1. **Select File -> New Database Connection...**
2. In the dialog that comes up, select the **MySQL** radio button -> **Next**
3. Fill in the following parameters from our earlier case study.

```
Database Name: nessus
Username: <user>
Password: <password>
Hostname or IP address: localhost
port: 3306
```

Assign a connection name. Mine is *Connect1*

You should by now have your database connection under Databases in the left side project browser as depicted in Figure 2.12

Database Query analyzer

You can query the datastore directly in Picalo thus;

1. Right click on *Connect1* (or whatever name you gave it) and select **New Query....**
2. Enter a query name in the '**Query Name:**' dialog box. I entered *Query1*

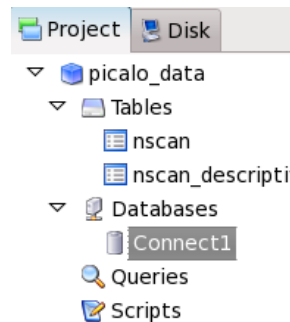


Figure 2.12:

3. Make sure *Connect1* is selected under '**Available Tables**' section
4. Proceed to double click on the results table

In the results section you should see the fields in my case I see the following

- id
- domain
- host
- service
- scriptid
- riskval
- msg1
- msg2

Let's for the moment assume that I need to find hosts with services with the highest number of vulnerabilities, I simply pick the *host*, *service* and *riskval* parameters under 'fields'

Then in the '**Select records where this field is**' pick '**greater than or equal to**' and in the dialog box input the value 2

In the query text window you will see

```
SELECT results.host, results.service, results.riskval
FROM results WHERE results.riskval >= 2
```

Click 'Run Query'

You should see an output shown in Figure 2.13

	host	service	riskval
0	196.200.65.249	snmp (161/udp)	2
1	196.200.65.129	snmp (161/udp)	2
2	196.200.65.145	snmp (161/udp)	2
3	196.200.65.70	snmp (161/udp)	3
4	196.200.65.71	snmp (161/udp)	3
5	196.200.65.66	snmp (161/udp)	3
6	196.200.65.70	https (443/tcp)	2
7	196.200.65.134	https (443/tcp)	2
8	196.200.65.66	https (443/tcp)	2
9	196.200.65.136	https (443/tcp)	2
10	196.200.65.134	https (443/tcp)	2
11	196.200.65.129	http-mgmt (280/tcp)	2
12	196.200.65.134	https (443/tcp)	2
13	196.200.65.67	snmp (161/udp)	3

Figure 2.13:

So we can see here that the hosts with the highest number of vulnerabilities are 196.200.65.66, 196.200.65.67, 196.200.65.70 and 196.200.65.71 and by far the most vulnerable service on the network is *snmp* (UDP port 161). All these can be exported in a number of file formats including CSV, TSV and XLS via **File -> Export** in Picalo. A number of open source database reporting tools can be used to generate reports including eye popping graphs and charts. This is discussed in the next section.

There are a number of activities that can be done within the Picalo interface. I suggest the reader get familiar with filtering and sorting data as they will be useful in daily security analysis activities.

2.4.2 Reporting

Once you are through with analysis, you no doubt need to generate dynamic reports. Even as most vulnerability and application scanners generate reports, there may be a need to generate

reports that are specific to your own needs in addition to being able to create and deliver rich, web-based reports to your stakeholders and corporate end-users alike. There are a number of database reporting tools and libraries out there ranging from the elaborate to the simple.

Here we will discuss an intuitive Java based report writing tool that can interface with most databases called LogiReport¹⁹ by LogiXML²⁰. It has a lot of bells and whistles and most of all the price is right - it is quintessentially FREE. The little downside to it is that the main component of the report writer tool is only available for Windows.

2.4.2.1 Logi Report

Logi Report Studio is a free reporting system that lets security analysts create and deliver rich, web-based reports. It has more useful features than several commercial reporting software solutions. It's functional specifications amongst others include report drill-down and drill-through, charting, data grouping and export functionality.

2.4.2.2 Reporting with Logi Report

LogiReport Studio makes the contents of most data sources easy to access and comprehend within the context of a browser. Its in built functionality through the use of various Web technologies allows for the general distribution of rich, interactive report applications. Logi Report brings several Web technologies together, allowing report developers to quickly deliver data in meaningful ways. The Logi framework consists of:

- An XML-based modeling language
- A runtime server (Logi Report Server)
- An integrated development environment (Logi Studio)

Logi Report Server is the hub between the presentation and data layers see Figure 2.14 . Logi Report Tools separate data handling, reporting and presentation. The Logi modeling language enables Web reports to be developed in report definition files. Logi Report Studio offers an integrated development environment with helpful wizards and tools to reduce report development time even though it is possible for reports to be developed with any text editor . The

¹⁹<http://www.freereporting.com>

²⁰<http://www.logixml.com>

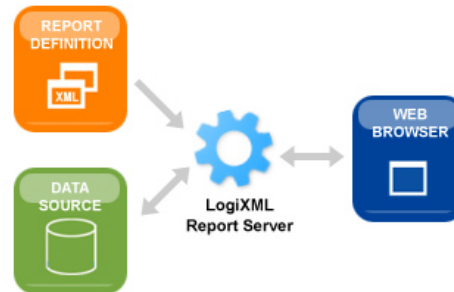


Figure 2.14:

Logi Report Server parses the XML constructs at runtime, creating HTML for browser based presentation.

2.4.2.3 Case Study 17: Report Generation with Logi Report Studio

In the following case study, we generate a report from our earlier nessus database in MySQL.

Logi Report Installation

Navigate to the download section of Logi Report and obtain the .Net version. At the moment the current version is 9.5.114. Make sure you have Microsoft IIS 5 or 6 and .Net Framework 2.0 or higher installed. Unzip and run the executable following the prompts.

Usage

Launch Logi Studio. You will be presented with the Report Development Environment consisting of three panes as shown in Figure 2.15

- Start the wizard by selecting the **New Application...** item on Studio's **File** menu
- Create a **new folder** for your application: `C:\Inetpub\wwwroot\nessus`
- Click **OK** to create the application. This populates the *MyNewApp* folder with the required basic application files and folders and opens the application in Studio (you may see a progress bar that says "Deploying application").

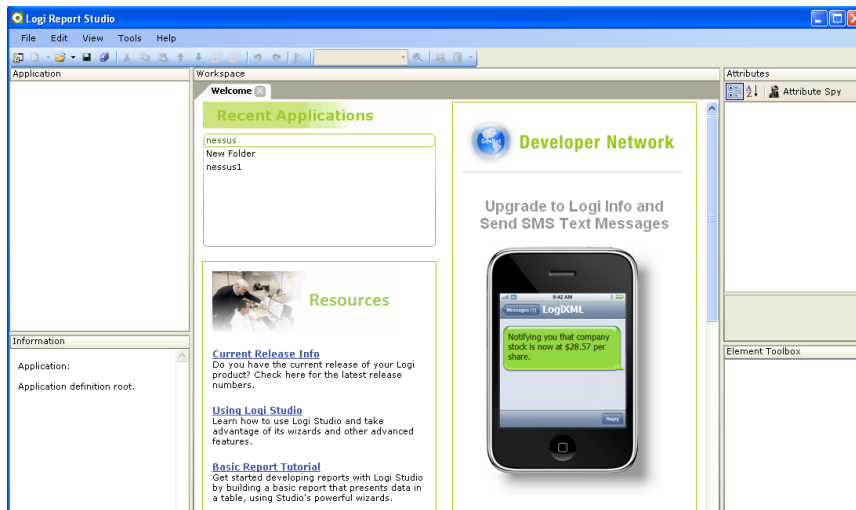


Figure 2.15:

- The wizard will now present a number of dialog boxes, starting with the one shown above left, prompting you for necessary information. Click **Next** and note the important information relating to IIS and the .NET framework displayed in the box, shown above right. Do not proceed if these conditions are not true. Click **Next** to continue.
- The next few dialog boxes will prompt you to provide the specifics of the connection. You will need to enter the following details in the ensuing dialog boxes;

```

Application Server Name: localhost
Application Caption: Nessus Report
Application Path (URL): http://localhost/nessus
Connection ID: NessusConnection
Connection Type: MySql
MySql Server Name: <ip address of mysql servr>
MySql database Name: nessus
MySql User Name: <your_user>
MySql password: <your_password>

```

- Logi Studio then tests for database connectivity. If your MySQL parameters were entered properly, the test should return a success.

- Next, a choice of **data presentations** is offered. Use the default value, Data Table and just click **Next** to continue. The new data table wizard will open; when it does click **Next**.
- The following examples refer to use of a table result in the nessus. If you're using a different database and table, adjust the names and selections as you deem appropriate. Tables require a unique **ID**; delete the suggested ID and enter ResultsDataTable. Click **Next** to continue. In the background, the wizard will open the **Default** report definition in the Workspace Panel and add a **Data Table** element to it.
- Tables use a **DataLayer** element to retrieve data and it also requires a unique ID; enter ResultsDataLayer. Click **Next** to continue. In the background, the wizard will add a **DataLayer.SQL** element to the report definition.
- The datalayer element uses the datasource connection you configured earlier to connect to the database. The wizard will present that connection as the default value. To use it, just click **Next**.
- The datalayer also needs to know what data to retrieve. Our **SQL query** is very simple: SELECT results.* FROM results, so just type it into the text box, as shown above, and click **Next** to continue. You can also use the **Build the SQL Query statement...** to build your query
- The data table wizard will now configure the **columns** by using the datalayer to query the table and then display a list of columns matching the SQL query result set. Check all the columns shown, then click **Next**. The wizard will add elements for each column to the report definition.
- Next, the wizard will ask if you want **paging controls** added. Use the default value, Yes, by clicking **Next** to continue. Similarly, take the default value, 20, for the maximum **number of table rows** per page and click **Next** again.
- 18. Next, use the default value, Graphical/Images, to select the **style** of the page navigation controls by clicking **Next** to continue. Then select a **style sheet** for the report, as shown above right, by using the default value, Blue.css, and clicking **Next**.
- The wizard will **finish** configuring your report definition and the completion dialog box, shown above, will appear. Click **Finish**.

- Your report definition is complete. Click the **Preview** link at the bottom of the Workspace Panel to preview your data table report. Click the **Browse Application** icon on Studio's toolbar or press the **f5** key to open your report in a full browser window. See Figure 2.16

Nessus Report
09/24/2009 23:43:14

Page 1 of 11

id	domain	host	service	scriptid	riskval	msg1	
1	196.200.65	196.200.65.249	snmp (161/tcp)	32767	True	This plugin reports the protocol version of the remote SNMP agent. By sending an SNMP 'get-next-request', it is possible to determine the protocol version of the remote SNMP agent.	See als service factor:
2	196.200.65	196.200.65.249	general/tcp	12053	True		196.20
3	196.200.65	196.200.65.249	telnet (23/tcp)	22964	True		A telne
4	196.200.65	196.200.65.249	http (80/tcp)	22964	True		A web
5	196.200.65	196.200.65.249	http-mgmt (280/tcp)	22964	True		A web
6	196.200.65	196.200.65.249	telnet (23/tcp)	10281	True	A telnet server is listening on the remote port The remote host is running a telnet server. Using telnet is not recommended as logins, passwords and commands are transferred in clear text. An attacker may eavesdrop on a telnet session and obtain the credentials of other users.	Solutio (CVSS:
7	196.200.65	196.200.65.255	general/tcp	12053	True		196.20
8	196.200.65	196.200.65.249	snmp (161/tcp)	10800	True	The System Information of the remote host can be obtained via SNMP. It is possible to obtain the system information about the remote host by sending SNMP requests with the OIDs.	Solutio going t Router.

Figure 2.16:

You may want to take some time to review the elements the wizard added to your Default report and *_settings* definitions. In the Default definition, notice the **Label** element beneath each **Data Column** element; look at its attributes and see the **@Data** token used to represent the data from the data layer.

2.5 Summary

Security data analysis is the overall process of transforming data with the purpose of drawing out useful information, suggesting conclusions, and supporting decision making. This chapter has presented the assumptions, principles, and techniques necessary to gain insight into raw data via statistical analysis of security data. We started by taking an in-depth look at security data extraction through protocol and traffic flow analysis and subsequently explored security data preservation as well as data iteration and report development process.

Part II

PROCESS

Chapter 3

Security Data Carving

Let me start right away by making a declaration: packetized network data reconstruction is difficult. Traditionally, applications could be identified by the TCP/UDP port number they use, for instance HTTP (TCP port 80) and SMTP (TCP port 25). This made it easy to monitor and control traffic as packets traversing a network could be blocked, optimized and accounted for by simply referring to their TCP/UDP port numbers. Making such assumptions based on port numbers alone doesn't suffice anymore. For example, unwanted and sometimes malicious applications can work over TCP port 80, and appear to be HTTP. Some applications allocate ports dynamically and even randomly implement port swapping so that there is no way of knowing what they are based purely on port number.

3.1 Application Layer Packet Classification

There are several reasons why you may want to be able to identify traffic on your network. It usually boils down to some combination of:

Monitoring - Finding out the sort of traffic and applications being run by users.

Accounting - Limiting the bandwidth used by certain protocols.

Blocking - Limiting the use or completely removing certain protocols.

To be able to classify packets, we need a mechanism to distinguish which packets belong to which protocols. However, traffic is not identifiable by simply looking at the port number anymore. This chapter seeks out ways of identifying known methods for protocol identification.

Application layer packet inspection will be used to analyze the payload of the packet in order to classify it. This will have the dual effect of allowing us both monitor and optimize traffic that masks as other traffic, for instance, peer-to-peer file sharing going over port 80. It also determines traffic running over unknown ports, for instance HTTP over TCP port 13331.

3.1.1 Layer 7 Monitoring

Packet classification should perform application layer monitoring automatically. It should examine application layer data in order to correctly identify and classify traffic flows, viz

- Traffic that systematically allocates ports for data transfer (e.g. passive FTP)
- Traffic using a non-standard port number (e.g. HTTP on a high port)
- Traffic using a port number it wouldn't normally use (e.g. Gnutella using TCP port 80)

3.2 Statistical Protocol ID

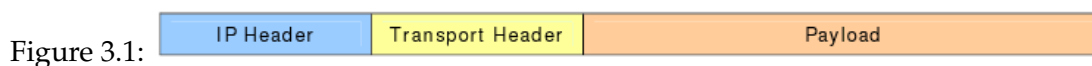
There is an ever growing need for reliable classification of network packets according to application layer protocols as understanding the type of traffic carried on networks enhances the possible detection of illegitimate traffic, such as network attacks and related security violations. Most firewalls and intrusion detection and prevention systems need to reliably identify network protocols in order to implement fine-grained and secure access policies. Most traffic classification techniques deduce which application layer protocol is being used in a session by using the list of well-known ports assigned by the International Assigned Number Authority (IANA). Other solutions that often use port numbers to deduce the application layer protocol are systems that assign Quality of Service (QoS) priorities and traffic shaping algorithms.

With the advent of P2P file sharing, more applications started using unpredictable dynamic port ranges and reused well-known ports of other applications, which yielded poor results

for port classification methods on modern network data. This development led to a wide use of deep packet inspection (DPI) for classification, which means inspection of packet payloads for known string patterns. DPI is currently the most reliable way to classify traffic, which explains its popularity in most commercial tools. However, examination of application layer data causes substantial legal and privacy concerns. Furthermore, DPI with static signatures is rather resource-expensive and does not work on encrypted traffic, which is becoming common as a reaction to legal threats.

Statistical Protocol IDentification¹ (SPID) algorithm designed by Erik Hjelmvik² is a protocol identification scheme which reliably identifies the application layer protocol by using statistical measurements of traffic flow data and application layer data. These attributes can be defined by all sorts of packet and flow data, ranging from traditional statistical flow features to application level data measurements, such as byte frequencies and offsets for common byte-values. In this sense SPID is a hybrid technique, utilizing efficient generic attributes, which can include deep packet inspection elements by treating them in the same way as statistical flow properties. A proof-of-concept (PoC) application for the SPID algorithm is available at Sourceforge³.

The concept of Protocol Identification is also known as Port Independent Protocol Identification (PIPI), Application Identification⁴, Protocol Discovery⁵, Application Recognition⁶ and Traffic Classification⁷. A typical packet may look like the example in Figure 3.1



3.2.1 SPID Algorithm Overview

SPID performs protocol identification by comparing the protocol model of an observed session to pre-calculated protocol models of known protocols. **Protocol models 3.2.4** contain a set of attribute fingerprints. Fingerprints are created through frequency analysis of various

¹http://sourceforge.net/apps/mediawiki/spid/index.php?title=Main_Page

²http://www.iis.se/docs/The_SPID_Algorithm_-_Statistical_Protocol_IDentification.pdf

³<http://sourceforge.net/projects/spid/>

⁴<http://www-rp.lip6.fr/~teixeira/bernaile-conext06.pdf>

⁵http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ftpdmib.html

⁶<http://www.cisco.com/web/go/nbar/>

⁷<http://www.caida.org/research/traffic-analysis/classification-overview/>

attributes, such as application layer data or flow features, and are represented as probability distributions.

The SPID algorithm also makes use of flow measurements (that do not require inspection of application layer data), such as *packet sizes*⁸, *packet inter-arrival times*⁹ and *packet order number- and direction combinations*¹⁰. Attribute fingerprints are represented in the form of probability distributions. This means that the data for each fingerprint is represented by two vectors or arrays of discrete bins: one array of counter bins and one of probability bins. Values of the counter bins represent the number of times an observation (analyzed packet) has caused the associated attribute meter to trigger that particular index number in the vector. Probability vectors are normalized versions of the counter vectors, with all values in every probability vector summing up to 1.0.

The SPID algorithm presents a reliable method for identifying protocols based on both flow and application data. The strong identification feature is a result of the innovative protocol attribute metering functions, the protocol models' rich data format and the model comparison algorithm.

3.2.2 SPID Algorithm Requirements

The main objective of the SPID algorithm is to reliably identify which application layer protocol is being used in a network communication session in an easy and efficient fashion. SPID should not only be able to classify traffic into rough, coarse-grained traffic classes (such as P2P or web), but in fine-grained classes on a per-protocol basis, which would enable detailed QoS assignments and security assessment of network flows. A key design goal of SPID is therefore to replace the use of pattern matching techniques with entropy based comparisons of probability distributions. Doing so eliminates the need for manually extracting inherent properties of protocols, since the SPID algorithm has the ability to automatically deduce properties from training data. The training data used, however needs to be preclassified, which can be done through manual classification by experts or by active approaches. A further goal of SPID is to allow protocol models to be updated easily as new training data becomes available, without having access to the previously used training data.

Below is a list of key operational requirements for the algorithm

1. Small protocol database size

⁸<http://sourceforge.net/apps/mediawiki/spid/index.php?title=First4OrderedDirectionPacketSizeMeter>

⁹<http://sourceforge.net/apps/mediawiki/spid/index.php?title=First4OrderedDirectionInterPacketDelayMeter>

¹⁰<http://sourceforge.net/apps/mediawiki/spid/index.php?title=First4OrderedDirectionPacketSizeMeter>

2. Low time complexity
3. Early identification of the protocol in a session
4. Reliable and accurate protocol identification

The motivation for requirements 1 and 2 is it should be possible to run the SPID algorithm in real-time on an embedded network device, which has limited memory and processing capabilities. The motivation for requirement 3 is it shall be possible to use the results from the SPID algorithm in a live traffic capturing environment to automatically take measures in order to, for example, provide quality of service (QoS) to an active session, block illicit traffic or store related traffic for off-line analysis¹. There is also the need for enterprises to degrade P2P services (via rate-limiting, service differentiation and blocking) on their networks in favour of performance for business critical applications. It is therefore required that the protocol must be identifiable, with the SPID algorithm, based on only the four first application data packets (i.e. not counting the flow control signalling packets) in a session. Requirement 4 does not need any further motivation than the obvious provisioning of high quality of service.

The SPID algorithm does not require support for advanced pattern-matching techniques, such as regular expressions. By providing a generic XML based format to represent protocol model fingerprints, SPID is designed to be both platform and programming language independent.

3.2.3 SPID Data Flow

As illustrated in Figure 3.2, SPID performs protocol identification by comparing the protocol model of an observed session to precalculated protocol models of known protocols.

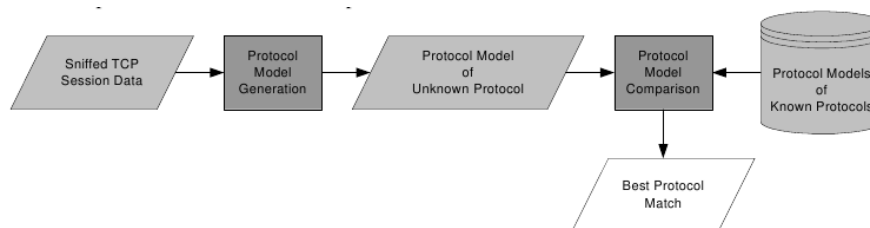


Figure 3.2:

The required manual efforts for adding a new protocol are thereby shifted from detailed protocol analysis to assembling training data for that particular protocol. This is an important

change since manual creation of static protocol patterns is a time consuming task, and new protocols continuously appear. Many new protocols are proprietary and undocumented binary protocols, which require advanced reverse engineering in order to manually generate the protocol patterns. The SPID algorithm does not require support for advanced pattern-matching techniques, such as regular expressions.

3.2.4 Protocol Models

The SPID algorithm performs protocol identification by using statistical fingerprints. The statistical fingerprints of a session or known protocol, are part of an object called **Protocol Model**. Protocol models contain a set of attribute fingerprints (Figure 3.3). These fingerprints are created through frequency analysis of various attributes, such as application layer data or flow features and are represented as probability distributions. The application layer protocol in a session is identified by comparing its protocol model to protocol models of known protocols.

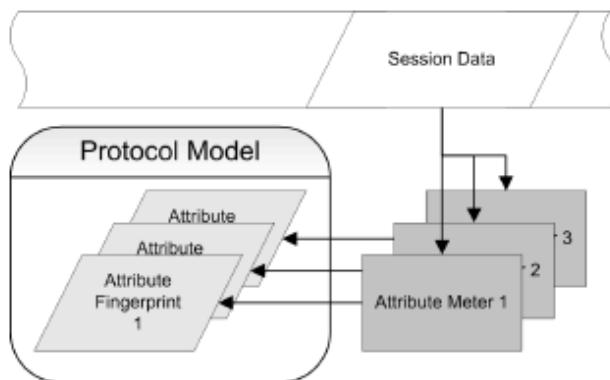


Figure 3.3:

A protocol model is created upon session establishment after the TCP three-way handshake consisting of a set of attribute fingerprints. Every packet with application layer data belonging to a session is called an observation. Each observation is then passed through the attribute meters, which provides measurements that are stored in the session's protocol model. Upon receiving such a measurement, the protocol model increments the fingerprint counters accordingly. For purposes of illustration, we assume an attribute fingerprint for the *ByteFre-*

*quencyMeter*¹¹ from the first data packet observed in a HTTP session, that is, an HTTP *GET* command. The counters would be incremented to:

- 3 for the counter at index 84 (since there are three T's in 'GET / HTTP/ 1.1')
- 2 for counters at index 32, 47 and 49 (space, '/' and '1')
- 1 for counters at index 71, 69, 72, 80 and 46
- 0 for all other counters

All other attribute fingerprints belonging to the same protocol model will also increase their counters based on the sets of indices that are returned from their respective attribute meter. Subsequent packets in the same session will induce an increment in the fingerprint counter values. However, since one of the design goals of SPID is to keep time complexity low, we want to prove that utilizing only the first few packets provides sufficient precision. Protocol models for known protocols are generated from real network packet traces. These traces need to be pre-classified, either manually or automatically, in order to be usable as training data for the SPID algorithm. The pre-classified training data is converted to protocol model objects (one per protocol) by generating protocol models for each session and merging (i.e. adding) the fingerprints of the same protocol and attribute type. The more sessions are merged together for each protocol, the more reliable the fingerprint will be. It was discovered that 10% of the fingerprints' vector lengths (i.e. approximately 25) turned out to be a rough measurement of the minimum number of training sessions needed to build a reliable protocol model.

3.2.5 Comparison of Protocol Models

Fingerprints of an observed session are compared to fingerprints of known protocol models by calculating the *Kullback-Leibler* (K-L) divergence otherwise known as relative entropy between the probability distributions of the observed session and each protocol model ranging from 0 (identical distributions) to 1. The K-L divergence is a value that represents how much extra information is required to describe the values in the observed session by using a code, which is optimized for the known protocol model instead of using a code optimized for the session protocol model. The best match for an observed session is the attribute fingerprint which yields the smallest K-L divergence.

¹¹<http://sourceforge.net/apps/mediawiki/spid/index.php?title=ByteFrequencyMeter>

Protocol models of observed sessions are finally compared to protocol models of known protocols by calculating the K-L divergences of the models' attribute fingerprints. The best protocol match is the one with the smallest average K-L divergence of the underlying attribute fingerprints. A good approach is to assign a threshold value, where only K-L divergence average values below the threshold are considered matches. If none of the known protocol models match, the session is classified as 'unknown' in order to avoid false positives for known models.

3.2.6 SPID Results

Information on the preliminary results of SPID can be found in this paper by Erik Hjelmvik and Wolfgang John¹².

A more detailed description of the inner workings of the "SPID Algorithm Proof-of-Concept" can be attained by looking at the source code, which is freely available from SourceForge¹³

3.2.7 SPID PoC Implementation

The proof of concept implementation of the SPID algorithm is available on *sourceforge.net*. The proof-of-concept application only allows for protocol identification of protocols which use the TCP protocol as transport layer. The proof-of-concept application therefore, makes use of the 5-tuple¹⁴ to identify the bi-directional flow¹⁵, which constitutes the TCP session. The SPID algorithm can, however, be used to identify protocols in any communication scheme where there is a notion of a session, i.e. a bi-directional flow. This implies that the SPID algorithm can (with various success rates) be used also to identify protocols that are transported or tunnelled within protocols such as UDP, HTTP, NetBIOS, DCE RPC or SSL.

The PoC SPID implementation infact uses over 30 attribute meters¹⁶, which are the functions that provide the distribution measurements for each specific attribute. An example of such an attribute meter is the basic *ByteFrequencyMeter*¹⁷, which measures the frequency with which all of the possible 256 bytes occur in the application layer data. Other attribute meters perform

¹²http://spid.sourceforge.net/sncnw09-hjelmvik_john-CR.pdf

¹³<http://sourceforge.net/projects/spid/>

¹⁴A 5-tuple is a set of: source IP, source port, destination IP, destination port and transport protocol

¹⁵A bi-directional flow consists of the data sent in both directions for a specific 5-tuple

¹⁶<http://sourceforge.net/apps/mediawiki/spid/index.php?title=AttributeMeters>

¹⁷<http://sourceforge.net/apps/mediawiki/spid/index.php?title=ByteFrequencyMeter>

much more advanced analysis of various properties in a session, such as measuring the frequency of various request-response combinations (e.g. HTTP behavior, where a 'GET' request is followed by an HTTP response or FTP behavior where a '220' message is replied to with a 'USER' command).

A vector length of 256 is used in the PoC implementation; an implementation of the SPID algorithm can, however, use any length for these vectors.

3.2.7.1 Case Study 18: SPID Algorithm PoC¹⁸

The proof-of-concept application for the SPID algorithm is written in C# using the Microsoft .NET framework. The application is designed to load two types of files; protocol model database files (in XML format) and single TCP-session capture files in *pcap* file format. The application automatically attempts to identify the application layer protocol when a TCP session capture file is loaded.

The proof-of-concept application makes use of over 30 different attribute meters in order to generate fingerprints. Many of these attribute meters are somewhat overlapping, so there is some potential to being able to reduce the number of attribute meters in future implementations.

Usage

At the time of writing this book, the SPID PoC tool is currently in version 0.4. Launch the *SPID.exe*¹⁹. You should immediately see the interface in Figure 3.4

Notice the table of protocols appears in the list view on the right of the user interface. The protocol models list also displays the number of sessions and observations that have been used, in form of pre-classified training data, to generate the fingerprints of each protocol model.

For this case study, we will be examining a simple http download capture file²⁰. Upload the *pcap* file thus;

File -> Open -> Pcap File

Upon loading the *pcap* file with a TCP session the application starts creating a protocol model based on the first 200 packets in the session. The protocol model's attribute fingerprints are

¹⁸http://sourceforge.net/project/platformdownload.php?group_id=58169

¹⁹You need to install Microsoft.Net Framework v2.0

²⁰<http://inverse.com.ng/trace/capture4.pcap>

The screenshot shows the 'SPID Algorithm Proof-of-Concept 0.4' application window. The main area is titled 'Classified Sessions (200 first displayed)' and contains a table with columns: Client IP, C. Port, Server IP, S. Port, Start T..., Obser..., and Protocol. Below this is a section for 'Protocol Models' with columns: Protocol, Sessions, and Obser... The following table represents the data in the Protocol Models section:

Protocol	Sessions	Obser...
IMAP	13	113
IMAP (cli...	13	53
IMAP (se...	13	60
IRC	31	297
IRC (cli...	31	101
IRC (ser...	30	196
MSE	19	221
MSE (cli...	19	103
MSE (ser...	19	118
MSN	23	227
MSN (cli...	23	93
MSN (se...	23	134
POP	26	262
POP (cli...	26	108
POP (ser...	26	154
SMTP	27	296
SMTP (c...	27	127
SMTP (s...	27	169
SSH	54	577
SSH (cli...	53	317
SSH (ser...	54	261
SSL	81	734
SSL (cli...	79	300
SSL (ser...	81	436

Figure 3.4:

then compared to those of the known protocols in the database. It correctly identifies the protocols as can be seen under the 'Protocol' column in Figure 3.5.

The screenshot shows the 'SPID Algorithm Proof-of-Concept 0.4' application window. The main area is titled 'Classified Sessions (200 first displayed)' and contains a table with columns: Client IP, C. Port, Server IP, S. Port, Start Time, Obser..., and Protocol. The following table represents the data in the Classified Sessions section:

Client IP	C. Port	Server IP	S. Port	Start Time	Obser...	Protocol
192.168.1.190	1198	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1199	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1200	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1201	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1202	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1196	209.85.135.103	80	9/11/2009 8:51:09 AM	10	HTTP
192.168.1.190	1203	66.235.120.110	80	9/11/2009 8:51:09 AM	2	HTTP
192.168.1.190	1206	80.157.170.89	80	9/11/2009 8:51:10 AM	9	HTTP
192.168.1.190	1204	69.147.121.161	443	9/11/2009 8:51:10 AM	11	SSL
192.168.1.190	1197	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1214	204.16.104.2	80	9/11/2009 8:51:19 AM	9	HTTP
192.168.1.190	1215	204.16.104.2	80	9/11/2009 8:51:20 AM	9	HTTP
192.168.1.190	1217	204.16.104.2	80	9/11/2009 8:51:20 AM	9	HTTP
192.168.1.190	1219	204.16.104.2	80	9/11/2009 8:51:20 AM	10	HTTP
192.168.1.190	1220	204.16.104.2	80	9/11/2009 8:51:20 AM	11	HTTP
192.168.1.190	1223	128.30.52.168	80	9/11/2009 8:51:22 AM	3	HTTP
192.168.1.190	1216	204.16.104.2	80	9/11/2009 8:51:20 AM	12	HTTP
192.168.1.190	1224	87.248.218.147	80	9/11/2009 8:51:23 AM	8	HTTP
192.168.1.190	1218	204.16.104.2	80	9/11/2009 8:51:20 AM	10	HTTP
192.168.1.190	1226	137.226.34.227	80	9/11/2009 8:51:34 AM	9	HTTP
192.168.1.190	1225	66.235.133.14	80	9/11/2009 8:51:23 AM	6	HTTP
192.168.1.190	1228	66.235.120.104	80	9/11/2009 8:52:02 AM	3	HTTP
192.168.1.190	1221	204.16.104.2	80	9/11/2009 8:51:22 AM	2	HTTP
192.168.1.190	1222	204.16.104.2	80	9/11/2009 8:51:22 AM	2	HTTP
192.168.1.190	1229	209.85.135.101	80	9/11/2009 8:52:23 AM	2	HTTP

Figure 3.5:

Highlighting any one of the packets in the TCP stream, the average Kullback-Leibler divergence between the observed session's fingerprints and those of the known protocol models, is displayed in the pane immediately below the Classified Sessions pane. The protocol that best matches the observed session is automatically selected as shown in Figure below:

Client IP	C. Port	Server IP	S. Port	Start Time	Obser...	Protocol
192.168.1.190	1198	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1199	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1200	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1201	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1202	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1196	209.85.135.103	80	9/11/2009 8:51:08 AM	10	HTTP
192.168.1.190	1203	66.235.120.110	80	9/11/2009 8:51:09 AM	2	HTTP
192.168.1.190	1206	80.157.170.89	80	9/11/2009 8:51:10 AM	9	HTTP
192.168.1.190	1204	69.147.121.161	443	9/11/2009 8:51:10 AM	11	SSL
192.168.1.190	1197	66.235.120.110	80	9/11/2009 8:51:08 AM	2	HTTP
192.168.1.190	1214	204.16.104.2	80	9/11/2009 8:51:19 AM	9	HTTP
192.168.1.190	1215	204.16.104.2	80	9/11/2009 8:51:20 AM	9	HTTP
192.168.1.190	1217	204.16.104.2	80	9/11/2009 8:51:20 AM	9	HTTP
192.168.1.190	1219	204.16.104.2	80	9/11/2009 8:51:20 AM	10	HTTP
192.168.1.190	1220	204.16.104.2	80	9/11/2009 8:51:20 AM	11	HTTP
192.168.1.190	1223	128.30.52.168	80	9/11/2009 8:51:22 AM	3	HTTP
192.168.1.190	1216	204.16.104.2	80	9/11/2009 8:51:20 AM	12	HTTP
192.168.1.190	1224	87.248.218.147	80	9/11/2009 8:51:23 AM	8	HTTP
192.168.1.190	1218	204.16.104.2	80	9/11/2009 8:51:20 AM	10	HTTP
192.168.1.190	1226	137.226.34.227	80	9/11/2009 8:51:34 AM	9	HTTP
192.168.1.190	1225	66.235.133.14	80	9/11/2009 8:51:23 AM	6	HTTP
192.168.1.190	1228	66.235.120.104	80	9/11/2009 8:52:02 AM	3	HTTP
192.168.1.190	1231	204.16.104.2	80	9/11/2009 8:51:20 AM	2	HTTP

Protocol Model	Divergence (avg)	Match Percentage
BitTorrent	3.7692	00.09% *
BitTorrent (client->server)	3.8244	00.07% *
BitTorrent (server->client)	3.5250	00.21% *
eDonkey	3.4486	00.27% *
eDonkey (client->server)	3.9354	00.05% *
eDonkey (server->client)	3.3019	00.45% *
FTP	3.4154	00.31% *
FTP (client->server)	3.9276	00.05% *
FTP (server->client)	3.3031	00.45% *
HTTP	1.8210	76.58% *
HTTP (client->server)	3.1676	00.72% ****
HTTP (server->client)	2.5046	07.17% ****
IMAP	3.1695	00.72% *
IMAP (client->server)	3.3486	00.36% *

Figure 3.6:

The application can also be used in order to generate a new protocol model database from scratch by adding new protocols to an empty protocol model database. Different protocol model databases can also be merged into a combined protocol model database simply by importing several protocol model databases (one at a time) into the application. The merged database can then be saved to disk by choosing **Save Protocol Model Database** from the file menu.

Metadata about protocols, such as their default port numbers, can be added to the protocol model database. The port number information is not, under any circumstances, used in order to perform the protocol identification. The purpose of adding metadata about default port numbers is merely in order to allow other applications to use the database to, for example, alert upon detection of protocols leeching on well known ports of other protocols (such as P2P

protocols running on TCP 80 or 443).

The protocol model database of proof-of-concept SPID application can be extend with more traffic and more training data in order to support more protocols and attain more reliable protocol identifications.

3.2.8 SPID Algorithm Tuning

The SPID algorithm needs to be tuned in order to adjust parameters such as:

- Vector lengths
- Kullback-Leibler divergence threshold value
- Number of attribute meters used

One other very important tuning aspect is to choose the best combination of attribute meters, which provides the most robust protocol identification service. The best combination of attribute metering functions may vary depending on the requirements of a specific application; early identification might for example be prioritized to actively block illicit traffic, while this functionality is not very important when performing post-event analysis. The recommendation is that all implementations should use the same combination of attribute meters. Doing so will make it easy to share attribute fingerprints for known protocols with others.

3.3 Passive Network Analysis

Going through network traffic on a packet-by-packet or byte-per-byte level can be very powerful at times, but as the captured packets grows the need for more advanced analysis tools becomes apparent. This section outlines the importance of analyzing captured network traffic and introduces some applications designed to support security analysis by extracting useful information from captured data - a method loosely referred to as data carving utilizing the algorithms already enumerated above.

3.3.1 Tcpextract²¹

Tcpextract is a tool for extracting files from network traffic based on file signatures. Extracting files based on file type headers and footers is an age old data recovery technique. Tcpextract uses this technique specifically for the application of intercepting files transmitted across a network. Tcpextract features the following:

- Supports 26 popular file formats out-of-the-box. New formats can be added by simply editing its config file.
- With a quick conversion, you can use your old Foremost config file with tcpextract.
- Custom written search algorithm is lightning fast and very scalable.
- Search algorithm searches across packet boundaries for total coverage and forensic quality.
- Uses libpcap, a popular, portable and stable library for network data capture.
- Can be used against a live network or a tcpdump formatted capture file.

3.3.1.1 Case study 19: Extract Data with Tcpextract

In this case study we will examine an http download pcap with embedded images²².

Installation

Download tcpextract here²³ and follow these procedures to install

```
# tar -xzf tcpextract-1.0.1.tar.gz
# cd tcpextract-1.0.1
# ./configure && make && make install
# cp tcpextract.conf /etc
```

The tcpextract executable should be in your \$PATH statement.

²¹<http://tcpextract.sourceforge.net/>

²²<http://inverse.com.ng/trace/capture5.pcap>

²³<http://prdownloads.sourceforge.net/tcpextract/tcpextract-1.0.1.tar.gz?download>

Usage

We extract and analyze our pcap file thus;

```
# mkdir images
# tcpxtract -f capture5.pcap -c /etc/tcpxtract.conf -o images
```

You should see the following output on *stdout*

```
Found file of type "html" in session [10.1.1.1:20480 -> 10.1.1.101:26892],
exporting to ./images/00000000.html
Found file of type "html" in session [10.1.1.1:20480 -> 10.1.1.101:29708],
exporting to ./images/00000001.html
Found file of type "jpg" in session [10.1.1.1:20480 -> 10.1.1.101:29964],
exporting to ./images/00000002.jpg
Found file of type "jpg" in session [10.1.1.1:20480 -> 10.1.1.101:30220],
exporting to ./images/00000003.jpg
Found file of type "html" in session [10.1.1.1:20480 -> 10.1.1.101:31500],
exporting to ./images/00000004.html
Found file of type "html" in session [10.1.1.1:20480 -> 10.1.1.101:31756],
exporting to ./images/00000005.html
Found file of type "html" in session [10.1.1.1:20480 -> 10.1.1.101:32012],
exporting to ./images/00000006.html
Found file of type "jpg" in session [10.1.1.1:20480 -> 10.1.1.101:32268],
exporting to ./images/00000007.jpg
Found file of type "jpg" in session [10.1.1.1:20480 -> 10.1.1.101:32524],
exporting to ./images/00000008.jpg
Found file of type "jpg" in session [10.1.1.1:20480 -> 10.1.1.101:32780],
exporting to ./images/00000009.jpg
```

You can then navigate to the images folder to view the extracted html and jpeg files. I viewed a random image say *00000008.jpg*. Figure 3.7 is a screenshot of the captured image.



Figure 3.7:

3.3.2 Chaosreader²⁴

Chaosreader is an open source tool to trace TCP and UDP sessions and fetch application data from Snoop (solaris), Wireshark or Tcpcap logs. This is a type of "any-snarf" program, as it will fetch telnet sessions, FTP files, HTTP transfers (HTML, GIF, JPEG, ...), SMTP emails, etc from the captured data inside network packet captures. An HTML index file is created that links to all the session details, including realtime replay programs for telnet, rlogin, IRC, X11 or VNC sessions; and reports such as image reports and HTTP GET/POST content reports. Chaosreader can also run in standalone mode - where it invokes Tcpcap or sSnoop (if they are available) to create the log files and then processes them.

