Global Information Assurance Certification Paper

Auditing a Distributed Intrusion Detection System: An Auditors Perspective

GSNA Practical Version 2.0 (amended 14 February 2002)

Author: Darrin Wassom
Date: 1 July 2002

Table of Contents

AUDITING A DISTRIBUTED INTRUSION DETECTION SYSTEM: AN AUDITORS PERSPECTIVE 1

ASSIGNMENT 1 – RESEARCH IN AUDIT, MEASUREMENT, PRACTICE AND CONTROL 4

IDENTIFY THE SYSTEM TO BE AUDITED 4
EVALUATE THE RISK TO THE SYSTEM 6
Intrusion Detection System Probe 6
Central Management Console 7
CURRENT STATE OF PRACTICE 8
IMPROVEMENT OF CURRENT METHODS AND TECHNIQUES 9

ASSIGNMENT 2 – CREATE AN AUDIT CHECKLIST 9

INTRODUCTION: 9
CONVENTIONS USED: 10
OBJECTIVES: 10
SCOPE: 10
A. ADMINISTRATIVE SECTION 11
B. ORGANIZATIONAL STRUCTURE / SYSTEM OVERVIEW, DOCUMENTATION, & TRAINING 13
C. RED HAT LINUX 7.2 OS 14
D. OPENSSH CONFIGURATION FOR IDS PROBE AND MANAGEMENT CONSOLE 21
E. SNORT CONFIGURATION FOR IDS PROBE(S) 24
F. MYSQL – MANAGEMENT CONSOLE 30
G. APACHE – MANAGEMENT CONSOLE 33
H. COMBINED PROBE(S) AND MANAGEMENT CONSOLE ASSESSMENT 37
END OF AUDIT STEPS 40

ASSIGNMENT 3 – CONDUCT THE AUDIT 41

CHECKLIST ITEM C2: PASS 41
Management Console 41
IDS Probe 42
CHECKLIST ITEM C5: PASS 42
Management Console 42
IDS Probe 43
CHECKLIST ITEM C6: PASS 43
Management Console/IDS Probe 43
CHECKLIST ITEM C14: PASS 44
Management Console 44
IDS Probe 44
CHECKLIST ITEM D1: FAIL 44
Management Console 44
IDS Probe 45
CHECKLIST ITEM E2: PASS 45
IDS Probe 45
CHECKLIST ITEM E3: PASS 45
IDS Probe 45
CHECKLIST ITEM H3: PASS 46
Accessing the Management Console from 192.168.1.100 46
Accessing the Management Console from 192.168.1.104 47
CHECKLIST ITEM H4: FAIL 48
SNScan 1.4 Results 49
Back Orifice Pinger 2.0 Results 51
Snot Results 53
Nmap Stealth Scan Results 55

CHECKLIST ITEM H5: FAIL 57
EVALUATE THE SYSTEM 58
EVALUATE THE AUDIT 59

ASSIGNMENT FOUR – FOLLOW UP 60

EXECUTIVE SUMMARY 60
FINDINGS 61
Red Hat Linux 7.2 Operating System 61
Observation: 61
Background/Risk: 61
Recommendation: 61
Cost: 61
Compensating Controls: 62
Observation: 62
Background/Risk: 63
Recommendation: 63
Cost: 63
OpenSSH Configuration for IDS Probe and Management Console 63
Snort Configuration for IDS Probe 64
Apache Web Server – Management console 65
Combined Probe(s) and Management Console Assessment 66
CONCLUSION 67

REFERENCES 68

Table of Figures

Figure 1 - Network Diagram 5
Figure 2 - Windows Pop-Up Box 46
Figure 3 - Successful Login to ACID 47
Figure 4 – Failed Login to ACID 48
Figure 5 - SNScan Client Configuration 49
Figure 6 - SNScan Results in ACID 50
Figure 7 - SNScan Real-Time Email 50
Figure 8 - BO Ping Client Configuration 51
Figure 9 - Back Orifice Ping Results in ACID 52
Figure 10 - Back Orifice Real-Time Email 52
Figure 11 - Snot Results in ACID 54
Figure 12 - Real-Time Response for Snot 54
Figure 13 - Stealth Scan Results in ACID 56
Figure 14 - Stealth Scan Real-Time Results 56

Assignment 1 – Research in Audit, Measurement, Practice and Control

Identify the system to be audited

I am auditing a Distributed Intrusion Detection System (IDS) that will be used by a healthcare organization to satisfy the proposed Health Insurance Portability and Accountability Act (HIPAA) security regulations which require a system to be in place to "guard data integrity, confidentiality and availability". If using a network, the following security measures must be in place to ensure HIPAA compliance: Alarm, Audit Trail, Entity Authentication and Event Reporting.[1] An intrusion detection system will aid in satisfying 3 of the 4 proposed regulations.

The IDS design being audited is currently residing in a test network for evaluation purposes only. The intent of this audit is to certify the design to ensure it will comply with stated security policies and guidelines set forth by the healthcare organization. In order for the IDS design to be certified it must undergo a detailed audit and any deficiencies must be addressed before the system can be installed into the production network.

Because this is a distributed IDS design, there are two main components; the IDS probe and a central management console. Specific operating system and software versions will be listed in the detailed audit checklist. The IDS probe is responsible for "listening" on a given network segment and reporting any signs of intrusion or electronic tampering. This is accomplished by configuring the probe with a set of pre-defined signatures that match known patterns of hostile network traffic. If the IDS probe sees network traffic that matches a signature string, it will log the event and report it to the central management console via a secure channel of communication. The probe will also trigger a real-time notification in the form of a text message sent to a pager and by "ringing" a bell on the central management console.

