Global Information Assurance Certification Paper

System Certifications:
An Administrative Makeover

John M Modransky
GSEC Practical
Version 1.4b, Option 2
Submitted January 9, 2004

Case Study: Documenting and Standardizing a Computer System Certification Procedure at a Financial Institution

Abstract

Described in this paper are the administrative controls that were implemented to certify and accredit UNIX (herein referred to as UN*X) and Microsoft Windows (herein referred to as Windows) based computer systems for a financial institution (herein referred to as The Firm).

This InfoSec consultant was assigned the following tasks:

1) perform the UN*X and Windows system certifications
2) develop, document, and publish a standardized methodology document containing the step-by-step actions, both administrative and technical, to perform UN*X or Windows system certifications
3) create a standardized accreditation statement certifying that UN*X and Windows systems conform to a standard configuration and pass a certification process

When initially given the project, there did not exist a formal, documented certification methodology or procedure within The Firm's Information Security (InfoSec) department. When a system certification was requested by other departments within The Firm, the current process was ad-hoc; verbal or email messages were used to request an certification, usually when the system administrator remembered.

After implementing and documenting the process, the system administrators requesting the certifications had a standardized request method. InfoSec engineers had standardized scanning tools and audit procedures, a standardized accreditation statement, and a standardized location where the final accreditation statements are published for review.

Overview

Definitions[5]:
Certification: The comprehensive evaluation of the technical and non-technical security features of an information system and the other safeguards, which are created in support of the accreditation process to establish the extent to which a particular design and implementation meets the set of specified security requirements.

Accreditation: A formal declaration by a Designated Approving Authority (DAA) where an information system is approved to operate in a particular security mode by using a prescribed set of safeguards at an acceptable level of risk.

The Firm has a security policy statement that outlines the system certification requirements. The statement is specific, as defined by SANS: “A policy is typically a document that outlines specific requirements or rules that must be

met. In the information/network security realm, policies are usually point-specific, covering a single area."[1]

The security policy specifies that UN*X and Windows systems are to be setup and configured per the Solaris Security: Step-by-Step version 2 and Securing Windows 2000: Step-by-Step version 1.5 technical documents, which are referenced in The Firm's UN*X and Windows configuration standards guides. Additional security controls are also outlined in The Firm's configuration standards' guides that are specific to their systems' and network needs that are above and beyond what are outlined in the SANS' security guides.

These additional constraints include directory structure names, user and group ownerships, file and directory permissions, use of directory permission sticky bits, group account membership, use of LAN IDs versus local system IDs, password life cycle, and an exception statement allowing RSH in tier one between application and utility servers as required by one third party vendor.

Further, The Firm's security policy also states that systems not certified and accredited cannot be put into production until it has passed the certification process and an accreditation statement has been posted in The Firm's document management system.

At The Firm, both UN*X and Windows system administrators are required to request a system certification for any new computer systems being built or upgraded in their production computing environments. These environments include development, test, and production. This requirement is defined in The Firm's security policy.

The Firm believes strongly in the following statement: "Any time new systems are added, system platforms are changed, or any major organizational modifications are undertaken, you need to redo that risk assessment. 'Security is not a one-time event—it's a practice. A practice that consists of tools, training, metrics, and a methodology. Anything less will be difficult (almost impossible) to maintain'"[4].

Being a financial institution that is responsible for millions of dollars a day in trading transactions, they are serious about implementing a thorough certification procedure that would help mitigate external and internal threats and also provide a controlled and documented process for certification and accreditation.

The Firm's security policy also required that when posting the accreditation statement to The Firm's document management system, support documentation is required within the accreditation statement. This includes a statement listing the date of accreditation, the system name, operating system, the next re-certification date, and the scanning tool output report. Re-certification is required every six months for production systems.

Before

When the certification methodology project was assigned to this InfoSec consultant, certification requests were made using voice mail and email correspondence, the scanning tool used had outdated configurations, system review methods were not standardized, and tracking outstanding certification requests was non-existent. In addition, no standard accreditation statement existed after a system was certified.

The Firm has a three tiered network with the third tier acting as the Internet facing tier, tier two the content delivery or user interfacing tier (i.e. web servers), and tier one is the data tier, hosting the database and application servers. Internet Security System's (ISS) Internet Scanner product was used to perform the network scans on the UN*X and Windows systems.

