Global Information Assurance Certification Paper

K. Chesley McColl
GSEC (v.1.3)

Know your network: a method for securing your network

For many systems administrators, the task of implementing network wide security is very daunting, but the task can be broken down into a methodical process. The first step to securing your network is to know what you are trying to secure. I work in a large university research division. We have 4 systems administrators, who are maintaining a network of many different computers (Mac's, PC's, Sparc's, and Alpha's running many different operating systems), and also different printers, switches, routers, and Network Attached Storage boxes. Documentation is key; at any given time any administrator should be able to tell if a new device has been added to the system, and has it been hardened. This can be accomplished by having the right set of tools and by having procedures in place for using the tools. This also means we need to develop a good baseline for comparison.

Once we have our baseline, our approach to securing the network is to start from the outside connection and work our way in to the hosts. By securing the outside connection first with a packet filter or firewall, we buy ourselves some time to work on host defenses. Always keep in mind that a secure system today, will not be secure tomorrow. The comparisons to the baseline and the host hardening should be done routinely.

Know your network: NMAP is a free tool that can be used to scan entire networks, informing you not only which hosts are up, but also the services running, and in some cases it can identify the operating system. It is important to note that all network machines (routers, switches, hubs printers, NAS boxes, and computers) can be running various services that can be compromised. Case in point, almost all devices use the Simple Network Management Protocol (SNMP), which if exploited could result in a denial-of-service condition, or allow the attacker to gain access to the device. For more information on the SNMP exploit check the URL http://www.cert.org/advisories/CA-2002-03.html.

Before you run NMAP, get permission to run the scans on your network. Once you have the go ahead, running "**nmap –n –v -sP –oN baseline.map network/mask**" will give a network snap shot of all machines that are up. By running this nmap command frequently, and piping the output to a new file, you can easily determine if a new device has been added or if a device isn't up by comparing the output with your baseline network map.

Having the baseline is great, but you really need to identify all the devices, at a minimum the type of device and the operating system it is running. It is possible

to run nmap to discover this information, however in a production environment this is not a wise decision. The basic method of OS fingerprinting is to send the device malformed packets and make note of the response to those packets. This type of finger printing can have adverse affects on some systems, so the brute force method is how we chose to get this information.

Next by running "**nmap -v –sT –oN baseline.tcp_services network/mask**" a snap shot of all the TCP services running is obtained, it is also a good idea to run this periodically, so that you can compare it against your baseline TCP network services map. And finally if you have root privileges, run "**nmap –v –sU –oN baseline.udp_services network/mask**" which gives a snap shot of all the UDP services running. I found that this scan works on a Solaris machine if scanning from another machine, unfortunately Solaris does not support this scan when run locally, but running with the options **–P0 –sT localhost** does work. As taken from the man pages for nmap:

> Unfortunately UDP scanning is sometimes painfully slow since most hosts implement a suggestion in RFC1812 (section 4.3.2.8) of limiting the ICMP error message rate. For example, the Linux kernel (innet/ipv4/icmp.h) limits destination unreachable message generation to 80 per 4 seconds, with a ¼ second penalty if that is exceeded. Solaris has much more strict limits (about 2 messages per second) and thus takes even longer to scan. *Nmap* detects this rate limiting and slows down accordingly, rather than flood the network with useless packets that will be ignored by the target machine.
>
> As is typical, Microsoft ignored the suggestion of the RFC and does not seem to do any rate limiting at all on Win95 and NT machines. Thus we can scan all 65K ports of a Windows machine **very** quickly. Woop!

Securing the outside connection: A new tool that runs on a unix machine with perl, can do a non-evasive audit of a Cisco router. The tool is rat-1.0 and can be downloaded from http://www.cisecurity.org.

Rat is a free tool that will either use your tftp dump file of the running configuration, or it will login and acquire your running configuration on the fly. It then audits against a set of rules and produces several output files. One output file is the rules that the router was tested for; another is a file that can be used to fix the configuration by simply cutting and pasting the output into the configuration mode on the router. The documentation is great. If you run the tool without any modifications "**rat –nosnarf cisco.cfg**", the rules checked for compliance are those found in the NSA Security Recommendation Guide.