Some of its features include:

- Reads Solaris snoop logs and four versions of tcpcap/libpcap logs
- Standalone mode generates a series of logs and then processes those
- Processes HTTP, FTP, telnet, SMTP, IRC, ... application protocols
- Processes any TCP and UDP traffic * Processes 802.11b wireless traffic
- Processes PPPoE traffic, tun device traffic
- Retrieves transferred files from FTP and HTTP traffic

²⁴<http://www.brendangregg.com/chaosreader.html>

- Creates HTML and text reports to list contents of the log
- Creates realtime replay programs for telnet or IRC sessions
- Creates red/blue coloured HTML reports for 2-way sessions such as telnet and FTP
- Creates red/blue coloured HTML reports for any TCP, UDP or ICMP traffic
- Creates image reports from HTTP, FTP transfers
- Creates HTTP GET and POST reports from queries
- Creates red/blue coloured HTML hex dumps for any TCP, UDP or ICMP traffic
- Creates plain text hex dumps for any TCP, UDP or ICMP traffic
- Creates HTTP proxy logs based on observed HTTP traffic, using the squid log format
- Creates X11 realtime replay programs to playback an X11 session. (experimental).
- Creates red/blue coloured HTML reports for X11 text and keystrokes.
- Creates realtime replay programs for X11 text communication.
- Creates VNC realtime replay programs to playback a VNC session. (experimental).
- Creates HTML reports for VNC keystrokes.
- Creates realtime replay programs for VNC keystrokes.
- SSH content analysis. reports, replays and keystroke delay data files
- Creates raw data files from TCP or UDP transfers
- Supports TCP out of sequence number delivery
- Supports IP fragmentation
- Supports IPv4 and IPv6
- Processes ICMP and ICMPv6
- Very configurable (including filtering on IPs and ports)

- Can sort data based on time, size, type or IP.
- Can skip sessions smaller than a min size.
- Runs on Solaris, RedHat, Windows,

As you can see it can accomplish quite a bit..

3.3.2.1 Case Study 20: Using Chaosreader

Installation

Chaosreader can be downloaded here²⁵. As it is a perl script, you need to have Perl installed to run it.

Usage

To get help on how to use

```
# perl chaosreader -h
```

Detailed help is available by typing;

```
# perl chaosreader --help
```

Enhanced help is further available by typing

```
# perl chaosreader --help2
```

To analyze our previous pcap capture file, *capture5.pcap*;

```
# perl chaosreader -ve capture5.pcap
```

You should see the following on *stdout*

²⁵<http://sourceforge.net/projects/chaosreader/files/chaosreader/0.94/chaosreader0.94/download>

```
Chaosreader ver 0.94
Opening, http_with_jpegs.cap
Reading file contents,
100% (326754/326754)
Reassembling packets,
100% (464/483)
Creating files...
Num Session (host:port <=> host:port) Service
0019 10.1.1.101:3200,10.1.1.1:80 http
0012 10.1.1.101:3193,209.225.0.6:80 http
0005 10.1.1.101:3185,209.225.0.6:80 http
0015 10.1.1.101:3196,10.1.1.1:80 http
0001 10.1.1.101:3177,10.1.1.1:80 http
0008 10.1.1.101:3189,10.1.1.1:80 http
0013 10.1.1.101:3194,209.225.0.6:80 http
0003 10.1.1.101:3183,209.225.0.6:80 http
0018 10.1.1.101:3199,10.1.1.1:80 http
0007 10.1.1.101:3188,10.1.1.1:80 http
0014 10.1.1.101:3195,10.1.1.1:80 http
0009 10.1.1.101:3190,10.1.1.1:80 http
0002 10.1.1.101:3179,209.225.11.237:80 http
0004 10.1.1.101:3184,209.225.0.6:80 http
0016 10.1.1.101:3197,10.1.1.1:80 http
0017 10.1.1.101:3198,10.1.1.1:80 http
0011 10.1.1.101:3192,209.225.0.6:80 http
0010 10.1.1.101:3191,209.225.0.6:80 http
0006 10.1.1.101:3187,209.225.0.6:80 http
index.html created.
```

This command tells Chaosreader to output everything (-e) about the capture file. You can then open the *index.html* file with a browser. Figure 3.8 shows an example of *index.html*

At this point you can click on any one of the **Image Report**, **GET/Post Report** and **HTTP Proxy Log** for further in-depth analysis of the packet capture. Obviously you can accomplish a lot more with Chaosreader. This exercise is left to the reader.

Note: Another tool that closely performs the same operations as Chaosreader is one called

Chaosreader Report

File: http_with_jpegs.cap, Type: tcpdump, Created at: Sat Sep 26 14:26:19 2009

[Image Report](#) - Click here for a report on captured images.

[GET/POST Report](#) - Click here for a report on HTTP GETs and POSTs.

[HTTP Proxy Log](#) - Click here for a generated proxy style HTTP log.

TCP/UDP/... Sessions

1.	Fri Nov 19 23:29:14 2004	0 s	10.1.1.101:3177 -> 10.1.1.1:80	http	911 bytes	<ul style="list-style-type: none"> • raw raw1 raw2 • as_html • hex • session_0001.part_01.html 160 bytes
2.	Fri Nov 19 23:29:14 2004	1 s	10.1.1.101:3179 -> 209.225.11.237:80	http	2217 bytes	<ul style="list-style-type: none"> • raw raw1 raw2 • as_html • hex
3.	Fri Nov 19 23:29:15 2004	1 s	10.1.1.101:3183 -> 209.225.0.6:80	http	3882 bytes	<ul style="list-style-type: none"> • raw raw1 raw2 • as_html • hex • session_0003.part_01.data 134 bytes
4.	Fri Nov 19 23:29:15 2004	1 s	10.1.1.101:3184 -> 209.225.0.6:80	http	3882 bytes	<ul style="list-style-type: none"> • raw raw1 raw2 • as_html • hex

Figure 3.8:

DataEcho Session Reconstruction Utility²⁶. DataEcho reconstructs historical web browsing and email traffic from captured network packets for monitoring insider security threats and policy compliance. It is however not nearly as comprehensive as Chaosreader (my highly subjective opinion) but nonetheless useful for session reassembly. It is a Microsoft .Net application so you need to have atleast v2.0 installed and it can be downloaded here²⁷. A sample screenshot showing a reconstructed webpage with an image from a captured packet stream is shown in Figure 3.9

3.3.3 NetworkMiner²⁸

NetworkMiner is a data carving tool for Windows. Infact it is much more than that. It is a Network Forensics Analysis Tool (NFAT). It lends itself to quite a lot - from being a passive packet capturing tool in order to detect operating systems, sessions, hostnames, open ports etc. (without putting any traffic on the network) to parsing PCAP files for off-line analysis and reconstructing transmitted files, directory structures and certificates.

NetworkMiner collects data about hosts rather than data regarding network traffic. It views data purely from the network host's perspective as opposed to the packet. So the output is

²⁶<http://data-echo.sourceforge.net>

²⁷<http://sourceforge.net/projects/data-echo/>

²⁸<http://networkminer.sourceforge.net>

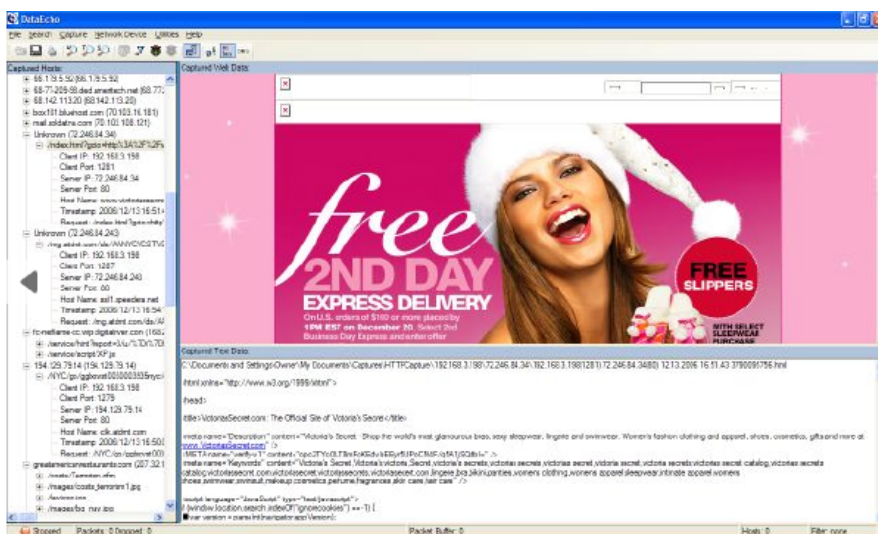


Figure 3.9:

human readable instead of frames and packets. NetworkMiner is also very useful in malware analysis since uploaded and downloaded files are extracted to disk.

3.3.3.1 Features

NetworkMiner features are listed below

- NetworkMiner performs OS fingerprinting based on TCP SYN and SYN+ACK packet by using OS fingerprinting databases from *p0f*²⁹ and *Ettercap*. NetworkMiner can also perform OS fingerprinting based on DHCP packets (which usually are broadcast packets) by making use of the *Satori*³⁰ OS fingerprinting database from *FingerBank*³¹. NetworkMiner also uses the MAC-vendor list from *Nmap*.
- NetworkMiner can extract files and certificates transferred over the network by parsing a PCAP file or by sniffing traffic directly from the network. This is a neat function that can be used to extract and save media files (such as audio or video files) which are streamed across a network. Supported protocols for file extraction are FTP, HTTP and SMB.

²⁹<http://lcamtuf.coredump.cx/p0f.shtml>

³⁰<http://myweb.cableone.net/xnih/>

³¹<http://www.fingerbank.org/>

- User credentials (usernames and passwords) for supported protocols are extracted by NetworkMiner and displayed under the 'Credentials' tab
- Another very useful feature is that the user can search sniffed or stored data for keywords. NetworkMiner allows the user to insert arbitrary string or byte-patterns that shall be searched for with the keyword search functionality.
- NetworkMiner also uses SPID to do protocol identification of a TCP session or UDP data. So instead of looking at the port number to guess which protocol is used on top of the TCP/UDP packet NetworkMiner will identify the correct protocol based on the TCP/UDP packet content. This way NetworkMiner will be able to identify protocols even if the service is run on a non-standard port.

3.3.3.2 Case Study 21: Processing with NetworkMiner

Installation

NetworkMiner can be downloaded here.³² At the time of writing, NetworkMiner has a version number of 0.89. Unzip the file and execute *NetworkMiner.exe*

Usage

Launching NetworkMiner, you are immediately presented with a GUI window with a network adapter drop down menu and a series of tabs totalling eleven in all giving you information on hosts, frames, files, images, credentials, sessions, DNS, parameters, keywords, cleartext and anomalies. See Figure3.10

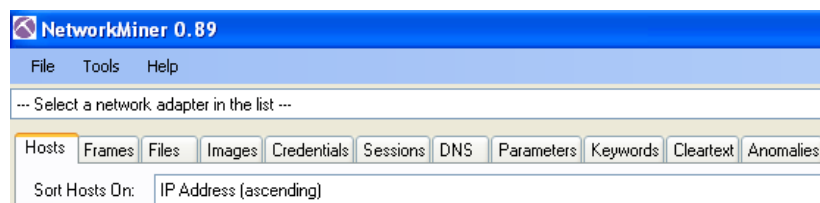


Figure 3.10:

³²http://sourceforge.net/project/showfiles.php?group_id=189429

For this case study, we will examine an earlier packet capture file *capture1.pcap* which happens to be a spyware capture. Remember that we have analyzed this file in Case Study 9. Simply load *capture1.pcap* into NetworkMiner using **File -> Open**. Figure 3.11 shows exactly what you will obtain

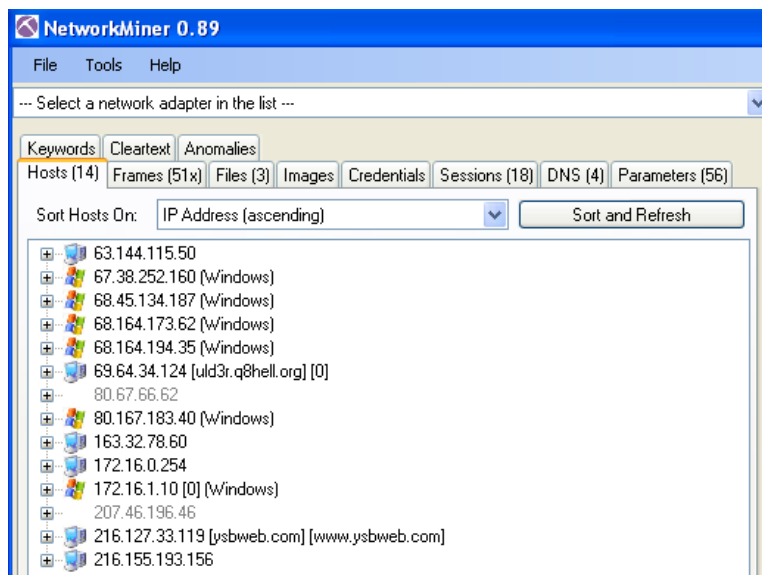


Figure 3.11:

It immediately gives a host view of the network, giving details about local and remote IP addresses with their corresponding operating system. You can click the + sign beside any host to drill down and obtain more information about hostname, MAC address, open ports, sent and received packets, incoming and outgoing sessions, etc. Most of the tabs are pretty self explanatory, the *Files and Images* tabs show reconstructed files and images within the packet stream, *Credentials* shows passwords that in the clear, *Sessions* shows just that - packet sessions, *DNS* gives DNS queries made, *Parameters* identifies web input variable parameters and strings passed to them, *Anomalies* for errant behaviour identification and *Cleartext* to narrow down keyword searches.

The *Keywords* tab is great for typical security data exploration. Using **Tools -> Delete Captured Data** before re-opening *capture1.pcap* file I added the keyword *.exe*, and then opened the capture. The result is shown in Figure 3.12

As we can see, the first two frames (70 and 71) points us to the *analiz.exe* file which we discov-

Frame...	Time...	Key...	Context	Source Host	Sour...	Destination H...
70	12/23...	.exe [...]D.>...E...f...anliz.exe.octet	172.16.1.10 [Windows]	UDP ...	68.164.173.62
71	12/23...	.exe [...]	*.....D.>...E...f...anliz.exe.octet	172.16.1.10 [Windows]	UDP ...	68.164.173.62
144	12/23...	.exe [...]	nload http://www.warees.net/bbnz.exe bbnz.exe 1...hurl3d.devilz.net	69.64.34.124 [jd3r:q8hell.org] [0]	TCP ...	172.16.1.10 [0]
148	12/23...	.exe [...]	URL: http://www.warees.net/bbnz.exe to: bbnz.exe...PRIVMSG #s02 :D	172.16.1.10 [0] [Windows]	TCP ...	69.64.34.124 [0]
164	12/23...	.exe [...]	com/ysb/exe/ysbinstall_1000489_3.exe...Transfer-Encoding: chunked.Co	216.127.33.119 [ysbweb.com]	TCP 80	172.16.1.10 [0]
167	12/23...	.exe [...]	com/ysb/exe/ysbinstall_1000489_3.exe...Transfer-Encoding: chunked.Co	216.127.33.119 [ysbweb.com]	TCP 80	172.16.1.10 [0]
170	12/23...	.exe [...]	com/ysb/exe/ysbinstall_1000489_3.exe...Transfer-Encoding: chunked.Co	216.127.33.119 [ysbweb.com]	TCP 80	172.16.1.10 [0]
183	12/23...	.exe [...]	ET /ysb/exe/ysbinstall_1000489_3.exe HTTP/1.1...User-Agent: Mozilla/4	172.16.1.10 [0] [Windows]	TCP ...	216.127.33.11
186	12/23...	.exe [...]	ET /ysb/exe/ysbinstall_1000489_3.exe HTTP/1.1...User-Agent: Mozilla/4	172.16.1.10 [0] [Windows]	TCP ...	216.127.33.11
199	12/23...	.exe [...]	28.6 KB to ysbinstall_1000489_3.exe @ 28.6 KB/sec...	172.16.1.10 [0] [Windows]	TCP ...	69.64.34.124 [0]
204	12/23...	.exe [...]	28.6 KB to ysbinstall_1000489_3.exe @ 28.6 KB/sec...	172.16.1.10 [0] [Windows]	TCP ...	69.64.34.124 [0]
205	12/23...	.exe [...]	ET /ysb/exe/ysbinstall_1000489_3.exe HTTP/1.1...User-Agent: Mozilla/4	172.16.1.10 [0] [Windows]	TCP ...	216.127.33.11
212	12/23...	.exe [...]	28.6 KB to ysbinstall_1000489_3.exe @ 28.6 KB/sec...	172.16.1.10 [0] [Windows]	TCP ...	69.64.34.124 [0]
214	12/23...	.exe [...]	D: Opened: ysbinstall_1000489_3.exe...	172.16.1.10 [0] [Windows]	TCP ...	69.64.34.124 [0]
216	12/23...	.exe [...]	D: Opened: ysbinstall_1000489_3.exe...PRIVMSG #s03 :[DDWNL0AD] Ope	172.16.1.10 [0] [Windows]	TCP ...	69.64.34.124 [0]

Figure 3.12:

ered as the spyware. The advantage of using the keyword search is the fact that it points us to the associated frame, seen in the Frames tab. see Figure 3.13 as well as source and destination host, source and destination port.

Frame	Time	Key	Context	Source Host	Sour...	Destination H...
70	12/23/2004 3:30:07 AM					
Ethernet2 [0-60]						
Destination MAC = 0001E10120E8						
Source MAC = 00D059AAAF80						
IPv4 [14-60]						
Total Length = 47						
TTL = 128						
Source IP = 172.16.1.10						
Destination IP = 68.164.173.62						
UDP [34-60]						
Source Port = 3735						
Destination Port = 69						
TFTP [42-60]						
71	12/23/2004 3:30:08 AM					
Ethernet2 [0-60]						
Destination MAC = 0001E10120E8						
Source MAC = 00D059AAAF80						
IPv4 [14-60]						
Total Length = 47						
TTL = 128						
Source IP = 172.16.1.10						
Destination IP = 68.164.173.62						
UDP [34-60]						
Source Port = 3735						
Destination Port = 69						
TFTP [42-60]						
72	12/23/2004 3:30:09 AM					
73	12/23/2004 3:30:09 AM					
74	12/23/2004 3:30:10 AM					
75	12/23/2004 3:30:10 AM					

Figure 3.13:

Frames informs us of the fact that there is indeed communication between 172.16.1.10 and

68.164.173.62 over UDP port 69 (TFTP). This pretty much gives it away.

3.3.3.3 Case study 22: TFTPgrab³³

This is another very nifty utility that comes in handy especially when analyzing TFTP protocol. TFTPgrab is a TFTP (Trivial File Transfer Protocol) stream extractor. It reads from Tcpdump/libpcap capture files and attempts to reconstruct data that has been transferred via TFTP. Reconstructed files are written to the current directory using the format,

```
src_ip.src_port-dst_ip.dst_port-filename
```

Non-alphanumeric characters in the filename are replaced with '_'. Since we have a TFTP connection, we can proceed to analyze it with this tool.

Installation

Download tftpgrab here³⁴ and extract

```
# wget http://pseudo-flaw.net/tftpgrab/tftpgrab-0.2.tar.gz
# tar xzvf tftpgrab-0.2.tar.gz
# cd tftpgrab-0.2
# ./configure && make && make install
```

Usage

To check for usage syntax type;

```
# tftpgrab -h
Usage: tftpgrab [OPTION]... [-r FILE] [EXPRESSION]
Reconstruct TFTP file contents from PCAP capture file.
With no FILE, or when FILE is -, read standard input.
-r PCAP file to read
-f overwrite existing files
```

³³<http://pseudo-flaw.net/content/tftpgrab/>

³⁴<http://pseudo-flaw.net/tftpgrab/tftpgrab-0.2.tar.gz>

```
-c print TFTP file contents to console
-E exclude TFTP filename when reconstructing
-v print verbose TFTP exchanges (repeat up to three times)
-X dump TFTP packet contents
-B check packets for bad checksums
-d specify debugging level
```

We proceed to the analysis of our capture file (*capture1.pcap*) thus;

```
# tftpgrab -r capture1.pcap
reading from file capture1.pcap, using datalink type EN10MB (Ethernet)
[warning] possible partial content for file: \
172.016.001.010.03735-068.164.173.062.00069-analiz.exe
```

The output is very obvious. We see that 172.16.1.10 port 3735 is communicating with 68.164.173.62 port 69 and calling the file *analiz.exe*. It tallies with frame 70 and 71 obtained with Network-Miner. Enough said.

3.4 Summary

This chapter reviewed active and passive security analysis through data mining and data carving techniques looking at statistical protocol Identification (SPID) algorithms. We also pored over advance techniques of packet reconstruction and reassembly for statistical analysis. In the next chapter we dig deeper into contextual network exploration.

Chapter 4

Security Data Exploration

Thus far, we have x-rayed methods of passive security through protocol agnostic packet reconstruction and reassembly techniques. This chapter takes it one step further by exploring the concept of packetized session reconstruction. As today's threats to enterprise networks are universally content based, when the need arises to view millions or even billions of network packets, the traditional approach, at best becomes insufficient and time consuming. We will take an in-depth look at a very valuable tool and one of its kind in network and security data exploration. It is called NetWitness® Investigator by NetWitness Inc. and it is absolutely free. Infact this tool is so noteworthy and perhaps revolutionary that I have devoted the entire chapter to its usage. It's a tool that is as relevant in digital forensics investigation as it is in data exploration and session reconstruction. But before we explore the tool, let's examine some methods of signature analysis

4.1 Signature Analysis

There are several methods of signature analysis techniques¹ used in identifying and classifying network traffic - from port, string match, numerical properties through to behavior and heuristics analysis methods.

¹<http://www.dpacket.org>

4.1.1 Port

Analysis by port is perhaps the easiest and most well known form of signature analysis. The rationale is the simple: the fact that many applications make use of either default ports or some chosen ports in a specified manner. An example is the POP3 protocol. The incoming POP3 uses port 110, and if it's POP3s, it uses port 995. The outgoing SMTP is port 25. But, since it is quite easy to detect application activity by port, this then becomes a weakness, particularly because many current applications masquerade themselves as other applications. The most common of these is Port 80, where many applications disguise as pure HTTP traffic.

In the case whereby some applications select random ports as opposed to fixed default ports, there is often some pattern involved in the port selection process - for instance, the first port may be random, but the next will be the subsequent one, and so forth. However in most cases the port selection process may be completely random. For all these reasons, it is often not feasible to use analysis by port as the only method for application identification, but rather as a form of analysis to be used together with other techniques.

4.1.2 String Match

String match analysis involves the search for a sequence of textual characters or numeric values within the contents of the packet. String matches may consist of several strings distributed within a packet or several packets. For instance, many applications declare their names within the protocol itself, such as *Kazaa*, where the string "Kazaa" can be found in the User-Agent field with an HTTP GET request. From this example, the relevance and importance of deep packet inspection becomes obvious in order to properly classify the packet. If analysis is performed by port analysis alone, then port 80 may indicate HTTP traffic and the GET request will further corroborate this observation. Since the User-Agent field information is missing however, the analysis will result in misleading and false classification i.e., *HTTP* and not *Kazaa* as depicted in Figure 4.1 which shows the analysis of *Kazaa* string match analysis.

This example further emphasizes that a combination of signature analysis techniques is required for assuring proper classification.

4.1.3 Numerical Properties

Analysis by numerical properties takes cognizance of arithmetic and numerical characteristics within a packet or several packets. Some examples of properties analyzed include length of

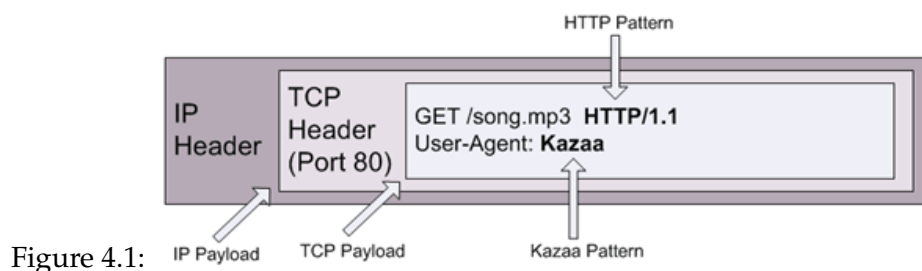


Figure 4.1:

payload, number of packets sent in response to a specific transaction, and the numerical offset of some fixed string or byte value within a packet.

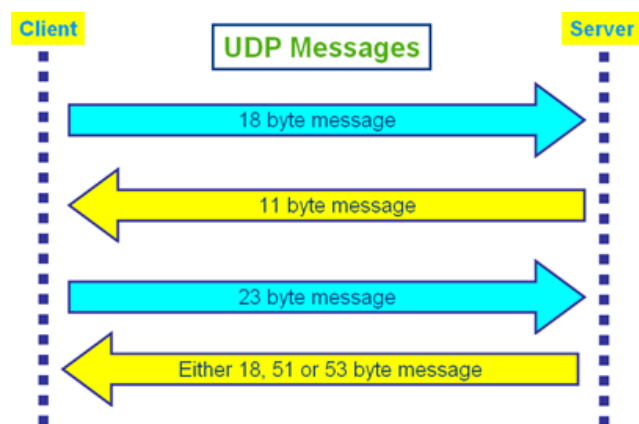


Figure 4.2:

For example, consider the process for establishing a TCP connection using some UDP transactions in *Skype* (versions prior to 2.0). The Client sends an 18 byte message, expecting in return an 11 byte response. This is followed by the sending of a 23 byte message, expecting a response which is 18, 51 or 53 bytes. See Figure 4.2 Similar to analysis by port and analysis by string match, analysis by numerical properties alone is inadequate, and can often lead to a lot of false positives.

4.1.4 Behavior and Heuristics

Behavioral analysis refers to the way a protocol acts and operates. Heuristic analysis typically boils down to the extraction of statistical parameters of examined packet transactions. Often, behavioral and heuristic analysis are combined to provide improved assessment capabilities.

For example, actions leading to other actions can clearly indicate a behavioral pattern which can be traced, as in the case where an active UDP connection eventually transforms into a TCP connection (using the same IP and port settings).

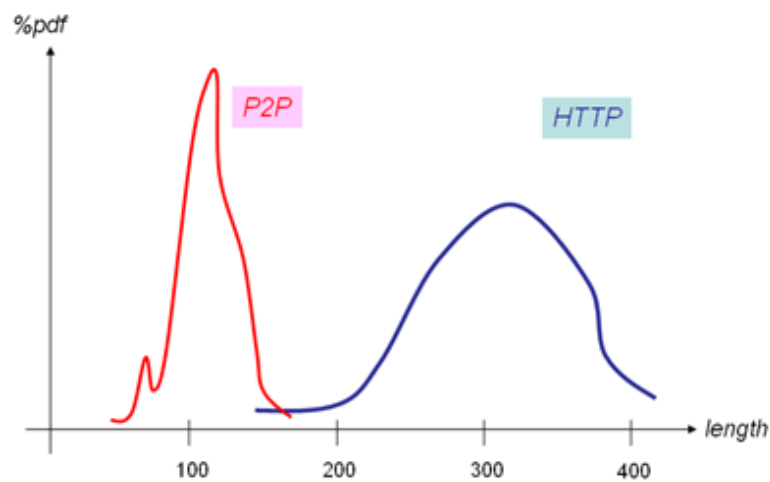


Figure 4.3:

Another example of behavior and heuristic analysis is shown in Figure 4.3, which compares HTTP and a typical P2P file sharing application. If the packet length histogram alone is examined while ignoring the file download or upload transaction itself (which tends to use large packet lengths), it becomes apparent that while pure HTTP packets tend to concentrate around a few hundred bytes in length, P2P control layer information tends to use shorter packet lengths. In this way, by examining some short-term statistics, it is possible to conclude whether a port 80 connection carries pure HTTP traffic or other P2P-related traffic.

4.2 Application Classification

There are certain situations where there are weak signatures, therefore not robust enough even though signatures are developed with the intention to uniquely and completely identify its related application or protocol. Classification of applications then becomes a problem.

False positives is the basic terminology referring to misclassification – or in simple terms - the likelihood that an application will be identified as something it is not. The implication is that it may lead to inaccurate conclusions. A typical example of such an inaccurate conclusion could be the lowering of priorities of time-sensitive streaming traffic and resultant introduction of unwanted latency or even packet loss. Consequently, when developing signatures, every effort must be made to achieve a false positive value of 0%. A common way to strengthen a weak signature is to use a combination of more than one pattern.

False negatives refers to situations where it is not possible to consistently identify an application – sometimes the identification is classified, while other times it is missed by the classification method. There are various reasons for this occurrence, the most common being that some applications can accomplish similar outcomes in several ways in different deployment scenarios. For example, some applications will behave differently if the client software operates through a proxy or firewall compared with the simpler case in which the client interacts with the web directly. Therefore, in these not so regular cases, if the signature was developed under the assumption of direct communications, it is likely that the application will not be correctly classified.

4.3 NetWitness Investigator²

NetWitness® Investigator provides security analysts and forensics investigators the ability to perform unparalleled free-form contextual analysis of raw network data. It was originally developed for the U.S. Intelligence Community but now used extensively by Law Enforcement, Defense, and other public and private organizations.

4.3.1 Primer

Investigator is a security intelligence application that audits and monitors all traffic on a network. To solve the concern of analyzing huge packet traces, it first flips the problem on its

²<http://www.netwitness.com/products/investigator.aspx>

head. It does this by not only creating a forensically valid network packet databases but a comprehensive log of all network activities and then interprets these activities into a format that network and security analysts alike can quickly recognize and understand. This is made possible because all Investigator's internal logic is performed at session level.

Novice and expert analysts alike can use Investigator to pivot terabytes of network traffic easily to delve deeply into the context and content of network sessions in real-time thereby making advance security analysis that once took days, take only minutes. It is this intersection of network metrics, rich application flow, and content information that differentiates Investigator from most other applications.

NetWitness converts each protocol into a common language, so knowledge of protocols is no longer needed. Performing analysis using Investigator can be as simple as looking for user names, e-mails, applications, resources, actions, computer names. Investigator can also be integrated with NetWitness Live³ to have access to multi-source threat intelligence. Some of its inherent capabilities are represented in Figure 4.4

4.3.2 Features

- Below is a list of the supported features
- New! Supports NetWitness® Live
- SSL Decryption (with server certificate)
- Interactive time charts, and summary view
- Interactive packet view and decode Hash Pcap on Export
- Enhanced content views
- Real-time, Patented Layer 7 Analytics
 - Effectively analyze data starting from application layer entities like users, email, address, files , and actions.
 - Infinite, free-form analysis paths
 - Content starting points
 - Patented port agnostic service identification

³<http://www.netwitness.com/products/live.aspx>

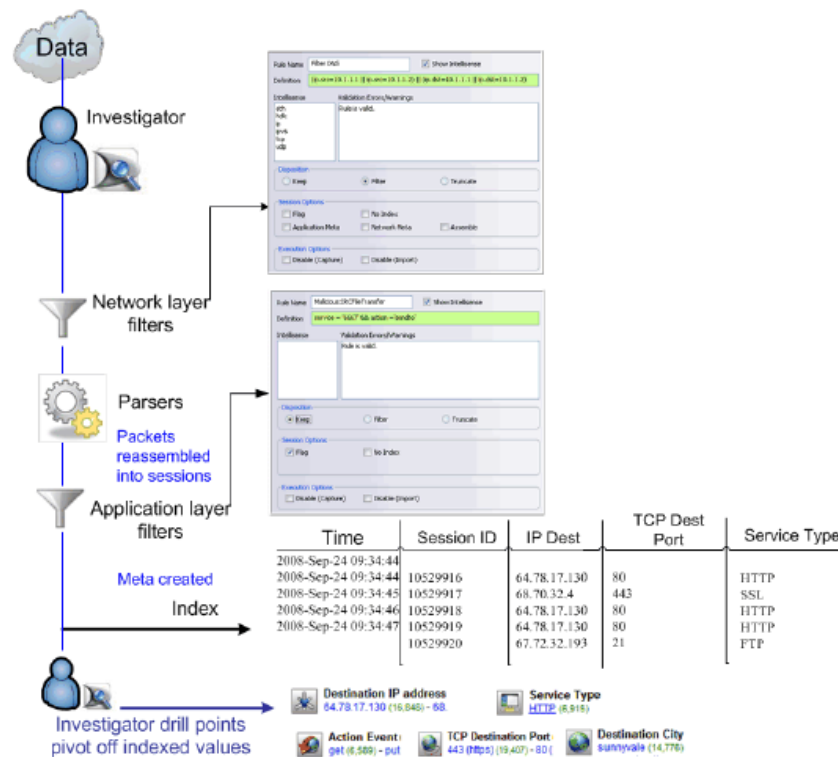


Figure 4.4:

- Extensive network and application layer filtering (e.g. MAC, IP, User, Keywords, Etc.)
- IPv6 support Captures live from any wired or wireless interface
- Full content search, with Regexp support Exports data in .pcap format
- Imports packets from any open-source, home-grown and commercial packet capture system(e.g. .pcap file import)
- Bookmarking & History
- Tracking Integrated GeoIP for resolving IP addresses to city/county, supporting Google Earth visualization

4.3.3 Concepts

Concepts relating to the use of Investigator are described below

Parser: Is a program, usually part of a compiler, that receives input in the form of sequential source program instructions, interactive online commands, markup tags, or some other defined interface and breaks them up into parts (for example, the nouns (objects), verbs (methods), and their attributes or options) that can then be managed by other programming. A parser may also check to see that all input has been provided that is necessary. The custom-defined Search and Flex parsers can be configured by the user, extending your analysis capabilities considerably.

Drill: The action of clicking on a link to the next level of detail. A drill point refers to focusing the analytic view on a specific subset of a collection defined by a particular metadata element.

Collection: A collection is a logically related group of packets. It consists of one or more capture or remote device files. A collection can be created either by the live capture capability within Investigator, by importing existing pcap files, or by connecting to another NetWitness appliance.

Collection Summary: A scalable high-level view of the characteristics (session count, session size, packet count) of a selected collection for a specific timeline.

Navigation View: The central mechanism for drilling into the extracted metadata.

Search View: The mechanism for locating individual sessions with specified string values or regular expressions.

Bookmark: Analogous to a web browser bookmark, Investigator bookmarks let the user create a reference to a single session or a group of sessions. A single-click mechanism returns the user to the selected session(s).

Breadcrumb: Breadcrumbs are a way to maintain a path from the root of the collection to the current drill point. The user can click on any element within the breadcrumb to jump back to that point in the drill path.

View: The relative position you are using to look at the captured data, in descending order: Summary, Collection, Report, Session, Search, Content

Sessions: A group of related data packets. These packets are grouped into sessions based on the transactional nature of the communication, as in the client/server request and response.

Content: The actual information or object represented in the data capture. The content of a session consists of every packet captured for that session. Session content can be viewed by its content type (web, e-mail, IM, text, etc.).

Metadata: Specific data types (Service Type, Action Event, Source IP Address, etc.) used by the parsers to count and itemize in the captured data. For instance, in an FTP session, the FTP parser will produce metadata such as login name, password, and file operations including get, put, or delete.

Index: Indexes are internal NetWitness data structures that organize the metadata elements of sessions and are generated during data processing for a collection. The content of the index, and consequently the metadata elements that are displayed in the Navigation view, are controlled by settings in effect during collection processing. Rebuilding a collection will regenerate the index.

4.3.4 Collection Navigation

Most security data exploration will be done in a collection. As a result, an understanding of the navigation features in Investigator is imperative so as to determine our location in the collection. Because there are multiple data items that can be drilled into at any point, it is almost trivial to beat about the bush within the application.

4.3.4.1 Navigation View

Figure 4.5 shows the navigation view of Investigator. The tab for the selected collection shows a listing of the processed reports (e.g. **IP Protocol**, **Service Type**, **Action Event**, etc.). Each of the report types lists report values and their associated session counts. In effect, Investigator blurs the distinction between navigation and queries.

4.3.4.2 Navigation Toolbar

The appearance of the collection reports and the data contained are determined by the combination of selections you make on the Navigation toolbar. For example, the Time Graph allows

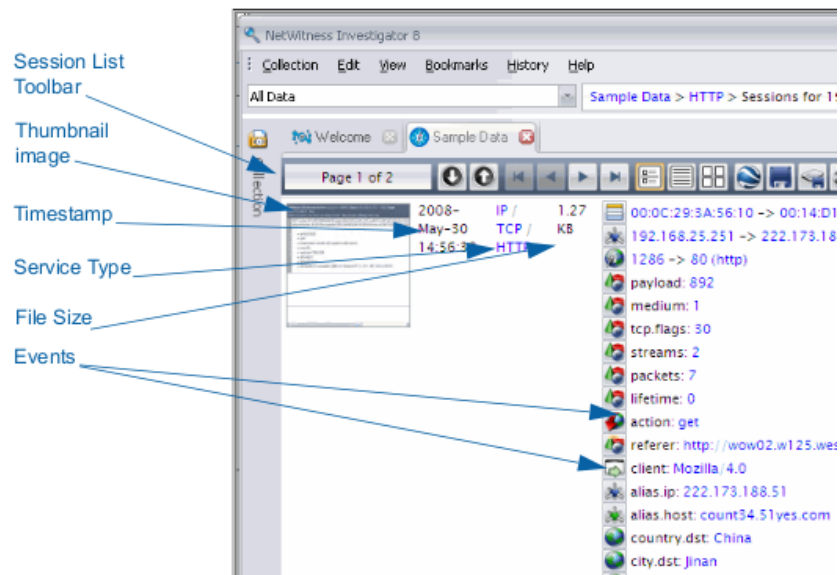


Figure 4.6:

4.3.4.4 Content View

A separate pane displays the content detail for a session if the Thumbnail image is clicked. You can select any of the formats from the Content Toolbar. We can continue to explore data through drilling into specific items, searching the session for a particular term, string, or other values.

4.4 Investigator Operations

Investigator can be downloaded here⁴. Installation is very direct, just follow the prompt and start Investigator. Figure 4.7 is the graphical display of the main window.

This window enables you to create new collections and manage the existing saved collections. We will expand further on the intricate features of Investigator by analyzing our initial packet captures from previous chapters through the following case studies.

⁴<http://download.netwitness.com/download.php?src=DIRECT>

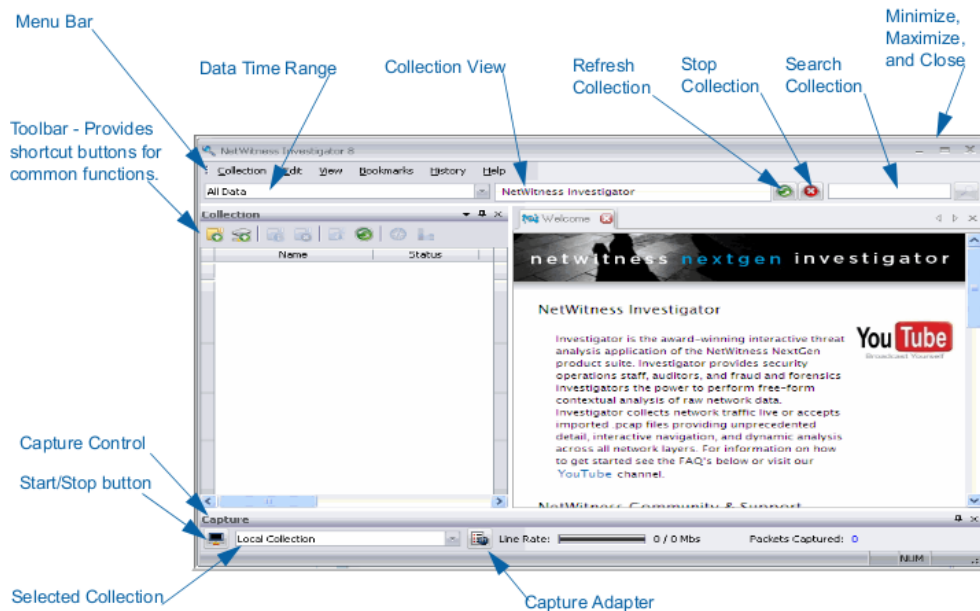


Figure 4.7:

4.4.1 Case Study 23: Basic Session Reconstruction

We start our case study by opening a sample http pcap⁵ file in Investigator⁶ thus;

- Click **Collection** -> **New Local Collection**
- Enter a Collection Name, say *httpdownload* and click **OK**
- Highlight *httpdownload* folder in the Collection pane,
- Click **Collection** -> **Import Packets**
- Navigate to the folder and pick the required *pcap* file.

In the Collection pane the status of the import will be shown under the **Status** column and once done, it will display *ready* from where you can double-click to display the content of the folder in the navigation window.

⁵http download with images

⁶<http://inverse.com.ng/trace/capture5.pcap>

Navigation consists of reports and values. Reports are categories of data that has been extracted such as source and destination IP, service type, hostname aliases, user accounts, etc and the values under each report represent the data extracted from the actual sessions. Every item within the navigation view is er navigable. Start by selecting any value or a combination of values to reconstruct the entire data set on that criteria. For instance from the navigation view:

- Select *HTTP* under **Service Type**

Investigator will dynamically reconstruct only sessions relating to the HTTP service type. Now all reports shown and all values of the report reference only HTTP sessions. We can further select additional values, such as *jpg* under **Extension**, the resulting data displayed is only data from that extention over the HTTP service. This effectively shows all jpeg files downloaded on the network. See Figure 4.8

In a large packet capture, if you want to see a report on the entire end to end HTTP transaction between two hosts, your criteria may include **HTTP Service Type**, **Source IP Address** and **Destination IP Address**. At any point in time you can step back to a previous selection within the breadcrumb window. We step back to our **HTTP Service Type** by clicking *HTTP* in the breadcrumb window as show in Figure 4.9

4.4.2 Case Study 24: Malware Investigation

Let us analyze our previous *capture2.pcap* packet capture file from Case Study 10. We follow the same procedure to import the pcap file. We immediately notice that there are 34 *HTTP*, 1 *DNS* and 1 *IRC* requests. We also observe 33 *CGI* extensions, 3 filenames and a TCP Destination port 5050. See Figure 4.10

As seen, Investigator in effect blurs the distinction between navigation and queries. So far we have been clicking on values. The report themselves are also selectable. For instance, if we wanted to view the content of *proxy.cgi* under **Filename** we simply select it. This details the entire session of the report.