System scanning was initiated from tier one to scan servers on tier two. Only one system was scanned at a time in order to produce reports with just a single server scanned in the output. This created somewhat of a bottle neck whenever a new environment was rolled out as it was not possible to start multiple ISS scans from the scanning system.

In addition, because the ISS Internet Scanner was run from a laptop in the InfoSec engineer's cubicle, routing constraints in the tier one desktop VLAN prevented scanning tier three. Tier three had to be scanned by taking the laptop into the data center and connecting it directly to a tier one or tier two switch. This problem was a very big productivity killer for the InfoSec engineers when new environments rolled out.

New environments were being rolled out every six months to support a new web based application deployment, and each time a new environment rolled out there were approximately eight to ten UN*X systems and six to eight Windows systems. An alternative vulnerability scanner was needed to support both certifying new and production environments as well as rolling out The Firm's business continuity initiative.

Before the UN*X and Windows certification procedure was revised, certification requests were made ad-hoc. The UN*X or Windows system administrator either verbally asked for or emailed his request. Upon receiving the request, the InfoSec engineer would begin scanning the system and visually checking the operating system configuration per The Firm's security standard's documents[3].

Whether or not the certification request was made for newly deployed systems or for productions systems, the scanning and visual configuration checks were performed during production hours.

When the scanning was complete, the InfoSec engineer would review the scan output and then either send a printed copy of the report back to the system administrator to address the vulnerabilities, or, for systems that passed the

vulnerability scan and visual configuration checks, file the scan report in The Firm's document management system. The final step was to verbally inform the system administrator that the system passed certification and was officially accredited.

Work-in-process tracking was not standardized and left to the individual InfoSec engineers to do on their own. This engineer initially used email printouts to track the requests. Other InfoSec engineers also used email printouts as well as Post-It notes or other paper based methods. There was also no standard turn-around time for completing the certification request.

The only documentation that was being copied to The Firm's document management system was the scanning tool's report output. A spreadsheet located in the main accreditation folder was being used to track system certifications and accreditations. It was maintained in the the document management system to track the systems that had been scanned, but there was no trigger mechanism to prompt the InfoSec department for further system re-certification after the initial system was certified and accredited.

The following Security Essentials material was useful in this stage in reviewing the existing certification procedure:

1. Basic Security Policy, Chapter 8 – evaluating and reviewing The Firm's security policy against the guidelines outlined in this chapter assisted in determining if the existing policy was adequate
2. Access Control and Password Management, Chapter 9 – The Firm's additional constraints, which are above an beyond the SANS Step-by-Step guides, were enhanced based on information from this chapter, specifically the use of strong passwords and minimum Windows service pack installation

During

An InfoSec web portal application was set up by the InfoSec programmer/analysts to handle the system certification and accreditation requests. The portal's data input form required the user to input the system name, operating system (OS), OS version, applications running on system, and IP address assigned to the system.

After the request was submitted through the portal, the portal application emailed the user's request to InfoSec email account, which was a list account composed of the InfoSec engineers in the department. The requester was also sent a copy of the email request and it contained the time frame that the scanning and auditing would be performed in.

Any InfoSec engineer could pick up the portal request email message and begin certifying the system. This method had the possibility of two InfoSec engineers working on the same request. Though this did not happen, since there were only

two engineers at the time who worked on the system certifications, this method was quickly seen as inadequate should the portal application be expanded for other InfoSec work requests.

Tracking certification requests was still left up to the individual InfoSec engineers. Discussion was initiated to expand the use of an existing ticketing system by integrating it with the InfoSec web portal application. InfoSec security programmer/analysts began work to see if the ticketing system had API hooks to allow the portal application to pass data to it, create a certification work ticket, and then let the InfoSec engineers assign the ticket to themselves.

Using the ticketing system would solve two problems immediately

1) it would give visibility to the InfoSec engineers on who was assigned the ticket and who was working on the request
2) it provided a tracking mechanism for both the InfoSec engineer, the InfoSec management team, and the requesting party.

