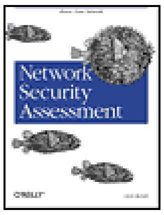
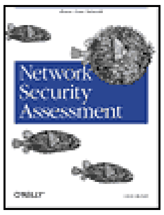


Network Security Assessment

By Chris McNab

Publisher: O'Reilly
Pub Date: March 2004
ISBN: 0-596-00611-X
Pages: 396
Slots: 1.0

Network Security Assessment offers an efficient testing model you can adopt, refine, and reuse to create proactive defensive strategies to protect your systems from the threats that are out there, as well as those still being developed. This thorough and insightful guide covers offensive technologies by grouping and analyzing them at a higher level--from both an offensive and defensive standpoint--helping administrators design and deploy networks that are immune to offensive exploits, tools, and scripts.



Network Security Assessment

By Chris McNab

Publisher: O'Reilly
Pub Date: March 2004
ISBN: 0-596-00611-X
Pages: 396
Slots: 1.0

PREV < Day Day Up > NEXT

Foreword

After managing the performance of over 20,000 infrastructure and applications penetration tests, I have come to realize the importance of technical testing and providing information security assurance.

This book accurately defines a pure technical assessment methodology, giving you the ability to gain a much deeper understanding of the threats, vulnerabilities, and exposures modern public networks face. The purpose for conducting the tens of thousands of penetration tests during my 20+ years working in information systems security was "to identify technical vulnerabilities in the tested system in order to correct the vulnerability or mitigate any risk posed by it." In my opinion, this is a clear, concise, and perfectly wrong reason to conduct penetration testing.

As you read this book, you will realize that vulnerabilities and exposures in most environments are due to poor system management, patches not installed in a timely fashion, weak password policy, poor access control, etc. Therefore, the principal reason and objective behind penetration testing should be to identify and correct the underlying systems management process failures that produced the vulnerability detected by the test. The most common of these systems management process failures exist in the following areas:

- System software configuration
- Applications software configuration
- Software maintenance
- User management and administration

Unfortunately, many IT security consultants provide detailed lists of specific test findings and never attempt the higher order analysis needed to answer the question of "why." This failure to identify and correct the underlying management cause of the test findings assures that, when the consultant returns to test the client after six months, a whole new set of findings will appear.

If you are an IT professional who is responsible for security, use this book to help you assess your networks; it is effectively a technical briefing of the tools and techniques that your enemies can use against your systems. If you are a consultant performing security assessment for a client, it is vital that you bear in mind the mismanagement reasons for the vulnerabilities, as discussed here.

Several years ago, my company conducted a series of penetration tests for a very large international client. The client was organized regionally; IT security policy was issued centrally and implemented regionally. We mapped the technical results to the following management categories:

OS configuration

PREV < Day Day Up > NEXT

About Bob Ayers

Bob Ayers is currently the Director for Critical Infrastructure Defense with a major IT company based in the United Kingdom. Previously, Bob worked for 29 years with the U.S. Department of Defense. His principal IT security assignments were with the Defense Intelligence Agency (DIA) where he served as the Chief of the DoD Intelligence Information System (DoDIIS). During this assignment, Bob developed and implemented new methodologies to ensure the security of over 40,000 computers processing highly classified intelligence information. Bob also founded the DoD computer emergency response capability, known as the Automated Systems Security Incident Support Team (ASSIST). Noticed for his work in DoDIIS, the U.S. Assistant Secretary of Defense (Command, Control, Communications, and Intelligence) selected Bob to create and manage a 155-person, $100-million-per-year DoD-wide program to improve all aspects of DoD IT security. Prior to leaving government service, Bob was the director of the U.S. DoD Defensive Information Warfare program.

Preface

It is never impossible for a hacker to break into a computer system, only improbable.

Network-based threats lie in wait around every corner in this information age. Even as I write this book, wireless networks are becoming a sore point for many companies and organizations that still don't understand how to secure their infrastructures. Networks are under siege from many different types of threat, including Internet-based hackers, worms, phone phreaks, and wireless assailants.

This book tackles one single area of information security in detail: that of undertaking IP-based network security assessment in a structured and logical way. The methodology presented in this book describes how a determined attacker will scour Internet-based networks in search of vulnerable components (from the network to the application level) and how you can perform exercises to assess your networks effectively. This book doesn't contain any information that isn't relevant to assuring the security of your IP networks; I leave listings of obscure techniques to behemoth 800-page "hacking" books.

Assessment is the first step any organization should take to start managing information risks correctly. My background is that of a teenage hacker turned professional security analyst, with a 100% success rate over the last five years in compromising the networks of financial services companies and multinational corporations. I have a lot of fun working in the security industry and feel that now is the time to start helping others by clearly defining an effective best practice network-assessment methodology.

By assessing your networks in the same way a determined attacker does, you can take a more proactive approach to risk management. Throughout this book, there are bulleted checklists of countermeasures to help you devise a clear technical strategy and fortify your environments at the network and application levels.

PREV < Day Day Up > NEXT

Recognized Assessment Standards

This book is written in line with the most important assessment standards, USA NSA IAM and UK CESG CHECK, which the United States and the United Kingdom use for government and critical national infrastructure testing and assurance.

NSA IAM

The United States National Security Agency (NSA) has provided an INFOSEC Assessment Methodology (IAM) framework to help consultants and security professionals outside the NSA provide assessment services to clients in line with a recognized standard. The NSA IAM homepage is http://www.nsa.gov/isso/iam/index.htm.

The IAM framework defines three levels of assessment related to testing of IP-based computer networks:

1. Assessment. This level involves discovering a cooperative high-level overview of the organization being assessed, including access to policies, procedures, and information flow. No hands-on network or system testing is undertaken at this level.
2. Evaluation. Evaluation is a hands-on cooperative process that involves testing with network scanning and penetration tools and the use of specific technical expertise.
3. Red Team. A Level 3 assessment is noncooperative and external to the target network, involving penetration testing to simulate the appropriate adversary. IAM assessment is nonintrusive, so within this framework, a Level 3 assessment involves full qualification of vulnerabilities.

This book covers only the technical network scanning and assessment techniques used within Levels 2 (Evaluation) and 3 (Red Team) of the IAM framework, since Level 1 assessment involves high-level cooperative gathering of information, such as security policies.

CESG CHECK

The Government Communications Headquarters (GCHQ) in the United Kingdom has an information assurance arm known as the Communications and Electronics Security Group (CESG). In the same way that the NSA IAM framework allows security consultants outside the NSA to provide assessment services, CESG operates a program known as CHECK to evaluate and accredit security testing teams within the United Kingdom to undertake government assessment work. The CESG CHECK homepage is accessible at http://www.cesg.gov.uk/site/check/index.cfm.

Unlike the NSA IAM, which covers many aspects of information security (including review of security policy, anti-virus, backups, and disaster recovery), CHECK squarely tackles the area of network security assessment. A second program is the CESG Listed Adviser Scheme (CLAS), which covers information security in a broader sense and tackles areas such as BS7799, security policy creation, and auditing.

To correctly accredit CHECK consultants, CESG runs an assault course to test the attack and penetration

Hackers Defined

Later in the book, I define hacking as being:

The art of manipulating a process in such a way that it performs an action that is useful to you.

Which I think is a true representation of a hacker in any sense of the word, whether a computer programmer who used to hack code on mainframes back in the day, so that it would perform an action useful to him, or a modern computer attacker with a very different goal and set of ethics. Please bear in mind that when I use the term hacker in this book, I am talking about a network-based assailant trying to compromise the security of a system, and I don't mean to step on the toes of hackers in the traditional sense, who have sound ethics and morals.

Organization

This book consists of 14 chapters and 2 appendixes. At the end of each chapter is a checklist that summarizes the threats and techniques described in that chapter along with effective countermeasures. The appendixes provide useful reference material, including listings of TCP and UDP ports, along with ICMP message types and their functions. Details of popular vulnerabilities in Microsoft Windows and Unix-based operating platforms are also listed. Here is a brief description of each chapter and appendix.

Chapter 1, discusses the rationale behind network security assessment and introduces security as a process, not a product.

Chapter 2, covers the various Unix-based operating systems and tool kits that determined attackers and network security professionals use.

Chapter 3, logically walks through the Internet-based options that a potential attacker has to map your network, from open web searches to DNS sweeping and querying of authoritative name servers.

Chapter 4, discusses all known IP network scanning techniques and their relevant application, also listing tools and systems that support such scanning types. IDS evasion and low-level packet analysis techniques are also covered.

Chapter 5, defines the techniques and tools that execute information leak attacks against services such as LDAP, *auth*, finger, and DNS. Some process manipulation attacks are discussed here when appropriate.

Chapter 6, comprehensively covers the assessment of web services including IIS, Apache, OpenSSL, and other components such as Frontpage Extensions and Outlook Web Access. Risk mitigation strategies are also detailed, including use of egress network filtering and web service configuration.

Chapter 7, details the tools and techniques used to correctly assess all common maintenance services (including SSH, VNC, X Windows, Microsoft Terminal Services, etc.). Increasingly, these services are targets of information leak and brute-force attacks, resulting in a compromise even though the underlying software isn't strictly vulnerable.

Chapter 8, outlines assessment strategies for testing FTP and database services correctly. I cover Unix-based FTP services along with common enterprise database services, such as Oracle and Microsoft SQL Server.

Chapter 9, comprehensively tackles security issues with each and every component (including MSRPC, NetBIOS, and CIFS) in a port-by-port fashion. Information-leak, brute-force, and process-manipulation attacks against each component are detailed, from the DCE locator service listening on port 135 through to the CIFS direct listener on port 445.

Chapter 10, details assessment of SMTP, POP-3, and IMAP services that transport email. Often, these services can fall foul to information-leak and brute-force attacks, and, in some instances, process manipulation.

Chapter 11, covers assessment of IP services that provide secure inbound network access, including IPsec, Check

Audience

This book assumes you are familiar with IP and administering Unix-based operating systems, such as Linux or Solaris. A technical network administrator or security consultant should be comfortable with the contents of each chapter. To get the most out of this book, you should be familiar with:

- The IP protocol suite, including TCP, UDP, and ICMP
- Workings of popular Internet network services, including FTP, SMTP, and HTTP
- At least one Unix-like operating system, such as Linux or a BSD-derived platform
- Configuring and building Unix-based tools in your environment
- Firewalls and network filtering models (DMZ segments, bastion hosts, etc.)

Mirror Site for Tools Mentioned in This Book

URLs for tools in this book are listed so that you can browse the latest files and papers on each respective site. If you are worried about Trojan horses or other malicious content within these executables, they have been virus-checked and are mirrored at the O'Reilly site: http://examples.oreilly.com/networksa/tools.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You don't need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book doesn't require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code doesn't require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but don't require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "Network Security Assessment, by Chris McNab. Copyright 2004 O'Reilly & Associates, Inc., 0-596-00611-X."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates commands, example URLs, passwords, error messages, filenames, emphasis, and the first use of technical terms

Constant width

Indicates IP addresses and Unix command-line examples

Constant width bold italic

Indicates replaceable text

`Constant width bold`

Indicates user input

This icon signifies a tip, suggestion, or general note.

This icon indicates a warning or caution.

Comments and Questions

Please address comments and questions concerning this book to the publisher:
O'Reilly Media, Inc.1005 Gravenstein Highway NorthSebastopol, CA 95472(800) 998-9938 (in the United States or Canada)(707) 829-0515 (international or local)(707) 829-0104 (fax)

There's a web page for this book that lists errata, examples, and any additional information. You can access this page at:
http://www.oreilly.com/catalog/networksa

To comment or ask technical questions about this book, send email to:
bookquestions@oreilly.com

For more information about books, conferences, Resource Centers, and the O'Reilly Network, see the O'Reilly web site at:
http://www.oreilly.com

Acknowledgments

I thank my family for their support, and my partner Georgina for putting up with all the hours I've spent sat in front of my PC writing this book!

As I look back over the last 23 years of my life, I realize that I have met a handful of key individuals to whom I owe a great deal, as I truly believe that I wouldn't have ended up here without their input in one form or another: Wez Blampied, Emerson Tan, Jeff Fay, Bryan Self, John McDonald, Geoff Donson, Kevin Chamberlain, Steve McMahon, Ryan Gibson, and Nick Baskett.

I am also extremely grateful for the positive support from the O'Reilly team (past and present), including Jim Sumser, Laurie Petrycki, Debby Russell, Tatiana Apandi Diaz, Nathan Torkington, and David Chu.

I thank the guys I've work alongside at Matta (http://www.trustmatta.com) for their support, and also MIS Corporate Defence Solutions (http://www.mis-cds.com)—the company that took me in and gave me an excellent start to my security career.

Online handles of those with whom I talk from time to time about the latest exploits and tools include the likes of: bond, twd, sinkhole, cr, Cold-Fire, ph0bos, gamma, i1l, cain, superluck, almauri, snare, Mixter, DiGiT-, Cybk0red, none, brc, dSan, gera, suid, caddis, duke, yowie, Joey_ _, tmoggie, biometrix, B-r00t, Haggis, giles, kraft, _sh, xfer, and skyper. I would like to thank these individuals for their time over the years, and show them my gratitude by acknowledging them here. Finally, thanks to Matt Lewis for his help with the application security chapter and to Michael Thumann for his Chapter 11 input.

Chapter 1. Network Security Assessment

This chapter discusses at a high level the rationale behind Internet-based network security assessment and penetration testing. To retain complete control over your networks and data, you must take a proactive approach to security, an approach that starts with assessment to identify and categorize your risks. Network security assessment is an integral part of any security life cycle.

1.1 The Business Benefits

From a commercial standpoint, assurance of network security is a business enabler. As a security consultant at the time of writing, I am helping a particular client in the retail sector to deploy and secure an 802.11b wireless network for use in nearly 200 stores across the United Kingdom. This wireless network has been designed in a security-conscious manner, allowing the retailer to embrace wireless technologies to improve efficiency and the quality of their service.

Shortcomings in network security and user adherence to security policy often allow Internet-based attackers to locate and compromise networks. High-profile examples of companies who have fallen victim to such determined Internet-based attackers over the last four years include:

- RSA Security (http://www.2600.com/hacked_pages/2000/02/www.rsa.com/)
- OpenBSD (http://lists.jammed.com/incidents/2002/08/0000.html)
- NASDAQ (http://www.wired.com/news/politics/0,1283,21762,00.html)
- Playboy Enterprises (http://www.vnunet.com/News/1127004)
- Cryptologic (http://lists.jammed.com/ISN/2001/09/0042.html)

These compromises have come about in similar ways, involving large losses in some cases. Cryptologic is an online casino gaming provider that lost $1.9 million in a matter of hours to determined attackers. In the majority of high profile incidents, the attackers used a selection of the following techniques:

- Compromising a poorly configured or protected peripheral system that is related to the target network space or host using publicly available exploits, such as scripts available from Packet Storm (http://www.packetstormsecurity.org) and other archives
- Directly compromising key network components using private exploit tools, such as scripts that the attacker or his hacking group have developed for their own personal use
- Compromising traffic and circumventing security mechanisms using ARP redirection and network sniffing
- Compromising user account passwords and using those passwords to compromise other hosts where the user may have an active account
- Abusing blatant system or network configuration issues, reading sensitive information from publicly accessible web folders, or bypassing poor firewall rules that open up the network to attack

1.2 IP: The Foundation of the Internet

The Internet Protocol Version 4 (IPv4) is the networking protocol suite all public Internet sites currently use to communicate and transmit data to one another. From a network security assessment methodology standpoint, this book comprehensively discusses the steps that should be taken during the security assessment of any IPv4 network.

IPv6 is an improved protocol that is gaining popularity among academic networks. IPv6 offers a 128-bit network space (3.4 x 1038 addresses) as opposed to the 32-bit space of IPv4 (only 4 billion addresses) that allows a massive number of devices to have publicly routable addresses. Eventually, the entire Internet will migrate across to IPv6, and every electronic device in your home will have an address.

Due to the large size of the Internet and sheer number of security issues and vulnerabilities publicized, opportunistic attackers (commonly referred to as script kiddies) will continue to scour the public IP address space seeking vulnerable hosts. The combination of new vulnerabilities being disclosed on a daily basis, along with the adoption of IPv6, ensures that opportunistic attackers will always be able to compromise a certain percentage of Internet networks.

1.3 Classifying Internet-Based Attackers

The first type of threat that all publicly accessible networks are at risk from is that posed by opportunistic attackers. These attackers use auto-rooting scripts and network scanning tools to find and compromise vulnerable Internet hosts. Most opportunistic attackers fall into two distinct groups:

- Those who compromise hosts for denial-of-service and flooding purposes
- Those who compromise hosts through which attacks can be bounced (including port scans, breaking into other hosts, or sending spam email)

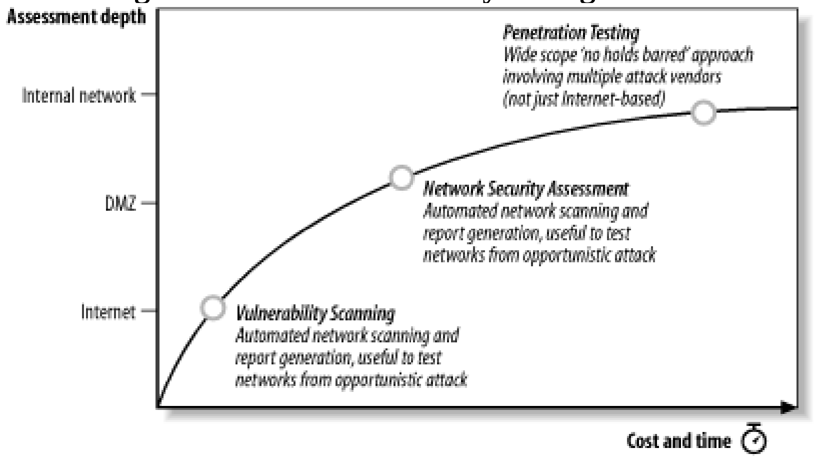
The second type of threat is that posed by determined attackers. A determined attacker will exhaustively probe every point of entry into a target network from the Internet, port scanning each and every IP address and assessing each and every network service in depth. Even if the determined attacker can't compromise the target network on his first attempt, he will be aware of areas of weakness. Detailed knowledge of a site's operating systems and network services allows the determined attacker to compromise the network upon the release of new exploit scripts in the future.

In light of this, the networks that are most at risk are those with sizeable numbers of publicly accessible hosts. Having many entry points into a network multiplies the exploitable vulnerabilities that exist at different levels; managing these risks becomes an increasingly difficult task as networks grow.

1.4 Assessment Service Definitions

Most security providers (both service and product companies) offer a number of assessment services branded in a variety of ways. Figure 1-1 shows the key service offerings along with the depth of assessment and relative cost. Each service type can provide varying degrees of security assurance.

Figure 1-1. Different security testing services



Vulnerability scanning uses automated systems (such as ISS Internet Scanner, QualysGuard, or eEye Retina) with minimal hands-on qualification and assessment of vulnerabilities. This is an inexpensive way to ensure that no obvious vulnerabilities exist, but it doesn't provide a clear strategy to improve security.

Network security assessment lies neatly between vulnerability assessment and full-blown penetration testing; it offers an effective blend of tools and hands-on vulnerability testing and qualification by trained analysts. The report is usually hand-written, giving professional advice that can improve a company's security.

Full-blown *penetration testing* is outside the scope of this book; it involves multiple attack vectors (e.g., telephone war dialing, social engineering, wireless testing, etc.) to compromise the target environment. Instead this book fully demonstrates and discusses the methodologies adopted by determined Internet-based attackers to compromise IP networks remotely, which in turn will allow you to improve IP network security.

1.5 Network Security Assessment Methodology

The best practice assessment methodology used by determined attackers and network security consultants involves four distinct high-level components:

- Network enumeration to identify IP networks and hosts of interest
- Bulk network scanning and probing to identify potentially vulnerable hosts
- Investigation of vulnerabilities and further network probing by hand
- Exploitation of vulnerabilities and circumvention of security mechanisms

This complete methodology is relevant to Internet-based networks being tested in a blind fashion with limited target information (such as a single DNS domain name). If a consultant is enlisted to assess a specific block of IP space, he skips initial network enumeration and commences bulk network scanning and investigation of vulnerabilities.

1.5.1 Internet Host and Network Enumeration

Publicly available reconnaissance techniques, including web and newsgroup searches, Network Information Center (NIC) WHOIS querying, and Domain Name System (DNS) probing, are used to collect data about the structure of the target network from the Internet without actually scanning the network or necessarily probing it directly.