The default rules contain checks for access control, routing, logging, and services. The default rules do not check for Loopback 0 existing (logging), forbid SNMP read-write (access), forbid SNMP without ACLs (access), tftp source-interface Loopback0 (access), no local logins (access), no tftp-server (services), Tunneling interfaces (routing). Also the default rules contain no checks for using AAA (TACACS+ or RADIUS), these include the following rules new-model (AAA), authentication login (AAA), authentication enable (AAA), accounting exec(AAA), accounting commands (AAA), accounting network(AAA), accounting connection(AAA), accounting system(AAA), source-interface Loopback0 (AAA). To check these rules specifically you need to add the option –limitclassto logging access AAA services routing.

As you can see from the output below, the importance of a given rule is given, 10 being the most important, whether or not the rule checked passed or failed, what the rule was, for which device, the line number in the configuration file, and the instance. Here is an example output from running rat with the default rule set.

Importance	Pass/Fail	Rule	Device	Line#	Instance
10	pass	IOS - enable secret	cisco.cfg		
10	pass	IOS - forbid SNMP community public	cisco.cfg		
10	FAIL	IOS - Apply telnet ACL	cisco.cfg	96	vty0 4
10	FAIL	IOS - Define telnet ACL	cisco.cfg	1	n/a
10	pass	IOS - login	cisco.cfg		
10	pass	IOS - require line passwords	cisco.cfg		
10	FAIL	IOS - no snmp-server	cisco.cfg	82	n/a
10	pass	IOS - no ip http server	cisco.cfg		
10	pass	IOS - forbid SNMP community private	cisco.cfg		
7	pass	IOS - Apply egress filter	cisco.cfg		
7	pass	IOS - Apply ingress filter	cisco.cfg		
7	pass	IOS - no service config	cisco.cfg		
7	pass	IOS 12 - no directed broadcast	cisco.cfg		
7	pass	IOS 12 - no tcp-small-servers	cisco.cfg		
7	FAIL	IOS - egress filter definition	cisco.cfg	1	n/a
7	pass	IOS 12 - no udp-small-servers	cisco.cfg		
7	pass	IOS - encrypt passwords	cisco.cfg		
7	FAIL	IOS - exec-timeout	cisco.cfg	90	con 0
7	FAIL	IOS - ingress filter definition	cisco.cfg	1	
7	FAIL	IOS - no cdp run	cisco.cfg	1	n/a
7	FAIL	IOS - no ip source-route	cisco.cfg	1	n/a
5	pass	IOS - clock summer-time	cisco.cfg		
5	FAIL	IOS - clock timezone GMT 0	cisco.cfg	1	n/a
5	pass	IOS 12 - no finger service	cisco.cfg		
5	pass	IOS - enable logging	cisco.cfg		
5	pass	IOS - no ip bootp server	cisco.cfg		

5	FAIL	IOS - no ip proxy-arp	cisco.cfg	32	FastEth0
5	FAIL	IOS - ntp server	cisco.cfg	1	n/a
5	FAIL	IOS - logging buffered	cisco.cfg	1	n/a
5	FAIL	IOS - service timestamps	cisco.cfg	1	n/a
5	FAIL	IOS - set syslog server	cisco.cfg	1	n/a
5	FAIL	IOS - vty transport telnet	cisco.cfg	96	vty 0 4
3	FAIL	IOS - disable aux	cisco.cfg	93	aux 0
3	FAIL	IOS - logging trap debugging	cisco.cfg	1	n/a
3	FAIL	IOS - logging console critical	cisco.cfg	1	n/a
3	FAIL	IOS - set syslog facility	cisco.cfg	1	n/a

Summary for cisco.cfg

#Rules	#Passed	#Failed	%Passed
37	17	20	45

PerfectWeightedScore ActualWeighedScore %WeightedScore
292 129 44
Overall Score (0-10)
4

Note: PerfectWeightedScore is the sum of the importance value of all rules.
ActualWeightedScore is the sum of the importance value of all rules passed, minus the sum of the importance each instance of a rule failed