There are some Investigator shortcuts that you need to be aware of. Firstly, if you select a collection that is already opened, Investigator will open an additional tab. This is useful when you like to explore multiple paths simultaneously. In addition use of the [ctrl] key while selecting any navigation option will launch it in a new tab with the destination reconstruction loaded. Furthermore, when you click on the report icon, it presents you with several choices (Figure 4.11) to create a custom query report using regular expressions and sub expressions.

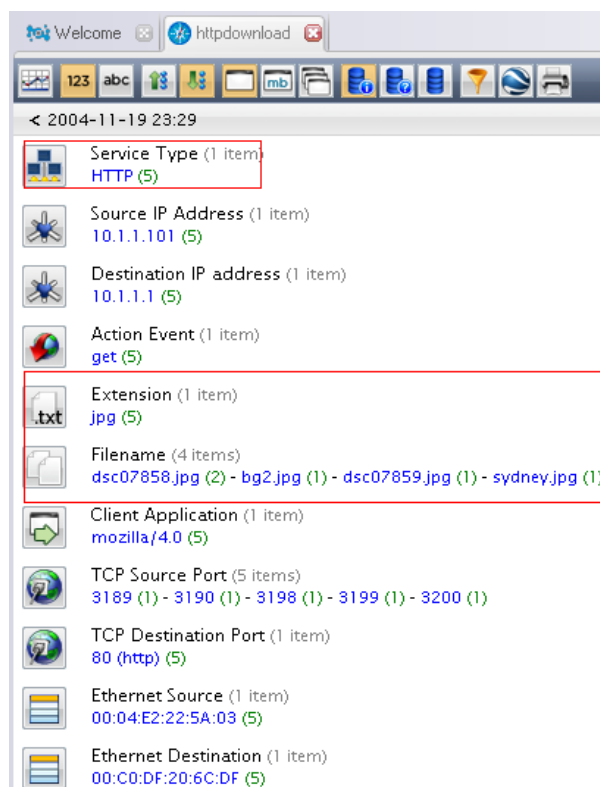


Figure 4.8:

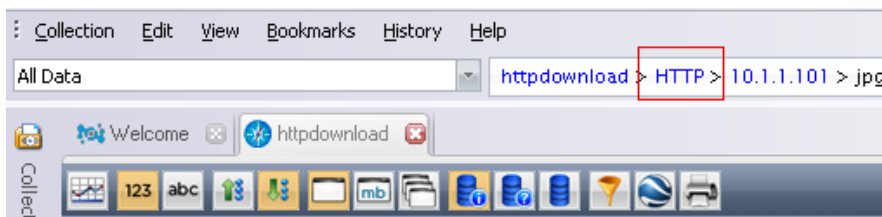


Figure 4.9:

Let's now examine some session content. On any drill value, if you select the number of matching sessions, Investigator will display the session list. To continue our investigation, we drill down to *proxy.cgi* report and click the value beside it which is 6 in our case. What we obtain is shown in Figure 4.12

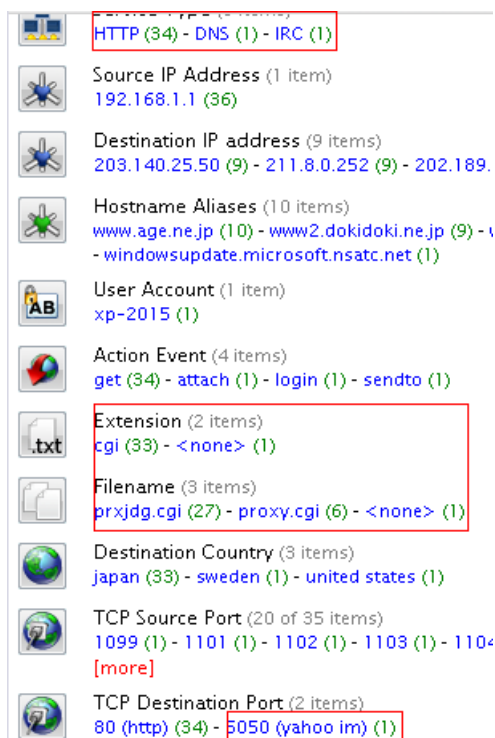


Figure 4.10:

Investigator gives you the opportunity to work with reconstructed data in different ways. The session list has three viewable options. The first shown in Figure 4.12 is the **hybrid** view and it provides the most session details under the *Events* column (which shows a listing of in order as data was detected of all extracted information within the session) as well as a thumbnail of the re-constructed session. It also has an **event** view which displays a condensed listing of matching sessions - used when an analyst needs to look for each session in order of time. Finally it provides a **thumbnail** view which displays a re-constructed thumbnail of the sessions - used by an analyst to quickly check known session types for the presence of content. I tend to lean more towards the default which is the hybrid view.

I have highlighted what is pretty much of importance to our analysis. Observe connectivity with destination IP 202.189.151.5 with full location `www.kinchan.net/cgi-bin/proxi.cgi`. Once you google the location as depicted in Figure 4.13

Notice the reference to a *Worm/SdBot.24A3@net* worm. This definitely is not good. Navigating

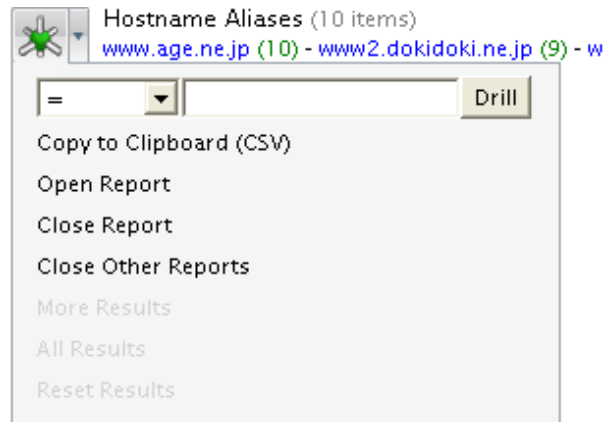


Figure 4.11:

	Time	Service	Size	Events
	2006-Oct-05 02:25:17	IP / TCP / HTTP	2.00 KB	<ul style="list-style-type: none"> 00:40:63:D7:A5:CA -> 00:30:48:11:B6:BC 192.168.1.1 -> 202.189.151.5 1109 (kpop) -> 80 (http) payload: 1.475 medium: 1 tcp.flags: 27 streams: 2 packets: 10 lifetime: 0 action: get directory: /cgi-bin/ filename: proxy.cgi extension: cgi alias.ip: 202.189.151.5 alias.host: www.kinchan.net content: text/html country.dst: Japan latdec.dst: 36.000000 longdec.dst: 138.000000 org.dst: MIXED MEDIA, INC. domain.dst: mxm.net
	2006-Oct-05 02:25:17	IP / TCP / HTTP	2.00 KB	<ul style="list-style-type: none"> 00:40:63:D7:A5:CA -> 00:30:48:11:B6:BC 192.168.1.1 -> 202.189.151.5 1108 -> 80 (http)

Figure 4.12:

to the link⁷ gives more details about the worm. See Figure 4.14
 The overview of the worm reads:

⁷<http://www.anchiva.com/virus/view.asp?vname=Worm/SdBot.24A3@net>

Google

Search: the web pages from Nigeria

Web [+ Show options...](#) R

[#608243 - Pastie](#)
7 Sep 2009 ... <http://66.197.169.200/cgi-bin/pp.cgi> ... <http://www.lintang.net/cgi-bin/prx.cgi> ... v2.30 <http://www.kinchan.net/cgi-bin/proxy.cgi> v2.28 ... pastie.org/608243/wrap - [Cached](#) - [Similar](#) - [🗨](#) [📄](#) [🗕](#)

[List of prxjdg.cgi - ProxyJudge, Updated weekly](#)
3 Feb 2008 ... <http://www.yoyolee.net/ck.cgi> v2.34 <http://www.mcreate.net/cgi-bin/envchk/prxjdg.cgi> v2.30 <http://www.kinchan.net/cgi-bin/proxy.cgi> ... web.freerk.com/proxyjudge/prxjdg.htm - [Cached](#) - [Similar](#) - [🗨](#) [📄](#) [🗕](#)

[charon help! - Proxy List Forum](#)
8 posts - 2 authors - Last post: 8 May 2005
<http://www.kinchan.net/cgi-bin/proxy.cgi> · <http://age.ne.jp/x/maxwell/cgi-bin/prxjdg.cgi> · <http://www.age.ne.jp/x/maxwell/cgi-bin/prxjdg.cgi> ... www.dcsproxy.com/proxy-questions/79-charon-help.html - [Cached](#) - [Similar](#) - [🗨](#) [📄](#) [🗕](#)

[Worm/SdBot.24A3@net](#)
<http://69.59.137.236/cgi/prxjdg.cgi> ... <http://tn0828-web.hp.infoseek.co.jp/cgi-bin/proxyjudge.cgi> ... <http://www.kinchan.net/cgi-bin/proxy.cgi> ... www.anchiva.com/virus/view.asp?vname..24A3@net - [Cached](#) - [Similar](#) - [🗨](#) [📄](#) [🗕](#)

Figure 4.13:

Worm/SdBot.24A3@net

Overall Risk Rating: Medium	Distribution Potential: High	Damage Potential: High
In The Wild: Yes	Vulnerability: MS00-035	Discovered: 2007-06-01
Size: 154,624 Bytes	Language: English	Compression: Unknown
Port(s): Random	Payload: None	
Aliases: Backdoor.Win32.SdBot.bjd (Kaspersky), WORM_SDBOT.EIC (Trendmicro), W32/Tilebot-JT (Sophos)		
Affected Platforms: Windows NT, Windows XP, Windows 2003, Windows 2000, Windows 95/98/ME		

Figure 4.14:

This worm belongs to the family of *Worm/SdBot* which is known to propagate via network shares and known vulnerabilities. It may arrive in the affected system as a result of unsecured system. It may also arrive as a result of successful exploit attack.

It propagates mainly by dropping a copy of itself into default shares of machines

connected to the network. It also takes advantages of SQL 'sa' account with blank password vulnerability to propagate across the network.

Furthermore, this worm has a backdoor capability. It connects to a remote IRC channel and joins a specific channel where it listens for commands from a remote attacker. The remote attacker can then perform malicious routines on the affected system.

This worm has also the capability to communicate to a remote server via HTTP and download proxy settings which, in turn, may convert the affected system into a proxy server.

4.4.3 Case Study 25: More Session Reconstruction

We now want to reconstruct a typical client/server session with authentication. For this case study, we use the Demo Collection that is already part of Investigator. It can also be downloaded here⁸

Depicted in Figure 4.15 and 4.16 is the navigation view of the packet.

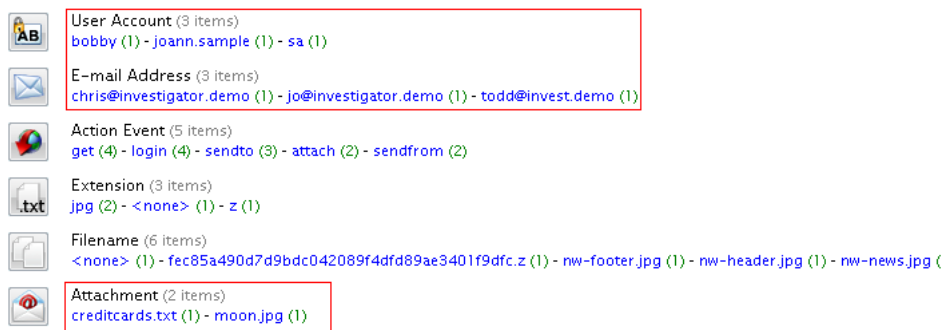


Figure 4.15:

I have highlighted a couple of interesting data. Observe *user accounts*, *email addresses*, *email attachments*, *passwords*, and even some *sql query* commands. Click on **Password** to open up its report then click on the value beside it. Figure 4.17 is a screenshot of the session view.

⁸http://inverse.com.ng/trace/netwitness_sample_pcap.pcap

```

Full Name (1 item)
bob smith (1)

Password (1 item)
mypassword1 (1)

Subject (1 item)
welcome to netwitness investigator (1)

Referer (1 item)
http://www.netwitness.com/samplepage/ (2)

Crypto (2 items)
aes256-ctr (1) - rsa-with-rc4-128-md5 (1)

SQL Query (20 of 35 items)
dbcc showfilestats (1) - dbcc sqlperf(logspace) (1) - exec dbo.dt_verstamp006 (1) - exec discovery_
n'discovery_db' (1) - exec sp_dbcmptlevel n'fe_db' (1) - exec sp_helpmergepullsubscription (1) - ex
sp_msdbuseraccess n'perm', n'discovery_db' (1) - exec sp_msdbuseraccess n'perm', n'fe_db' (1)
sp_tables (1) - execute master.dbo.xp_regread n'hkey_local_machine', n'software\microsoft\mssql
master..sp_mssqladm70_version else select 0 (1) [more]

```

Figure 4.16:

We can view the entire reconstructed session by clicking on the thumbnail. This opens in another navigation window which can be detached from the main Investigator menu as shown in Figure 4.16. From here we can read the mail together with the header. Investigator gives the option of extracting attachments in the reconstructed data within itself, with a default or external application and saved to a location on the hard disk. Investigator can reconstruct any kind of network traffic from instant messaging over common IM protocols to audio and video content over common protocols. Full extractions and reconstructions however depend on protocol versions, complete capture of both sides of the conversation and myriad of other factors.

If for any reason the session cannot be rendered, there will always be a *text*, *hex* and *packet* views of the session in the navigation toolbar. Investigator will try as much as possible to reconstruct packets as best as it can while suppressing scripting and executable content. In such a case, this may affect the way the session is displayed.

Investigator provides a mechanism for exporting all sessions within the session and content views to standalone pcap files. In addition to pcap, Investigator supports *raw*, *xml*, *csv*, *text* and the netwitness proprietary data format *nwd*. Also with the content view, it is possible to open the session with any other tool associated with a pcap file, such as Wireshark. Lastly, there is a **Google Earth** icon present in both the navigation and session views. This allows the visualization of a group of sessions by sending raw xml output to google earth for display using GeoIP information gathered during analysis.

There you go, full reconstruction of packetized session data. This application should be con-

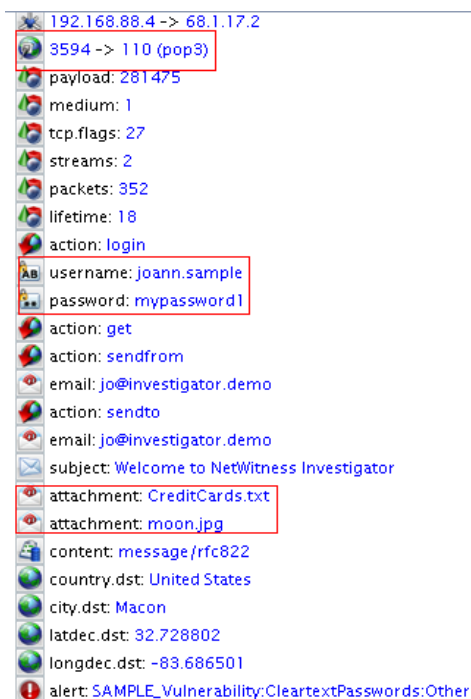


Figure 4.17:

sidered packet analysis on steroids.

4.4.4 Case Study 26: Searching Content

In the top right hand corner of Investigator is the search bar. Basic searches can be conducted from this dialog box or you can select the search icon to bring up the search window. Under the **Preferences** dialog you can choose to *Search Content*, *Search Metadata*, *Decode Sessions* and perform *Case Insensitive* searches. The *Decode Sessions* option will decode sessions prior to searching which includes amongst others gzipped and uuencode sessions. Any type of string, byte sequence or regular expression can be searched for.

For this case study we go back to our *capture1.pcap* spyware file. Import it into Investigator if you haven't already done so and select the search icon. Click the **Preferences** dialog and make sure all the options are checked. In the search dialog enter the search item *.exe* then click **Search** button. Figure 4.18 shows the result of the search

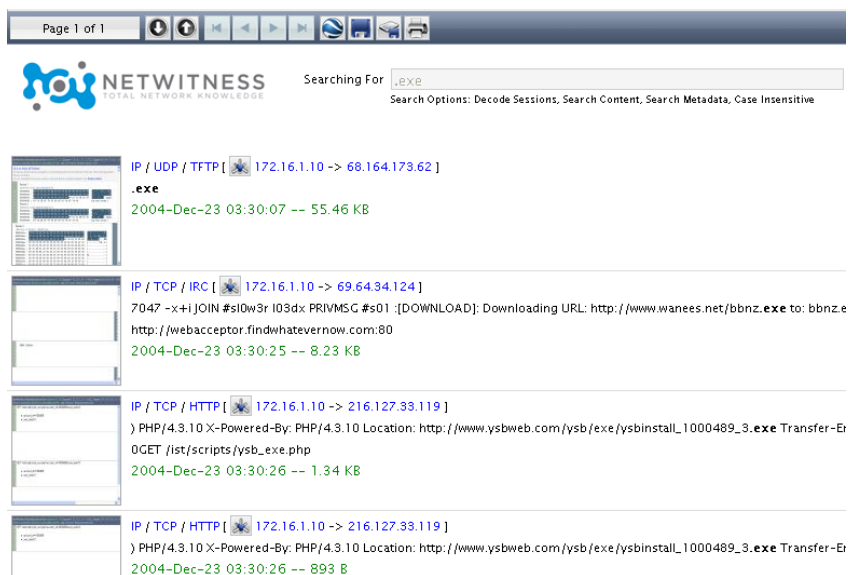


Figure 4.18:

We can see all executables downloaded together with full source and destination IP and size of executable. There is also individual thumbnails beside each packet stream. The first one grabs my attention as it is over TFTP which is a UDP protocol. I then select the thumbnail and there goes our friend *analiz.exe* lurking around. See Figure 4.19

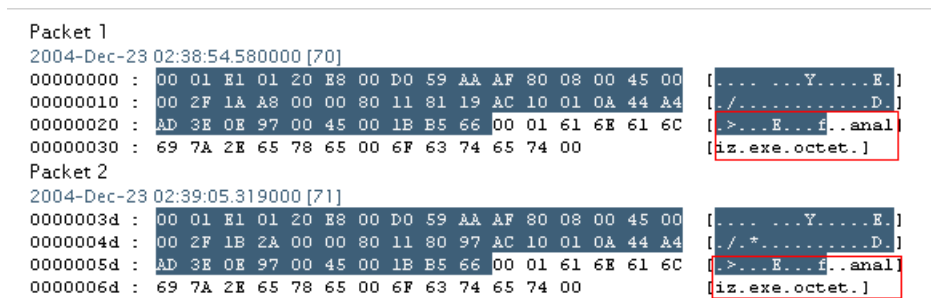


Figure 4.19:

Note: Content searching is context specific to the drill allowing you to optimize your searches. There is also an option on the top left for advance searches. With the **Advanced Search** you

can write and save regular expression searches that you perform often, many even ship as examples in the **Search Name** dialog box. Also in the **Advanced Searches**, there is also an option for *Search & Export* matches into a new collection in one simple step.

4.4.5 Case Study 27: Packet Capture

Yes we have basically analyzed pcap files with Investigator, but can also be set up to capture packets directly off the network, even though I personally don't subscribe to this method. In this case study, we examine how Investigator can be configured to act as a packet sniffer. Remember that you need to install the latest version of Winpcap packet capture library in Windows prior to capturing.

- First enable **Capture Bar** under **View** menu
- Below you will see the **Capture** bar at the bottom of Investigator
- Select the **Configure the capture adapter** icon
- Select your network adapter interface and other options such as *Max Disk Usage* under **Capture** tab and click **OK**
- Then select your Collection where Investigator will send captures to or create a new one by clicking on the **Start capturing packets to selected location** icon
- Select the same icon to stop capture when done.
- You can then proceed to Collection to view your data capture by selecting the *fid* folder

Figure 4.20 is the screenshot of Investigator in action capturing to a location called *fid*

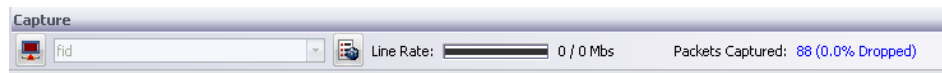


Figure 4.20:

4.4.6 Investigator Options

Clicking on Options under Edit allows you to customize Investigator in more ways than one. It allows you to customize the **General** look and feel of Investigator, its interface **Display** format,

enabling and suppressing **Reports**, **Capture** method, **Process** various application parsers, view **Audio Codecs** and setting up **Advance** configurations such as *indexing options*, *threads* and *log messages*.

4.5 Summary

This chapter discussed deep inspection and packet reconstruction through the creation of valid network packet databases and a comprehensive log of all network activities. Through the use of Netwitness Investigator and its internal session reconstruction logic, we were able to interpret these activities into a format that is well understood by network and security analysts alike. Hopefully this application will become a valuable component of your security analysis toolkit and network defence.

Part III

VISUALIZE

Chapter 5

Security Visual Mapping

What is information visualization and how does it relate with data mining? Information visualization is emerging as an important fusion of graphics, network and security visualization as well as human-computer interaction (HCI). Scores of innovative visualizations tools and techniques for 1-, 2-, 3- and multi-dimensional data have been developed, along with creative designs for temporal, hierarchical, and network data. This third part of the book examines the various tools, design and logic as human factor evaluation for visualization.

Security visual mapping and transformation is a technique to create and manipulate a graphic representation and are more often than not employed to create a two- and three-dimensional graphics of a typical security dataset. As a matter of fact, data transformation is concerned with converting free form raw data source to graphics and visuals for the purpose of decision making. As we explore the different ways of data representation and transformation, we should note that some transformation tools will only be appropriate for some specific scenarios while others will be more generic data representation formats. Nevertheless the goal of data transformation is to understand the underlying data relationship and not necessarily the data itself.

Some representation tools further allow the animation of the graphic through a specific data dimension sequence. Transformation tools such as bar and pie charts as well as line graphs have been used for decades, and to this day most organizations still rely on the them for, by and large most of their information and data representation needs. But with advancements in new graphical and visualization techniques, organizations have realized that they can quickly and productively employ and adopt a few graphical illustrations to substitute pages of tabular security reports. Some organizations even use these graphics to supplement their traditional

reports.

This chapter discusses the techniques of data mapping as well as graphical data representation. But before we go head on, we start by examining a subtle but non the less important concept - visual perception.

5.1 Visual Perception

“For the most part, we do not first see, and then define; we define first and then see” - Walter Lippmann

What is visual perception and why do we comprehend some forms of data presentation faster than others? The answer is important to security analysts as this aids in the decision-support process.

Analysis and effective communication is key to making sense of information and data. Security graphs and to a lesser degree, tables communicate visually. The level of effective and efficient communication of data presentation is directly proportional to the use of visual perception. Presentation of data isn't just making pretty graphs and charts, it is far deeper and to do that, you must realize what works, what doesn't, and why. What is needed is a vivid illustration of raw dataset in clear and concise manner that will make end users see and understand the big picture at a go.

The author Colin Ware tells us in the preface to his book¹ why we should be interested in the understanding of visual perception.

“Because the human visual system is a pattern seeker of enormous power and subtlety. The eye and the visual cortex of the brain form a massively parallel processor that provides the highest-bandwidth channel into human cognitive centers. At higher levels of processing, perception and cognition are closely interrelated, which is the reason why the words "understanding" and "seeing" are synonymous. However, the visual system has its own rules. We can easily see patterns presented in certain ways, but if they are presented in other ways, they become invisible.... If we can understand how perception works, our knowledge can be translated into rules for displaying information. Following perception-based rules, we can present our data in such a way that the important

¹Information Visualization: Perception for Design

and informative patterns stand out. If we disobey the rules, our data will be incomprehensible or misleading.”

The sense of sight (vision) is much like a superhighway when compared with the sense of smell, hearing, taste, and touch which are more like cobblestone paved narrow pathways.

5.1.1 Memory Limitations

Whilst we will not go in-depth into the theory behind visual perception, it is worth noting that when we look at a physical object, we don't actually see the object, we see light reflected off their surfaces. Much like computers, our brains use various types of mechanisms to store information while it's being processed and, in some cases, to store it for future reference. Pertinent to say there are three basic types of memory in the brain: *iconic*, *short-term*, and *long-term*.

Iconic memory is not dissimilar to the graphics buffer of a PC. Its job is to briefly store what the eyes see until it is moved either to short-term memory referred to as working memory for conscious processing or is discarded as irrelevant.

Short-term memory is like the RAM in a PC. It is readily accessible for high-speed processing but limited in its capacity to retain data. Information that is classified essential for later reference is taken from short-term memory into long-term memory where it's stored and labeled and cross referenced in one or more ways for later permanent storage, just like on the hard disk of a PC. Despite the common misconception of retaining memories of everything we experience in life, we really do not.

Long-term memory is limited in its capacity, but is very flexible in application. It maintains and constantly indexes a complex network of memory relationships in an effort to keep them available and useful.

Now to the point. Everything about sense-making is done in *Short-term memory*. New data is passed through the senses and old data is swapped in from *long-term memory*, working quicker than the speed of thought to help us in making sense of raw data. Given man's superior cognitive abilities, it's amazing to know that all of this is done using short-term memory that can only hold from three to seven slices of data at a time. This limitation must be considered when designing any data presentation.

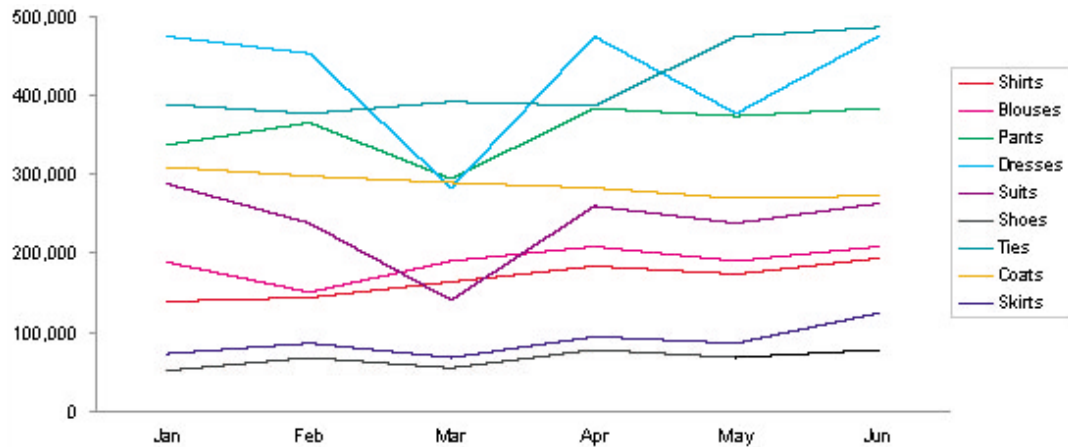


Figure 5.1:

Figure 5.1 depicts a notable problem in plotting graphs. Taking a look at the nine differently coloured lines, it is not easy to break down the the meaning of the separate datasets because they can't be held in *short-term memory* simultaneously. Most times attention is shifted back and forth between the legend and the lines of the graph to be reminded of what each line represents. Therefore, for any sense making of the graph to be made, a cap or limit must be placed on the number of data components. This limit should be at most seven, with five the recommendation.

5.1.2 Envisioning Information

Figure 5.2 depicts a visual representation of the major components of visual perception. As we have already emphasized, we don't actually see physical objects, we see light and this light enters our eyes through a small opening in the iris called the pupil. When focus is directly on physical objects, the light reflected shines on a small area on the retina at the back of the eye called the fovea. The retina consists of light receptors (millions of them) which are divided into two basic types, rods and cones. Rods detect dim light and record it in black and white. Cones detect bright light and record it in color.

Cones are further categorized into three, each of which senses a different range of the color spectrum: roughly blue, red and green. The fovea is just an area with a compact collection of cones, therefore light shining on the fovea can be seen in minute and fine detail. A human being is capable of negotiating up to 625 separate data points in a one-inch square area, such

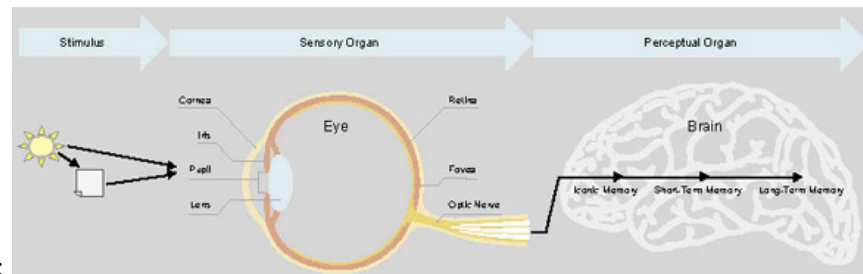


Figure 5.2:

as a condense collection of dots in a scatter plot. Visual perception stimulus detected by parts of the retina other than the fovea is not as detailed, but is capable of simultaneously processing vast amounts of information throughout one’s span of vision, ready to notice a point of interest that invites attention, which then leads to a quick shift in one’s gaze to that area of interest. Rods and cones translate what they detect into electrochemical signals and pass them on, through the optic nerve, to the brain where they can be processed. In essence, our eyes sense visual stimuli, then our brains take over perceiving the data and making sense of it.

As we begin to understand the advantages and disadvantages of visual perception, we will be in a better position to make use of human abilities to detect patterns in data when it’s visually displayed. One of the foremost exponents of visual perception, Edward R. Tufte, summarizes:

“We thrive in information-thick worlds because of our marvelous and everyday capacities to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flit through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsise, winnow the wheat from the chaff, and separate the sheep from the goats. Visual displays rich with data are not only an appropriate and proper complement to human capabilities, but also such designs are frequently optimal”

There is virtually unlimited freedom in how data is transformed. The difficult question is how best to represent it.

5.2 Information Visualization

Information visualization is very distinct. It is the communication of abstract data through the use of interactive visual interfaces. It produces interactive visual representations of abstract data to reinforce human cognition; thus enabling the viewer to gain knowledge about the internal structure of the data and causal relationships in it. As an image or an object is visualized, it is easier to comprehend since it becomes almost tangible in the mind. Visualization involves the enhancement of three-dimensional perception, color, and patterns through the application of computerized systems. The primary objective is to convert massive amounts of data into easily discernable shapes and colors in order to envision things that are too small, too big, too far away, or simply out of sight.

Security visualization is not just a method used in security analysis, it is a process of transforming information into a visual form enabling the viewer to observe, understand and make sense of the information in question. It employs computers to process and view the information using methods of interactive graphics, imaging, and visual design. It relies on the visual system to perceive and process the information.

5.2.1 Visualization Pipeline

The visualization pipeline describes the (step-wise) process of creating visual representations of data. It is made up of the following:

1. **Data Analysis:** data is prepared for visualization (e.g., by applying a smoothing filter, interpolating missing values, or correcting erroneous measurements) – usually computer-centered, little or no user interaction.
2. **Filtering:** selection of data portions to be visualized – usually user-centered.
3. **Mapping:** focus data is mapped to geometric primitives (e.g., points, lines) and their attributes (e.g., color, position, size); most critical step for achieving Expressiveness and Effectiveness.
4. **Rendering:** geometric data is transformed to image data.

Figure 5.3 is a diagrammatic flow of the visualization pipeline

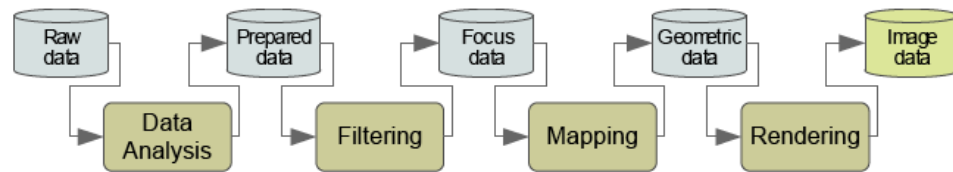


Figure 5.3:

5.2.2 Visualization Objectives

There are three major goals of Information Visualization which are to:

- enable the viewers develop a mental model upon viewing. The visual attributes have to match the properties of the data in a well defined manner
- enable the viewers reconstruct its context in the real world as well as assisting them in matching the recognized structures with the real world.
- aid in the analysis, the understanding and the communication of data models and concepts

The ultimate goal of information visualization is to bring to the fore and make visible to non-experts data relationships that may otherwise be hidden.

5.2.3 Visualization Benefits

Visualization enables users to make discoveries, decisions, or explanations about patterns (trend, cluster, gap, outlier...), groups of items, or individual items. Some of its other benefits include

Explorative Analysis - It takes only the data as basic interactive search for information and structures

Confirmative Analysis - In addition to data hypothesis, it also checks and verifies those hypothesis

Presentation and Communication - The analyst has to take the target group into account. It allows facts and statements be presented to viewers not familiar with results.

In discussing the benefits, we must also be aware that there are certain factors that influence the quality of the visuals. Some of the factors include the ratio of information recognized by the analyst compared to the information which was understood in that same timeframe as well as the defined and recognizable visual representation that is most suitable to display specific information. Other influencing factors include:

- Type and structure of the data
- Working goal of the visualization
- Domain knowledge of the user
- Visual capabilities or preferences of the analyzer
- Common metaphors and conventions in the domain
- Characteristics of the media

In essence information visualization must be expressive, effective and appropriate. Whilst expression and effectiveness are necessary conditions for the visualization which take the viewer into account, they do not measure the costs for generating the visualization. Being appropriate simply describes the amount of resources used to create the visualization.

5.2.4 Visualization Process

The visualization process is typically made up of three distinct but interrelated components. These components are: Data Tables - **Data Transformations**, Visual Structures - **Visual Mappings**, Views - **View Transformations** and is represented in Figure 5.4

5.2.4.1 Data Tables

Raw data format is transformed into structured data tables that have some meaning represented through metadata. It is quite trivial to apply modeling tools to data and come to a conclusion about the value of resulting models based on their predictive or descriptive value. This does not however diminish the role of careful attention to data preparation efforts. Data transformation process is loosely broken down into selection, cleaning, formation (of new composite data) and formatting.

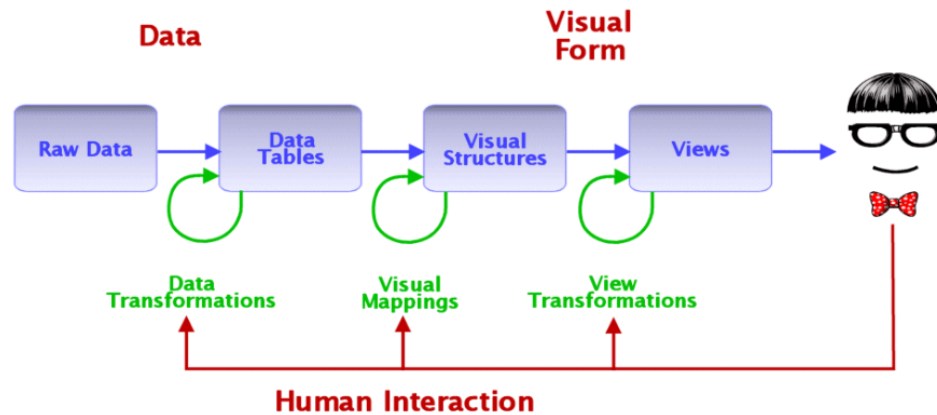


Figure 5.4:

Selection

A subset of data acquired in the previous stage (Raw Data) is selected based on certain criteria:

- Data quality properties: completeness and correctness
- Technical constraints such as limits on data volume or data type: this is basically related to data mining tools which are planned earlier to be used for modeling

Cleaning

This step immediately follows from data selection. It is also the most involving as a result of the different techniques that can be implemented so that data quality can be optimized for later stages. Possible techniques for data cleaning include:

- Data normalization - For example decimal scaling into an acceptable range or normalization of standard deviation
- Data smoothing - Discretization of numeric attributes is one example, this is helpful or even necessary for logic based methods.
- Missing values solution - There isn't a simple and straightforward remedy for the cases where some of the attributes have significant number of missing values. Generally, it is

good to experiment with and without these attributes in the data tables phase, in order to find out the importance of the missing values. Simple solutions are:

- a) replacing all missing values with a single global constant,
 - b) replacing a missing value with its feature mean,
 - c) replacing a missing value with its feature and class mean.
- The main disadvantage of simple solutions like these is that the values substituted is not correct. This means that the data will be a bit tilted towards a particular direction. However, if the missing values can be isolated to only a few features, then a solution can be advanced by deleting examples containing the missing values, or deleting attributes containing most of the missing values. Another more sophisticated solution is to try and predict missing values with a data mining tool in which case missing values predicted then becomes a problem of the special data mining tool.
- Data dimension reduction - There are at least two reasons that can be given for data reduction: data over bloat and length of time for problem resolution. The mechanisms for data reduction are usually effective but not perfect or optimal. The most usual step for data reduction is to examine the attributes and consider their predictive potential. Some of the attributes can usually be disregarded, either because they are poor predictors or are redundant relative to some other good attribute. Some of the methods for data reduction through attribute removal are:
- a) merging features using linear transform.
 - b) attribute selection from means and variances
 - c) using principal component analysis

Data Formation

This step represents constructive operations on selected data which includes:

- derivation of new attributes from two or more existing attributes
- generation of new records (samples)
- data transformation: data normalization (numerical attributes), data smoothing

- merging tables: joining together two or more tables having different attributes for same objects
- aggregations: operations in which new attributes are produced by summarizing information from multiple records and/or tables into new tables with "summary" attributes

Data Formatting

Final data transformation step which represents syntactic modifications to static data, but are required by the particular modeling tool include:

- reordering of the attributes or records: some modeling tools require reordering of the attributes (or records) in the dataset: putting target attribute at the beginning or at the end, randomizing order of records (required by neural networks for example)
- changes related to the constraints of modeling tools: removing commas or tabs, special characters, trimming strings to maximum allowed number of characters, replacing special characters with allowed set of special characters

5.2.4.2 Visual Structures

The prepared data is transformed into a geometric model by selecting geometric primitives such as points, lines and polygons and assigning the data attributes. For instance a 3-Dimensional data table can be transformed in a 3-Dimensional graphic using each one of the columns associated with a particular variable. The same table can be used to produce a 2-Dimensional representation with the third variable represented by the size or colour of the points placed in the chart according to the other two variables.

5.2.4.3 Views

Finally the representation can be viewed from different standpoints. Geometric data is further transformed into an image. This is done by viewing transformations that scale, translate, zoom and clip the graphic representation. Interaction allows the user to bring feedback to the system by changing the parameters that control the types of transformations already mentioned.

5.2.5 Feedback Loops

There are basically three feedback loops associated with the visualization process: *data gathering*, *data manipulation* and *data exploration*. We can obtain a series of data, inject it into the process and after a preprocessing and transformation stage obtain a graphical representation of it that eventually activates our visual and cognitive system. We can then manipulate the way in which the graphics engine shows the data once preprocessed and transformed.

Analysis of the result can also be explored by selecting between different pre-processes. For instance selecting different subsets of data or transforming the data to produce derived magnitudes like differences between data or statistical magnitudes computed on it and displaying it on a new graphical representation.

The physical and social environments also play a vital role in the information gathering process with the physical environment acting as the data source while the social environment determines what is collected and how it is interpreted.

5.3 Visualization vs Graphing

Information visualization differs from other forms of graphical communication in several distinct ways. Whilst most of the common visual tools are designed for presentation of known objects or quantities like chart of vulnerabilities by host, information visualization is more open-ended. It is a way of using graphical tools to see things previously invisible: structures, relationships or data obscured by other data.

When you set out to create a graph, you know the nature of the indices to be represented in advance; the graph is a recreation of those existing numbers, however, visualization is part of the process.

Therefore, in order to allow an analyst have a conversation with the data, information visualization tools perform actions that even the most sophisticated charting and graphing packages can't. First of all, the mechanism employed makes it easier to handle multivariate and dense dataset in a logical way, thus helping to make hidden correlations visible. Instead of the standard menu of bar chart, pie chart and line chart these tools offer presentation methods customized for particular domains and densities of information. The degree of customization of the tool also gives the user a greater degree of freedom to change the axes ordering the data, or to highlight or suppress certain kinds of data, in order to explore the underlying relationships.

To an extent, a number of different comparisons of data could be done in a spreadsheet like

Microsoft Excel, but combining these comparisons into one visual interface produces an output that no spreadsheet can even manage. In the same way that a sports team is more than its individual members because the whole is greater than the sum of its parts, the overall pattern of data often exhibits patterns that emerge from the individual pieces of data, patterns that can be impossible to spot by merely re-sorting the data in spreadsheet format.

Information visualization techniques offer a more varied palette of visual properties than the typical graphical interface: Characteristics such as color, brightness, contrast, surface area, line weight, and visual pattern, amongst others, can be used to represent differences between pieces of data. Indeed, one of the greatest challenges of information visualization is knowing when to say no to this embarrassment of riches – the hardest choices are often among the few visual aspects that will best represent the underlying data set in a way that’s easy to grasp.