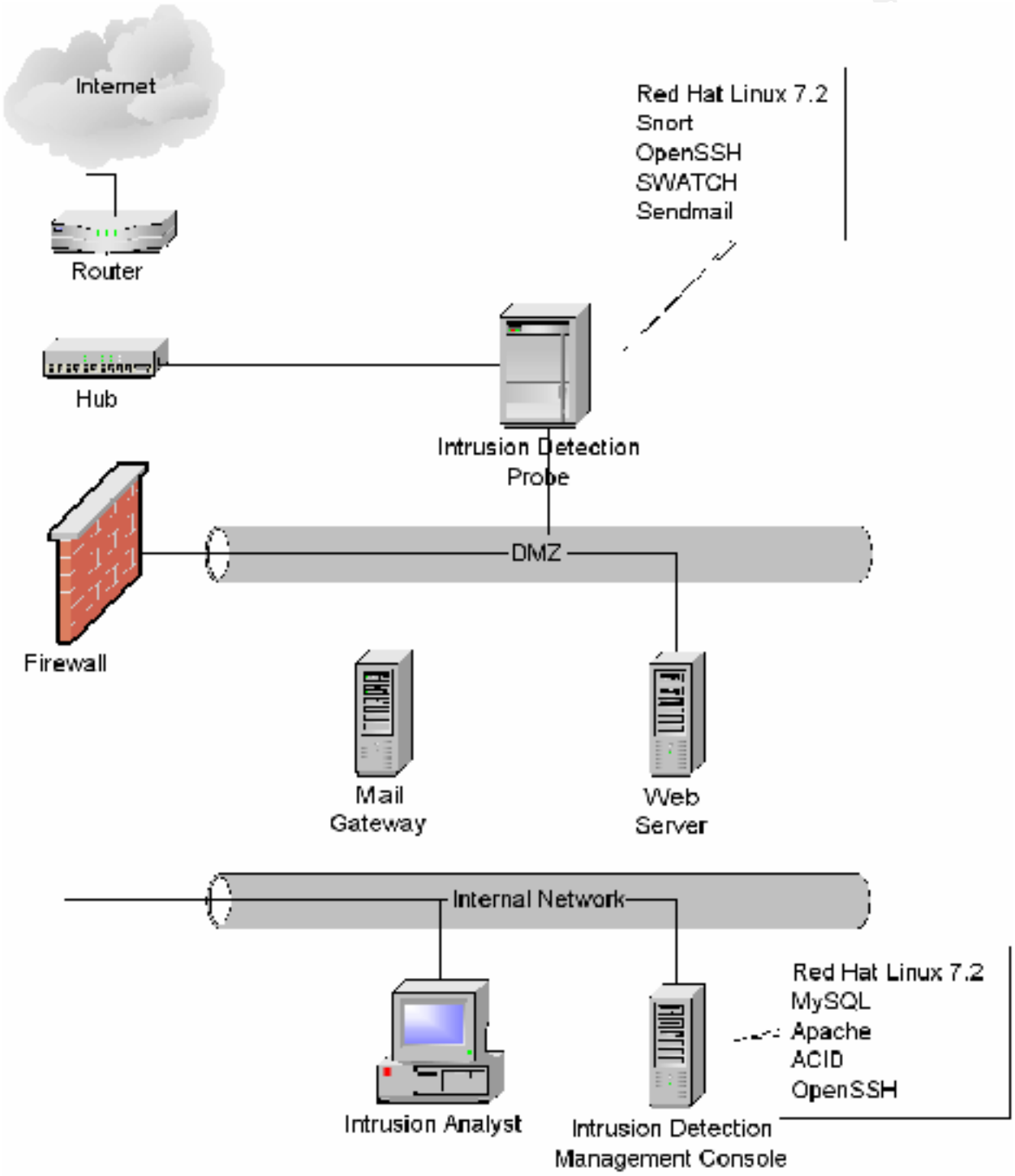
The central management console is responsible for receiving data from the IDS probe and storing it in a MySQL database for analysis. Event data is presented to the intrusion analyst via a web server running Apache and ACID (Analysis Console for Intrusion Databases). Access to the web server is restricted through the use of ACL's (Access Control Lists) that require the intrusion analyst to access from a specific host. In addition, access to the central management console is restricted by a User ID/Password combination. From the management console, the intrusion analyst can perform a variety of tasks to include event analysis, archiving, grouping, event correlation and limited email functions.

The following network diagram shows the IDS probe and central management console placement on the network. As shown, the IDS probe sits between the router and the

[1] "Technical Security Services to Guard Data Integrity, Confidentiality and Availability". Proposed HIPAA Security Regulations. URL: http://www.hipaadvisory.com/regs/securityandelectronicsign/technicalsecur.htm (23 May 2002).

firewall to listen to inbound/outbound traffic while the management console sits on the internal (trusted) network.

Figure 1 - Network Diagram



Evaluate the risk to the system

Based on the documentation and network diagrams I received prior to conducting the audit, it was evident that this system was designed with a "defense in depth" methodology. The design makes effective use of routers, firewalls, screened networks and secure channels of communication between the IDS probe and the management console. In addition, access control lists are being used to restrict access to both components to only those specific hosts that need access. In this case, access is restricted to only the Intrusion Analyst at IP address, 192.168.1.100.

However, given the inherent nature of the Internet and the fact new exploits are released on a daily basis, there is a need to focus the attention of the audit to those areas most prone to vulnerabilities. Given the distributed nature of this design, I am going to divide the areas of risk into two categories; the IDS probe and the management console.

Intrusion Detection System Probe

The IDS probe is the proverbial workhorse in this design. It is charged with listening to all inbound/outbound network traffic and detecting any known signature matches that could indicate an intrusion or electronic tampering. The success of this endeavor is dependent almost totally on the availability of a current signature database. For example, if an attacker is attempting to exploit a particular web server vulnerability and the IDS probe does not have a signature to match, the exploit will go undetected by the probe. Essentially, a probe is rendered useless without the availability of current signatures.

The placement of the IDS probe on the network makes it vulnerable to exploit because it has minimal layers of protection compared to the management console. It is clear from the design that some thought was given to this and that is reflected in the "stealth" interface being used to monitor the network and the policy on the firewall that protects the live interface from being access from outside hosts. Stealth mode simply means that an interface is configured to listen only. The interface does not have an IP address assigned to it and it is virtually impossible to detect its presence on a network. Exposure is further limited by closing all ports except the port needed for secure communication (OpenSSH) on port 22. Despite the access controls, OpenSSH could still be prone to vulnerabilities as evidenced by the most recent announcement from Internet Security Systems (ISS) and OpenBSD.[2] The likelihood of this exploit actually occurring is minimized by restricting access to the probe to a single IP address on the internal network but it is still a risk and must be addressed. Since OpenSSH provides the secure means of communication to the management console, its availability and integrity is vital to the success of intrusion detection.

[2] "Internet Security Systems Security Advisory – OpenSSH". 26 June 2002. URL: http://www.openssh.com/txt/iss.adv (1 July 2002).

The IDS probe relies on sendmail and SWATCH for real-time alert notification. There is a potential for a denial of service attack if too many alerts are detected in a short time frame. This could result in hundreds and possibly thousands of messages queuing up in the outbound mail queue which would impact system and network performance. This chance of this happening is mitigated by the use of thresholds within the SWATCH program but it should still be carefully monitored.

The presence of an IDS probe on a monitored network could alert a would-be attacker that he is being watched. This could result in an amplification of packets being directed toward the IDS probe in an attempt to cause a denial of service attack while also attacking other hosts on the network. The goal would be to "knock" the probe out of commission so the other attacks would go unnoticed. The use of a stealth interface and closing all but the necessary ports limit the exposure of the probe to these types of attacks. If the probe is properly configured then the chance of this exploit actually happening is dramatically decreased.

