SECURITY

Network Security Auditing

The complete guide to auditing network security, measuring risk, and promoting compliance

ciscopress.com

Chris Jackson, CCIE® No. 6256

Network Security Auditing

Chris Jackson, CCIE No. 6256

Published by:

Cisco Press

800 East 96th Street

Indianapolis, IN 46240 USA

ISBN-13: 978-1-58705-352-8

ISBN-10: 1-58705-352-7

Printed in the United States of America

First Printing June 2010

Library of Congress Cataloging-in-Publication Data: Library of Congress Cataloging-in-Publication data is on file.

Warning and Disclaimer

Introduction

Mention the word audit to IT professionals and you will probably see their eyes glaze over as they imagine frightening visions of auditors with pointy tails, pitchforks, and checklists running around and pointing out all of the things they have done wrong to their manager. The purpose of a security audit is not to place blame or pick apart network design, but to ensure the integrity, effectiveness, and compliance of corporate security policies. Auditing provides the ability to test the assumptions companies have about how secure they think they are from threats and to gauge whether or not policies map to industry best practices and compliance laws. An organization's level of risk is quantified by placing a value on the assets of the business and analyzing what impact the exploitation of vulnerabilities can have to the business as a whole. Auditors find risk and check to see whether the appropriate controls are in place to mitigate exposure to that risk.

Auditing is not just about running a bunch of hacker tools in an attempt to break into the network. There are many types of audits, and the scope of an audit defines what the auditor inspects and how often. Many organizations require an annual audit of key systems by an outside firm (external audit), whereas others also mandate internal audits every six months or before and after any major IT project. If you are subject to PCI compliance requirements, you might need to have an audit preformed every quarter. The bottom line is if you aren't auditing today, you will be forced to through regulations or encouraged to by industry best practices. It simply makes good business sense to measure the effectiveness of your security investment.

The ultimate benefit of auditing is to continuously improve the processes, procedures, and controls put in place to secure valuable corporate assets. Businesses today have a responsibility to their customers to safe guard their confidential data. Numerous high-profile security failures have shattered that trust through carelessness while handling backup media and allowing millions of credit cards and financial records to fall into the hands of individuals determined to illegally profit at the expense of others. It takes only one major breach to appear in the news for a company to experience significant loss of shareholder value and sometimes even the total loss of the company itself. Having a policy and enforcing it are essential to protecting your business. Auditing that policy plays a key role in making sure that the policy actually accomplishes the goal of reducing risk and therefore protects key assets from loss. A large percentage of security failures can be minimized or prevented with a strong risk-based auditing program.

Goals

The goal of this book is not to be yet another hacker book devoted to the latest tools and techniques for breaking into networks. Those skills are useful, but are not the primary focus of a security audit. There have been many books devoted to that topic and they are typically out of date by the time they come to press because of the speed in which technology changes. This book is about measuring the deployment of Cisco security technologies to mitigate risk. Baseline technical testing is covered from a process standpoint, but the focus is not on penetration testing.

This book provides the reader with a practical guide to building an auditing and assessment program that factors in regulatory and industry security requirements, with real examples of how Cisco products can help address those needs. Recognizing that security is a system that relies on strong policy is the key principle. The value of the book lies in its ability to show real applications of Cisco security technology in the context of an auditing framework. Here are the key benefits of the book:

- Provides an overview of the auditing process and introduces important regulations and industry best practices.
- Demonstrates how to use commercial and open source tools to assist with the auditing process and validate security policy assumptions.
- Introduces IT governance frameworks such as Cobit, ITIL, and ISO 17799/27001 while providing guidance about how to leverage each with Cisco security products.
- Shows the reader how to segment security architectures into domains that provide a systems approach to auditing Cisco networks.
- Supplies a detailed auditing checklist after each domain for the reader to utilize in an auditing program.
- Provides design guidance for meeting auditing requirements and shows how complementary security solutions greatly increase the overall security posture of a company.
- Guides the reader to build an auditing program that utilizes the techniques presented in the book.

Who Should Read This Book?

This book is geared toward beginner to intermediate-level auditing and more specifically, auditing as it pertains to Cisco networks. The content is useful to anyone who wants to build a program to measure the effectiveness of Cisco security products. IT governance and auditing have common roots with financial auditing, and in many cases, it is ultimately the responsibility of the CFO in larger organizations. The language and procedures an IT auditor follows are similar to how a CPA might examine the books to certify that a business is keeping its records accurately and paying its taxes on time. Both disciplines keep there eyes open for fraud and try to anticipate how a system of controls can be circumvented. Every aspect of auditing, such as database auditing or web applications, is not covered as the focus of this book is on auditing the network. Numerous other books are dedicated to application and website auditing and would be better at providing a deeper understanding in those areas. If you are an IT auditor, security consultant, InfoSec manager, or someone who wants to assess his own network for good security deployment practices, then this book is for you.

How This Book Is Organized

The organization of this book breaks the material up into two major parts. The first part covers the principles of auditing and strives to teach the language and key components of the auditing process. This overview pulls together a number of techniques for identifying risk and shows how we must think like auditors in our network designs and device configurations. It also covers the major regulatory, industry compliance, and security framework initiatives. The section ends with a description of common auditing tools and techniques that can be used to assess and verify that the policy is enforced by technical controls.

The second part, consisting of Chapters 5 through 12, covers the major Cisco security solution domains, which break down Cisco security technologies into seven categories that enable the auditor to examine network security as a system of integrated components rather than individual products. Each chapter discusses the risks, threats, policies, procedures, and technical controls that can be deployed to defend each domain. Best practices on network security design and configuration are covered, too. The reader is also supplied with a checklist that can be used as a starting point or reference for auditing.

The following provides more detail on the contents of each chapter:

Part I, "Principles of Auditing"

Chapter 1, "Principles of Auditing": This chapter defines security fundamentals including policies, procedures, standards, and controls. The basics of risk management and the how, what, and why audits are performed. In addition, the auditing process is outlined with a six-step methodology that can be used in performing an audit.

Chapter 2, "Security and the Law": This chapter is about IT security laws and regulatory compliance with an overview of many of the major federal and state statutes governing IT Security. SOX, HIPAA, and GLBA are covered in addition to the PCI standard.

Chapter 3, "Security Governance, Frameworks, and Standards": Security governance frameworks such a COSO, Cobit, and ITIL help businesses coordinate people, process, and technology around security objectives. This chapter covers these frameworks, and also includes where to find source material that can be useful in building standards, procedures, and guidelines for security technology deployment.

Chapter 4, "Auditing Tools and Techniques": This chapter addresses the basics for evaluating security controls through technical testing. A combination of open source, commercial, and integrated Cisco testing tools are presented.

Part II, "Mapping Cisco Security Controls to Auditing Requirements"

Chapter 5, "Security Solutions Domains": Security solution domains are introduced in this chapter as a method for assessing network security as an interconnected system. This chapter also discusses building checklists for security audits.

Chapter 6, "Policy and Compliance": Policy and compliance is the first auditing domain and is focused on assessing security policies. This chapter provides an overview of key security policies that businesses should have and how they should be constructed.

Chapter 7, “Infrastructure Security”: This chapter covers assessing baseline security features and configuration that should be implemented on Cisco routers, switches, and wireless devices.

Chapter 8, “Perimeter Intrusion Prevention”: Assessing perimeter defenses is covered in this chapter, with a focus on firewalls and intrusion prevention systems.

Chapter 9, “Access Control”: Access control technologies enable the enforcement of role-based access requirements that follow the principle of least privilege. This chapter describes how to assess identity-based networking solutions and network admission control.

Chapter 10, “Remote Access”: This chapter covers how to assess VPN technologies including site-to-site and mobile-user VPNs. Best practices for deployment and testing methods are also discussed.

Chapter 11, “Endpoint Protection”: Endpoint protection is about preventing and detecting attacks targeted at users and their network devices. This chapter discusses methods that can be used to assess policies, procedures, and controls to protect endpoints from web, email, malware, and data loss.

Chapter 12, “Unified Communications”: This chapter addresses auditing Unified communications systems policies, procedures, and security controls used to maintain confidentiality and defend against fraud.

Chapter 4

Auditing Tools and Techniques

Assessing security controls involves more than simply scanning a firewall to see what ports are open and then running off to a quiet room to generate a report. It is natural for security engineers to gravitate toward technology and focus on technical security control testing (otherwise known as penetration testing), because it is likely the "fun" part of security for most engineers. Conducting a penetration test is like throwing down the gauntlet to security professionals, and it gives them an opportunity to flex their hacker skills. Testing security as a system, however, involves significantly more than launching carefully crafted evil packets at the network to see what happens. This chapter discusses software tools and techniques auditors can use to test network security controls.

It is important to note that this is not a chapter about hacking. You will not learn all of the techniques and tools available today for breaking into networks. Do a search at your favorite online bookseller for the terms hacking, hacker, or penetration testing and you will find a slew of books devoted to the topics. Security testing as a process is covered, but the focus is on gathering the evidence useful for an audit. Thoroughly assessing security controls serves a vital part in determining whether or not a business is compliant with its policies, procedures, and standards. Through security controls testing, you can determine whether the organization meets its goals for reducing risk and keeping evildoers out of the network and away from critical systems.

Evaluating Security Controls

Security controls are the safeguards that a business uses to reduce risk and protect assets. Policy determines what security controls are needed, and those controls are selected by identifying a risk and choosing the appropriate countermeasure that reduces the impact of an undesirable event (such as a customer database being stolen). The evaluation of security controls in its simplest form validates whether or not the control adequately addresses policy, best practice, and law. Testing security controls for effectiveness and measuring them against standards are of the best ways to help an organization meet its obligations to shareholders and regulatory responsibilities.

As discussed in Chapter 1, "The Principles of Auditing," the main security control types are administrative, technical, and physical. Under each category, the specific controls that can be implemented are preventative, detective, corrective, or recovery. These control types work together, and in general, you must provide controls from each category to effectively protect an asset. When testing controls, make sure that each functional category is addressed and all controls are implemented in a way that doesn't allow someone easy circumvention. You can have the most advanced firewall in the world as a preventative control, but without monitoring its effectiveness through detective controls, such as log reviews and IPS, you would never know for sure if it enforced policy. These missing pieces are typically what hackers exploit to break into systems, and it's the auditor's job to identify and report on weaknesses in the system.

When evaluating security effectiveness, you need to examine three primary facets for every control. All security incidents, from break-ins to lost customer records, can usually be traced back to a deficiency that can be attributed to people, process, or technology. Testing these areas enables you to analyze security from a big picture perspective, gives you a better understanding of how an organization performs today, and recommends improvements for tomorrow. Following are the three facets to examine:

- People are users, administrators, data owners, and managers of the organization with varying levels of skills, attitudes, and agendas. If users are not following security policies, there might be a need for stronger administrative controls such as security awareness training or penalties for noncompliance (this is the "up to and including getting fired" clause that HR puts in the employee manual). An organization can also implement a detective/corrective control to enforce policies such as having the latest antivirus updates or operating system patches before the user is allowed on the network. People also represent the organizational structure and policies that drive security.
- Process represents how the organization delivers the service of IT. These are the procedures and standards that are put into place to protect assets. Processes must be up to date, consistent, and follow best practices to be effective. Process is one of the most important areas to test, because most attacks that result in significant loss have a component in which process has failed. Take, for example user account creation and decommission. Someone is hired, and a request is put into IT to create the appropriate accounts the new hire. Who is allowed to send the request? Is it any hiring manager or does it have to be one from Human Resources? How is the request validated as legitimate? Without strong process and the appropriate controls in place to prevent, detect, and correct, anyone can call and impersonate a hiring manager and request an account be created. This is significantly easier (and quicker) than trying to run a brute force, password-cracking tool against a server.
- Technology represents the facilities, equipment, computer hardware, and software that automate a business. Technology enables people to accomplish repetitive jobs faster and with less error. Of course, technology also enables someone to do stupid things just as efficiently (and faster). Misconfigurations and poorly implemented software can take a mistake and multiply its impact exponentially. Imagine leaving the door unlocked on a room that houses hardcopy files. Someone could potentially

walk into the room and take files, but it would take a long time (not to mention effort) to hand carry those documents out to a car. Now, imagine misconfiguring a server in the DMZ to allow for access from the Internet to a key database server. Someone could download the entire database and not even leave a trace that they were there. This is why it is so important for a business to standardize on best practices and configurations that are known to work. Best practices tend to anticipate many of these scenarios.

Evaluating security controls requires the auditor to look at a system with the eyes of a hacker and anticipate how things could be exploited to gain unauthorized access. Just because something "shouldn't" be exploitable, doesn't mean that it isn't. The only way to know is to test the system and the individuals who are tasked with monitoring and maintaining it should do the testing.

Auditing Security Practices

The first step for evaluating security controls is to examine the organization's policies, security governance structure, and security objectives because these three areas encompass the business practices of security. Security controls are selected and implemented because of security policies or security requirements mandated by law. Security is a service provided by IT to the business, so measuring it as such enables you to see many of the connections to the various functions of the business. As discussed in Chapter 3, "Information Security Governance, Frameworks, and Standards," there are standards, laws, and benchmarks that you can use as your baseline to compare against. Normally, you include content from multiple areas, as businesses may have more than one regulation with which they must comply. It is easiest to start with the organization's policies and build your security auditing plan from there. Some criteria you can use to compare the service of security against are:

- Evaluation against the organization's own security policy and security baselines
- Regulatory/industry compliance—Health Insurance Portability and Accountability Act (HIPAA), Sarbanes-Oxley Act (SOX), Grahmm-Leach-Bliley Act (GLBA), and Payment Card Industry (PCI)
- Evaluation against standards such as NIST 800 or ISO 27002
- Governance frameworks such as COBIT or Coso

After you have identified the security audit criteria that the organization needs to comply with, the next phase is to perform assessments to determine how well they achieve their goals. A number of assessments are usually required to determine appropriate means for referring back to the scope, which defines the boundaries of the audit. The following are types of assessments that might be preformed to test security controls:

- **Risk assessments:** This type of assessment examines potential threats to the organization by listing areas that could be sources of loss such as corporate espionage, service outages, disasters, and data theft. Each is prioritized by severity, matched to the

identified vulnerabilities, and used to determine whether the organization has adequate controls to minimize the impact.