The scanning tool used during this stage of the project was changed from the ISS Internet Scanner to the Nessus scanner[2]. The reason for this was threefold. First, the Nessus scanner has the ability to be launched multiple times on the scanning system by a single InfoSec engineer[2]. This is important because scanning reports must be created with only one system in each report. It was more time consuming when using the ISS scanner as only one scan could be run at a time. When running the Nessus scanner, multiple scans could be run simultaneously.

Secondly, the Nessus scanner was deployed on a single workstation on the main tier one subnet. This accomplished two things right away.

1) scans could be scheduled and batched processed when needing to scan production systems during non-production hours
2) tier three systems could be scanned without having to take a laptop into the data center and performing a scan from there

And, lastly, Nessus was deployed as a way of allowing multiple InfoSec engineers to run scans at the same time. The Firm only had one license for the ISS Internet Security scanner and this created a productivity bottleneck. By having the Nessus scanner on the main tier one subnet, multiple system scanning could be performed simultaneously by multiple InfoSec engineers eliminating the bottleneck caused by single scans and having to take the scanner into the data center.

Having the Nessus scanner running on UN*X also gave InfoSec engineers different ways to run the scanner:

1) from the UN*X command line (to support batch mode)
2) from an X-Window's front end
3) from a JAVA console
4) from a Windows based client[2].

The InfoSec Engineers did not change any of the steps when actually performing the certification for either UN*X or Windows based systems. Systems were scanned by the Nessus scanner and the output reviewed for vulnerabilities.

Several new items were added to The Firm's Windows configuration standard's guide. Specifically, minimum service pack versions were added and minimum password lengths were changed to be consistent with the UN*X configuration standard's guide.

Redlining the reports was also implemented as a way of identifying and highlighting the problems that required the system administrators' attention versus just sending the canned report back to the requester. Although this may sound redundant, it was found that when administrators received the canned report, they thought that every item required their attention or a response to the item.

By sending a standardized email message with brief instructions specifying what was required of them and redlining the vulnerabilities that needed attention, the time line for sending the report, getting the issues fixed, and rescanning the system shrunk significantly. See Appendix A for a sample standardized rejection email message that contains a redlined report.

Visual checks were performed as before, using the UN*X or Windows configuration standard's guides as a checklist. Any additional problems found during this step were also included in the redlined report sent to the system administrators.

After fixing the vulnerabilities or issues found during the scanning and visual audit step, the requester emailed back the redlined report with their own comments explaining how the vulnerability was addressed. Some issues required an exception to be made in those situations where a vulnerable services was required in production.

These exceptions were only granted if there is no patch or work around available and required the requester to get a statement from the third party vendor when a patch or upgrade would be available to address the vulnerability.

After getting the comments back from the requester that the vulnerabilities were addressed, the InfoSec engineer rescanned the system and either certified it or emailed another scan report back to the administrator, redlining the vulnerabilities that need to be addressed. This back-and-forth process continued until the system is certified.

A standardized accreditation statement was drafted and approved for usage by the InfoSec management team. This standardized accreditation statement fulfilled a security policy requirement for an accreditation statement to be published with the scanning report when posted to The Firm's document management system.

The accreditation statement formally notifies the requester that the system has been certified, passing both the configuration requirements review and network vulnerability assessment. See Appendix B for a sample statement that contains the passing scan report.

After emailing the statement to the requester, the InfoSec engineer posted the accreditation statement to The Firm's document management system and sent a copy of the accreditation statement to the system environment owner notifying them that their system was ready for production deployment.

In developing a revised and standardized administrative procedure to use for certifying systems, this consultant found the following Security Essentials sections and chapters particularly helpful in revising the existing system certification process:

1. Attack Strategies and Mitigation, Chapter 13 – review the use of and change the exception of using the Berkeley "R" utilities (rsh, rlogin, and rcp)
2. Vulnerability Scanning, Chapter 15 – the use of open source scanners, how-to, and penetration testing techniques helped to determine the alternative scanning software
3. Risk Management and Auditing, Chapter 18 – quantitative and qualitative assessment approaches, and the use of checklists validated the existing visual review procedure
4. Operations Security, Chapter 24 – auditing types and configuration management helped determine if the new certification process was adequate

After

After deploying the new administrative controls and certification procedure, several changes were made to the process to address business changes within The Firm's computing environment.