Central Management Console

The management console performs the analysis of all the data it receives from the IDS probe(s) configured to send traffic to its database. This "magic" is performed through the use of various software applications that include OpenSSH, MySQL, Apache, ACID and PHP. While all components listed are critical the success of the design, the primary risk to the system is compromised integrity of the data stored in the MySQL database. If this data becomes corrupted in any way, the information is useless to the intrusion analyst. Sending all MySQL data (port 3306) through a secure tunnel via SSH reduces this risk. Access to the MySQL database is limited to only two accounts and a single IP address on the internal network. Particular attention is paid to this aspect of the design in the Detailed Audit Plan in Assignment Two.

The Apache web server is the mechanism used to present data to the intrusion analyst via a web browser. Access to the ACID console (served from the Apache web server) is limited to the intrusion analyst only but it could still be vulnerable to exploit. A recent announcement on the Apache Project website indicates that denial of service attacks are possible and, therefore, should be closely monitored for abuse.[3] While the likelihood is minimal because of the access control and layers of defense in place, the potential still exists.

Because the management console resides on the internal (trusted) network, it is afforded the most protection from outside attack. However, it has relatively little protection from internal attacks because its protection is limited to an access control list for the web server and only minimal ports being open on the console itself. Port 22 (SSH) and 80 (HTTP) is open on the internal interface but limited to only the intrusion analyst. The use of a port scanner and vulnerability assessment tools in the detailed audit plan (assignment two) will help decrease the chance of exposure to potential vulnerabilities.

[3] "Apache Security Bulletin". URL: http://httpd.apache.org/info/security_bulletin_20020620.txt. (23 June 2002).

Current State of Practice

Quite surprisingly, there is very little information available on the technical aspects of conducting an intrusion detection system audit. I conducted a thorough search of checklist repositories such as:

- AuditNet (http://www.auditnet.org/)
- ISACA – Information Systems Audit and Control Association (http://www.isaca.org/)
- SANS Reading Room (http://rr.sans.org)
- SANS Posted Practicals for GIAC Systems and Network Auditor (GSNA) and GIAC Certified Unix Security Administrator (GCUX) – (http://www.giac.org/cert.php)

I enjoyed some limited success when I found material relating to penetration testing of IDS[4] and an article from 1999[5] on how to test various IDS engines. The penetration testing methodology was too generic to be used to develop a checklist but it did contain some interesting information regarding steps that should be performed to test IDS sensitivity and sustained packet rates over a period of time. It was interesting to note that Snort was not covered in the article from 1999 which indicates it has only recently gained widespread acceptance as a viable IDS engine. While this material was very interesting and thought provoking, it really didn't contain enough information to use this as a sole source of material for a checklist.

There is a wealth of information regarding auditing various releases of the Linux operating system to include versions of Red Hat Linux. However, there is a lot more material available for earlier releases (Red Hat Linux 6.2) as compared to the latest release from Red Hat (7.3). Information regarding Red Hat 6.2 was plentiful and covered extensively by Lance Spitzner[6] of "Project Honeynet" fame and, until recently, the SANS Institute offered a step-by-step guide to securing Linux in their bookstore. As newer releases of Red Hat were made available, there seems to be less attention paid to creating auditing checklists. This is frustrating because there are some fundamental differences between versions that do not lend themselves well to using outdated checklists. For example, the inetd file structure is different in Red Hat 7.2 because it now uses xinetd.conf rather than the inetd.conf of earlier Red Hat releases.[7] The Center for Internet Security (http://www.cisecurity.org) has a very concise and clearly written Benchmark and Scoring Tool for Linux that contained valuable information.[8]

---

[4] Herzog, Pete. "Open-Source Security Testing Methodology Manual". 26 February 2002. URL: http://www.ideahamster.org/download.html (23 May 2002).
[5] Shipley, Greg. "Intrusion Detection, Take Two". 15 November 1999. URL: http://www.networkcomputing.com/1023/1023f19.html (23 May 2002).
[6] Spitzner, Lance. "Armoring Linux". 19 September 2000. URL: http://www.enteract.com/~lspitz/linux.html (23 May 2002).
[7] Laude, Mary. "Auditing Red Hat Linux 7.0" 23 July 2001. URL: http://www.giac.org/practical/Mary_Laude_GSNA.zip (23 May 2002).
[8] "CIS Level-1 Benchmark and Scoring Tool for Linux". URL: http://www.cisecurity.org/bench_linux.html (15 June 2002).

Because the Apache Project (http://httpd.apache.org/) has been around for several years, there is a wealth of information pertaining to auditing the web server component. Many of the checklists are extremely technical and deal primarily with source code modification and the various add-on modules available. I found the associated manual pages and documentation listed on the site extremely helpful in learning about how Apache works and what steps are needed to ensure a secure configuration. Those specific documents are referenced in the actual checklist I developed and will not be listed here for the sake of brevity.

Finding audit checklists pertaining specifically to the OpenSSH and MySQL aspects of the IDS design was an effort in futility. I found the product web sites (http://www.openssh.com and http://www.mysql.com) to contain an abundance of good material. I also found Gerhard Mourani's text (referenced in the audit plan) to contain the most useful information regarding OpenSSH and was used almost exclusively when preparing audit steps pertaining to OpenSSH.

Improvement of Current Methods and Techniques

Because it was impossible for me to find a concise audit plan that encompassed the many different pieces of software used in the distributed intrusion detection system design, I decided I would create my own checklist using many of the sources listed above. This was an extremely time consuming process but I think the result was worth the effort. I knew going into the planning phases of the audit that I wasn't likely to find a readily available checklist and I also wanted to ensure that the IDS design was taken into account when devising the checklist.

Using the system documentation and network diagrams I received from the healthcare organization requesting the audit, I familiarized myself with the design and researched those areas where I did not have deep technical understanding. In the process of researching the design, I found it beneficial to install the software used to gain better understanding. I wasn't able to duplicate the design but just having some familiarity (issuing commands, noting results, etc) with several pieces of software made it easier to develop the checklist. While developing the checklist, I paid particular attention to the system design and the risks associated with each component used. In addition, I focused on the layered defense deployed in the design and thought carefully about where the IDS probe and management console resides on the network, who has access to these components and the known vulnerabilities associated with each.

Assignment 2 – Create an Audit Checklist

Introduction:

The healthcare organization has requested an audit to certify the design of a proposed intrusion detection system using a distributed architecture. The system being audited is currently residing on a lab network with limited access. Because this is a lab

environment, the hardware used does not meet current production standards and, as such, can not be expected to mirror that of a production network.