And similarly the output for fixing the above Failed rules:

```
! The following commands may be entered into the router to fix problems found.
! They must be entered in config mode (IOS).
! Fixes which require specific information (such as uplink interface device name
! or specific access list numbers) are listed but commented out.  Examine them,    ! edit and uncomment.
!
! THESE CHANGES ARE ONLY RECOMMENDATIONS.
! CHECK THESE COMMANDS BY HAND BEFORE EXECUTING.  THEY MAY BE WRONG.  THEY MAY BREAK YOUR ROUTER.
! YOU ASSUME FULL RESPONSIBILITY FOR THE APPLICATION OF THESE CHANGES.
!

!
!line vty 0 4
!access-class EDIT-BY-HAND in
!exit
!
!
!no access-list EDIT-BY-HAND
!access-list EDIT-BY-HAND permit tcp EDIT-BY-HAND 0.0.0.255 any eq telnet
!access-list EDIT-BY-HAND permit tcp host EDIT-BY-HAND any eq telnet
!access-list EDIT-BY-HAND deny   ip any any log
!
no snmp-server
!no access-list 108
!access-list 108 deny   ip any 10.0.0.0 0.255.255.255 log
!access-list 108 deny   ip any 127.0.0.0 0.255.255.255 log
```

```
!access-list 108 deny   ip any 172.16.0.0 0.15.255.255 log
!access-list 108 deny   ip any 192.168.0.0 0.0.255.255 log
!access-list 108 permit EDIT-BY-HAND any
!access-list 108 deny   ip any any log
line con 0
exec-timeout 5 0
exit

!no access-list 107
!access-list 107 deny   ip 10.0.0.0 0.255.255.255 any log
!access-list 107 deny   ip 127.0.0.0 0.255.255.255 any log
!access-list 107 deny   ip 172.16.0.0 0.15.255.255 any log
!access-list 107 deny   ip 192.168.0.0 0.0.255.255 any log
!access-list 107 deny   ip EDIT-BY-HAND any
! YOUR INTERNAL ADDRS HERE ^^^^^^^^^^^
!access-list 107 deny   ip any 10.0.0.0 0.255.255.255 log
!access-list 107 deny   ip any 127.0.0.0 0.255.255.255 log
!access-list 107 deny   ip any 172.16.0.0 0.15.255.255 log
!access-list 107 deny   ip any 192.168.0.0 0.0.255.255 log
!access-list 107 permit ip any any
!
no cdp run
no ip source-route
clock timezone GMT 0

int FastEthernet0/1
no ip proxy-arp
exit

ntp server \d+\.\d+\.\d+\.\d+
!ntp source EDIT-BY-HAND
logging buffered
service timestamps log datetime show-timezone
!logging EDIT-BY-HAND <== Put your syslog server IP addr there.

line vty 0 4
transport input telnet
exit

line aux 0
no exec
transport input none
exit
```

logging trap debugging
logging console critical
logging facility local1

So as you can see, the rat tool can be very effective, and extremely easy to use, in a short period of time.

If you don't have a Cisco router you could use known good practice, a good starting place for guidance is the NSA Security Recommendation Guide. Distribution of the guide is currently subject to the terms of an NSA Legal Notice, available at **http://nsa2.www.conxion.com/cisco/notice.htm**

Securing the outside connection step-by-step: Physical security is the first line of defense; any knowledgeable person can do password recovery procedures quickly at the console. Passwords are the second line of defense, there should not be any local user accounts configured on the router, and the passwords themselves should be encrypted if possible.

The SNMP community strings public and admin should be removed, and if SNMP isn't needed, disable the SNMP server. Otherwise set up hard-to-guess community strings.
Management of your router should be done from a secure network, preferably from the console itself. SNMP version1 uses clear text authentication, and usually is implemented with periodic polling, thus the password is easily sniffed. You should seriously consider not using SNMP over the public internet.