Initial reconnaissance is very important because it identifies hosts that aren't properly fortified from attack. A determined attacker invests time in identifying peripheral networks and hosts, while companies and organizations concentrate their efforts on securing obvious public systems (such as public web and mail servers) but neglecting hosts and networks that lay off the beaten track.

It may well be the case that a determined attacker also enumerates networks of third party suppliers and business partners that, in turn, have access to the target network space. Nowadays such third parties often have dedicated links into areas of internal corporate network space through VPN tunnels and other links.

Key pieces of information that are gathered through initial reconnaissance include details of Internet-based network blocks, internal IP addresses gathered from DNS servers, insight into the target organization's DNS structure (including domain names, subdomains, and hostnames), and IP network relationships between physical locations.

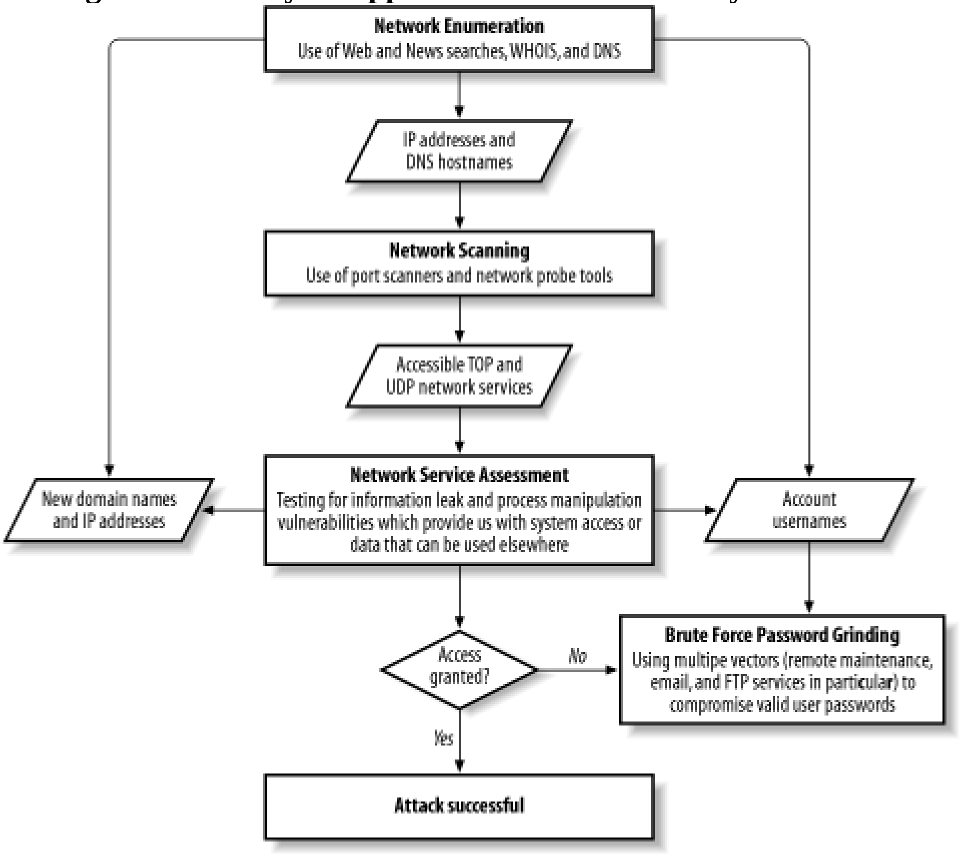
This information is then used to perform structured bulk network scanning and probing exercises to assess further the target network space and investigate potential vulnerabilities. Further reconnaissance involves extracting user details (including email addresses), telephone numbers, and office addresses.

1.5.2 Bulk Network Scanning and Probing

1.6 The Cyclic Assessment Approach

Assessment of large networks in particular can become a very cyclic process if you are testing the networks of an organization in a blind sense and are given minimal information. As you test the network, information leak bugs can be abused to find different types of useful information (including trusted domain names, IP address blocks, and user account details) that is then fed back into other processes. Figure 1-2s flowchart defines this approach and the data being passed between processes.

Figure 1-2. The cyclic approach to network security assessment


Network Enumeration
Use of Web and News searches, WHOIS, and DNS
IP addresses and DNS hostnames
Network Scanning
Use of port scanners and network probe tools
Accessible TCP and UDP network services
Network Service Assessment
Testing for information leak and process manipulation vulnerabilities which provide us with system access or data that can be used elsewhere
New domain names and IP addresses
Account usernames
Access granted?
No
Yes
Brute Force Password Grinding
Using multipe vectors (remote maintenance, email, and FTP services in particular) to compromise valid user passwords
Attack successful

This flowchart starts with network enumeration, then bulk network scanning, and finally specific service assessment. It may be the case that by assessing a rogue non-authoritative DNS service an analyst may identify previously unknown IP address blocks, which can be fed back into the network enumeration process to identify further network components. In the same way, an analyst may enumerate a number of account usernames by exploiting public folder information leak vulnerabilities in Microsoft Outlook Web Access, which can be fed into a brute-force password grinding process later on.

Chapter 2. The Tools Required

This chapter describes the operating systems and some key tools required to undertake an IP-based network security assessment. Many advanced TCP/IP assessment utilities are available only for Unix-based systems such as Linux, so you will often find that a competent security consultant uses a variety of tools under different operating systems to assess and successfully penetrate a network. These tools and their respective uses are discussed in detail throughout the book, and they are listed here so that you can select and start to prepare your assessment platform before moving forward.

All tools listed in this book can also be found in the O'Reilly archive at http://examples.oreilly.com/networksa/tools. I have listed the original sites in most cases so that you can freely browse other tools and papers on each respective site.

2.1 The Operating Systems

Selecting the operating platforms to use during a network security assessment depends on the type of network you are going to test (e.g., completely Microsoft Windows), and the depth to which you will perform your assessment. Often it is the case that to successfully launch exploit scripts against Linux or Unix systems, access to a Unix-like platform (usually Linux or BSD-derived) is required to correctly compile and run specialist exploit tools. What follows is a discussion of the operating systems that are commonly used.

2.1.1 Windows NT Family Platforms

As Windows NT systems (NT 4.0, 2000, XP, 2003 Server, etc.) start to mature and become more flexible, many more network assessment and hacking tools are available that run cleanly on the platform. Previous Windows releases didn't give raw access to network sockets, so many tools used by consultants had to be run from Unix-based platforms. This is no longer the case; increasing amounts of useful security utilities have been ported across to Windows, including *nmap* and powerful tools within the *dsniff* package, such as *arpspoof*.

2.1.2 Linux

Linux is the choice of most hackers and security consultants alike. The Linux platform is versatile, and the system kernel provides low-level support for leading-edge technologies and protocols (Bluetooth being a good example at the time of writing). All mainstream IP-based attack and penetration tools can be built and run under Linux with no problems, due to the inclusion of extensive networking libraries such as *libpcap*.

I use Red Hat (http://www.redhat.com) and Debian (http://www.debian.org) Linux distributions on laptops and servers within the office. Debian is useful because of its *apt-get* package search and installation tool that can be used to install and update system packages. Red Hat packages are easily installed using the *rpm* command along with various wrappers that hook into sites such as RPMfind (http://www.rpmfind.net) to automatically update and install packages.

2.1.3 MacOS X

MacOS X is a BSD-derived operating system. The underlying system looks and feels very much like any Unix environment, with standard command shells (such as *sh*, *csh*, and *bash*) and useful network utilities that can be used during an IP-based network security assessment (including *telnet*, *ftp*, *rpcinfo*, *snmpwalk*, *host*, and *dig*).

MacOS X is supplied with a compiler, and many header and library files that allow for specific assessment tools to be built. Three useful tools easily built under MacOS X include *nmap*, Nessus, and *nikto*.

2.1.4 VMware

VMware is an extremely useful program that allows you to run multiple instances of operating systems easily on a single laptop or workstation. VMware Workstation (Version 4 at the time of writing) is a fully supported commercial package, available from http://www.vmware.com for both Windows and Linux. To register and purchase the full VMware workstation product costs a single user in the region of $300.

I run VMware from my Windows 2000 workstation to run and access Linux in parallel, as needed during a network

2.2 Free Network Scanning Tools

Following is an introduction to a small number of scanning tools that I will discuss throughout the book.

2.2.1 nmap

The command-line driven *nmap* utility is a port scanner designed to scan large networks and determine which hosts are up and which TCP and UDP network services they offer. *nmap* supports a large number of popular ICMP, TCP, and UDP scanning techniques, also offering a number of advanced features such as service protocol fingerprinting, IP fingerprinting, stealth scanning and low-level filter analysis.

nmap is available from http://www.insecure.org/nmap/. Currently *nmap* can be run under Windows 2000 and Unix operating systems, including Linux and MacOS X.

2.2.2 Nessus

Nessus is a vulnerability assessment package that can perform many automated tests against a target network, including:

- ICMP sweeping
- TCP and UDP port scanning
- Banner grabbing and network service assessment
- Brute force against common network services
- IP fingerprinting and other peripheral functions

I know of auditing teams within the big five accounting firms who use Nessus to undertake much of their network scanning and assessment work. Nessus has two components (daemon and client) and deploys in a distributed fashion that permits effective network coverage and management.

Nessus has a good reporting engine that can present comprehensive results along with relevant CVE entries. CVE is a detailed list of common vulnerabilities maintained by the MITRE Corporation (accessible at http://cve.mitre.org).

Nessus is available for download from http://www.nessus.org. At the time of writing, the daemon component is available only for Unix-based systems such as Linux, Solaris, and FreeBSD. The Unix Nessus client software is bundled with the daemon component in a single package; Windows clients are also available.

2.2.3 NSAT

2.3 Commercial Network Scanning Tools

Commercial scanning packages are used by many network administrators and those responsible for the security of large networks. Although not cheap (with software licenses often in the magnitude of tens of thousands of dollars), commercial systems are supported and maintained by the respective vendor, so vulnerability databases are kept up-to-date. With this level of professional support, a network administrator can assure the security of his network to a certain level.

Here's a selection of popular commercial packages:

- Core IMPACT (http://www.corest.com/products/coreimpact/)
- ISS Internet Scanner (http://www.iss.net)
- Cisco Secure Scanner (http://www.cisco.com/warp/public/cc/pd/sqsw/nesn/)

A problem with such one-stop automated vulnerability assessment packages is that increasingly, they record false positive results. When professionally scanning large networks, it is often advisable to use a commercial system such as ISS Internet Scanner to perform an initial bulk scanning and network service assessment of a network, then fully qualify vulnerabilities and investigate network components by hand to produce accurate results.

2.4 Protocol-Dependent Assessment Tools

When assessing the security of specific services, specialist tools can perform assessment in specific areas, such as enumeration and brute-force password grinding. What follows here is an introduction to a number of freely available tools you can use to assess Windows networking, DNS, and web services.

2.4.1 Microsoft NetBIOS, SMB, and CIFS

NetBIOS, Server Message Block (SMB), and Common Internet File System (CIFS) protocols are used primarily within Microsoft Windows networks for user authentication, file sharing, and access to services such as Microsoft Exchange over RPC. CIFS is a relatively new incarnation of SMB over NetBIOS; it's for vendors seeking to move away from NetBIOS and toward CIFS. Windows 2000, for example, runs SMB over NetBIOS on port 139 and CIFS on port 445. CIFS is the native protocol used in Windows 2000 networks, so SMB access through NetBIOS provides backward compatibility.

NetBIOS and CIFS assessment tools fall into two categories: enumeration and information gathering, and brute-force password guessing. Enumeration tools are used to gather system information using anonymous null sessions and other techniques. Brute-force tools are then used to compromise account passwords and gain access to shared files and resources.

2.4.1.1 Enumeration and information gathering tools

enum (http://razor.bindview.com/tools/files/enum.tar.gz)

Jordan Ritter's *enum* utility is a Windows command-line tool that extensively queries target hosts running NetBIOS through TCP port 139. The tool can list usernames, password policy, shares, and details of other hosts including domain controllers.

epdump (http://www.packetstormsecurity.org/NT/audit/epdump.zip)

The *epdump* Windows command-line utility queries the RPC end-point mapping service at TCP port 135 to enumerate network interfaces along with details of RPC services and named pipes that are accessible.

nbtstat

The *nbtstat* command is found within all recent Microsoft Windows systems. *nbtstat* queries the NetBIOS name service running on UDP port 137, resulting in the NetBIOS name table being returned (including the hostname, domain name, details of logged-in users, shared resources, and the MAC address of the network interface).

usrstat

The *usrstat* utility is part of the Windows NT 4.0 Resource Kit (http://www.microsoft.com/ntserver/nts/downloads/recommended/netkit/default.asp). It can run against a target NetBIOS session service to enumerate user details (using anonymous null sessions) against the IPC$ administrative share. Information that is returned includes the login name, full name, and last logon date for each user.

PREV < Day Day Up > NEXT

Chapter 3. Internet Host and Network Enumeration

This chapter focuses on the first steps you should take when assuming the role of an Internet-based attacker. An early avenue that any competent attacker would pursue involves querying entirely legal and public sources of information, such as WHOIS, DNS, and even web and newsgroup search engines including Google. Attackers can often build a clear picture of your network by launching indirect probes, without most network administrators even knowing. By identifying systems of interest (such as development or test systems), attackers can focus on specific areas of the target network later on.

This chapter comprehensively covers enumeration through Web and newsgroup searches, NIC querying, DNS querying, and SMTP probing.

The reconnaissance process is often interactive, repeating the full enumeration cycle when a new piece of information (such as a domain name or office address) is found. The scope of the assessment exercise usually defines the boundaries, which sometimes include testing third parties that you identify while performing in-depth enumeration. I know of a number of companies whose networks were compromised by extremely determined attackers breaking home user PCs that were using always-on cable modem connections and then "piggy backing" into the corporate network.

3.1 Web Search Engines

As web crawlers scour the Internet's web sites for content, they catalog pieces of potentially useful information. Search engines, such as Google, now provide advanced search functions that allow attackers to build a clearer picture of the network that they plan to attack later.

In particular, the following types of information are easily found:

- Employee contact details and information
- Email addresses
- DDI telephone numbers
- Physical addresses of offices from which the employees are based
- Details of internal email systems
- DNS layout and naming convention, including domains and hostnames
- Documents that reside on publicly accessible servers

Direct-dial telephone numbers are especially useful to determined attackers, who may later launch war dialing and other telephone-based attacks. It is very difficult for organizations and companies to prevent this information from being ascertained; for example, it is made freely available every time a user posts to a mailing list with his signature. To manage this risk more effectively, companies should go through public record querying exercises to ensure that the information an attacker can collect doesn't lead to a compromise.

3.1.1 Google Advanced Search Functionality

Using a powerful advanced search function, Google can indirectly map networks and gather potentially useful information. The advanced search function itself is directly accessible at http://www.google.com/advanced_search?hl=en. In terms of the functionality, searches can be refined in the following ways:

Filtering words

Exclude pages that don't include specific words or phrases, for example

Language

3.2 NIC Querying

Network Information Centers (NICs) store useful information in WHOIS databases, primarily as *network*, *route*, or *person* objects. WHOIS database objects define which areas of Internet space are registered to which organizations, with other information such as routing and contact details in the case of abuse.

There are three primary regions under which all public Internet-based network blocks and IP address spaces fall. The following international registrars around the world can retrieve useful information (including names of technical IT staff, details of IP network blocks, and physical office locations):

- American Registry for Internet Numbers (ARIN) at http://www.arin.net
- Asia Pacific Network Information Centre (APNIC) at http://www.apnic.net
- Réseaux IP Européens (RIPE) at http://www.ripe.net

Each respective regional registrar's WHOIS database contains information relevant to that particular region. For example, the RIPE WHOIS database doesn't contain information about network space and other objects that are found in the Americas.

3.2.1 NIC Querying Tools and Examples

Tools that are used to query NIC WHOIS databases include:

- The Sam Spade Windows client (available from http://www.samspade.org)[1]

  [1] URLs for tools in this book are mirrored at the O'Reilly site, http://examples.oreilly.com/networksa/tools.

- The *whois* client found within Unix-based environments
- Direct querying via the appropriate regional WHOIS

3.2.1.1 Using the Sam Spade Windows client

The Sam Spade client is a powerful and easy-to-use Windows tool that can perform many public-record query functions, as shown in Figure 3-4.

Figure 3-4. The Sam Spade Windows client

Spade - [IP block 144.51.92.35, finished]
File Edit View Window Basics Tools Help

3.3 DNS Querying

Using tools such as *nslookup*, *host,* and *dig*, you can launch DNS requests and probes against domains and IP address blocks identified during the web search and NIC querying phases. Other tools also perform reverse DNS sweeps against IP network blocks to identify hostnames and other domains.

DNS requests and probes can be launched to retrieve parts of, or in some cases, entire DNS zone files for specified domains or network spaces. Most DNS servers around the Internet can be quizzed for useful information, including:

- Authoritative DNS server information from name server (NS) records
- Domain and subdomain information
- Hostname information from A, PTR, and CNAME records
- Public points of presence that list mail exchanger (*MX*) records

In some cases, poorly configured DNS servers also allow you to enumerate:

- Operating-system and platform information of hosts from the host information (*HINFO*) record
- Names and IP addresses of internal or nonpublic hosts and networks

You can very often uncover previously unknown network blocks and hosts during DNS querying. If new network blocks are found, I recommend launching a second round of WHOIS queries and web searches to get further information about each new network block.

DNS probing in this fashion is stealthy in the sense that there is no active scanning or probing of the target networks. Instead, you simply probe and query the authoritative DNS servers for those domains or network blocks that are often run by ISPs. Most name servers aren't even configured to pick up on potential sweeps of this sort, because it resembles standard DNS traffic.

3.3.1 Forward DNS Querying

Forward DNS records are required for organizations and companies to integrate and work correctly as part of the Internet. Two examples of legitimate forward queries are when an end user accesses a web site and during the receipt of email when SMTP mail exchanger information is requested about the relevant domain. Attackers issue forward DNS queries to identify mail servers and other obvious Internet-based systems.

Tools that query DNS servers directly include:

3.4 Enumeration Technique Recap

It is an interesting and entirely legal exercise to enumerate the CIA and other organizations' networks from the Internet by querying public records. As a recap, here is a list of public Internet-based querying techniques and their application:

Web and newsgroup searches

Using Google to perform searches against established domain names and target networks to identify personnel, hostnames, domain names, and useful data residing on publicly accessible web servers.

NIC querying

Querying NIC databases such as ARIN, APNIC, and RIPE to retrieve network block, routing, and contact details related to the target networks and domain names. NIC querying gives useful information relating to the sizes of reserved network blocks (useful later when performing intrusive network scanning).

DNS querying

Querying publicly accessible DNS servers to enumerate hostnames and subdomains. Misconfigured DNS servers can also be abused to download DNS zone files that categorically list subdomains, hostnames, operating platforms of devices and internal network information in severe cases.

SMTP probing

Sending email to nonexistent accounts at target domains to map internal network space by analyzing the responses from the SMTP system.

3.5 Enumeration Countermeasures

Use the following checklist of countermeasures to effectively reconfigure your Internet-facing systems not to give away potentially sensitive information:

- Configure web servers to prevent indexing of directories that don't contain *index.html* or similar index files (*default.asp* under IIS, for example). Also ensure that sensitive documents and files aren't kept on publicly accessible hosts, such as HTTP or FTP servers.
- Always use a generic, centralized network administration contact detail (such as an IT help desk) in Network Information Center databases, to prevent potential social engineering and war dialing attacks against IT departments from being effective.
- Configure all name servers to disallow DNS zone transfers to untrusted hosts.
- Ensure that nonpublic hostnames aren't referenced to IP addresses within the DNS zone files of publicly accessible DNS servers, to prevent reverse DNS sweeping from being effective. This practice is known as *split horizon DNS*, using separate DNS zones internally and externally.
- Ensure that *HINFO* and other novelty records don't appear in DNS zone files.
- Configure SMTP servers either to ignore email messages to unknown recipients or to send responses that don't include the following types of information:
    - Details of mail relay systems being used (such as Sendmail or MS Exchange).
    - Internal IP address or host information.