Lastly, Information visualization has the potential to accomplish two things: to provide a new visual language for letting the average user get their work done, while at the same time increasing the amount of work they can do. In the past, the spread of information visualization has been hampered by the *GIGO* problem: Garbage In, Garbage Out. Tools that help users visualize the structure of security data are useless if the security data is unstructured. There’s also the less well-known *NINO* problem: Nothing In, Nothing Out. Organizations that are not aggressive about collecting, indexing, and storing data can’t visualize it.

5.4 Visual Data Representation

There are many different ways of representing security dataset. Some depend on the nature of the data or the nature of the analysis while others are largely equivalent. Visual representation can be illustrated in a holistic and understandable fashion by presenting them graphically using statistical and data presentation tools. However, when creating graphic displays, the following must be put into consideration:

- Type and status of audience
- Object or focal point of communication
- Hindrances to fully grasping and comprehending graphic
- Entire picture capture of display

Several types of visualization graphs and tools exist especially for multidimensional data visualization. These include

Frequency charts: Used for simple percentages or comparisons of magnitude

- Pie Charts
- Bar Charts
- Pareto Charts

Trends chart: Used to gauge trends over time

- Line Graph
- Run Charts
- Control Charts

Distributions chart: For finding out variation not related to time (distributions)

- Histograms
- Frequency Polygons

Associations chart: Used to look for a correlation between two things

- Scatter Plots
- Parallel Coordinates

Different types of data require different kinds of data representation tools. Data types are of two types and varieties. The first called *Attribute data* is data that is countable or data that can be categorized for instance the number of items, percentage of tall to short people etc. while the second called *Variable data* is based on measurement that is based on some continuous scale for instance duration, size, capacity etc. The following table lists the aforementioned display graphs, usage and the required data.

Visualization type	Usage	Required Data
Frequency Chart	Bar Chart Pie Chart Pareto Chart	Tallies by category (data can be attribute data or variable data divided into categories)
Trend Chart	Line Graph Run Chart Control chart	Measurements taken in chronological order (attribute or variable data can be used)
Distribution Chart	Histograms Frequency Polygons	Used to compare the distribution of distinct values for one or more discrete columns
Association Chart	Scatter Plots Parallel Coordinates	Used to investigate the relationship between two or more continuous columns

5.4.1 Frequency Charts

These are used for simple percentages or comparisons of magnitude.

5.4.1.1 Pie Charts

Pie charts are best suited to basic visualization needs such as comparing values of a dimension as parts or percentages of a whole. Example of a pie chart is given in Figure 5.5

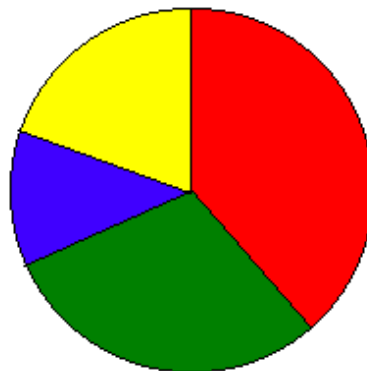


Figure 5.5:

Pie charts are employed in the definition and analysis of problems, verifying causes, or seeking solutions. They make it easier to understand data because they present the data as a graphical image by highlighting the output. This is particularly useful in presenting results to team members, managers, as well as non technical people. Pie charts present results that compare continuous data dimensions across discrete data dimensions in an x - and y -coordinate system. They can also be used with variable data that have been grouped. Pie charts are used to display relative proportions of various items making up the whole that is how the constituents of the pie.

5.4.1.2 Bar Chart

Bar chart is suited to a visual representation of the count of individual values of a data dimension. Bar charts are used to visualize comparisons between different categories. Example of a bar chart is given in Figure 5.6

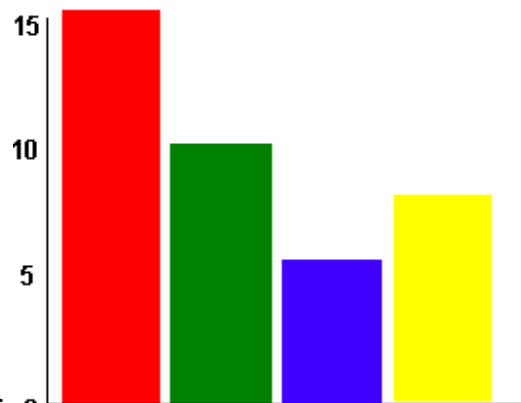


Figure 5.6: n

A bar graph is a graphical representation of frequency distributions of ungrouped data. It is a pictorial representation of the numerical data by a number of bars (rectangles) of uniform width erected vertically (or horizontally) with equal spacing between them. There are however, three types of bar charts to choose from, depending on the type data and stress points of the data. They are:

Simple bar charts: Used for sorting data into basic and simple categories.

Grouped bar charts: Used for breaking up data into different components within each sub category and showing comparisons between the individual components and between the sub categories. (It renders more usable information than a combined total of all the components.)

Stacked bar charts: These are like grouped bar charts and make use of grouped data within sub categories. (They make clear both the sum of the parts and each component's contribution to that total.)

Figure 5.7 are examples of the above named bar chart types.

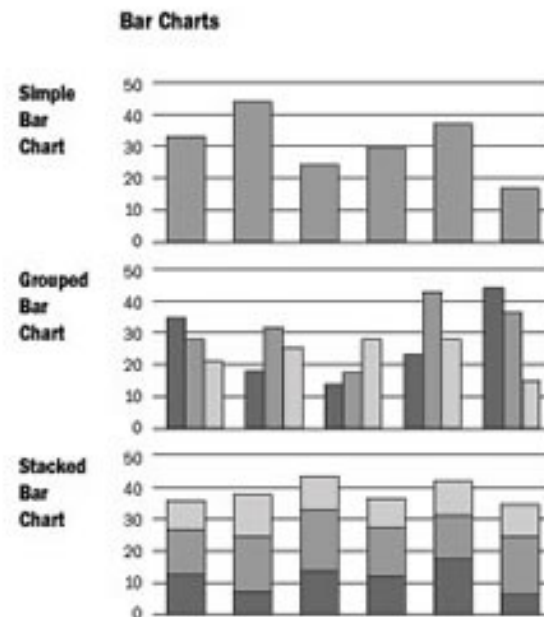


Figure 5.7:

Note Do not use too many notations on the charts. Keep them as simple and as direct as possible and include only the information necessary to interpret the chart. Scales must also be at regular intervals.

5.4.1.3 Pareto Chart

Pareto chart outputs facts needed for setting priorities. It organizes and displays information to show the relative importance of various problems or causes of problems. It is essentially a special form of a vertical bar chart that puts items in order (from the highest to the lowest) relative to some measurable effect of interest: frequency, cost, time. The chart is based on the Pareto principle, which states that when several factors affect a situation, a few factors will account for most of the impact. The Pareto principle describes a phenomenon in which 80 percent of variation observed in everyday processes can be explained by a mere 20 percent of the causes of that variation. An example is illustrated in Figure 5.8

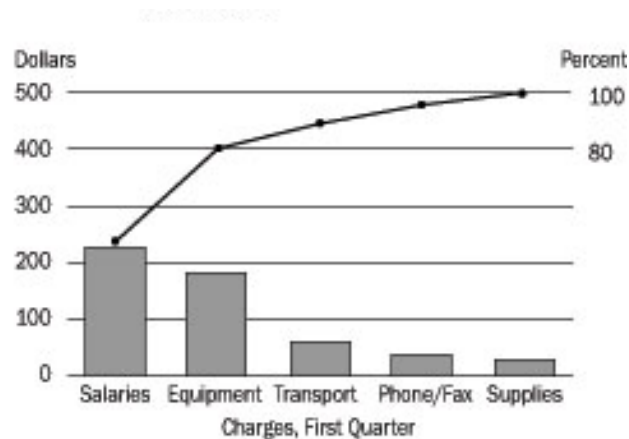


Figure 5.8:

Placing the items in descending order of frequency makes it easy to discern those problems that are of greatest importance or those causes that appear to account for most of the variation. Thus, a Pareto chart helps analysts focus their efforts where they can achieve the greatest potential impact. Comparing Pareto charts of a given situation over time can also determine if an implemented solution reduced the relative frequency or cost of that problem or cause.

5.4.2 Trend Charts

Trend charts are used to gauge trends over time.

5.4.2.1 Line Graphs

A line graph is a set of data points connected by line segments. In other words, it is simply a graph that uses points connected by lines to show variable changes over a period of time. Line graphs normally show how the values of one column (data dimension) compare to another column within an x - and y -coordinate system. Line and spline segments will connect adjacent points from the values of the data column. An illustration is given in Figure 5.9

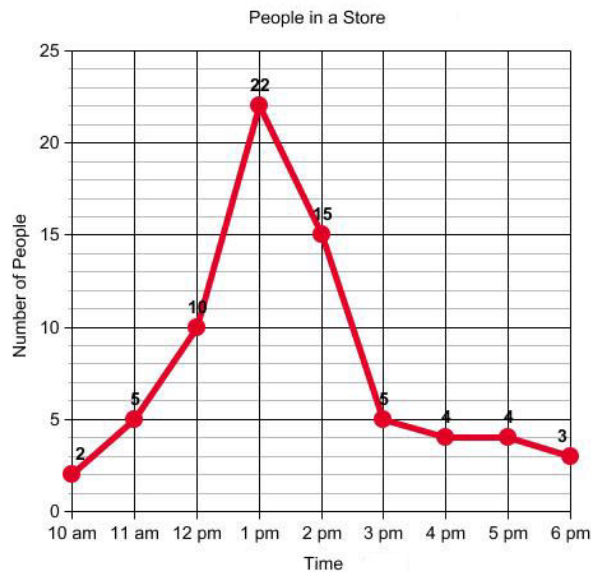


Figure 5.9:

5.4.2.2 Run Chart

Run charts also referred to as run-sequence plots give a graphical visualization of a variation in some process in a time sequence and help detect special (external) causes of that variation. They make trends or other non-random variation in the process easier to see and understand. With the understanding of patterns and trends of the past, groups can then use run charts to help predict future direction. An example is given in Figure 5.10

Run charts are an easy way to graphically summarize a univariate data set. Sometimes if analysis focuses on providing only the big picture (such as average, range, and variation),

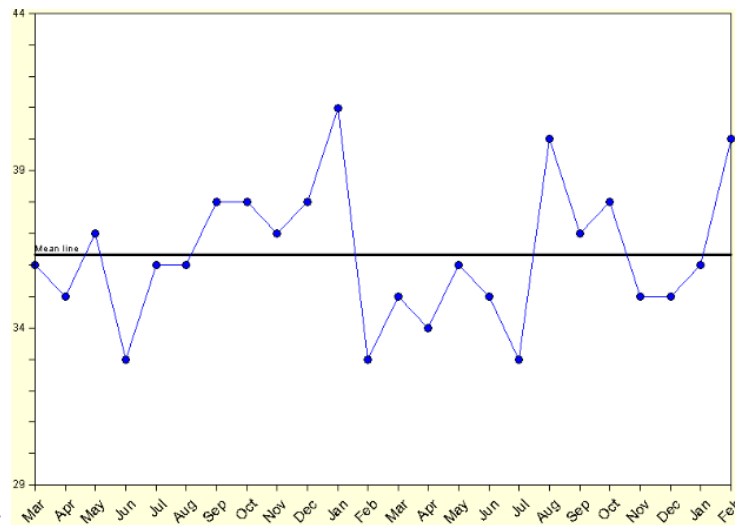


Figure 5.10:

trends over time can often be lost. Changes could be hidden from view and latent problems left unresolved. Run charts graphically display shifts, trends, cycles, or other non-random patterns over time. They can be used to identify problems by showing a trend away from the desired results and to monitor progress when solutions are implemented.

A run is the consecutive points running either above or below the center line (mean or median). The points in a run chart mark the single events (how much occurred at a certain point in time). A run is broken once it crosses the center line. Values on the center line are ignored: they do not break the run, nor are they counted as points on the run.

Note Be careful not to use too many notations on a run chart. Keep it as simple as possible and include only the information necessary to interpret the chart. Whenever possible, use a run chart to show the variation in the process. Do not assume that the variation is so clear and obvious that a run chart is unnecessary.

5.4.2.3 Control Charts

Control charts are used to analyze variation and process stability over time. If the run chart provides sufficient data, it is possible to calculate "control limits" for a process; the addition of these control limits creates a control chart. Control limits indicate the normal level of variation

that can be expected; this type of variation is referred to as common cause variation. Points falling outside the control limits, however, indicate unusual variation for the process; this type of variation is referred to as special cause variation. This analytical tool helps to distinguish between common and special causes of variation, allowing analysts focus on quality improvement efforts on eliminating special causes of variation (e.g., unplanned events). See Figure 5.11 for an illustration

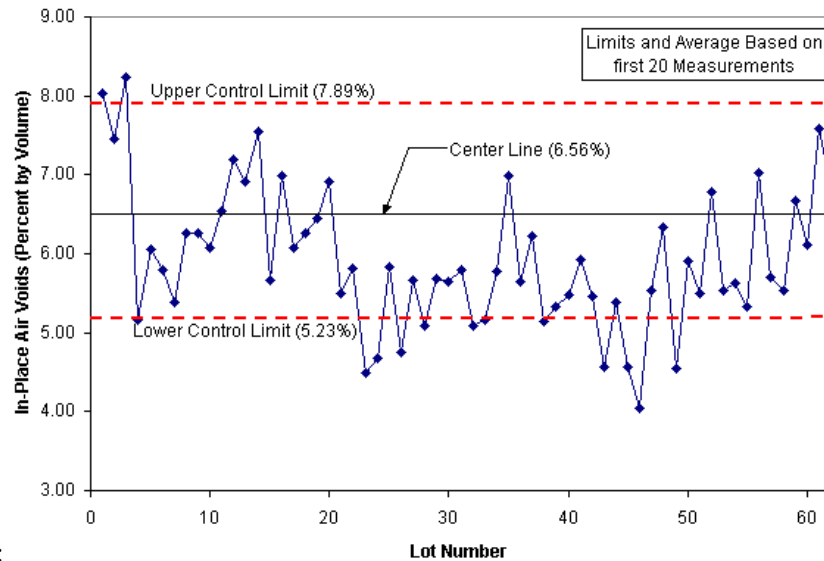


Figure 5.11:

Note Do not draw conclusions that are not justified by the data. Certain trends and interpretations may require more statistical testing to determine if they are significant.

5.4.3 Distribution Charts

They are used to compare the distribution of distinct values for one or more discrete columns.

5.4.3.1 Histograms

Histograms are similar to bar charts apart from the consideration of areas. In a bar chart, all of the bars are the same width and the only thing that matters is the height of the bar. In a

histogram, the area is the important thing. It is a graphical representation of a continuous frequency distribution i.e. grouped frequency distributions including vertical rectangles, with no spaces between the rectangles. The class-intervals are taken along the horizontal axis and the respective class frequencies on the vertical axis using suitable scales on each axis. For each class, a rectangle is drawn with base as width of the class and height as the class frequency. The area of the rectangles must be proportional to the frequencies of the respective classes. An example is shown in Figure 5.12

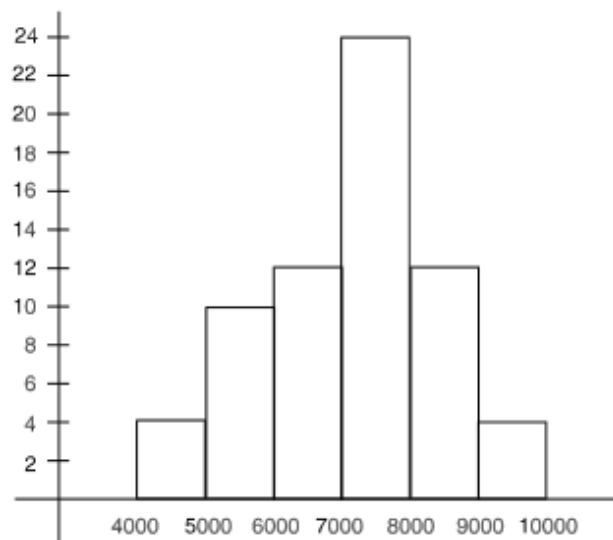


Figure 5.12:

5.4.3.2 Frequency Polygon

A frequency polygon is the join of the mid-points of the tops of the adjoining rectangles in a histogram. The mid-points of the first and the last classes are joined to the mid-points of the classes preceding and succeeding respectively at zero frequency to complete the polygon. A frequency polygon is shown in dotted lines in Figure 5.13

5.4.4 Association Charts

These are used to investigate the relationship between two or more continuous columns

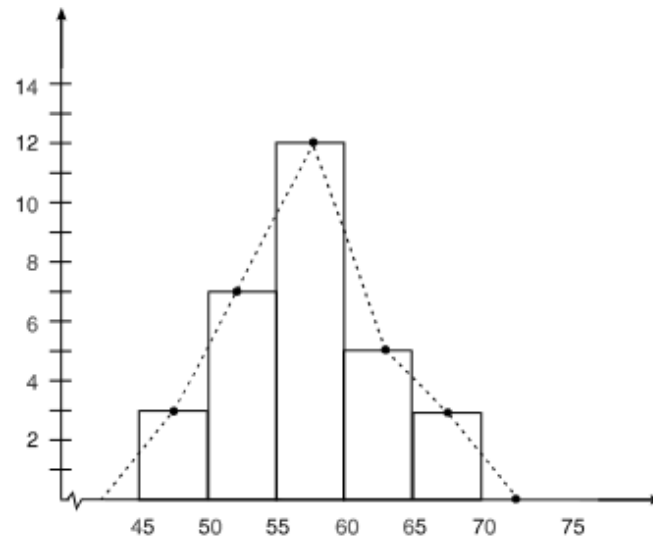


Figure 5.13:

5.4.4.1 Scatter Plots

The scatter plot is another method of visually displaying data. It shows the association between two variables acting continuously on the same item. The scatter plot illustrates the strength of the correlation between the variables through the slope of a line. This correlation can point to, but does not prove a relationship. Therefore, it is important not to immediately conclude the relationship between variables as there may be another variable that modifies the relationship. The scatter plot is easy to use, but should be interpreted with caution as the scale may be too small to see the relationship between variables since other unknown factors may be involved. An example is shown in Figure 5.14

Typical Uses

- Scatter plots make the relationship between two continuous variables stand out visually in a way that raw data cannot
- They can be used in examining a cause-and-effect relationship between variable data that is continuous data measurement.

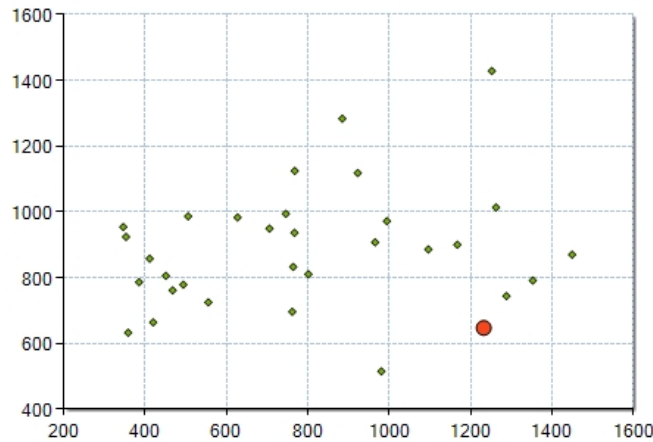


Figure 5.14:

- They show relationships between multiple effects to see if these effects stem from a common cause.
- They are also used to examine the relationship between two causes.

5.4.4.2 Parallel Coordinates

Statement of Problem

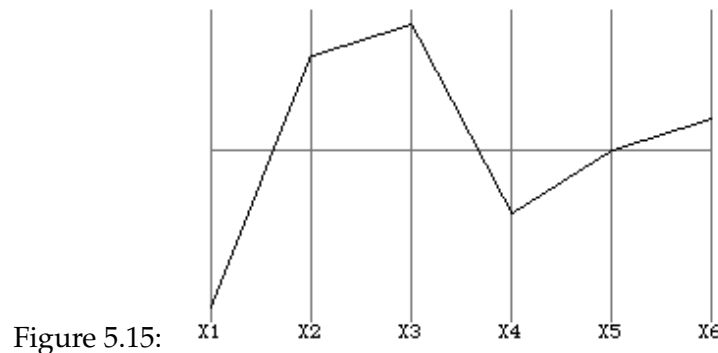
One of the challenges with visualization of multidimensional dataset is that human perception operates in a three-dimensional space with orthogonal coordinates - any data set of higher dimension has to be mapped or projected into this three dimensional space to be explored by humans. This projection implies a loss of information and/or clarity, because there are only three space dimensions available, and the exceeding information has to be either omitted or mapped to "intrinsic" dimensions or properties of the projected data point (like color or size, for example). For some problems, this might be a feasible approach or even the most intuitive way of visualization. For other problems, the asymmetry between extrinsic and intrinsic coordinates is not at all intuitive and can be misleading.

In the mean time, available output devices for computer generated visualizations only worsen the problem: practically all available output technology is purely two-dimensional (screens, printers, plotters), and so the three-dimensional mapping of the data set has to be projected

again onto a 2d projection surface, again causing data loss and possible confusion. Emerging 3d output technology might solve this problem in the future, but these systems are nowhere near the production quality and availability of current 2d-output media.

Parallel coordinates Algorithm

With computers, trying to map the n-dimensional problem into 2- or 3-dimensional orthogonal space shouldn't even be attempted. Instead, all the coordinate axes $x_n Z_n$ of the problem are arranged parallel along the x-axis of (regular) 2d-space. A single point in Cartesian space is then represented by a profile of lines connecting all the coordinates of the point in the computer space (Figure 5.15)



A parallel coordinate plot is a graphical data analysis technique for examining the relative behavior of all variables in a multivariate data set. It consists of a sequence of spaced vertical spikes with each spike representing a different variable in the multivariate data set. A single parallel coordinate plot examines the behavior of all such variables but only for a specified subset of the data. The total length of a given spike is uniformly set to unity for sake of reference. The "data length" of a given spike is proportional to the magnitude of the variable for the subset relative to the maximum magnitude of the variable across all subsets. (Thus we are looking at the ratio of the "local" value of the variable to the "global" maximum of the variable.) An interconnecting line cutting across each spike at the "data length" gives the parallel coordinate plot its unique appearance and name.

The real strength of parallel coordinates isn't in the ability to communicate some truth in the data to others, but rather in the ability to bring meaningful multivariate patterns and

comparisons to the front burner when used interactively for analysis. So in essence it is a graphical analysis technique for plotting multivariate data. An example is given in Figure 5.16

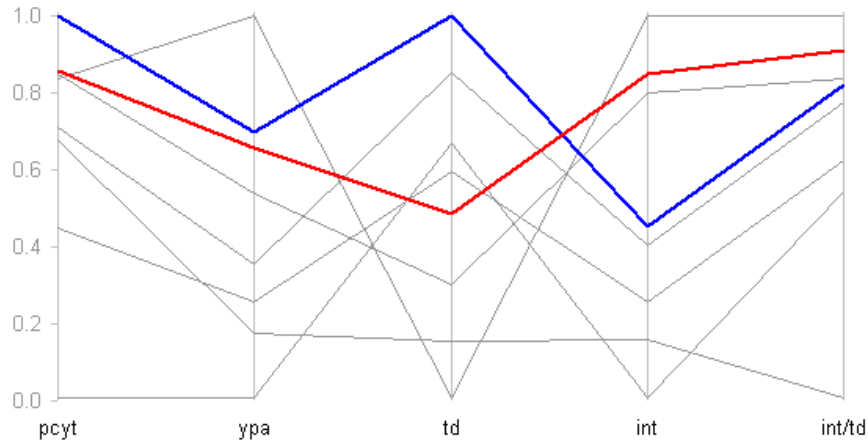


Figure 5.16:

The main advantages of computer visualization is that the number of dimensions is only limited by horizontal screen space and that correlations of variables can be easily spotted. The display of complex datasets in the computer space gives an intuitive overview of properties of the dataset that might not be obvious or not visible at all in other visualizations. Complex queries can be formulated graphically by selecting regions or locations on the coordinate axes, possibly combining single operations to complex queries.

To recognize the worth of a parallel coordinates display, it should not be thought of in the same light as a normal line graph. Lines are predominantly used to represent time-series data. The up and down slopes of the lines indicates a change through time from one value to the next. The lines in parallel coordinate displays, however, don't indicate change. Instead, a single line in a parallel coordinates graph connects a series of values - each associated with a different variable - that measures multiple aspects of a particular object. Parallel coordinates can reveal correlations between multiple variables. This is particularly useful when you want to identify which conditions correlate highly to a particular outcome. Multivariate analysis requires specialized visualizations and methods of interaction with data.

Hierarchical parallel coordinates

One problem with computer displays is that they can get extremely complex and cluttered when the number of data points increases. Filtering and brushing are two methods used to reduce the complexity or increase the clarity of a computer display, but both of these focus on subsets of the data and not on overview, which is one of the strengths of PC visualization. [Fua et al. 1999] proposed a clustering approach to reduce the complexity of the display and still be able to overview the whole data set. A cluster of lines is represented by a line denoting the average value of the cluster, and by regions of decreasing opacity above and below this line representing the other values in the cluster. See Figure 5.17

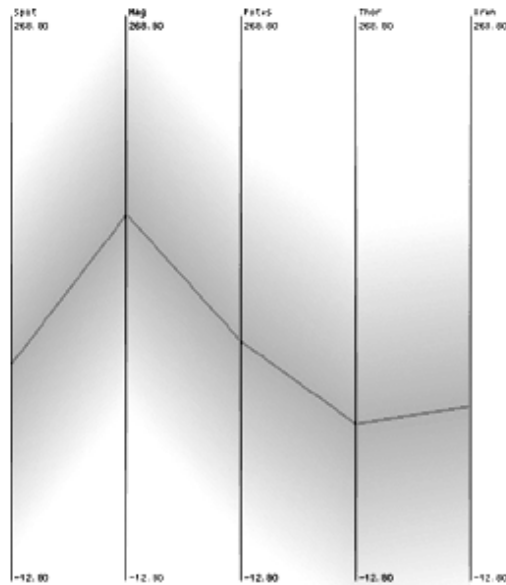


Figure 5.17:

By selecting individual clusters or intervals the user can then drill down from an overview perspective to the detailed view of the individual data points.

5.4.5 Tree Graphs

Treemap is a space-constrained visualization of hierarchical structures. It is very effective in showing attributes of leaf nodes using size and color coding. Treemap enables users to

compare nodes and sub-trees even at varying depth in the tree, and help them spot patterns and exceptions. It is a method for displaying tree-structured data by using nested rectangles. An example of a treemap is shown in Figure 5.18

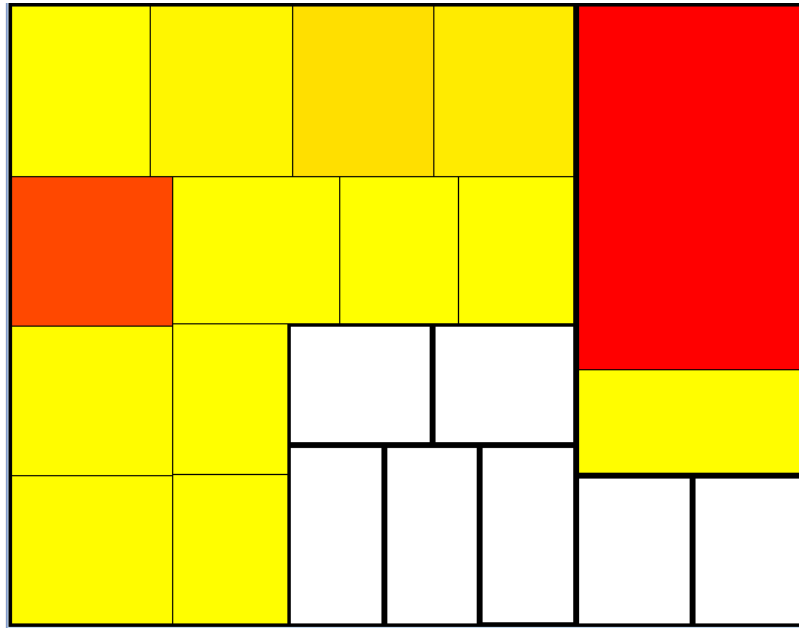


Figure 5.18:

A treemap is created by defining a tiling algorithm which is a way to break up a rectangle into multiple sub-rectangles of specified areas. In essence, they use 2D visualization of trees where the tree nodes are encapsulated into the area of their parent node. The size of the single node is determined proportionally in relation to all other nodes of the hierarchy by an attribute of the node. Unfortunately, these properties have an inverse relationship. As the aspect ratio is optimized, the order of placement becomes less predictable. As the order becomes more stable, the aspect ratio is degraded.

5.4.6 Link Graphs

A link graph is best-suited for visualizing relationships in a dataset. An example may be to display an interactive visualization of the connections between hosts on a network. For instance, in a packet capture file, our analysis may include the relationship between a source

address and a destination address. Link graphs are sometimes referred to as network maps or relationship graphs.

5.5 Limitations of Visualization

After all said and done, visualization is not a silver bullet. Information visualization tools have also encountered problems that have kept them from mainstream use. The most obvious one is the display. The surface area of the largest commodity PC screen is only a fraction of the size and visual density of a plain paper, and unlike the microprocessor and storage, screen size and resolution are not improving very quickly. Some industries, such as finance and oil and gas get around this problem by equipping their personnel with two or more large LCD screens. Any company looking to offer information visualization products or services to a large audience has to target display technology that is going to see only incremental short-term improvement.

Another problem is that of skill: Computers make it easier to make bad visual presentations than to make good ones: After the arrival of desktop publishing, the simplest birthday party invitation usually arrived in five different fonts. Graphics tools like Photoshop and presentation tools like Powerpoint have had a similar effect, vastly increasing the number of people capable of creating graphics without doing anything to increase the number of people capable of creating useful graphics. Given these restrictions, information visualization is usually sold as both a product and a service, with per-client customization and some degree of user training as a core part of any contract.

Lastly there is the problem of human work patterns. As mentioned earlier, information visualization tools can handle multivariate datasets by mapping different axes of data to items such as line weight, surface pattern, and motion. However, the resulting interfaces can often leave the user spending more time trying to understand the interface than the data itself. Most times when we refer to 'intuition' we often mean 'familiar'. Information visualization tools create symbolic overhead: a user has to learn new interfaces, software behaviours or visual symbols, and to the degree that these interfaces are unfamiliar, they feel unintuitive even though companies marketing information visualization tools typically work to convert the rather obscure interfaces developed in technical environments to something the end user can easily comprehend.

5.6 Summary

This chapter examined data mapping and information visualization tools and techniques. We discussed visualization benefits and processes and at the same time enumerated standard data representation methods that are employed in discovering trends, distribution and anomalies in datasets. We looked at different categories of tools and how to choose the right graph for data representation. Furthermore, we looked at how graphical visualization is used to explore security dataset for decision support. We must however note that more often than not, the right graph isn't necessarily going to magically solve all the challenges associated with dataset representation.

Chapter 6

Security Visualization Techniques

Information visualization has existed as a separate area of inquiry since the 1987 National Science Foundation report, “Visualization in Scientific Computing”. In the last 10 years, a variety of new tools have been developed by different people and companies. Though the approaches to information visualization differ considerably, the common theme is for the tools to be simple enough to use without significant retraining, and be tolerant of partial or incomplete datasets if information visualization is to become mainstream.

A number of open source security visualization applications have evolved over the last couple of years. Most are turn key applications that are focused on particular visualization type such as time based network visualizers, high dimensional data visualization, 3D scatter plot visualization and so forth. These are applications targeted more at the analyst than at graphics programmers. These tools are sometimes referred to as Modular Visualization Environments (MVE).

With MVEs, software components are connected to create a visualization framework. These components are referred to as modules. MVEs allow the modules for data tables, visual structures and views be combined into executable flow networks. Once the structure of the visualization application has been established, the MVEs execute the network and display the computer generated image. Each image that is produced is a visual representation of the scene defined by the modules. The analyst can then interact with the image by moving or changing direction or by modifying surfaces, rotating, shifting or resizing objects, or by changing the point and angle of view.

6.1 Evolution

Visualization is an ingredient technology, not an application in itself. Majority of visualization tools are made up of 2-D visualization such as MapQuest - a GIS program, 3-D visualization, multidimensional data, hierarchical data, and temporal data. Two previous evolutions have relied on visual presentation – the spread of PCs, which began reaching a much wider audience with the advent of the GUI, and the spread of the Web, with Mosaic as a browser. The need is enormous as well. When we move from our desktop to the Web, we go from a compact and highly controlled environment to a vast, interlinked one, and yet the visual tools for handling the complexities of the Web are much less well-developed than the desktop. Even the desktop is an advanced filing system, all but useless for extracting patterns from large amounts of data. In addition, our ability to handle large and growing datstores is limited, as most of the tools used in the searching and extracting process are visually light and thin.

It has been said that information visualization branched off from scientific visualization only a decade ago. Information visualization may depict either physical or abstract data, while scientific visualization mostly depicts physical data. We can maintain, however, that information visualization's present generation of interfaces derive from spreadsheet applications. Visible Calculator — VisiCalc — emerged in the late 1970s with data organized in rows and columns. The value of spreadsheets was to simply offer data overviews but was so successful in the 1970s that it became the incentive for a lot of people to even acquire a PC. In the late 1980s, Excel was one of the first programs released for the Windows operating system.

Since then, much of the work in information visualization has gone on in academic and commercial fields, with only a small percentage of the research crossing over into commercial applications. Two of these labs stand out. The first is Ben Shneiderman's Human-Computer Interaction Laboratory at the University of Maryland. Shneiderman's work on tree maps later became heat maps, the concept behind a certain Smart Money's Market Map.

Quite a lot of the information visualization techniques used by companies have evolved from research and development that treat information visualization not as an end in itself but a means to an end. Lumeta, which was spun off Lucent in 2000, uses these techniques as a way of representing existing network data. Likewise, the enormous growth in the number of social network analysis visualization techniques from labs such as Microsoft's Social Computing Group, IBM's Social Computing Group, and MIT's Social Media Group are optimizing information visualization to come up with solutions to peculiar challenges in comprehending social organization. Another traditional area of strength of research is in data interpretation of the Internet. The Center for Advanced Spatial Analysis (CASA) at University College of Lon-

don, maintains *cybergeography.org*, a site dedicated to listing and displaying the wide variety of representations of not just the technological but the social view of the Internet much like the site managed by the Cooperative Association for Internet Data Analysis (CAIDA) having a number of visualization tools dedicated to the “engineering and maintenance of a robust, scalable global Internet infrastructure.”

Despite the variety of techniques resulting from all this research, the fact that most of the information visualization in the commercial world falls into a few categories suggests that there is no clear or obvious path from the lab to the field. HCIL has had a bit of success as a commercial research group in getting its tools and techniques adopted commercially. The relatively short history of the field suggests that despite the R&D title, lab work in information visualization is almost pure “Research” even in commercial labs, and that the “Development” has been outsourced to firms determined to make the tools user-friendly.

6.2 Tools

Information visualization tools are visual by definition and interactive in implementation; they work more like the whiteboard than the bar chart. A user of a visualization tool is not simply presented with static data; instead, the user can explore a set of data interactively by changing the data modeling or the hypotheses being tested and seeing the resulting visual representation. This feedback loop is part of the exploration; the tools support the process rather than simply expressing the results.

Information visualization tools and applications can display results quickly when words match words. That is, a text query might match with an object metadata rather than the query by going through the full text of a document or examining the object itself. While speeding the search, this places a burden on the accuracy of the metadata. Images generally include metadata in text form to speed up the matching. Graphical illustrations however differ with different information visualization tools as will be shown in the following sections.

6.3 AfterGlow

AfterGlow¹ is really just a collection of scripts which facilitate the process of generating link graphs. The tool is written in Perl and needs to be invoked via the Unix command line as

¹<http://afterglow.sourceforge.net/>

there is no GUI. Nevertheless the the tool is quite simple to use. AfterGlow accepts a CSV file as input file. The file can contain two or three columns of data. A common way of generating the CSV files are parsers which take a raw input file, analyze it and output a comma separated list of records based on the data they found. The output of AfterGlow is one of two formats. Either it generates a dot attributed graph language file - the input required by the graphviz library - or it can generate input for the large graphing library (LGL).

6.3.1 Parsers in Afterglow

AfterGlow provides a couple of example parsers to generate CSV input files. The first one to parse Tcpcdump or *pcap* output and the second one to parse sendmail log files. Recommended use of AfterGlow usually includes piping all steps into a single command, but the Tcpcdump-to-CSV conversion is more useful for illustration purposes.

6.3.2 Graph Representation

Depending on the number of columns fed into AfterGlow, it can be run in two modes - two- or three-column mode. In the former mode, you define nodes that are being connected with each other and in the latter, the graph output is represented with three nodes, where the first column represents the source node, the second one the event node and the third one the target node. A sample output for a three node configuration is shown in Figure 6.1 Make sure that when feeding AfterGlow only two columns, you use the *-t* parameter!

6.3.3 Configuration File

Most of the capabilities in AfterGlow is driven by the *color.properties* file used to configure the color output. The following is a sample of a configuration file:

```
# AfterGlow Color Property File
#
# @fields is the array containing the parsed values
# color.source is the color for source nodes
# color.event is the color for event nodes
# color.target is the color for target nodes
#
```

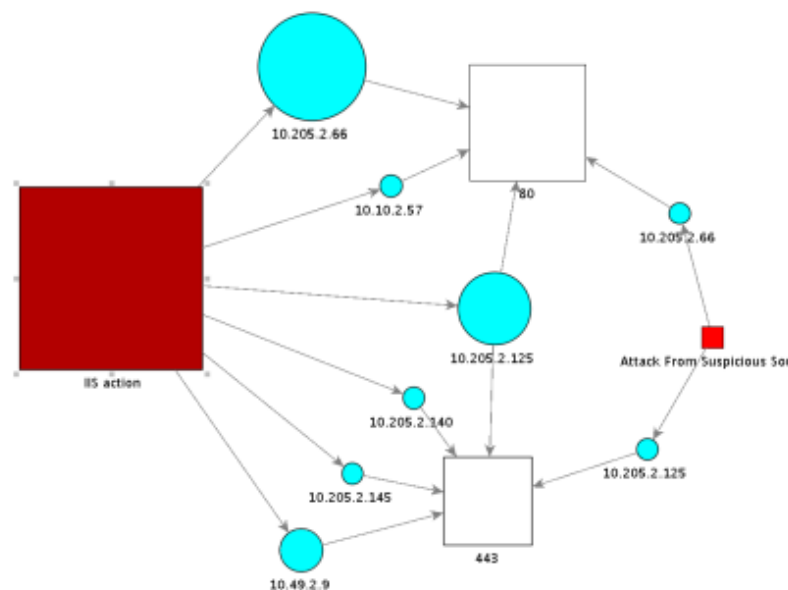


Figure 6.1:

```
# The first match wins
#
color.source="yellow" if ($fields[0]=~/^192\.168\..*/);
color.source="greenyellow" if ($fields[0]=~/^10\..*/);
color.source="lightyellow4" if ($fields[0]=~/^172\.16\..*/);
color.source="red"
color.event="yellow" if ($fields[1]=~/^192\.168\..*/);
color.event="greenyellow" if ($fields[1]=~/^10\..*/);
color.event="lightyellow4" if ($fields[1]=~/^172\.16\..*/);
color.event="red"
color.target="blue" if ($fields[2]<1024)
color.target="lightblue"
```

This might look a bit intimidating at first. Before we look at the individual entries, we need to know what input corresponds to this configuration. We are processing data which consists of three columns. The first two containing IP addresses and the third column representing

ports. This means our input shows the source address for an event in the first column, the destination address in the second and the destination port in the third column. Now back to the configuration file, there are basically three assignments: *color.source*, *color.event*, and *color.target*. Well, these values correspond to the three nodes in Figure 6.2



Figure 6.2:

An assignment is a perl expression which has to return a color name. The expressions are evaluated top to bottom. As soon as an expression matches, the color for this node is assigned. Another important fact is that color configurations can reference the values of the current record, which are made available in the `@fields` array. The first column of the data is therefore accessible with `$fields[0]`.

Getting back to our example, you should now understand what the first three lines are doing. Whenever the first column of the data (`$fields[0]`) starts with 192.168., the node is colored yellow. If it starts with 10., the node is greenyellow and if it starts with 172.16, it is colored in lightyellow4. If none of these conditions are true, red is the default color that will be used. The same logic applies to the event nodes. This time referencing the second column (`$fields[1]`). For the target nodes, we want to color them blue if the target port is below 1024 and lightblue if it is equal or bigger than 1024.

