

Systems Security Assessment: A Simple Baseline

Russ McRee

GSEC, MCSE/MCSA: Security, Security+

January 2005

Table of Contents

Abstract	Page 3
Introduction	Page 3
Servers:	
Physical Assessment	Page 3
Tools for Assessment	Page 3
Nmap	Page 4
Nessus	Page 4
OS Assessment – Microsoft Windows	Page 4
Best Practices for Securing a Windows Server	Page 6
OS Assessment – *nix	Page 8
Best Practices for Securing a *nix Server	Page 8
Server Summary	Page 9
Client Holes – Securing the Desktop Environment	Page 9
Conclusion	Page 11
Endnotes	Page 11
Addendum A – Remove Services Script & sc	Page 12
Addendum B - Known Ports	Page 13

Should questions arise regarding this document or its contents please email rmcree@yahoo.com.

Abstract

This document is intended to provide basic guidelines to systems administrators and engineers with regard to assessing vulnerabilities for two distinct environments. It is not intended to be a complete doctrine or the only solution as the effort to maintain good systems security never ends. Instead, use this paper as a path to a reasonably sound foundation on which to build.

This guideline will describe a list of vulnerabilities as they apply to servers, at the physical, OS and infrastructure in any environment.

Additionally, this document will provide a checklist for securing the desktop environment; in essence, a list of best practices as deemed essential by many wise security administrators. The author believes in a technological environment where security is everyone's concern and even the most junior administrator can and should participate. This paper is written with that goal in mind.

Introduction

It is assumed that in a current server environment a security administrator is likely to assess and maintain either a Microsoft Windows server of some kind or a Unix derivative, likely AIX, Solaris or Linux. This paper will dedicate equal consideration to both platforms and describe a detailed list of both assessment tools and the parameters with which to utilize them. Additionally, the intent is to suggest some basic steps to eliminate unnecessary vulnerabilities on all platforms.

Servers

Physical Assessment

While not as glamorous as analyzing OS or network weaknesses, physical security is no less critical. These procedures, as defined in your Security Manual (you have one, right?), regarding physical security should be considered mandatory and assessment of all environments should determine adherence to these procedures as well as create suggestions for meeting a state of compliance.

Tools for Assessment

There are an endless number of applications and platforms with which to analyze vulnerability. An equal number of vendors suggest buying their tools and additional support. For the purposes of this document only open source applications will be describe in order to avoid licensing costs and legalities. Furthermore, all said applications fall under the GPL (General Public License).

Nmap

Nmap ("Network Mapper") is a free open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. Nmap runs on most types of computers and both console and graphical versions are available. Nmap is free software, available with full source code under the terms of the GNU GPL.¹

Nmap is an ideal starting point in a vulnerability assessment or security scan. It can be run from various operating systems and will yield an assortment of open ports, services and enumerated account information.

Nessus

The big gun of the bunch, Nessus is capable of assessing numerous vulnerabilities on numerous platforms. It employs a multitude of plug-ins that are configured to specific operating systems, vulnerabilities, exploits, hacks, backdoors and applications. As this document will describe analyzing both Microsoft and *nix operating systems the details as to what plug-ins to use for each OS will be described in their respective summaries. In all, there are 2159 plug-ins in the database, covering 1349 unique CVE (Common Vulnerabilities and Exposures) ids and 1578 unique Bugtraq ids.

Nessus enumerates a number of services and mountpoints that stand alone tools like rpcinfo, showmount and netcat might, namely the RPC service, NFS mount points and banners. Additional precautions that can be taken will be described later to eliminate these weaknesses.

Disclaimer:

Improperly used to assess a server, Nessus can do great harm!

Nessus assesses vulnerability by actually exploiting the weaknesses via known exploits and hacks. Proceed with extreme caution and understanding before engaging this tool.

OS Assessment – Microsoft Windows

Run Nmap first to get a good snapshot of the server and its potential weaknesses.

Once a list of open ports and available services is in hand, Nessus can be used to target those specific vulnerabilities and identify the corrections that need be made.