Chapter 4. IP Network Scanning

This chapter focuses on the technical execution of IP network scanning. After undertaking initial reconnaissance to identify IP address spaces of interest, network scanning builds a clearer picture of accessible hosts and their network services. Network scanning and reconnaissance is the real data gathering exercise of an Internet-based security assessment. The rationale behind IP network scanning is to gain insight into the following elements of a given network:

- ICMP message types that generate responses from target hosts
- Accessible TCP and UDP network services running on the target hosts
- Operating platforms of target hosts and their configuration
- Areas of vulnerability within target host IP stack implementations (including sequence number predictability for TCP spoofing and session hijacking)
- Configuration of filtering and security systems (including firewalls, border routers, switches, and IDS sensors)

Performing both network scanning and reconnaissance tasks paints a clear picture of the network topology and its security mechanisms. Before penetrating the target network, further assessment steps involve gathering specific information about the TCP and UDP network services that are running, including their versions and enabled options.

4.1 ICMP Probing

The Internet Control Message Protocol (ICMP) identifies potentially weak and poorly protected networks. ICMP is a short messaging protocol that's used by systems administrators and end users for continuity testing of networks (e.g., using the *ping* or *traceroute* commands). From a network scanning and probing perspective, the following types of ICMP messages are useful:

Type 8 (echo request)

Echo request messages are also known as ping packets. You can use a scanning tool such as *nmap* to perform ping sweeping and easily identify hosts that are accessible.

Type 13 (timestamp request)

A timestamp request message requests system time information from the target host. The response is in a decimal format and is the number of milliseconds elapsed since midnight GMT.

Type 15 (information request)

The ICMP information request message was intended to support self-configuring systems such as diskless workstations at boot time, to allow them to discover their network address. Protocols such as RARP, BOOTP, or DHCP do so more robustly, so type 15 messages are rarely used.

Type 17 (subnet address mask request)

An address mask request message reveals the subnet mask used by the target host. This information is useful when mapping networks and identifying the size of subnets and network spaces used by organizations.

Firewalls of security-conscious organizations often blanket-filter inbound ICMP messages and so ICMP probing isn't effective; however, ICMP isn't filtered in most networks because ICMP messages are often useful for network troubleshooting purposes.

There are a handful of other ICMP message types that have relevant security applications (such as ICMP type 5 redirect messages sent by routers), but they aren't related to network scanning.

Table 4-1 outlines popular operating systems and their responses to certain types of direct ICMP query messages.

Table 4-1. Operating system responses to direct ICMP query messages

Operating system	**Direct ICMP message types (non-broadcast)**			

4.2 TCP Port Scanning

Accessible TCP ports can be identified by port scanning target IP addresses. The following nine different types of TCP port scanning are used in the wild by both attackers and security consultants:

Standard scanning methods
 Vanilla connect( ) scanningHalf-open SYN flag scanning

Stealth TCP scanning methods
 Inverse TCP flag scanningACK flag probe scanningTCP fragmentation scanning

Third-party and spoofed TCP scanning methods
 FTP bounce scanningProxy bounce scanningSniffer-based spoofed scanningIP ID header scanning

What follows is a technical breakdown for each TCP port scanning type, along with details of Windows and Unix-based tools that can perform scanning.

4.2.1 Standard Scanning Methods

Standard scanning methods, such as vanilla and half-open SYN scanning, are extremely simple direct techniques used to identify accessible TCP ports and services accurately. These scanning methods are reliable but are easily logged and identified.

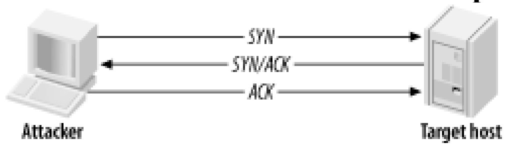
4.2.1.1 Vanilla connect( ) scanning

TCP connect( ) port scanning is the most simple type of probe to launch. There is no stealth whatsoever involved in this form of scanning because a full TCP/IP connection is established with TCP port one of the target host, then incrementally through ports two, three, four, and so on.

TCP/IP's reliability as a protocol, vanilla port scanning is a very accurate way to determine which TCP services are accessible on a given target host. Figures Figure 4-2 and Figure 4-3 show the various TCP packets and their flags, as they are sent and received by the attacker and the host he is scanning.

In Figure 4-2, the attacker first sends a SYN probe packet to the port he wishes to test. Upon receiving a packet from the port with the SYN and ACK flags set, he knows that the port is open. The attacker completes the three-way handshake by sending an ACK packet back.

Figure 4-2. A vanilla TCP scan result when a port is open



If, however, the target port is closed, the attacker receives an RST/ACK packet directly back, as shown in Figure 4-3.

Figure 4-3. A vanilla TCP scan result when a port is closed

4.3 UDP Port Scanning

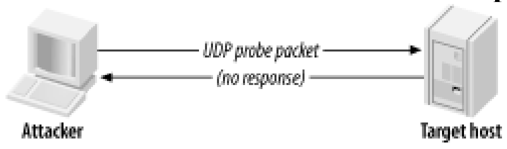
Because UDP is a connectionless protocol, there are only two ways to effectively enumerate accessible UDP network services across an IP network:

- Send UDP probe packets to all 65535 UDP ports, then wait for "ICMP destination port unreachable" messages to identify UDP ports that aren't accessible.
- Use specific UDP service clients (such as *snmpwalk*, *dig*, or *tftp*) to send UDP datagrams to target UDP network services and await a positive response.

Many security-conscious organizations filter ICMP messages to and from their Internet-based hosts, so it is often difficult to assess which UDP services are accessible via simple port scanning. If "ICMP destination port unreachable" messages can escape the target network, a traditional UDP port scan can be undertaken to deductively identify open UDP ports on target hosts.

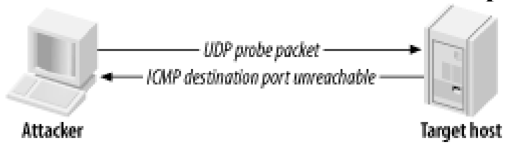
Figures Figure 4-12 and Figure 4-13 show the UDP packets and ICMP responses generated by hosts when ports are open and closed.

Figure 4-12. An inverse UDP scan result when a port is open



UDP port scanning is an inverted scanning type in which open ports don't respond. What is looked for, in particular, are ICMP destination port unreachable (type 3 code 3) messages from the target host, as shown in Figure 4-13.

Figure 4-13. An inverse UDP scan result when a port is closed



4.3.1 Tools That Perform UDP Port Scanning

nmap supports UDP port scanning with the -sU option. The latest version of Foundstone's *SuperScan* also supports UDP port scanning. However, both tools wait for negative "ICMP destination port unreachable" messages to identify open ports (i.e., those ports that don't respond). If these ICMP messages are filtered by a firewall as they try to travel out of the target network, inaccurate results are gleaned.

During a comprehensive audit of Internet-based network space, you should send crafted UDP client packets to popular services and await a positive response. The *scanudp* utility developed by Fryxar (http://www.geocities.com/fryxar/) does this very well.

4.4 IDS Evasion and Filter Circumvention

IDS evasion, when launching any type of IP probe or scan, involves one or both of the following tactics:

- Use of fragmented probe packets, assembled when they reach the target host
- Use of spoofing to emulate multiple fake hosts launching network scanning probes, in which the real IP address of the scanning host is inserted to collect results

Filtering mechanisms can be circumvented at times using malformed or fragmented packets. However, the common techniques used to bypass packet filters at either the network or system-kernel level are as follows:

- Use of source routing
- Use of specific TCP or UDP source ports

First, I'll discuss IDS evasion techniques of fragmenting data and emulating multiple hosts, and then filter circumvention methodologies. These techniques can often be mixed to launch attacks using source routed, fragmented packets to bypass both filters and IDS systems.

4.4.1 Fragmenting Probe Packets

Probe packets can be fragmented easily with *fragroute* to fragment all probe packets flowing from your host or network or with a port scanner that supports simple fragmentation, such as *nmap*. Many IDS sensors can't process large volumes of fragmented packets because doing so creates a large overhead in terms of memory and CPU consumption at the network sensor level.

4.4.1.1 fragtest

Dug Song's *fragtest* utility (available as part of the *fragroute* package from http://www.monkey.org/~dugsong/fragroute/) can determine exactly which types of fragmented ICMP messages are processed and responded to by the remote host. ICMP echo request messages are used by *fragtest* for simplicity and allow for easy analysis; the downside is that the tool can't assess hosts that don't respond to ICMP messages.

After undertaking ICMP probing exercises (such as ping sweeping and hands-on use of the *sing* utility) to ensure that ICMP messages are processed and responded to by the remote host, *fragtest* can perform three particularly useful tests:

- Send an ICMP echo request message in 8-byte fragments (using the frag option)
-

4.5 Low-Level IP Assessment

Tools such as *nmap*, *hping2*, and *firewalk* perform low-level IP assessment. Sometimes holes exist to allow certain TCP services through the firewall, but the expected service isn't running on the target host. Such low-level network details are useful to know, especially in sensitive environments (e.g., online banking environments), because very small holes in network integrity can sometimes be abused along with larger problems to gain or retain access to target hosts.

Insight into the following areas of a network can be gleaned through low-level IP assessment:

- Uptime of target hosts (by analyzing the TCP timestamp option)
- TCP services that are permitted through the firewall (by analyzing responses to TCP and ICMP probes)
- TCP sequence and IP ID incrementation (by running predictability tests)
- The operating system of the target host (using IP fingerprinting)

nmap automatically attempts to calculate target host uptime information by analyzing the TCP timestamp option values of packets received. The TCP timestamp option is defined in RFC 1323; however, many platforms don't adhere to RFC 1323. This feature often gives accurate results against Linux operating systems and others such as FreeBSD, but your mileage may vary.

4.5.1 Analyzing Responses to TCP Probes

A TCP probe always results in one of four responses. These responses potentially allow an analyst to identify where a connection was accepted, or why and where it was rejected, dropped, or lost:

TCP SYN/ACK

If a SYN/ACK packet is received, the port is considered open.

TCP RST/ACK

If a RST/ACK packet is received, the probe packet was either rejected by the target host or an upstream security device (e.g., a firewall with a reject rule in its policy).

ICMP type 3 code 13

If an ICMP type 3 code 13 message is received, the host (or a device such as a firewall) has administratively

4.6 Network Scanning Recap

Different IP network scanning methods allow you to test and effectively identify vulnerable network components. Here is a list of effective network scanning techniques and their applications:

ICMP scanning and probing

By launching an ICMP ping sweep, you can effectively identify poorly protected hosts (as security conscious administrators filter inbound ICMP messages) and perform a degree of operating-system fingerprinting and reconnaissance by analyzing responses to the ICMP probes.

Half-open SYN flag TCP port scanning

A SYN port scan is often the most effective type of port scan to launch directly against a target IP network space. SYN scanning is extremely fast, allowing you to scan large networks quickly.

Inverse TCP port scanning

Inverse scanning types (particularly FIN, Xmas, and NULL) take advantage of idiosyncrasies in certain TCP/IP stack implementations. This scanning type isn't effective when scanning large network spaces, although it is useful when testing and investigating the security of specific hosts and small network segments.

Third-party TCP port scanning

Using a combination of vulnerable network components and TCP spoofing, third-party TCP port scans can be effectively launched. Scanning in this fashion has two benefits: hiding the true source of a TCP scan and assessing the filters and levels of trust between hosts. Although time consuming to undertake, third-party scanning is extremely useful when applied correctly.

UDP port scanning

Identifying accessible UDP services can be undertaken easily only if ICMP type 3 code 3 (destination port unreachable) messages are allowed back through filtering mechanisms that protect target systems. UDP services can sometimes be used to gather useful data or directly compromise hosts (the DNS, SNMP, TFTP, and BOOTP services in particular).

IDS evasion and filter circumvention

Intrusion detection systems and other security mechanisms can be rendered ineffective by using multiple spoofed decoy hosts when scanning or by fragmenting probe packets using *nmap* or *fragroute*. Filters such as firewalls, routers, and even software (including the Microsoft IPsec filter) can sometimes be bypassed using specific source TCP or UDP ports, source routing, or stateful attacks.

4.7 Network Scanning Countermeasures

Here is a checklist of countermeasures to use when considering technical modifications to networks and filtering devices to reduce the effectiveness of network scanning and probing undertaken by attackers:

- Filter inbound ICMP message types at border routers and firewalls. This forces attackers to use full-blown TCP port scans against all of your IP addresses to map your network correctly.
- Filter all outbound ICMP type 3 unreachable messages at border routers and firewalls to prevent UDP port scanning and firewalking from being effective.
- Consider configuring Internet firewalls so that they can identify port scans and throttle the connections accordingly. You can configure commercial firewall appliances (such as those from Check Point, NetScreen, and WatchGuard) to prevent fast port scans and SYN floods being launched against your networks. On the open source side, there are many tools such as *portsentry* that can identify port scans and drop all packets from the source IP address for a given period of time.
- Assess the way that your network firewall and IDS devices handle fragmented IP packets by using *fragtest* and *fragroute* when performing scanning and probing exercises. Some devices crash or fail under conditions in which high volumes of fragmented packets are being processed.
- Ensure that your routing and filtering mechanisms (both firewalls and routers) can't be bypassed using specific source ports or source-routing techniques.
- If you house publicly accessible FTP services, ensure that your firewalls aren't vulnerable to stateful circumvention attacks relating to malformed PORT and PASV commands.
- If a commercial firewall is in use, ensure the following:
  - The latest service pack is installed.
  - Antispoofing rules have been correctly defined, so that the device doesn't accept packets with private spoofed source addresses on its external interfaces.
  - Fastmode services aren't used in Check Point Firewall-1 environments.
- Investigate using inbound proxy servers in your environment if you require a high level of security. A proxy server will not forward fragmented or malformed packets, so it isn't possible to launch FIN scanning or other stealth methods.
- Be aware of your own network configuration and its publicly accessible ports by launching TCP and UDP

Chapter 5. Assessing Remote Information Services

Remote information services can collect information for later use (such as username and internal IP address information) and run arbitrary commands on the target server by exploiting process manipulation vulnerabilities. This chapter focuses on the assessment of these services and lists relevant tools and techniques that can test and assure the security of your services.

5.1 Remote Information Services

Unix-based systems and various device platforms, such as Cisco IOS, run remote information services that provide system, user, and network details over IP. Such services can be probed to collate username listings and details of trusted networks and hosts, and, in some cases, compromise systems directly.

I derived a basic list of remote information services from the */etc/services* file:

```
systat          11/tcp
```

5.2 systat and netstat

The *systat* and *netstat* services are interesting because current network and system information can be found easily by connecting to the services using *telnet*. The */etc/inetd.conf* file on a system running *systat* and *netstat* typically includes the following lines:

```
systats stream  tcp  nowait  root /usr/bin/ps      ps -ef
```

5.3 DNS

In Chapter 3, I covered the use of Domain Name System querying to enumerate and map IP networks. This involves launching forward and reverse queries, along with DNS zone transfers. DNS servers use two ports to fulfill requests: UDP port 53 to serve standard direct requests (e.g., to resolve names to IP addresses and vice versa) and TCP port 53 to serve DNS information during a zone transfer.

To fully assess DNS services (to identify exploitable vulnerabilities and other risks) you must do the following:

- Retrieve DNS service version information
- Attempt to perform DNS zone transfers against known domains
- Attempt to perform mass reverse-lookup queries against internal address space
- Test for process manipulation vulnerabilities

5.3.1 Retrieving DNS Service Version Information

DNS server version information can be gleaned directly across UDP port 53 by issuing a version.bind chaos txt request through the Unix dig utility. In Example 5-3, BIND 9.2.1 is running against mail.hmgcc.gov.uk.

Example 5-3. Using dig to glean BIND version information

```
# dig @mail.hmgcc.gov.uk version.bind chaos txt
```

5.4 finger

The *fingerd* service is commonly found listening on TCP port 79 of Cisco IOS routers. Default out-of-box builds of many commercial Unix-based systems also run the service, including Solaris and BSDI.

The service can be queried using a *finger* client (found in most operating platforms) or by directly using *telnet* to connect to port 79. Two examples of this follow, in which I show the differences in results from querying a Cisco IOS device and a Solaris server.

Here's a *finger* query against a Cisco router using *telnet*:

```
# telnet 192.168.0.1 79
```

5.5 auth

The Unix *auth* service (known internally as *identd*) listens on TCP port 113. The primary purpose of *auth* is to provide a degree of authentication through mapping local usernames to TCP network ports in use. IRC is a good example of this: when a user connects to an IRC server, an *auth* request is sent to TCP port 113 of the host to retrieve the user's current login name.

The *identd* service can be queried in line with RFC 1413 to match open TCP ports on the target host with local usernames. The information gathered has two different uses to an attacker: to derive the owners of processes with open ports and to enumerate valid username details.

nmap has the capability to cross reference open ports with the *identd* service running on TCP port 113. Example 5-12 shows such an *identd* scan being run to identify a handful of user accounts.

Example 5-12. Finding service ownership details through identd

```
# nmap -I -sT 192.168.0.10
```

5.6 SNMP

The Simple Network Management Protocol (SNMP) service listens on UDP port 161. SNMP is often found running on network infrastructure devices such as managed switches, routers, and other appliances. Increasingly, SNMP can be found running on Unix-based and Windows servers for central network management purposes.

SNMP authentication is very simple and is sent across networks in plaintext. SNMP Management Information Base (MIB) data can be retrieved from a device by specifying the correct *read community string*, and SNMP MIB data can be written to a device using the correct *write community string*. MIB databases contain listings of Object Identifier (OID) values, such as routing table entries, network statistics, and details of network interfaces. Accessing a router MIB is useful when performing further network reconnaissance and mapping.

Two useful tools used by attackers and security consultants alike for brute-forcing SNMP community strings and accessing MIB databases are *ADMsnmp* and *snmpwalk*.

5.6.1 ADMsnmp

ADMsnmp is available from the ADM group home page at http://adm.freelsd.net/ADM/. The utility is an effective Unix command-line SNMP community string brute-force utility. Example 5-13 shows the tool in use against a Cisco router at 192.168.0.1 to find that the community string private has write access.

Example 5-13. ADMsnmp used to brute-force SNMP community strings

```
# ADMsnmp 192.168.0.1
```

5.7 LDAP

The Lightweight Directory Access Protocol (LDAP) service is commonly found running on Windows 2000 Active Directory, Exchange, and Lotus Domino servers. The system provides user directory information to clients. LDAP is highly extensible and widely supported by Apache, MS Exchange, Outlook, Netscape Communicator, and others.

5.7.1 Anonymous LDAP Access

You can query LDAP anonymously (although mileage varies depending on the server configuration) using the *ldp.exe* utility from the Microsoft Windows 2000 Support Tools Kit found on the Windows 2000 installation CD under the *\support\tools\* directory.

The *ldapsearch* tool is a simple Unix-based alternative to *ldp.exe* that's bundled with OpenLDAP (http://www.openldap.org). In Example 5-16, I use the tool to perform an anonymous LDAP search against 192.168.0.65 (a Lotus Domino server on Windows 2000).

Example 5-16. Searching the LDAP directory with ldapsearch

```
 # ldapsearch -h 192.168.0.65
```

5.8 rwho

The Unix *rwhod* service listens on UDP port 513. If found to be accessible, you can query it using the Unix *rwho* client utility to list current users who are logged into the remote host, as shown:

```
# rwho 192.168.189.120
```

5.9 RPC rusers

The Unix *rusers* service is a Remote Procedure Call (RPC) service that listens on a dynamic TCP port. The *rusers* client utility first connects to the RPC portmapper on TCP port 111, which returns the whereabouts of the *rusersd* service if it is active.

During initial TCP port scans, if the RPC portmapper service isn't found to be accessible, it is highly unlikely that *rusersd* will be accessible. If, however, TCP or UDP port 111 is found to be accessible, the *rpcinfo* client can check for the presence of *rusersd* and other accessible RPC services, as shown in Example 5-17.

Example 5-17. Enumerating RPC services with rpcinfo

```
# rpcinfo -p 192.168.0.50
```

5.10 Remote Information Services Countermeasures

- There is no reason to run *systat*, *netstat*, *fingerd*, *rwhod*, or *rusersd* services in any production environment; these services completely undermine security and offer little benefit.
- DNS should be accessible over TCP only if inbound DNS zone transfers are offered because standard DNS queries are served over UDP. Diligently check all publicly accessible hosts to ensure that unnecessary DNS services aren't publicly accessible.
- Most Linux *identd* packages are vulnerable to public and privately known attacks; therefore, refrain from running *identd* on mission-critical Linux servers.
- SNMP services running on both servers and devices should be configured with strong read and write access community strings to minimize brute-force password-grinding risk. Network filtering of SNMP services from the Internet and other untrusted networks ensures further resilience and blocks buffer overflow and other process-manipulation attacks.
- Ensure that your accessible LDAP and Windows 2000 AD Global Catalog services don't serve sensitive information to anonymous unauthenticated users. If LDAP or Global Catalog services are being run in a high-security environment, ensure that brute-force attacks aren't easily undertaken by logging failed authentication attempts.
- Always keep your publicly accessible services patched to prevent exploitation of process-manipulation vulnerabilities. Most DNS, SNMP, and LDAP vulnerabilities don't require an authenticated session to be exploited by a remote attacker.