- **Policy assessment:** This assessment reviews policy to determine whether the policy meets best practices, is unambiguous, and accomplishes the business objectives of the organization.
- **Social engineering:** This involves penetration testing against people to identify whether security awareness training, physical security, and facilities are properly protected.
- **Security design review:** The security design review is conducted to assess the deployment of technology for compliance with policy and best practices. These types of tests involve reviewing network architecture and design and monitoring and alerting capabilities.
- **Security process review:** The security process review identifies weaknesses in the execution of security procedures and activities. All security activities should have written processes that are communicated and consistently followed. The two most common methods for assessing security processes are through interviews and observation:
  - **Interviews:** Talking to the actual people responsible for maintaining security, from users to systems administrators, provides a wealth of evidence about the people aspect of security. How do they feel about corporate security methods? Can they answer basic security policy questions? Do they feel that security is effective? The kind of information gathered helps identify any weakness in training and the organization's commitment to adhering to policy.
  - **Observation:** Physical security can be tested by walking around the office and observing how employees conduct themselves from a security perspective. Do they walk away without locking their workstations or have sensitive documents sitting on their desks? Do they leave the data center door propped open, or do they not have a sign-out procedure for taking equipment out of the building? It is amazing what a stroll through the cubicles of a company can reveal about the security posture of an organization.
- **Document review:** Checking the effectiveness and compliance of the policy, procedure, and standards documents is one of the primary ways an auditor can gather evidence. Checking logs, incident reports, and trouble tickets can also provide data about how IT operates on a daily basis.
- **Technical review:** This is where penetration testing and technical vulnerability testing come into play. One of the most important services an auditor offers is to evaluate the competence and effectiveness of the technologies relied upon to protect a corporation's assets.

This section covered evaluation techniques for auditing security practices within an organization. Many of the security practices used to protect a company are process- and

policy-focused. They represent the primary drivers for technology purchases and deployment. Technology can automate many of these processes and policies and needs a different approach to testing effectiveness. The remainder of this chapter covers tools that can be used to test security technologies.

Testing Security Technology

There are many terms used to describe the technical review of security controls. Ethical hacking, penetration test, and security testing are often used interchangeably to describe a process that attempts to validate security configuration and vulnerabilities by exploiting them in a controlled manner to gain access to computer systems and networks. There are various ways that security testing can be conducted, and the choice of methods used ultimately comes down to the degree to which the test examines security as a system. There are generally two distinct levels of security testing commonly performed today:

- **Vulnerability assessment:** This technical assessment is intended to identify as many potential weaknesses in a host, application, or entire network as possible based on the scope of the engagement. Configurations, policies, and best practices are all used to identify potential weaknesses in the deployment or design of the entity being tested. These types of assessments are notorious for finding an enormous amount of potential problems that require a security expert to prioritize and validate real issues that need to be addressed. Running vulnerability scanning software can result in hundreds of pages of items being flagged as vulnerable when in reality they are not exploitable.
- **Penetration test:** The penetration test is intended to assess the prevention, detection, and correction controls of a network by attempting to exploit vulnerabilities and gain control of systems and services. Penetration testers (also known as pentesters) scan for vulnerabilities as part of the process just like a vulnerability assessment, but the primary difference between the two is that a pentester also attempts to exploit those vulnerabilities as a method of validating that there is an exploitable weakness. Successfully taking over a system does not show all possible vectors of entry into the network, but can identify where key controls fail. If someone is able to exploit a device without triggering any alarms, then detective controls need to be strengthened so that the organization can better monitor for anomalies.

Security control testing is an art form in addition to a technical security discipline. It takes a certain type of individual and mindset to figure out new vulnerabilities and exploits. Penetration testers usually fit this mold, and they must constantly research new attack techniques and tools. Auditors, on the other hand, might not test to that degree and will more than likely work with a penetration tester or team if a significant level of detailed knowledge in required for the audit. When performing these types of engagements, four classes of penetration tests can be conducted and are differentiated by how much prior knowledge the penetration tester has about the system. The four types are:

- **Whitebox:** Whitebox testing is where the tester has complete information about the design, configuration, addressing, and even source code of the systems under test.

This type of test is generally used to simulate a worst-possible scenario of an attacker who has intimate knowledge of the network and systems.

- **Blackbox:** Blackbox testing is the classical penetration test in which the tester simulates an external hacker and is given no information about the subject under test, other than what he can glean from the testing methods. The concept of this type of test is to identify weaknesses that can be exploited based on publicly available information.
- **Graybox:** This is a test that falls in the middle of the other two types in that some information is disclosed to the tester to "get him started." Intended to simulate the insider threat, the penetration tester might be provided network diagrams, IP addressing, and user-level access to systems.
- **Red Team/Blue Team assessment:** The terms Red and Blue Team come from the military where combat teams are tested to determine operational readiness. In the computer world, a Red and Blue Team assessment is like a war game, where the organization being tested is put to the test in as real a scenario as possible. Red Team assessments are intended to show all of the various methods an attacker can use to gain entry. It is the most comprehensive of all security tests. This assessment method tests policy and procedures, detection, incident handling, physical security, security awareness, and other areas that can be exploited. Every vector of attack is fair game in this type of assessment.

Auditors should have a base knowledge of testing tools and techniques. Using testing frameworks is a useful way to develop a technical testing planning. The next section introduces a couple of well known testing frameworks.

Security Testing Frameworks

There are numerous security testing methodologies being used today by security auditors for technical control assessment. Four of the most common are as follows:

- Open Source Security Testing Methodology Manual (OSSTMM)
- Information Systems Security Assessment Framework (ISSAF)
- NIST 800-115
- Open Web Application Security Project (OWASP)

All of these frameworks provide a detailed, process-oriented manner in which to conduct a security test, and each has its particular strengths and weaknesses. Most auditors and penetration testers use these frameworks as a starting point to create their own testing process, and they find a lot of value in referencing them.

OSSTMM

OSSTMM was developed under the Creative Commons License as a free methodology to conduct security testing in a thorough and repeatable manner. The current released version 2.2 of the manual highlights the systems approach to security testing by dividing assessment areas into six interconnected modules:

- **Information Security:** Competitive intelligence, data leakage, and privacy review
- **Process Security:** Access granting processes and social engineering testing
- **Internet Technologies Security:** Network mapping, port scanning, service and operating system (OS) identification, vulnerability scanning, Internet app testing, router/firewall testing, IDS testing, malicious code detection, password cracking, denial of service, and policy review
- **Communications Security:** Private branch exchange (PBX)/phone fraud, voicemail, fax, and modem
- **Wireless Security:** 802.11, Bluetooth, handheld scanning, surveillance, radio frequency identification (RFID), and infrared
- **Physical Security:** Perimeter, monitoring, access control, alarm systems, and environment

The OSSTMM has a strong following in the community and provides a good reference for what areas need to be examined and what types of results to expect. It is not a "click here, do that" type of document; rather, it requires a level of knowledge of various tools and techniques to accomplish the goals of the tests. Version 3.0 of the OSSTMM is a significant update that is still a work in progress. As of this writing, it is in beta with no timeline announced for release. Becoming a member of the project will provide access to the current beta draft and other documents such as templates and spreadsheets that can be used in conducting an audit with this methodology.

ISSAF

The ISSAF is one of the largest free-assessment methodologies available. Weighing in at 1200 pages, it provides a level of detail that is staggering. The authors believe that is it better to provide all of the information possible that an auditor might need than to limit it to high-level objectives. Each control test has detailed instruction for operating testing tools and what results to look for. It is split into two primary documents. One is focused on the business aspect of security, and the other is designed as a penetration test framework. The framework has not been updated in sometime (file date is 2006), but it is still useful as source material for controls testing and as a full-assessment methodology. The level of detailed explanation of services, security tools to use, and potential exploits is high and can help an experienced security auditor and someone getting started in auditing.

NIST 800-115

The NIST 800-115, Technical Guide to Information Security Testing, provides guidance and a methodology for reviewing security that is required for the U.S. government's various departments to follow. Like all NIST-created documents, 800-115 is free for use in the private sector. It includes templates, techniques, and tools that can be used for assessing many types of systems and scenarios. It is not as detailed as the ISSAF or OSSTMM, but it does provide a repeatable process for the conduction of security reviews. The document includes guidance on the following:

- Security testing policies
- Management's role in security testing
- Testing methods
- Security review techniques
- Identification and analysis of systems
- Scanning and vulnerability assessments
- Vulnerability validation (pentesting)
- Information security test planning
- Security test execution
- Post-test activities

OWASAP

The OWASP testing guide was created to assist web developers and security practitioners to better secure web applications. A proliferation of poorly written and executed web applications has resulted in numerous, easily exploitable vulnerabilities that put the Internet community at risk to malware, identity theft, and other attacks. As a nonprofit organization, OWASP has created a number of tools, guides, and testing methodologies that are free for anyone to use. The OWASP testing guide has become the standard for web application testing. Version 3 was released in December of 2008 and has helped increase the awareness of security issues in web applications through testing and better coding practices.

The OWASP testing methodology is split as follows:

- Information gathering
- Configuration management
- Authentication testing
- Session management
- Authorization testing

- Business logic testing
- Data validation testing
- Denial of service testing
- Denial of service testing
- Web services testing
- AJAX testing

Each test provides a summary of the issues, tools that can be used to assess the service, and examples of expected results. The information and examples given are thorough, and reference materials on the tools used or issues discussed are included at the end of each of the individual tests. The OWASP project also has a subproject called WEBGOAT that enables you to load a vulnerable website in a controlled environment to test these techniques against a live system.

Whatever your approach is to testing security controls, you must ensure that it is consistent, repeatable, and based on best practices. Your audits will be more thorough and you will be less likely to miss major issues that might slip by if you are "winging" your tests. Leverage the great resources that are available free from the security community and feel free to contribute your own ideas, so that everyone can benefit.

Security Auditing Tools

One thing is certain about security auditing tools: The power and sophistication of tools that auditors have at their disposal increase exponentially every year. Not only are the authors of these tools truly brilliant individuals (and some scary ones, too), they have also helped the security community significantly through the automation of advanced testing techniques.

If you attend Blackhat, DefCon, or other security conferences, you can see the latest and greatest additions to this growing list of powerful applications. Fyodor, the author of NMAP, has conducted a yearly survey of the members of his mailing list (over 4,000 high-energy security professionals) to rank the top 100 security tools. This list includes a number of the tools discussed in this section. There are many books written from the security tool perspective, with indepth discussions of the various uses, switches, and techniques to implement these programs. Consider this an introduction to the uses of these tools, and auditors are encouraged to read *Security Power Tools* from O'Reilly Press for a fantastic discussion of security tools and their many configuration options. There are also a number of free whitepapers and guides on the Internet. The following sections discuss a few commercial and open source assessment tools that can be used to effectively audit Cisco networks.

Service Mapping Tools

Service mapping tools are used to identify systems, remote services, and open ports. These types of tools can be used to test a firewall rule base or response given different real or crafted IP packets.

Nmap

Nmap is the network and service scanning tool of choice for most security professionals. It is a free, open source application available on all UNIX and Windows operating systems. The tool is command-line based, but there are a number of graphical frontends for those who want a point-and-click experience.

Nmap can be used to scan for service ports, perform operating system detection, and ping sweeps. Nmap uses an "operating systems normal" response to a valid connection request or "tear down" response to determine whether a port is open (listening and responding) or if it is not enabled. A typical TCP connection follows a three-way handshake to set up communications.

Step 1. Computer A sends a Syn packet to computer B to initiate communication-Syn.

Step 2. Computer B replies to computer A with an acknowledgement packet-Ack.

Step 3. Computer A sends a Syn acknowledgement packet to computer B to start the session-Syn Ack.

Step 4. A connection is established and data communications can begin.

Auditors can use Nmap to get a quick idea of what hosts and services are available on a network. It can be used to scan a single subnet or much larger networks. Nmap performs a ping sweep to identify hosts that are active on the network and then proceed to identify what services respond. You can also check the configuration of firewalls and access policies for critical systems.

Before using Nmap on UNIX type systems (LINUX, BSD, and Mac OS X), you need to obtain root privileges via SUDO to use any features that cause Nmap to create custom packets. Nmap can be run without administrative privileges, but some of the advanced scanning techniques such as SYN scanning and anything that needs to access the raw IP stack will fail.