Consider the following example:

Nmap yields the fact that port 21 is open on a server destined only to act as domain controller. Given that that a DC does not require ftp it would be essential to study further the nature of this open port. On a Windows 2000 server to determine if the Windows server is offering ftp navigate to Add/Remove Programs then to Add/Remove Windows Components. Then under IIS if you discover ftp checked you can disable it

along with IIS if web services aren't required. Best practices would certainly indicate that IIS has no place on a domain controller to begin with.

After walking through these steps it is possible that they don't show ftp running via the Windows OS. The implication then is that another app is opening this port and must be identified. Nessus can then be tuned to seek more information. Keep in mind that there are almost 100 plug-ins specific to ftp so choose those that seem relevant to a Windows environment. Obviously, BSD ftpd won't be running on Windows 2000. The following plug-ins should provide more detailed information:

IIS FTP server crash

CuteFTP multiple flaws

Windows Administrator NULL FTP password

Multiple Overflows in WS_FTP client

Perhaps another administrator engaged a third party ftp service like CuteFTP, LeapFTP, CesarFTP and SmartFTP. Nessus will discover it if the correct ftp plug-in is utilized.

Nessus offers a family of plug-ins specifically written to analyze every aspect of the Windows platforms.

Given that there are thousands of plug-ins overall and hundreds specific to Windows listing them all here would be a disservice. It should be noted however that the list of top 10 most popular of plug-ins used with Nessus all explore Windows vulnerabilities. The list is as follows, and are without doubt, critical to use when initiating a security scan for on a Windows OS.

<u>Plug-in ID</u>	<u>Name</u>
11835	Microsoft RPC Interface Buffer Overrun (KB824146)
10394	SMB log in
10150	Using NetBIOS to retrieve information from a Windows host
10264	Default community names of the SNMP Agent
12054	ASN.1 Parsing Vulnerabilities (NTLM check)
11412	IIS: WebDAV Overflow (MS03-007)
12209	Microsoft Hotfix for KB835732 (SMB check)
10114	icmp timestamp request
10077	Microsoft Frontpage exploits
11921	Buffer Overflow in the Workstation Service (828749)

For complete lists of Windows-related plug-ins see the following pages at the Nessus web site.

<http://cgi.nessus.org/plugins/dump.php3?family=Windows>

<http://cgi.nessus.org/plugins/dump.php3?family=Windows%20%3A%20User%20management>

Best Practices for Securing a Windows Server

Preface:

There are some remarkable documents and tools available for server hardening at <http://www.nsa.gov/snac/> for NSA's Security Configuration Guides², <http://www.cisecurity.org/> for the Center for Internet Security³ standards, and <http://www.sans.org/top20/> for the SANS Top 20 Internet Security Vulnerabilities⁴. These tools are essential for establishing great recommended best practices and baselines.

These practices should and can be deployed on any Windows server (NT4, Windows 2000, Windows Server 2003). Above all else be certain you need that which is installed. If you're not running web services via IIS, don't install it. The same holds true for DNS, WINS, Services for UNIX and SNMP. Don't need it? Don't install it or leave it on the server if it's there. Even IIS can be limited in its scope. By default it includes ftp, nntp and smtp out of the box. Not allowing file transfers, network news or email services? Then ftp, nntp and smtp aren't necessary. Eliminate them.

1) Auditing

Enable logging first and foremost as follows (at a minimum):

Audit account logon events	Success, failure
Audit account management	Failure
Audit directory service access	Failure
Audit logon events	Success, failure
Audit object access	Failure
Audit policy change	Failure
Audit privilege use	Failure
Audit process tracking	No auditing (auditing this buries the server)
Audit system events	Success, failure

Read the logs regularly and consider a log manager like GFI's S.E.L.M (Security Event Log Manager)

2) Services

Disable all unnecessary services. Keep in mind that, as an additional step, the services deemed unnecessary can be removed entirely via script (see Addendum A) or the Resource Kit tool sc (Service Controller).

This list might typically include Automatic Updates, DHCP Client (assuming your not assigning server addresses dynamically), Fax Service, Indexing Service, Internet Connection Sharing, NetMeeting Remote Desktop Sharing, Remote Registry Service, Telnet, Windows Time and Wireless Configuration.