Chapter 6. Assessing Web Services

This chapter focuses on the technical execution of web service assessment. These services are commonly accessible in corporate environments and over the Internet, and they require a high level of security assurance due to their public nature. In this chapter I discuss the techniques and tools used to fully test HTTP and HTTPS services, along with their enabled components, subsystems, and any custom-written code that may be present.

6.1 Web Services

The assessment of various web services and individual subsystems can fill its own book. Web services run over two protocols: HTTP (found on TCP port 80, and sometimes 81, 8080, and others) and HTTPS (an SSL-enhanced web service usually found on TCP port 443).

Many security consultants run simple CGI scanning tools (such as *whisker*) against web services, which doesn't fully identify and categorize all the risks at hand. In broad terms, professional assessment of web services involves the following five steps:

1. Identifying the web service running (such as IIS 4.0 or Apache 1.3.27)
2. Identifying subsystems and enabled components (such as FrontPage Extensions)
3. Investigating known vulnerabilities in the web service and its enabled components
4. Identifying and accessing poorly protected sensitive information
5. Assessing CGI scripts and custom ASP pages running server-side

Automated web service scanning tools, such as *nikto* (http://www.cirt.net/code/nikto.shtml)[1] and N-Stealth (http://www.nstalker.com/nstealth/), are good at performing Steps 1, 2, and 4. The Open Web Application Security Project (OWASP) site at http://www.owasp.org is a great source of tools and white papers discussing the assessment of custom CGI scripts and ASP pages for SQL injection and other vulnerabilities.

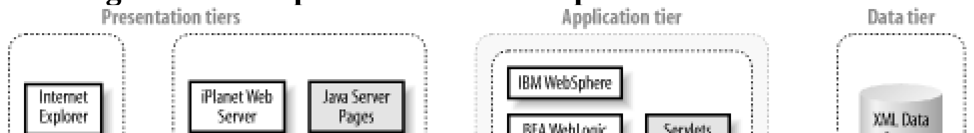
[1] URLs for tools in this book are mirrored at the O'Reilly site, http://examples.oreilly.com/networksa/tools.

All in all, basic web assessment can be automated. It is imperative, however, that you perform hands-on testing and qualification after automatically identifying all the obvious security flaws, especially when assessing complex environments such as custom-built e-commerce sites.

Buffer overflow and memory corruption vulnerabilities are difficult to identify in most environments, so you have to download exploit scripts and launch them against the target web server (following accurate service identification) to qualify vulnerabilities that manipulate areas of remote server memory.

To show how various components at presentation, application, and data tiers are often arranged in complex enterprise web environments, the OWASP team put together the diagram shown in Figure 6-1.

Figure 6-1. Components used in enterprise web environments



6.2 Identifying the Web Service

You can identify both standard plaintext and SSL web services through analyzing responses to simple HTTP methods such as HEAD and OPTIONS. Error pages can also determine the version and service pack level of IIS web servers. Many security-conscious system administrators modify the server-information field of their web services, so deeper analysis of responses is sometimes required.

6.2.1 HTTP HEAD

In Example 6-1, I use *telnet* to connect to www.trustmatta.com on port 80 and issue a HEAD / HTTP/1.0 request (followed by two carriage returns).

Example 6-1. Using the HTTP HEAD method against Apache

```
# telnet www.trustmatta.com 80
```

6.3 Identifying Subsystems and Components

Increasing numbers of exposures and vulnerabilities are identified in web-service subsystems and components used in complex environments. Here are some examples of popular subsystems that can be exploited to gain access to a target web server:

- ASP.NET
- WebDAV
- Microsoft FrontPage
- Microsoft Outlook Web Access (OWA)
- Default IIS ISAPI Extensions
- PHP
- OpenSSL

Through ascertaining the core web-service version and clear details of subsystems and enabled components, security analysts can properly investigate and qualify vulnerabilities and catalog exploit scripts to test later. What follows are examples and details of what to look for when identifying these subsystems.

6.3.1 ASP.NET

Microsoft IIS 5.0 and 6.0 servers can often be found running .NET framework components. If ASP.NET pages are in use (commonly with *.aspx* file extensions as opposed to *.asp*), H D Moore of Digital Defense, Inc., wrote the *dnascan.pl* utility to enumerate details of the ASP.NET subsystem and its configuration (http://www.digitaloffense.net/dnascan.pl.gz).

Example 6-10 shows the tool identifying the version of ASP.NET running on http://www.patchadvisor.com as 1.1.4322.573).

Example 6-10. Performing ASP.NET enumeration

```
# ./dnascan.pl http://www.patchadvisor.com
```

6.4 Investigating Web Service Vulnerabilities

You can search vulnerability information sites (such as MITRE CVE, SecurityFocus, and ISS X-Force) to investigate current web service vulnerabilities. Often vulnerabilities are described, but public working exploit scripts can't be found. Increasing numbers of vulnerabilities are exploitable only under certain circumstances, so full qualification is very important.

6.4.1 The Tools

N-Stealth (http://www.nstalker.com/nstealth/) and *nikto* (http://www.cirt.net/code/nikto.shtml) are two excellent tools for performing initial automated investigation of known web service vulnerabilities and issues.

When performing a full web-service assessment, it's best practice to perform service-identification tasks by hand and launch automated sweeps to check for known issues and obvious attack vectors. This information helps to build a clear picture of the server and its configuration, enabling efficient investigation and testing of vulnerabilities.

6.4.1.1 nikto

nikto is a Perl script that can be run from Unix-like environments, as well as Windows and other platforms. Example 6-13 shows *nikto* being launched against an IIS 4.0 server with no obvious serious vulnerabilities.

Example 6-13. nikto in use against www.example.org

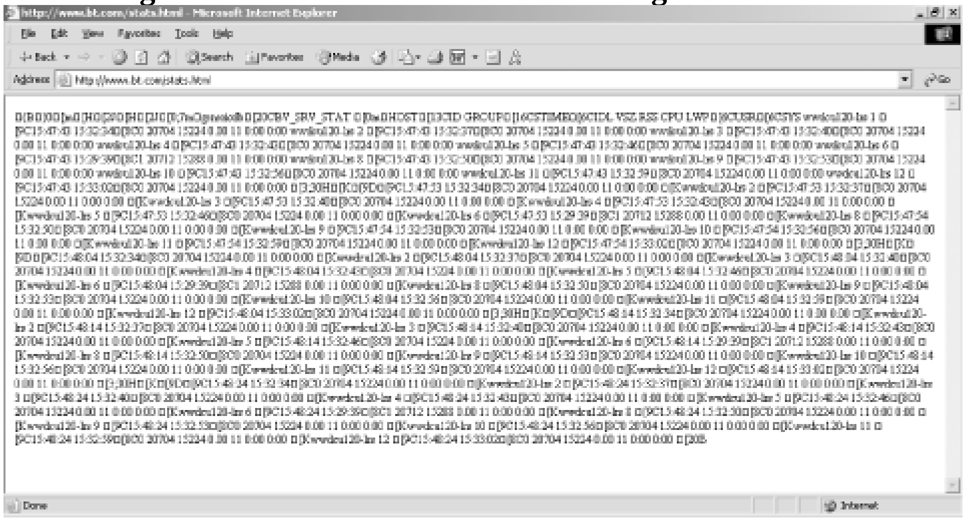
```
# perl nikto.pl -host www.example.org
```

6.5 Accessing Poorly Protected Information

You can find server backup files and other sensitive data if you look hard enough. I know a handful of cases in which administrators have set up private areas of web service space to store such files, with predictable directory names (for example, */backup*, */private*, or */test*). In one such instance, I downloaded a 500-MB backup image of a Linux web server, containing the */etc/passwd*, */etc/shadow*, and other useful system files.

Automated web service scanning tools are proficient at identifying these obvious file locations and directories. The *stats.html* page on the BT corporate web site reveals potentially sensitive information You can find it at http://www.bt.com/stats.html and see it in Figure 6-15.

Figure 6-15. The BT web site reveals usage information


http://www.bt.com/stats.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.bt.com/stats.html
Go
Done
Internet

```
[illegible]
```

A casual look through this page reveals a table with column headings of HOST, GROUP, TIME, and CPU.

6.5.1 Brute-Forcing HTTP Authentication

When assessing large environments, analysts often encounter basic HTTP authentication prompts. By launching brute-force password-grinding attacks against these authentication mechanisms, an attacker can gain access to potentially sensitive information or system components (web application back-end management systems, etc.).

In particular, the Brutus and Hydra brute-force tools are exceptionally good at launching parallel brute-force password grinding attacks against web authentication mechanisms. The tools are available from the following locations and are discussed throughout this book with working examples:
http://www.hoobie.net/brutus/brutus-download.htmlhttp://www.thc.org/releases.php

6.6 Assessing CGI Scripts and Custom ASP Pages

All the large e-commerce and online banking environments that I compromised recently have been broken through abuse of custom-written ASP and CGI scripts. Nowadays, it is relatively straightforward for a company to perform its own initial security testing to negate all the obvious risks that are exploited by propagating worms and opportunistic attackers, but application-level risks are left unchecked.

Custom scripts and ASP pages that create dynamic e-commerce and online banking environments are difficult to secure. Often it is the case that the developer who has written the ASP pages has little secure-programming experience, so doesn't truly understand the threats posed to his code from determined attackers.

Sometimes there are very small vulnerabilities in web environments, which can be combined to result in a full compromise. For example, you can combine filter evasion with an unbounded file call to create an ASP page or CGI script in an executable directory, and then call it directly.

The Apache Foundation web site (http://www.apache.org) was compromised in a similar way in May 2000: an attacker uploaded a simple PHP script to a directory under ftp.apache.org that also existed on the web site. From there, the PHP script was called and executed, creating a backdoor to the Apache web server. MySQL was found running as root, so it wasn't long before the attacker had super-user privileges and defaced the site. For a more in-depth analysis of this compromise, check out:
http://www.securityfocus.com/archive/1/58478http://www.dataloss.net/papers/how.defaced.apache.org.txt
http://www.attrition.org/mirror/attrition/2000/05/03/www.apache.org

Defense in depth is required to prevent compromise from determined attackers who exploit a series of small vulnerabilities. In the Apache case, MySQL shouldn't have been running as root, and the FTP directory structure should not have been directly linked to the accessible web site.

I discuss individual vulnerabilities that are often exploited to gain various degrees of access to a web environment. From an external perspective, the following major threats are present:

- Parameter manipulation and filter evasion
- SQL and operating system command injection
- Error-handling problems

The Open Web Application Security Project (OWASP) team have written some excellent papers that cover in-depth testing and resolution of problems within custom-built web applications. If you require further information relating to web application security (such as details of cross-site scripting vulnerabilities or issues relating to access control), check out http://www.owasp.org.

6.6.1 Parameter Manipulation and Filter Evasion

When providing input to a script or web application running server-side (such as a search engine, feedback form, or

PREV < Day Day Up > NEXT

6.7 Web Services Countermeasures

- You should ensure that all Internet-based server software and components (Microsoft IIS, Apache, OpenSSL, PHP, *mod_perl*, etc.) have up-to-date patches and are configured to prevent known public exploits and attack techniques from being successful.
- If you don't use script languages (such as PHP or Perl) in your web environment, ensure that associated Apache components such as *mod_perl* and PHP are disabled. Increasingly, vulnerabilities in these subsystems are being identified as attackers find fewer bugs in core server software.
- Many buffer overflow exploits use connect-back shellcode to spawn a command shell and connect back to the attacker's IP address on a specific port. In a high security environment I recommend using aggressive firewalling to prevent unnecessary outbound connections (so that web servers can send traffic outbound only from TCP port 80, for example). In the event of new vulnerabilities being exploited, good egress network filtering can flag suspicious outbound connections from your web servers and buy you time.
- Prevent indexing of accessible directories if no index files are present (e.g., *default.asp*, *index.htm*, *index.html*, etc.) to prevent web crawlers and opportunistic attackers from compromising sensitive information.

Here are some database and custom-written web application recommendations:

- Don't expose debugging information to public web users if a crash or application exception occurs within your web server or application-tier software.
- If you use backend SQL databases, tie down the SQL user accounts used by public web servers so that they have limited access to potentially damaging stored procedures (if any) and have decent permissions relating to reading and writing of fields and tables from the database.

Here are Microsoft IIS and Outlook Web Access-specific recommendations:

- Microsoft has published security checklists and tools for best practice IIS configuration, including URLscan and the IIS lockdown tool available from http://www.microsoft.com/technet/security/tools/tools.asp.
- Under IIS, ensure that unnecessary ISAPI extension mappings are removed (such as *.ida*, *.idq*, *.htw*, *.htr*, and *.printer*).
- Don't run Outlook Web Access at a predictable web location (for instance, */owa*, */exchange*, or */mail*), and use SSL in high-security environments to prevent eavesdropping. Ideally, remote access to Exchange and other services should be provided through a VPN tunnel.
-

Chapter 7. Assessing Remote Maintenance Services

This chapter covers the assessment of remote maintenance services that give direct access to servers and devices for administrative purposes. Common remote maintenance services include SSH, Telnet, X Windows, VNC, and Microsoft Terminal Services. Determined attackers concentrate on breaking remote maintenance services to obtain direct access to the target host.

7.1 Remote Maintenance Services

Services used by network administrators to directly manage remote hosts over TCP/IP (e.g., SSH, Telnet, VNC, and others) are threatened by three categories of attack:

- Information leak attacks, from which user and system details are extracted
- Process manipulation attacks (buffer overflows, format string bugs, etc.)
- Brute-force guessing of user passwords to gain direct system access

An online bank may be running the Telnet service on its Internet routers for administrative purposes. This service may not be vulnerable to information leak or process-manipulation attacks, but a determined attacker can launch a brute-force attack against the service to gain access. Brute force is an increasingly popular attack vector for attackers attempting to break moderately secure networks.

I have derived this list of common remote maintenance services from the */etc/services* file:

```
ssh               22/tcp
```

7.2 SSH

Secure Shell (SSH) provides encrypted access to Unix and Win32 command shells. Weaknesses in plaintext services such as Telnet were often abused by attackers to compromise networks, so SSH was introduced to provide encrypted access to Unix-based hosts for maintenance purposes.

Before 1999, the only SSH servers available were for commercial use and provided by SSH Communications (http://www.ssh.com) and F-Secure (http://www.f-secure.com). In late 1999, the OpenBSD team worked to provide SSH support in Version 2.6 of their operating system, and OpenSSH 1.2.2 was born. Commercial versions provided by SSH Communications and F-Secure remain supported and are sold, but OpenSSH has proved to be extremely popular and is now included with most Linux distributions.

Due to its cryptographic nature, an SSH client is required to connect to and authenticate with SSH. The free OpenSSH package can be downloaded from http://www.openssh.com.

For Windows users, PuTTY is a freely downloadable tool available with a host of other SSH client utilities (including PSCP, PSFTP, and Plink) available from http://www.chiark.greenend.org.uk/~sgtatham/putty/.

7.2.1 SSH Fingerprinting

To correctly ascertain vulnerabilities that may be present in the target SSH service, first perform banner grabbing by using *telnet* or *nc* to connect to the SSH service. Example 7-1 shows how *telnet* can do this: the banner shows the host is running OpenSSH 3.5 patch level 1 using the SSH 2.0 protocol.

Example 7-1. Grabbing the SSH service banner using telnet

```
# telnet 192.168.0.80 22
```

7.3 Telnet

Telnet is a plaintext remote management service that provides command-line access to multiple operating systems including Unix, VAX/VMS, Windows NT, and devices such as Cisco routers and managed switches.

From a security perspective, the Telnet protocol is weak because authentication details are transmitted in plaintext and can be sniffed by determined attackers. When authenticated users are connected through Telnet, their sessions can also be hijacked and commands injected to the underlying operating system by attackers with access to the same network segment.

7.3.1 Telnet Service Fingerprinting

From a remote Internet-based perspective, you can use automated software, such as *telnetfp*, to fingerprint Telnet services. A second approach is to manually grab the service banner and compare it with a known list of responses. I discuss these two approaches with practical examples.

7.3.1.1 telnetfp

You can use *telnetfp* to accurately fingerprint the Telnet services of Windows, Solaris, Linux, BSD, SCO, Cisco, Bay Networks, and other operating platforms, based on low-level responses. The tool even has a scoring system to guess the service if an exact match isn't seen. *telnetfp* can be downloaded from http://packetstormsecurity.org/groups/teso/telnetfp_0.1.2.tar.gz.

After downloading and compiling the tool, you can run it as follows:

```
# ./telnetfp
```

7.4 R-Services

Unix r-services are common to commercial platforms, including Solaris, HP-UX, and AIX. I have assembled a list from the */etc/services* file as follows:

```
exec             512/tcp
```

7.5 X Windows

X Windows is commonly used by most major Unix-like operating systems as the underlying system for displaying graphical applications. For example, Gnome, KDE, and applications including *xterm* and *ghostview* run using the X Windows protocol.

X Windows was developed at MIT in 1984, with Version 11 first released in 1987. The X Window system is currently at release six of Version 11 (commonly referred to as X11R6). Over the past few years since release two, the X Window system has been maintained by the X Consortium, an association of manufacturers supporting the X standard.

7.5.1 X Windows Authentication

X servers listen on TCP ports 6000 to 6063 (depending on the number of concurrent displays). Most of the time users simply access their local X server, although X can be accessed over a network for remote use. The two authentication mechanisms within X Windows are *xhost* and *xauth*, which I discuss in the following sections.

7.5.1.1 xhost

Host-based X authentication allows users to specify which IP addresses and hosts have access to the X server. The *xhost* command is used with + and - options to allow and deny X access from individual hosts (i.e., xhost +192.168.189.4). If a + option is used with no address, any remote host can access the X server.

xhost authentication is dangerous and doesn't provide the granularity required in complex environments. By issuing an xhost - command, host-based authentication is disabled, and only local access is granted.

7.5.1.2 xauth

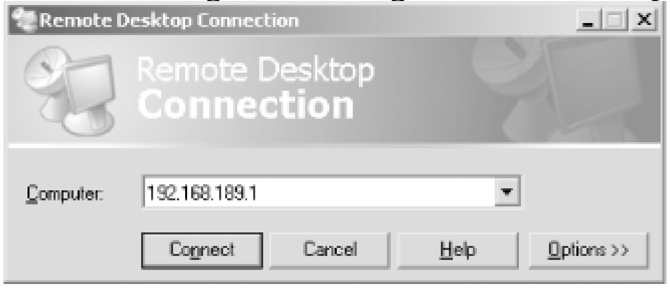
When a legitimate user logs in locally to X Windows, a magic cookie is placed into the *.Xauthority* file under the user's home directory. The *.Xauthority* file contains one cookie for each X display the user can use, which can be manipulated using the *xauth* utility as shown here:

```
# xauth list
```

7.6 Microsoft Remote Desktop Protocol

Remote Desktop Protocol (RDP, also known as Microsoft Terminal Services) provides thin client access to the Windows desktop. The Windows 2000, XP, and 2003 Server platforms usually run these services. The RDP service runs by default on TCP port 3389, accessed using the remote desktop client as shown in Figure 7-2.

Figure 7-2. Connecting to RDP using the remote desktop client



Computer: 192.168.189.1

The Microsoft RDP client is available at http://download.microsoft.com/download/whistler/tools/1.0/wxp/en-us/msrdpcli.exe.

7.6.1 RDP Brute-Force Password Grinding

After locating accessible RDP servers (by port scanning for TCP 3389) and performing enumeration through anonymous NetBIOS sessions (see Chapter 9) to identify potentially weak user accounts, an attacker can launch brute-force password-grinding attacks. The Administrator account is usually a good place to start because it can't be locked locally upon multiple failed logon attempts.

Tim Mullen (http://www.hammerofgod.com) put together a useful tool called *tsgrinder* for brute-forcing terminal services. *tsgrinder* (Version 2.03 at the time of writing) is available at http://www.hammerofgod.com/download.htm.

Example 7-16 shows the *tsgrinder* usage from a Win32 command prompt.