If you execute Nmap with its default settings, and assuming you have root privileges, Nmap performs a SYN scan:

```
nmap 192.168.1.3
```

Nmap sends a SYN to all of the ports listed in its services file (over 1,000 ports) and looks for a SYN/ACK response. If it gets a response, it assumes that the port is open and immediately sends a RST (reset) to close the connection and then move on to the next port to be tested. If there is no response, Nmap assumes that the port is closed. The SYN scanning process is simple and is why Nmap can scan a host so quickly.

```
Starting Nmap 5.21 ( http://insecure.org )
```

```
Interesting ports on 172.16.1.3:
Not shown: 1707 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-term-serv
MAC Address: 00:1A:92:0A:62:B1 (Asustek Computer)

Nmap done: 1 IP address (1 host up) scanned in 2.226 seconds
```

Scanning for UDP ports is handled differently. Because UDP doesn't have a handshake process like TCP, the UDP packet must be crafted in a manner that causes the operating system to respond back. If you send a UDP packet to a closed port on a server, the TCP/IP stack is supposed to send an ICMP port unreachable message back. If a host does not send this response, it is assumed that the port is open. Obviously, a firewall can wreak havoc with a UDP scan, so it is a major limitation of searching for open UDP ports with tools like Nmap.

```
sudo nmap –sU 172.16.1.3
Starting Nmap 5.21 ( http://insecure.org )
Interesting ports on 172.16.1.3:
Not shown: 1481 closed ports
PORT     STATE          SERVICE
123/udp  open¦filtered ntp
137/udp  open¦filtered netbios-ns
138/udp  open¦filtered netbios-dgm
500/udp  open¦filtered isakmp
1434/udp open¦filtered ms-sql-m
1900/udp open¦filtered UPnP
4500/udp open¦filtered sae-urn
MAC Address: 00:1A:92:0A:62:B1 (Asustek Computer)

Nmap done: 1 IP address (1 host up) scanned in 62.419 seconds
```

Utilizing the OS detection and versioning features of Nmap is also useful for identifying the type of OS and versions of services that run on a remote system. Nmap enables you to perform versioning (-sV) and OS detections (-O) separately or together as a combined command (-A):

```
nmap –A 127.0.0.1
Starting Nmap 5.21 ( http://insecure.org )
Interesting ports on 172.16.1.253:
Not shown: 1707 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     Cisco SSH 1.25 (protocol 1.99)
23/tcp  open  telnet  Cisco router
```

```
80/tcp  open  http    Cisco IOS administrative httpd
443/tcp open  https?
MAC Address: 00:19:E8:3C:EE:40 (Cisco Systems)
Device type: switch
Running: Cisco IOS 12.X
OS details: Cisco Catalyst C2950 or 3750G switch (IOS 12.1 - 12.2)
Network Distance: 1 hop
Service Info: OS: IOS; Device: router
Nmap done: 1 IP address (1 host up) scanned in 18.877 seconds
```

Nmap provides several ways to mask your identity when scanning. One of the more popular mechanisms is through an idle scan. This is a clever technique that utilizes unique identifiers for every IP communication stream (IPIDS). Some operating systems simply increment the IPID every time a new connection is made. If you can find a host that is not being used, you can use it to bounce scans off of and make the remote system think the scan is coming from the idle host. To pull this off, you have to first find a host with incremental IPIDs.

```
nmap –sT –O –v 172.16.1.3
Starting Nmap 5.21 ( http://insecure.org )
Initiating ARP Ping Scan at 17:28
Scanning 172.16.1.3 [1 port]
Completed ARP Ping Scan at 17:28, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:28
Completed Parallel DNS resolution of 1 host. at 17:28, 0.05s elapsed
Initiating Connect Scan at 17:28
Scanning 172.16.1.3 [1711 ports]
Discovered open port 3389/tcp on 172.16.1.3
Discovered open port 135/tcp on 172.16.1.3
Discovered open port 139/tcp on 172.16.1.3
Discovered open port 445/tcp on 172.16.1.3
Completed Connect Scan at 17:28, 1.62s elapsed (1711 total ports)
Initiating OS detection (try #1) against 172.16.1.3
Host 172.16.1.3 appears to be up ... good.
Interesting ports on 172.16.1.3:
Not shown: 1707 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-term-serv
MAC Address: 00:1A:92:0A:62:B1 (Asustek Computer)
Device type: general purpose
Running: Microsoft Windows Vista
OS details: Microsoft Windows Vista
```

```
Uptime: 0.926 days (since Fri Jan  4 19:15:18 2008)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=260 (Good luck!)
IP ID Sequence Generation: Incremental
Read data files from: /opt/local/share/Nmap
OS detection performed. Please report any incorrect results at
http://insecure.org/Nmap/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.802 seconds
           Raw packets sent: 17 (1460B) ¦ Rcvd: 17 (1408B)
```

Now that you have found a host that can be used for stealth scanning, you simply need to use one of the TCP services to bounce off of. In this example, port 445 (Microsoft directory services) is used. It is important to disable the initial ping that Nmap sends by default (-P0) to see whether a host is up before scanning to prevent any packets from your computer being sent to the destination system you are trying to scan.

```
nmap -P0 -sI 172.16.1.3:445  172.16.1.253
Starting Nmap 5.21 ( http://insecure.org )
Idle scan using zombie 172.16.1.3 (172.16.1.3:445); Class: Incremental
Interesting ports on 172.16.1.253:
Not shown: 1707 closed¦filtered ports
PORT    STATE SERVICE
22/tcp  open  ssh
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https
MAC Address: 00:19:E8:3C:EE:40 (Cisco Systems)
Nmap done: 1 IP address (1 host up) scanned in 17.770 seconds
```

Going through the hundreds of ways an auditor can use Nmap is beyond the scope of this book. Suffice it to say, you should read the manual pages of Nmap carefully if you intend to fully exploit its capabilities. There is an excellent Nmap tutorial that can be read for free at http://nmap.org/bennieston-tutorial/. For a more thorough Nmap exploration, read *NMAP Network Scanning*, written by the tools creator Gordon "Fyodor" Lyon. Some examples of useful Nmap commands for auditors are included in Table 4-1.

Table 4-1 *Useful Nmap Commands*

Nmap Command Example	Description
nmap –sP 192.168.1.0/24	Ping the entire 192.168.1.0 subnet to see which hosts respond.
nmap –P0 192.168.1.5-11	Scan IP hosts at .5–11. Assume hosts are available for scanning, don't ping to check and perform a SYN scan. (By default, Nmap doesn't scan a host if it doesn't receive a ping response.)

continues

Table 4-1 *Useful Nmap Commands (continued)*

Nmap Command Example	Description
nmap –A 192.168.1.4	Scan host and attempt identification of services running on ports and the OS.
nmap –O 172.16.2.3	Scan host and attempt to identify what OS it runs.
nmap –p22,23,25 10.10.1.1	Scan a host to see whether ports 22, 23, and 25 are available.
nmap –sT –A –v 192.12.1.24	Scan a host with full a TCP connect and perform OS and service version detection with verbose reporting.

Hping

Hping is a tool that expands on basic ping functionality by providing the capability to create custom IP packets for the auditing and testing of security controls. Hping enables the sending of arbitrary packets, the manipulation of IP options and fields, and basic port-scanning capabilities. Not only does Hping send packets, but it also enables the auditor to set up a listening mode that displays any packets that return matching a certain pattern. This can be useful when testing security controls such as firewalls or intrusion detection system (IDS) and intrusion prevention system (IPS).

Note—Hping2 is the version used in this book, but it is also worth checking out Hping3, which is written in TCL for integrated scripting support and sports an interactive command-line interface. Hping3 is command-compatible with Hping2.

Some of the uses of Hping are:

- **Port scanning:** Hping provides basic port-scanning capabilities including an incremental option (++ before the port number) that enables an auditor to scan a range of ports with custom packets and TCP options. This tool doesn't replace Nmap, but provides a high level of control about exactly what packets get sent on the wire.
- **Network protocol testing:** Hping can create practically any packet you want to manufacture to test how a system responds to malformed communications.
- **Access control and firewall testing:** Hping can be used to test firewall and IDS rules to ensure they work as expected. Hping can accept input from a text file to create payload data that can be packaged and sent to a remote system (like exploit code). This feature can be used to verify IPS signatures and monitoring systems.

The following example shows Hping scanning ports from 134 to 140. Notice the SA flags in the response denoting a SYN ACK response on the live ports, and RA flags or Reset Ack on closed ports:

```
hping2 172.16.1.3 -S -p ++134
HPING 172.16.1.3 (en1 172.16.1.3): S set, 40 headers + 0 data bytes
```

```
len=46 ip=172.16.1.3 ttl=128 DF id=4802 sport=134 flags=RA seq=0 win=0 rtt=0.6 ms
len=46 ip=172.16.1.3 ttl=128 DF id=4803 sport=135 flags=SA seq=1 win=8192 rtt=0.8 ms
len=46 ip=172.16.1.3 ttl=128 DF id=4804 sport=136 flags=RA seq=2 win=0 rtt=0.8 ms
len=46 ip=172.16.1.3 ttl=128 DF id=4805 sport=137 flags=RA seq=3 win=0 rtt=0.9 ms
len=46 ip=172.16.1.3 ttl=128 DF id=4806 sport=138 flags=RA seq=4 win=0 rtt=0.8 ms
len=46 ip=172.16.1.3 ttl=128 DF id=4807 sport=139 flags=SA seq=5 win=8192
rtt=0.8 ms
len=46 ip=172.16.1.3 ttl=128 DF id=4808 sport=140 flags=RA seq=6 win=0 rtt=0.8 ms
....Truncated for brevity
```

Some useful Hping commands are included in Table 4-2.

Table 4-2 *Useful Hping2 Commands*

hping2 Command Example	Description
hping2 172.16.1.4 –p 80	Sends a TCP Null packet to port 80 on host 172.16.1.4. Most systems respond with a Reset/Ack flag if they are up and not firewalled.
hping2 192.168.1.4 –p 80 –S	Sends a SYN connect packet to host 192.168.1.4 at port 80. If the port is open, you will see a SYN/ACK response.
hping2 172.16.1.10 –S –p ++22	Sends a SYN connect packet to host 172.16.1.10 port 22 and increments the port number by 1 after each packet sent. Open ports respond with SA flags and closed ports respond with RA flags. It is useful for mapping ports sequentially.

Vulnerability Assessment Tools

There are many vulnerability assessment tools available today, from commercial applications to well-known open source tools. A vulnerability scanner's purpose is to map known vulnerabilities in products and present a report of potential vulnerabilities. This type of tool is great for automating the assessment of multiple hosts and usually provides nice severity categorization and output for reports. Obviously, you need to be careful when performing vulnerability tests on business systems because some of the assessment mechanisms these tools use to find vulnerabilities can crash services or cause an outage. Auditors should have a plan in place for restoring service in the event of a problem and perform testing outside of peak utilization times. Taking down the accounting server in the middle of processing payroll will not win you any friends and could be a career-limiting move. The following sections discuss vulnerability assessment tools that are good examples of the types of applications auditors can use to find control weaknesses.

Nessus

Nessus is a popular vulnerability scanner that looks for known vulnerabilities in operating systems, networking gear, and applications. Currently at version 4, Nesus has expanded its functionality significantly since it was introduced as an open source project more than

10 years ago. With the release of Version 4, Nessus has become a closed source product owned by Tennable Network Security. While the scanner is still free for home use to scan your personal devices, if you use it in any other capacity outside of the home, a professional feed license is required. The professional feed provides access to the latest updates and advanced features such as compliance checks (PCI NIST or CIS), SCAP protocol support, the ability to load it as virtual appliance, and product support from Tenable. The yearly professional license fee for Nessus is around $1,200.

Nessus is only as good as its latest vulnerability database update so it is imperative that you keep it up to date. If your organization conducts vulnerability assessments on a regular basis, opting for the commercial plugin feed adds support and access to the latest updates (often many times a day). The free plugin feed lags the commercial by seven days and does not include the auditing plugins that can be used to look for policy violations and specific types of data that don't belong on an end users' systems (such as credit card information).

Nessus is available for Windows, Linux, and Mac OS X. The only differences between the versions are cosmetic for the most part, but network-scanning performance is better on Linux-based systems. A well-written installation guide and videos are available on Tennable's website. These walk you through the process for getting Nessus up and running on your operating system.

Scanning a system with Nessus is straightforward and doesn't require a whole lot of effort to do. The first thing to do after logging in to the web interface for Nessus is configure the policies you will use to assess the network. This section is where you configure scanning preferences and the plugins that you assess the network against. Plugins are at the heart of the Nessus engine and provide the assessment intelligence used to find vulnerabilities and compliance violations. Thousands of plugins can be used during a scan, but it is recommended you enable only plugins for the devices you are assessing to greatly speed up the process. If you scan routers and switches, it doesn't make sense to turn on nonapplicable plugins like AIX security checks (unless you truly like watching the digital equivalent of paint drying).

Optionally, you can input login credentials and SNMP strings for databases and windows domain credentials to get a more thorough scan of operating system files and networking equipment settings. Figure 4-1 shows the plugin selection process used to configure scanning policies.

After scanning policies have been configured, select the device IP addresses that will be assessed. To start a scan, simply provide target addresses to scan, and then the scan policy that you want to use. You can select individual IPs, entire subnets, or you can import a text file with all of the addresses for the entire organization. After your targets are selected, select launch scan and Nessus will start its vulnerability analysis. Figure 4-2 shows the scan selection and launch process.

After the scan has been launched, Nessus performs all of the hard work gathering vulnerability information in the background. Depending on the complexity and depth of your scan, it can take a few minutes or a number of hours. After Nessus has finished, you will have a nice list of items it discovered that you can browse by severity level. Nessus ranks