There were no changes to the administrative process used. However, continual security education was noted as being the primary method that InfoSec engineers can stay abreast of new security technologies and vulnerabilities. The UN*X and Windows configuration standard's guides would be a continual work in process, going through revisions to insure that the latest controls and constraints are properly applied to computer systems going into or that are already in production.

The Firm's InfoSec programmer/analysts completed the programming interface between the InfoSec web portal application and the ticketing system. Now, when a UN*X or Windows system certification request is made in the web portal application, the request opens up a ticket instead of emailing the entire InfoSec group.

InfoSec engineers are responsible for monitoring the ticketing system for system certification requests and assigning themselves the ticket. The web portal to ticketing system integration was so successful that the web portal was further enhanced to accept requests for other InfoSec tasks

As The Firm moved forward with their business continuity initiative, a second site, code named “The Baha” site, was built out. During this time, the number of system certification requests doubled and tripled at times. Although the InfoSec web portal and ticketing system integration came in toward the end of the business continuity site build out, the newly deployed certification procedure assisted in administrating and controlling the system certifications.

At The Baha site, a second Nessus scanner was deployed in tier one. This aided the InfoSec engineers with their network scanning. Now, they would not have to rely on The Firm's main site for network scanning the Baha site. The link between the two sites could be brought down, and InfoSec engineers at both sites could continue to work without interruption.

As new UN*X and Windows system were deployed and system re-certifications came due, the web portal and ticketing system integration greatly assisted in tracking open system certification work orders. With two to three InfoSec engineers now at each site, the old tracking system would have hindered productivity greatly if relying on broadcast emails for work requests.

An additional change was made to The Firm's document management system when posting the accreditation statements. The old method had the accreditation statement and scan report in one file name “Accreditation and Scan Report”. This file was in a folder named “<System_Name>”. These system folders were further organized under folders that were named either UNIX or Windows.

The naming convention change assisted with the lack of a trigger mechanism for systems requiring re-certification. The naming convention compressed the three layer file and folder hierarchy into a two layer hierarchy. In the new convention, the filename remained the same, but the folder name changed to “yymmdd-<System_Name>”. The operating system folder layer was eliminated.

Now, when listing the top level folder, which contains the date stamped folders, the oldest system folder appears at the top. Although this is not an automated trigger, it does provide for a visual reminder to the InfoSec engineers when listing the folder, as the oldest certified systems appear at the top.

An automated trigger mechanism integrated with The Firm's ticketing system was under consideration when this consultant's contract ended.

The final certification process and standardized administrative procedure was implemented using the suggestions from the following Security Essentials

sections and chapters:

1. The Windows Security Infrastructure, Chapter 25 – active directory considerations as The Firm considers moving from legacy Primary Domain Controller account administration to Microsoft's Active Directory
2. UNIX Security, Chapter 34 – changes in password expiration and aging was implemented on both Windows and UNIX systems and information in this chapter assisted with identifying the capabilities and limitations that UNIX has in this area

Conclusion

There is further room for improvement with the current certification procedure. System re-accreditation needs to be automated, password management administration, and patch management for both UN*X and Windows systems would assist in the certification process by having consistent and automated methods. However, compared with the initial certification process, the existing one today is efficient, conforms to the security policy, and will stand up to an external audit.

Appendix A

To: admin@<company>.com
From: Infosec@<company>.com

Attached is a summarized report outlining vulnerabilities and/or services that need identification on SAMSON. Use the attached report to comment back in, or if you reply in an email, provide the service name/number you're commenting on. All items marked with "**!!!!!**" required feedback, so do not leave any items un-replied to.
Here's what needs to be done:

- See the section "**Analysis of Host**" or "**Description**" for the details. Those items that begin with "**!!!!!**" require fixing or an exception requested if the service is required that is outside the UNIX/Solaris Security Standards <doc-mgmt-sys>/dscgi/ds.py/Get/File-87491/SolarisStds_8_12_02-v2-1.doc and the Solaris SANS <doc-mgmt-sys>/dscgi/ds.py/Get/File-10297/SANS-Solaris.pdf guidelines.
- For the "***unknown***" services/ports (if any) or services identified as "***sometimes-XX***" run the `checkports` script (copy it from SALLEY:`/apps/software/misc`), or `lsof`, on each port and note the application running on the port. If no application is reported, then note this as well.

After you take care of these issues, email me back with the replies for the information requested above, and InfoSec will rescan the system.