Example 7-16. Using tsgrinder

```
D:\tsgrinder> tsgrinder
```

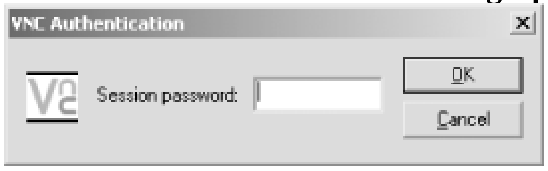
7.7 VNC

AT&T's Virtual Network Computing (VNC) package is available from http://www.uk.research.att.com/vnc/. VNC is a free and simple remote desktop access system for Windows, and it runs over the following TCP ports:

- Port 5800 for HTTP access to VNC using a Java client through a web browser
- Port 5900 for direct access to VNC using the native *vncviewer.exe*

From a security perspective, VNC is relatively straightforward to compromise. A major issue with VNC security is its authentication mechanism, shown in Figure 7-3.

Figure 7-3. VNC authentication relies on a single password



VNC requires only one piece of data for authentication purposes: a session password with a maximum length of eight characters. On the target server, the VNC password string is stored in the Windows registry under the following keys:

```
\HKEY_CURRENT_USER\Software\ORL\WinVNC3
```

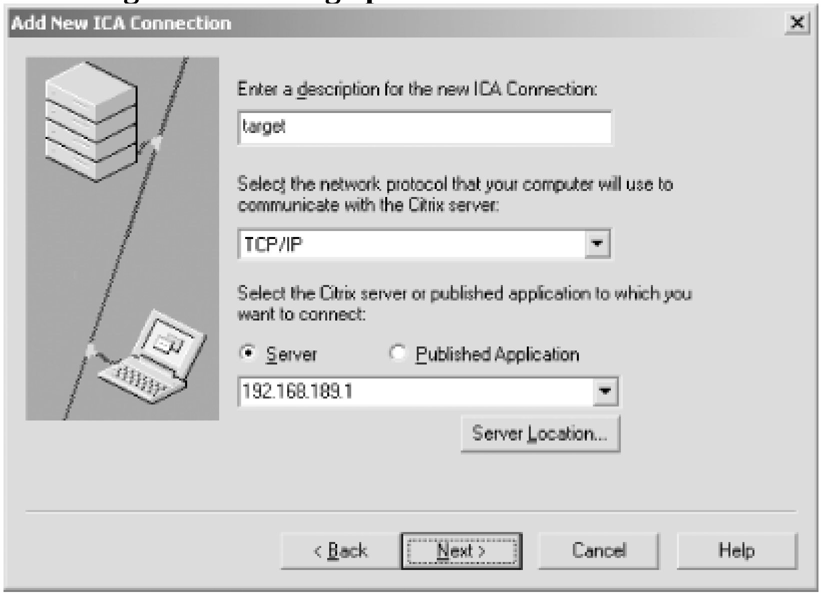
7.8 Citrix

Citrix is a scalable thin-client Windows service that is accessed directly through TCP port 1494 server-side. The protocol that Citrix uses is known as Independent Computing Architecture (ICA). After finding a server with TCP port 1494 open, you should use a Citrix ICA client to connect to the service for further investigation (available from http://www.citrix.com/download/ica_clients.asp).

7.8.1 Using The Citrix ICA Client

When you run the client software, you should add a new ICA connection, using TCP/IP to communicate with the server and provide the IP address of the host with port 1494 open as in Figure 7-4.

Figure 7-4. Setting up the ICA client to connect



Add New ICA Connection

Enter a description for the new ICA Connection: target

Select the network protocol that your computer will use to communicate with the Citrix server: TCP/IP

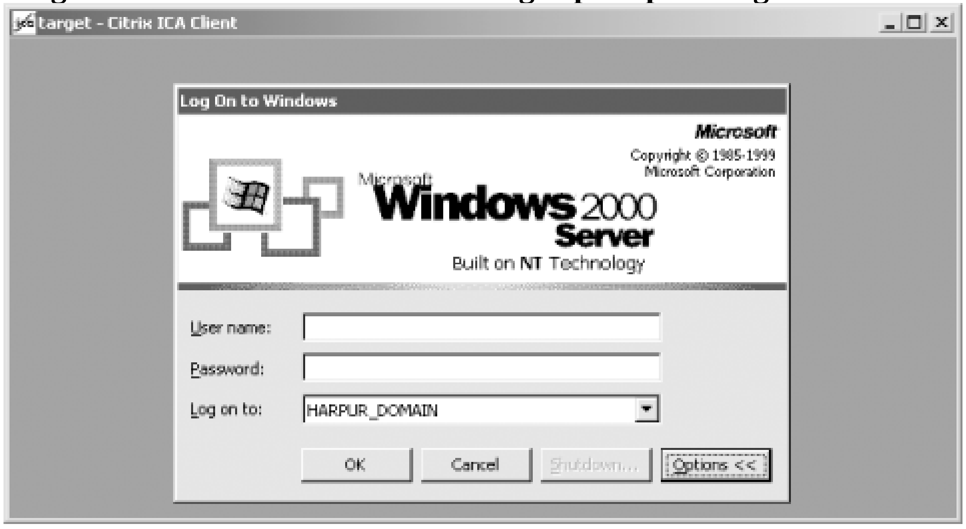
Select the Citrix server or published application to which you want to connect:

[✓ Server] Published Application

192.168.189.1

Username, password, and application details can all be left blank if you have no insight into the Citrix configuration. Upon entering the details correctly and connecting, a login screen like that shown in Figure 7-5 (depending on the server configuration) appears.

Figure 7-5. A Windows 2000 Server logon prompt through Citrix ICA



target - Citrix ICA Client

Log On to Windows

Microsoft
Copyright © 1985-1999
Microsoft Corporation

Microsoft Windows 2000 Server
Built on NT Technology

User name:
Password:
Log on to: HARPUR_DOMAIN

In some instances, you log into a Windows desktop environment with access to published applications such as

7.9 Remote Maintenance Services Countermeasures

- Don't run Telnet services on publicly accessible devices. Cisco IOS and decent appliance servers and operating platforms can run either SSH or OpenSSH (http://www.openssh.com).
- Ensure resilience of your remote maintenance services from brute-force password guessing attacks. Ideally, this involves setting account lockout thresholds and enforcing a good password policy.
- Don't run r-services (*rsh*, *rexec*, or *rlogin*) because they are vulnerable to spoofing attacks, use very weak authentication, and are plaintext.
- In secure environments, don't use services such as VNC because they have weak authentication, and determined attackers can compromise them. You should use Microsoft RDP and Citrix ICA services with Secure Socket Layer (SSL) encryption to prevent sniffing and hijacking attacks.
- Read the guide to hardening terminal services that's published by Microsoft (http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/deploy/part4/chapt-16.asp).
- To improve authentication and completely negate brute-force attacks, use two-factor authentication mechanisms such as Secure Computing Safeword and RSA SecurID. These solutions aren't cheap, but they can be useful when authenticating administrative users accessing critical servers.

Chapter 8. Assessing FTP and Database Services

This chapter focuses on the remote assessment of FTP and SQL database services used in most corporate networks to facilitate file distribution and central storage of data. If these services aren't configured or protected correctly at both application and network levels, they can be used to great effect to compromise networks and sensitive data.

8.1 FTP

File Transfer Protocol (FTP) services are usually found running on servers to provide public access to files over IP. The FTP service uses the following two ports to function in its native mode:

TCP port 21

The inbound control port, used to receive and process FTP commands

TCP port 20

The outbound data port, used to send data from the server to client

Historically, the way that FTP services have been used to compromise networks include:

- Brute-force guessing of user passwords to gain direct access
- FTP bounce port-scanning and exploit payload delivery
- Issuing crafted FTP commands to circumvent stateful filtering devices
- Process manipulation, including overflow attacks involving malformed data

I will discuss each of these attack types; however, the first task to undertake when identifying an accessible FTP service is that of enumeration, to ascertain the FTP service that's running and its configuration.

8.2 FTP Banner Grabbing and Enumeration

When finding a server running FTP, the first piece of information discovered by connecting to the service is the FTP server banner:

```
# ftp 192.168.0.11
```

8.3 FTP Brute-Force Password Guessing

Many tools are available to perform FTP brute-force password-guessing attacks. As discussed earlier, Hydra is a Unix-based brute-force utility for FTP, POP3, IMAP, HTTP, LDAP, and many other services; Brutus is a similar Windows tool. The tools are available at:[1]

[1] URLs for tools in this book are mirrored at the O'Reilly site, http://examples.oreilly.com/networksa/tools.

http://www.thc.org/releases.phphttp://www.hoobie.net/brutus/brutus-download.html

8.4 FTP Bounce Attacks

As outlined in Chapter 4, FTP services bundled with the following operating platforms are vulnerable to bounce attacks in which port scans or malformed data can be sent to arbitrary locations via FTP:

- FreeBSD 2.1.7 and earlier
- HP-UX 10.10 and earlier
- Solaris 2.6 / SunOS 5.6 and earlier
- SunOS 4.1.4 and earlier
- SCO OpenServer 5.0.4 and earlier
- SCO UnixWare 2.1 and earlier
- IBM AIX 4.3 and earlier
- Caldera Linux 1.2 and earlier
- Red Hat Linux 4.2 and earlier
- Slackware 3.3 and earlier
- Any Linux distribution running WU-FTPD 2.4.2-BETA-16 or earlier

If you know that an accessible FTP service is running on an internal network and is accessible through NAT, bounce attacks can be used to probe and attack other internal hosts, and even the server running the FTP service itself.

8.4.1 FTP Bounce Port Scanning

You can use the *nmap* port scanner in Unix and Windows environments to perform an FTP bounce port scan, using the -P0 and -b flags in the following manner:

```
nmap -P0 -b username:password@ftp-server:port <target host>
```

Example 8-5 shows an FTP bounce port scan being launched through the Internet-based 142.51.17.230 to scan an internal host at 192.168.0.5, a known address previously enumerated through DNS querying.

Example 8-5. FTP bounce scanning with nmap

8.5 Circumventing Stateful Filters Using FTP

At Black Hat Briefings 2000 in Las Vegas, Thomas Lopatic, John McDonald, and Dug Song presented "A Stateful Inspection of Firewall-1" (available as a Real Media video stream and Powerpoint presentation from http://www.blackhat.com/html/bh-usa-00/bh-usa-00-speakers.html), which documented a raft of security issues with Checkpoint Firewall-1 4.0 SP4. One area covered was abusing FTP access to a host through a stateful firewall in order to open ports and gain access to services that should otherwise be filtered.

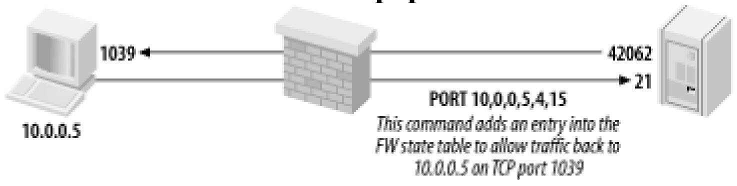
By its very specification, FTP is a complex protocol used to transfer files that have two channels: the control channel (using TCP port 21) and the data channel (using TCP port 20). The PORT and PASV commands are issued across the control channel to determine which dynamic high ports (above 1024) are used to transfer and receive data.

8.5.1 PORT and PASV

The PORT command defines a dynamic high port from which the client system receives data. Most firewalls perform stateful inspection of FTP sessions, so the PORT command populates the state table.

Figure 8-3 shows a client system that connects to an FTP server through a firewall and issues a PORT command to receive data. A short explanation of the command follows.

Figure 8-3. The PORT command populates the firewall state table

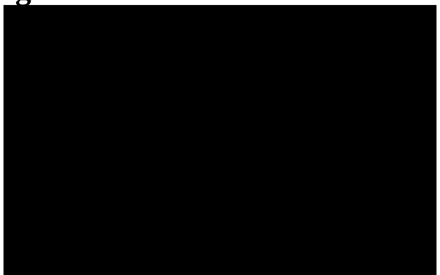

1039
10.0.0.5
42062
21
PORT 10,0,0,5,4,15
This command adds an entry into the FW state table to allow traffic back to 10.0.0.5 on TCP port 1039

The reason that port 1039 is opened is because the last two digits in the PORT command argument (4 and 15) are first converted to hexadecimal:

- 4 becomes 0x04
- 15 becomes 0x0F

The two values then concatenate to become 0x040F, and a tool such as the Base Converter application found in Hex Workshop (available from http://www.bpsoft.com) is used to find the decimal value, as shown in Figure 8-4.

Figure 8-4. Converting the concatenated hex value to a port number



8.6 FTP Process Manipulation Attacks

If an attacker can accurately identify the target FTP service and the operating platform and architecture of the target server, it is relatively straightforward to identify and launch process-manipulation attacks to gain access to the server.

Most serious remote buffer overflows in FTP services are post-authentication issues; they require authenticated access to the FTP service and its underlying commands. Increasingly, write access is also required to create complex directory structures server-side that allow exploitation.

8.6.1 Solaris and BSD FTP Globbing Issues

The following glob( ) bug is present in default Solaris installations up to Solaris 8. By issuing a series of CWD *~username* requests, an attacker can effectively enumerate valid user accounts without even logging into the FTP server. This issue is described in detail at http://www.iss.net/security_center/static/6332.php and demonstrated in Example 8-6.

Example 8-6. Exploiting Solaris FTP glob( ) issues remotely

```
# telnet 192.168.0.12 21
```

8.7 FTP Services Countermeasures

- Don't provide anonymous FTP access, especially anonymous writable FTP access. Most serious overflows in FTP services require a degree of access to the server, in order to overflow nested functions within the program.
- Ensure aggressive firewalling both into and out of your public FTP servers. Most publicly available exploits use connect-back or bindshell shellcode, which allow attackers to compromise your server if it isn't fully protected at network level. If possible, avoid running other public network services (for example, web or mail services) on the same machine as an FTP server.
- If you offer public FTP access, ensure that your firewall is patched with the latest vendor service pack or security hot fixes; this will defuse any circumvention attacks.

8.8 Database Services

Three popular SQL database services that are often found in small, medium, and large network environments are Microsoft SQL Server, Oracle, and MySQL, accessible through the following network ports:

```
ms-sql          1433/tcp
```

8.9 Microsoft SQL Server

The Microsoft SQL Server service can be found running by default on TCP port 1433. Sometimes I find that the SQL Server service is running in hidden mode, accessible via TCP port 2433. Yes, this is what Microsoft means by hidden.

The SQL Server Resolution Service (SSRS) was introduced in Microsoft SQL Server 2000 to provide referral services for multiple server instances running on the same machine. The service listens for requests on UDP port 1434 and returns the IP address and port number of the SQL server instance that provides access to the requested database.

8.9.1 SQL Server Enumeration

You can use Chip Andrews' *sqlping* Win32 command-line utility to enumerate SQL Server details through the SSRS port (UDP 1434). The *sqlping* tool is available at http://www.sqlsecurity.com/uploads/sqlping.zip.

Example 8-11 shows the *sqlping* utility in use against a SQL 2000 Server, revealing the server name, database instance name, clustering information, along with version details and network port/named pipe information.

Example 8-11. Using sqlping to enumerate a Microsoft SQL Server

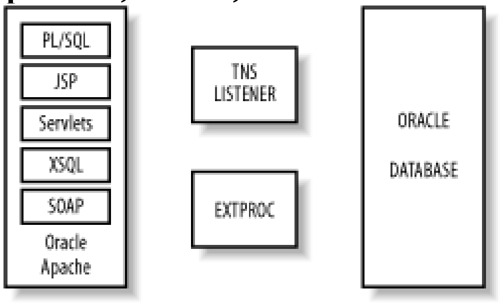
```
D:\SQL> sqlping 192.168.0.51
```

8.10 Oracle

Here I describe user and database enumeration techniques, password grinding, and remote buffer overflow attacks launchable against the Oracle TNS Listener service.

The Transparent Network Substrate (TNS) protocol is used by Oracle clients to connect to database instances via the TNS Listener service. The service listens on TCP port 1521 by default (although it is sometimes found on ports 1526 or 1541) and acts as a proxy between database instances and the client system. Figure 8-6 shows an example Oracle web application architecture.

Figure 8-6. Application, listener, and backend Oracle components



8.10.1 TNS Listener Enumeration and Information Leak Attacks

The listener service has its own authentication mechanism and is controlled and administered outside the Oracle database. In its default configuration, the listener service has no authentication set, which allows commands and tasks to be executed outside the database.

tnscmd.pl is an excellent tool you can use to interact with the TNS Listener. It's a Perl script that's available at http://www.jammed.com/~jwa/hacks/security/tnscmd/tnscmd.

8.10.1.1 Pinging the TNS Listener

You can use *tnscmd.pl* to issue various commands to the TNS Listener service. Example 8-15 shows the default *ping* command being issued to the listener to solicit a response.

Example 8-15. Pinging the TNS Listener using tnscmd

```
# perl tnscmd.pl -h 192.168.189.45
```

8.11 MySQL

MySQL is commonly found running on TCP port 3306 on Linux and FreeBSD servers. The database is relatively straightforward to administer, with a much simpler access model than the heavyweight, but more scalable, Oracle.

8.11.1 MySQL Enumeration

The version of the target MySQL database can be easily gleaned simply by using *nc* or *telnet* to connect to port 3306 and analyzing the string received, as shown here:

```
# telnet 10.0.0.8 3306
```

8.12 Database Services Countermeasures

- Ensure that database user passwords (*sa* and *probe* accounts found in Microsoft SQL Server, *root* under MySQL, etc.) are adequately strong.
- Filter and control public Internet-based access to database service ports to prevent determined attackers from launching brute-force password-grinding attacks in particular. In the case of Oracle with the TNS Listener, this point is extremely important.
- Don't run publicly accessible remote maintenance services on database servers; you will thus deter Oracle TNS Listener user *.rhosts* file creation and other types of grappling-hook attacks. If possible, use two-factor authentication for remote access from specific staging hosts; with public keys, use something like SSH .
- If SQL services are accessible from the Internet or other untrusted networks, ensure they are patched with the latest service packs and security hot fixes to ensure resilience from buffer overflows and other types of remote attack.

Chapter 9. Assessing Windows Networking Services

This chapter focuses squarely on Windows NetBIOS and CIFS services that are used in corporate networks to provide access to SMB for file sharing, printing, and other useful functions. If these services aren't configured or protected correctly by network filtering devices, they can be used to great effect to enumerate system details and cause a complete network compromise.

9.1 Microsoft Windows Networking Services

Microsoft Windows networking services use the following ports:

```
 loc-srv         135/tcp
```

9.2 Microsoft RPC Services

The Microsoft RPC endpoint mapper (also known as the DCE locator service) listens on both TCP and UDP port 135, and works much like the Sun RPC portmapper service found in Unix environments. Examples of Microsoft applications and services that use port 135 for endpoint mapping include Outlook, Exchange, and the Messenger Service.

> Depending on the host configuration, the RPC endpoint mapper can be accessed through TCP and UDP port 135, via SMB with a null or authenticated session (TCP 139 and 445), and as a web service listening on TCP port 593. For more information, see Todd Sabin's presentation titled "Windows 2000, NULL Sessions and MSRPC". Look for it at http://razor.bindview.com/publish/presentations/files/nullsess.ppt.

Through the Microsoft RPC service, you can attempt to:

- Enumerate system information, including IP addresses of network interfaces
- Gather user details via the SAMR and LSARPC interfaces
- Brute-force passwords of users in the Administrators group
- Execute commands through the Task Scheduler interface
- Run arbitrary code or crash the host entirely (through overflow issues)

Following is a breakdown of these exposures, along with details of tools and techniques you can adopt to assess MSRPC services properly.

9.2.1 Enumerating System Information

Through the RPC endpoint mapper, you can enumerate IP addresses of network interfaces (which will sometimes reveal internal network information), along with details of RPC services using dynamic high ports. The following four tools can mine information from the endpoint mapper:[1]

[1] URLs for tools in this book are mirrored at the O'Reilly site, http://examples.oreilly.com/networksa/tools.

epdump

http://www.packetstormsecurity.org/NT/audit/epdump.zip

9.3 The NetBIOS Name Service

The NetBIOS name service is accessible through UDP port 137. In particular the service can process NetBIOS Name Table (NBT) requests, commonly found in environments where Windows is being used along with workgroups, domains, or active directory components.

9.3.1 Enumerating System Details

You can easily enumerate the following system details by querying the name service:

- NetBIOS hostname
- The domain of which the system is a member
- Authenticated users currently using the system
- Accessible network interface MAC addresses

The inbuilt Windows *nbtstat* command can enumerate these details remotely. Example 9-13 shows how it can be run against 192.168.189.1.