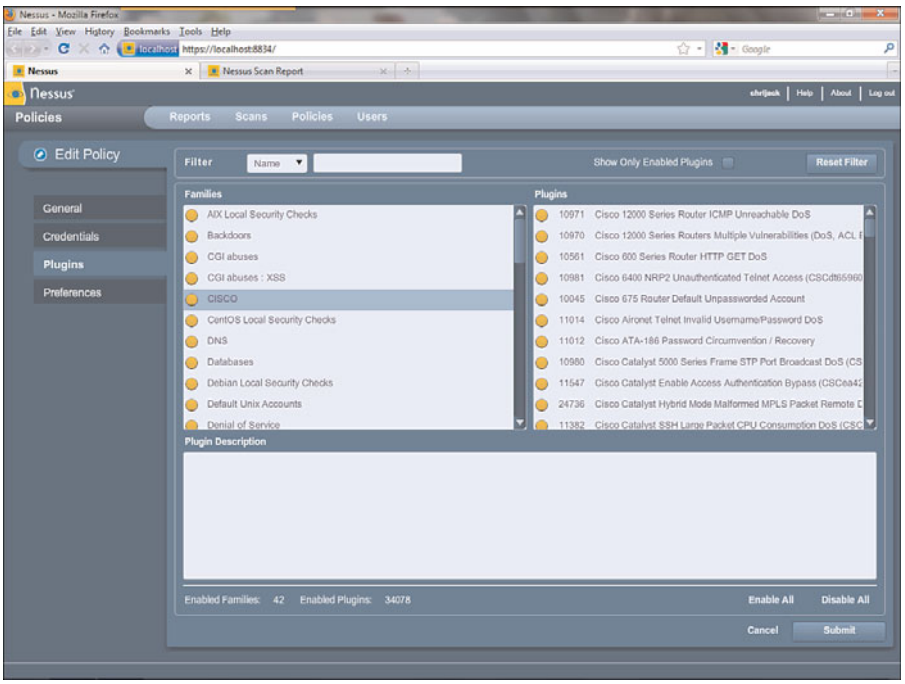

Figure 4-1 *Selecting Plugins in Nessus*

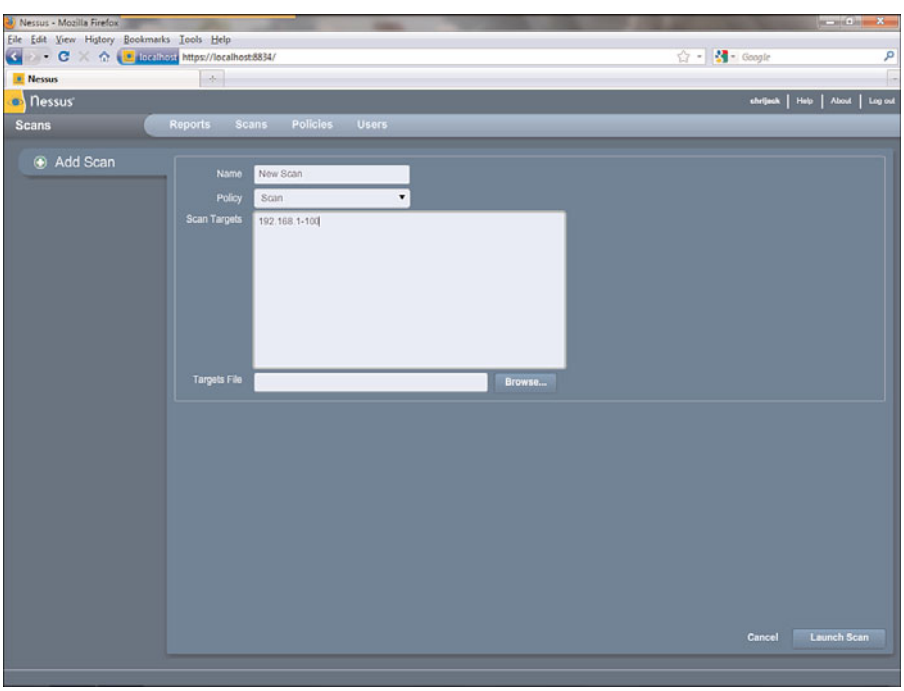

Figure 4-2 *Starting a Scan with Nessus*

vulnerabilities by severity using a high, medium, and low scale. Low severity is most commonly found and usually represents difficult-to-exploit weaknesses, information disclosure, or other potential security issues to be aware of that are not cause for alarm. Medium and high levels are the ones to be most concerned with and represent major vulnerabilities with known exploits that should be patched immediately. Figure 4-3 shows a Nessus scan summary with severity ranking of vulnerabilities found.

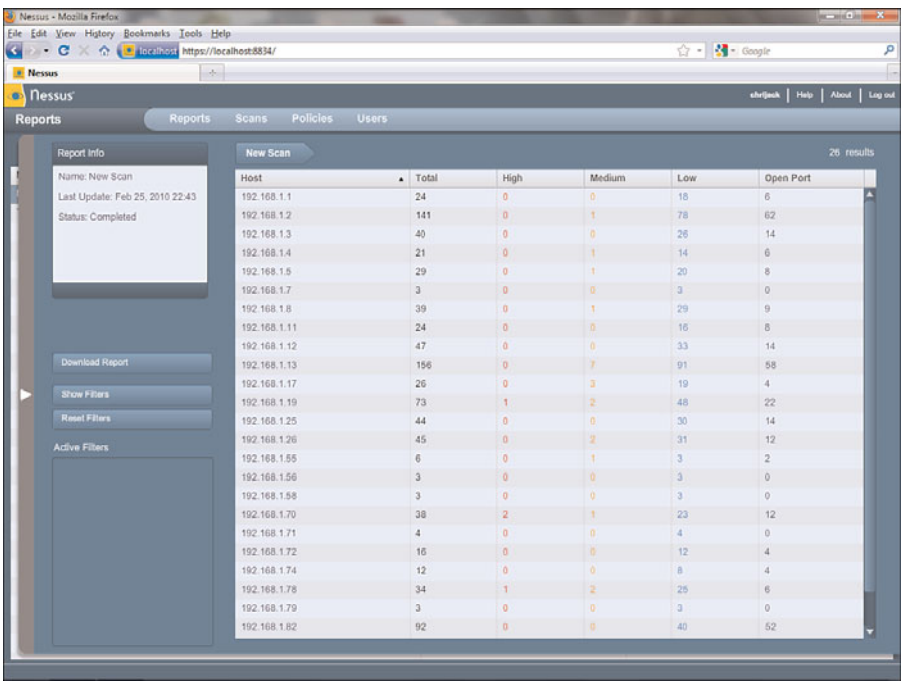

Figure 4-3 *Nessus Scan Vulnerability Ranking*

Detailed explanations of each vulnerability can be seen by clicking on the vulnerability and reviewing the informative description provided. There are also recommended solutions to address the problem and links to technical documents that analyze the vulnerability to a greater degree. Common Vulnerability Scoring System (CVSS) ranking is also applied to each vulnerability as a standard way to categorize the vulnerability. The complete report can be downloaded in a wide range of formats to incorporate the vulnerability information into an auditor's report. Figure 4-4 shows the detailed view of a medium-ranked vulnerability identified during scanning.

While basic Nessus scans are relatively simple, there are numerous advanced configuration options that serious auditors must become familiar with to get the most value out of their vulnerability scans. Auditors should not just launch Nessus against the entire organization's address range without a plan and expect to get anything of significant value. These types of shotgun approaches can cause a lot of trouble, especially because some of the plugins are potentially disruptive to servers and networking gear. There's nothing like taking down the company database or WAN links to win friends and influence management's opinion of your value to the organization.

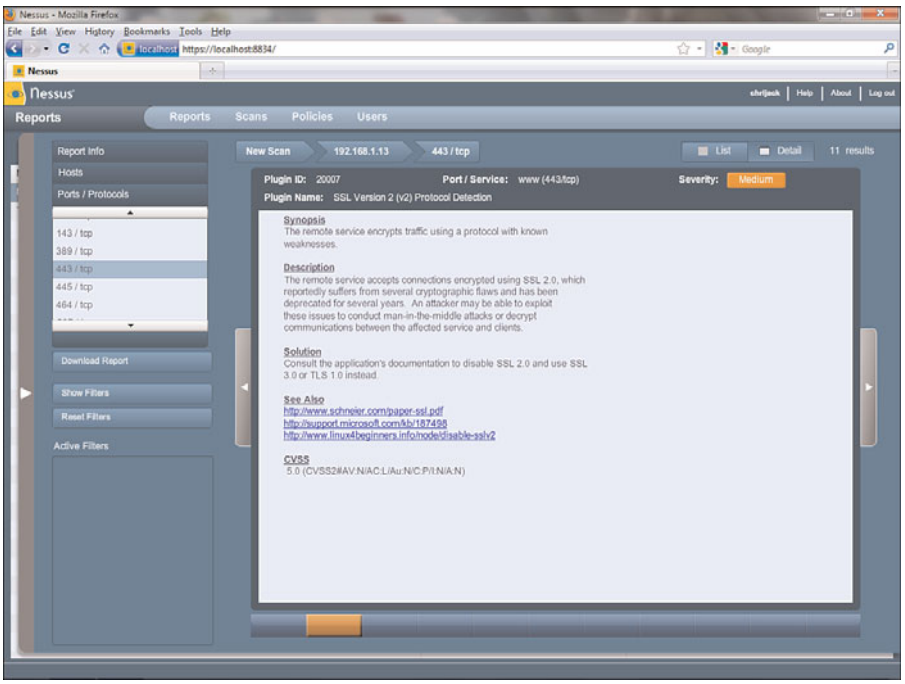

Figure 4-4 *Detailed Vulnerability Analysis*

For more information on using Nessus, the book *Nessus Security Auditing*, written by Mark Carey, is a great reference that can help an auditor learn the nuances of using Nessus. Check out the video demos on Tennable's website to see the product in action: http://www.tenablesecurity.com/demos/index.php?view=demo_videos.

RedSeal SRM

RedSeal Security Risk Manager (SRM) is a commercial risk management and threat identification application that eases the burden of analyzing a network to find vulnerabilities in configurations and visualizes the severity of what could happen if network security controls are compromised. The power of this application is that it enables an auditor to identify, prioritize, and report on the risk an organization faces at every point in the network. SRM builds a model of the network by importing configurations from network devices, vulnerability data from scanners, and the applications that are present. It performs Network Configuration Checks (NCC) that compare device configurations against standards and that identify vulnerabilities leveraging the National Vulnerability Database hosted by NIST. The NCCs ferret out any misconfiguration in access lists and identify unneeded services and potential policy violations. SRM also analyzes network configurations for compliance with corporate policy and PCI standards. These checks are continuously updated in the form of RedSeal's Threat Reference Library (TRL) files, which are imported into the application.

SRM comes in two flavors: an appliance version that you can install in a network and use as a dedicated risk analysis tool or a software-only install that can be loaded on a Windows laptop, desktop, or server that meets the minimum hardware requirements. The architecture of both versions is client-server, where interaction with the application requires loading a Java-based client.

After it is installed, SRM needs to be fed data about your network. You can either import the configuration files from your devices and vulnerability scan information directly to the application, or you can configure it to poll your devices and retrieve configuration data on a periodic basis. The ability to import the data "offline" without having to interact with the remote devices directly is a benefit for auditors and organizations that don't want to install the product and leave it running all of the time or would prefer a portable risk-management solution.

After you have imported your configuration files and vulnerability assessment information, you can begin modeling your networks security posture. Launching the client brings up the SRM dashboard shown in Figure 4-5, which gives the user a quick glance at the current risks identified through a simple graphical representation that shows best practice violations, warning, and a pass/fail assessment of network policy.

Figure 4-5 *SRM Home Tab*

The Maps and views tab enables an auditor to examine the network topology for access vulnerabilities by simply clicking on any one of the network devices represented on the map. The detail viewer at the bottom of the screen shows where packets generated from computers behind the device selected would be able to reach on the network. When an auditor assesses policy compliance, this one feature can reduce the amount of work the auditor has to do to assess access lists and other security controls in the network. This network path exploration function can easily show what types of traffic are allowed between segments and what threats different areas of the network pose to critical services. Figure 4-6 shows what parts of the network are accessible by Internet users and the protocols that are allowed through.

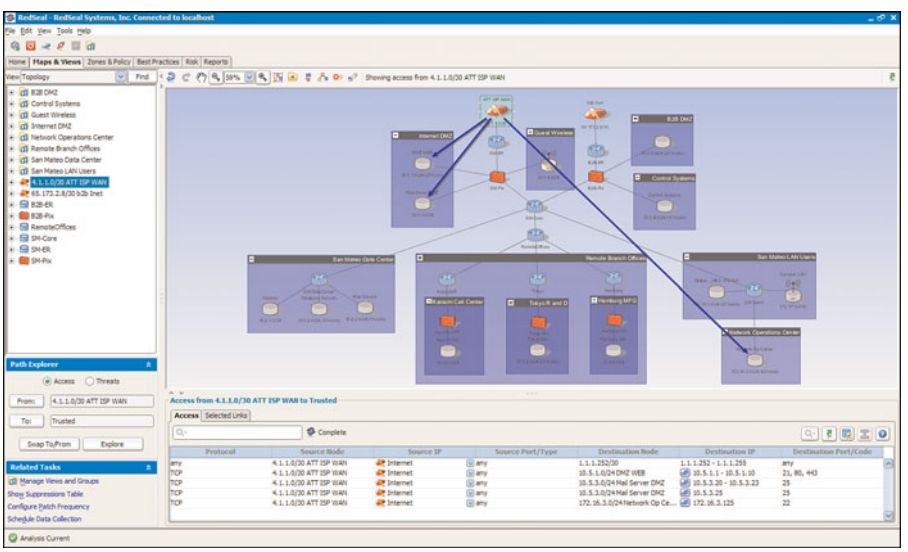

Figure 4-6 *SRM Maps & Views*

The Zones and Policy tab gives the auditor a compliance view of the network that assesses topology against corporate policy and regulatory requirements. The SRM has built-in rules for PCI DSS standards and the capability to add custom business policies that can be used for analysis of the network. Figure 4-7 shows the Zones and Policy tab and a PCI compliance assessment.

SRM can also automatically generate a PCI compliance report that can be used for ensuring that the appropriate controls are in place to meet the PCI DSS standard. Figure 4-8 shows a sample PCI report.

Configuration comparison of network devices against NIST security best practices is accomplished from the Best Practices tab. This is a quick way to identify misconfigured devices that represent poor security implementation. Figure 4-9 shows best practice configuration compliance failures found by SRM.