From the services.msc remember to both stop and disable the service in question.

3) Local Security Policy

- a) Be sure to enable account lockout duration and threshold and reset account lockout counter. Default settings are fine as a minimum.
- b) Password policies are critical. For basic parameters a minimum of eight characters is required, both alpha and numeric. It is suggested that password be a minimum of 12 characters, alpha numeric and symbolic.
- c) Under user rights assignments remove all unnecessary accounts. On servers this typically is the Power User. The author believes that one is either a user or an administrator. Additional accounts like Backup Operators may or may not be in use, if not, eliminate them. Windows Server 2003 automatically installs Remote Help accounts that should be destroyed on sight.
- d) Under security options set additional restrictions for anonymous connections to "do not allow enumeration of SAM accounts and shares." None is too little, no access is too much. Set cached logons to 0 and rename both the administrator and guest account here. Remember to logout and log back in again after doing so assuming you were making these changes under the built-in administrator account.

4) Patches

As Microsoft releases patches for the server OS's it is critical to patch them in a timely manner. In the case of application servers with high availability, it is imperative to check with the vendor to insure that the patches have been tested before deployment. Regardless, patching is one of the most critical administrative tasks in a secure environment.

5) Antivirus

Where possible, so long as it creates no adverse effect on applications, run an antivirus agent. Any product is better than none, and the known names including Trend Micro, McAfee and Symantec is acceptable so long as it is maintained on a daily basis with current DAT updates.

6) Host based firewall or IDS (Intrusion Detection Systems)

Windows Server 2003 offers a built in firewall that can be highly effective in filtering traffic. So long as it is configured to allow the ports required by your applications all other traffic can be blocked at the server regardless of other infrastructure security.

7) Lock the server or log out. Leaving a server unattended that is still logged in with administrative privileges a huge hazard and should be avoided at all costs. Screensavers can be enabled to lock out the server after a set period requiring the user to log back in. Set this interval at ten minutes. Ctrl-Alt-Delete-Enter should you need to walk away will lock the server instantly. Do it without hesitation.

OS Assessment – *nix

Brute force attacks against services on the UNIX platform are most common.

This list may include telnet, ftp, rlogin, rsh, ssh, SNMP, POP and HTTP.

Other vulnerabilities to analyze include data driven attacks creating a buffer overflow, tftp, sendmail, RPC, NFS and X.

Accordingly, direct Nessus to assess weaknesses with plug-ins relevant to any of the following as listed by the SANS Institute⁴:

BIND Domain Name System, Remote Procedure Calls (RPC), Apache Web Server General UNIX Authentication Accounts with No Passwords or Weak Passwords, Clear Text Services, Sendmail, Simple Network Management Protocol (SNMP), Secure Shell (SSH), Misconfiguration of Enterprise Services NIS/NFS and Open Secure Sockets Layer (SSL).

Best Practices for Securing a *nix Server

The same general best practices that apply in a Window environment apply in a UNIX environment. The following will certainly aid in protecting possibly vulnerable services or accounts:

1) Passwords

All users must have a valid password.

Force password changes every 30 to 60 days

The same minimum password length and difficulty as described in the Windows section should apply in *nix environments.

The same account lockout and duration policy as described in the Windows section should apply in *nix environments.

2) Server Holes

Again, if the service is not necessary, disable or uninstall it.

a) BIND/DNS

Don't allow zone transfers, don't give hosts reverse DNS that don't need it and split DNS. If an inside server does not need to resolve in the outside world, see to it that it doesn't.

b) Install IP firewalling

Block inbound UDP not destined to open ports. See Addendum B for more specific firewall details. Iptables is innate on most Linux systems and easily implemented.

Block ICMP echo.

c) Disable unneeded services, disable unneeded services, disable unneeded services. Use ACL services where possible. Access Control Lists can police ingress/egress quite successfully.

d) Eliminate banners. Why announce what you're running? When possible, obfuscate.

e) Alter or eliminate version numbers. No reason why the world should know a given service you may be running.

f) Patch, patch, patch. As new kernels are released or newer versions of vital services like Apache or SSH become available it is imperative to research them. Often they are released as a function of a known vulnerability or achieved exploit. Testing is important and as with Windows, confirmation with any vendors is imperative prior to patching or upgrading.