Example 9-13. Using nbtstat to dump the NetBIOS name table

```
C:\> nbtstat -A 192.168.189.1
```

9.4 The NetBIOS Datagram Service

The NetBIOS datagram service is accessible through UDP port 138. As the NetBIOS name service is vulnerable to various naming attacks (resulting in denial of service in some cases), so can the NetBIOS datagram service be used to manipulate the target host and its NetBIOS services.

Anthony Osborne of PGP COVERT labs published an advisory in August 2000 that documented a NetBIOS name cache corruption attack that can be launched by sending crafted UDP datagrams to port 138. The full advisory is available at http://www.securityfocus.com/advisories/2556.

RFC 1002 defines the way in which Windows NetBIOS host information is encapsulated within the NetBIOS datagram header. When a browse frame request is received (on UDP port 138), Windows extracts the information from the datagram header and stores it in the NetBIOS name cache. In particular, the source NetBIOS name and IP address are blindly extracted from the datagram header and inserted into the cache.

A useful scenario in which to undertake this attack would be to send a crafted NetBIOS datagram to the target host, that mapped a known NetBIOS name on the internal network (such as a domain controller) to your IP address. When the target host attempts to connect to the server by its NetBIOS name, it instead connects to your IP address. An attacker can run SMBRelay or LC4 to capture rogue SMB password hashes in this scenario (which he can then crack and use to access other hosts).

Interestingly, Microsoft didn't release a patch for this issue: due to the unauthenticated nature of NetBIOS naming, it's a fundamental vulnerability! The MITRE CVE contains good background information within CVE-2000-1079.

9.5 The NetBIOS Session Service

The NetBIOS session service is accessible through TCP port 139. In particular, the service facilitates authentication across a Windows workgroup or domain, and provides access to resources (such as files and printers). You can perform the following attacks against the NetBIOS session service:

- Enumerate details of users, shared folders, security policies, and domain information
- Brute-force user passwords

After authenticating with the NetBIOS session service as a privileged user, you can:

- Upload and download files and programs
- Schedule and run arbitrary commands on the target host
- Access the registry and modify keys
- Access the SAM password database for cracking

> The CESG CHECK guidelines specify that candidates should be able to enumerate system details through NetBIOS (including users, groups, shares, domains, domain controllers, and password policies), including user enumeration through RID cycling. After enumerating system information, candidates are required to brute-force valid user passwords and access the filesystem and registry of the remote host upon authenticating.

9.5.1 Enumerating System Details

Various tools can enumerate sensitive information from a target Windows host with TCP port 139 open. Information can be collected either anonymously by initiating what is known as a *null session*, or through knowledge of a valid username and password. A null session is when you authenticate with the IPC$ share of the target host in the following manner:

```
net use \\target\IPC$ "" /user: ""
```

By specifying a null username and password, you gain anonymous access to IPC$. By default, Windows NT family hosts allow anonymous access to system and network information through NetBIOS, so the following can be gleaned:

-

9.6 The CIFS Service

The Common Internet File System (CIFS) is found running on Windows 2000, XP, and 2003 hosts through both TCP and UDP port 445. CIFS is the native mode for SMB access within these operating systems, but NetBIOS access is provided for backward compatibility.

Through CIFS, you can perform exactly the same tests as with the NetBIOS session service, including enumeration of user and system details, brute-force of user passwords, and system access upon authenticating (such as file access and execution of arbitrary commands).

9.6.1 CIFS Enumeration

In the same way that system and user information can be gathered through accessing SMB services through NetBIOS, CIFS can be directly queried to enumerate the same information: you just need the right tools for the job.

The SMB Auditing Tool (SMB-AT) is a suite of useful utilities, available as Win32 executables and source code (for compilation on Linux and BSD platforms in particular) from http://www.cqure.net.

9.6.1.1 User enumeration through smbdumpusers

The *smbdumpusers* utility is a highly versatile Windows NT user enumeration tool that can query SMB through both NetBIOS session (TCP 139) and CIFS (TCP 445) services. A second useful feature is the way the utility can enumerate users through a direct dump that works with RestrictAnonymous=0, but also using the RID cycling technique that can evade RestrictAnonymous=1 settings by attempting to reverse each ID value to a username. Example 9-20 shows the usage and command-line options for *smbdumpusers*.

Example 9-20. smbdumpusers usage and command-line options

```
D:\smb-at> smbdumpusers
```

9.7 Unix Samba Vulnerabilities

The Samba open source suite (http://www.samba.org) allows Linux and other Unix-like platforms to operate more easily within Windows NT domains and provides seamless file and print services to SMB and CIFS clients. Over the last six years, a number of remote vulnerabilities have been found in Samba services that allow attackers to compromise mostly Linux systems.

At the time of writing, the ISS X-Force vulnerability database (http://xforce.iss.net) lists a number of serious remotely exploitable issues in Samba (not including denial of service or locally exploitable post-authentication issues), as shown in Table 9-6.

Table 9-6. Remotely exploitable Samba vulnerabilities

XF ID	Date	Notes
12749	27/07/2003	Samba 2.2.7a and prior reply_nttrans( ) overflow
11726	07/04/2003	Samba 2.2.5 through 2.2.8 and Samba-TNG 0.3.1 and prior call_trans2open( ) remote overflow
11550	14/03/2003	Samba 2.0 through 2.2.7a remote packet fragment overflow
10683	20/11/2002	Samba 2.2.2 through 2.2.6 password change request overflow
10010	28/08/2002	Samba 2.2.4 and prior enum_csc_policy( ) overflow
6731	24/06/2001	Samba 2.0.8 and prior remote file creation vulnerability
3225	21/06/1999	Samba 2.0.5 and prior messaging service remote overflow
337	01/09/1997	Samba 1.9.17 and prior remote password overflow

9.8 Windows Networking Services Countermeasures

- Filter public or nontrusted network access to high-risk services, especially the MSRPC service that are accessible through TCP and UDP port 135, and the NetBIOS session and CIFS services (TCP ports 139 and 445), which can be attacked and used to compromise Windows environments.
- Ensure local administrator accounts passwords are set because these are often set to NULL on workstations when domain authentication is used. If possible, disable the local computer Administrator accounts across your network.
- Enforce a decent user account lockout policy to minimize the impact of brute-force password-grinding attacks.

Here are Microsoft RPC service-specific countermeasures:

- If RPC services are accessible from the Internet, ensure that the latest Microsoft security patches relating to RPC components are installed. At the time of writing, these are MS03-026 and MS03-039.
- Disable the Task Scheduler and Messenger services if they aren't required. The Task Scheduler can be used by attackers to remotely execute commands upon authenticating, and both services have known memory-management issues.
- Disable DCOM support if it isn't required because this will minimize the current and future threat presented by RPC service attacks (such as the Blaster worm in 2003). Microsoft KB article 825750 discusses this; you can find it at http://support.microsoft.com/default.aspx?kbid=825750.
- Be aware of threats presented by RPC over HTTP functionality within Microsoft IIS web services (when COM Internet Services is installed). Ensure that the RPC_CONNECT HTTP method isn't allowed (unless required) through any publicly accessible web services in your environment.

Here are NetBIOS Session and CIFS service-specific countermeasures:

- Enforce RestrictAnonymous=2 under Windows 2000, XP, and 2003 hosts to prevent enumeration of system information through NetBIOS. The registry key can be found under HKLM\SYSTEM\CurrentControlSet\Control\Lsa. Microsoft KB articles 246261 and 296405 should be reviewed and are accessible from http://support.microsoft.com.
- Enforce NTLMv2 if possible. Fast multithreaded brute-force tools, such as SMBCrack, take advantage of weaknesses within standard NTLM, and therefore don't work against the cryptographically stronger NTLMv2.
-

Chapter 10. Assessing Email Services

Email services can relay information across the Internet and private networks. Due to the nature of these services, channels between the Internet and corporate network space are opened, which determined attackers can abuse to compromise internal networks. This chapter defines a strategy for assessing email services, through accurate service identification, enumeration of enabled options, and testing for known issues.

10.1 Email Service Protocols

Here are the common network ports used for email delivery and collection through SMTP, POP-2, POP-3, and IMAP:

```
smtp            25/tcp
```

10.2 SMTP

Most organizations with an Internet presence use email to communicate and to do business. Simple Mail Transfer Protocol (SMTP) servers provide email transport via software packages such as Sendmail, Microsoft Exchange, Lotus Domino, and Postfix. Here I discuss the techniques used to identify and exploit SMTP services.

10.2.1 SMTP Service Fingerprinting

Accurate identification of the SMTP service enables you to make sound decisions and efficiently assess the target system. Two tools in particular perform a number of tests to ascertain the SMTP service in use:[1]

[1] URLs for tools in this book are mirrored at the O'Reilly site, http://examples.oreilly.com/networksa/tools.

smtpmap

http://freshmeat.net/projects/smtpmap

smtpscan

http://www.greyhats.org/outils/smtpscan/smtpscan-0.2.tar.gz

Both tools are launched from Unix-like platforms. Example 10-1 shows the *smtpmap* command in use, identifying the mail service on mail.trustmatta.com as Lotus Domino 5.0.9a.

Example 10-1. The smtpmap tool in use

```
# smtpmap mail.trustmatta.com
```

10.3 POP-2 and POP-3

Post Office Protocol Versions 2 and 3 (POP-2 and POP-3) are end-user email services. POP-2 services are rare nowadays because most organizations use POP-3 rather than TCP port 110. Common POP-3 email services include Qualcomm QPOP (also known as *qpopper*; it runs on many Unix platforms) and the POP-3 component of Microsoft Exchange. These services are traditionally vulnerable to brute-force password grinding and process-manipulation attacks, as discussed next.

10.3.1 POP-3 Brute-Force Password-Grinding

After performing enumeration and identifying local user accounts through Sendmail and other avenues, it is trivial to perform a brute-force password-grinding attack. As I've discussed throughout the book so far, tools such as Brutus and Hydra offer parallel password grinding to the masses.

You can use most POP-3 servers to launch frequently effective brute-force password-grinding attacks, for three reasons:

- They don't pay attention to account lockout policies.
- They allow a large number of login attempts before disconnecting.
- They don't log unsuccessful login attempts.

Many specific Unix-based POP-3 brute-force tools exist and can be found in the Packet Storm archive, including: http://packetstormsecurity.org/groups/ADM/ADM-pop.chttp://packetstormsecurity.org/Crackers/Pop_crack.tar.gz http://packetstormsecurity.org/Crackers/hv-pop3crack.pl

10.3.2 POP-3 Process Manipulation Attacks

Both unauthenticated and authenticated process-manipulation attacks pose a serious threat to security. Most users who pick up email via POP-3 shouldn't be allowed to execute arbitrary commands on the POP-3 server; however, they can do so via post-authentication overflows in user commands such as LIST, RETR, or DELE.

10.3.2.1 Qualcomm QPOP process-manipulation vulnerabilities

At the time of writing the MITRE CVE list details a handful of vulnerabilities in Qualcomm QPOP (not including denial of service issues), as shown in Table 10-4. Serious post-authentication vulnerabilities are also listed in Table 10-4 because they allow users to execute arbitrary code.

Table 10-4. Remotely exploitable QPOP vulnerabilities

CVE name	Date	Notes

PREV < Day Day Up > NEXT

10.4 IMAP

Internet Message Access Protocol (IMAP) services are commonly found running on TCP port 143. The IMAP protocol is much like POP-3; a user authenticates with a plaintext network service and can then collect and manage their email.

Most accessible IMAP servers on the Internet today run the Washington University IMAP service (known as both UW IMAP and WU-IMAP), distributed from the official UW IMAP site at http://www.washington.edu/imap/. Mark Crispin (http://staff.washington.edu/mrc/) invented and maintains IMAP, which currently uses IMAP4rev1 as the standard server protocol (RFC 3501).

10.4.1 IMAP Brute Force

As with many other simple plaintext protocols (Telnet, FTP, POP-3, etc.), Brutus and Hydra do an excellent job brute-forcing valid user-account passwords from both Unix-based and Win32 GUI environments. As mentioned earlier, they can be downloaded from:
http://www.hoobie.net/brutus/brutus-download.htmlhttp://www.thc.org/releases.php

Like POP-3, IMAP services are notoriously susceptible to brute-force password-grinding attack because they don't pay attention to account lockout policies and often don't log unsuccessful authentication attempts.

10.4.2 IMAP Process Manipulation Attacks

Since 1997, a handful of remotely exploitable security vulnerabilities within IMAP2bis and IMAP4rev1 services have been publicized, which are summarized in Table 10-5.

Table 10-5. Remotely exploitable IMAP vulnerabilities

CVE name	Date	Notes
CVE-1999-0005	17/07/1998	Washington University IMAP 4 (IMAP4rev1 10.234) and prior AUTHENTICATE command overflow
CVE-1999-0042	02/03/1997	Washington University IMAP 4.1beta and prior LOGIN command overflow
CVE-2000-0233	27/03/2000	SuSE Linux IMAP server allows remote attackers to bypass IMAP authentication and gain privileges

10.5 Email Services Countermeasures

- Don't run Sendmail in high-security environments, because the software contains many bugs and is heavily bloated. Sound Unix-based alternatives include *qmail* (http://www.qmail.org) and *exim* (http://www.exim.org), neither of which is as complex or susceptible to Internet-based attack.
- To minimize the impact of a user-enumeration and password-grinding attack, ensure that all user accounts on SMTP and POP-3 mail servers have strong passwords. Ideally, SMTP servers shouldn't also run remote maintenance or email pickup services to the public Internet.
- If you do offer public POP-3 or IMAP mail services, investigate their resilience from brute-force attack, including logging provisions and whether an account lockout policy can be deployed.
- Using SSL-enhanced versions of POP-3 and IMAP services will minimize the risk of plaintext user account password details from being sniffed. Plaintext services are open to determined attack, so you need either SSL or VPN client software to protect both passwords and the email data sent from point to point.
- Ensure that inbound commercial SMTP relay and antivirus scanners (such as Clearswift MAILsweeper and InterScan VirusWall) are patched and maintained to prevent circumvention attacks from being effective.
- Cisco PIX, Check Point Firewall-1, and other firewall systems can run SMTP proxy services to scrub traffic flowing to and from SMTP mail servers. This SMTP proxy functionality is known as the "SMTP Security Server" under Check Point and the "SMTP fixup protocol" under Cisco. While these proxy components aren't bulletproof, they do provide valuable protection.

Chapter 11. Assessing IP VPN Services

This chapter tackles assessment of services found running on network boundaries that provide secure remote access over IP. Increasingly, VPN services provide access for both home users and branch offices, using IPsec, proprietary Check Point FWZ, or Microsoft PPTP. These services are under threat primarily from offline preshared key-grinding and information-leak attack, which are described in the following sections.

11.1 IPsec VPNs

VPN technologies and their underlying protocols and key exchange mechanisms fill entire books already. One excellent book I used to research and present IPsec key exchange and authentication protocols is IPSec: Securing VPNs, by Carlton R. Davis (McGraw-Hill). If you require detailed low-level information about IPsec and its various modes and protocols, you should definitely read a book dedicated to the subject. Here I tackle the key protocols and mechanisms at a high level, and discuss known remotely exploitable weaknesses and attacks.

Standard Internet (IP) packets are inherently insecure. IPsec was developed to provide security options and enhancements to IP and to negate the following security weaknesses:

- IP spoofing and packet-source forgery issues
- Modification of data within IP packets
- Replay attacks
- Sniffing attacks

IPsec VPNs use the Internet Security Association and Key Management Protocol (ISAKMP) service to provide authentication and key exchange when establishing and maintaining an IPsec connection. After authenticating, a Security Association (SA) is established between the client and VPN gateway. The SA defines the IPsec protocol to be used, as well as cryptographic algorithms, cryptographic keys, and their lifetime.

11.1.1 ISAKMP and IKE

ISAKMP is accessible through UDP port 500, and provides Internet Key Exchange (IKE) support for IPsec VPN tunnels. IKE is used as the authentication mechanism when establishing an IPsec connection; it supports three authentication methods: preshared keys, public key encryption, or digital signatures.

IKE can be run in two primary modes: main and aggressive. Each accomplishes a phase one exchange, generating authenticated keying material for use during phase two. In late 1999, Check Point developed hybrid mode, which supports a number of extra authentication methods (including LDAP, RADIUS, and RSA SecurID). At the time of writing, hybrid mode IKE is an IETF draft, available by searching the IETF site (http://search.ietf.org) for "hybrid mode authentication." The following sections detail a breakdown of main and aggressive mode IKE.

11.1.1.1 Main mode IKE

Main mode is a phase-one key-exchange mechanism that protects the identity of the client and authentication data by using a Diffie-Hellman exchange to generate a mutual secret key. Figure 11-1 shows the main mode IKE messages sent between the initiator and responder.

Figure 11-1. Main mode IKE messages in transit

11.2 Attacking IPsec VPNs

To fully assess the security of an IPsec VPN, as with any target network or system, you need to perform enumeration, initial testing, investigation, and exploitation. Here I discuss how to enumerate, probe, and investigate vulnerable IPsec VPN components efficiently. If you have access to the wire, there are a number of complex man in the middle (MITM) and sniffing attacks that can be launched to compromise IPsec VPN tunnels; however, these attacks lie outside of the scope of this book.

11.2.1 IPsec Enumeration

ipsecscan is a Win32 command-line utility that can identify IPsec enabled devices and hosts; it's available at http://ntsecurity.nu/toolbox/ipsecscan/.[1]

[1] URLs for tools in this book are mirrored at the O'Reilly site, http://examples.oreilly.com/networksa/tools.

Example 11-1 shows *ipsecscan* in action, scanning from 10.0.0.1 to 10.0.0.10 for IPsec support.

Example 11-1. ipsecscan in use to identify IPsec enabled devices

```
D:\> ipsecscan 10.0.0.1 10.0.0.10
```

11.3 Check Point VPN Security Issues

Organizations using Check Point Firewall-1 or NG to provide remote user access (through SecuRemote or SecureClient software) are susceptible to active attacks through both IPsec (ISAKMP running on UDP port 500) and proprietary FWZ (RDP running on TCP port 259) avenues that enumerate valid usernames and collect interface and network topology information.

11.3.1 Check Point IKE Username Enumeration

From a remote Internet-based perspective, attackers can perform username enumeration attacks against Check Point Firewall-1 4.1 and NG appliances that support aggressive mode IKE for authentication. Roy Hills of NTA Monitor (http://www.nta-monitor.com) demonstrated this issue in a post to the BugTraq mailing list during September 2002, mirrored at:

http://www.securityfocus.com/archive/1/290202/2002-08-29/2002-09-04/0
http://lists.insecure.org/lists/bugtraq/2002/Sep/0107.html

Roy wrote a utility called fw1-ike-userguess that enumerates valid Check Point SecuRemote users through UDP port 500. The tool isn't publicly available but is demonstrated in Example 11-4.

Example 11-4. Using fw1-ike-userguess to enumerate valid VPN usernames

```
# fw1-ike-userguess --file=testusers.txt --sport=0 172.16.2.2
```

11.4 Microsoft PPTP

Microsoft's Point to Point Tunneling Protocol (PPTP) uses TCP port 1723 for communication. Due to PPTP model complexity and reliance on MS-CHAP for authentication, PPTPv1 and PPTPv2 are vulnerable to several offline cryptographic attacks.

No active information-leak or user-enumeration vulnerabilities have been identified in PPTP to date, and so the service is adequately secure from determined remote attack (if the external attack has no access to the PPTP traffic).

For details of the multiple cryptographic weaknesses within PPTP, see Bruce Schneier's page that's dedicated to the protocol: http://www.schneier.com/pptp.html. A number of publicly available network sniffers can compromise PPTP MS-CHAP challenge/response hashes from the wire, including:
http://packetstormsecurity.org/sniffers/anger-1.33.tgzhttp://packetstormsecurity.org/sniffers/dsniff/dsniff-2.3.tar.gz
http://packetstormsecurity.org/sniffers/pptp-sniff.tar.gz

11.5 VPN Services Countermeasures

- Ensure that firewall or VPN gateway appliances have the latest security hot fixes and service packs installed to minimize the risk of a known publicized attack from being successful.