The audit will be conducted in the presence of the network engineer responsible for the design, installation and maintenance of the proposed system. Any commands that require administrator/root level access will be entered by the network engineer and observed/noted by the auditor. Both the network engineer and the auditor have agreed to this testing methodology.

Conventions Used:

- Commands to be performed are listed in **bold** throughout the audit plan. Each command must be performed in order to ensure the integrity of the audit.
- Any command that is subjective in nature will be noted as such and documented.

Objectives:

The purpose of this audit is to certify the security design and application of a distributed intrusion detection system for a healthcare organization. Once the design has been certified the system will be scheduled to move into the production network.

More specifically, the audit will:

- Determine if the design meets the security standards of the organization.
- Determine if the design will meet the intrusion detection requirements of the organization.
- Determine whether the design will handle the current network bandwidth as well as future growth of the organization network.
- Determine if the Operating System chosen can be hardened to meet the security guidelines set forth by the organization.
- Determine whether the intrusion detection system will satisfy proposed HIPAA requirements.

Scope:

This audit focuses primarily on the security design and application of a system residing in a lab environment. As such, only limited testing can be performed in some areas of the audit.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
A. ADMINISTRATIVE SECTION *Objective(s):* To allow the auditor necessary time and/or resources to research the system being audited and develop an effective audit plan. *Source(s):* Ratliff, Richard L. Internal Auditing: Principles and Techniques. Altamonte Springs: The Institute of Internal Auditors, 1996. 187-193. NOTE: The source cited above refers to all of Section A unless otherwise noted.			
1. Prepare a Strategic Audit Plan for the area(s) or function(s) to be reviewed.	The expected result is the very plan you are currently reading.	O	Deficiencies may be overlooked if the audit plan is poorly designed.
2. Prepare a Detail Audit Budget for the area(s) or function(s) to be reviewed.	The expected result is an audit budget that takes into account the time, labor and resources needed to conduct the full assessment.	S	Time needed to conduct a proper assessment may exceed budgeted money. In turn, this could result in an audit that is hastily prepared and/or conducted.
3. Prepare Statement of Scope and Methods memorandum for the area(s) or function(s) to be reviewed. Address the memorandum to the appropriate level of management. Request copies or access to information necessary to begin work on the review. This information includes but is not limited to: • Policies and procedures and system	The expected result is to foster effective communication between the auditor and the entity being audited. Generally, an auditor will make personal contact and then send a formal memorandum requesting various pieces of information before the audit begins.	O	This is considered a common courtesy and the only real risk involved with not including this step is a possible delay in getting the information needed to conduct the audit.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
user manual. • Reports concerning the activity, i.e. special projects, studies, other audit reports including prior years' audit reports. • Organizational charts from CEO down to area(s) being reviewed • Job descriptions • Flow charts • Any types of network and/or system documentation			
4. Document Opening Conference: • Notification Memo • Meeting Agenda • Management's Comments / Meeting Notes • Exit Conference	The opening conference will outline the audit plan with management and will be the starting point to coordinate time and resources associated with the audit. This step also allows for appropriate response from management before a formal exit conference is scheduled.	O	It is imperative to present a professional appearance at the opening conference. The opening conference is the not the place to not be prepared. Image goes a long way in the minds of management and the area being audited.
5. Comparison of Budgeted Hours to Actual Hours	This step is needed to ensure that initial estimates were correct and, if required, change the estimate for future audits being conducted on the same or similar systems.	S	This step is needed to ensure accurate estimates are made when developing audit plans for the fiscal year. The difference between 50 and 100 hours is significant to an audit team trying to allocate time and resources to varying audit plans.
6. Prepare audit issues. Discuss audit issues and recommendations with appropriate	Chances are that some issues or deficiencies will be noted during the	O	Integrity of the audit is extremely important. If an

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
management personnel during course of audit as appropriate and at Exit Conference.	course of the audit. Changes can be made during the actual audit but they must still be reported to maintain the integrity of the audit plan. This step is usually a joint effort between the auditor and the entity being audited to develop a corrective course of action to remedy any issues that may arise.		administrator makes on the spot changes during the course of an audit it still needs to be entered into the final report.
B. ORGANIZATIONAL STRUCTURE / SYSTEM OVERVIEW, DOCUMENTATION, & TRAINING *Objective(s):* Determine the reporting structure of the organization being audited. Determine if system documentation exists and if policies and procedures defining the usage or rational for the system being deployed are present. *Source(s):* Ratliff, Richard L. Internal Auditing: Principles and Techniques. Altamonte Springs: The Institute of Internal Auditors, 1996. 195. NOTE: The source referenced above applies to all of Section B.			
1. Determine the reporting structure from the area(s) to be reviewed up to the CEO. • Initial memo should request a copy of the organization chart. • If a formal chart is not available then interview the entity being audited to determine the reporting structure.	A copy of the organization chart should be available. This will be helpful in determining the reporting structure of the organization and what areas of management need to be included in any meetings and/or published reports.	S	Knowing the reporting level of the organization is beneficial. It will allow the auditor to address reports accordingly but most importantly, it will establish boundaries and give the auditor a good idea what parts

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
			of the system/network are owned by the organization being assessed and what parts are owned by a third-party vendor. An example would be off-site DNS hosting. You would not want to launch commands against a DNS server that is not owned by the organization under review.
2. Determine if policies and/or procedures exist that define the usage and rational for the system(s) being audited.	Ideally, written policies and procedures will be available for review. It is possible that informal policies exist and if that is the case then a subjective interview will be necessary.	S	It is virtually impossible to conduct a thorough audit without a baseline for comparison. A policy is needed to determine what exactly needs to be audited and how the tests should be written and conducted.
3. Determine if documentation exists for the system(s) being reviewed. This would include any system documentation, scripts, and network diagrams. • Initial memo should request any available documentation. • If documentation does not exist then an interview of the organization being audited will be necessary	Copies of all system documentation, diagrams, etc should be available for review. This is essential in understanding the system, why it is being used and if it is being properly maintained and administered.	S	In addition to policies, it is extremely helpful to have access to any and all system documentation. Without formal documentation, it is nearly impossible to conduct an exhaustive audit in a timely manner.
C. RED HAT LINUX 7.2 OS			
1. Obtain general system information for the intrusion detection probe and the management console.	The uname –a command will show all system information to include OS, hostname, kernel version, date/time	O	You want to ensure you are assessing the right machine. Entering commands or