Note regarding "unknown" or "sometimes-XX" services: if you can include, in future accreditation requests, all of the known web server ports that are in use, then the step in identifying the ports with `checkports` or `lsof` can be avoided.

<your_name>
x1212

23.10.2003

Network Vulnerability Assessment Report

Sorted by host names

Session name: UNIX	**Start Time:** 23.10.2003 13:23:58
	Finish Time: 23.10.2003 13:29:22
	Elapsed: 0 day(s) 00:05:23
Total records generated: 57	
high severity: 9	
low severity: 20	
informational: 28	

Summary of scanned hosts

Host	Holes	Warnings	Open ports	State
samson	9	20	28	Finished

samson

Service	Severity	Description
swa-rpc-swift (32786/tcp)	**Info**	Port is open
x11 (6000/tcp)	**Info**	Port is open
bpcd (13782/tcp)	**Info**	Port is open

wbem-rmi (5987/tcp)	**Info**	**!!!!! Identify the service running on this port**
vopied (13783/tcp)	**Info**	Port is open
VeritasNetbackup (13722/tcp)	**Info**	Port is open
unknown (32816/tcp)	**Info**	**!!!!! Identify the service running on this port**
unknown (32856/tcp)	**Info**	**!!!!! Identify the service running on this port**
unknown (33236/tcp)	**Info**	**!!!!! Identify the service running on this port**
discard (9/tcp)	**Info**	**!!!!! Identify the service running on this port**
dt-session-rpc9 (32773/tcp)	**Info**	**!!!!! Identify the service running on this port**
dtlogin-cde (32783/tcp)	**Info**	Port is open
nds (1106/tcp)	**Info**	Port is open
unknown (33095/tcp)	**Info**	**!!!!! Identify the service running on this port**
unknown (33094/tcp)	**Info**	**!!!!! Identify the service running on this port**
unknown (33093/tcp)	**Info**	**!!!!! Identify the service running on this port**
unknown (33092/tcp)	**Info**	**!!!!! Identify the service running on this port**
sunrpc (111/tcp)	**Info**	Port is open
unknown (33091/tcp)	**Info**	**!!!!! Identify the service running on this port**
sometimes-rpc15 (32776/tcp)	**Info**	!!!!! Identify the service running on this port
unknown (32785/tcp)	**Info**	**!!!!! Identify the service running on this port**
nds (1105/tcp)	**Info**	Port is open
nfs-clnt-lockd (4045/tcp)	**Info**	Port is open
nfs-nas (2049/tcp)	**Info**	Port is open
unknown (32815/tcp)	**Info**	**!!!!! Identify the service running on this port**
unknown (32784/tcp)	**Info**	**!!!!! Identify the service running on this port**
bmc-perf-sd (10128/tcp)	**Info**	Port is open
UC4 (2345/tcp)	**Info**	Port is open
dtspc (6112/tcp)	**High**	**!!!!! Disable this service** The 'dtspcd' service is running. This service deals with the CDE interface for the X11 system. Some versions of this daemon are vulnerable to a buffer overflow attack which may allow an attacker to gain root privileges on this host. Solution : See http://www.cert.org/advisories/CA-2001-31.html to determine if you are vulnerable or deactivate this service (comment out the line 'dtspc' in /etc/inetd.conf and restart the inetd process) Risk factor : High CVE : CVE-2001-0803 BID : 3517
ddi-tcp-1 (8888/tcp)	**High**	**!!!!! Disable this service or implement the solution. If this services is needed, provide an explanation for it's requirement.** The remote web server is vulnerable to a format string attack. A cracker may exploit this vulnerability to make your web server crash continually or even execute arbirtray code on your system. Solution : upgrade your software or protect it with a filtering reverse proxy Risk factor : High BID : 5384