Disable the following:

1. CDP – Cisco discovery protocol
2. TCP small servers
3. UDP small servers
4. finger
5. http server
6. bootp
7. ip source routing (rarely used for legitimate traffic)
8. proxy ARP
9. IP directed broadcast ("smurf" denial of service attack)
10. classless routing
11. IP redirects
12. NTP service
13. SNMP (as noted above)
14. DNS

Basic Filtering should be done for the following:

1. Inbound Traffic should not contain private addresses or IP address from the internal network. (anti-spoofing)
2. Outbound Traffic should not contain private addresses or external IP addresses. (anti-spoofing)

If you cannot apply an ACL that is based on blocking everything except a specific list of ports, then the following website could be quite helpful in determining some ports that should be blocked. Such as RPC bind UDP port 111 and NetBIOS ports UDP 135-139. http://www.securityspace.com/smysecure/daudit_ports.html

System logging should be turned on, and can be sent to console, syslogd of a unix machine or kept locally in the routers RAM buffer. Some of the most important events to log would be interface status changes, changes to the system configuration, and matches to any packet filters you have put in place.

Finally the setting up of a banner message helps to prosecute later if a cracker does get in. Typical content of the banner should contain:

> A notice that the system is to be logged in to or used only by specifically authorized personnel, and perhaps information about who may authorize use. A notice that any unauthorized use of the system is unlawful, and may be subject to civil and/or criminal penalties. A notice that any use of the system may be logged or monitored without further notice, and that the resulting logs may be used as evidence in court.

Of course a firewall that can do more then just packet filtering is better yet. Selecting the firewall that is appropriate for your needs is the first step. There are proxy, stateful packet inspection, or a hybrid firewalls. Proxy firewalls work at the application level, basically protecting your network by hiding the ip address for the application running. Stateful packet inspection works at the network level, by examining the packet headers, and remembering prior connections the stateful firewall evaluates if the connect should be terminated by a set of rules. And the hybrid is a combination of the two of these.

Stay up to date: visit the vendors website for known vulnerabilities. A couple of good sites to have bookmarked for new vulnerabilities are:

1. The Internet Storm Center http://www.incidents.org/
2. The CERT Coordination Center http://www.cert.org

Switch Security is often overlooked, but after the recent SNMP exploit everyone should be looking at their switches. On each switch:

Enable the following (if possible)

1. password protected login
2. banner

3. ssh logins
4. tacacs

Disable the following (if possible):

1. SNMP
2. http server
3. ip source routing
4. bootp

Solaris Host Hardening: This can be very time consuming and requires a significant amount of expertise to secure a system properly. Fortunately we don't have to start from scratch. For a Solaris system, there are several good free tools out there.

An excellent comparison of the tools can be found at:
http://www.netsecurity.pl/www.boran.com/security/sp/comparison1.html

JASS- Jumpstart Architecture and Security Scripts is a solution provided by Sun for hardening a Solaris 2.5.1-2.8 system for use in the DMZ. Although JASS itself is not supported by Sun, the final system configuration is supported by Sun. One nice feature of this tool is that it can be run after patch installs to verify that the system is still hardened.

YASSP- Yet another Solaris Security Package, this package was written with the intent of automating the tasks of hardening a Solaris 2.6-2.8 system for use in the DMZ.
It includes the installation of GNU gzip, GNU RCS, tcpwrapper with rpcbind, tripwire and some cronjob for checking logs and archiving. The script SECclean, actually removes some files and edits other system files, this is all controlled with a configuration file that you can modify.

TITAN- design goals for this collection of modules are to improve or audit the security of a UNIX system. Titan helps to consistently produce consistent secure systems, and because of its modularity it is easily customized for a particular site/machine. Titan does not fix or patch known security bugs, but the end product should be a more secure system then what you started with.

In addition to the changes recommended by the above programs it is a good idea to disable inetd and use standalone applications. Inetd provides an easy way for a hacker to leave a back door into your system, which could be easily overlooked by a systems administrator. If you can't totally disable inetd, then weed out as many services as possible and use tcpwrapper to secure the remaining services, which can be downloaded from ftp://ftp.porcupine.org/pub/security/index.html. Tcpwrapper is a tool that will log all attempts to start a service, it will do access

control, not only checking for a password but also verifying the host that the request came from, and if everything checks out it will start the service. This is all accomplished unbeknownst to the user.