3) Logging

As with Windows log auditing is critical in a UNIX environment. While a bit more difficult to keep track of in the UNIX environment it is no less essential. Kernel and system logs will track core data while individual applications log themselves like Apache and sendmail.

4) The root account

If you have root, you have everything. If work can be done without being logged in as root consider using an alternative account. Above all else, remember to log out regardless of what account is being used.

Server Summary

A consistent theme runs through the server section and is best remembered in any environment on any platform.

- 1) If you don't need a given service don't leave it running or installed.
- 2) Strong passwords – the longer and more complex the password the less likely it is to be brute forced or compromised.
- 3) Change passwords regularly.
- 4) Watch the logs. One of the best tools for centralized log management available on the market is Kiwi Syslog Daemon at <http://kiwisyslog.com/>. The possibilities with this application are endless. Manage logs from almost any device with output to almost any database. Indispensable to say the least.

Client Holes – Securing the Desktop Environment

It becomes much more difficult to achieve near absolute security in the desktop environment but applying the following checklist can provide an excellent starting point. Given its 90+% ownership of the desktop, it is assumed that all desktop operating systems participating in an average enterprise will be Microsoft. Should at anytime an alternative like Linux be considered on the desktop these practices are still relevant.

1) Education

If users aren't made aware of security concerns and the possible consequences they are far less likely to care. Communication and training go a long way towards attaining a mutual state of understanding and security.

2) Passwords

As with the server environment, here too strong passwords are critical.

A good desktop standard includes eight character alpha numeric passwords, changed every 30 to 60 days. Preferably, security policy for desktops is maintained at domain

level. 5 missed password attempts should be a maximum and lockout of no less than 30 minutes should ensue.

3) Operating systems

Windows 2000 should be the minimum OS in any Windows environment.

It offers better service control and security policy than its predecessors and runs on dated hardware quite admirably. Any OS not utilizing NTFS and its resulting strengthened security cannot be considered viable in the production environment. Windows XP is better still as it, like Windows Server 2003 offers a built in firewall.

4) Patches

Patches on the Windows desktop are entirely critical. Desktops should be current no less than 96 hours after Microsoft releases its monthly patch updates on the second Tuesday of each month. As Microsoft currently releases patches on a rigid timeline, planning and coordination for enterprise patching can be planned for accordingly. Microsoft and others like GFI offer products that support patching in large environments with limited resources. In essence, a central server either pushes or offers patches to the desktops as they log on. This is a viable consideration when personnel resources are limited. Given the repetitive vulnerabilities in Internet Explorer staying patched is vital.

5) Service Packs

Service Packs typically roll together all patches released since the last service pack as well as any system improvements identified by Microsoft.

Windows 2000 Service Pack 4 should be a minimum on Windows 2000 clients at the time of this writing. Stay current accordingly.

Windows XP SP 2 is the most recent on XP clients.

6) Antivirus

As critical as MS patches, antiviral agents will keep Trojans and worms at bay should they pass the firewall bastion or enter via email or rogue laptops. As stated in the server section, any vendor's offering is acceptable so long as it is maintained. A daily update schedule should be configured on the agent so DATs are kept current daily.

7) Lock the workstation

Screen savers in Windows 2000 or later can be configured to automatically lock the PC after a set time of no use. Ten minutes is suggested. Better still, as the user gets up and walks away from a PC, the simple act of typing Ctrl-Alt-Delete-Enter will instantly lock the workstation. If a malicious internal user can't get on the network the effort required to create havoc just increase exponentially. Physical security can go a long way.

8) Services

Here too unnecessary services should be disabled or eliminated much as on servers. The list might typically include Automatic Updates, Fax Service, Indexing Service, Internet Connection Sharing, NetMeeting Remote Desktop Sharing, Remote Registry Service, Telnet, Windows Time and Wireless Configuration. Assess these requirements the same way you did with servers.