Here are IPsec-specific countermeasures:

- Preshared keys used with both main and aggressive mode IKE key exchange mechanisms are open to sniffing and offline brute-force grinding attacks to compromise the shared secret. You should use digital certificates or two-factor authentication mechanisms to negate these risks.
- Pre-shared keys and aggressive mode IKE support is a recipe for disaster. If you must support aggressive mode IKE, use digital certificates for authentication.
- Aggressively firewall and filter traffic flowing through VPN tunnels so that, in the event of a compromise, network access is limited. This point is especially important when providing mobile users network access, as opposed to branch offices.
- Where possible, limit inbound IPsec security associations to specific IP addresses. This ensures that even if an attacker compromises a preshared key, she can't easily access the VPN.

Check Point Firewall-1- and NG-specific countermeasures:

- Filter access to TCP ports 256 and 264 if they aren't required for remote access through SecuRemote, SecureClient, or similar VPN client software. These ports can be abused to collect user and network topology information.
- Check Point Firewall-1 and NG are open to active attack to enumerate valid usernames through aggressive mode IKE. If possible, disable aggressive mode support.
- If you use FWZ VPN tunnels, ensure that the latest Check Point service pack and any relevant security hot fixes are installed to negate circumvention techniques. Due to the fact that RDP runs over UDP, this service is susceptible to various types of spoofing and encapsulation attack.

Chapter 12. Assessing Unix RPC Services

Vulnerabilities in Unix RPC services have led to many large organizations falling victim to hackers over the last 10 years. One recent incident in April 1999 resulted in the web sites of Playboy, Sprint, O'Reilly Media, Sony Music, Sun Microsystems, and others being mass-defaced by H4G1S and the Yorkshire Posse (HTML mirrored at http://www.2600.com/hackedphiles/current/oreilly/hacked/). In this chapter, I cover remote RPC service vulnerabilities in Solaris, IRIX, and Linux, exploring how these services are exploited in the wild and how you can protect them.

12.1 Enumerating Unix RPC Services

A number of interesting Unix daemons (including NIS+, NFS, and CDE components) run as Remote Procedure Call (RPC) services using dynamically assigned high ports. To keep track of registered endpoints and present clients with accurate details of listening RPC services, a *portmapper* service listens on TCP and UDP port 111.

The RPC portmapper (also known as *rpcbind* within Solaris) can be queried using the *rpcinfo* command found on most Unix-based platforms, as shown in Example 12-1.

Example 12-1. Using rpcinfo to list accessible RPC service endpoints

```
# rpcinfo -p 192.168.0.50
```

PREV < Day Day Up > NEXT

12.2 RPC Service Vulnerabilities

Due to the number of different RPC services, associated prognum values, CVE references, and vulnerable platforms, it is difficult to simply group bugs and talk about them individually (as I do elsewhere in this book). I have put together the following matrix of popular services and vulnerable platforms, shown in Table 12-1. A small number of obscure IRIX services (*rpc.xfsmd*, *rpc.espd*, etc.) aren't listed; they can be investigated through MITRE CVE and other sources.

Table 12-1. RPC services, affected platforms, and CVE references

Program number	**Service**	**Platforms affected**				**CVE references**
		Solaris	**Linux**	**IRIX**	**Other**	
100000	*portmapper*	Yes	No	No	No	CVE-1999-0190
100004	*ypserv*	No	Yes	No	No	CVE-2000-1042 CVE-2000-1043
100005	*mountd*	No	Yes	No	No	CVE-1999-0002 CVE-2003-0252
100007	*ypbind*	Yes	Yes	No	No	CVE-2000-1041 CVE-2001-1328
100008	*rwalld*	Yes	No	No	No	CVE-2002-0573

12.3 Unix RPC Services Countermeasures

- Don't run *rexd*, *rusersd*, or *rwalld* RPC services, because they are of minimal use and provide attackers with both useful information and direct access to your hosts.
- In high-security environments, don't offer any RPC services to the public Internet. Due to the complexity of these services, it is highly likely that zero-day exploit scripts will be available to attackers before patch information is released.
- To minimize the risk of internal or trusted attacks against necessary RPC services (such as NFS components, including *statd*, *lockd*, and *mountd*), install the latest vendor security patches.
- Aggressively filter egress traffic, where possible, to ensure that even if an attack against an RPC service is successful, a connect-back shell can't be spawned to the attacker.

Chapter 13. Application-Level Risks

In this chapter, I focus on application-level vulnerabilities and mitigation strategies. The effectiveness of firewalls and network segmentation mechanisms is severely impacted if vulnerabilities exist within accessible network services. In recent years, major security flaws in Unix and Windows systems have been exposed, resulting in large numbers of Internet-based hosts being compromised by hackers and worms alike.

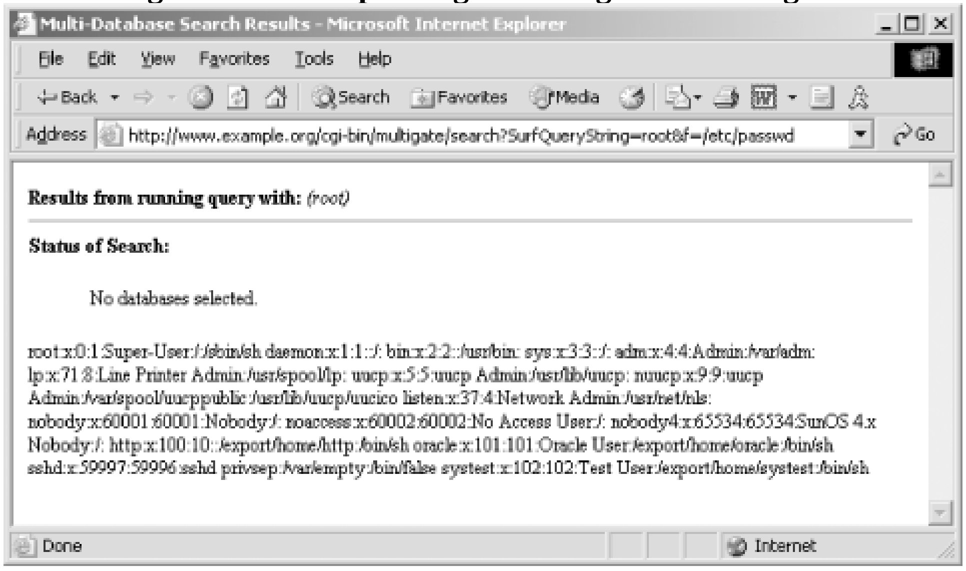
13.1 The Fundamental Hacking Concept

Hacking is the art of manipulating a process in such a way that it performs an action that is useful to you.

A simple example is to look at a search engine; the program takes a query, cross references it with a database, and provides a list of results. Processing occurs on the web server itself, and by understanding the way search engines are developed and their pitfalls (such as accepting both the query string and database filename values), a hacker can attempt to manipulate the search engine to process and return sensitive files.

Many years ago, the main U.S. Pentagon, Air Force, and Navy web servers (http://www.defenselink.mil, http://www.af.mil, and http://www.navy.mil) were vulnerable to this very type of search engine attack. They used a common search engine called *multigate*, which accepted two abusable arguments: SurfQueryString and f. The Unix password file could be accessed by issuing a crafted URL, as shown in Figure 13-1.

Figure 13-1. Manipulating the multigate search engine


Multi-Database Search Results - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go
Done
Internet

```
http://www.example.org/cgi-bin/multigate/search?SurfQueryString=root&f=/etc/passwd
```

Results from running query with: *(root)*

Status of Search:

No databases selected.

```
root:x:0:1:Super-User:/:/sbin/sh daemon:x:1:1::/: bin:x:2:2::/usr/bin: sys:x:3:3::/: adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp: uucp:x:5:5:uucp Admin:/usr/lib/uucp: nuucp:x:9:9:uucp
Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/: noaccess:x:60002:60002:No Access User:/: nobody4:x:65534:65534:SunOS 4.x
Nobody:/: http:x:100:10::/export/home/http:/bin/sh oracle:x:101:101:Oracle User:/export/home/oracle:/bin/sh
sshd:x:59997:59996:sshd privsep:/var/empty:/bin/false systest:x:102:102:Test User:/export/home/systest:/bin/sh
```

High-profile military web sites are properly protected at network level by firewalls and other security appliances. However, by the very nature of the massive amount of information stored, a search engine was implemented, which in turn introduced vulnerabilities at application level.

Nowadays, a lot of vulnerabilities are more complex than simple logic flaws. Stack, heap, and static overflows, along with format string bugs, allow remote attackers to manipulate nested functions and often execute arbitrary code on accessible hosts.

13.2 The Reasons Why Software Is Vulnerable

In a nutshell, software is vulnerable due to complexity and inevitable human error. Many vendors (e.g., Microsoft, Sun, Oracle, and others) that developed and built their software in the 90's didn't write code that was secure from heap overflows or format string bugs, because these issues were not widely known at the time.

Software vendors are now in a situation where, even though it would be the just thing to do, it is simply too expensive to secure their operating systems and server software packages from memory manipulation attacks. Code review and full black box testing of complex operating system and server software would take years to undertake, and severely impact future development and marketing plans, along with revenue.

In order for adequately secure programs to be developed, the interaction of that program with the environment in which it is run should be controlled at all levels—no data passed to the program should be trusted or assumed to be correct. Input validation is a term used within application development to ensure that data passed to a function is properly sanitized before it is stored in memory. Proper validation of all external data passed to key network services would go a long way toward improving the security and resilience of IP networks and computer systems.

13.3 Network Service Vulnerabilities and Attacks

In this section, I concentrate on Internet-based network service vulnerabilities, particularly how software running at both the kernel and system daemon levels processes data. These vulnerabilities can be categorized into two high-level groups: memory manipulation weaknesses and simple logic flaws.

Memory manipulation attacks are detailed here to help you understand the classification of bugs and the respective approaches that can be taken to mitigate risks. Simple logic flaws are identified and tackled throughout the book already (see Section 6.6) and are a much simpler threat to deal with.

13.3.1 Memory Manipulation Attacks

Memory manipulation attacks involve sending malformed data to the target network service in a manner so that the logical program flow is affected (the idea is to execute arbitrary code on the host, although crashes sometimes occur, resulting in denial of service).

Here are the three high-level categories of remotely exploitable memory manipulation attacks:

- Classic buffer overflows (stack, heap, and static overflows)
- Integer overflows (technically an overflow delivery mechanism)
- Format string bugs

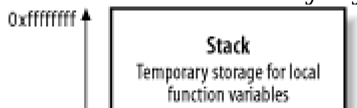
I discuss these three attack groups and describe individual attacks within each group (such as stack saved instruction and frame pointer overwrites). There are a small number of exotic bug types (e.g., index array manipulation and static overflows) that unfortunately lie outside the scope of this book, but which are covered in niche application security publications and online presentations.

Through understanding how exploits work, you can effectively implement changes to your critical systems to protect against future vulnerabilities. To appreciate these low-level issues, you must first have an understanding of runtime memory organization and logical program flow.

13.3.2 Runtime Memory Organization

Memory manipulation attacks involve overwriting values within memory (such as instruction pointers) to change the logical program flow and execute arbitrary code. Figure 13-2 shows memory layout when a program is run, along with descriptions of the four key areas: text, data and BSS, the stack, and the heap.

Figure 13-2. Runtime memory layout



PREV < Day Day Up > NEXT

13.4 Classic Buffer-Overflow Vulnerabilities

By providing malformed user input that isn't correctly checked, you can often overwrite data outside the assigned buffer in which the data is supposed to exist. Commonly you do this by providing too much data to a process, which overwrites important values in memory and causes a program crash.

Depending on exactly which area of memory (stack, heap, or static segments) your input ends up in, and overflows out of, you can use numerous techniques to influence the logical program flow, and often run arbitrary code.

What follows are details of the three classic classes of buffer overflows, along with details of individual overflow types. Some classes of vulnerability are easier to exploit remotely than others, which limits the options an attacker has in some cases.

13.4.1 Stack Overflows

Since 1988, stack overflows have led to the most serious compromises of security. Nowadays, many operating systems (including Microsoft Windows 2003 Server, OpenBSD, and various Linux distributions) have implemented nonexecutable stack protection mechanisms, and so the effectiveness of traditional stack overflow techniques is lessened.

By overflowing data on the stack, you can perform two different attacks to influence the logical program flow and execute arbitrary code:

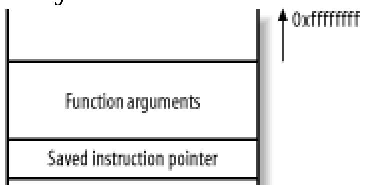
- A stack smash, overwriting the saved instruction pointer
- A stack off-by-one, overwriting the saved frame pointer

These two techniques can change logical program flow, depending on the program at hand. If the program doesn't check the length of the data provided, and simply places it into a fixed sized buffer, you can perform a stack smash. A stack off-by-one bug occurs when a programmer makes a small calculation mistake relating to lengths of strings within a program.

13.4.2 Stack Smash (Saved Instruction Pointer Overwrite)

As stated earlier, the stack is a region of memory used for temporary storage. In C, function arguments and local variables are stored on the stack. Figure 13-4 shows the layout of the stack when a function within a program is entered.

Figure 13-4. Stack layout when a function is entered



PREV < Day Day Up > NEXT

13.5 Heap Overflows

Not all buffers are allocated on the stack. Often an application doesn't know how big to make certain buffers until it is running. The heap is used by applications to dynamically allocate buffers of varying sizes. These buffers are susceptible to overflows if user-supplied data isn't checked, leading to a compromise through an attacker overwriting other values on the heap.

Where the details of stack overflow exploitation rely on the specifics of hardware architecture, heap overflows are reliant on the way certain operating systems and libraries manage heap memory. Here I restrict the discussion of heap overflows to a specific environment: a Linux system running on an Intel x86 platform, using the default GNU libc heap implementation (based on Doug Lea's *dlmalloc*). While this situation is specific, the techniques I discuss apply to other systems, including Solaris and Windows.

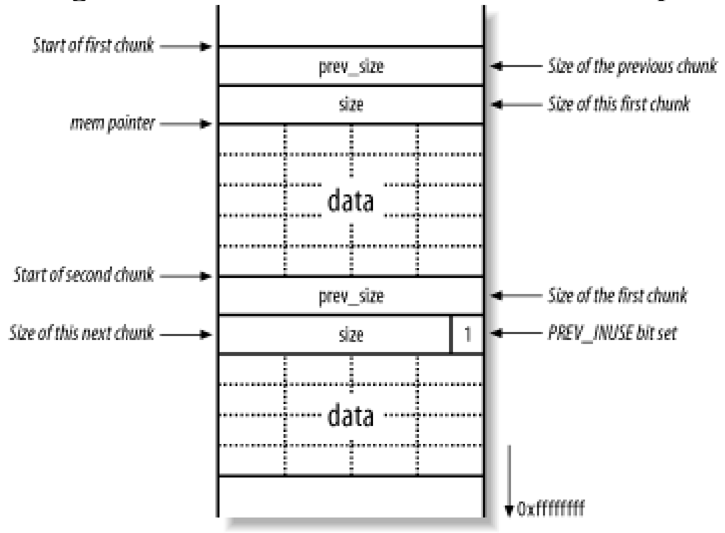
Heap overflows can result in compromises of both sensitive data (overwriting filenames and other variables on the heap) and logical program flow (through heap control structure and function pointer modification). I discuss the threat of compromising logical program flow here, along with a conceptual explanation and diagrams.

13.5.1 Overflowing the Heap to Compromise Program Flow

The heap implementation divides the heap into manageable chunks and tracks which heaps are free and in use. Each chunk contains a header structure and free space (the buffer in which data is placed).

The header structure contains information about the size of the chunk and the size of the preceding chunk (if the preceding chunk is allocated). Figure 13-12 shows the layout of two adjacent allocated chunks.

Figure 13-12. Two allocated chunks on the heap



In Figure 13-12, mem is the pointer returned by the malloc( ) call to allocate the first chunk. The size and prev_size 4-byte values are used by the heap implementation to keep track of the heap and its layout. Please note that here I have drawn these heap diagrams upside down (when compared with the previous stack diagrams), therefore 0xffffffff is downward in these figures.

The *size* element does more than just hold the size of the current chunk, it also specifies whether the previous chunk is free or not. If a chunk is allocated, the size element of the next chunk has its least significant bit set, otherwise this

13.6 Integer Overflows

The term integer overflow is often misleading. An integer overflow is simply a delivery mechanism for a stack, heap, or static overflow to occur (depending on where the integer ends up in memory).

Arithmetic calculations are often performed on integers to calculate many things, such as the amount of data to be received from the network, the size of a buffer, etc. Some calculations are vital to the logic of a program, and if they result in erroneous values, the program's logic may be severely corrupted or hijacked completely.

Calculations can sometimes be made to give incorrect results because the result is simply too big to be stored in the variable to which it is assigned. When this happens, the lowest part of the result is stored, and the rest (which doesn't fit in the variable) is simply discarded, as demonstrated here:

```
int a = 0xffffffff;
```

13.7 Format String Bugs

Buffer overflows aren't the only type of bug that can control a process. Another fairly common programming error is the situation in which a user can control the format parameter to a function, such as printf( ) or syslog( ). These functions take a format string as a parameter that describes how the other parameters should be interpreted.

For example, the string %d specifies that a parameter should be displayed as a signed decimal integer, while %s specifies that a parameter should be displayed as an ASCII string. Format strings give you a lot of control over how data is to be interpreted, and this control can sometimes be abused to read and write memory in arbitrary locations.

13.7.1 Reading Adjacent Items on the Stack

Example 13-11 shows a vulnerable C program, much like the *printme* program in Example 13-1.

Example 13-11. A simple C program containing a format string bug

```
int main(int argc, char *argv[])
```

PREV < Day Day Up > NEXT

13.8 Memory Manipulation Attacks Recap

Variables can be stored in the following areas of memory:

- Stack segment (local buffers with known sizes)
- Heap segment (dynamically allocated buffers with varying sizes)
- BSS and data segments (static buffers used for global variables)

If bounds checking of data that is copied and stored in memory isn't performed, logical program flow can be compromised. The following common threats are posed:

Stack smash bugs

The saved instruction pointer for the stack frame is overwritten, which results in a compromise when the function epilogue occurs, and the instruction pointer is popped. This executes arbitrary code from a location of your choice.

Stack off-by-one bugs

The least significant byte of the saved frame pointer for the stack frame is overwritten, which results in the parent stack frame existing at a slightly lower memory address than before (into memory that you control). You can overwrite the saved instruction pointer of the new stack frame and wait for the function to exit (requiring two returns in succession) or overwrite a function pointer or other variable found within the new stack frame. This attack is only effective against little endian processors, such as Intel x86 and DEC Alpha.

Heap overflows

If you supply too much data to a buffer on the heap, you can overwrite both heap control structures for other memory chunks or overwrite function pointers or other data. Some heap implementations (such as BSD PHK, used by FreeBSD, NetBSD, and OpenBSD) don't mix heap data and control structures, so are only susceptible to function pointers and adjacent heap data being overwritten.

Static overflows

Not discussed here, but static overflows are very similar to heap and off-by-one attacks. Logical program flow is usually compromised using a static overflow to overwrite a function pointer, generic pointer, or authentication flag. Static overflows are rare, due to the unusual global nature of the variable being overflowed.

Integer overflows (delivery mechanism for stack, heap, and static overflows)

13.9 Mitigating Process Manipulation Risks

There are a number of techniques that can be mitigate underlying security issues, so that even if your applications or network services are theoretically vulnerable to attack, they can't be practically exploited.

Here are the five main approaches:

- Nonexecutable stack and heap implementation
- Use of canary values in memory
- Running unusual server architecture
- Compiling applications from source
- Active system call monitoring

As with any bolt-on security mechanism, there are inherent positive and negative aspects. Here I discuss these approaches and their shortfalls in some environments.

13.9.1 Nonexecutable Stack and Heap Implementation

An increasing number of operating systems support nonexecutable stack and heap protection (including OpenBSD, Solaris, and a small number of Linux distributions). This approach prevents the instruction pointer from being overwritten to point at code on the stack or heap (where most exploits place their shellcode in user-supplied buffers).

To defeat this kind of protection, *return-into-libc*, or a similar attack executes in-built system library calls that can be used to compromise the system. These attacks require accurate details of loaded libraries and their locations, which can only practically be gained through having a degree of local system access in the first place.

From a network service protection perspective, implementing nonexecutable stack and heap elements can certainly prevent remote exploitation of most memory manipulation bugs.

13.9.2 Use of Canary Values in Memory

Windows 2003 Server, OpenBSD, and a number of other operating systems place canary values on the stack (and sometimes heap) to protect values that are critical to logical program flow (such as the saved frame and instruction pointers on the stack).