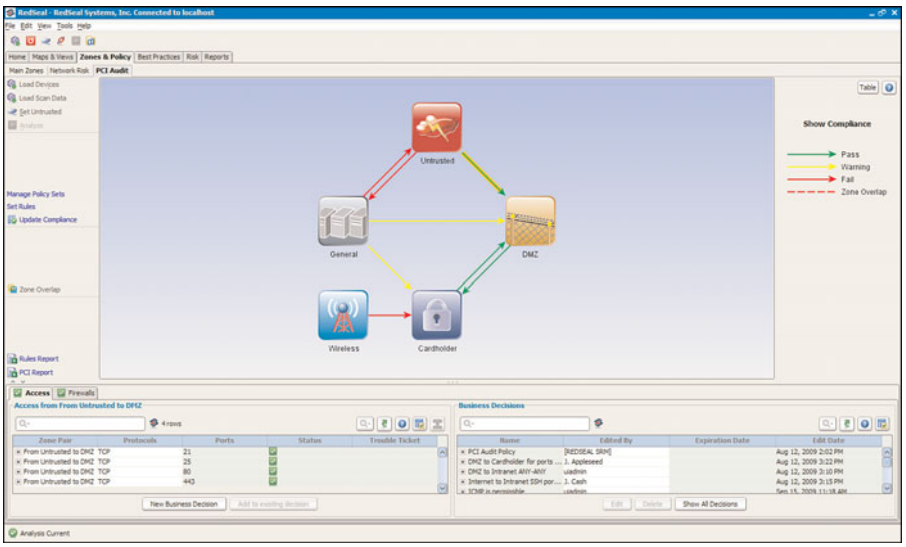

Figure 4-7 *SRM Zones and Policies*

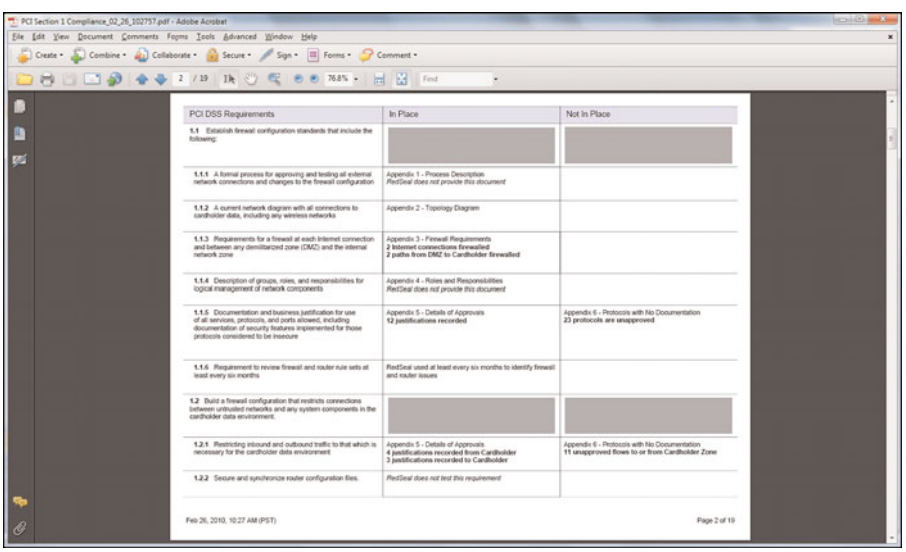

Figure 4-8 *SRM PCI Report*

Selecting the Risk tab takes you to the risk map, as shown in Figure 4-10, which shows risk in a graphical display by protocol, host, vulnerability, and mitigation priority. You can also export the data from this screen to a jpeg or as a text file for inclusion in a report.

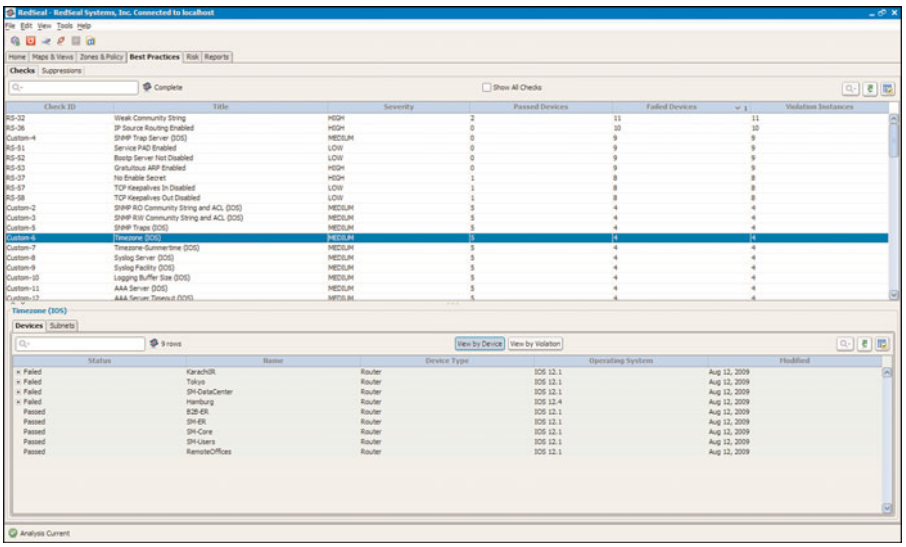

Figure 4-9 *SRM Best Practices Tab*

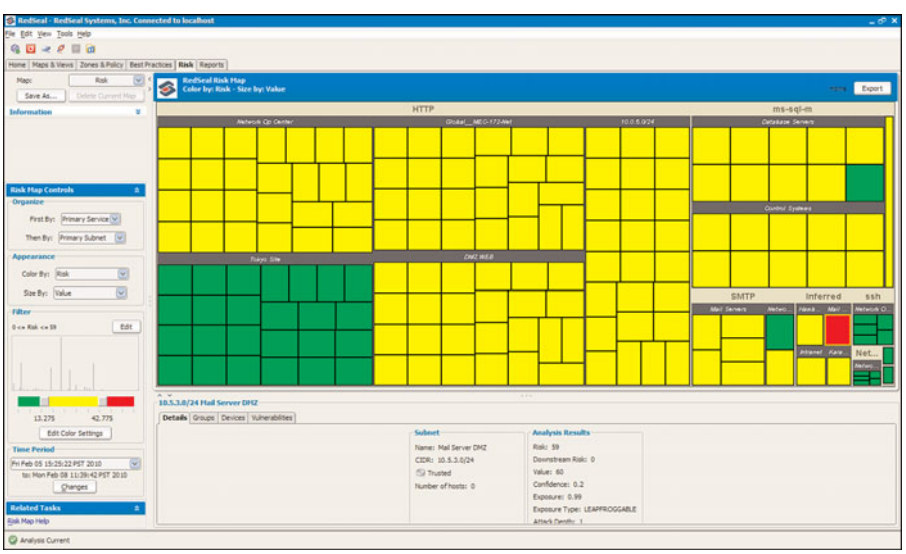

Figure 4-10 *SRM Risk Tab*

The last tab is the Reporting tab. It houses the various built-in reports that SRM provides. The reports can be run on the fly and saved to PDF for archiving. Figure 4-11 shows a consolidated security posture report that provides an overview of key findings. Running historical reports can also be helpful to show how risk is reduced over time as identified risks are mitigated. Many organizations use this information as a performance indicator for the success of their security programs.

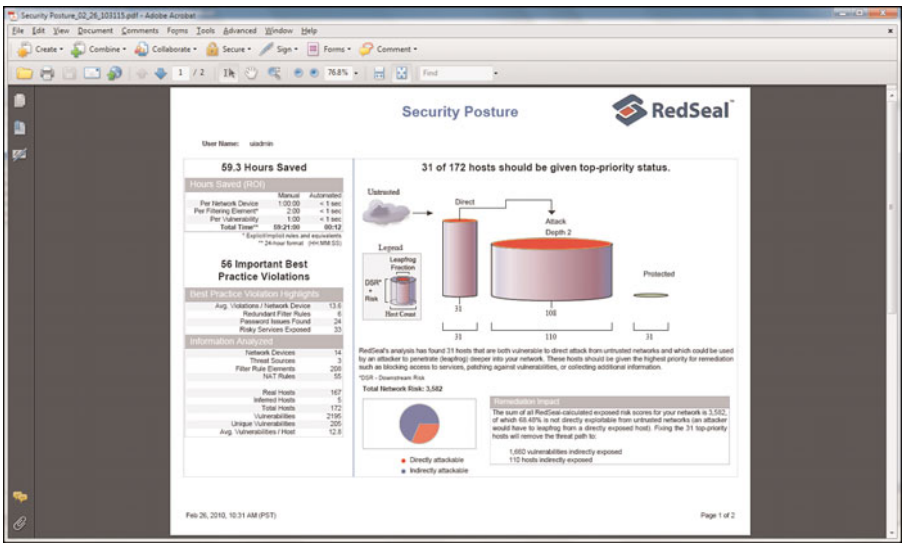

Figure 4-11 *SRM Reporting Tab*

RedSeal Security Risk Manager is a useful tool for visualizing and reporting on risk. Auditors can use it to aide in identifying whether a network is configured according to best practices, but also as a means to interpret business risk by assigning asset values and automatically quantifying the risk. Most auditors use a number of discrete tools that pull portions of this data, but having the ability to identify potential vulnerabilities and then extrapolate downstream attack potential is a compelling aspect of this product. For example, you may wonder whether a web server can be compromised and how much access the current configuration affords that web server to the internal network. Simply click on the Threats To tab and see visually what could potentially happen. Threat modeling is a powerful way to increase the security posture of the network.

Some of the other uses for SRM are:

- Prioritizing what host or devices to remediate first based on the overall risk and downstream threat to the organization
- Modeling a potential perimeter breach to determine what types of compensating technologies or controls need to be in place to reduce the risk of leapfrogging from one system to another
- As a measuring tool for management to correlate the changes in risk over time and as systems are remediated
- As new vulnerabilities are identified in applications, quickly modeling the impact of those vulnerabilities to the network as a whole

- As new services or business-to-business connections are brought online, modeling the risk to connected systems
- The ability to conduct a best-practices audit per device with the click of a button

Packet Capture Tools

Validation and testing of security controls are the most important aspects of conducting an audit. Auditors shouldn't just assume a firewall or IPS will enforce policy; they must test it and gather evidence abour how well those controls do their jobs. Packet capture tools are familiar to anyone who has had to troubleshoot a challenging network redesign or configuration. Packet capture tools are also extremely valuable when testing firewall rules, IPS signatures, and practically any other scenario where you need to see exactly what is going across the wire. Tcpdump and Wireshark are two free tools that should be in every auditor's repertoire.

Tcpdump

Tcpdump is a free packet capture program that operates as a simple command-line based "sniffer". It has been compiled for practically every operating system and leverages the UNIX Libpcap library (Winpcap on Windows) to copy traffic from the wire and display it on the screen or save it to a file. This simple packet sniffer provides a detailed view into the actual bits and bytes flowing on a network. Tcpdump is a simple application that doesn't have a graphical interface that abstracts the details of the packet capture process to automatically detect problems. It is left to the auditor to use his knowledge and experience to identify anomalies or issues. That doesn't mean that Tcpdump doesn't decode traffic; it just doesn't perform higher-level interpretation like Wireshark.

The other benefit of Tcpdump is that it can be used to grab the raw communications off of the wire in a format that a slew of other analysis tools can use. Tcpdump data files can be used as input into Snort, PDF, Wireshark, and may other packet-analysis applications. Tcpdump's capability to load on virtually any computing platform provides a portability that makes it the de facto standard for security testing.

Tcpdump is an easy tool to get started using. Simply open a command prompt, type in the command Tcpdump, and it happily starts displaying all of the packets seen by the first interface it finds on the machine. To be more specific about the interface you use (wireless or wired), you can type:

```
tcpdump -D
1.en0
2.fw0
3.en1
4.lo0
```

Tcpdump lists the interfaces available on your computer so that you can then select by number which one you want to use. This is especially useful on the Windows version (Windump) because Windows stores device information in the registry and assigns a cryptic address to your interfaces. After you have the appropriate interface, in this case Ethernet0 (en0), you can begin capturing traffic by issuing the command tcpdump –i 1 (or tcpdump –I en0):

```
tcpdump -i 1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listing on en0, link-type EN10MB (Ethernet), capture size 68 bytes
17:16:15.684181 arp who-has dhcp-10-90-9-126.cisco.com tell dhcp-10-90-9-
126.cisco.com
17:16:15.746744 00:1a:a1:a7:8c:d9 (oui Unknown) > 01:00:0c:cc:cc:cd (oui Unknown)
 SNAP Unnumbered, ui, Flags [Command], length 50
```

Using the default capture parameters, Tcpdump captures only the first 68 bytes of any packet it sees and will not decode any packets. This mode is useful for a cursory glance of traffic data, but doesn't provide the level of detail necessary for testing security. To increase the amount of data captured, you can modify the snaplen (snapshot length) with the **–s** option. For any Ethernet segment, the max length is typically 1514, so issuing the command **tcpdump –s 1514** copies every bit of data your interface receives.

Not all data is interesting or necessary to see when testing devices. Tcpdump has a simple, yet powerful filtering system that can be employed to sort through all of the noise on the wire to get to the traffic you are looking for. There are four basic filter options to help fine-tune your search.

- **Net:** Display all traffic to/or from a selected network; for example:

  ```
  tcpdump net 172.16.1.0/24,tcpdump net 192.168.0.0/16
  ```

- **Host:** Display packets to/or from a single host; for example:

  ```
  tcpdump host 192.168.32.2
  ```

- **Protocol:** Select IP protocol to capture (TCP, UDO, or icmp); for example:

  ```
  tcpdump udp 172.16.23.2
  ```

- **Source/Destination port:** Display traffic from a specific port; for example:

  ```
  tcpdump dst port 80
  tcpdump src port 22
  ```

You can add advanced filtering logic by stringing together the basic filter options with AND, OR, and NOT to get exactly the traffic you want to see. For example, if you want to see all UDP traffic from a host with the IP address 10.2.3.1 with a source and destination port of 53 (DNS,) you would use:

```
tcpdump host 10.2.3.1 and udp dst port 53
```

Another example would be if you wanted to see any nonSSH traffic from a user's subnet to a firewall management address at 192.168.23.1.

```
tcpdump dst 192.168.23.1 and not tcp port 22
```

Beyond the simple filters, Tcpdump can also allow someone who understands how the TCP/IP headers are formed to specify combinations of bits to examine. This is done through advanced options that require you to know what bits equal what flags in the TCP headers. You can find a good reference for the TCP/IP headers and fields created by the SANS institute at http://www.sans.org/resources/tcpip.pdf.

If you want to display all of the TCP packets captured that have both a SYN and a FIN flag set in the same packet (obviously a crafted packet), you would need to have a Tcpdump key on the flag fields you were looking for, and it would help to consult a chart that shows the offset in the TCP header and the bits you wanted to test against.

|C|E|U|A|P|R|S|F|
|—————————-|
|0 0 0 0 0 0 1 1|
|—————————-|
|7 6 5 4 3 2 1 0|
$2^1 + 2^0=3$

This provides a binary representation of 3 to check for SYN and FIN being present in the TCP flags. Consulting the TCPIP table, you can see that the TCP flags start at hex offset 13, which gives you a filter that looks like the following:

```
tcpdump -i eth0 (tcp[13] & 0x03)=3
```

Filtering can be complex, and if you make a mistake with the filters when capturing, you can miss the data stream you are looking for. It is usually best to do a raw capture, write it to a file, and then run your filters and other tools on the captured data file. Doing this enables you to examine the traffic in many different ways.