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
• Verify hostname, kernel version and system time is correct and/or updated to the most recent version. ○ **uname -a** • Obtain IP address information. ○ **ifconfig -a** *Source(s):* Personal experience	and hardware class. I have found it is always a good idea to verify that I am looking at the right machine. Output should reveal: IDS Probe: • hostname = probe1 • kernel = 2.4.7-10 Mgmt Console: • hostname = mgmt1 • kernel = 2.4.7-10 The ifconfig –a command will reveal the IP address information for each interface configured for the machine(s). IDS Probe: • eth0 = 192.168.1.103 • eth1 = 0.0.0.0 (stealth mode) Management Console: • eth0 = 192.168.1.102		performing certain actions could cause an availability problem. This can be troublesome in a production environment so it is always a wise idea to do a quick check to verify you are in the right location.
2. Verify latest Operating System patches have been installed. • Compare output with the latest packages available from the Red Hat errata pages. ○ **rpm –qa > package.txt** *Source(s):* Laude, Mary. "Auditing Red Hat Linux 7.0" 23 July 2001. URL: http://www.giac.org/practical/Mary_Laude_GSNA.zip (22 May 2002).	The rpm –qa command will result in a long list of all the installed packages for the system. This can be a tedious job to cross-reference the installed packages against the Red Hat errata page so I would recommend using the up2date feature if it is installed on the machine. If using the up2date feature then the	S	New exploits come out on a daily basis so it is imperative to ensure the system(s) being assessed are using the most up to date packages available. Vendors typically do a great job in releasing updated packages when a potential vulnerability exists but it is really up to the end-user to ensure that

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
• If the machine has been registered with the Red Hat Network, run the command: ○ **up2date -l** *Source(s):* Personal experience	listing should only show packages that were skipped because of their configuration. You will typically see the kernel and kernel-headers listed.		patches are installed in a timely manner.
3. Verify xinetd services are disabled. • Examine the /etc/xinetd.conf file for any services that are not needed. The file may contain an include statement for /etc/xinetd.d/ and the contents of that directory should also be examined. ○ **cat /etc/xinetd.conf** (-or-) ○ **more /etc/xinetd.conf** (-and-) ○ **ls –l /etc/xinetd.d** • Another option would be to ensure that xinetd is totally disabled. ○ **chkconfig –-list xinetd** *Source(s):* "CIS Level-1 Benchmark and Scoring Tool for Linux". URL: http://www.cisecurity.org/bench_linux.html (15 June 2002).	The contents of the /etc/xinetd.conf file should not contain any services. The /etc/xinetd.d/ directory should be empty. If security policy dictates that xinetd should be disabled completely then the result of the chkconfig command should reveal the daemon is turned OFF in all run levels.	O	Xinetd is typically used to initialize services such as telnet, ftp, and rpc services, which are known to be vulnerable to exploit.
4. Verify boot services that are not needed for the operation of the IDS are disabled. • Show daemons/services that are set to run at system boot. ○ **chkconfig --list |grep :on** *Source(s):* "CIS Level-1 Benchmark and Scoring Tool for Linux". URL:	The following daemons/services SHOULD NOT be listed as running: apmd, autofs, gpm, innd, IrDA, isdn, kdcrotate, lpd, lvs, mars-nwe, named, netfs, nfs, nfslock, oki4daemon, portmap, routed, rstatd, rusersd, rwalld, rwhod, sendmail, smb, snmpd, webmin, ypbind, ypserv,	O	Disabling services that are not needed greatly decreases the potential of a vulnerability being compromised by an attacker.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
http://www.cisecurity.org/bench_linux.html (15 June 2002).	yppasswdd		
5. Verify only the needed daemons are active and listening on their respective port(s). • Use the netstat command to see what daemons are running and on which ports. ○ **netstat –an** (-or-) ○ **netstat –at** • Use the lsof command to see what active network sockets are being used. ○ **lsof –i +M** *Source(s):* Laude, Mary. "Auditing Red Hat Linux 7.0" 23 July 2001. URL: http://www.giac.org/practical/Mary_Laude_GSNA.zip (22 May 2002).	The daemons/services listed below should be the only ones running. • For the IDS probe: • SSH (sshd) • Established SSH Tunnel to the management console • MySQL (localhost only) • For the management console: • SSH (sshd) • Established SSH tunnel to the IDS probe(s) • HTTP (httpd) • MySQL (mysqld)	O	Anything not explicitly needed for the operation of the probe(s) and management console is just a potential target. New vulnerabilities come out on a daily basis and it is wise to ensure any services not used are disabled.
6. Verify only needed ports are open on probe(s) and management console. • Use nmap to scan both the probe and management console. ○ **nmap –sS –p 1-65000 –v 192.168.1.[102-103]** *Source(s):* "Nmap Network Security Scanner Man Page". URL: http://www.nmap.org/nmap/nmap_manpage.html (22 May 2002).	Output from the nmap scan should reveal the following: 192.168.1.102: • port 22 (SSH) open • port 80 (HTTP) open 192.168.1.103: • port 22 (SSH) open	O	Again, the goal is to minimize exposure. If a port is not needed for the successful operation of the IDS configuration then it should not be open.
7. Verify all logging functions. Pay particular attention to login attempts, errors and warnings.	The /etc/syslog.conf file should contain entries for auth, authpriv, error and warning conditions.	O	The success of an intrusion detection system relies on the quality of the logging.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
• Examine the /etc/syslog.conf file to verify what is being logged. ○ **cat /etc/syslog.conf** (-or-) ○ **more /etc/syslog.conf** • Verify actionable items are being written to the appropriate logfile. ○ **ls –l /var/log** *Source(s):* Laude, Mary. "Auditing Red Hat Linux 7.0" 23 July 2001. URL: http://www.giac.org/practical/Mary_Laude_GSNA.zip (22 May 2002).	Any action that should be logged is written to the appropriate file in /var/log directory.		Therefore, logging should be configured to be as complete as possible in order to not miss any potential attacks.
8. Determine if log files are being reviewed. How often? By whom? *Source(s):* Personal Experience	Ideally, an automated process such as logcheck or swatch should review log files daily. If an automated method is not used then they should be reviewed by the system administrator on a daily basis.	S	Log files are of absolutely no use if they are not reviewed on a regular (daily) basis. This is often the only way to tell if an attacker has attempted to access the system.
9. Verify that logs are being rotated on a regular basis. • Examine /etc/syslog.conf to determine log rotation schedule. ○ **cat /etc/logrotate.conf** (-or-) ○ **more /etc/logrotate.conf** • Verify that logrotate is included in /etc/cron.daily directory. ○ **ls –l /etc/cron.daily** *Source(s):* "CIS Level-1 Benchmark and Scoring Tool for Linux". URL: http://www.cisecurity.org/bench_linux.html	Based on the log rotation policy for the organization being audited, the following entries must be present in /etc/syslog.conf: • rotate log files weekly • logs are kept 4 weeks • create new log after rotating old logs • wtmp files are rotated monthly and kept for 2 months	O	Log files that get too big are difficult to analyze and review. Log files should be kept for a minimum of 4 weeks in case further review is needed or even correlation of similar events needs to be conducted.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
(15 June 2002).			
10. Verify that there are no user accounts with empty password fields. • Use the awk command to verify the accounts do not have empty password fields. ○ **awk –F: '($2 == "") {print $1}' /etc/shadow** *Source(s):* "CIS Level-1 Benchmark and Scoring Tool for Linux". URL: http://www.cisecurity.org/bench_linux.html (15 June 2002).	This should result in no output after the command has been entered.	O	A user account with an empty password field is basically a wide open door. Many attacks look for default user accounts with no password.
11. Verify that no UID 0 accounts exist other than root. • Use the awk command to verify. ○ **awk -F: '($3 == 0) {print $1}' /etc/passwd** *Source(s):* "CIS Level-1 Benchmark and Scoring Tool for Linux". URL: http://www.cisecurity.org/bench_linux.html (15 June 2002).	The only account that should be listed is root.	O	Any account listed as UID 0 is considered a 'Super User' account. There is no need to have more than one account of this nature on a system. Accounts with UID 0 should be limited to only 'root' and only used when absolutely necessary.
12. Verify minimum password length and maximum password age is being enforced on the system(s). • Examine the contents of /etc/login.defs and check entries for PASS_MIN_LEN and PASS_MAX_DAYS. ○ **cat /etc/login.defs** (-or-) ○ **more /etc/login.defs**	Based on the organization security policy, the following parameters must be defined: • PASS_MAX_DAYS 90 • PASS_MIN_LEN 8 The test should not reveal the ability to create a new user with a password length less than 8 characters.	O	Passwords are easily cracked or guessed. Therefore, a good security policy should require passwords be a minimum of 8 characters and changed every 90 days. Passwords are often deployed as the first and last line of defense so it is critical to the