A *canary value* is a hashed word that is known by the system and checked during execution (e.g., before a function returns). If the canary value is modified, the process is killed, preventing practical exploitation.

13.10 Recommended Secure Development Reading

Prevention is the best form of protection from application-level threats such as overflows and logic flaws. The following four books discuss secure programming techniques and approaches (primarily with C programming examples across Unix and Windows platforms):

- Writing Secure Code, by Michael Howard and David LeBlanc (Microsoft Press)
- Secure Coding: Principles and Practices, by Mark Graff and Kenneth van Wyk (O'Reilly Media, Inc.)
- Building Secure Software, by Gary McGraw and John Viega (Addison Wesley)
- Secure Programming Cookbook for C and C++, by Matt Messier and John Viega (O'Reilly)

PREV < Day Day Up > NEXT

Chapter 14. Example Assessment Methodology

In this final chapter, I walk through a remote security assessment of a small network protected by a firewall. By reading through this process from start to finish, you will have a good understanding of the overall process. The exercise will identify, attack, and penetrate systems in a class-c network space, from my launch system on a remote network.

PREV < Day Day Up > NEXT

14.1 Network Scanning

Increasingly, network scanning is becoming a cyclic process, primarily due to the finite amount of time you have to perform a network security assessment exercise and the fact that most firewalls repel fast SYN port scans.

The best practice approach to network scanning is to undertake the following:

- Initial network scanning to identify poorly protected hosts and common services
- Full scanning to identify all remotely accessible TCP and UDP services
- Low-level network testing to gain insight into firewall and host configuration

In this section I perform these tests against the target 192.168.10.0/24 network. By coming up against the hurdles placed in my way by firewalls and defensive mechanisms, you will see how my approach is applied to get accurate results.

14.1.1 Initial Network Scanning

In Example 14-1, I use *nmap* with the -sP option to perform an initial sweep of the target network and identify any obvious accessible hosts that are poorly protected. If I don't specify the -PI option, *nmap* also sends TCP probes to port 80 of each host in the target range.

Example 14-1. Using nmap to perform an ICMP ping sweep

```
# nmap -sP -PI 192.168.10.0/24
```

14.2 Accessible Network Service Identification

After identifying accessible TCP and UDP network services with *nmap* (which also performs IP and service fingerprinting), you must perform analysis and further identification of complex network services. The five services I want at this point in the example are:

- The Telnet service running on the Cisco router (192.168.10.1)
- The SSH and SMTP services running on the Sun mail server (192.168.10.10)
- The HTTP and HTTPS services running on the Windows 2000 web server (192.168.10.25)

The SNMP and NTP services that are accessible via UDP on the Cisco router don't require further investigation, as they use a connectionless protocol.

14.2.1 Initial Telnet Service Assessment

nmap has already identified the router at 192.168.10.1 as running Cisco IOS 12.2.8. Example 14-9 shows how to obtain insight into the authentication mechanism in use and brute-force options, by connecting to the acessible Telnet service.

Example 14-9. Connecting to the Cisco IOS Telnet service

```
# telnet 192.168.10.1
```

14.3 Investigation of Known Vulnerabilities

After performing full TCP and UDP port scanning, along with initial investigation of accessible network services to qualify *nmap* results and obtain further useful information, you usually know enough to properly investigate known vulnerabilities.

Sites such as MITRE CVE, SecurityFocus, ISS X-Force, and Packet Storm provide bug details, along with publicly accessible exploit scripts. To fully qualify vulnerabilities by hand, you often need to use such tools. What follows is a breakdown of the results I obtained from these sites in relation to the accessible network services I identified in this case study.

14.3.1 Cisco IOS Accessible Service Vulnerabilities

Telnet, NTP, and SNMP services are accessible on the Cisco IOS 12.2.8 router at 192.168.10.1. Through checking MITRE CVE, SecurityFocus, and ISS X-Force, no remotely exploitable issues were identified that affect this version of IOS.

Therefore, the two particular threats to this Cisco IOS router are from:

- Telnet service password grinding
- SNMP service community string grinding

14.3.2 Solaris 8 Accessible Service Vulnerabilities

OpenSSH 3.1p1 and Sendmail 8.11.6 were found running on the Solaris 8 mail server at 192.168.10.10. Table 14-3 shows the remotely exploitable issues identified through checking MITRE CVE, SecurityFocus, and ISS X-Force databases for issues relating to OpenSSH 3.1p1.

Table 14-3. Relevant OpenSSH vulnerabilities identified

CVE	BID	XFID	Notes
CVE-2002-0639	5093	9169	OpenSSH 3.3 and prior contains vulnerabilities in challenge-response handling code.
CVE-2003-0190	7467	11902	OpenSSH 3.6.1p1 and earlier, with PAM support enabled, allows remote attackers to determine

14.4 Network Service Testing

After investigating the potential low-level weaknesses within each accessible network service, I launch exploit scripts and attack techniques against each service to qualify and test the vulnerabilities.

14.4.1 Cisco IOS Router (192.168.10.1)

The router is susceptible to brute-force attack through its Telnet and SNMP services. A full-blown brute-force attack (which often takes days to complete) should be launched if initial brute-force attacks, using obvious common passwords, fail. Example 14-17 shows *hydra* in use to brute-force the Cisco IOS Telnet service password, using a list of default Cisco passwords from *pass.txt*.

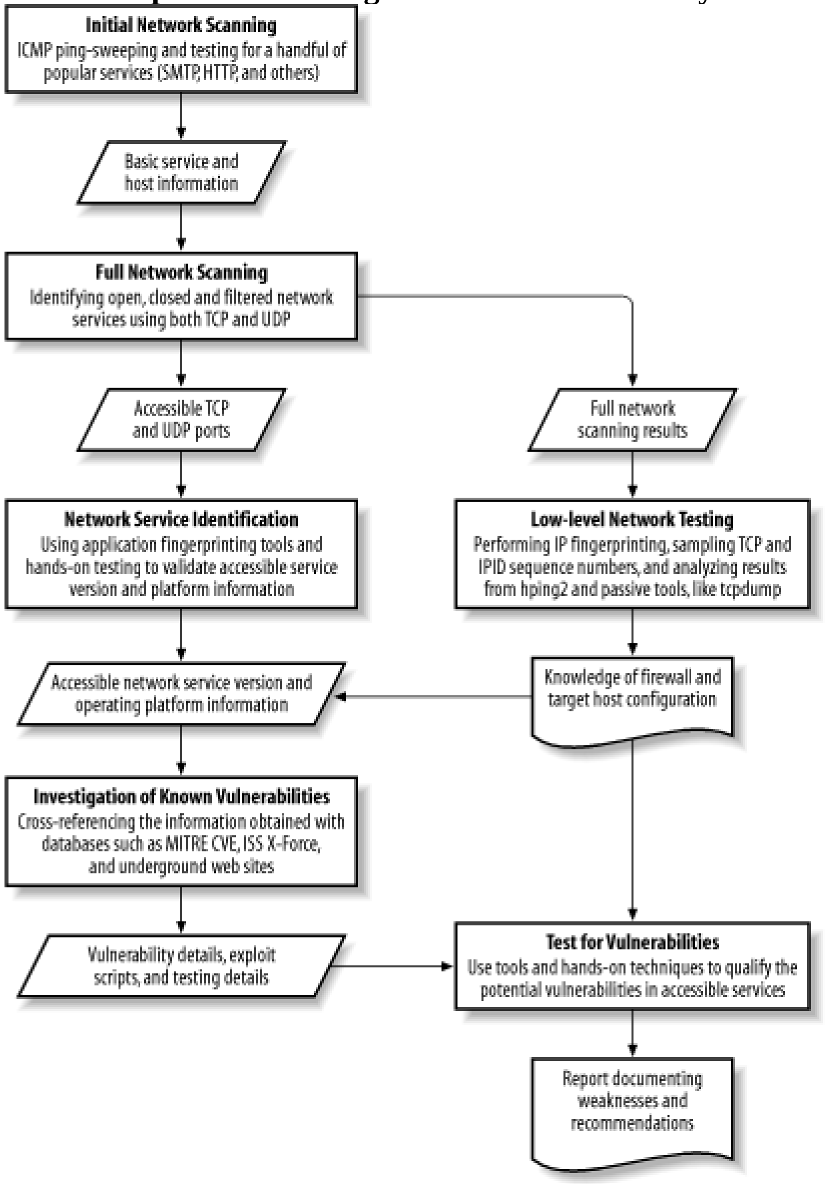
Example 14-17. Performing initial Telnet brute force using Hydra

```
# cat pass.txt
```

14.5 Methodology Flow Diagram

The overall methodology is relatively straightforward; it covers initial and full network scanning, low-level network testing (depending on the type of network and filtering mechanisms), accessible service identification, investigation of vulnerabilities, and qualification of vulnerabilities. Figure 14-1 shows this flow diagram at a high-level and the data passed between each process.

Figure 14-1. A process flow diagram for network security assessment


Initial Network Scanning
ICMP ping-sweeping and testing for a handful of popular services (SMTP, HTTP, and others)
Basic service and host information
Full Network Scanning
Identifying open, closed and filtered network services using both TCP and UDP
Accessible TCP and UDP ports
Full network scanning results
Network Service Identification
Using application fingerprinting tools and hands-on testing to validate accessible service version and platform information
Low-level Network Testing
Performing IP fingerprinting, sampling TCP and IPID sequence numbers, and analyzing results from hping2 and passive tools, like tcpdump
Accessible network service version and operating platform information
Knowledge of firewall and target host configuration
Investigation of Known Vulnerabilities
Cross-referencing the information obtained with databases such as MITRE CVE, ISS X-Force, and underground web sites
Vulnerability details, exploit scripts, and testing details
Test for Vulnerabilities
Use tools and hands-on techniques to qualify the potential vulnerabilities in accessible services
Report documenting weaknesses and recommendations

If you are new to security assessment, you will soon realize that it is highly time-consuming to search and cross reference various web sites and information sources for accurate vulnerability information. The "Investigation of Known Vulnerabilities" component shown in Figure 14-1 will prove hard to carry out the first few times you try, but after a while, you will be able to read through the port scan results, and get a good idea of the vulnerabilities to test for, and the exploits to use.

14.6 Recommendations

Upon performing the assessment exercise, and qualifying the vulnerabilities at hand, a plan should be put forward to improve security. Recommendations fall into two categories: quick wins and long-term recommendations.

14.6.1 Quick Win Recommendations

The quick win recommendations for the immediate improvement of security in this case are as follows, broken down by target host.

14.6.1.1 Cisco IOS router

A router Access Control List (ACL) should be implemented to prevent public access, particularly to the Telnet and SNMP services. NTP doesn't pose a security issue within Cisco IOS at the time of writing, although it would be diligent to filter access to this service also.

14.6.1.2 Solaris mail server

Public access to the OpenSSH service should be filtered, allowing only trusted hosts to connect. OpenSSH should also be upgraded to the latest stable release (3.7.1p2 at the time of writing, available from http://www.openssh.com), to negate the risks posed by the four remote memory manipulation attacks, and the one user enumeration bug.

The Sendmail service should be upgraded to the latest stable release (8.12.10 at the time of writing, available from http://www.sendmail.org) to negate the risks posed by the recent prescan( ) vulnerabilities that can permit a remote compromise. A catch-all email account should also be implemented so that RCPT TO: local user enumeration attacks are no longer effective.

14.6.1.3 Windows 2000 web server

Basic hardening of the IIS 5.0 web service ensures that bugs in components and subsystems that are rarely used aren't remotely exploitable. In particular, the following should be undertaken in this case:

- Install the latest Windows 2000 service pack and IIS 5.0 security hot fixes.
- Disable unnecessary ISAPI extensions, including *.ida*, *.idq*, and *.printer*.
- Install Microsoft URLScan to filter requests and block dangerous HTTP methods.

14.6.1.3.1 Disable unnecessary ISAPI extensions

You can disable unnecessary ISAPI extensions by clicking through the following Internet Services Manager (ISM) menus and options:

1.

14.7 Closing Comments

I was at a book signing in the United Kingdom a few weeks ago, where the author Neal Stephenson was asked about his writing processes, and how much time and effort goes into outlining the plot and characters before writing the novel. Neal replied that any skilled professional (whether a blacksmith, race car driver, or computer programmer) can do their job with a minimal amount of planning because the human mind can be trained in such a way that relevant information can be stored and manipulated in a very fluid sense.

By reading through this book, you should be aware of the major Internet-based security issues that have been publicized over the last 10 years. The trick now is for you to keep up-to-date with the latest threats, and learn to manage and use this data in an efficient and fluid way. By performing assessment exercises and building up your own knowledge base, you will become proficient at protecting IP networks from remote attack.

Appendix A. TCP, UDP Ports, and ICMP Message Types

I list useful TCP, UDP ports, and ICMP message types in this appendix. There exist a small number of remotely exploitable network services I don't cover in the book, but list here—for example, the Solaris *dtspcd* service on port 6112 and the X font server on port 7100.

A comprehensive list of registered TCP and UDP services may be found at http://www.iana.org/assignments/port-numbers. The *nmap-services* list of ports provided with *nmap* is also a good reference, particularly for backdoors and other unregistered services.

A.1 TCP Ports

TCP ports of interest from a remote security assessment perspective are listed in Table A-1. I have included references to chapters within this book, along with other details that I deem appropriate, including MITRE CVE references to known issues.

Table A-1. TCP ports

Port	**Name**	**Notes**
1	*tcpmux*	TCP port multiplexer, indicates the host is running IRIX
11	*systat*	System status service; see Chapter 5
15	*netstat*	Network status service; see Chapter 5
21	*ftp*	File Transfer Protocol (FTP) service; see Chapter 8
22	*ssh*	Secure Shell (SSH); see Chapter 7
23	*telnet*	Telnet service; see Chapter 7
25	*smtp*	Simple Mail Transfer Protocol (SMTP); see Chapter 10
42	*wins*	Microsoft WINS name service
43	*whois*	WHOIS service; see Chapter 3
53	*domain*	Domain Name Service (DNS); see Chapter 5
79	*finger*	Finger service, used to report active

A.2 UDP Ports

UDP ports of interest from a remote security assessment perspective are listed in Table A-2. I have included references to chapters within this book, along with other details that I deem appropriate, including MITRE CVE references to known issues.

Table A-2. UDP ports

Port	**Name**	**Notes**
53	*domain*	Domain Name Service (DNS); see Chapter 5
67	*bootps*	BOOTP (commonly known as DHCP) server port
68	*bootpc*	BOOTP (commonly known as DHCP) client port
69	*tftp*	Trivial File Transfer Protocol (TFTP), a historically weak protocol used to upload configuration files to hardware devices
111	*sunrpc*	RPC portmapper (also known as rpcbind); see Chapter 12
123	*ntp*	Network Time Protocol (NTP), often on Cisco IOS devices
135	*loc-srv*	Microsoft RPC server service; see Chapter 9
137	*netbios-ns*	Microsoft NetBIOS name service; see Chapter 9
138	*netbios-dgm*	Microsoft NetBIOS datagram service; see Chapter 9

A.3 ICMP Message Types

ICMP message types of interest from a remote security assessment perspective are listed in Table A-3. Both the message types and individual codes are listed, along with details of RFCs and other standards in which these message types are discussed.

Table A-3. ICMP message types

Type	Code	Notes
0	0	Echo reply (RFC 792)
3	0	Destination network unreachable
3	1	Destination host unreachable
3	2	Destination protocol unreachable
3	3	Destination port unreachable
3	4	Fragmentation required, but don't fragment bit was set
3	5	Source route failed
3	6	Destination network unknown
3	7	Destination host unknown
3	8	Source host isolated
3	9	Communication with destination network is administratively prohibited
		Communication with destination host

Appendix B. Sources of Vulnerability Information

To maintain the security of your environment, it is vital to be aware of the latest threats posed to your network and its components. You should regularly check Internet mailing lists and hacking web sites to access the latest public information about vulnerabilities and exploit scripts. I've assembled the following lists of web sites and mailing lists that security consultants and hackers use on a daily basis.

B.1 Security Mailing Lists

BugTraq, http://www.securityfocus.com/archive/1 VulnWatch, http://www.vulnwatch.org NTBugTraq, http://www.ntbugtraq.com Full Disclosure, http://lists.netsys.com/pipermail/full-disclosure/ Pen-Test, http://www.securityfocus.com/archive/101 Web Application Security, http://www.securityfocus.com/archive/107 Honeypots, http://www.securityfocus.com/archive/119 CVE Announce, http://archives.neohapsis.com/archives/cve/ Nessus development, http://list.nessus.org/ Nmap-hackers, http://lists.insecure.org/nmap-hackers/

B.2 Vulnerability Databases and Lists

MITRE CVE, http://cve.mitre.orgISS X-Force, http://xforce.iss.netOSVDB, http://www.osvdb.orgBugTraq, http://www.securityfocus.com/bid/CERT vulnerability notes, http://www.kb.cert.org/vuls/Secunia, http://www.secunia.com

B.3 Underground Web Sites

The Hacker's Choice, http://www.thc.orgPacket Storm, http://www.packetstormsecurity.orgInsecure.org, http://www.insecure.orgZone-H, http://www.zone-h.orgPhenoelit, http://www.phenoelit.denewroot.de, http://www.newroot.dePulhas, http://p.ulh.as/Digital Offense, http://www.digitaloffense.netGOBBLES Security, http://www.immunitysec.com/GOBBLES/cqure.net, http://www.cqure.netTESO, http://www.team-teso.netADM, http://adm.freelsd.net/ADM/Netric, http://www.netric.orgHack in the box, http://www.hackinthebox.orgOutsiders, http://www.0x333.orgcnhonker, http://www.cnhonker.com.dtors, http://www.dtors.netSoft Project, http://www.s0ftpj.orgPhrack, http://www.phrack.orgLSD-PLaNET, http://www.lsd-pl.netw00w00, http://www.w00w00.orgAstalavista, http://astalavista.comBlack Sun Research Facility, http://blacksun.box.sk

B.4 Security Events and Conferences

Black Hat Briefings, http://www.blackhat.comHEX2005, http://www.hex2005.orgCCC Camp, http://www.ccc.de/camp/ToorCon, http://www.toorcon.orgCanSecWest, http://www.cansecwest.comSummerCon, http://www.summercon.orgDEF CON, http://www.defcon.org

Colophon

Our look is the result of reader comments, our own experimentation, and feedback from distribution channels. Distinctive covers complement our distinctive approach to technical topics, breathing personality and life into potentially dry subjects.

The animals on the cover of Network Security Assessment are porcupine fish (Diodon hystrix). This fish is found in oceans throughout the world, most often among or near coral reef areas. Its tube-shaped body ranges in length from 3 to 19 inches with relatively small fins. When threatened, the fish inflates itself by taking in tiny gulps of water until the stomach is full; the body expands in seconds to double or triple size, and its spines become erect. (Smaller species have spines that are permanently bristly.) The porcupine fish is covered with evenly spaced dark spots, which distinguishes it from other puffers.

The fish has a a single tooth in each jaw; fused at the midline, they form a parrotlike beak. A nocturnal hunter, it moves its body over a small area of sand and spurts tiny jets of water to uncover its prey, usually mollusks and crustaceans. The porcupine fish is popular as an aquarium specimen; it's also blown up, dried, and sold as a souvenir.

In earlier centuries, certain Pacific island warriors used the porcupine fish to fashion a battle helmet. They would catch a fish, let it inflate, and then bury it in sand for about a week. When dug up, the fish, now a hard ball, would be cut open to make a hard, head-shaped piece that looked most formidable. The porcupine fish isn't listed as endangered or vulnerable with the World Conservation Union.

Mary Anne Weeks Mayo was the production editor and proofreader, and Derek DiMatteo was the copyeditor for Network Security Assessment. Reg Aubry and Claire Cloutier provided quality control. Jamie Peppard, Mary Agner, and Marlowe Shaeffer provided production assistance. Julie Hawks wrote the index.

Emma Colby designed the cover of this book, based on a series design by Edie Freedman. The cover image is a 19th-century engraving from the Dover Pictorial Archive. Emma produced the cover layout with QuarkXPress 4.1 using Adobe's ITC Garamond font.

The online edition of this book was created by the Safari production group (John Chodacki, Becki Maisch, and Madeleine Newell) using a set of Frame-to-XML conversion and cleanup tools written and maintained by Erik Ray, Benn Salter, John Chodacki, and Jeff Liggett.

PREV < Day Day Up >

PREV < Day Day Up >

< Day Day Up >