Check for and secure your setuid and setgid programs. These programs if exploited can easily permit a local user root access.

Setup a host based firewall. There are several free packages out there. Ipfilter is one packet filter that can be used on a Sun server and can be downloaded from http://coombs.anu.edu.au/~avalon/ip-filter.html.

Even though your system is hardened today, tomorrow a new vulnerability will come out. Be proactive, check for new patches on a regular basis, a cluster of recommended patches can be downloaded from: http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access.

If you have Sun maintenance there is a great tool to automate the patch update process, this can be downloaded from: http://sunsolve.sun.com/private-cgi/show.pl?target=resources/patchdiag

Windows 2000/NT Host Hardening: Unfortunately for the Windows systems there are not many good free tools out there and hardening the system is very time consuming and requires a significant amount of expertise. There are however many white papers on how to secure your systems by hand to name a few:

1. Windows 2000 Security step-by-step, SANS, http://www.sansstore.org/
2. Hardening windows 2000, Part One: Seeing the Forest in spite of the trees http://online.securityfocus.com/infocus/1296
3. Windows 2000 and IIS 5.0 Hardening Checklist/Guide http://www.shebeen.com/w2k/

One free tool, the Cerberus Internet Scanner written and maintained by Cerberus Information Security, Ltd, will locate and fix security holes on Windows NT or 2000. CIS can be downloaded from http://www.cerberus-infosec.co.uk/cis.shtml. This tool is updated on a regular basis, is modular and implemented as a DLL's, so the administrator can run each scan individually. The modules include WWW, SQL, ftp various NT checks, NMTP, POP3, DNS, and finger to name a few. It checks for over 300 different security holes, and can be run in batch mode or through a GUI.

On a routine basis the windows systems should be checked for patch levels and updates, this can be done through the Microsoft website: http://windowsupdate.microsoft.com/

There are several free personal firewalls for consumers rank the pc sygate, tiny and zone alarm as the best. For a small price BlackIce Defender, Zone Alarm Professional, or Norton can be purchased. For a comparison of these firewalls: http://www.firewallguide.com/software.htm.

The methodology we have used can be summarized as follows:

1. Secure your outside connection
2. Scan the network for services running
3. Protect the services that are running
4. Be proactive, run checks routinely and monitor for known vulnerabilities

References:

Boran, Sean. "Hardening Solaris". December 11 2001.
URL: http://www.boran.com/security/sp/Solaris_hardening3.html

Chouanard, Jean. "How to install Solaris and have a good host security". Xerox Palo Alto Research Center. November 19 2001. URL: http://www.yassp.org/

Marks, Howard. "Design Tools Come into Focus". September 2001.
URL: http://www.networkcomputing.com/1219/1219f3.html

Taylor, Laura. "Select the right firewall: Part 1". March 2001.
URL:
http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2694089,00.html

Wilson, Jamie. "Securing your Solaris server ". Unix Insider. February 16 2001.
URL: http://www.itworld.com/Comp/2377/UIR010216hardening/

"Cerberus Internet Scanner". URL: http://www.cerberus-infosec.co.uk/cis.shtml

"Improving Security on Cisco Routers". Cisco. October 17 2001.
URL: http://www.cisco.com/warp/public/707/21.html

"Know Your Enemy: Passive Fingerprinting". Honeynet Project. March 2002.
URL: http://project.honeynet.org/papers/finger/

"NMAP Project". 28 December 2001.
URL: http://www.insecure.org/nmap/nmap_relatedprojects.html

"NSA Router Security Configuration Guidelines". NSA. September 2001.
URL: http://nsa2.www.conxion.com/cisco/download.htm

“Solaris[tm] Security Toolkit ("JASS")”. December 10,2001.
URL: http://dcb.sun.com/practices/profiles/solaris_security.jsp

“TITAN 4.0Beta 2”. Aug 31 2001.

URL: http://www.fish.com/titan/TITAN_documentation.html