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
• Attempt to create a user with a password length shorter than the minimum length allowed by the security policy. • Time permitting, use a password-cracking program (e.g. John the Ripper) to test the strength of the passwords used on the system(s). *Source(s):* "login.defs". URL: http://docs.csoft.net/cgi-bin/man.cgi?section=5&topic=login.defs (23 May 2002).			security of a system to have a strong password policy.
13. Verify only the needed user id's are present on each system(s). • Examine the contents of the /etc/passwd file for any unnecessary user id's. ○ **cat /etc/passwd** (-or-) ○ **more /etc/passwd** *Source(s):* "CIS Level-1 Benchmark and Scoring Tool for Linux". URL: http://www.cisecurity.org/bench_linux.html (15 June 2002).	Probe – snort and root Console – snort, mysql, apache, root	O	The goal here is to minimize exposure by reducing the amount of user id's that can be used for a particular system. There are many default user id's that have easy to guess passwords. Removing any user id's that are not needed for proper performance is necessary to reduce the risk of compromise.
14. Verify the system time/date for each system(s) being audited is correct. • Check the current system(s) time by entering the following at the command prompt. ○ **date** *Source(s):* Andrews, James. "Time for Linux". URL:	Each probe should be consistent with the time displayed on the IDS management console. Ideally, the organization being audited should be using a time synchronization server (e.g. NTP) to keep their entire network time-synced. According to the security policy, this organization utilizes a public time	O	Time synchronization is important when it comes to event correlation and incident analysis. If the IDS logs are not in sync with a server that was compromised then it becomes very difficult to prove how an attack happened if pursuing criminal prosecution.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
http://www.linuxplanet.com/linuxplanet/tutorials/215/1/ (23 May 2002). • Compare the system time with the nist.gov time server. ○ **rdate –p time-a.nist.gov** *Source(s):* "rdate manual page". URL: http://linuxcommand.org/man_pages/rdate1.html (23 May 2002).	server to keep synced. The time displayed on the system(s) console should match that of the public time server.		
D. OPENSSH CONFIGURATION FOR IDS PROBE AND MANAGEMENT CONSOLE *Objective(s):* To verify the use and secure configuration of OpenSSH as a means for secure communication between the IDS probe(s) and the management console.			
1. Verify that OpenSSH is being used and that it is the latest version. • Check to see if SSH is running. ○ **netstat –at |grep ssh** (-and-) ○ **netstat –an |grep :ssh** • Verify latest version of OpenSSH is installed. ○ **ssh –V** • Compare version number with the current listing at the OpenSSH web site (http://www.openssh.com/). *Source(s):* Personal experience	The netstat commands should indicate that ssh is running and listening on default port 22. The command should also reveal an established ssh connection between the IDS probe and the management console. The version of OpenSSH being used must be OpenSSH_3.4p1 or later.	O	As stated earlier, vulnerabilities come out on a daily basis and it is imperative to always run the latest version to mitigate the effects of "vulnerability dujour". Note: Shortly after creating this checklist, a serious vulnerability in OpenSSH was released by ISS and OpenSSH regarding a potential flaw that could allow an attacker to escalate privilege to root.
2. Verify that /etc/ssh/ssh_config is properly	The /etc/ssh/ssh_config file must	O	Proper configuration is