smtp (25/tcp)	**High**	**!!!!! Upgrade this service to that latest working version. See URLs below for reference.** The remote sendmail server, according to its version number, may be vulnerable to a remote buffer overflow allowing remote users to gain root privileges. Sendmail versions from 5.79 to 8.12.7 are vulnerable. Solution : Upgrade to Sendmail ver 8.12.8 or greater or if you cannot upgrade, apply patches for 8.10-12 here: http://www.sendmail.org/patchcr.html NOTE: manual patches do not change the version numbers. Vendors who have released patched versions of sendmail may still falsely show vulnerabilty. see http://www.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=21950 http://www.cert.org/advisories/CA-2003-07.html http://www.kb.cert.org/vuls/id/398025 Risk factor : High CVE : CAN-2002-1337, CVE-2001-1349 BID : 6991
snmp (161/udp)	**High**	**!!!!! Disable this service** The device answered to more than 4 community strings. This may be a false positive or a community-less SNMP server HP printers answer to all community strings. SNMP Agent responded as expected with community name: private SNMP Agent responded as expected with community name: public SNMP Agent responded as expected with community name: write CVE : CAN-1999-0517, CAN-1999-0186, CAN-1999-0254, CAN-1999-0516 BID : 177, 7081, 7212, 7317
ddi-tcp-1 (8888/tcp)	**High**	**!!!!! Disable this service or implement the solution. If this services is needed, provide an explanation for it's requirement.** It was possible to read the content of /EXT.INI (BadBlue configuration file) by sending an invalid GET request. A cracker may exploit this vulnerability to steal the passwords. Solution : upgrade your software or protect it with a filtering reverse proxy Risk factor : Medium CVE : CAN-2002-1021 BID : 5226
ssh (22/tcp)	**High**	**!!!!! Upgrade the OpenSSH software to version 3.7p2. Also, check to verify that OpenSSL is at version 0.9.7c or higher.** You are running a version of OpenSSH which is older than 3.7.1 Versions older than 3.7.1 are vulnerable to a flaw in the buffer management functions which might allow an attacker to execute arbitrary commands on this host. If you are running a RedHat host, make sure that the command : rpm -q openssh-server Returns : openssh-server-3.5p1-11 (RedHat 9) Solution : Upgrade to OpenSSH 3.7.1 See also : http://marc.theaimsgroup.com/?l=openbsd-misc&m=106375452423794&w=2 http://marc.theaimsgroup.com/?l=openbsd-misc&m=106375456923804&w=2 Risk factor : High CVE : CAN-2003-0693, CAN-2003-0695 BID : 8628
font-service (7100/tcp)	**High**	**!!!!! Disable this service** The remote X Font Service (xfs) might be vulnerable to a buffer overflow. An attacker may use this flaw to gain root on this host remotely. *** Note that Nessus did not actually check for the flaw *** as details about this vulnerability are still unknown Solution : See CERT Advisory CA-2002-34 Risk factor : High CVE : CAN-2002-1317

telnet (23/tcp)	**High**	**!!!!! Change the default password on this account to the standard 'patrol' password.** A valid telnet account has been found by brute force : login: patrol password: patrol Solution: Use strong passwords and difficult to guess usernames Risk factor : High CVE : CAN-1999-0502, CAN-1999-0505, CAN-1999-0516, CAN-1999-0518
finger (79/tcp)	**Low**	**!!!!! Disable this service** The 'finger' service provides useful information to attackers, since it allows them to gain usernames, check if a machine is being used, and so on... Here is the output we obtained for 'root' : Login Name TTY Idle When Where root Super-User pts/10 5 Thu 11:38 sally.domain.com root Super-User pts/3 1d Tue 13:11 172.19.6.155 Solution : comment out the 'finger' line in /etc/inetd.conf Risk factor : Low CVE : CVE-1999-0612
time (37/tcp)	**Low**	**!!!!! Disable this service** A time server seems to be running on this port
smtp (25/tcp)	**Low**	An SMTP server is running on this port Here is its banner : 220 samson.domain.com ESMTP Sendmail 8.9.3+Sun/8.9.3 Thu, 23 Oct 2003 13:18:33 -0500 (CDT)
smtp (25/tcp)	**Low**	**!!!!! Per the solution below, fix this issue.** The remote SMTP server allows anyone to use it as a mail relay, provided that the source address is set to '<>'. This problem allows any spammer to use your mail server to spam the world, thus blacklisting your mailserver, and using your network resources. Risk factor : Medium Solution : reconfigure this server properly CVE : CVE-1999-0819
shell (514/tcp)	**Low**	**!!!!! Disable this service and replace it with SSH/scp/sftp** The rsh service is running. You should Identify the service running on this port and use ssh instead. Solution : Comment out the 'rsh' line in /etc/inetd.conf. Risk factor : Low CVE : CAN-1999-0651
printer (515/tcp)	**Low**	A LPD server seems to be running on this port
login (513/tcp)	**Low**	**!!!!! Disable this service and replace it with SSH/scp/sftp** The remote host is running the 'rlogin' service, a remote login daemon which allows people to log in this host and obtain an interactive shell. You should Identify the service running on this port and use openssh instead (www.openssh.com) Solution : Comment out the 'login' line in /etc/inetd.conf and restart the inetd process. Risk factor : Low CVE : CAN-1999-0651
general/udp	**Low**	For your information, here is the traceroute to 172.19.9.24 : 172.18.4.8 ? 172.19.9.24
ftp (21/tcp)	**Low**	**!!!!! Disable this service and replace it with SSH/scp/sftp** An FTP server is running on this port. Here is its banner : 220 samson FTP server (Version wu-2.6.2(4) Wed Oct 22 10:12:56 CDT 2003) ready.