6.3.4 Case Study 28: Visualization with Afterglow and Graphviz

Graphviz is a rich set of graph drawing tools and it comprises the following graph layout programs - *dot*, *naoto*, *twopi*, *circo* and *fdp*. This will assist us in visualizing our packet capture file. This case study makes use of the *capture5.pcap* which is a simple Web download thus:

Graphviz Installation

```
# yum -y install graphviz
```

Then proceed with Afterglow thus to generate the equivalent CSV file

```
# tcpdump -vtttttnnelr capture5.pcap | afterglow/src/perl/parsers/tcpdump2csv.pl \  
"sip dip dport" > capture5.csv
```

where *sip* is source IP, *dip* is destination IP and *dport* is destination port

Note The parser remembers the source of a transmission and automatically inverts the responses to reflect that behavior. It outputs the direction of the communication (client to server) and not the direction of the packets. This is very useful when visualizing network traffic. Think about it!

tcpdump2csv.pl allows you to select a number of possible fields to be written to the CSV output, including timestamp, destination and source IP, MAC, and port, as well as time-to-live and other parameters. you can look at the script itself for more options. This should generate the *capture5.csv* file in your current working directory which can then be piped into afterglow thus:

```
# cat capture5.csv | ./afterglow/src/perl/graph/afterglow.pl \  
-c ./afterglow/src/perl/parsers/color.properties \  
| dot -Tjpg -o capture5.jpg
```

Dot

Dot is a Unix filter for drawing directed graphs. It works well on graphs that can be drawn as hierarchies. It reads attributed graph files and writes drawings. By default, the output format dot is the input file with layout coordinates appended. Figure 6.3 shows the image obtained

This can also be done.

```
# cat capture5.csv | ./afterglow/src/perl/graph/afterglow.pl \  
-c ./afterglow/src/perl/parsers/color.properties \  
| neato -Tjpg -o capture5.jpg
```

Naeto

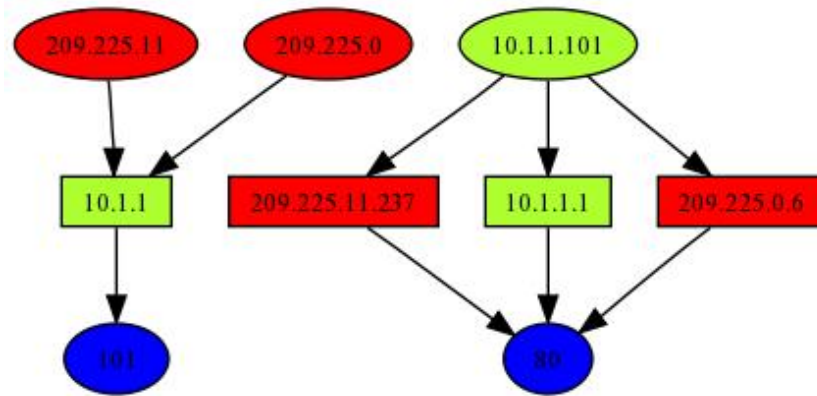


Figure 6.3:

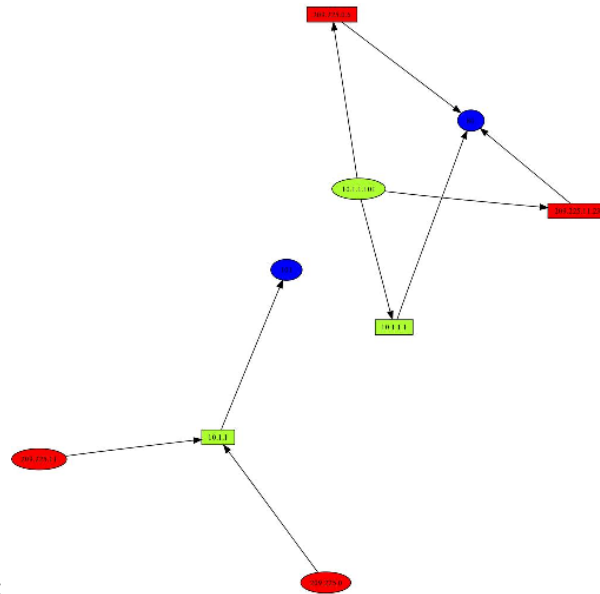


Figure 6.4:

Neato is a Unix filter for drawing undirected graphs. It creates “spring models”. Input files must be formatted in the dot attributed graph language. By default, the output of *neato* is the input graph with layout coord nates appended. Figure 6.4 shows the resulting image

```
# cat capture5.csv | ./afterglow/src/perl/graph/afterglow.pl \
-c ./afterglow/src/perl/parsers/color.properties \
| twopi -Tjpg -o capture5.jpg
```

Twopi

Twopi is a filter for radial layouts of graphs. Basically, one node is chosen as the center and put at the origin. The remaining nodes are placed on a sequence of concentric circles centered about the origin, each a fixed radial distance from the previous circle. All nodes distance 1 from the center are placed on the first circle; all nodes distance 1 from a node on the first circle are placed on the second circle; and so forth. Figure 6.5 shows the given output

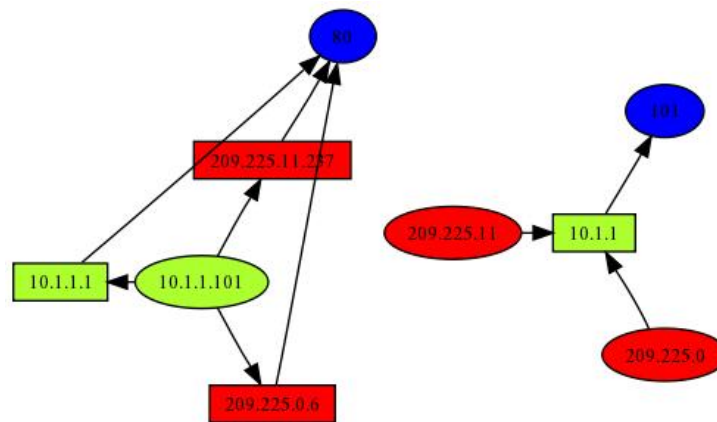


Figure 6.5:

```
# cat capture5.csv | ./afterglow/src/perl/graph/afterglow.pl \
-c ./afterglow/src/perl/parsers/color.properties \
| circo -Tjpg -o capture5.jpg
```

Circo

Is a filter for circular layout of graphs. The tool identifies biconnected components and draws the nodes of the component on a circle. The block-cutpoint tree is then laid out using a recursive radial algorithm. Edge crossings within a circle are minimized by placing as many

edges on the circle's perimeter as possible. In particular, if the component is outerplanar, the component will have a planar layout. Figure 6.6 shows the image of the *pcap* file

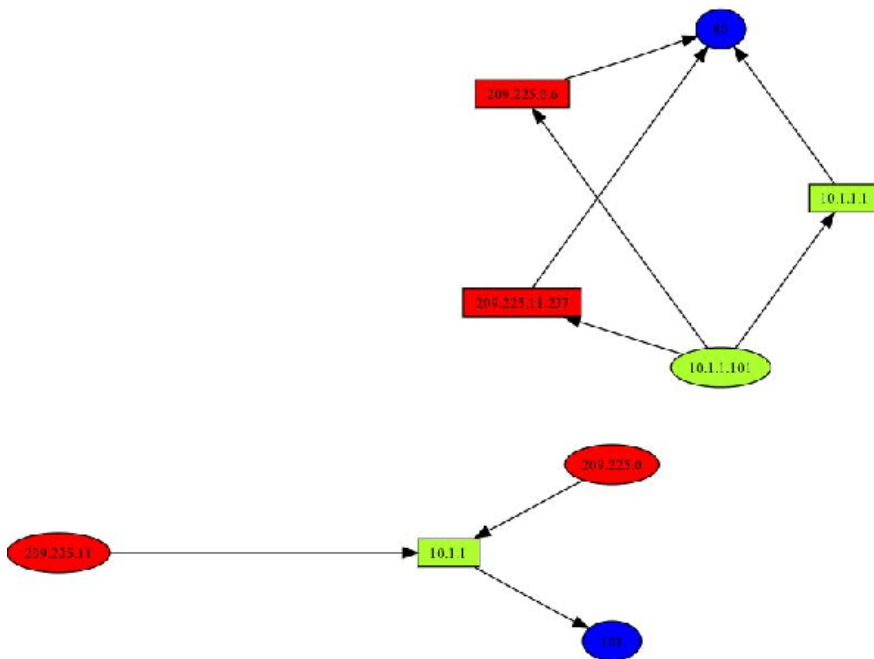


Figure 6.6:

```
# cat capture5.csv | ./afterglow/src/perl/graph/afterglow.pl \
-c ./afterglow/src/perl/parsers/color.properties \
| circo -Tjpg -o capture5.jpg
```

Fdp

Is also a filter for undirected graphs using a “spring” model. It relies on a force-directed approach. Figure 6.7 depicts the *pcap* file image

6.3.5 Functions in Afterglow

In later versions of AfterGlow, functions have been introduced. These functions can be used in the property file to accomplish clustering, filtering and coloring. The functions are listed in

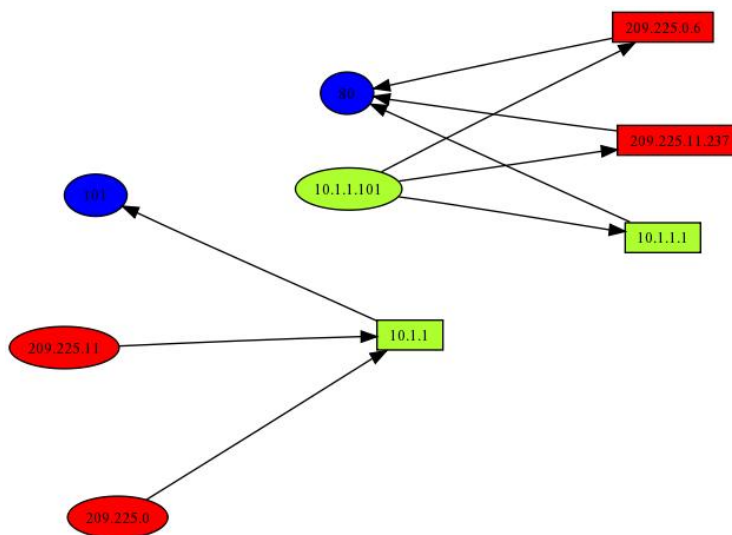


Figure 6.7:

the following:

field()

```
if ($type eq "source") { return $fields[0];}
if (($type eq "event" || ($twonodes)) { return $fields[1];}
if (($type eq "target" && (!$twonodes)) { return $fields[2];}
```

match()

```
($regex) = @_;
return $globalField =~ /$regex/;
```

regex_replace()

```
($regex) = @_;
return ($globalField =~ /$regex/)[0];
```

subnet()


```

my ($value,$value2) = @_;
my @temp = split(/\./,$value);
# return if not an IP address
return(0) if (scalar(@temp) != 4); # very simplistic test!
my $ip=unpack("N",pack("C4",@temp));
my ($network,$mask) = $value2 =~ /([\^\/]+)\/(.*)/;
$network=unpack("N",pack("C4",split(/\./,$network)));
$mask = (((1 << $mask) - 1) << (32 - $mask));
$newNet = join(".",unpack("C4",pack("N",$ip & $mask)));
$newNetwork = join(".",unpack("C4",pack("N",$network & $mask)));
if ($newNetwork eq $newNet) {
return 1;
} else {
return 0;
}

```

6.4 TNV

TNV² referred to as The Network Visualizer or Time-based Network Visualizer depicts network traffic by visualizing packets and links between local and remote hosts. It displays network traffic as a matrix with the packet capture timeline on the *x-axis* and all of the host IP addresses in the dataset on the *y-axis*. TNV is used as a tool to learn the the constituents of 'normal' activity by investigating packet details security events. It can also be use in troubleshooting network performance.

The main visualization shows remote hosts along the left side and a reorderable matrix of local hosts on the right, with links drawn between them. The local host matrix shows aggregated packet activity as background color, and shows network packets as triangles, with the point representing the packet direction. Packets and links are color coded and matched with protocol, and the analyst can optionally show the flags for TCP packets.

By selecting a cell within the matrix, representing a local host for a certain time period, the user can show either the packet details or the port activity related to that host. The main interaction mechanism for moving through the data is a scroll bar that sets the viewable selection, while at the same time showing areas of relative network activity in a bar graph - providing an

²<http://tnv.sourceforge.net/>

overview of the entire dataset with a more detailed display in the main visualization. TNV interface is shown in Figure 6.8

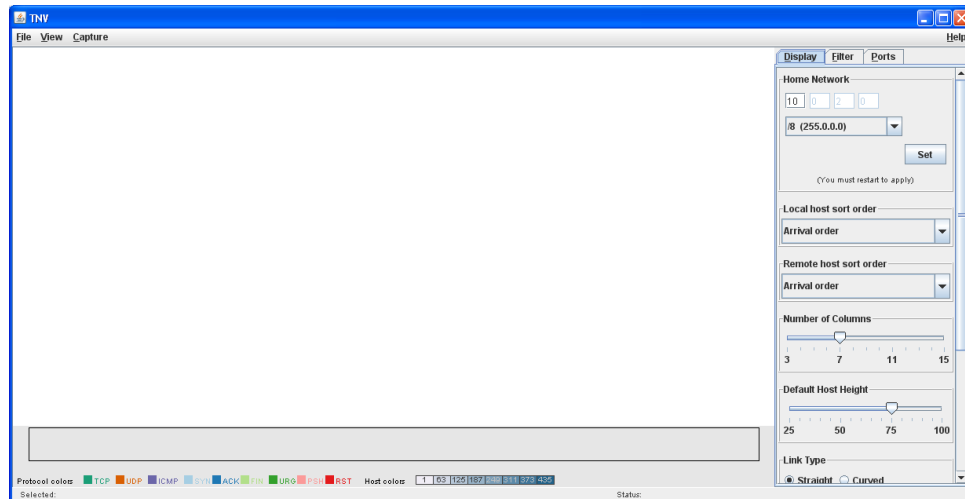


Figure 6.8:

6.4.1 Case Study 29: Visualization with TNV

The tool is pretty simple. Execute `tnv_startup.sh` (Linux) or `tnv_startup.bat` (Windows). Import the `capture5.pcap` file by choosing the menu item **File -> Import pcap file...** It can be used to capture packets off the wire by selecting **Capture -> Capture Packets...** Figure 6.9 shows the output

The memory size available to TNV will determine the number of packets that can be visualized at once, but generally anything less than 25,000 will be fine. Up to 60,000 will work, but the user interaction will be a bit sluggish.

TNV shows remote hosts at the left of the screen and local hosts in a matrix of host rows and time columns at the right with links drawn between them. The matrix cells' colors represent the number of packets for that time interval. Packets can optionally be superimposed on the aggregated local host cells. Figure 6.10 shows the different aspects of the TNV interface

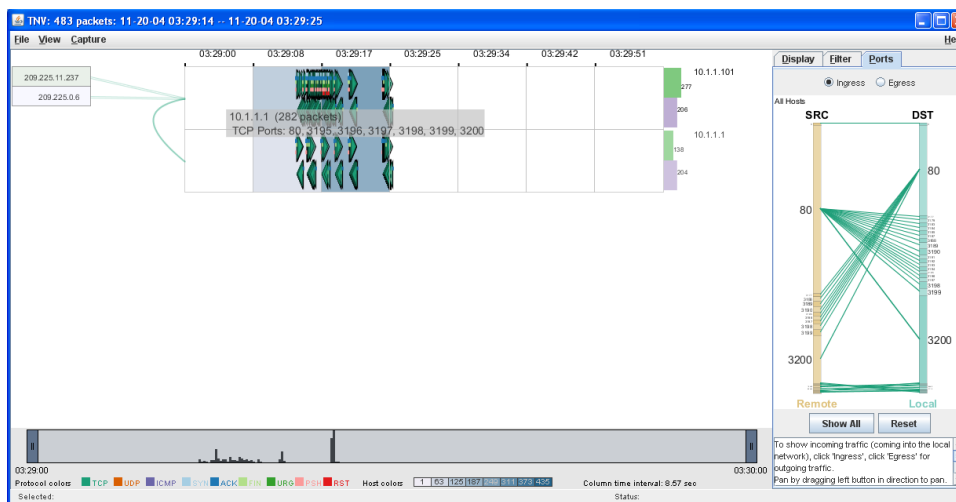


Figure 6.9:

6.5 Rumint

RUMINT is an open source network and security visualization tool coined from rumour intelligence developed by Dr. Greg Conti. This section will go through the primary techniques found in RUMINT and how to interpret them. It also accepts *pcap* files as input without conversion.

Rumint gives multiple views of a data packet capture.

ASCII and hexadecimal provides this functionality. While the canonical hexadecimal and ASCII view is useful for very detailed analysis, it lacks the ability to compare larger numbers of packets.

Text Rainfall allows 24 to 40 packets to be displayed at the same time. From the output it becomes obvious that many packets are behaving in the same manner. While this is an improvement, the text rainfall view is still limited to approximately 40 packets at one time.

Byte Frequency view compresses this information by plotting one packet per horizontal line. Each horizontal line is 256 pixels long and pixels along this line are illuminated based on the presence of bytes within the packet. Vertical lines show constant values between packets, in many instances these are header fields such as the IP Version (typically 4 for

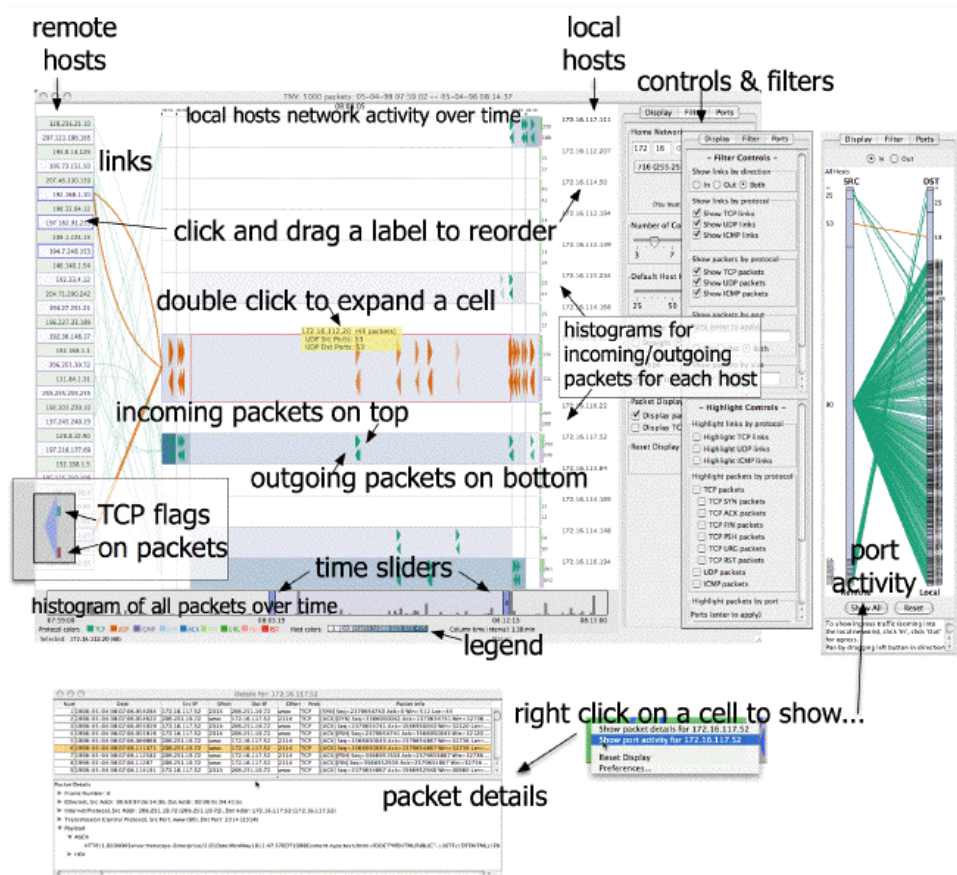


Figure 6.10:

IPv4). Diagonal lines indicate values that change at a constant rate between packets. The slope of the line indicates the direction and rate of change. The primary advantage of this view is the ability to compare up to about 1,000 packets at one time as we are only limited by the horizontal resolution of the monitor. This technique is largely independent of packet length as larger packets, even up to the traditional Ethernet limit of 1518 bytes, can contain only 256 possible bytes. The primary disadvantages are the loss of human readable packet contents and the inability to see which byte values occur more frequently. We can address this second shortcoming by color-coding the view by byte frequency instead of simply the presence or absence of bytes.

Binary Rainfall directly maps values in packets to pixels on the screen. Again there is one packet per horizontal line, but each pixel is illuminated based on the value of the byte at that offset in the packet. Bytes have 256 possible values so the binary rainfall view uses a 256 level color scale, to depict packet contents. By using this technique the actual structure of the packet can be seen.

Parallel coordinate plot allows from 2 to about 30 variables to be compared at one time. The concept is straightforward and it generates very distinctive fingerprints. This will form the basis of our next case study.

6.5.1 Case Study 30: Parallel Coordinate Plot

Load the pcap file into rumint thus

File -> Load PCAP Dataset

Select **Parallel Plot** under the **View Menu**

Change the **Number of Axes** Value to 3 and pick *Source IP*, *Destination Port* and *Destination IP* respectively for the axes.

Hit the **Play** button

Figure 6.11 shows the final image.

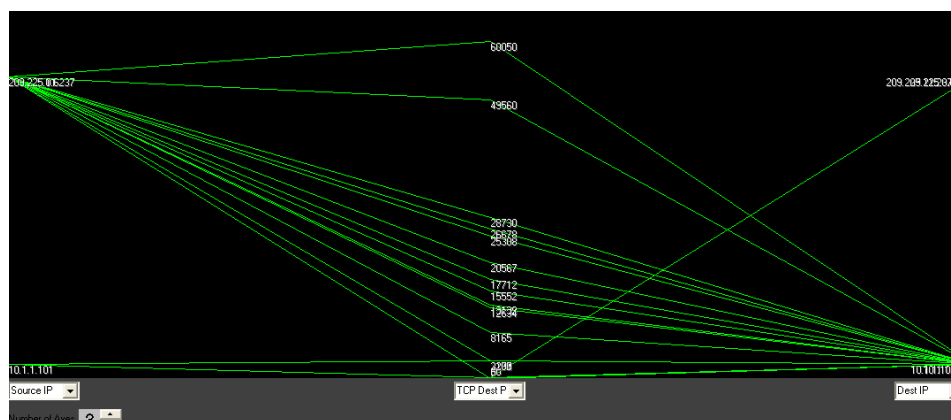


Figure 6.11:

The parallel coordinate plot is useful for rapidly characterizing large sets of packets and determining which fields are constant, near-constant, random and sequential, but it does suffer

from one significant shortcoming that is one or a few packets may take the same path and it is difficult to tell the difference. A solution will be to use the combined visualization technique.

6.5.2 Case Study 31: Combined Imagery Analysis

The combined visualization combines a two axis parallel coordinate plot with animated glyphs for each packet. It overcomes many occlusion problems because packets are animated and move outward off the screen. While we choose to look at the source IP address alongside the Destination Port, the drop down boxes in both the traditional parallel coordinate plot (Figure 6.11) and the combined visualization (Figure 6.12) allow you to flexibly choose any set of header fields based on your needs. The process is as follows:

File -> Load PCAP Dataset

Select **Combined** under the **View Menu**

Select *Source IP* for the first axis and select *Destination Port* for the second axis. Leave the **packets per quantum** value as *1*

Hit the **Play** button

Figure 6.12 shows the image representation

6.6 EtherApe

The EtherApe³ network monitor is another alternative for monitoring network data traffic. It is open source and offers a dynamic graphical interface, features IP and TCP modes, supports Ethernet, FDDI, PPP, and slip devices, filters traffic, and reads traffic from a *pcap* generated file and is capable of capturing straight off the network. In the following case study we will explore EtherApe and its customization features.

6.6.1 Case Study 32: Etherape Visualization

We start by installing. And as usual we call on *yum*

³<http://etherape.sourceforge.net>

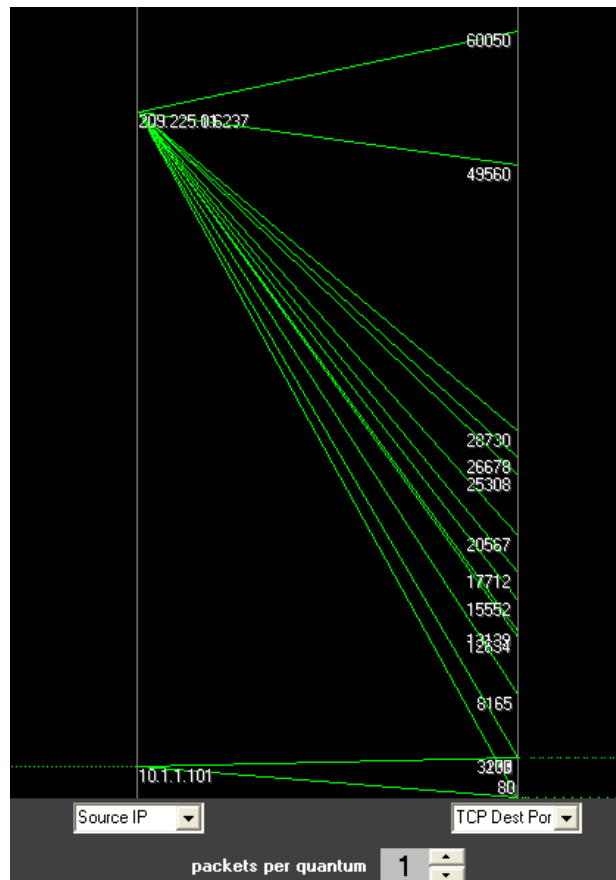


Figure 6.12:

Etherape Installation

```
# yum -y install etherape
```

And thats all.

Usage

EtherApe can be launched thus:

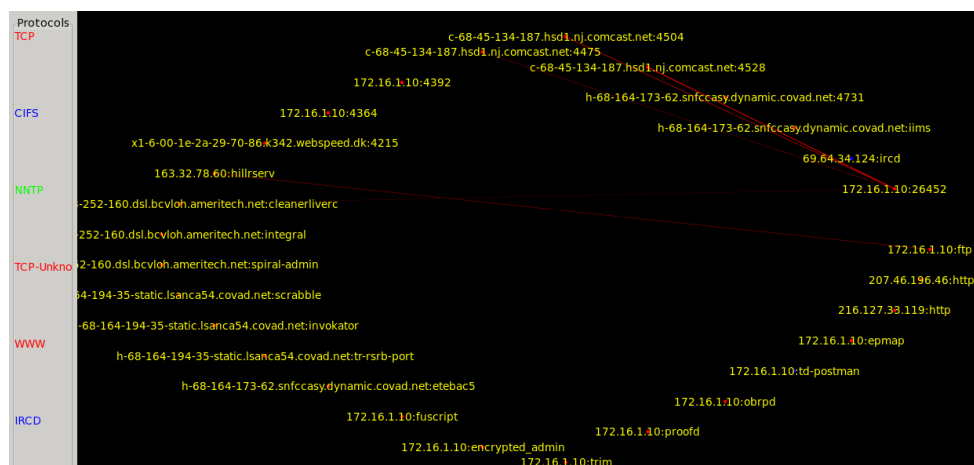


Figure 6.14:

6.6.2 EtherApe protocols

The Protocols window is a great tool that can be used for isolating erroneous packets on the network. From the *IP mode* view, we see connections to a TFTP and IRC server. We can use EtherApe to check on the this traffic that's moving through the network by opening up the Protocols window. This is done by selecting the **Protocols** check box under the **View** Menu. Figure 6.15 shows the output

Protocol	Inst Traffic	Accum Traffic	Last Heard	Packets
CIFS	0 bps	214 bytes	2'51" ago	1
DOMAIN	0 bps	783 bytes	1'19" ago	6
IRCD	0 bps	7.372 Kbytes	1'7" ago	32
NNTP	0 bps	242 bytes	2'50" ago	3
TCP	979 bps	8.441 Kbytes	0" ago	152
TCP-Unknown	0 bps	7.775 Kbytes	1'15" ago	40
TFTP	5.291 Kbps	67.119 Kbytes	0" ago	258
WWW	0 bps	11.350 Kbytes	47" ago	22

Figure 6.15:

As shown, TFTP alone accounts for 258 packets with accumulated traffic of 67.119 Kbytes by far the highest on the network. Even superseding that of WWW. This is a major concern.

Switching back to the main panel, we can then trace the offending host. Armed with this information, a filter or an access list can be developed quickly to block the offending host and port. The EtherApe protocols window is also a good way of optimizing the network.

6.6.3 Additional Options

EtherApe can be further fine tuned to a granular level by way of its configuration options. To do this, click the **Stop** button on the main window and then click the **Pref.** (preferences) button to open the Configuration window as shown in Figure 6.16

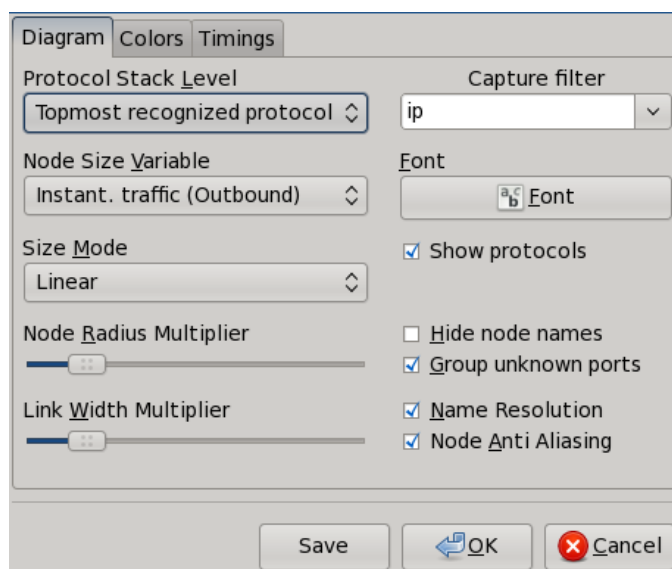


Figure 6.16:

The first tab on the Configuration window, the **Diagram** tab is used to set some of the monitor's protocol specifics. With the *Protocol Stack Level* configuration, we can specify the level of packet we want to monitor. There are five levels of the stack to watch: the *Topmost Recognized Protocol* (Level 1, physical medium), *Level 2 - ETHII*, *Level 3 - IP*, *Level 4 - TCP*, and *Level 5 - HTTP*. Using the Topmost level gives more specific information about the packets traversing the network as a result I tend to use this view quite a lot because I get a better holistic idea and picture of the packets traversing the network.

Node Size Variable is the next configuration. Node Size allows us to dictate the direction of

monitoring in EtherApe. Traffic is classed as instant and accumulated, and each class has three different directional patterns viz *In+Out*, *Inbound*, and *Outbound*.

On the **Timings** tab, we can adjust the *Diagram Refresh Period*. This period is in milliseconds, so don't let the default 800 deceive you. You must note that with this particular configuration the faster the refresh rate, the more difficult it is to follow the traffic. If you set this to the fastest possible setting (50 milliseconds), the monitor becomes useless because at this high refresh rate, the size of the traffic and the host addresses move around so quickly, it is practically unviewable. At a much slower rate of say 2,000 milliseconds, for example, too much traffic will also be missed. I suggest that on a fairly larger network, you should use a refresh rate of say between 400 and 800 milliseconds.

Also on this tab are *Node Timeouts*, *Link Timeouts* and *Global Protocol timeouts* option. Node Timeout dictates how long a node will remain in the Diagram without activity. The default setting is 6,000 milliseconds. In a multinode network, it would be advisable to set this number to a lower number so as to make the Diagram more easily readable. For instance, with a four-node network, the number of clients/servers and amount of traffic might be overwhelming. At this level of the network, there will be too many destination and source addresses shown on the screen at one time, which will prevent you from actually seeing the traffic. By allowing nodes to drop off the display (after a given amount of inactivity), the network traffic will be much more easily read. Same logic can be applied to Link and Global Protocol Timeouts.

6.6.4 Filters in EtherApe

One of the most important aspects of EtherApe is filters. In a network analyzer like Wireshark, the filter utility allows you to analyze the packets at a more granular level. For instance, if we have a large network that is struggling because of excessive erroneous traffic or malware, we will be unable to figure out where the bottleneck is coming from so by specifying which host or hosts we want EtherApe to monitor, we can more quickly troubleshoot the problem. Once a filter has been entered, we can save it and they will appear in the **Capture Filter** drop-down list. Thus if we have more than one filter we can switch between them quickly without having to reenter them again.

6.7 NetGrok

NetGrok⁴ applies well-known information visualization techniques (overview, zoom & filter, details on demand) and employs a group-based graph layout and a treemap to visually organize network data. NetGrok also integrates these tools with a shared data store that can read *pcap* formatted network captures, capture traces from a live interface, and dynamically filter dataset by bandwidth, number of connections and time. These techniques can be applied to static and real-time streaming packet data as shown in case study 33. NetGrok serves as an “excellent real-time diagnostic,” enabling fast understanding of network traffic and easy problem detection. Netgrok is a Java based visualization tool and imports *pcap* files directly. It can also listen on any available network interface.

6.7.1 Case Study 33: The Drill on NetGrok

NetGrok can be downloaded here⁵. It is pretty self contained. Just unzip and launch either the *netgrok20080928.bat* file (Windows) or *netgrok20080928.jar* (Linux). You need the Java Runtime Environment JRE 1.5 and above

Installation

The NetGrok visualization tool has two dependencies, both of which are provided in the download archive, although each requires additional installation steps. Unpack NetGrok and then follow the following procedure:

```
# wget http://netgrok.googlecode.com/files/netgrok20080928.zip
# cd Netgrok/lib/linux
```

Intermediate Procedure

There are a couple of things that must be done to get it working. On my notebook, I copied *jpcap.jar* and *libjpcap.so* files to the *ext* and *lib* subdirectories of my Java JRE installation as follows:

⁴<http://www.cs.umd.edu/projects/netgrok/>

⁵<http://netgrok.googlecode.com/files/netgrok20080928.zip>

```
# cp jpcap.jar /usr/java/jre1.6.0_16/lib/ext/  
# cp libjpcap.so /usr/java/jre1.6.0_16/lib/i386/
```

There is a slight typographical error in the *groups.ini* file found in the NetGrok root folder. Open it up. Either you take out the Private1-Wireless entry or replace the first '=' with a '-' so the second line in the file reads *Private1-Wireless=192.168.0.0/16* (instead of *Private1=Wireless=192.168.0.0/16*.)

Usage

To Launch on Linux just type:

```
# java -jar netgrok20080928.jar
```

or if you are using Sun JRE

```
# /usr/java/jre1.6.0_16/bin/java -jar netgrok20080928.jar
```

Figure 6.17 shows the NetGrok Interface.

For this case study we still examine the *capture1.pcap* file. Import it by selecting **Open Pcap File...** under the **File** menu. Figure 6.18 shows the screenshot of the interface.

As can be seen, The Netgrok interface has three tabs corresponding to three different views of network traffic. These are the *graph* view, the *treemap* view and the *edge* table.

6.7.1.1 Edge Table

The edge table can be used to view the underlying data being visualized by Netgrok at any given time. Figure 6.19 corresponds to the edge table of *capture1.pcap* file.

6.7.1.2 Graph View

The graph view represents network hosts as nodes in the graph and connections that exists between them. Connections can be seen by hovering over a host. Hosts sizes are represented according to the number of connections they make, so in this case the host 172.16.1.10 makes the most connections. Nodes in red are the nodes that utilize the most bandwidth, green

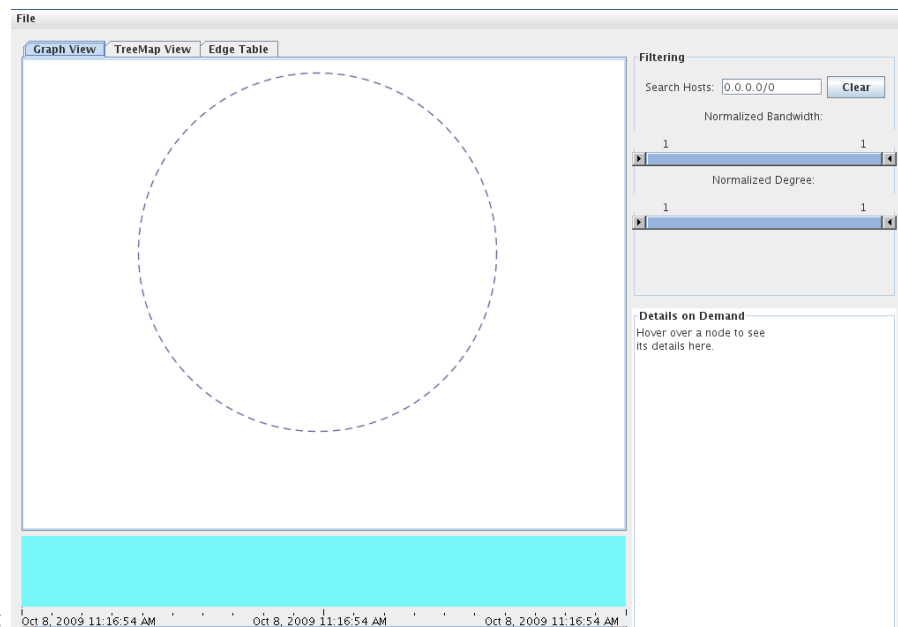


Figure 6.17: Oct 8, 2009 11:16:54 AM Oct 8, 2009 11:16:54 AM Oct 8, 2009 11:16:54 AM

utilize the least, and clear nodes mark zero-byte hosts. Hosts found inside the dashed ring are local, and all other hosts are regarded as foreign to the local network.

To zoom in on a node, double click on it; hovering over the node will produce details on demand in the NetGrok UI. Hovering over the red node (the node utilizing the most bandwidth) reveals its IP address as 172.16.1.10. Hovering over the brown node (the node utilizing the second highest amount of bandwidth) displays its IP address as 68.164.173.62. We can see the victim, 172.16.1.10, conversing most consistently with a server 68.164.173.62. Right clicking on a host also allows you to perform actions such as DNS lookups or displaying the network edges.

6.7.1.3 TreeMap View

NetGrok lets us display TreeMap views as shown in Figure 6.20. The TreeMap view displays hosts as rectangles within a hierarchy of network groups. A TreeMap view is better suited for displaying large PCAP files since they can view more hosts and connections without occlusion than the graph view. The size of a host within this view is proportional to the number of

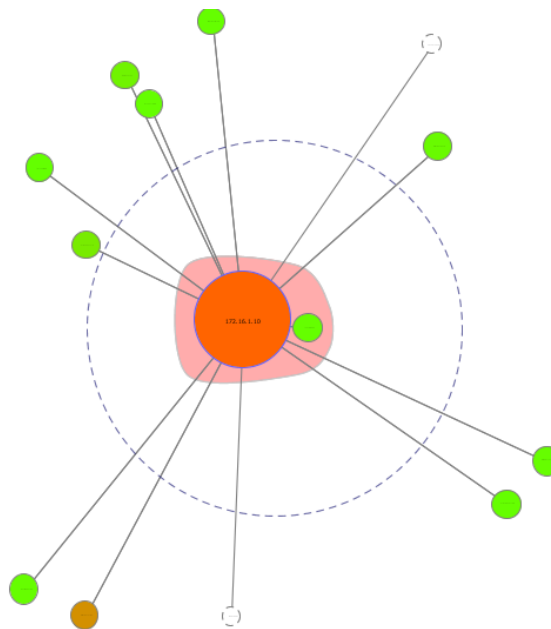


Figure 6.18:

target	address	groups	bandwidth	degree	lastseen	bandwidth	degree_rank	temp_time	time_selec...	is_zero_byte	host_addr...	hostn
13	null	null	40	1	1103769...	6.208866...	0.0	true	true	true		<unres
1	null	null	60	1	1103769...	9.313299...	0.0	true	true	true		<unres
12	null	null	120	1	1103769...	0.001862...	0.0	true	true	true		<unres
1	null	null	144	1	1103769...	0.002235...	0.0	true	true	true		<unres
11	null	null	968	1	1103769...	0.015025...	0.0	true	true	true		<unres
1	null	null	2336	1	1103769...	0.036259...	0.0	true	true	true		<unres
1	null	null	10222	1	1103769...	0.158667...	0.0	true	true	true		<unres
10	null	null	2130	1	1103769...	0.033062...	0.0	true	true	true		<unres
1	null	null	5593	1	1103769...	0.086815...	0.0	true	true	true		<unres
9	null	null	2110	1	1103769...	0.032751...	0.0	true	true	true		<unres
1	null	null	521	1	1103769...	0.008087...	0.0	true	true	true		<unres
8	null	null	178	1	1103769...	0.002762...	0.0	true	true	true		<unres
7	null	null	480	1	1103769...	0.007450...	0.0	true	true	true		<unres
6	null	null	6202	1	1103769...	0.096268...	0.0	true	true	true		<unres
1	null	null	64424	1	1103769...	1.0	0.0	true	true	true		<unres
5	null	null	480	1	1103769...	0.007450...	0.0	true	true	true		<unres
1	null	null	576	1	1103769...	0.008940...	0.0	true	true	true		<unres
4	null	null	520	1	1103769...	0.008071...	0.0	true	true	true		<unres
1	null	null	624	1	1103769...	0.009685...	0.0	true	true	true		<unres
3	null	null	80	1	1103768...	0.001241...	0.0	true	true	true		<unres
1	null	null	80	1	1103768...	0.001241...	0.0	true	true	true		<unres
2	null	null	188	1	1103768...	0.002918...	0.0	true	true	true		<unres
0	null	null	40	1	1103768...	6.208866...	0.0	true	true	true		<unres
1	null	null	200	1	1103768...	0.003104...	0.0	true	true	true		<unres

Figure 6.19:

connections it makes.