Writing a Tcpdump data file named capture.dmp:

```
Tcpdump -s 1514 -w capture.dmp
```

Reading a Tcpdump data file named capture.dmp:

```
Tcpdump -s 1514 -r capture.dmp
```

Table 4-3 lists useful Tcpdump commands.

Table 4-3 *Useful Tcpdump Commands*

Tcpdump Command Example	Description
tcpdump –r file_name –s 1514 -vv	Read the capture file name with a snaplen of 1514 and decode of very verbose.
tcpdump –w file_name –s 1514 -e	Write capture to file_name with a snaplen of 1514.
tcpdump –I eth0 –s 1514 –vv -e	Capture packets from interface Ethernet 0, decode very verbose, and include Ethernet header information.
tcpdump host 10.2.3.1 and udp dst port 53	Capture packets from host 10.2.3.1 that are UDP going to port 53 (DNS).
tcpdump –i 3 (tcp[13] & 0x03)=3	Capture and display packets on interface 3 with SYN and FIN bits set in TCP header.

Wireshark/Tshark

For those looking for a more full-featured GUI-based sniffer, you would be hard pressed to find a better one than the open source project known as Wireshark. Wireshark started life as Ethereal, written by Gerald Combs in 1998. Due to a trademark issue with the name Ethereal being owned by his former employer, the project was renamed in 2006 to Wireshark. Wireshark has become one of the most widely used and arguably the best packet capture application available. Best of all, it is completely free to use and actively developed by a team of over 500 volunteers.

Wireshark operates very much like Tcpdump in that it captures live traffic from the wire, reads traffic from a captured file, and decodes hundreds of protocols. Where Tcpdump has a simpler decode mechanism, Wireshark supports vastly more protocols and has a protocol decode framework that allows for the creation of custom packet decoders in the form of plugins. The display capabilities and advanced features such as stream following and packet marking make it easy to see what you want very quickly.

The filtering capabilities in Wireshark also allow for highly granular display and capture filters that follow the Tcpdump filter creation syntax. So, if you know Tcpdump, you will feel at home using Wireshark. Of course, Wireshark also has its own more detailed filtering language that can use specific keywords to search for fields of interest that don't require you to figure out what the offset is and what bits are required.

Using Wireshark is simple. After launching the application, select an interface to capture on, select start, and you will see captured traffic streaming from your interface. If you select an option before start, you will be presented with a screen, as shown in Figure 4-12, that allows you to limit the types of traffic you want through capture filters and a slew of other settings to finetune Wireshark's behavior.

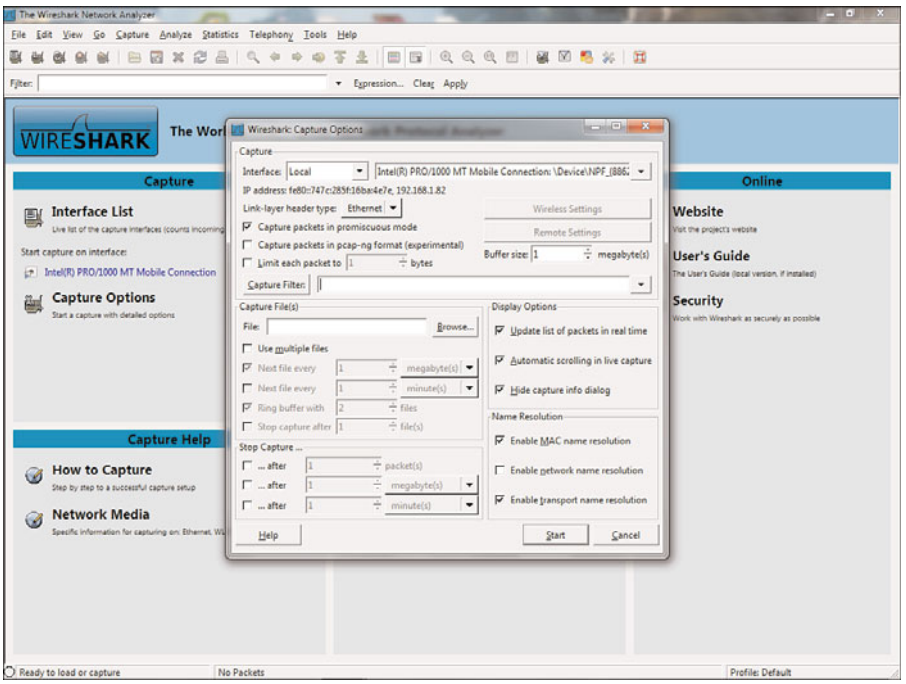

Figure 4-12 *Wireshark Capture Options*

The Wireshark GUI display provides a great way to visualize communications. All of the information you would see scrolling by on the command line can be viewed on screen. If you select a packet that interests you, you can drill down into the details of that packet by simply clicking the portion of the packet you want to see. In the example shown in Figure 4-13, we have selected an SSL version 3 packet. Wireshark decodes the packet and shows in HEX and Ascii what is in the payload. Looks like SSLv3 encryption does work!

One of the most valuable features of a packet-capture application for auditors is the capability to save and load captures. Wireshark supports many different file formats including commercial sniffing products and Tcpdump. By saving it in Tcpdump format, you ensure that the captures are able to be read by the widest variety of analysis tools. It is common for auditors to capture packets on a network and then use the capture files with other security tools for later analysis, such the open source intrusion detection tool Snort. Captures can also be replayed through the network interface of an auditor's laptop for security device testing purposes.

Tshark is the command-line equivalent of Wireshark, and uses the same major commands and options. Decodes provide the same level of detail as the GUI, but without the display flexibility or point and click operation. Tshark reads and writes capture files and is compatible with Tcpdump.

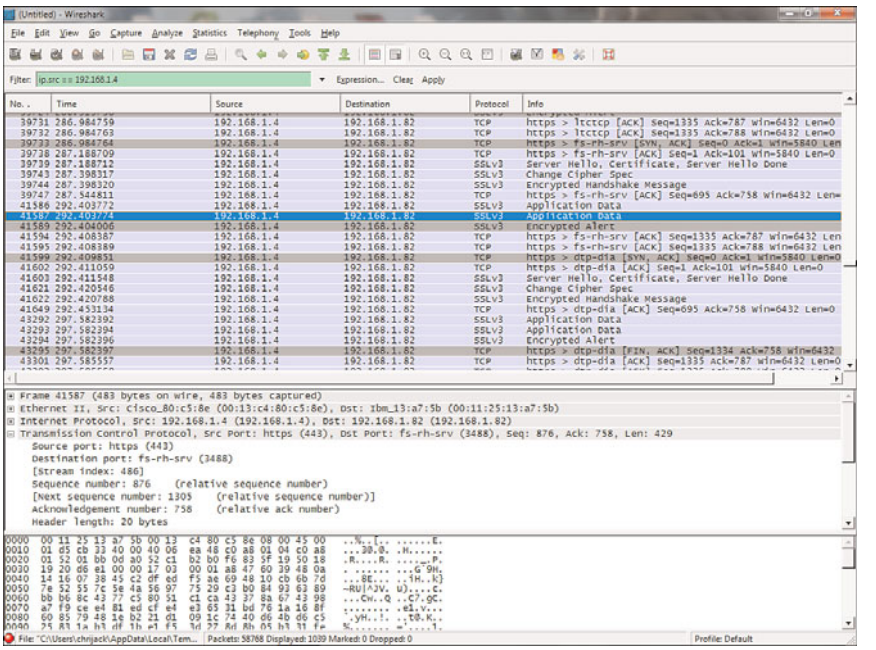

Figure 4-13 *Wireshark Protocol Decode*

Penetration Testing Tools

Auditors can leverage high-quality penetration testing tools to make auditing security controls significantly easier. Most professional penetration testers use a combination of general purpose exploit frameworks such as Core Impact and Metasploit in addition to their own custom scripts and applications. Not everyone in security is an uber hacker or has the time to build their own tools to test for exploitable services. These two applications are powerful and represent the best of the commercial and open source penetration testing tools available.

Core Impact

In the world of penetration tools, Core Impact is widely considered the best commercial product available. Developed by Core Security Technologies, this software package is a comprehensive penetration testing suite with the latest commercial grade exploits and a drag-and-drop graphical interface that can make anyone look like a security penetration testing pro. Writing exploit code and delivering it to a remote system is not a trivial task, but Core Impact makes it look easy. The framework Core has developed provides a modular platform to create custom exploits and making the tool appropriate for even the most advanced penetration test. Core Impact boasts a significant array of tools to test security controls. This product identifies vulnerabilities and automatically selects the appropriate exploits to gain control of remote systems (no way to have a false positive here). It does this without having to worry about tweaking and manipulating multiple tools and by including all of the functionality you need built right into the application itself.

Remotely exploitable network vulnerabilities are the Holy Grail of the security world, but Core Impact doesn't just rely on those types of exploits. It also provides client-side attacks designed to test how well the users follow security policy. You can embed Trojans into Excel files or other applications and email them to a user to see if they are following policy. If the user opens the suspicious file against policy, then Core Impact gains control of the computer and takes a screenshot of the desktop (suitable for framing!). There are also phishing capabilities that allow you to gather e-mail addresses and other information (useful for social engineering) off of the corporate website. This information can be used to target specific users and test their response, just like the bad guys do.

Core Impact also includes web application penetration testing features to test web security controls. Cross-site scripting and SQL injection attacks can be launched from the tool providing a complete penetration testing suite.

The Core Impact dashboard shown in Figure 4-14 is the first screen you see when launching this product and includes general information about the number and types of exploits available, and what operating systems are exploitable via the tool. There is also a link to update the exploits to download the latest attacks and modules.

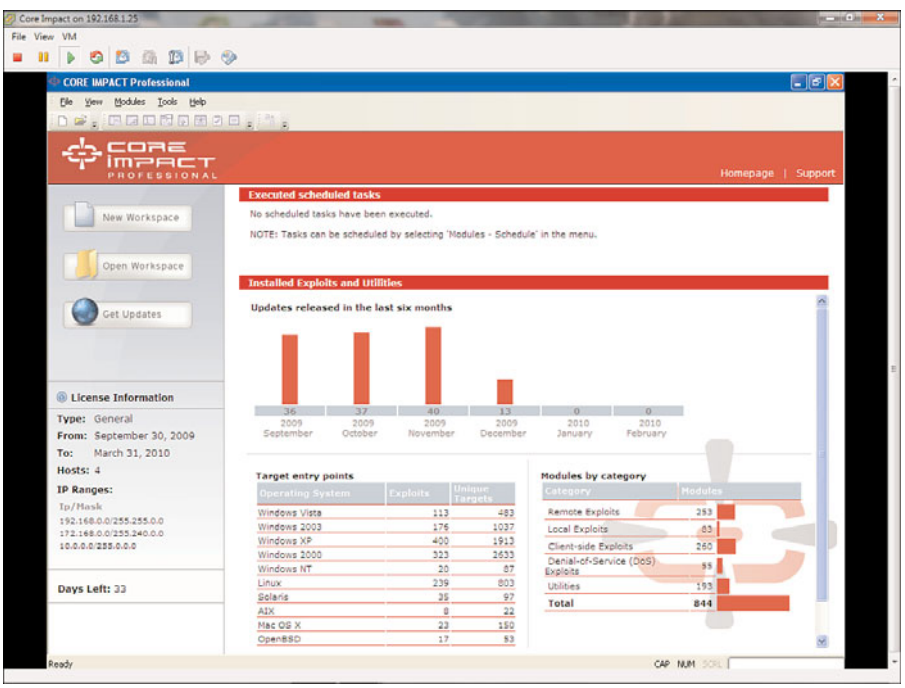

Figure 4-14 *Core Impact Dashboard*

In Core Impact, you can define workspaces to segment individual assessment engagements. Each workspace is password-protected and encrypted on the system to prevent sensitive data from falling into the wrong hands. These workspaces store a complete record of all of the activities and modules run during the penetration test.

After you have created a workspace or loaded an existing workspace, you are presented with the main console. This is where you decide what types of modules and exploits you are going to initiate. Core divides the exploits into the following categories:

- **Remote exploit:** These are attacks that can be initiated from a remote system usually in the form of a buffer overflow against a vulnerable service.
- **Local exploit:** These are privilege escalation attacks (gaining administrative access) that take advantage of weaknesses in applications or running processes on a system.
- **Client-side exploit:** Client-side exploits are designed to trick a user into executing code, surfing to a website, or launching malicious e-mail attachments. These types of exploits include phishing, Trojans, Keyloggers, and similar tools that target users.
- **Tools:** These are various components that can be used to assist with the exploitation process of a client, such as injecting an agent into a virtual machine.

Knowing what exploit to run against a system is the part that makes penetration testing a challenge. It requires playing detective to figure out what services are available and what versions, which usually necessitates using various tools such as Nmap and Nessus. Finding these vulnerabilities and matching them to the appropriate exploit is where Core Impact shines. Core Impact uses a wizard-based interface labeled RPT, which means Rapid Penetration Test; it follows a six-step penetration testing process for network and client tests. The web penetration testing wizard has a six-step process and all three are described in the following step lists.

The six-step network penetration test consists of:

Step 1. **Network information gathering:** Runs Nmap and Portscan against common services to identify operating systems and patch levels.

Step 2. **Network attack and penetration:** Uses the vulnerability information gathered in the first step to select possible exploits to use based on operating system type and services available. Sends real exploits and attempts to gain access to load an agent kit, which is a piece of code loaded into the memory of the remote system, enabling Core Impact to interact with the compromised computer.

Step 3. **Local information gathering:** Leverages the agent kit loaded to identify applications loaded, software patch levels, directory lists, and screen shots of the desktop. This can be used to prove that remote access was achieved.

Step 4. **Privilege escalation:** Some exploits work against user level processes only and do not give you complete control of the operating system at the kernel level. This wizard is used to upgrade access to root or administrative privileges by exploiting user level access processes.

Step 5. **Cleanup:** Removes all traces of the agent kits and cleans up logs on the compromised systems.

Step 6. **Network report generation:** Generates a report that details all of the activities the penetration tester engaged in and all of the vulnerabilities and exploits successfully used. This also provides an audit trail of the test.