9) Use the Windows Firewall in Windows XP. One more layer of security can only enhance the big picture.

Conclusion

Common sense can go a long way in this endeavor. Use it. If you believe you can be compromised you might be. If your instincts tell you to change your password you should do so. There's no such thing as too paranoid in the effort to secure your environment. If a weakness can be exploited, somehow and somewhere it will be. Do the simple things first. Change and strengthen passwords, remove that which is not needed and monitor all activity. From there fine-tune your efforts with continued best practices and policy. Keep your users informed and information available. Never rely on the notion that existing protections are enough. Statements like "We're behind a firewall, that's enough" could be disastrous. A talented hacker can navigate around firewalls and if additional layers of security aren't available he will quickly achieve end game. But should he arrive inside your network space only to find obfuscated versioning, nonexistent services, highly difficult passwords and fully patched systems he might just move on to another target. And thus, through some very simple action, the end game is yours.

¹ Nmap, www.insecure.org/nmap

² NSA Security Configuration Guides, www.nsa.gov/snac/

³ Center for Internet Security <http://www.cisecurity.org/>

⁴ SANS Top 20 Internet Security Vulnerabilities <http://www.sans.org/top20/>

⁵ SANS <http://www.sans.org>

Addendum A

Remove service script (Windows 2000 or later)

Cut and paste this code to a text editor, modify <enter service name here> with the name of the service you wish to remove and save the files as remove_service.vbs or a name of your choosing. Double clicking on the file from Explorer will execute it. Refresh your Services view and the service should be gone.

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:" _
    & "{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")
Set colListOfServices = objWMIService.ExecQuery _
    ("Select * from Win32_Service Where Name = '<enter service name here>'")
For Each objService in colListOfServices
    objService.StopService()
    objService.Delete()
Next
```

Remove services remotely with sc (Service Controller) from the Windows 2000 or Windows Server 2003 Resource Kit

- 1) Install the Resource Kit on your admin host
- 2) Confirm sc exists in your path by typing sc /? at command line
- 3) Use runas /user: domain\username cmd to open a command window with privileges for the domain the server resides in.
- 4) Create a .bat to modify the server and services you wish and execute.

Syntax for remote removal of services from server or PC:

```
sc \\
```

Addendum B

If we assume that firewalls are critical to enterprise security than a list of certain ports to be blocked is critical. As always, caution when blocking ports is suggested to avoid breaking services that may be required. The following is a list provided by the SANS Institute. Another great source for port discovery can be found at <http://www.cirt.net/cgi-bin/ports.pl>

Name	Port	Protocol	Description
Small services	<20	tcp/udp	small services
FTP	21	tcp	file transfer
SSH	22	tcp	login service
TELNET	23	tcp	login service
SMTP	25	tcp	mail
TIME	37	tcp/udp	time synchronization
WINS	42	tcp/udp	WINS replication
DNS	53	udp	naming services
DNS zone transfers	53	tcp	naming services
DHCP server	67	tcp/udp	host configuration
DHCP client	68	tcp/udp	host configuration
TFTP	69	udp	miscellaneous
GOPHER	70	tcp	old WWW-like service
FINGER	79	tcp	miscellaneous
HTTP	80	tcp	web
alternate HTTP port	81	tcp	web
alternate HTTP port	88	tcp	web (sometimes Kerberos)
LINUXCONF	98	tcp	host configuration
POP2	109	tcp	mail
POP3	110	tcp	mail
PORTMAP/RPCBIND	111	tcp/udp	RPC portmapper
NNTP	119	tcp	network news service
NTP	123	udp	time synchronization
NetBIOS	135	tcp/udp	DCE-RPC endpoint mapper
NetBIOS	137	udp	NetBIOS name service
NetBIOS	138	udp	NetBIOS datagram service
NetBIOS/SAMBA	139	tcp	file sharing & login service
IMAP	143	tcp	mail
SNMP	161	tcp/udp	miscellaneous
SNMP	162	tcp/udp	miscellaneous
XDMCP	177	udp	X display manager protocol

BGP	179	tcp	miscellaneous
FW1-secureremote	256	tcp	CheckPoint FireWall-1 mgmt
FW1-secureremote	264	tcp	CheckPoint FireWall-1 mgmt
LDAP	389	tcp/udp	naming services
HTTPS	443	tcp	web
Windows 2000 NetBIOS	445	tcp/udp	SMB over IP (Microsoft-DS)
ISAKMP	500	udp	IPSEC Internet Key Exchange
REXEC	512	tcp	} the three
RLOGIN	513	tcp	} Berkeley r-services
RSHELL	514	tcp	} (used for remote login)
RWHO	513	udp	miscellaneous
SYSLOG	514	udp	miscellaneous
LPD	515	tcp	remote printing
TALK	517	udp	miscellaneous
RIP	520	udp	routing protocol
UUCP	540	tcp/udp	file transfer
HTTP RPC-EPMAP	593	tcp	HTTP DCE-RPC endpoint mapper
IPP	631	tcp	remote printing
LDAP over SSL	636	tcp	LDAP over SSL
Sun Mgmt Console	898	tcp	remote administration
SAMBA-SWAT	901	tcp	remote administration
Windows RPC programs	1025	tcp/udp	} often allocated
Windows RPC programs	to		} by DCE-RPC portmapper
Windows RPC programs	1039	tcp/udp	} on Windows hosts
SOCKS	1080	tcp	miscellaneous
LotusNotes	1352	tcp	database/groupware
MS-SQL-S	1433	tcp	database
MS-SQL-M	1434	udp	database
CITRIX	1494	tcp	remote graphical display
WINS replication	1512	tcp/udp	WINS replication
ORACLE	1521	tcp	database
NFS	2049	tcp/udp	NFS file sharing
COMPAQDIAG	2301	tcp	Compaq remote administration
COMPAQDIAG	2381	tcp	Compaq remote administration
CVS	2401	tcp	collaborative file sharing
SQUID	3128	tcp	web cache
Global catalog LDAP	3268	tcp	Global catalog LDAP
Global catalog LDAP SSL	3269	tcp	Global catalog LDAP SSL
MYSQL	3306	tcp	database
Microsoft Term. Svc.	3389	tcp	remote graphical display
LOCKD	4045	tcp/udp	NFS file sharing

Sun Mgmt Console	5987	tcp	remote administration
PCANYWHERE	5631	tcp	remote administration
PCANYWHERE	5632	tcp/udp	remote administration
VNC	5800	tcp	remote administration
VNC	5900	tcp	remote administration
X11	6000-6255	tcp	X Windows server
FONT-SERVICE	7100	tcp	X Windows font service
alternate HTTP port	8000	tcp	web
alternate HTTP port	8001	tcp	web
alternate HTTP port	8002	tcp	web
alternate HTTP port	8080	tcp	web
alternate HTTP port	8081	tcp	web
alternate HTTP port	8888	tcp	web
Unix RPC programs	32770	tcp/udp	} often allocated
Unix RPC programs	to		} by RPC portmapper
Unix RPC programs	32899	tcp/udp	} on Solaris hosts
COMPAQDIAG	49400	tcp	Compaq remote administration
COMPAQDIAG	49401	tcp	Compaq remote administration
PCANYWHERE	65301	tcp	remote administration

ICMP: block incoming echo request (ping and Windows traceroute), block outgoing echo replies, time exceeded, and destination unreachable messages except "packet too big" messages (type 3, code 4). (This item assumes that you are willing to forego the legitimate uses of ICMP echo request in order to block some known malicious uses.)

In addition to these ports, block spoofed addresses: packets coming from outside your company sourced from internal addresses, private addresses (RFC1918) and IANA reserved addresses (for details, see <http://www.iana.org/assignments/ipv4-address-space>). It is also suggested that you block packets bound for broadcast or multicast addresses. Specifically blocking source routed packets or any packets with IP options set will be advantageous as well.

You should also apply egress filters on your border routers to block spoofed packets from originating from your network. Only allow packets sourced from your assigned addresses to be routed out of your organization.