Again we can see from the illustration that the spyware server is constantly in communication with our internal host and creates a log clutter. The resulting NetGrok TreeMap view defines

two clear facts. 172.16.1.10 is quite clearly the top talker (98316 bytes – denoted as a large red cube as seen in Details on Demand), and it is on the local network, indicated by the thicker black line separating it from external hosts. Thin borders are used to separate hosts within each network group. NetGrok also includes useful filtering mechanisms to allow host isolation by IP, bandwidth, and degree (ingress vs. egress).

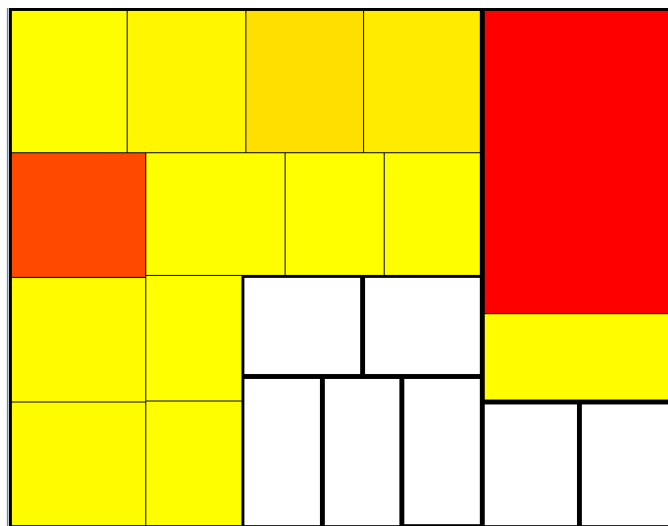


Figure 6.20:

The histogram at the bottom of the NetGrok screen shows a **Line Chart** see Figure 6.21. This is the chart of the number of connections made over a period of time.

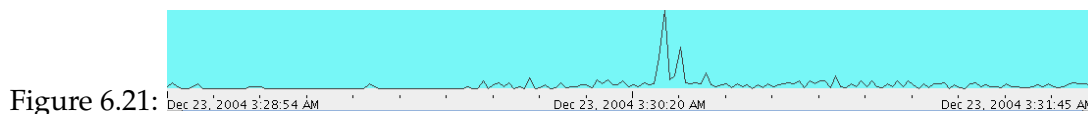


Figure 6.21:

The slider below can be used to inspect the number of connections made at a specific time. The results of sliding can be seen in the graph view. NetGrok is also capable of filtering by hosts, normalized bandwidth and normalized degree which can show a large number of zero byte hosts⁶ on the network.

⁶hosts that receive communication but sends nothing

6.7.2 NetGrok Advantages

Advantages of Netgrok are;

- Identification of DOS , DDOS attacks including attempts both internal and external angles.
- Locating individual zombie machines utilizing large proportions of bandwidth.
- Identification of hosts using covert communication channels or p2p applications.
- Attack detection (short or long period of attack)
- Spam detection on network layer

6.8 RadialNet

RadialNet⁷ is a network visualization tool. It visualizes network structures mapped by Nmap1.5.3.4 providing a graphical overview of the nodes on the network whilst identifying potential security holes. RadialNet is written in Python. To use the application, you need a Python interpreter, along with the PyCairo, PyGTK, and PyGObject packages for the graphics. Most distributions include the packages by default. Launch your favorite distribution's software management tool to complete the installation.

Note RadialNet visualization tool now comes prepackaged with the Nmap frontend called Zenmap. I have just chosen to view the application as a standalone. However. the operations of Zenmap network topography component is not too dissimilar to it.

6.8.1 Case Study 34: Visualization with RadialNet

RadialNet can be downloaded here⁸. At the time of writing, RadialNet is currently in version 0.44.

⁷<http://www.dca.ufrn.br/~joaomedeiros/radialnet/>

⁸<http://www.dca.ufrn.br/~joaomedeiros/radialnet/files/radialnet-0.44.tar.gz>

Installation

After downloading RadialNet unpack the tool in a terminal window by typing

```
# yum -y install pycairo, pygtk, and pygobject
# wget http://www.dca.ufrn.br/~joaomedeiros/radialnet/files/radialnet-0.44.tar.gz
# tar xfvz radialnet-0.44.tar.gz.
```

The data fed into RadialNet must be in the Nmap XML file format and can be passed in to the program either at launch time, by adding a `-f <filename>` flag, or interactively by selecting **Open**.

Following on from 1.5.3.6, we will attempt to visualize the *dmz.xml*⁹ output file generated by Nmap.

To launch the program, type

```
# cd radialnet && python radialnet.pyw -f dmz.xml
```

Even if you do not have any XML file for now, RadialNet ships with example in the *share/sample/* subfolder in the root folder. Figure 6.22 shows a typical example of a RadialNet generated graphic of the *dmz.xml* file.

By default, your computer (localhost) will be at the center of the map, shown as the black dot in the Figure 6.22. The colored nodes show the devices analyzed by Nmap. The color indicates the number of ports that are opened. Because open ports are potential security risks, computers with very few open ports are shown in green. Yellow is indicative of a medium-scale risk, and red nodes are high risk. There is not port references for white nodes. Squares depict routers, switches, or WLAN access points. The type is indicated by a light blue icon. Circles are “real” computers. Other icons might also appear. A yellow padlock stands for a computer with filtered ports, and a red wall is a firewall.

Clicking a circle or a square takes it to the center of the display. Right-clicking opens a context menu with detailed information on the selected network node (Figure 6.23). The **General** tab takes you to general operating system information and the active network interface. The **Services** tab lists the open ports, and **Traceroute** gives the the hop information from the localhost to the node selected.

⁹<http://inverse.com.ng/trace/dmz.xml>

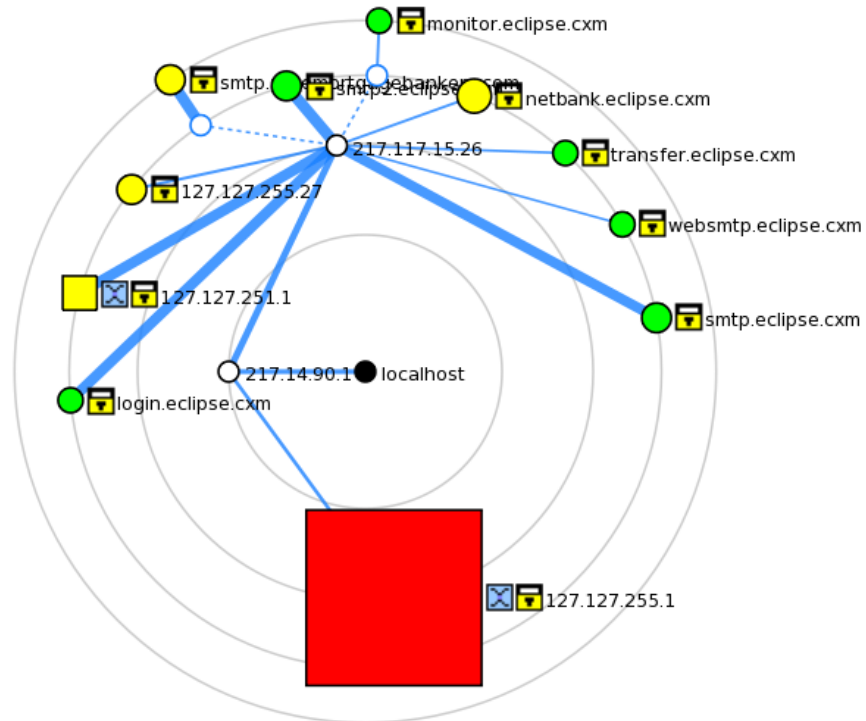


Figure 6.22:

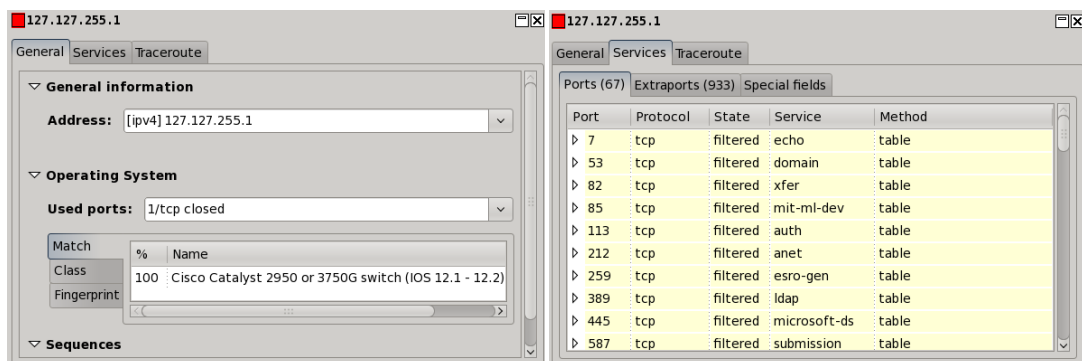


Figure 6.23:

The **Tools** -> **Host Viewer** menu item takes you to a scalable overview (Figure 6.24) of the detailed information. The left hand side of the window shows the nodes analyzed, with the information from the pop-up window on the right. Traceroute shows connections between individual nodes on the map, indicating the routes that data will take from localhost to the border nodes. If traceroute information is missing, the path is shown as a dotted line.

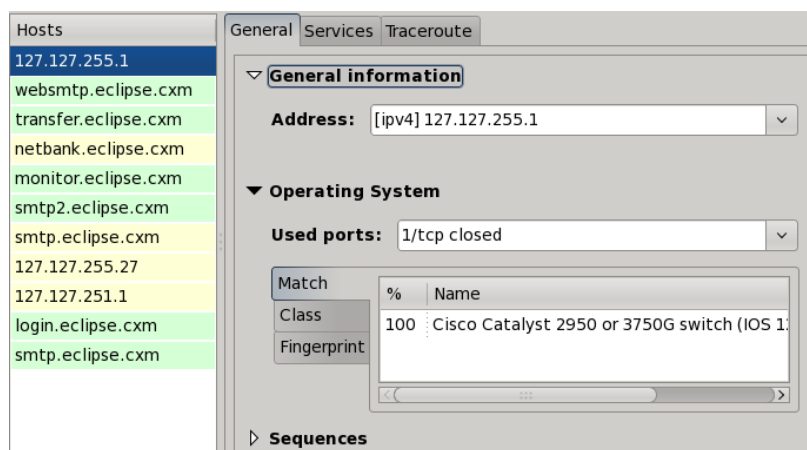


Figure 6.24:

RadialNet has three useful extra buttons in the top right of the window. **Fullscreen** toggles the full screen view on or off. The **Fisheye** button lets you toggle between a flat display and a fisheye view. The fisheye view assigns more space to the center of the map than to the border areas, thus making the information at the center easier to read. A slider appears at the bottom of the window, which you can use to change the fisheye view aspect. The flat view allocates the same amount of space to all the nodes on the map. The **Controls** displays a navigation aid on the right side of the window. With this tool, we can zoom in or out of the map or toggle between address and hostname views. Also, you might want to try out a few of the parameters to get that perfect view mode for your own requirements. All these can be viewed in Figure 6.25

RadialNet gives analysts a tool for visualizing the network that clearly identifies potential risks and with its integration into Zenmap, visual vulnerability analysis and network mapping are no longer restricted to text-based output.

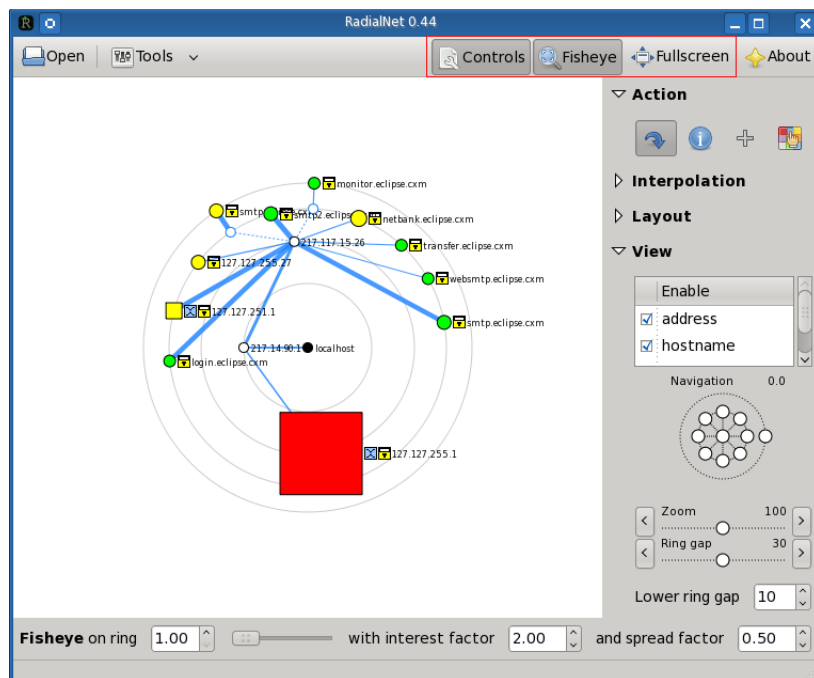


Figure 6.25:

6.9 InetVis

InetVis¹⁰ also referred to as Internet Visualization is a 3-D scatter-plot visualization for network traffic. It is like a media player for network traffic - a visual tool for network telescope traffic analysis. InetVis was adopted from Stephen Lau's *Spinning Cube of Potential Doom*. It is a tool for reviewing packet capture files, but may be useful in other endeavours. For example, InetVis has been used to verify and critique the accuracy of scan detection algorithms in the Snort and Bro IDSes.

Whilst many other visualization tools make use of lines as a metaphor for connection, the 3-D scatter-plot of points proves to scale well with larger volumes of data. The visualization makes network and port scanning readily evident as horizontal and vertical lines respectively. InetVis visualizes packet captures of network traffic using Libpcap to either capture live traffic from the default interface or replay traffic from a pcap file

¹⁰<http://www.cs.ru.ac.za/research/g02v2468/inetvis.html>

6.9.1 InetVis Plotting Scheme

Below itemizes the scheme InetVis employs to visualize the 3-D scatter plot.

- Destination address (home network) plotted along blue x-axis (horizontal)
- Source address (external Internet range) plotted along red z-axis (depth)
- Ports (TCP and UDP) plotted along green y-axis (vertical)
- ICMP traffic plotted below TCP/UDP cube grey/white ICMP plane

Figure 6.26 shows this representation graphically.

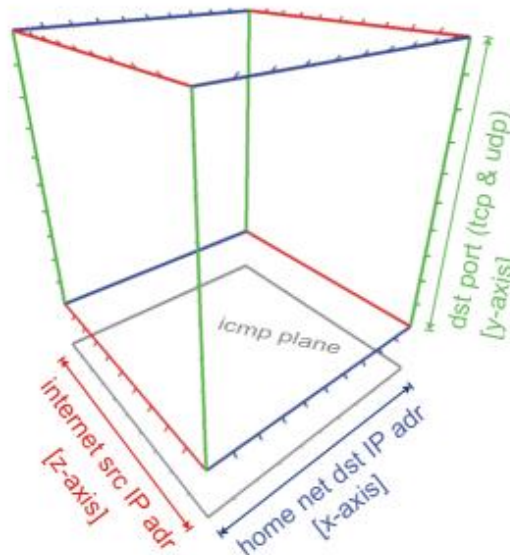


Figure 6.26:

Note that when you start to work with packet captures, InetVis will demand that you set the Home Network Range (Blue x-axis) but it isn't necessary, largely because InetVis sees the home network as the destination and all Internet addresses as the source. Most cases just click OK when it prompts. But as is typically the case, when visualizing a malware infection, our perspective is exactly the opposite, where the host infected is the source and all the other partakers are the destination.

6.9.2 Case Study 35: Scatterplot Visualization

InetVis runs on both Windows and Linux and they can be obtained here¹¹. To demonstrate the workings of InetVis we will use a Storm infection¹² pcap file *capture6.pcap*¹³

Installation

Unzip the download and double click on the *inetvis.exe* to launch.

Usage

Leave all the defaults settings

Select **Open...** under the **File** menu.

Navigate to the *capture6.pcap* file and double click it.

Note that when you start to work with packet captures, InetVis will demand that you set the Home Network Range (Blue *x-axis*) but it isn't necessary, largely because InetVis sees the home network as the destination and all Internet addresses as the source. Most cases just click **OK** when it prompts. But as is typically the case, when visualizing a Storm infection, our perspective is exactly the opposite where the host infected is the source and all the other P2P partakers are the destination. Figure 6.27 shows the output graphically.

Streamlining the range of ports slightly from the general default of 0-65535 to the range of 4000-35000 to create a more focused view, you'll notice that the resulting visualization shows a single point of reference along the red *z-axis* (source) as the infected host is utilizing a Class C address of 192.168.248.105. We can also observe that it sends a prism of visualized disease across the blue *x-axis* (destination) given the hundreds of unique hosts in the botnet's P2P mesh that the victim chatted with over the four minute capture. Finally, note the full rainbow across the *y-axis* (green) that represents the port range as this particular capture spread the whole spectrum. See Figure 6.28

InetVis allows the rotation of the cube. This is done by hold down the left mouse button on the left side of the view and dragging it to the right - the cube will rotate in the direction of the pointer. Lastly InetVis allows recording by supporting all the three recording methods

¹¹<http://www.cs.ru.ac.za/research/g02v2468/inetvis.html>

¹²June 2008 ISSA Journal

¹³<http://inverse.com.ng/trace/capture6.pcap>

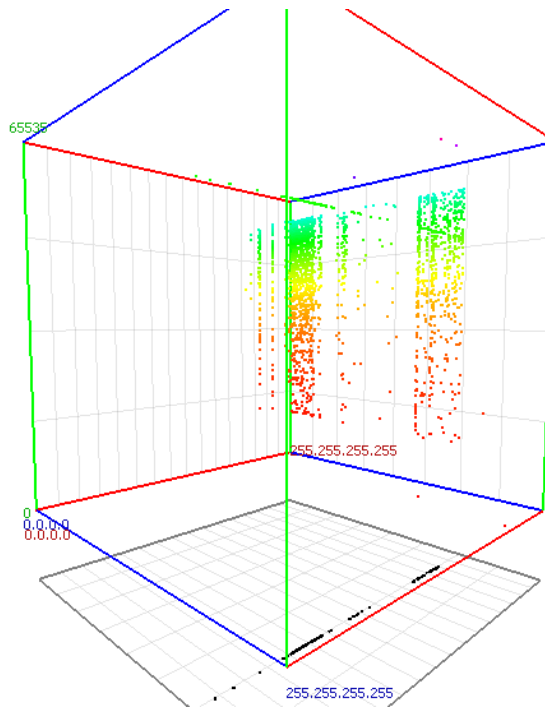


Figure 6.27:

- recording to capture file, taking a single image snapshot, or dumping rendered frames to image files and they can be used simultaneously in conjunction with playback. For more details view the online manual here¹⁴.

6.10 GGobi

GGobi¹⁵ is an open source visualization program for exploring high-dimensional data. It is used in explorative data analysis and it provides highly dynamic and interactive graphics such as tours and familiar graphics like the scatterplot, bar chart and parallel coordinates plots. Plots are interactive and linked with brushing and identification. It includes 2-D displays of projections of points and edges in high-dimensional spaces, scatterplot matrices, parallel coor-

¹⁴<http://www.cs.ru.ac.za/research/g02v2468/inetvis/0.9.3/doc/inetvisdoc.html>

¹⁵<http://www.ggobi.org>

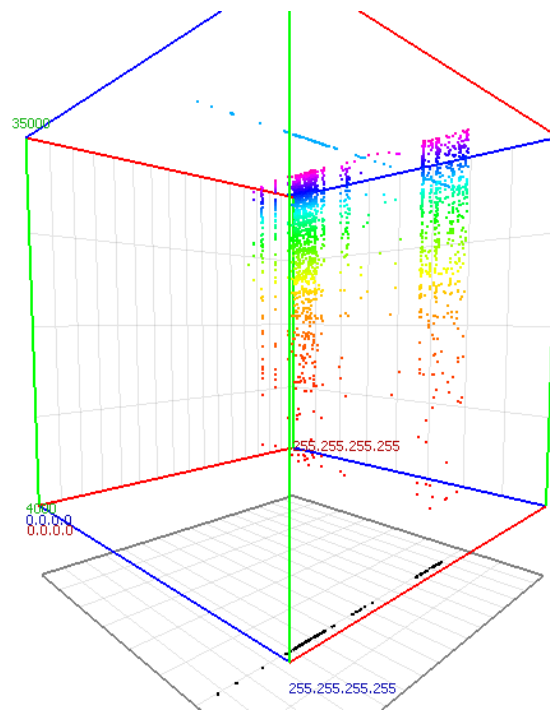


Figure 6.28:

dinate, time series plots and bar charts. Projection tools include average shifted histograms of single variables, plots of pairs of variables, and grand tours of multiple variables. Points can be labeled and brushed with glyphs and colors. Several displays can be open simultaneously and linked for labeling and brushing. Missing data is accommodated and their patterns can be examined.

6.10.1 Case Study 36: Interactive GGobi

GGobi can be downloaded here¹⁶. make sure to download the latest version of GTK for Windows if you intend running it there.

¹⁶<http://www.ggobi.org/downloads/>

Installation

Installing on Windows is quite direct. Just launch the *GTK* executable file followed by the *ggobi.exe* file and follow the instructions thereafter. The easiest way to install on Linux (Fedora) is the trusty *yum*.

```
# yum -y install ggobi
```

This will install *ggobi* with the associated dependencies.

Usage

For this case study we will examine the http download pcap (*capture5.pcap*). *GGobi* takes XML or CSV file format as input so we need to convert our pcap file first. The easiest is to convert to CSV with the *tcpdump2csv.pl* script packaged with *Afterglow*. I have made this available for general use by copying the script into a directory in my *\$PATH* (*/usr/local/bin*) and given it necessary permissions. I then start thus;

```
# tcpdump -vtttnnelr capture5.pcap | tcpdump2csv.pl "sip dip dport" \  
> capture5.csv
```

Then I open this file in *GGobi* thus

```
# ggobi capture5.csv
```

This can also be done by selecting **Open** in the **File** Menu and navigating the file system to find the *capture5.csv* file upon launching the *GGobi* GUI. Two windows will appear, the *GGobi* console and a scatterplot, as shown in Figure 6.29

The console has a panel of controls on the left, labeled **XY Plot**, and a variable selection region on the right. You can see that the scatterplot contains a 2-dimensional projection of the data, a plot of Area vs Region. Move the mouse to one of the variable labels in the variable selection region, and leave it in place until the tooltip appears, explaining how to select new variables for the plot. Begin to get a feeling for the data by looking at several of the 2d plots.

Using the **Interaction** menu, choose **Identify** (Figure 6.30). Look at the buttons inside the left-most portion of the *GGobi* console. Notice that they're contained by a frame labeled *Identify*,

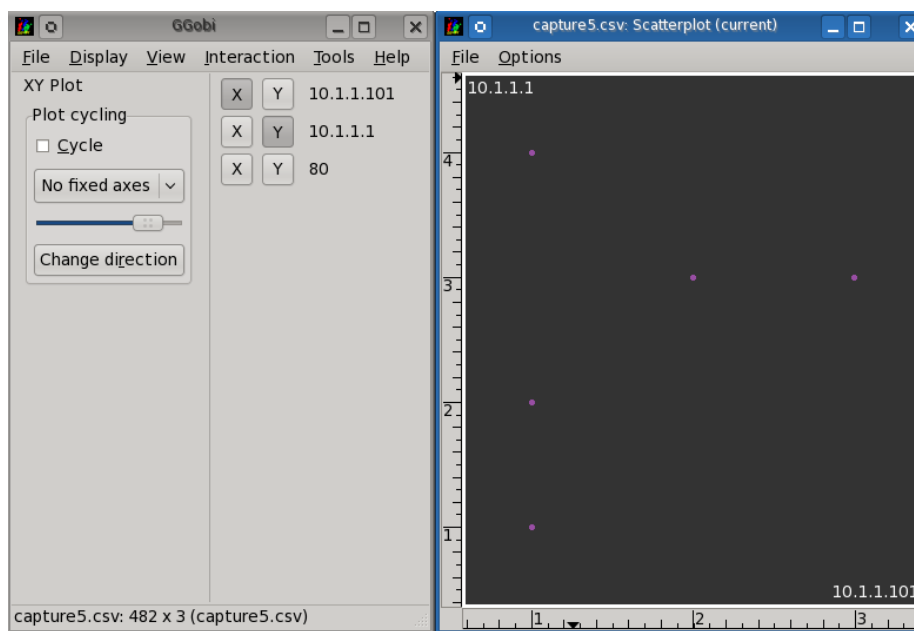


Figure 6.29:

which is the interaction mode of the current plot. This frame contains most of the row labeling controls. There are several options for writing labels in the plot window. The default is the record label. Highlight the IP addresses 10.1.1.101, 10.1.1.1 and the port 80, then the labels will be the geographic area. Move the cursor around the plot window using the mouse, and observe the labels of the point closest to the cursor. The labels show the geographic area where the sample was taken.

Open a second display window, showing a bar chart (Figure 6.31), using the **Display** menu. Notice that the new window has a narrow white band drawn around the outside of the plotting area: that means that the new window is now the “current display” and its plot is the “current plot.” Click in the plotting region of the other scatterplot to make it the current plot, and notice what happens in the console when you alternate between the two. Both the controls and the variable selection region correspond to the current plot.

Now set up a plot in the first scatterplot, and display Region in the bar chart, and make the bar chart the current plot. Using the Interaction menu, choose **Brush**.

Look at the buttons and menus inside the leftmost portion of the GGobi console. Notice that they’re contained by a frame labeled **Brush**, which is the view mode of the current plot. This

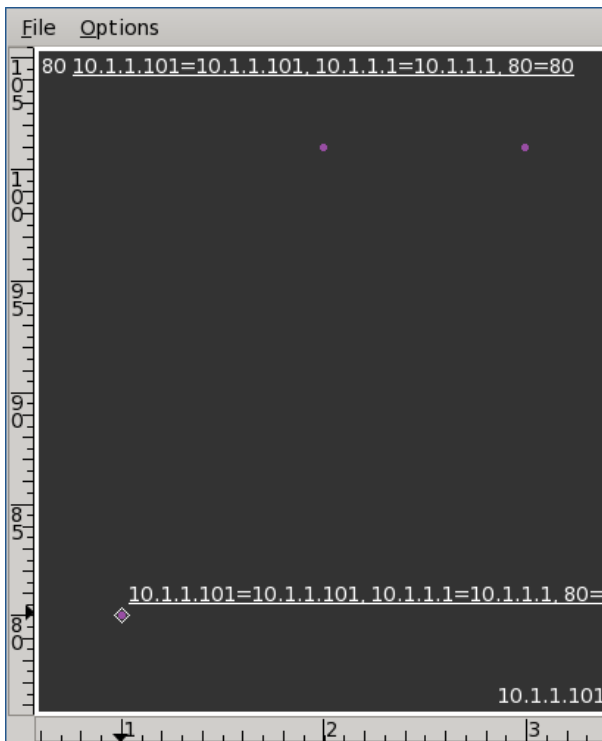


Figure 6.30:

frame contains most of the brushing controls. A few of the brushing controls, though, are in the **Brush** menu in the display menu bar, and some associated tools are in the **Tools** menu.

The rectangle that appears in the current plot is the “paintbrush,” and dragging it over bars (or in a scatterplot the points) changes their color. Change the color of the brush by opening up the **Choose color & glyph** panel. Hold down the left button and drag the mouse to paint the first region, then the second and third regions, or click on the bars. Since you’re brushing in the **Transient** style, the points resume their original color when the paintbrush no longer surrounds them (Figure 6.32)

While you brush, keep an eye on the plot and notice where the painted points fall in that scatterplot (Figure 6.33). Using two linked 2d plots is one way to explore the relationship among three variables.

You can also view the **Time Series graphs** (Figure 6.34), **Parallel Coordinates display** (Figure 6.35) and **Scatterplot Matrix** (Figure 6.36) all under the Display menu.

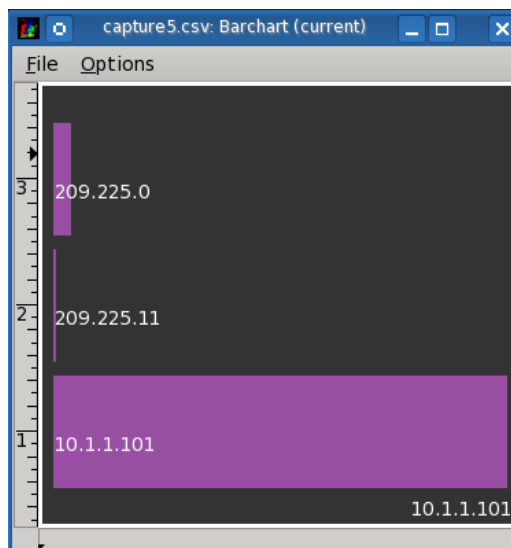


Figure 6.31:

GGobi is a massive tool and this has just been a basic usage of GGobi. Detailed data analysis and visualization functionality of GGobi can be obtained from its manual located here¹⁷

6.11 DAVIX

DAVIX¹⁸, is a live CD for data analysis and visualization. It puts together the most important free tools for data processing and visualization in the form of a live CD. There is no need for any hassle to build the necessary tools to get started with visualization. It allows full concentration on data analysis.

The DAVIX live CD is based on SLAX and features out-of-the-box hardware support for graphic cards and network adapters. SLAX is based on Slackware and follows a modularized approach. It can even be installed on USB drives and provides mobile analysis capabilities. All tools are accessible through a start menu and accompanied with links to external manuals and tutorials. All the necessary information to get started with the tools is just a click away. The important tools in DAVIX are organized in three categories depending on their use within

¹⁷<http://www.ggobi.org/docs/manual.pdf>

¹⁸<http://davix.secviz.org/>

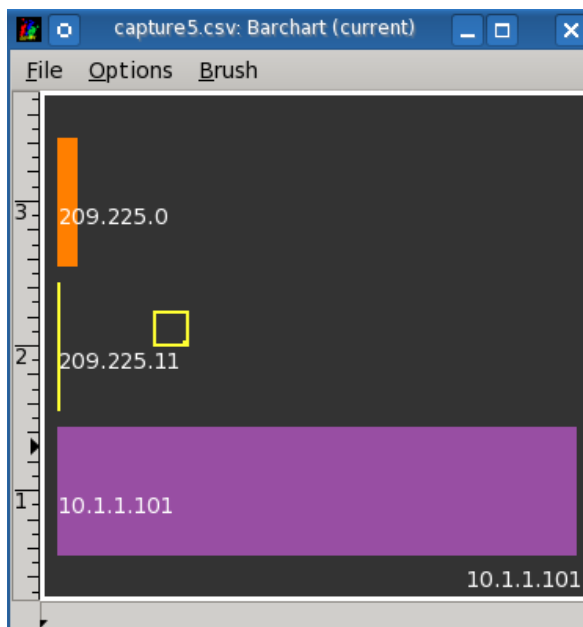


Figure 6.32:

the analysis process:

- Capture (C)
- Process (P)
- Visualize (V)

6.11.1 DAVIX Tools

The following table displays a list of all the visualization tools packaged with the DAVIX distribution. It is well worthwhile to download and give it a spin.

Afterglow	TreeMap	GGobi	GraphViz	MRTG/RRD
InetVis	EtherApe	Dot and Neato	Rumint	R Project
RT3DG	Walrus	NVisionIP	LGL	Guess
TimeSearcher	TNV	Parvis	Cytoscape	Shoki
Ploticus	Mondrian	Tulip	glTail	Gnuplot

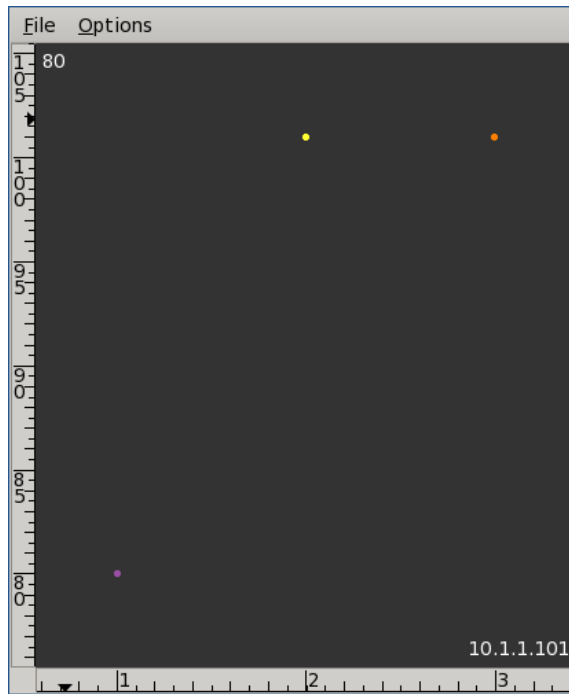


Figure 6.33:

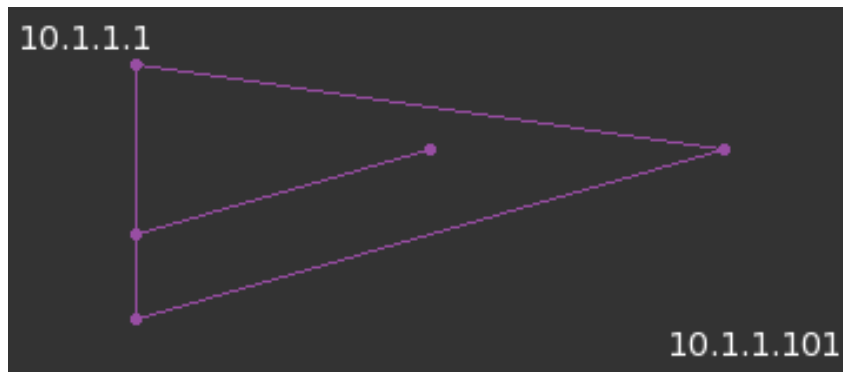


Figure 6.34:

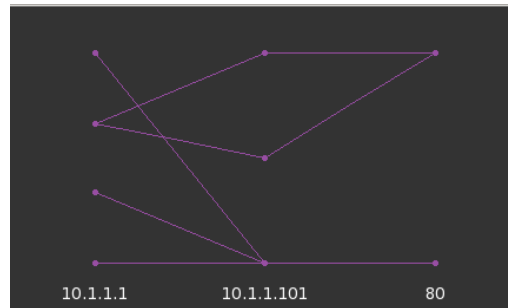


Figure 6.35:

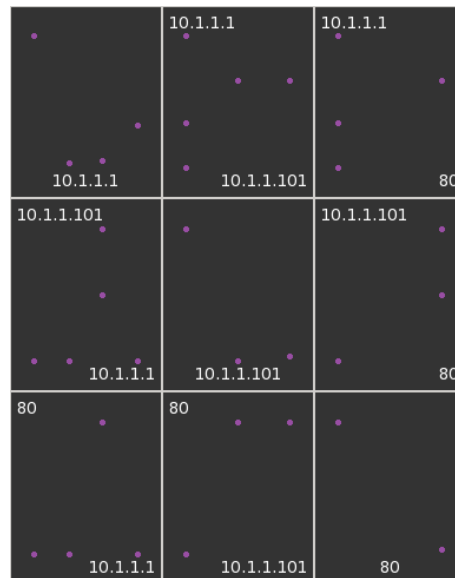


Figure 6.36:

6.12 Summary

In this chapter, we focused on using data visualization techniques to analyze, evaluate, and gain insights into the results of security datasets. We started by looking at the evolution of tools of visualization before categorizing and examining majority of the free and open source security visualization tools that will add veritable value to an analysts daily security visualization activities.

Part IV

GOVERN

Chapter 7

Security Data Management

This chapter is all about the management of enterprise information. Security Data Management (SDM) is the process of managing information security data resources. Data management also refers to the overall management of the availability, usability and integrity of the security data acquired and employed in an enterprise. It is a field that has security data governance at its core and incorporates data quality, data protection, data leak prevention, data retention and privacy of data in an organization. Through security data management, enterprises are looking to exercise positive control over the processes and methods used by their custodians to handle data.

In addition, the management of security data is an involving process of maximizing the value of data within an organization and the management and usage of acquired data. It helps minimize the risks that come with handling data and helps organizations comply with an intricate array of regulatory compliance that surrounds data.

With proper data management, organizations can cut costs and improve control over its information, which not only increases data security and privacy but improves responses to changing regulatory requirements. Not having in place any data management strategy increases the risks of data breaches, including identity theft and fraud, which can erode trust in an organization, trigger financial or legal penalties, or reduce confidence among employees, customers, and stakeholders.

A top-down approach to information security management starts with an understanding of the data life cycle—the acquisition, extraction, preservation, analysis, processing and the eventual deletion of information. It also requires all relevant players to understand regulatory demands and translate them into clear and specific business and technology requirements.

Armed with that fore knowledge, organizations can then create and adopt a technology framework which includes controls for safeguarding privacy.

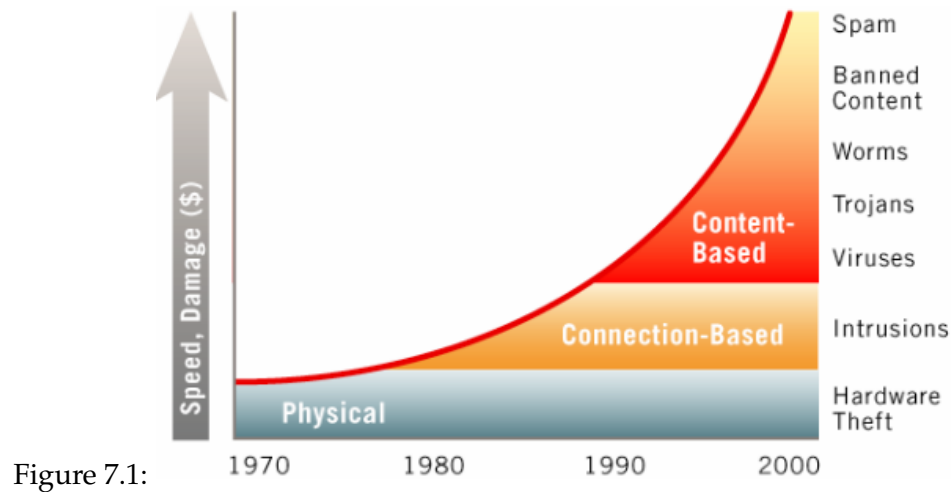
7.1 Threat Evolution

The proliferation of public and private networks and the increasing sophistication of network protocols and applications have driven a rapid escalation in the number and severity of attacks against computing systems.

Security threats have evolved from relatively simple, connection-based attacks to more complex content-based attacks such as viruses, worms, and trojans. See Figure 7.1. At the same time, organizations are saddled with the management of other content-based threats, such as email spam and inappropriate application content that reduces productivity and exposes them to substantial liability. These new content-based attacks are neither detected nor stopped by traditional measures like firewalls, causing a search for newer, more effective technologies. Recently, many industry practitioners have heralded the benefits of a technology referred to as Deep Packet Inspection, promising better protection against content-based threats. Deep Packet Inspection is more effective than Stateful Inspection for these types of attacks. Specifically, Deep Packet Inspection can detect active viruses, trojans and worms, and is completely effective for viewing erroneous Web content and spam. With an appropriate platform, DPI technology can be deployed in high speed networks without impacting the performance of network based applications.

Early network protocols, such as FTP, Telnet and RPC to mention a few were relatively simple and required action by a dedicated intruder with a sustained connection to a remote system to launch an attack. The response to these types of attacks was the development of connection-oriented security systems, called Stateful Inspection firewalls, which control access to computing resources on a network by selectively allowing or denying the establishment of remote connections based primarily on the identity of the sender and receiver using the TCP/IP three-way handshake.

However, we are in an era where applications have become a lot more complex where protocols are used to carry much broader and richer content. These changes have been exploited by attackers to develop more effective, content-based threats that circumvent connection-oriented security and that also have the ability to reproduce and spread automatically. Content based threats are able to bypass connection-oriented Stateful-Inspection firewalls because they are typically delivered through connections that are allowed and trusted. Content-based



threats include viruses, Trojans, worms, banned content and spam, and are readily propagated through email, web pages and other real-time communications applications. One such example is the recent conficker worm. The propagation speed of content-based threats and the resulting damage they can cause is considerable.

7.2 Shallow Packet Inspection

Much of this book has centered around one form of packet analysis or the other. In this section we take a look at packet inspection in a generic form. Typical packet inspection process extracts basic protocol information such as source and destination IP addresses and other low-level connection states. This information mostly resides within the header of the packet and consequently reveals the communication intent. Figure 7.2 shows the packet inspection process¹

Packet filtering devices all share this same basic mechanism: As an IP packet traverses the packet filter, the headers are parsed, and the results are compared to a set of rules defined by an administrator. The ruleset, commonly based upon source and/or destination IP address, source and/or destination port, or a combination of the two, defines the type of traffic that is

¹Digging Deeper into Deep Packet Inspection - Allot Communications

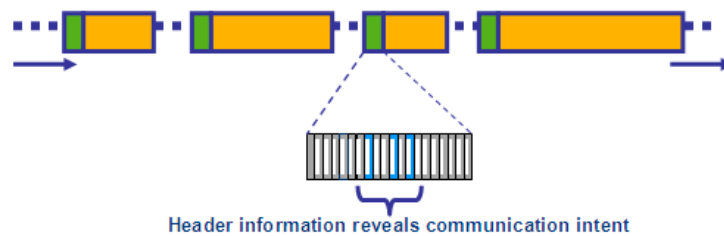


Figure 7.2:

allowed or denied. Interestingly, some early packet filtering implementations required that the system admin define specific byte fields with the packet headers, and the specific byte patterns to match against. The point here is that packet filtering as well as the code that performs these tasks based upon parsing of IP headers have been available for many a year.