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
configured. • Examine the contents of the /etc/ssh/ssh_config file. o **cat /etc/ssh/ssh_config** (-or-) o **more /etc/ssh/ssh_config** *Source(s):* Mourani, Gerhard. Securing and Optimizing Linux: The Ultimate Solution. Montreal: Open Network Architecture, Inc, 2001. 290-292	contain the following information: Host * ForwardAgent no ForwardX11 no RhostsAuthentication no RhostsRSAAuthentication no RSAAuthentication yes PasswordAuthentication no FallBackToRsh no UseRsh no BatchMode no CheckHostIP yes StrictHostKeyChecking yes IdentityFile ~/.ssh/identity IdentityFile ~/.ssh/id_dsa IdentityFile ~/.ssh/id_rsa Port 22 Protocol 2,1 Cipher blowfish EscapeChar ~		essential to the security of a system. Because SSH is the method being used to remotely administer the server and as the secure channel of communication between the IDS probe(s) and management console, it is important to ensure the application is configured properly.
3. Verify that /etc/ssh/sshd_config is properly configured. • Examine the contents of the /etc/ssh/sshd_config file. o **cat /etc/ssh/sshd_config** (-or-) o **more /etc/ssh/sshd_config** *Source(s):* Mourani, Gerhard. Securing and Optimizing Linux: The Ultimate Solution. Montreal: Open Network Architecture, Inc, 2001. 293-295.	The /etc/ssh/sshd_config file must contain the following information: Port 22 ListenAddress 0.0.0.0 HostKey /etc/ssh/ssh_host_key HostKey /etc/ssh/ssh_host_dsa_key HostKey /etc/ssh/ssh_host_rsa_key ServerKeyBits 768 LoginGraceTime 60 KeyRegenerationInterval 3600 PermitRootLogin no IgnoreRhosts yes IgnoreUserKnownHosts yes	O	Proper configuration is essential to the security of a system. Because SSH is the method being used to remotely administer the server and as the secure channel of communication between the IDS probe(s) and management console, it is important to ensure the application is configured properly.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
	StrictModes yes X11Forwarding no PrintMotd yes KeepAlive yes SyslogFacility AUTH LogLevel INFO RhostsAuthentication no RhostsRSAAuthentication no RSAAuthentication yes PasswordAuthentication yes PermitEmptyPasswords no AllowUsers snort Subsystem sftp /usr/libexec/openssh/sftp-server		
4. Verify that OpenSSH PAM password authentication support is being used. • Examine the contents of the /etc/pam.d/sshd file. ○ **cat /etc/pam.d/sshd** (-or-) ○ **more /etc/pam.d/sshd** *Source(s):* Mourani, Gerhard. Securing and Optimizing Linux: The Ultimate Solution. Montreal: Open Network Architecture, Inc, 2001. 295.	The /etc/pam.d/sshd file must contain the following: auth required /lib/security/pam_stack.so service=system-auth auth required /lib/security/pam_nologin.so account required /lib/security/pam_stack.so service=system-auth account required /lib/security/pam_access.so account required /lib/security/pam_time.so password required /lib/security/pam_stack.so service=system-auth	O	PAM (Pluggable Authentication Module) adds an additional layer of security by enabling the separation of authentication schemes from the applications themselves. Utilizing PAM with SSH adds to the overall security of the application. As stated before, because SSH is the sole means of communicating with the probe(s) or management console remotely, additional security is needed.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
	session required /lib/security/pam_stack.so service=system-auth session required /lib/security/pam_limits.so session optional /lib/security/pam_console.so		
E. SNORT CONFIGURATION FOR IDS PROBE(S) *Objective(s):* To verify the installed configuration of Snort is current and reflects the IDS policy established by the organization being audited.			
1. Verify Snort is running and that it is automatically started when the probe(s) is powered on or rebooted. • Check to see if Snort is running. ○ **ps ax |grep snort** (-or-) ○ **service snortd status** • Examine contents of /etc/rc.d/init.d/ to verify the Snort initialization script is present. ○ **ls –l /etc/rc.d/init.d/** • Use chkconfig to verify the runtime levels. ○ **chkconfig –list |grep snortd** *Source(s):* Personal Experience	Snort should be listed as running on the IDS probe(s). Examining the associated init.d script should reveal that the service is started automatically whenever the system is booted. Using the 'chkconfig' command will reveal which runtime levels snort is started and available on when the system boots.	O	If Snort is not configured to run whenever the system is booted then there is a risk in missing intrusion attempts to the system being monitored. It is not a wise idea to manually start Snort because there is a good chance that a system could reboot after normal business hours without the administrator being aware of the problem.
2. Verify the latest 'stable' release of Snort is installed.	The latest version of Snort is 1.8.6. The command should reveal the IDS probe(s) running the most current	O	Because Snort is an open source project, it can be prone to significant changes or