finger (79/tcp)	**Low**	**!!!!! Disable this service** here is a bug in the remote finger service which, when triggered, allows a user to force the remote finger daemon to display the list of the accounts that have never been used, by issuing the request : finger 'a b c d e f g h'@target This list will help an attacker to guess the operating system type. It will also tell him which accounts have never been used, which will often make him focus his attacks on these accounts. Solution : disable the finger service in /etc/inetd.conf and restart the inetd process, or apply the relevant patches from Sun Microsystems. Risk factor : Medium BID : 3457
telnet (23/tcp)	**Low**	**!!!!! Disable this service and replace it with SSH/scp/sftp** The Telnet service is running. Solution: If you are running a Unix-type system, OpenSSH can be used instead of telnet. For Unix systems, you can comment out the 'telnet' line in /etc/inetd.conf. In addition, many different router and switch manufacturers support SSH as a telnet replacement. You should contact your vendor for a solution which uses an encrypted session. Risk factor : Low CVE : CAN-1999-0619
exec (512/tcp)	**Low**	**!!!!! Disable this service** The rexecd service is open. This service is design to allow users of a network to execute commands remotely. However, rexecd does not provide any good means of authentication, so it may be abused by an attacker to scan a third party host. Solution : comment out the 'exec' line in /etc/inetd.conf and restart the inetd process Risk factor : Medium CVE : CAN-1999-0618
echo (7/tcp)	**Low**	**!!!!! Disable this service** The remote host is running the 'echo' service. This service echoes any data which is sent to it. Solution : - Under Unix systems, comment out the 'echo' line in /etc/inetd.conf and restart the inetd process - Under Windows systems, set the following registry key to 0 : HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableTcpEcho HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableUdpEcho Then launch cmd.exe and type : net stop simptcp net start simptcp To restart the service. Risk factor : Low CVE : CVE-1999-0103, CAN-1999-0635
ddi-tcp-1 (8888/tcp)	**Low**	A web server is running on this port
uucp (540/tcp)	**Low**	**!!!!! Disable this service** An UUCP server seems to be running on this port

daytime (13/tcp)	**Low**	**!!!!! Disable this service** The remote host is running a 'daytime' service. This service is designed to give the local time of the day of this host to whoever connects to this port. The date format issued by this service may sometimes help an attacker to guess the operating system type of this host, or to set up timed authentication attacks against the remote host. In addition to that, the UDP version of daytime is running, an attacker may link it to the echo port of a third party host using spoofing, thus creating a possible denial of service condition between this host and a third party. Solution : - Under Unix systems, comment out the 'daytime' line in /etc/inetd.conf and restart the inetd process - Under Windows systems, set the following registry keys to 0 : HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableTcpDaytime HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableUdpDaytime Then launch cmd.exe and type : net stop simptcp net start simptcp To restart the service. Risk factor : Low CVE : CVE-1999-0103
chargen (19/tcp)	**Low**	**!!!!! Disable this service** The remote host is running a 'chargen' service. When contacted, chargen responds with some random characters (something like all the characters in the alphabet in a row). When contacted via UDP, it will respond with a single UDP packet. When contacted via TCP, it will continue spewing characters until the client closes the connection. An easy attack is 'ping-pong' in which an attacker spoofs a packet between two machines running chargen. This will cause them to spew characters at each other, slowing the machines down and saturating the network. Solution : - Under Unix systems, comment out the 'chargen' line in /etc/inetd.conf and restart the inetd process - Under Windows systems, set the following registry keys to 0 : HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableTcpChargen HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableUdpChargen Then launch cmd.exe and type : net stop simptcp net start simptcp To restart the service. Risk factor : Low CVE : CVE-1999-0103
xdmcp (177/udp)	**Low**	The remote host is running XDMCP. An attacker may use this flaw to capture all the keystrokes of the users using this host through their X terminal, including passwords. Risk factor : Medium Solution : Disable XDMCP