The inspection level in the process is not deep enough to reach any application level related outcome. For instance, if a packet is the result of an application trying to set up additional connections for its core operation, examining the source or destination IPs as they appear within the packet header will not reveal any valuable information regarding the connections to be used in future as requested by the application. In addition, it is quite common that the necessary information is spread over several packets so examining the header alone overlooks the complete handshake.

7.3 Deep Packet Inspection

As a result of the limitation of the standard packet inspection, a cursory look at deep packet inspection is perhaps in order. DPI provides application awareness. DPI is one of the foremost technologies used in identifying and authenticating protocols and applications (IP flows or sessions in general) conveyed by IP. The "deep" in DPI simply refers to the fact that in addition to viewing the header information as packets pass through them, these devices also move beyond the IP and TCP header information to view the payload of the packet. See 7.3. The goal is to identify the applications and reassemble sessions on the network

Whilst privacy is a legitimate concern, the importance of DPI is growing exponentially and instances of its value are provided shortly. A more general term called Deep Packet Processing (DPP) which encompasses actions taken on the packet such as modification, blocking or filtering is also gaining in popularity. DPI and DPP are commonly used interchangeably.

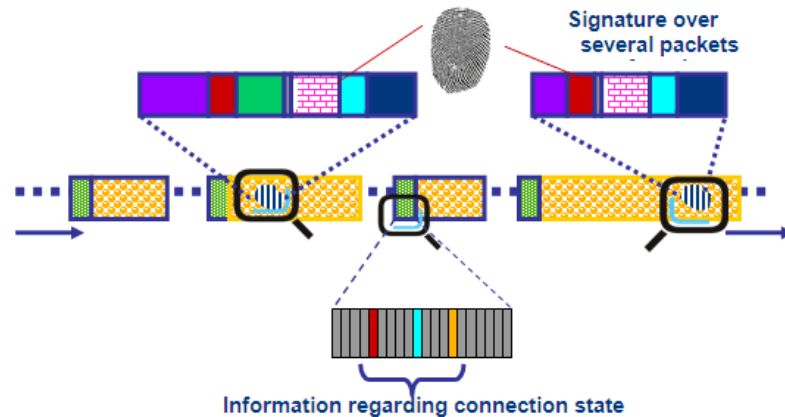


Figure 7.3:

DPI is a breakthrough concept in information security that serves as the foundation of many current and next generation services. Today's IT spending tend to be proportional to technologies that leverage DPI. These may include routing and switching devices, firewalls, packet filters, intrusion detection & prevention (IDS / IPS), regulatory compliance and leak prevention, traffic management, etc.

DPI promises a more secure and self containing network infrastructure. However, it also serves up scenarios that need comprehensive technical, operational, legal and ethical consideration by the security community and DPI stakeholders in general in order to gain universal acceptance and deliver its full benefits. Some of the issues regarding DPI include:

- Its potential impact on privacy and censorship and the appropriate public policy and regulatory frameworks
- Its relevance to network architecture evolution so as to address the needs for adequate compute power, throughput, flexibility and functionality
- Trends and efforts that will enable or improve interoperability, network and service management, or consolidation of various DPI technologies
- Driving forces for DPI solutions to expand into new markets, bigger footprint and higher value-added solutions?
- Its role in national security.

This chapter will try to enlist solutions to some of the DPI challenges especially also as it relates with security data management and governance.

7.3.1 DPI Analogy

Analogies are relatively interesting instruments to illustrate deep technical concepts for a non-technical audience. Sometimes it becomes a bit difficult to call them right. A poor analogy is intentionally used to convey a personal opinion instead of technical facts – or worse, to bring about fear, uncertainty and doubt (FUD).

An analogy that is particularly popular in the recent debate about DPI is that of the post office system. An analogy where DPI is compared to *opening* a letter and *reading* its content. One could argue about opening because a sealed letter is clearly marked ‘Do not open’, a network packet on the other hand, is not (that is if not encrypted). But this is highly debatable. DPI systems by no means ‘read’ or even ‘understand’ the communication content. Instead, they scan for certain patterns to classify the protocol or application that generated the packets used in transmitting the content. Such systems only find what they are looking for, that is if they do not scan for words such as ‘guns’ or ‘weapon’, they will not know if they are available or not. In other words, there is no content indexing of network packets by DPI unlike what search engines do for web pages.

If we really have to use the analogy of ‘letter reading’, then it has to be postcards as opposed to letters, and the ‘reader’ should be the one who does not understand the language of the letter and who only scans certain parts of the contents for matching symbols from a list of symbols – or patterns. It is therefore, important to understand that DPI is not necessarily a violation of privacy

7.3.2 Header Vs Payload

The architecture of any network node follows a standardized model. Each implements a subset of functions necessary for end-to-end information communication and data transmission. Defined interfaces between these layers provide a hand-over point for data. In a transmitting node, each layer receives data via this interface from its upper layer. These data makes up the payload for the current layer. Data is processed and a header is added at the head of the packet. (Sometimes, trailing information is also added to the packet, usually redundancy checks or padding information). This process repeats itself at every layer. For instance, let

us take a look at what happens when we send a mail. After composing the mail say with Thunderbird, and sending, the following is what transpires:

1. The message including mail header fields such as *subject*, *from*, *to*, *attachments*, etc is encoded in the Internet Message Format (IMF).
2. The IMF-encoded message is delivered to the SMTP handler, which further encapsulates the IMF payload by adding its own header.
3. The SMTP packet is then handed to the sending host's TCP instance, that again adds its header (with port numbers identifying the communicating application, plus other, connection-state and flow control information) to the SMTP payload data.
4. The TCP segment is passed on to the IP instance, that adds an IP header with IP source and destination addresses.
5. The data link layer (Ethernet in most cases) takes the IP packet and encapsulates it in an Ethernet frame, again adding a header with Ethernet addressing information (source and destination MAC addresses).
6. Only now this Ethernet frame is put as an electromagnetic signal, representing the bit values (0s and 1s) that comprises the frame, onto the copper or fiber-optical cable.

The same process happens again at the receiver but in the reverse order. This illustration shows that there is no distinction between header and payload on the Internet. An IP packet is an Ethernet frame's payload, a TCP segment (i.e. the TCP packet) is the payload of an IP packet, and so on.

So where exactly does DPI start? There is no real answer to this, but a clear line can be drawn by looking at the Internet's layer packet delivery process. For packet forwarding on the Internet, the applications sending and receiving the packets are irrelevant. Packets are sent from one host, represented by the source IP address to another represented by the destination IP address. These two addresses are the only information required by Internet nodes (the sending and receiving hosts and the intermediate routers) to deliver a packet. So one could assume a situation where only the transmitting end systems should look beyond the IP header at TCP/UDP port numbers. That is necessary to deliver data to the correct application, of which several may run on any given host. At any other point along the network path between the transmitting hosts, the look up needs only go as deep as the IP header, as this is all that is necessary to send the packet on its way.

7.3.3 DPI Uses

DPI technology is unique. It is currently the only known way to accomplish certain governmental security directives. DPI also has the potential to be put to real good use. For example, Distributed Denial of Service (DDoS) attacks are almost impossible to thwart. If DPI technology was in place and properly configured, it would detect any trace of DDoS packets and filter them out. Some more potential uses are listed below:

Network Security DPI's ability to inspect data streams at such low level will prevent malware from either gaining entry or exiting the network.

Network Access DPI creates conditions where access policies are easily enforceable due to the its inherent deep inspection of packets.

Enforcing AUP Organizations can use DPI to ensure that their acceptable use policy is enforced. For example, DPI can locate illegal download as well as sharp utilization of available bandwidth.

Quality of Service DPI would allow the organizations to dynamically allocate bandwidth and network resources.

7.3.4 DPI Concerns

In the course of writing this book, I conducted an extensive reasearch on DPI and its growing concerns. However, there was one major aspect that perhaps was an oversight in the argument for or against - that of spyware, viruses, trojans, DoS and general malware behavioral analysis.

As a consultant of several years experience within the information security arena, I regularly conduct security assessments viz-a-viz penetration tests and as far as I know I track quite a number of zombies and bots participating in delivering millions of spam messages on a daily basis, resulting in sometimes a distributed denial of service attack (DDoS), identity theft, fraud and so on. I also typically identify various legitimate sites that have been broken into with backdoor leftovers mostly for subsequent connection.

The magnitude and scale of the problem is mind boggling, and isn't getting any better. It is however not recognized by entities not specialists in the field because by and large organized crime is experienced at concealing their activities and hiding their traces - infact easily deceiving most experts. Take for instance, DNS attacks, an attack on domain name resolution. It is one of the more recent attacks and perhaps the most threatening. Most times, when you think

you're transacting business with say your financial service provider, or bank, someone may in fact be snooping and eavesdropping on your conversation after DNS lookup compromise. Deploying Secure Sockets Layer (SSL) helps in some cases, but not always because there are attacks that can compromise that too. It is becoming increasingly dangerous to your online transaction and your Internet privacy. The fraudsters are getting smarter with new tools and techniques, security has to simply keep pace.

Overall, discussions about network neutrality, privacy and other concerns won't amount to much if users can no longer trust the services they use, nor indeed even their own computers. Encryption is seldom a panacea. Whilst true that some service providers are using DPI in ways that may not be ethical such as spying and monitoring, these acts, to all intents and purposes, were still possible even without DPI and will remain so, whether or not DPI exists. Perhaps the advantage for DPI within Internet providers and businesses is the detection and interception of malicious traffic and identifying the hosts with these infections. DPI can detect whether the email you just received contains a malicious component, when a trojan or worm activates and tries to propagate, where the attacks originate etc.

We need to just view DPI as another vital component of enterprise network security, which can also be misused much like most open source penetration testing tools. As the saying goes "Guns don't kill people, people kill people".

7.3.5 Privacy

In discussing DPI, one cannot but make mention of privacy. It is a storm cloud in the bright blue skies of DPI. As a result of its deep inspection nature, it raises all sorts of questions from privacy advocates concerned about the trivial collection of private personal information. Most current devices are so sophisticated that they can reconstitute e-mails and IM conversations out of asymmetric traffic flows and can essentially peek "under the hood" of any non-encrypted packet to take a look at its content as demonstrated by NetWitness Investigator. The greatest threat to privacy are the following factors².

Opportunity: Because most providers serve as the gateway between private networks and the Internet, every VoIP call, mail, IM sent and received; every web page and file you download travels through the service providers network.

Means: Few years ago, most service providers lacked the means to efficiently analyze traffic traversing their network, but over the past decade, while network bandwidth has

²<http://dpi.priv.gc.ca/index.php>

increased, computing power has also increased exponentially and ISPs can now analyze more traffic inexpensively. An entire industry – the DPI industry – has sprung up to provide technologies for massive, widespread, automated network monitoring and surveillance.

Motive: Third parties are placing pressure on service providers to spy on users Internet activities in unprecedented ways. Advertisers are willing to pay higher rates for what is known as behavioral advertising. To enable behavioral advertising, companies like Phorm have been trying to convince ISPs to collect user web-surfing data they do not collect today. Similarly, the copyright industries seem willing to pay ISPs to detect, report, and possibly block the transfer of copyrighted works.

Highlighted below are some privacy implications of DPI;

- Consent to monitoring is a waiver of privacy rights Including automated, non-human inspection
- All privileges are waived on an inspection network
- Private communications will be available to others through a 3rd party subpoena to the service provider with a showing of mere relevance, and without user notice
- ISP Terms of Service require businesses to consent to the monitoring of their online communications
- Information gleaned from inspection can be used for any and all purposes by the service provider
- Trade secrets, proprietary information, confidential communications, transaction records, customer lists, etc are all exposed
- Businesses risk violating customer privacy laws Allowing third party access to financial and browsing records is often prohibited

Since DPI has lawful and legitimate uses, it need not be banned. However, privacy can be ensured by requiring direct and voluntary permission as well as informed user consent for wiretapping. Service providers must also ensure full and complete disclosure of inspection practices and legal consequences to users. However requiring consent as a condition of receiving service be deemed involuntary. There is also a need for an administrative or legislative

declaration of a public policy against Internet access contracts that fail to disclose practices and privacy implications and/or require waiver of privacy rights as a condition of service. Finally privacy should be preserved without regulation.

In reality though, the problem isn't that of DPI technology, it is really that of its implementation. The executive vice president of Verison once said, *"The perceived problem with 'packet inspection' is not the technology. Many useful technologies can be used for nefarious purposes. The problem arises if packet inspection is used to inappropriately track customers' online activity without their knowledge and consent and invade their personal privacy."*

Ultimately, it is again a matter of regulation and social discourse to decide what levels of DPI and what applications are considered acceptable. But it is also naive to believe that intelligence services will refrain from using the latest available technology for wiretapping. This too is a matter of regulation.

7.3.6 Net Neutrality

Net neutrality is the principle that data packets on the Internet should be delivered without regard to content, source or destination. Net neutrality is sometimes referred to as the *First Amendment of the Internet*. It follows the principle that Internet users control the content viewed and applications used on the Internet. As a matter of fact, the Internet was built with this principle in mind. Indeed, it is this neutrality that has hitherto allowed a number of companies innovate and develop. Fundamentally, net neutrality is about equal and unparalleled access to the Internet. Service providers should however not be allowed to use their market power to control online activities. Note that this doesn't suggest every network has to be neutral to be useful. Discriminatory, private networks can be extremely useful for other purposes. What the principle suggests is that there is such a thing as a neutral public network, which has a particular value that depends on its neutral nature.

Beside the technical reasons, ISPs typically have other justifications for adopting DPI. By demonstrating that they can identify, sort out and filter all types of data, ISPs may be increasing their responsibility for the uses for which their customers put their networks to. For instance, until recently, providers have been able to defend themselves on accusations of facilitating various copyright infringement activities by saying that, much like the post office system, they do not know what envelopes or in this case, packets actually contain. This no longer holds true, and the situation creates new incentives for data sniffing that have nothing to do with resource allocation.

If net neutrality is to be understood in a way that guarantees equal access to all its users,

then certainly the Internet is by no means neutral. Statistics obtained across many regions gathered over a number of years agree that less than 20 percent of network users generate over 80 percent of the traffic. This phenomenon cannot only be explained by mere differences in demand. Instead, tech-savvy Internet users can get significantly more than their fair share of the available bandwidth, which is particularly critical in times of network congestion when this inequality adversely affects the performance of other users. An example is a P2P file sharing application such as BitTorrent. To maximize download speeds, they open many, sometimes hundreds of parallel connections to many different peers. Legacy client-server applications such as Web browsing only open one connection to a server, or in some cases, a few connections to a small number of servers. The multi connection applications will also win in the competition for bandwidth in a network without QoS. Worse still, they can completely displace low-data rate, realtime applications like Internet telephony and online games, effectively rendering them unusable. Hence, an unregulated network cannot be called neutral because it does not guarantee fairness among users.

Lastly, there are regulations to keep the Internet open even when the Internet thrives on lack of regulation. However, basic values have to be preserved for the simple fact that the economic system depends on the rule that money cannot be duplicated or photocopied much the same way as democracy depends on freedom of speech. Freedom of connection, with any application, to any party, is the fundamental basis of the Internet, and now, the society is based on it. You can read more about net neutrality here³

7.4 DPI Evasion Techniques

If your ISP is using DPI technologies to 'spy' on you without your knowledge (in a way that is unethical) and you are concerned about privacy issues, then this section is for you. There are ways to protect yourself and we explore one such way using secure *SSH* tunnels in the following case study. See Figure 7.4

7.4.1 Case Study 37: DPI Evasion with OpenSSH

We start by looking at using *OpenSSH* to create a secure SOCKS proxy tunnel.

³<http://dpi.priv.gc.ca>

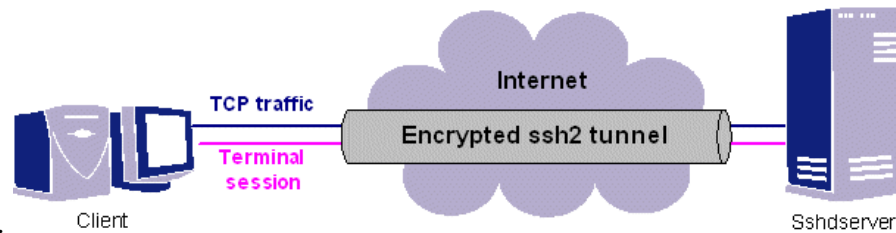


Figure 7.4:

Requirements:

- Remote shell account on a machine running an SSH server
- An SSH client - openssh
- Programs that allow you to proxy traffic with SOCKS such as a browser or mail client

One very effective way to evade deep packet inspection is to create a secure, encrypted tunnel that you send all of your traffic through. The ISPs cannot modify the content of the packets if they are passed through an encrypted channel, since they have no way of seeing what the content actually is. The idea is to encrypt the packets so as to get them out of the reach of your ISP, and once they arrive at a remote server where you have shell access, that server unwraps the traffic and sends it out on its way. The remote server that you have access must be secure and trusted. If it is not, you may inadvertently open up yourself to a man-in-the-middle attack. If you have a remote SSH connection, you can use openSSH to create a secure SOCKS proxy on a specific port of your local machine, which forwards all traffic to the remote machine before reaching its final destination.

OpenSSH Installation

Openssh server and clients should be installed by default in most Linux operating systems, but if not you can simply use the *yum* tool

```
# yum -y install openssh-clients openssh-server
```

That's all

Usage

Here we are going to establish an SSH connection from the Local Server to the Remote Server. The following is the structure and break down of the command.

```
# ssh -D localhost:port -f -C -q -N user@remote.host
```

where port is the local port you want opened.

Example

```
# ssh -D 127.0.0.1:5150 -f -C -q -N abc@myremotesshserver.com
```

Now that we have the connection established from the Local SSH Server (port 5150) to the Remote SSH Server we can now login to the Remote SSH Server using the reverse SSH Tunnel. Note that when this command is issued for the first time, it generates a hex fingerprint. Make a note of this. If at a later time you get a warning that there is a fingerprint mismatch, your traffic may be intercepted. This fingerprint acts as verification that you are indeed communicating with the intended remote server.

First let us verify that the tunnel is setup and listening on port 8080 on the Local SSH Server by running the following as root:

```
# netstat -antp | grep :5150
```

You should see an output like

```
tcp 0 127.0.0.1:5150 0.0.0.0:* LISTEN
```

To bring up a web page that is running on any Remote Network Server, just configure your Local Workstation Browser for Socks 5 Proxy and put in the IP address of the Local SSH Server (127.0.0.1) and port (5150).

Note: When configuring certain programs to use the SOCKS proxy, there is a potential for DNS leakage. This means that even though the traffic between your host and the remote server is encrypted, the name resolution may not be. This may present a problem, but certain programs such as firefox allows you to ensure that there is no DNS leakage. Some firefox plugins such as *FoxyProxy* take care of this for you in their settings.

7.4.2 Case Study 38: DPI Evasion with Putty

This time we employ the Windows based *Putty* program to create a secure SOCKS proxy tunnel

Requirement

- As above you will need a remote server you can connect to using SSH. This is typically a remote Unix or Linux server that supports SSH logins.
- The *Putty* Software which can be obtained here⁴

Once these are in place we're ready to get started.

Usage

We will set up *Putty* to create an SSH tunnel and connect to the remote server.

- In the field labeled **Host Name (or IP address)**, enter the hostname or TCP/IP address of the remote again assume that our remote ssh server is `myremotesshserver.com`
- In the field labeled **Saved Sessions**, enter a name that you want to use to identify this configuration. This is typically the hostname or IP address of your remote server, but it can also be something like "My Secure Tunnel". In this case I will just input my hostname here, that is `myremotesshserver.com`

At this point your Putty window should look like the Figure 7.5

Next, on the left side of the putty window there is a navigation tree. In that tree we simply select **Tunnels**. If this isn't already visible, you can find it by selecting **Connection** node, then **SSH**, and then **Tunnels**.

Under the section labeled **Add a new forwarded port** type in a port like 5150 for the source port. Leave the Destination field blank, then select the *Dynamic* and *Auto* radio buttons. Then click the Add button, and you should see the text *D5150* show up in the text area immediately above the **Add a new forwarded port**. as show in Figure 7.6

⁴<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

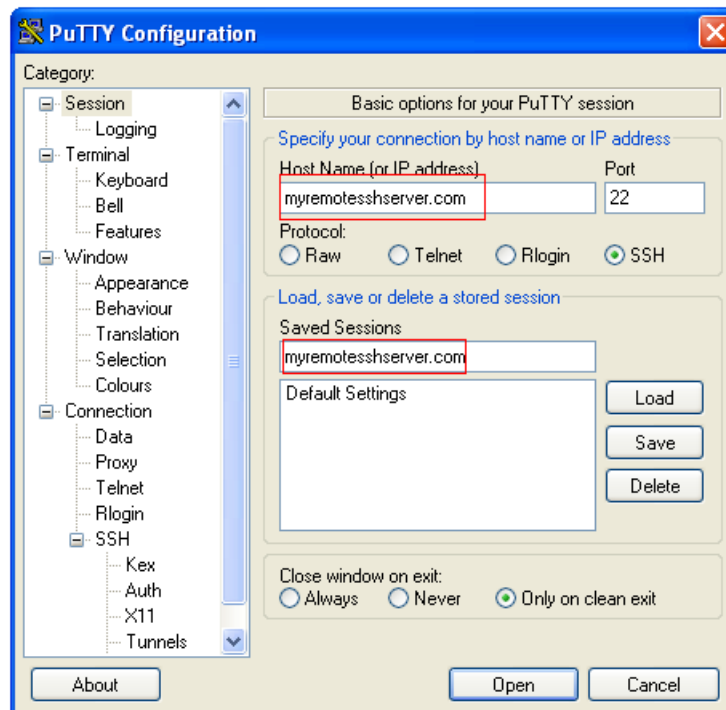


Figure 7.5:

- In the Putty navigation tree on the left click on the **Session** node (at the top of the tree), and then click the **Save** button on the right side of the screen to save this configuration. That's all to configuring *Putty*

Now all you have to do is login to your remote server.

- To do this, just click the **Open** button at the bottom of this window. You should see a Putty login shell open up to your remote server. Just login to your remote server with your *username* and *password*, and you're done.

Now we are ready to configure our browser or SOCKS enabled client by supplying 127.0.0.1 as the host and 5150 as the port. This will completely evade any DPI monitoring device.

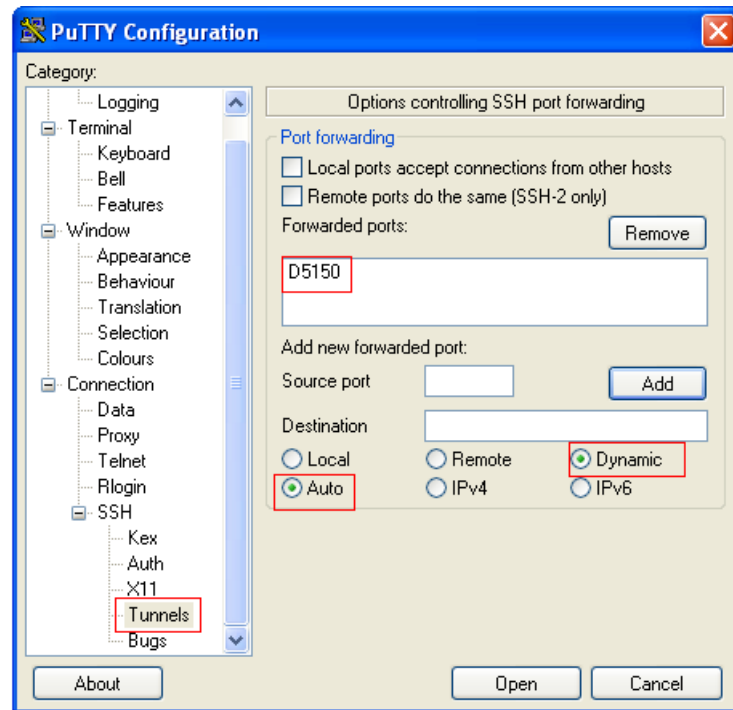


Figure 7.6:

7.5 Data Leak Prevention (DLP)

Scenario: Confidential data leaks out of the company’s network and gets into the hands of malicious users.

Data Leakage is perhaps one of the major energy sapping and exhausting challenges for most organizations. No matter how robust your technology is, or how intuitive your detection systems are, proprietary and confidential data still manages to find a way to slip through the least guarded nooks and crannies of the enterprise. The term DLP, which stands for Data Leak Prevention, had a head start in 2006 and gained some popularity in early 2007. Just as we have witnessed the growth of firewalls, intrusion detection and prevention systems (IDS/IPS) and numerous security products, DLP has already improved considerably and is beginning to influence the security industry. DLP in simple terms is truly a technology that provides visibility at content level into one’s network.

Whether it's email, instant messaging, webmail, a form on a website, or file transfer, electronic communications exiting organizations still go largely uncontrolled and unmonitored on their way to their destinations – with the ever-present potential for confidential information to fall into the wrong hands. Should sensitive information be exposed, it can wreak havoc on the organization's bottom line through fines, bad publicity, loss of strategic customers, loss of competitive intelligence and legal action.

Given today's strict regulatory and ultra-competitive environment, data loss prevention (DLP) is one of the most critical issues facing CIOs, CSOs and CISOs. For those creating and implementing a DLP strategy, the task can seem daunting. Fortunately, effective technical solutions are available. This section sets to provide some effective strategies that organizations can leverage as they seek solutions for preventing leaks, enforcing compliance, and protecting the company's brand value and reputation.

7.5.1 DLP Operating Levels

DLP is in fact not new as there are commonly three different levels of operation known: *data at rest*, *data in-motion* and *data in use*.

Data at rest - This is the content discovery level that combs an organization's hosts for sensitive data. This level usually also includes a prevention component. This applies to anything that holds data such as file shares, databases, etc. Data discovery has primary two uses. Primary use of this feature is for discovering sensitive data on data depositories. This uses the existing policy to look for any sensitive data. Discovery scanning can also be used to fingerprint data to be used identifying unstructured data elsewhere. See Figure 7.7

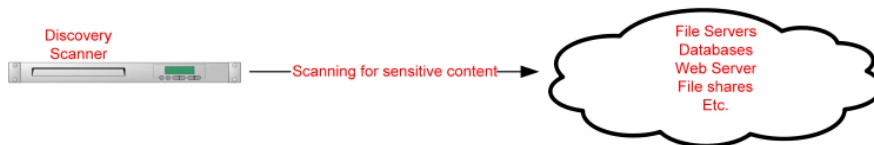


Figure 7.7:

Data in motion - This corresponds to a combination of monitoring and prevention component. It is used to monitor and control all data on the wire as well as outgoing traffic as shown in Figure 7.8

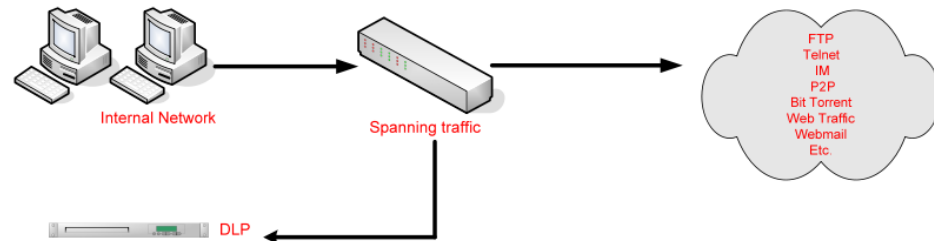


Figure 7.8:

Data in use - This level consists of agents that run on end-servers and end-user's laptops or desktops, keeping watch on all activities related to data. They typically monitor and prevent activity on file systems and removable media options like USB thumb drives. Due to its agent based approach, it really has not been a favorable solution among organizations. However, it does provide a great deal of protection against data leaving via removable devices. Implementation of this solution is comparable to a host-based IDS (Intrusion Detection System).

These individual solutions can be combined to create a much more effective DLP setup. For instance, data at rest could be utilized in identifying sensitive information, fingerprinting and deploying those fingerprints with data in motion and data in use products for an all purpose DLP solution

DLP is in effect a solution for identifying, monitoring and protecting sensitive data or information in an organization according to policies. Organizations can have different policies, but typically most tend to focus on preventing sensitive data from leaking out of the organization and identifying people or places that should be restricted to certain data or information. DLP is also known in some quarters as: data loss prevention, content monitoring and filtering (CMF), extrusion detection, outbound content management, insider thread protection, information leak prevention (ILP) and so on.

7.5.2 Data Leakage Problem

A number of organizations have experienced some loss of data and with the exponential use of email and messaging systems with data being increasingly mobile this is an ever increasing problem. The challenge is not necessarily how often data is lost, but the potential damage of every single event when there is leakage. The disturbing trend is that most incidents are not

from the outside but from internal employee who knowingly or accidentally let confidential information slip out of the organization. Unfortunately, there isn't any local statistics to work with but globally, analyst firms report that the majority of all leaks are the result of unintentional information loss from employees and partners, both external and internal leaks. Organizations need to be more proactive than ever to monitor and control employees' information activities.

Over the years, organizations have spent a tremendous amount of resources in the hope of protecting their information. However, all efforts have been directly focused on preventing outsiders from breaking into the organization, educating employees, and securing data at rest. As organizations invest millions in business systems increasing the availability of information to build or maintain a competitive edge, there remain a slew of security-related considerations, including:

- Where the organization's confidential and sensitive data reside.
- How, where and when the data is transmitted and by whom.
- How data can be controlled and protected.
- The organization's financial risk from a leak.

7.5.3 Business Case for DLP

How you may ask, is DLP different from any other security solution out there? Whilst security devices such as firewalls and IDS/IPS look for anything that can pose a threat to an organization, DLP is interested in identifying sensitive data. It looks for critical content particular to an organization. It would seem as though DLP is a solution whose only purpose is to prevent data breaches from intruders. While it can prevent such data leaks, more often than not this solution is used as a mechanism for discovering broken processes in the normal course of business. For instance the presence of sensitive data on an employees laptop. As organizations spend a lot of valuable time, energy and money on user awareness training, one would assume data leaks as a result of actions by an unwitting user would be very minimal. However, this is not the case. We know for a fact that majority of all malware outbreaks companies suffer are due to such user actions. This trend has not changed much even with the ongoing user awareness training.

Up until recently, most organizations thought of data security only in terms of protecting their network from intruders. But with growing amount of data, rapid growth in the sizes of

organizations, rise in number of data points and easier modes of communication, intentional or accidental data leakage from the organization has become a rather sad reality. This has led to a growing awareness about information security in general and about outbound content management in particular.

Some of the following are the major reasons behind the adoption of DLP solution by most organizations:

- Growing cases of data and IP leakages for example financial data being leaked inadvertently or being hacked
- Regulatory mandates such as SOX, PCI DSS, ISO/IEC 27001 to protect private and personal information
- Protection of brand value and reputation
- Internal policies for example, Facebook leaking some pieces of their code
- Profiling for weaknesses for instance finding out employees level of access to data, discovering sensitive data lying on public servers and finding out if employees are doing what they are not supposed to do with data

7.5.4 DLP Workings

DLP solutions classify data in motion, at rest, and in use, and then dynamically apply the desired type and level of control, including the ability to perform mandatory access control that can't be circumvented by the user. It monitors multiple transmission paths and routes for specific outbound and inbound content. DLP solutions typically:

- Perform content-aware DPI on outbound network communication including email, Web, VoIP and other TCP/IP protocols
- Track complete sessions for analysis, not individual packets, with full understanding of application semantics
- Detect (or filter) content that is based on policy-based rules
- Use linguistic analysis techniques beyond simple keyword matching for monitoring (e.g. advanced regular expressions, partial document matching, Bayesian analysis and machine learning)

7.5.5 Overview of DLP Components

Broadly, the core DLP process has four components: *identification, monitoring, prevention* and *control*.

Identification: It is a process of discovering the constituents of sensitive content within an organization. An organization has to first and foremost define what “sensitive” data is. This can be done by using policies, which are composed of rules, which in turn could be composed of words or patterns or any other variation. These rules are then fed to a content discovery engine that scours data sources in the organization for sensitive content. Data sources may include application data like Web servers, mail servers, portals and database servers, repositories like network attached storage NAS and SANs, as well as end-user data sources like laptops, desktops and removable media. There could be different policies for different classes of data sources; for example, the policies for the portal could try to identify design documents whereas those for database could be tuned to discover financial information. Most DLP products have built-in policies for well-known scenarios, such as PCI compliance.

Monitoring: It is a solution typically deployed at the network egress point or on end-user endpoints. It is used to flag data that should not be traversing the boundaries of the organization. This flagging is done using rules and policies, which could be written independently, or be derived from information gleaned during the identification process. The monitoring component taps into raw data going over the wire, does some semantic reconstruction and applies policies on it. DPI can be utilized here for session reconstruction of raw network data.

Prevention: It is the process of taking some action on the data flagged by the identification or monitoring component. A number of actions on the data are possible here – blocking, quarantining, deleting, encrypting, compressing and notifying. Preventive actions are also typically configured using policies and hooks into identification and/or monitoring policies. This component is mostly deployed side by side with the identification or monitoring component.

Control: This is component is just a feature that allows a user to centrally manage and monitor the whole DLP process. This typically includes the GUI, policy/rule definition and deployment module, process control, reporting and various other dashboards.

7.5.6 DLP Phases

The phases of DLP are typically discovery, monitoring and prevention phases. We take an in-depth look at these phases in this section.

Discovery component makes use of crawlers to find sensitive content in an organization's network of machines. Each crawler is made up of a connector, browser, filtering module and reader. A connector is a specific module that assists in the connection to, and browsing and reading from a data source. So, there are connectors for various types of data sources like CIFS, NFS, HTTP, FTP, databases and so on. The browser module lists all data that is accessible within a data source. It is then filtered depending on the nature of discovery requirements. For instance, if the requirement is to discover and analyze only word (.doc) and pdf formatted files, then all other types of files will be filtered out of the listing. There are different dimensions, depending on metadata, on which filtering can be done: it can be name, size, folder, content type, sender, author, date, subject etc. Once the filtered list is ready, the reader module carries out the function of actually downloading the data and any related metadata information.

Monitoring component is typically composed of following modules: data tap, reassembly, protocol analysis, content analysis, indexing engine, rule engine and incident management. Further analysis can be done on the captured data. As already alluded to earlier, this capture can take place at any protocol level. After data is captured from the wire, it is processed into a form that is suitable for further analysis. For example, captured TCP packets could be reassembled into a higher level protocol like HTTP and further into application level data like GMail. After data is parsed for analysis, the first level of policy/rule evaluation is done using protocol analysis. Here, the data is parsed for protocol specific fields like IP addresses, ports, possible geographic locations of IPs, *To*, *From*, *Cc*, FTP commands, GMail XML tags and so on. Rules that depend on any of the protocol-level information are evaluated at this stage. An example is – outbound FTP to any IP address in China. If there is a match, it is then stored in a database with all relevant information. The next step, content analysis, is more involved: first, actual data and meta-data is extracted out of assembled packet, and then the content type of the data (e.g. PPT, PDF, ZIP, C source, Python source) is determined using signatures and rule-based classification techniques (a similar but less powerful option is *file* command in Unix). Depending on the content type of data, text is extracted along with as much meta-data as possible. Now, content based rules are applied – for example, disallow all Java script code. Again, matches are stored. Depending on the rules, more involved

analysis like classification, entity recognition, tagging and clustering can also be done. The text and meta-data extracted are passed to the indexing engine where it is further indexed and made searchable. Another set of rules, which depend on contents of data, are evaluated at this point; an example: stop all MS Office or PDF files containing the words “confidential” and “top secret” with a frequency of at least one per page. The indexing engine typically makes use of an inverted index, but there are other ways also. This index can also be used later to perform ad-hoc searches e.g. for deeper analysis of a policy match. All along this whole process, the rule engine keeps evaluating many rules against many pieces of data and keeping a track of all the matches. The matches are collated into what are called incidents that is actionable events – from an organization’s perspective with as much detail as possible. These incidents are then notified or shown to the user and/or also sent to the prevention module for further action

Prevention module contains a rule engine, an action module and (possibly) connectors. The rule engine evaluates incoming incidents to determine action(s) that needs to be taken. Then the action module kicks in and does the appropriate thing, like blocking the data, encrypting it and sending it on, quarantining it and so on. In some scenarios, the action module may require help from connectors for taking the action. For example, for quarantining, a NAS connector may be used or for putting legal hold, a CAS system may be deployed. Prevention during content discovery also needs connectors to take actions on data sources like Exchange, databases and file systems. As an example SMTP blocking capability is enabled by integrating into MTAs. All email messages are inspected by an SMTP prevent server and once the content is examined, an action item is sent to the MTA. An action item can be block, encrypt, quarantine and/or notify sender.

HTTP and HTTPS blocking are enabled by integrating with HTTP proxy server and HTTPS proxy server respectively. This is very similar to applications like WebSense except HTTP prevent servers allow or block request based on content. This feature prevents sensitive information leaving a company via web mail, any external blogging sites, news groups, etc. as shown in Figure 7.9

FTP prevention server functions the same way as well. It integrates with FTP proxy servers to enable blocking

7.5.7 Extending DLP Solution

There are many “value-added” options that can be performed on top of the functionality described above. These are sometimes provided as separate features or products altogether.

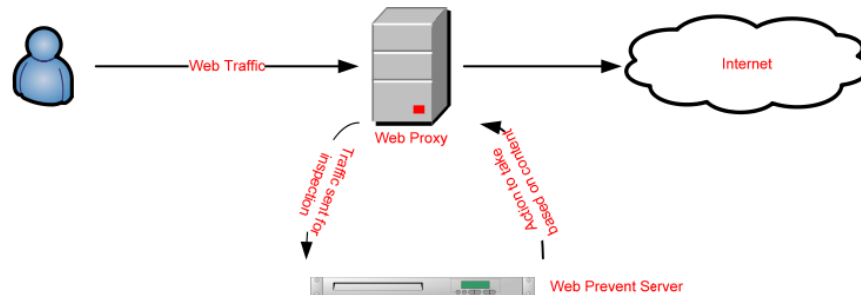


Figure 7.9:

Reporting and OLAP - Information from matches and incidents is fed into cubes and data warehouses so that OLAP and advanced reporting can be carried out.

Data Mining - Incident/match information or even stored captured data is mined to discover patterns and trends, plot graphs and generate fancier reports. The possibilities here are endless and this seems to be the hottest field of research in DLP right now.

E-discovery - Here, factors important from an e-discovery perspective are extracted from the incident database or captured data and then pushed into e-discovery products or services for processing, review or production purposes. This process may also involve a bit of data mining.

Learning - Incident and mined information are then used to provide a feedback into the DLP setup. Eventually, this can improve existing policies and even provide new policy options.

Integration - Integration with third-parties. For example, integration with some applications provides setups that can capture and analyze HTTPS/SSL traffic.⁵

7.5.8 DLP Use Case Summary

Below is a diagrammatic representation of use cases for Data Leakage Prevention in relation to data at rest, data in motion and data in use

⁵<http://punetech.com/data-leakage-prevention-overview/>

Data at rest.

Figure 7.10

1. Compliance (GLBA, PCI, HIPAA) CCN data stored unencrypted on disks	2. Investigate Users Search all hard disk contents stored on an employee's system who is caught surfing adult sites	3. eDiscovery Discover and index content stored on systems or repositories such as Sharepoint
4. Data Classification Determine where sensitive data exists and the type of data it is	5. Data Access Audit Search for payroll or HR data on sales computers	6. Laptop and Back-up Tape Loss Manifest of contents stored on a particular system

Figure 7.10:

Data in motion

Depicted in Figure 7.11

Data in use

shown in Figure 7.12

7.6 Encryption

Encryption is a general method of scrambling data, such as passwords, in order to protect it in the event of interception. Depending on the encryption method used, it could conceivably take a cracker several thousand years to decrypt the data using conventional methods. Most encryption methods are done between the client application and the server, making the process transparent to all users. However, encryption is something that most people do not understand. Administrators feel that it is a nuisance to integrate into their network services,

<p>1. Control of Rogue Business Processes</p> <p>Monitor unauthorized leaks over instant messaging or P2P transfers</p>	<p>2. Regulatory Compliance</p> <p>Patient data sent to personal webmail accounts by medical employees for working at home</p>	<p>3. Encryption</p> <p>Automatically encrypt sensitive data destined for a business partner or a client</p>
<p>4. Conditional Blocking or Quarantine</p> <p>Hold & review emails sent to competitors that contain company financials or Intellectual Property</p>	<p>5. Monitor and/or Block SSL Channels</p> <p>Visibility into SSL-encrypted web mail transmissions or PGP emails</p>	<p>6. Employee Education</p> <p>Auto-Notify employees and/or management when privacy or harassment training guidelines are violated</p>
<p>7. Investigate Unknown Leaks</p> <p>Investigate all communications of an employee leaking trade secrets to competition</p>	<p>8. Acceptable Use</p> <p>Determine if questionable images or materials are being sent</p>	

Figure 7.11:

<p>1. Data Protection While disconnected</p> <p>Mobile employee sending out next generation product plans while sitting in Starbucks</p>	<p>2. Confidential Data Abuse /Theft</p> <p>Protect information leaving through USB or Wifi etc.</p>	<p>3. Employee Education</p> <p>Inform employees in realtime about policy violations as they occur on their systems or ask for justification</p>
---	---	---

Figure 7.12:

even though, in most cases, encrypting network traffic can be a relatively simple procedure. The advantages of using encryption vastly outweigh its liabilities.