The six-step client-side penetration test wizard consists of:

Step 1. **Client-side information gathering:** Searches websites, search engines, DNS, and WHOIS to harvest e-mail addresses to target specific users through social engineering. You can also import addresses from raw text files.

Step 2. **Client-side attack and penetration:** This wizard walks you through the process of crafting an e-mail to send to a user to try to entice them to load an attached Trojan or mail client exploit. You can also exploit web browsers by e-mailing links to exploits served by the Core Impact tools built in web server. The goal is to load an agent kit that will provide access to the system.

Step 3. **Local information gathering:** Same as with the network wizards, this wizard gathers information on the remote system.

Step 4. **Privilege escalation:** Uses subsequent vulnerabilities to gain admin or root level access to the system.

Step 5. **Cleanup:** Removes all agent kits and traces of access.

Step 6. **Client-side report generation:** Repots are created on which users "fell" for the attacks and what vulnerabilities were used and exploited.

The four-step Web Penetration test wizard consists of:

Step 1. **WebApps information gathering:** This process analyzes the website's structure and gathers information on the type of webserver software and code levels in use.

Step 2. **WebApps attack and penetration:** The Web Attack and Penetration Wizard sniffs out vulnerabilities in the web applications and attempts to exploit them. It performs cross-site scripting, SQL injection, and PHP attacks.

Step 3. **WebApps browser attack and penetration:** Cross-site scripting is used to exploit a user's web browser in this wizard. E-mail addresses are gathered for the target organization, and links are sent to get the user to click on and download an agent kit.

Step 4. **WebApps report generation:** Reports are generated for the web exploit process including all of the activities the penetration tester performed and which systems were compromised.

Figure 4-15 shows the Core Impact tool in action.

A remote computer at IP address 192.168.1.61 was compromised using a buffer overflow vulnerability in the Microsoft RPC service, and a Core Impact Agent was loaded in memory. After this occurs, the penetration tester has full control of the remote machine and can use the remote computer to attack other machines, sniff information off of the local

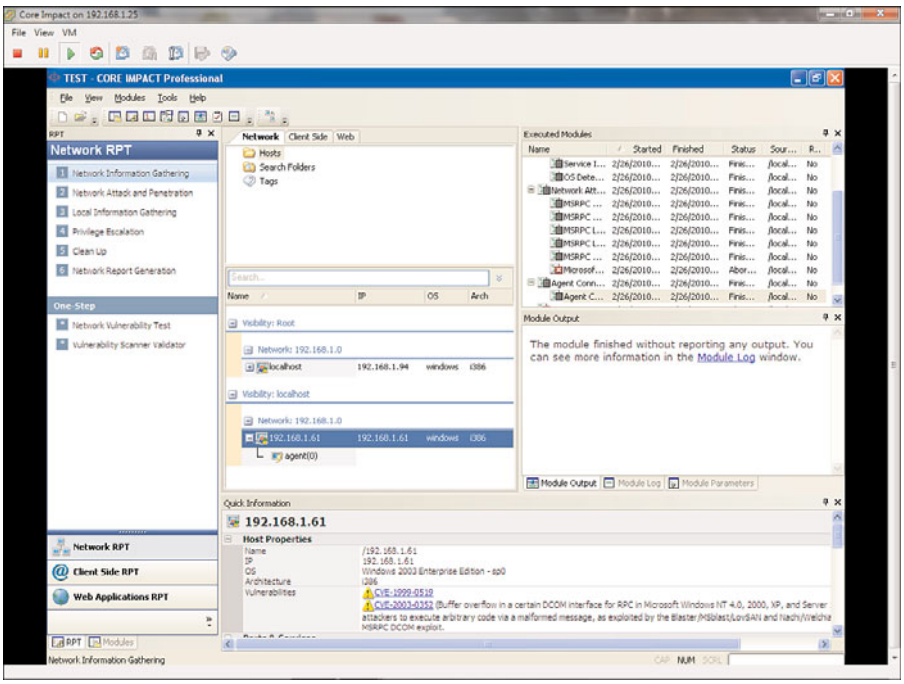

Figure 4-15 *Core Impact Vulnerability Exploit*

network, or a wide range of other attacks. Figure 4-16 shows a remote shell that was opened on the compromised computer, giving the auditor direct command-line access. As the old saying goes, "A picture is worth a thousand words."

Auditing requires the testing of controls and sometimes requires sending exploits to remote systems and testing the response of controls such as firewall, IPS, or HIPS products. This information can be exported into a variety of formats for reporting and correlating with vulnerability findings. With all of the advanced exploit techniques and reporting capabilities in Core Impact, it can be one of the best tools an auditor has in assessing security device capabilities and validating whether or not a vulnerability is actually exploitable.

Metasploit

The Metasploit project is responsible for providing the security community with one of the most important and useful security tools available today. Originally conceived and written by H.D. Moore in 2003 to assist with the development and testing of security vulnerabilities and exploits, the project has developed a life of its own through the contributions of many of the brightest security researchers today. The Metasploit Framework takes many of the aspects of security testing from reconnaissance, exploit development, payload packaging, and delivery of exploits to vulnerable systems and wraps them into a single application. The power of the framework comes from its open nature and extensibility. If you want to add a feature or integrate it into other tools, you can add support

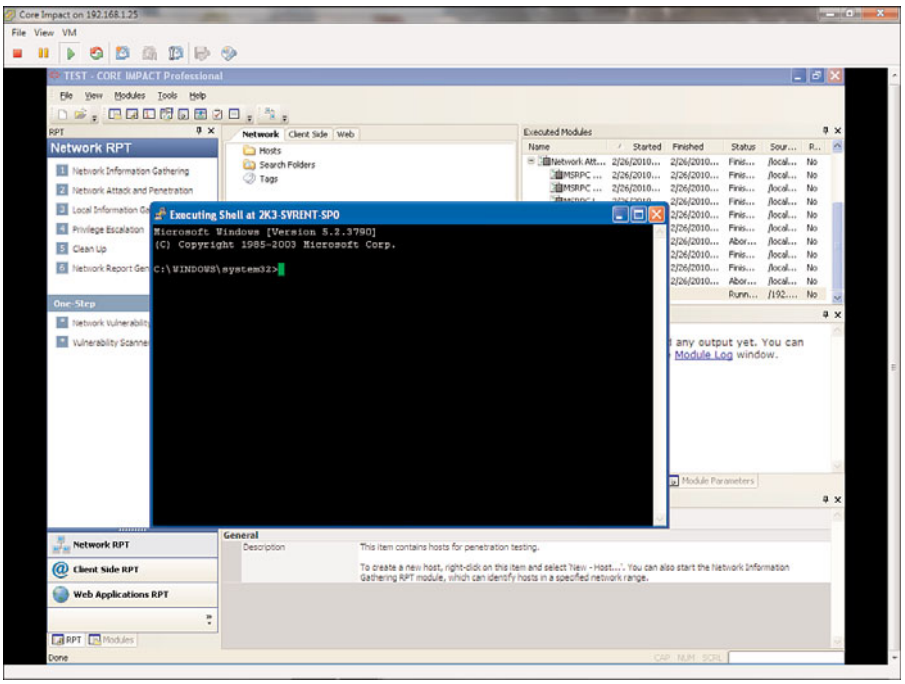

Figure 4-16 *Core Impact Opening a Remote Command Shell*

via new modules. Written in the Ruby programming language, Metasploit is available for all of the major operating systems: Windows, UNIX, Linux, and Mac OSX. The project is located at www.metasploit.com.

Unlike commercial products like Core Impact, there isn't the same level of polish or features designed for less experienced security professionals. There are no reporting capabilities or the simple wizard-based GUIs; this tool is designed for those security professionals who want to directly control every aspect of a penetration test. The current version 3.3 has improved dramatically and includes four choices for the user interface.

- **Msfconsole:** This is the primary console. It provides access to all of Metasploits exploits, payloads, and auxiliary modules through an intuitive command driven interface. Every portion of the interface has help features either through the command **help** or **–h**. You can easily find exploits and payloads by issuing the search command.
- **Msfcli:** This is a **-ine** interface executed from a UNIX or Windows command prompt that provides access to Metasploit. Designed to provide quick access to a known exploit or auxiliary module, it is also useful for scripting.
- **Msfweb:** MSFweb provides control of Metasploit through an interactive web interface. By default, it uses the built-in web brick web server and binds to the loopback address at port 55555. You can, however, select a real IP address and access the Metasploit from another computer's web browser. Firefox, Internet Explorer, and

Safari are all supported.

- **Msfgui:** In version 3.3, the Metasploit GUI has advanced considerably and is available for UNIX platforms (3.2 supports a GUI on Windows). The interface has integrated search functions and status and session connection information to exploited systems:
  - **Payloads:** Payloads provide the commands to add users, execute commands, copy files, launch a VNC session, or just initiate a command shell back to the attacker. Payloads are what are sent with the exploit to provide the attack a mechanism to interact with the exploited system. These payloads are available for a wide number of operating systems, including BSD, UNIX, Windows, OSX, Solaris, and PHP web environments.
  - **Exploits:** Exploits are the code and commands that Metasploit uses to gain access. Many of these are in the form of buffer overflows that enable the remote attacker to execute payloads (arbitrary software). There are hundreds of exploits for Windows, UNIX, and even a few for the Apple iPhone.
- **Encoders:** Buffer overflows are targeted against specific processor types and architectures. Metasploit's encoders enable the user to make the payloads readable for PowerPC, SParc, and X86 processors. You can also modify the encoder settings to change the payload to try to evade IDS and IPS signatures.
- **NOPS:** NOPS (no operation) are used when added to payloads in a buffer overflow because the exact location in memory of where the overflow occurs is not always known. NOPS allows there to be a margin of error in the coding of an exploit, because when the processor sees a NOP, it ignores it and moves on to the next bit of code in the buffer. After it reaches the payload, it executes the hacker's commands. Most IDS/IPS trigger on a string of NOPS (known as a NOP sled). These modules in Metasploit allow for the customization of the NOP sled to try to evade IDS/IPS systems.
- **Auxiliary:** The Auxiliary modules in Metasploit provide many useful tools including wireless attacks, denial of service, reconnaissance scanners, and SIP VoIP attacks.

After you install Metasploit, you have a choice about how you interact with it by picking the appropriate interface. Using Metasploit from the interactive console allows direct access to the most powerful components of the framework. However, if you want a point-and-click experience, the new GUI or web interface is available. Figure 4-17 shows the Metasploit console and commands displayed for help.

To launch the GUI, enter the command **msfgui** or click the icon under the Metasploit installation menu. The interface loads and you are presented with a simple interface that lists the different modules and a session list and module output window. Figure 4-18 shows the GUI under Linux.

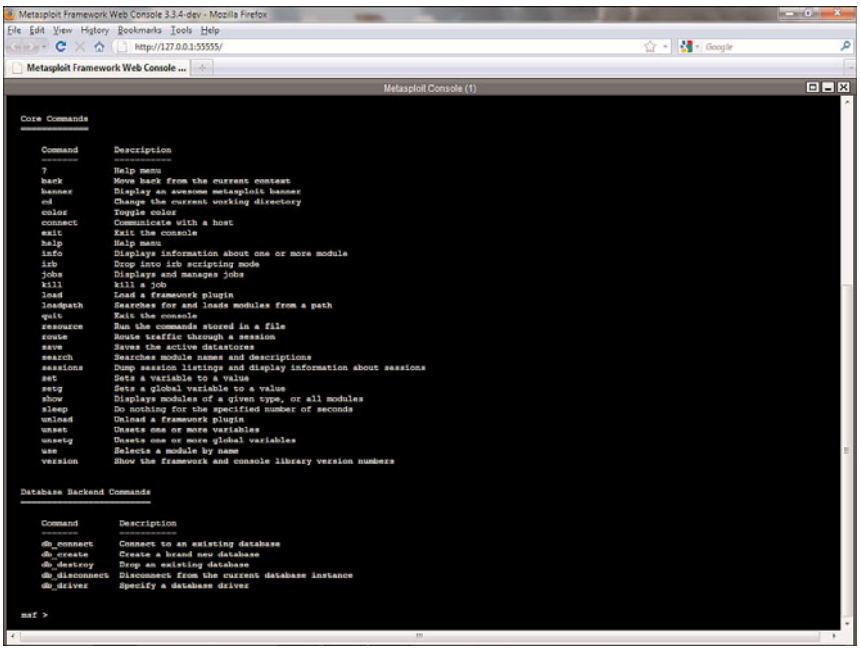

Figure 4-17 *Metasploit Console and Commands*

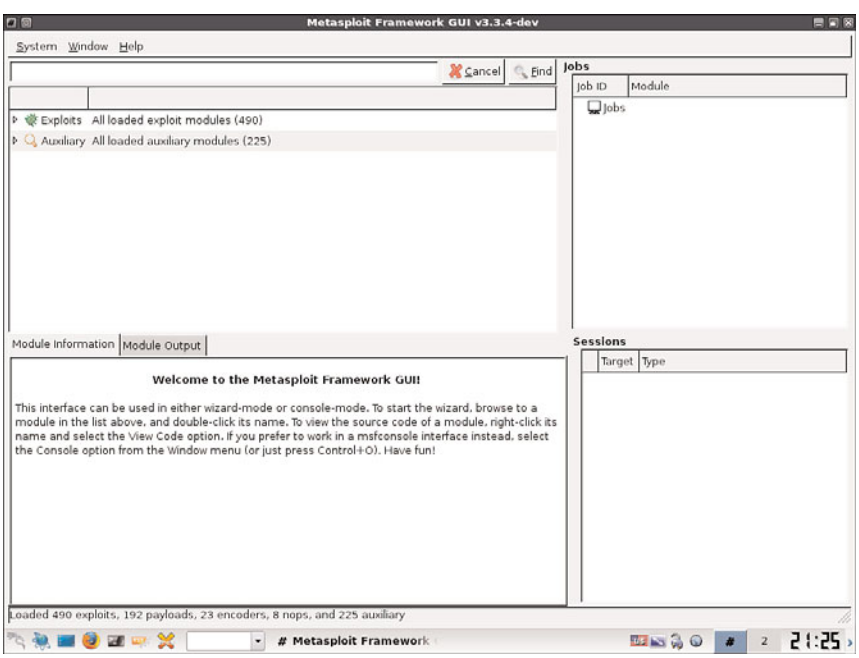

Figure 4-18 *Metasploit GUI*

In this example, the remote system is a Windows 2003 Server we are attempting to exploit. The easiest way to find exploits for a particular operating system is to use the built-in search function of the GUI. Entering **windows 2003** in the search window displays a list of modules where Windows 2003 is listed in the description of the module as being applicable. Scrolling through the list and selecting the RPC DCOM buffer overflow that gave us worms like Blaster, the interface presents a four-step process for configuring the exploit, which is illustrated in Figure 4-19.