Appendix B

Accreditation Statement for SAMSON
Information Security Services
Accreditation Date: 10/15/03
Next Scheduled Accreditation Date: 3/15/04

Information Security (InfoSec) accredits Sun Solaris system SAMSON as of 10/15/03. InfoSec has verified that SAMSON complies with Sun Solaris production configuration standard. InfoSec has also performed network vulnerability scans against SAMSON. InfoSec finds that SAMSON is in compliance with Sun Solaris production configuration standards.

Network Vulnerability Assessment Report

23.10.2003

Sorted by host names

Session name: UNIX	**Start Time:** 23.10.2003 13:23:58
	Finish Time: 23.10.2003 13:29:22
	Elapsed: 0 day(s) 00:05:23
Total records generated: 38	
high severity: 0	
low severity: 9	
informational: 29	

Summary of scanned hosts

Host	Holes	Warnings	Open ports	State
samson	0	9	29	Finished

samson

Service	Severity	Description
swa-rpc-swift (32786/tcp)	Info	Port is open
bpcd (13782/tcp)	Info	Port is open
x11 (6000/tcp)	Info	Port is open
dt-session-rpc9 (32773/tcp)	Info	Port is open
dtlogin-cde (32783/tcp)	Info	Port is open
VeritasNetbackup (13722/tcp)	Info	Port is open
nds (1105/tcp)	Info	Port is open
nds (1106/tcp)	Info	Port is open
nfs-clnt-lockd (4045/tcp)	Info	Port is open

nfs-nas (2049/tcp)	**Info**	Port is open
dt-session (32784/tcp)	**Info**	Port is open
smtp (25/tcp)	**Info**	Port is open
snmp (161/udp)	**Info**	Port is open
dt-session (32776/tcp)	**Info**	Port is open
sunrpc (111/tcp)	**Info**	Port is open
wbem-rmi (5987/tcp)	**Info**	Port is open
vopied (13783/tcp)	**Info**	Port is open
UC4 (2345/tcp)	**Info**	Port is open
printer (515/tcp)	**Low**	A LPD server seems to be running on this port
general/udp	**Low**	For your information, here is the traceroute to x.x.x.x : x.x.x.y ? x.x.x.w
ddi-tcp-1 (8888/tcp)	**Low**	A web server is running on this port
ftp (21/tcp)	**Low**	**Note: FTP is required until customer implements SSH on the client side.** An FTP server is running on this port. Here is its banner : 220 samson FTP server (Version wu-2.6.2(4) Wed Oct 22 10:12:56 CDT 2003) ready.
xdmcp (177/udp)	**Low**	The remote host is running XDMCP. This protocol is used to provide X display connections for X terminals. XDMCP is completely insecure, since the traffic and passwords are not encrypted. An attacker may use this flaw to capture all the keystrokes of the users using this host through their X terminal, including passwords. Risk factor : Medium Solution : Disable XDMCP

References

1) SANS, “The SANS Security Policy Project/Is it a Policy, a Standard or a Guideline?”, URL: http://www.sans.org/resources/policies/

2) Nessus. Nessus Security Scanner
URLs: http://www.nessus.org/intro.html http://www.nessus.org/features.html

3) SANS, “Security Essentials CISSP 10 Domains Courses ILOT, Section: SANS Security Essentials III: Internet Security Technologies: Risk Management and Auditing”, p 845.

4) McCarthy, Linda. “Intranet Security: stories from the trenches”, Sun Microsystems Press, p118.

5) Ronald L. Krutz and Russell Dean Vines, “The CISSP Study Guide: Gold Edition, 2003”, Wiley Publishing, Inc., p 267