In information security, encryption is primarily used to protect data in one of two instances. The first is to protect *data at rest*. The second is to protect *data in motion*. Different encryption approaches are needed to protect these two different states of data, even if it is the same information. Whatever the encryption application used, it should minimally be compatible with all major financial and insurance institutions and should be certified to provide a level

of encryption that is compliant with stringent open systems standards. SSL, SSH and HTTPS encryption technologies should also be used to protect data in motion.

7.6.1 Data Types

The following are some common types of data that can be encrypted. The first list is various types of data at rest, while the second is of data in motion.

Data at rest

- Individual files and folders Many free and commercial products are designed explicitly for the purpose on encrypting specific files and folders of data as determined by the user.
- Partitions or Volumes Some products are designed to encrypt all of the data on an entire partition or volume requiring little interaction from the user.
- Entire Physical Disks Some manufacturers are producing hard drives with encryption built in, to allow for the encryption of all data on the disk.

Data in motion

- Secure shell (ssh) SSH encrypts all data between two ssh enabled computers.
- Web traffic (https) Most web browsers support the exchange of sensitive data between the client and host when necessary using HTTPS for communication.
- Instant Messaging Several instant messaging clients support plugins that allow for the encryption of all messages sent between parties via instant messaging.
- Email Some email clients support plugins that allow for the transmission of encrypted email messages sent between parties via SMTP

7.6.2 Encryption Tools

Here are some common tools that can be used to encrypt data. The tools I highlight here are free. Also, listed with each tool are the types of data the tool is designed to encrypt, as well as information about whether the tool is designed to protect data in motion or data at rest.

Encryption Tools for Data at rest

By far the best free tool that i have used for this is **TrueCrypt**. It is a free and open source software used for encryption of volumes or folders. Great for individuals, small work groups, and portable media.

Encryption Tools for Data in motion

- **GNU Privacy Guard (GnuPG)** Free software is a complete cryptographic system usable for a variety of encryption uses.
- **EnigMail** is a plugin for Mozilla email clients (including Thunderbird). It uses GnuPG for sending and receiving encrypted and digitally signed email.

Strong encryption provides a powerful mechanism that can be applied to many parts of an organization's data security practices, offering effective, continuous protection of data. This protection can encompass a range of uses, from end-point devices in the field to the core of the central servers where vital information resides.

7.7 Data protection

With the explosion of information in computer systems, data protection has become an important issue for most organizations. Data at rest is subject to threats from hackers and other malicious threats. To prevent this data from being accessed, modified or stolen, organizations often employ security protection measures such as password protection, data encryption, or a combination of both. The security options used for this type of data are commonly referred to as data at rest protection. Protection is basically a step-by-step process and often times an effective way out of this insecurity of data loss is to employ layered data protection.

7.7.1 Continuous Data Protection

Most people use the Internet through their computer. The use of the information superhighway is generally considered a reason for leakage of data due to disclosure of identity. It hardly matters what the size of your business is , loss of data is a serious issue that needs addressing. The huge amount of data being moved may carry important information from the perspective

of the business. To get hold of this vast data without any data corruption, losses and leakages is very difficult. But with the use of some data protection measures such as back up software, there is still a silver lining of hope. These are certain data protection measures that greatly assist in safeguarding the data from getting lost. Of these, CDP is definitely the most effective solution.

Continuous Data Protection (CDP) CDP is a back up procedure that takes a backup of your files by automatically saving a copy of the data and using this copy of data at the time of data loss. This means that if a virus attacks or corrupts data at any point of time, a copy can immediately be received through the restoration of that data again. The procedure is continuous in a way that it initially saves every copy of the changes made by the user in a file to a different storage space so that if the data is lost at any point, it can be recovered. CDP is a service that captures changes to data to a separate storage location. There are multiple methods for capturing the continuous changes involving different technologies that serve different needs. CDP-based solutions can provide fine granularities of restorable objects ranging from crash-consistent images to logical objects such as files, mail boxes, messages, and database files and logs.

Continuous data protection differentiates itself from traditional backup in that there is no need to specify the point in time in which you would like to recover until you are ready to perform a restore. Traditional backups can only restore data to the point at which the backup was taken. With continuous data protection, there are no backup schedules. When data is written to disk, it is also asynchronously written to a second location, usually another computer over the network. This introduces some overhead to disk-write operations but eliminates the need for scheduled backups.

CDP also differs from RAID, replication, or mirroring in that these technologies only protect one—the most recent—copy of the data. If data becomes corrupted in a way that is not immediately detected, these technologies will simply protect the corrupted data. Continuous data protection will protect against some effects of data corruption by allowing to restore a previous, uncorrupted version of the data. Transactions that took place between the corrupting event and the restoration will be lost, however. They could be recovered through other means, such as journaling.

There is a proliferation of CDP services now and could be seen on any dedicated server or web hosting plans. With such remarkable benefits, this is the best way to safeguard data effectively.

7.7.1.1 CDP vs Near Continuous

There is a big debate between the advantages of "True" CDP against Near-Continuous. Near Continuous Data Protection is a derivative of Continuous Data Protection, where protection is done in such a manner that many "point in time" data recovery options are available. These recovery points should be close enough in time (e.g., a few minutes apart) to minimize data loss in case of disaster.

True CDP is a variation of replication technology and to be a True Continuous Backup, an application must provide at least one second granularity between recovery points. This is why this True CDP Technology is only available to Storage Area Networks and also has limited application particularly to streaming type applications.

The file system maintains a cache of changes made to files in memory. This is commonly called the System Cache in Windows and buffers/cache in Linux. More accurately this is a Page Cache. This in-memory representation of files that is found on all modern operating systems like Windows and Linux is only flushed to disk periodically and this technique is called lazy write.

The important thing to note is that a critical part of server performance is that writes to disk don't actually get written to the disk platters except periodically and this period is determined by the operating system. True CDP applications do not see changes written to the Storage Area Network until the operating system flushes its Disk Cache. This means the view of the raw SAN disk is in the average case corrupt and out of date compared to the actual state of the system. Near-Continuous backup applications in contrast to True CDP have user scheduled synchronizations. Realistically these can only be performed as frequently as every 15 minutes as that is about as often as you can safely flush the disk cache without losing any performance.

7.7.2 Data Protection Act

The Data Protection Act came into existence with an objective *to protect data from being processed and misused by anyone*. The act was started in 1984. At the time, it was implemented to protect personal data from being processed automatically. Then, it was followed by the Data Protection Act 1998 which remitted the Data Protection Registrar position and renamed it to Data Protection Commissioner. Freedom of Information Act 2000 further expanded the role of the commissioner to Office of the Information Commissioner.

Data refers to information that needs to be processed by the computer as a word document

or a spreadsheet and is stored on the computer for future use or a source of information in the form of a record. Personal data is discriminated from personal sensitive data in the Data Protection Act and the act only applied to the personal information.

The act covers any data that is relevant to the identification of a living person. For example, name, address, phone numbers, birth days, Fax numbers, E-mail addresses etc. The act only applies to the data which is held or is subjected to be held on the computer. It is a right for people whose information is stored on the computer, for those who store this information and for those who collect such personal information.

The act lays down eight principles of handling the information related to personal data:

- The personal data can only be used for the purpose for which it is collected.
- The other parties cannot access the data without the consent of the individual, whose data is in use.
- An individual has the right to access the information that is processed about him.
- As soon as the information is processed, it becomes no longer in use.
- As the act is under the United Kingdom, the information can not pass outside the European Economic Area (EEA).
- All entities that process personal information must register themselves with the Information Commissioner.
- Entities processing information must have substantial security measures.

However, Data Protection Act is based on certain data protection principles that are provided below:

- Personal data cannot be processed until it meets at least one condition from schedule 2 and schedule 3 of the Act.
- Personal data shall be obtained only for one or two purposes lawfully and then it cannot be used further.
- Personal data has to be accurate and must be kept up to date.
- Appropriate measures should be taken to avoid unauthorized and unlawful use of the data.

- Perfect data cannot be transferred across the country.
- Full attention has to be paid towards the security towards data corruptions and losses.

The Data Protection Act encompasses the convenience of the person whose information is being processed and for all those who process and collect data.

7.7.3 Formulating a Data Protection Policy

There is no particular data protection policy that can be used as a standard for all the organizations as it became evident with the Data Protection Act in the year 1998.

When you are protecting your data, you are not required to follow some fixed rules that other organizations follow. The importance lies in being aware of common principles that are apt for your situation.

7.7.3.1 Organizational Data Protection Policy

Depending upon situation to situation, actions are taken. These actions depend a lot on the working methodology followed by your organization and the type of clients you work with. Depending on these factors, you should decide your own policy. You should not just go by the policy of other company as their working methodology may be different and they may be working with different sets of clients. Applying the same policy to your own profession may prove to be detrimental to your business interests.

While deciding a data protection policy for your own company, you should not just consider data protection. There are other aspects that you should consider. Consider whether the decision taken by your organization regarding its legal issues is correct; what are the responsibilities of each and every individual in the organization and whether they are aware of the importance of protecting data from unethical disclosures or not.

7.7.3.2 Data protection policy and People

A data protection policy is not just about protecting the data. By protecting valuable data, people are protected in some way or the other by preventing any data leakage. This leakage of data is harmful to people as the data once leaked may go to the wrong people. This can be misused because of an inefficient security system. Suppose a data is very crucial and some

important decisions are to be made based on the same data. Unfortunately you provided the wrong data, imagine the consequences. If you are not very clear about data protection, this may create some unwanted problems and it is your professional competency that becomes questionable in the organization.

It is therefore important to ensure data is secured. Policies that relate to data updating and revisions should be carried out regularly. The importance of confidentiality should be made clear to the others in the organization. You should always have a discussion within teams or other staff members or some groups may also be formed to keep track of data protection processes. Once the data protection policy is made, it should be circulated among working groups to take their feedback.

7.8 Summary

We investigated the discipline that embodies a convergence of governance and data management surrounding the handling of security data in an organization. We also an in-depth analysis of data governance concepts such as deep packet inspection, privacy, data loss prevention, encryption and data protection.

Appendix A: Glossary

A

Abstract data Refers to (heterogeneous) data that has no inherent spatial structure; thus it does not allow for a straightforward mapping to any geometry, but relies upon means provided by information visualization for its visual representation.

Accuracy The measure of a predictive model that reflects the proportionate number of times that the model is correct when applied to data.

Animation is the outcome of the process of taking a series of individual static images and stringing them together in a timed sequence to give the impression of continuous movement.

Application Programming Interface (API). The formally defined programming language interface between a program (system control program, licensed program) and its user.

Appropriateness refers to the trade off between efforts required for creating the visual representation and the benefits yielded by it. If this trade off is balanced, the visualization is considered to be appropriate.

Artificial Intelligence The scientific field concerned with the creation of intelligent behavior in a machine.

Association Rule. A rule in the form of “if this then that” that associates events in a database.

B

Bar Chart A graphical depiction of the frequency distribution, relative frequency distribution, or percent frequency distribution of a qualitative variable or dataset.

Binning The process of breaking up continuous values into bins. Usually done as a preprocessing step for some data mining algorithms. For example breaking up age into bins for every ten years.

Brushing is the process of interactively selecting data items from a visual representation. The original intention of brushing is to highlight brushed data items in different views of a visualization.

Brute Force Algorithm A computer technique that exhaustively uses the repetition of very simple steps repeated in order to find an optimal solution. They stand in contrast to complex techniques that are less wasteful in moving toward an optimal solution but are harder to construct and are more computationally expensive to execute.

C

CART Classification and Regression Trees. A type of decision tree algorithm that automates the pruning process through cross validation and other techniques.

Causal Relationship A relationship that holds true on the average due to the influences of uncertain factors besides those of the Independent Variables that are explicitly identifiable.

CHAID Chi-Square Automatic Interaction Detector. A decision tree that uses contingency tables and the chi-square test to create the tree. Classification. The process of learning to distinguish and discriminate between different input patterns using a supervised training algorithm.

Chartjunk is a term for unnecessary or confusing visual elements in charts and graphs. Markings and visual elements can be called chartjunk if they are not part of the minimum set of visuals necessary to communicate the information understandably.

Classification The process of determining that a record belongs to a group.

Clustering is the process of grouping similar input patterns together using an unsupervised training algorithm. unsupervised learning technique.

Cognition The intellectual process that produces knowledge from perception or ideas. It describes the conscious operations of the mind by which one becomes aware of thoughts and perceptions, including all aspects of perceiving, thinking, and remembering.

Collinearity The property of two predictors showing significant correlation without a causal relationship between them.

Colour is the perceptual result of light in the visible region of the spectrum, having wavelengths in the region of 400 nm to 700 nm, incident upon the retina.

Conditional Probability The probability of an event happening given that some event has already occurred. For example the chance of a person committing fraud is much greater given that the person had previously committed fraud.

Confirmatory analysis, one or more hypotheses about the data serve as a starting point. The process can be described as a goal-oriented examination of these hypotheses. As a result, visualization either confirms these hypotheses or rejects them.

Covariance A numerical measure of linear association between two variables. Positive values indicate a positive relationship, and negative values indicate a negative relationship.

Coverage A number that represents either the number of times that a rule can be applied or the percentage of times that it can be applied.

Cross Validation (and Test Set Validation). The process of holding aside some training data which is not used to build a predictive model and to later use that data to estimate the accuracy of the model on unseen data simulating the real world deployment of the model.

Cumulative Frequency Distribution A tabular summary of a set of quantitative data showing the number of items/cases having values less than or equal to the upper class limit of each class interval. The cumulative relative frequency distribution shows the fraction or proportion of the items having values less than or equal to the upper class limit of each class; while the cumulative percent frequency distribution shows the percentage of items/cases having values less than or equal to the upper class limit of each class.

D

Data Measurements or facts that are collected from a statistical unit/entity of interest. They are classified as quantitative (continuous and discrete) if they contain numeric information.

Data Type Data type of a parameter is given by the statistic attributes of the values.

Data Visualization is the process of using graphical presentation to represent complex data in a way that provides the viewer with a qualitative understanding of its information contents, turning complicated sets of data into visual insights.

Descriptive Statistics A branch of statistics that is concerned with the use of tabular, graphical, and numerical methods to summarize data.

Details on demand technique allows interactively selecting parts of data to be visualized more detailed while providing an overview of the whole informational concept.

Deterministic Relationship A relationship that holds true in a mathematical sense according to some preconceived rule or formula. For example, $A = WL$ describes the relationship between the Area (A), the Width (W) and the Length (L) of a rectangle.

Dimension Each of the independent variables that contain the data we intend to visualize.

Direct manipulation — proposed among others by Ben Shneiderman — allows a viewer to interact with a visualization corresponding to the real-world analogue it follows. It avoids the barrier of having to translate ideas into commands meaningful to a computer by building a graphical user interface that is semantically in line with the representation.

Dynamic queries continuously update the data that is filtered from the database and visualized. They work instantly within a few milliseconds as users adjust sliders or select buttons to form simple queries or to find patterns or exceptions; the dynamic-query approach thus applies the principles of direct manipulation to the database.

E

Effectiveness A visualization is effective if it addresses the capabilities of the human visual system. Since perception, and hence the mental image of a visual representation, varies

among users, effectiveness is user-dependent. Nonetheless, some general rules for effective visualization have been established in the visualization community.

Exploration denotes an undirected search for interesting features in a data set.

Exploratory data analysis (EDA) was introduced by John Tukey as an approach to analyze data when there is only a low level of knowledge about its cause system as well as contextual information. EDA aims at letting the data itself influence the process of suggesting hypotheses instead of only using it to evaluate given (a priori) hypotheses. Explorative - opposed to Confirmatory - Data Analysis is like detective work looking for patterns, anomalies or in general new insights and is usually done via graphical representations of the underlying dataset.

Expressiveness criteria identify graphical languages that express the desired information. [...] A set of facts is expressible in a language if it contains a sentence that (1) encodes all the facts in the set, (2) encodes only the facts in the set.

F

Fact A verified data or sample evidence used along with probability theory to support hypothesis testing procedures.

Filtering is one of the basic interaction techniques often used in information visualization used to limit the amount of displayed information through filter criteria.

Fish-eye view magnify the center of the field of view, with a continuous fall-off in magnification toward the edges. Degree-of-interest values determine the level of detail to be displayed for each item and are assigned through user interaction.

Focus-plus-context-visualizations The basic idea of this is to enable viewers to see the object of primary interest presented in full detail while at the same time getting a overview-impression of all the surrounding information — or context — available.

Frequency Distribution A table that shows the number of cases/items that fall in each of several non-overlapping classes of the data. The numbers in each class are referred to as frequencies. When the number of cases/items are expressed by their proportion in each class, the table is referred to as the a relative frequency distribution or a percentage distribution.

G

Glyphs are basically composite graphical objects where different geometric and visual attributes are used to encode multidimensional data structures in combination. A graphical object designed to convey multiple data values.

Graphic Design is the applied art of arranging image and text to communicate a message. It may be applied in any media such as print, digital media, motion picture, animation, product decoration, packaging, signs, identities, etc.

Grouped Data Data that has been organized into a frequency distribution. Thus, for a variable X the individual values (X_i) in the original data set are unobservable.

H

Histogram A graphical depiction of the frequency distribution, relative frequency distribution, or percent frequency distribution of a quantitative variable or data.

Human-Computer Interaction. The study of how people work with computers and how computers can be designed to help people effectively use them.

Heterogeneous data refers to n -dimensional data acquired from different sources that includes a mixture of different data types such as temporal, spatial, ordinal or nominal.

Hypothesis consists either of a suggested explanation for an observable phenomenon or of a reasoned proposal predicting a possible causal correlation among multiple phenomena. The scientific method requires that one can test a scientific hypothesis. A hypothesis is never to be stated as a question, but always as a statement with an explanation following it.

I

Icon In computer terminology (as opposed to graphic design terminology), an icon is a small image used most often to represent files or label a button.

Information is data put within context. It's a concept bound to that of metadata, data that refers to the meaning of other data.

Infographics are traditionally viewed as visual elements such as charts, maps, or diagrams that aid comprehension of a given text-based content.

Information visualization (InfoVis) produces (interactive) visual representations of abstract data to reinforce human cognition; thus enabling the viewer to gain knowledge about the internal structure of the data and causal relationships in it.

Insight We define insight as an individual observation about the data by the participant, a unit of discovery.

Interactivity Interactivity means controlling the parameters in the visualization reference model. This naturally means that there are different types of interactivity, because the user could control the parameters to data transformations, to visual mappings, or to view transformations. It also means that there are different forms of interactivity based on the response cycle of the interaction.

K

Knowledge Information given meaning and integrated with other contents of understanding. What differentiates knowledge from Information is the complexity of the experiences that you need to reach it. Knowledge is not transferable, you have to build it yourself by experiencing the information.

Knowledge Crystallization The goal of a knowledge crystallization–process is to get the most compact description possible for a set of data relative to some task without removing information critical to its execution. Knowledge that is proven effective, useful, and objective is maintained — knowledge irrelevant in this case is removed.

Knowledge visualization means the use of visual representations to transfer knowledge between at least two persons. It aims to improve the transfer of knowledge by using computer and non-computer based visualization methods complementary.

L

Linking and brushing are interaction techniques. They can be used to enhance the work with scatterplot matrices, parallel coordinates and many other InfoVis techniques.

M

Mental Model We are explanatory creatures: We develop explanations for the events of the world and for the actions of people, both ourselves and others. We find reasons, causes, explanations. All of these require us to create "mental models," mental scenarios in which we construct representative explanatory descriptions. Mental models allow us to understand prior experiences, the better to predict future ones. Mental models also give us guidance and assistance in knowing what to expect and how to respond in novel or dangerous situations.

Metadata Data about data. In data processing, meta-data is definitional data that provides information about or documentation of other data managed within an application or environment.

Model A model in science is a physical, mathematical, or logical representation of a system of entities, phenomena, or processes. Basically a model is a simplified abstract view of the complex reality. Models are meant to augment and support humans reasoning, and further can be simulated, visualized and manipulated.

Multiple Views Multiple views of the same data set (3 views: Parallel Coordinates?, a 2D Graph and a Treemap). A multiple view-system uses two or more distinct views to support the investigation of a single conceptual entity.

N

Navigation is directed movement in discrete or continuous information space.

Normal Probability Distribution A probability distribution of a continuous random variable.

O

Objective A statement of purpose.

Outlier(s) One or more data values that depart significantly from the rest of the values either by being too big [maximum value outlier(s)] or too small [minimum value outlier(s)].

Outliers can cause trouble with statistical analysis, so they should be identified and acted on prior to analysis.

Overview provides a general context for understanding the data set; it paints a "picture" of the whole data entity that the information visualization represents.

P

Pan Panning is an interaction technique that helps to navigate within a view. Instead of using a bounded view, an user interface that supports panning typically has no bounds. That means, it is possible to navigate in any direction without limitations given through bounds.

Pattern is an expression in some language describing a subset of the data or a model applicable to the subset.

Perception Representing the basic component in the mechanism of forming new concepts, perception is the process of becoming aware of something by use of the senses.

Perceptual principle Perceptual and spatial representations are more natural and therefore to be preferred over non perceptual, non spatial representations, but only if the mapping between the representation and what it stands for is neutral - analogous to the real perceptual and spatial environment.

Pie Chart A graphical device for presenting qualitative data where the area of the whole pie represents 100% of the data being studied and the slices (or subdivisions of the circle) correspond to the relative frequency for each class (or subdivision or sector).

Pixel The smallest element of a video screen that can be assigned a determined colour, brightness and luminance.

Population The set of all elements in the universe of interest to the researcher. A frame comprises the elementary units with the appropriate restrictions imposed on the target population.

Parameter A summary measure whose value is contained/embedded in a population of data. In most instances this value is unknown; hence must be estimated from that of the corresponding sample statistic.

Probability distribution A table, graph, or mathematical function that describes how the probabilities are distributed over the values that the random variable of interest (X) can assume.

Q

Qualitative Data Data that provide or contain non-numeric information; they serve merely as labels or names for identifying special attributes of the statistical entity/unit of interest.

Quantitative Data Data that provide or contain information as to how much or how many; hence they are always numeric. A variable that assumes quantitative values is called a Quantitative variable.

R

Random Sample A sample drawn in such a way that each member of the population has an equal chance of being selected.

Regression Analysis A statistical technique for measuring/quantifying the type of causal relationship among variables; one of which is the Dependent Variable (DV) while the others are the Independent Variables (IVs). The analysis is called Simple Regression if there is only one IV in the model; it is called Multiple Regression if there are two or more IVs in the model. A regression model whether in the Simple or Multiple form can be used for prediction purposes as well as for testing existing economic theories, among others.

S

Sample A subset of the population of interest to the researcher. The size is often denoted as n. In practice, we will be interested in a random sample for the purpose of making reasonable inferences about the population being studied/analyzed.

Sample Statistic A summary measure/value computed from a sample of data. Thus, this value is always known.

Small multiples are small, thumbnail-sized representations of multiple images displayed all at once, which allows the reader to immediately, and in parallel, compare the inter-frame differences.

Standard Deviation A measure of dispersion for a body/set of data, found by taking the positive square root of the variance.

Statistical Analysis (Types) A statistical analysis is said to be Univariate if the applicable technique involves only one statistical variable, it is said to be Bivariate if the applicable technique involves two variable and it is said to be Multivariate if the applicable technique involves more than two variables.

Symbol is Image, figure or object that represents an abstract, moral or intellectual concept. The symbol has to be distinguished from the sign. A symbol implies more than its immediate meaning.

T

Temporal data refers to data, where changes over time or temporal aspects play a central role or are of interest.

Timeline A timeline is a graphical or textual display of events in chronological order and is the most used technique for interacting with time-linear visual information. It also allows the user to explore relationships among historical events.

Transformation Replacing each data value by a different number (such as its logarithm) to facilitate statistical analysis. The logarithm often transforms skewness into symmetry by stretching the scale near zero, thus spreading out all the small values that had been bunched together. It also pulls together the very large data values which had been thinly scattered at the at the high end of the scale.

U

Uniform Probability Distribution A probability distribution in which equal probabilities are assigned to all values of a random variable. The distribution can be a pdf or a pmf depending on whether the random variable X is continuous or discrete.

User-centered design is an iterative process involving task analysis?, design, prototype implementation, and testing, as illustrated in Fig. 3. Users are involved as much as possible at each design phase.

V

Variable A characteristic or an attribute of the statistical unit/entity of interest with values that are numeric (in the case of a quantitative variable) or non-numeric (in the case of a qualitative variable). The standard notation for a variable is X in the case of a univariate analysis, X and Y in the case of a bivariate analysis, or X , Y and Z in the case of a three-variable multivariate analysis.

Visual analysis aims for supporting the verification or falsification of given hypotheses about a dataset visually. This means that users perform a directed search for information. A high degree of interactivity is required here as well. Since hypotheses are given a priori visual analysis techniques should be able to guide the users during the search process. Visual analysis techniques are often tailored to one certain analysis problem (i.e., a limited set of hypotheses).

Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces.

Visual Clutter Clutter is the state in which excess items, or their representation or organization, lead to a degradation of performance at some task.

Visual Cue A visual cue is a signal and reminder of something; aiming to be self-explanatory and preattentive, it brings to mind knowledge from previous experiences providing a framework for its own interpretation.

Visual Exploration The aim pursued with visual exploration is to give an overview of the data and to allow users to interactively browse through different portions of the data. In this scenario users have no or only vague hypotheses about the data; their aim is to find some. In this sense, visual exploration can be understood as an undirected search for relevant information within the data. To support users in the search process, a high degree of interactivity must be a key feature of visual exploration techniques.

Visual mapping is a mapping between data aspects and visual variables, i.e., assigning specific visual characteristics to data attributes in order to facilitate visual sense-making.

Visual presentation aims for communicating and sharing information in a dataset visually to others (e.g., experts for detailed analysis). The visual presentation must emphasize the relevant information among the rest of the data.

Visualization A graphical representation of data or concepts, which is either an internal construct of the mind or an external artifact supporting decision making.

Visualization Pipeline The visualization pipeline describes the (step-wise) process of creating visual representations of data.

Visualization Session The interactive use of visual representations is considered a visualization session.

Visual Variables are specified set of modifications that can be applied to objects in order to encode information.

W

Wisdom The ultimate level of understanding. With it we understand a broad enough set of patterns and meta-patterns in such a way that we can use and combine them in new ways and situations that are completely different to those that served us to learn.

Z

Zoom zooming is one of the basic interaction techniques of information visualizations. Since the maximum amount of information can be limited by the resolution and color depth of a display, zooming is a crucial technique to overcome this limitation.

Appendix B: GNU Free Documentation License

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, \LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of

copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this

License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with

the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

Appendix C: General Public License version 3 (GPLv3)



Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this
license document, but changing it is not allowed.

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you

want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

1. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

2. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

3. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control

copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

4. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

5. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

6. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- (a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

- (b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- (c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- (d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

7. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- (a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- (b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange,

for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

- (c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- (d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- (e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, autho-

rization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

8. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.)

You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- (a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- (b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- (c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- (d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- (e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- (f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

9. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

10. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

11. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations.

If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

12. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other

readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

13. No Surrender of Others’ Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program,

the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

14. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

15. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

16. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK

AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

17. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

18. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
<one line to give the program's name
and a brief idea of what it does.>\par Copyright (C) <textyear> <name
of author>\par This program is free software: you can redistribute
it and/or modify it under the terms of the GNU General Public License
as published by the Free Software Foundation, either version 3 of
the License, or (at your option) any later version.\par This program
is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.\par You should have received a copy of the GNU
General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name
of author>\par This program comes with ABSOLUTELY NO WARRANTY; for
details type 'show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type 'show c' for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Index

- 3D scatter plot, 198
- 5-tuple, 92
- Accounting, 85
- Advance Scan with Unicornscan, 28
- Advance Security Analysis, 68
- Afterglow, 169
- Afterglow Functions, 176
- Afterglow Parsers, 170
- animated glyphs, 183
- Application Classification, 115
- Application Identification, 87
- Application Layer, 7, 35
- Application Recognition, 87
- Argus, 48, 49
- Argus and Netflow, 54
- Argus Client, 50
- Argus Installation, 49
- Argus sensor, 48
- ARP Poisoning, 10
- ARP Poisoning with Ettercap, 12
- ASCII, 180
- Associations chart, 150
- attribute meter, 88
- Availability**, 3
- backup, 242
- Bar Chart, 152
- basic statistics, 74
- Behavior and Heuristics, 114
- behavioral baselining, 49
- bi-directional, 51
- Binary Rainfall, 182
- Bison, 49
- BitTorrent, 224
- Blocking, 86
- Bookmark, 118
- Breadcrumb, 118
- Business Case for DLP, 232
- Byte Frequency, 180
- byte-values, 87
- ByteFrequencyMeter*, 90
- capinfos*, 16
- CAPTURE, 1
- CDP, 242
- Chaosreader, 99
- Circo, 175
- Cisco Netflow, 54
- Cleaning, 145
- Closed port, 30
- Collection, 118
- Collection**, 3
- Collection Navigation, 119
- Comma-Separated Values, 59
- Command menus, 19
- common cause variation, 157

- Comparison of Protocol Models, 91
- complex aggregations, 54
- components of visual perception, 140
- compression, 7
- con, 57
- cones, 140
- Confirmative Analysis, 143
- connection, 7
- connection oriented, 6
- connectionless, 6
- Content, 119
- content monitoring and filtering, 231
- Continuous Data Protection, 241
- Control, 234
- Control Charts, 156
- control totals, 74
- counter bins, 88
- Creating firewall rules with Wireshark, 46
- CSV, 59
- Data Analysis**, 142
- Data Analysis with Picalo, 70
- Data at rest, 230, 238, 240
- data conversion, 60
- Data dimension reduction, 146
- data exploration*, 148
- Data Formation, 146
- Data Formatting, 147
- data gathering*, 148
- Data in motion, 230, 238, 240
- Data in use, 231, 238
- Data Leak Prevention, 229
- Data Leakage, 229
- Data Leakage Problem, 231
- data loss prevention, 231
- data manipulation*, 148
- Data Mining, 237
- Data normalization, 145
- Data protection, 241
- Data Protection Act, 243
- Data Protection Policy, 245
- Data smoothing, 145
- data structure, 70
- Data Tables, 144
- Data Transformations**, 144
- data transformation, 137
- Data Types, 240
- data unit, 6
- DataEcho, 103
- DAVIX, 206
- DDoS, 220
- decompression, 7
- decryption, 7
- Deep Packet Inspection, 216
- Deep Packet Processing, 216
- Depth**, 3
- descriptives, 74
- Diagram Refresh Period*, 188
- Discovery, 235
- Distributions chart, 150
- DLP, 229
- DLP Components, 234
- DLP Operating Levels, 230
- DLP Phases, 235
- DLP Workings, 233
- DNS attacks, 220
- Dot, 173
- DPI, 87, 216
- DPI Analogy, 218
- DPI Concerns, 220
- DPI Evasion Techniques, 224
- DPI Uses, 220

- Drill, 118
- dumpcap*, 22
- dynamic*, 24
- E-discovery, 237
- Edge Table, 190
- Encryption, 238
- encryption, 7
- Encryption Tools, 240
- Enforcing AUP**, 220
- EnigMail**, 241
- entropy, 88
- Envisioning Information, 140
- EtherApe, 183
- Ethernet switches, 8
- Ettercap, 10
- Ettercap Installation, 12
- Ettercap Plugins, 14
- Evolution, 168
- Explorative Analysis, 143
- Extracting and Preserving, 21
- extrusion detection, 231

- False negatives, 115
- False positives, 115
- Fdp, 176
- fear, uncertainty and doubt, 218
- Fedora Core 10, 12
- Feedback Loops, 148
- field*, 71
- Filtered port, 30
- Filtering**, 142
- Filters in EtherApe, 188
- FingerBank*, 104
- Fingerprints, 90
- First Amendment of the Internet*, 223
- Flex, 49

- flow key, 52
- flow status records, 51
- fovea, 140
- FoxyProxy*, 226
- frame*, 6
- fraud detection, 68
- frequency analysis, 90
- Frequency charts, 150
- Frequency Polygon, 158

- gateways*, 6
- gathering, 68
- GGobi, 201
- GIGO, 149
- Global Protocol timeouts*, 188
- gnmap*, 61
- GnuPG**, 241
- Grant Privileges, 66
- Graph View, 190
- Graphing, 148
- Grouped bar charts, 153

- Header, 218
- header, 6
- heat maps, 168
- hexadecimal, 180
- Hierarchical parallel coordinates, 163
- histogram, 193
- Histograms, 157
- hop-count, 48
- Host Name**, 227
- HTTPS, 240
- Human-Computer Interaction, 168
- human-computer interaction, 137

- IANA, 24, 86
- Iconic memory, 139

- Identification, 234
- IETF IPFIX, 49
- Imagery Analysis, 183
- Index, 119
- indexing engine, 236
- InetVis, 198
- information leak prevention (ILP), 231
- Information Visualization, 142
- Information visualization, 137
- insider thread protection, 231
- Integration, 237
- Interactive GGobi, 202
- Internet Layer, 6, 23
- Investigating Spyware Infection, 44
- Investigator, 115
- Investigator Operations, 121
- Investigator Options, 132
- IP forwarding, 11
- ISO27001, 233

- K-L divergence, 91
- Kullback-Leibler divergence*, 91

- L2 addresses, 48
- L4 transport identification, 48
- Layer 7 Monitoring, 86
- leak prevention, 217
- Learning, 237
- Libpcap*, 43
- Limitations of Visualization, 165
- Line Chart**, 193
- Line Graphs, 155
- Link Graphs, 164
- Link Timeouts*, 188
- Linux, 8
- Logi Report, 78
- Logi Report Server, 78

- Logi Studio, 78
- Long-term memory, 139

- MAC address, 10
- Malware Behaviour Analysis, 45
- Malware Investigation, 123
- Man-in-the-middle, 9
- mandatory access control, 48
- Mapping**, 142
- MapQuest, 168
- Means, 221
- Memory Limitations, 139
- Metadata, 119
- Missing values solution, 145
- modeling, 68
- Modular Visualization Environments, 167
- Monitoring, 85, 234, 235
- Motive, 222
- MySQL, 65
- MySQL client, 65
- MySQL server, 65

- Naeto, 173
- Navigation, 118
- Ncat**, 31
- Near Continuous, 243
- Nessus, 36
- Nessus Installation, 37
- Net Neutrality, 223
- NetGrok, 189
- NetGrok Advantages, 194
- NetWitness Investigator, 115
- NetWitness Live, 116
- Network Access**, 220
- Network Flow, 51
- Network forensics Analysis Tool, 103
- Network Interface Layer, 5, 8

- Network Packet Sniffing, 14
- Network Security**, 220
- NetworkMiner, 103
- NFAT, 103
- NINO, 149
- Nmap, 30, 34
- Nmap Installation, 31
- nmapxmlparser, 62
- Node Size Variable*, 187
- Node Timeouts*, 188
- NumEmpty*, 74
- Numerical Properties, 112
- NumNone*, 74
- NumZero*, 74

- occlusion, 183
- ODBC, 70
- Open port, 30
- OpenSSH, 224
- Opportunity, 221
- orthogonal coordinates, 160
- outbound content management, 231

- p0f*, 104
- P2P, 224
- packet*, 6
- Packet Analysis, 41
- Packet Analysis with Wireshark, 44
- Packet Capture, 132
- Packet Capture Libraries, 42
- Packet Capture with Tcpdump, 16
- Packet Capture with Wireshark, 20
- Packet Classification, 85
- packet display filter field, 20
- packet inter-arrival times*, 88
- packet order number- and direction combinations*, 88

- Packet Redirection, 10
- packet sizes, 88
- Packet-contents window, 20
- Packet-header details window, 20
- Packet-listing window, 20
- Packets*, 6
- parallel coordinate, 183
- Parallel Coordinates, 160
- Parallel coordinates Algorithm, 161
- Parallel Coordinates display**, 205
- Pareto Chart, 154
- Parser, 118
- Parsing Nessus Output, 63
- Parsing Nmap Output, 60
- Parsing Security Data, 60
- Passive Network Analysis, 96
- pattern matching, 88
- Payload, 218
- PCAP, 103
- pcap, 42
- pcap*, 18
- PCI DSS, 233
- physical address, 6
- Picalo, 68
- Picalo Database Connectivity, 75
- Picalo Database Query analyzer, 75
- Picalo Descriptives, 74
- Picalo Sorting, 74
- Pie Charts, 151
- Plotting Scheme, 199
- Port, 112
- port mirroring*, 9
- port monitoring*, 9
- Port Scanning, 25
- Ports, 23
- Presentation and Communication, 143

- Prevention, 234, 236
Privacy, 221
probability bins, 88
probability distributions, 88, 90, 91
PROCESS, 83
Processing Netflow Data, 56
Protocol Analysis, 41
Protocol Discovery, 87
protocol ID, 48
Protocol Model, 90
Protocol models, 91
Protocol models, 87
protocol representation key, 185
Protocol Stack Level, 187
Putty, 227
- QoS, 86
Quality of Service, 220
- ra*, 51, 56
racluster, 51, 57
RadialNet, 194
radium, 55
ragraph, 60
RAID, 242
rasort, 53
Ratop, 58
Reading Netflow Data, 55
record, 71
record counts, 74
registered, 24
regulatory compliance, 217
relative entropy, 91
Rendering, 142
Report Development Environment, 79
Reporting and OLAP, 237
resource allocation, 223
- rods, 140
router, 6
Rumint, 180
Run Chart, 155
run-sequence plots, 155
- SA, 58
Satori, 104
Saved Sessions, 227
Scatter Plots, 159
Scatterplot Matrix, 205
Search, 118
Searching Content, 130
secure SSH tunnels, 224
Security analysis, 68
security analysis, 3
Security Analysis with Argus, 49
Security Data, 4, 21
Security Data Analysis, 41
Security Data Capture, 7
Security Data Carving, 85
Security Data Exploration, 111
security data exploration, 119
Security Data Management, 213
Security Reporting, 77
Security Visual Mapping, 137
Security Visualization Techniques, 167
Selection, 145
sense-making, 139
session, 6
Session Reconstruction, 122, 128
Sessions, 119
Shallow Packet Inspection, 215
Short-term memory, 139
Signature Analysis, 111
signature analysis techniques, 111

- Simple bar charts, 152
- Situation Awareness, 58
- SLAX, 206
- snort, 48
- Socket, 24
- SOCKS, 224
- SOX, 233
- SPID, 87
- SPID Algorithm Overview, 87
- SPID Algorithm Requirements, 88
- SPID Algorithm Tuning, 96
- SPID Data Flow, 89
- SPID PoC, 92
- SPID Results, 92
- Spinning Cube of Potential Doom*, 198
- SSH, 240
- SSL, 240
- Stacked bar charts, 153
- statistical fingerprints, 90
- Statistical Protocol Identification, 86
- stdout*, 34
- Storage Area Networks, 243
- stream extractor, 108
- String Match, 112
- Switch Port Mirroring, 8
- Switched Port Analyzer*, 9
- syntactic analysis, 60
- System Cache, 243

- TCP/IP Reference Model, 5
- Tcpdump, 15
- Tcpdump Installation, 15
- tcpdump2csv*, 203
- Tcpxtract, 97
- Text Rainfall, 180
- TFTPgrab, 108

- The Network Visualizer, 178
- The parallel coordinate plot, 182
- Threat Evolution, 214
- threat intelligence, 116
- Time Series graphs**, 205
- Time-based Network Visualizer, 178
- tissynbe, 64
- TNV, 178
- traffic accounting, 48
- Traffic analysis, 48
- Traffic Classification, 87
- traffic flow, 48
- Traffic Flow Analysis, 48
- trailer, 6
- transforming, 68
- transmission error, 6
- Transport Layer, 7
- Transport Layers, 23
- Tree Graphs, 163
- TreeMap View, 191
- Trends chart, 150
- TrueCrypt**, 241
- tunnel identifiers, 48
- Twopi, 175

- uni-directional flow, 56
- Unicornscan, 26
- Unicornscan Installation, 26

- VA, 36
- variation of replication technology, 243
- View, 118
- View Transformations**, 144
- Views, 144, 147
- virtual circuit, 7
- Visibility**, 3
- VisiCalc, 168

Visual Data Representation, 149
visual mapping and transformation, 137
Visual Mappings, 144
Visual Perception, 138
Visual Structures, 144, 147
Visualization, 148
Visualization Benefits, 143
Visualization Objectives, 143
Visualization Process, 144
Visualization Tools, 169
VISUALIZE, 135
Vulnerability Assessment, 36

well known, 24
Windump, 15
Winpcap, 43
Wireshark, 17
Wireshark Installation, 18

xml, 61

yum, 15
yum, 12

Zenmap, 31