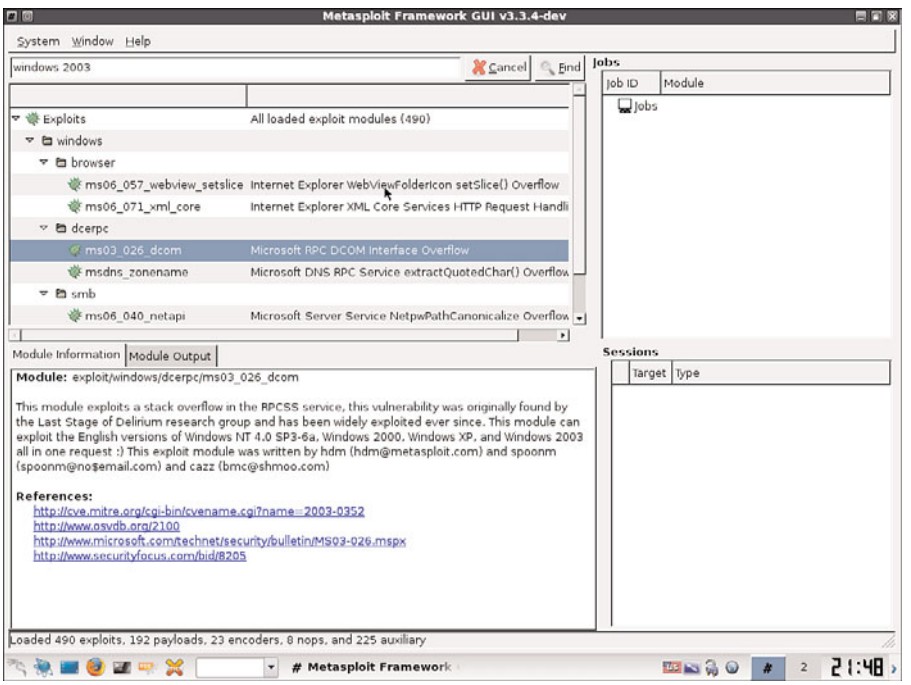

Figure 4-19 *Selecting an Exploit for Metasploit*

First, define the payload that you would like to use to execute code on the remote machine. Metasploit provides a number of methods to interact with the remote system after it is compromised. Grabbing a command shell or even using the Meterpreter to launch attacks on other systems through this compromised machine is possible. One of the slickest payloads available injects a VNC process into memory and gains access through remote control of the machine. Figure 4-20 shows the selection of a payload that will create a VNC terminal session with the target.

Next, enter configuration options and runtime parameters for executing the attack. LHOST is the local IP address you will use to connect back to, and RHOST is the target's IP address. Everything else is set as default. Figure 4-21 shows how the attack is configured.

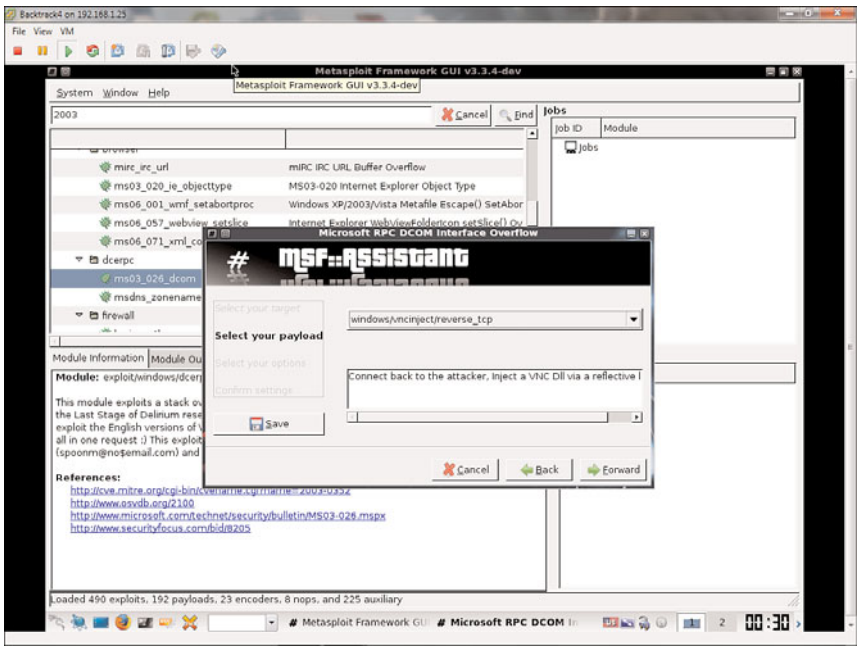

Figure 4-20 *Selecting VNC dll Injection*

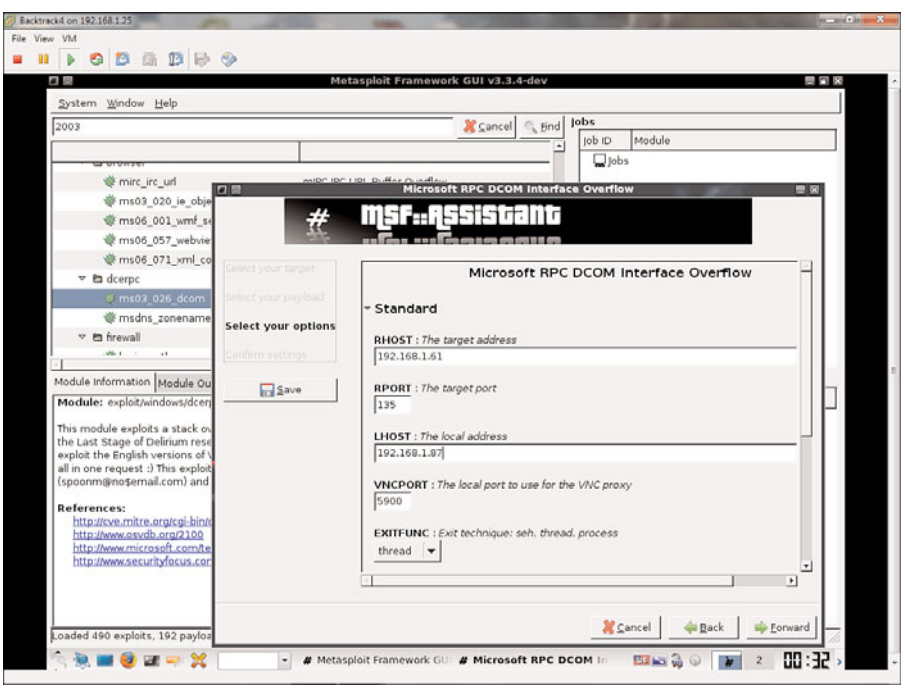

Figure 4-21 *Configuring Metasploit Attack Parameters*

After selecting forward, you are presented with a screen that shows the selected options and your settings for the exploit. After you have approved the configuration, you can launch the exploit. Metasploit sends the buffer overflow and payload to the remote system and list a connection coming back from the exploited host. If the attack works, then VNC Viewer automatically loads and you have full control of the remote host. Figure 4-22 shows a VNC session that was created from the exploit sent to the Windows 2003 server. Metasploit is even kind enough to launch a "courtesy" command for you.

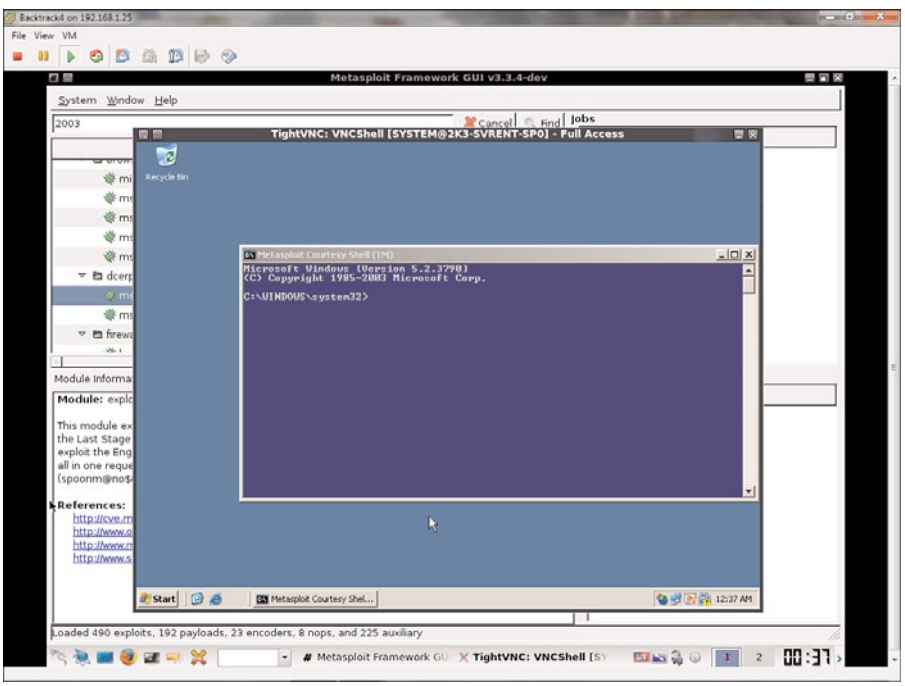

Figure 4-22 *VNC Session from Remote Computer*

Metasploit is a great tool for auditors, the price is right (as in free), and the capabilities are powerful. The biggest challenge in using Metasploit is the learning curve required for the average auditor with limited experience with host or network attacks. From an educational standpoint, Metasploit is a wonderful tool to hone your penetration-testing skills and enhance your understanding of vulnerabilities and how hackers exploit them. As a penetration-testing framework for research and development of new exploits, it is unmatched. If, however you are more interested in a commercial grade product with a vendor's technical support services and easy-to-use wizards with excellent reporting capabilities, tools such as Core Impact become a compelling choice.

BackTrack

BackTrack is a Linux live CD distribution built on Slackware Linux that doesn't require any installation and can be run from practically any PC with a CD ROM. You can also configure BackTrack to boot off of a USB memory stick making it an extremely portable, easily available security-testing environment. BackTrack4 is one of the most complete suites of security assessment tools ever assembled, saving security professionals countless hours of finding, installing, and compiling hundreds of different security applications. There are other security-focused distributions available, but none are as widely regarded and supported as BackTrack.

BackTrack is offered as a free distribution from www.remote-exploit.org and is available for download directly from the website or Bit-torrent network. Once downloaded, you can use it from a CD, USB memory stick, or load it into VmWare. The benefit of loading to a read/writeable format is obvious in that you can store settings, update packages, and customize the environment. Regardless of your preferred method of use, the tools included are extensive and are organized by the Open Source Security Testing Methodology. The categories are:

- **Information gathering:** DNS mapping, Whois, Finger, and mail scanning
- **Network mapping:** Port and services mapping, OS fingerprinting, and VPN discovery
- **Vulnerability identification:** Tools to identify service, SQL, VoIP, and HTTP vulnerabilities
- **Web application analysis:** Web application hacking tools for the frontend services (XSS, PHP) and the backend database (SQL injection)
- **Radio network analysis:** Wireless sniffers, scanners, and cracking tools
- **Penetration:** Tools to exploit vulnerabilities and compromise systems (Metasploit is the primary application.)
- **Privilege escalation:** LAN sniffers, password sniffers, and spoofing tools
- **Maintaining access:** Backdoors, rootkits, and tunneling applications for retaining access after exploiting
- **Digital forensics:** Disk editors, file system dump tools, and hex editors for recovering evidence from deleted and hidden files
- **Reverse engineering:** Malware analysis tools, application debug tools, and hex and assembly tools
- **Voice over IP:** VoIP cracking and recording tools
- **Miscellaneous:** Tools that don't fit in any other category that can assist with penetration testing

Summary

This chapter introduced security testing methodologies and some of the tools used to conduct those tests. It is not an exhaustive list of all potentially useful security testing tools, but should give a sampling of some of the most popular that any auditor can find useful. If you are interested in learning more about penetration testing or want to take a class with hands on practice, the SANS Institute offers a fantastic class called Security 560: Network Penetration Testing and Ethical Hacking.

In summary:

- Evaluating security controls requires testing three elements: people, process, and technology. If one area is weak, it can leave an organization vulnerable to attack.
- Penetration testing is a discipline that requires a structured and repeatable methodology. Without one, you are simply launching exploits and hoping to get in.
- Commercial tools such as Core Impact and open source tools such as Metasploit assist with testing security controls. Which one you choose depends on your budget, skill level, and desired reportability.
- The easiest way to get access to many of the tools discussed in this chapter is to download and launch Backtrack3. Not only does it save you many hours of setup, but it also gives you a powerful suite of tools with strong community support.

References in This Chapter

Security Testing Frameworks

Information Systems Security Assessment Framework, http://www.oissg.org/

Open Source Security Testing Methodology Manual, http://www.isecom.org/osstmm/

NIST 800-115: Technical Guide to Information Security Testing and Assessment, http://csrc.nist.gov/publications/nistpubs/800-115/SP800-115.pdf

Open Web Application Security Project, http://www.owasp.org/index.php/Main_Page

Security Testing Tools

NMAP, http://nmap.org

Hping, http://hping.org

Nessus, http://nessus.org

RedSeal SRM, http://www.redseal.net

TcpDump, http://sourceforge.net/projects/tcpdump/

Wireshark, http://www.wireshark.org/

Core Impact, http://www.coresecurity.com/

Metasploit Project, http://www.metasploit.com/

Backtrack, http://www.backtrack-linux.org/

Index

Numbers

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y-Z