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
• Check the version being used and compare it against the current version listed at http://www.snort.org/dl/ and/or http://www.snort.org/binaries/RPMS/ ○ **snort –V** *Source(s):* Roesch, Marty. "Snort Users Manual". URL: http://www.snort.org/docs/writing_rules/ (23 May 2002).	version.		rewrites. As attackers devise ways to evade intrusion detection systems, Snort will continue to be updated. Running an older version of Snort could cause an intrusion attempt to be missed.
3. Determine if a stealth interface is being used. • Check for the presence of a second network interface card for the purpose of "listening" only. • **ifconfig –a** • Check the /etc/rc.d/init.d/snortd script to determine which interface is used by Snort. • **cat /etc/rc.d/init.d/snortd** (-or-) • **more /etc/rc.d/init.d/snortd**	The 'ifconfig' command should reveal the presence of three interfaces: • eth0 = IP address of the probe • eth1 = stealth interface (no ip address assigned) • lo = loopback interface (127.0.0.1)	O	A stealth interface should be used for the "listening" interface. A stealth interface does not have an IP address assigned to it, which makes it virtually impossible to detect on the network. The idea is to hide the presence of an intrusion detection probe from a would-be attacker. It should also be noted that in a switched environment where a network tap is used, only a stealth interface would work.
4. Examine the 'network variables' section of the /etc/snort/snort.conf file to determine if the IDS probe is properly configured for the system(s) and/or network(s) being protected. ○ **cat /etc/snort/snort.conf** (-or-) ○ **more /etc/snort/snort.conf** *Source(s):* Roesch, Marty. "Snort Users	Based on the IDS policy developed by the organization and the network diagrams provided, the following settings should be present in the /etc/snort/snort.conf file. • HOME_NET 192.168.1.0/24 • EXTERNAL_NET any	S/O	An IDS probe that is not properly configured runs the risk of missing a potential intrusion or attack. Particular attention needs to be paid to the HOME_NET variable to ensure the right network is being monitored.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
Manual". URL: http://www.snort.org/docs/writing_rules/ (23 May 2002).	• SMTP 192.168.1.101 • HTTP_SERVERS 192.168.1.102 • HTTP_PORTS 80 • SQL_SERVERS 192.168.1.102 • DNS_SERVERS 192.168.1.1 NOTE: Snort allows for extreme flexibility when it comes to configuration. It is possible that the settings reflected above will not be present in the snort.conf file. In that case, ask the administrator to explain their reason for the settings that are present. The goal is to ensure that the settings accurately reflect the network being protected/monitored.		
5. Examine the 'preprocessors' section of the /etc/snort/snort.conf file to determine if the IDS probe is properly configured for the system(s) and/or network(s) being protected. ○ **cat /etc/snort/snort.conf** (-or-) ○ **more /etc/snort/snort.conf** *Source(s):* Roesch, Marty. "Snort Users Manual". URL: http://www.snort.org/docs/writing_rules/ (23 May 2002). *Source(s):* Poppi, Sandro. "Snort-Setup for Statistics HOWTO". 23 February 2002.	Based on security policy, network diagrams and the open source documentation used to build the intrusion detection system, the following 'preprocessor' variables should be present in the snort.conf file: • preprocessor frag2 • preprocessor stream4: detect_scans detect_state_problems • preprocessor stream4_reassemble: ports all • preprocessor unidecode: 80	S/O	An IDS probe that is not properly configured runs the risk of missing a potential intrusion or attack. In this instance, we want to probe to look detect as many possible intrusion attempts as possible while making sure not to impact performance. There is a risk in looking for too many things which could result in packets being dropped.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
URL: http://www.tldp.org/HOWTO/Snort-Statistics-HOWTO/configuration.html#SNORT-CONFIG (31 May 2002).	8080 • preprocessor rpc_decode: 111 • preprocessor bo: -nobrute • preprocessor telnet_decode • preprocessor portscan: 0.0.0.0/0 6 3 /var/log/snort/portscan.log • preprocessor portscan-ignorehosts: $DNS_SERVERS NOTE: Snort allows for extreme flexibility when it comes to configuration. It is possible that the settings reflected above will not be present in the snort.conf file. In that case, ask the administrator to explain their reason for the settings that are present. The goal is to ensure that the settings accurately reflect the network being protected/monitored		
6. Examine the 'output plugins' section of the /etc/snort/snort.conf file to determine if the IDS probe is properly configured for the system(s) and/or network(s) being protected. ○ **cat /etc/snort/snort.conf** (-or-) ○ **more /etc/snort/snort.conf** *Source(s):* Roesch, Marty. "Snort Users Manual". URL: http://www.snort.org/docs/writing_rules/ (23 May 2002).	Based on security policy, network diagrams and the open source documentation used to build the intrusion detection system, the following 'output plugins' variables should be present in the snort.conf file: • output alert_syslog: LOG_AUTH LOG_ALERT LOG_PID • output database: alert, mysql, user=snort password=******** dbname=snort host=localhost	S/O	Snort needs to know where to log the data it collects. Because we are running a central management console, it is important to ensure that all alerts generated by the IDS probe(s) make their way to the database on the management console. If this section is not properly configured then there is a risk that potential intrusions will go undetected.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
Source(s): Poppi, Sandro. "Snort-Setup for Statistics HOWTO". 23 February 2002. URL: http://www.tldp.org/HOWTO/Snort-Statistics-HOWTO/configuration.html#SNORT-CONFIG (31 May 2002).	sensor_name=probe1		
7. Examine the 'customized ruleset' section of the /etc/snort/snort.conf file to determine if the IDS probe is properly configured for the system(s) and/or network(s) being protected. ○ **cat /etc/snort/snort.conf** (-or-) ○ **more /etc/snort/snort.conf** *Source(s):* Roesch, Marty. "Snort Users Manual". URL: http://www.snort.org/docs/writing_rules/ (23 May 2002). *Source(s):* Poppi, Sandro. "Snort-Setup for Statistics HOWTO". 23 February 2002. URL: http://www.tldp.org/HOWTO/Snort-Statistics-HOWTO/configuration.html#SNORT-CONFIG (31 May 2002).	Because the ruleset being used will be determined by the network(s) being monitored, this is a pretty subjective test. Check to see what rules are being used for the particular probe and verify with the administrator that they are meeting the needs of the organization. • Are all servers being monitored on the network segment protected by the ruleset? • Does the administrator understand the choices made in what rules to include?	S	If a particular ruleset is not enabled then there is the risk of an intrusion going undetected. An example would be an IIS server on the network being monitored that is not patched and vulnerable to Code Red. If the IDS probe is not configured to look for IIS related attacks, the intrusion would go unnoticed and the IIS server could become infected. It is imperative to use all the rulesets necessary to guarantee complete coverage of the network being monitored.
8. Verify snort rules are up to date and what methods are being used to keep the snort rules current. • Compare the version number and date of the snort.conf file being used with the snort.conf file available in the latest rulesets	The snort.conf version number and date should match that of the latest ruleset available at: http://www.snort.org/dl/signatures/ Automated scripts are available to aid in keeping the snort rules current.	S/O	New signatures are released on a regular basis as new vulnerabilities are announced or uncovered. An example of this is the recent OpenSSH vulnerability. Without updated signatures, the IDS probe

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
available from the Snort web site. • **cat /etc/snort/snort.conf | grep Id** (-or-) • **more /etc/snort/snort.conf| grep Id** • Interview the Administrator to determine what methods are being used to keep the snort rules current. *Source(s):* Personal Experience	At a minimum, the administrator should be checking the Snort web site on a regular basis and applying new rulesets as necessary.		would not be able to alert to any attacker searching for the OpenSSH vulnerability.
9. Verify that sendmail (needed for real-time alerting) is configured for localhost traffic only and set to retrieve mail from the queue every 30 seconds. • Check running processes for sendmail configuration. ○ **ps ax |grep sendmail**	Output should indicate that sendmail is set to poll the queue every 30 seconds. The ps ax |grep sendmail command should reveal the following: /usr/sbin/sendmail -q30s	O	Sendmail is usually started as a daemon/service. In this instance, the probe is not acting as a mail gateway or relay so it does not need to be listening on an active port and can be configured for local outbound traffic only.
10. Determine if real-time alerting (SWATCH) is enabled and verify the configuration to ensure it is compliant with the IDS security policy. • Check to see if SWATCH is running. ○ **ps ax |grep swatch** • Verify /etc/swatch/swatchrc is configured properly. ○ **cat /etc/swatch/swatchrc** (-or-) ○ **more /etc/swatch/swatchrc** *Source(s):* Poppi, Sandro. "Snort-Setup for Statistics HOWTO". 23 February 2002. URL: http://www.tldp.org/HOWTO/Snort-	Output from the ps ax |grep swatch command should reveal that swatch is, indeed, running. The /etc/swatch/swatchrc file should contain: • 'watchfor' listing containing the snort variables being monitored. • Recipient (email, pager, etc) for any alert notifications. • 'throttle' variable to aid in cutting down the number of alerts received for a string of detections.	S/O	Without a real-time alerting function in place, intrusion attempts of a critical nature could go undetected. SWATCH can be configured to pop-up messages on the analyst's console, send email to a pager or even ring a bell. The 'throttle' variable is necessary to cut down on the potential deluge of messages for persistent intrusion attempts like the Code Red Worm.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
Statistics-HOWTO/configuration.html#SWATCH-CONFIG (31 May 2002).			
F. MYSQL – MANAGEMENT CONSOLE *Objective(s):* To ensure basic security settings for the MySQL database exist and meet the security needs for the organization.			
1. Verify MySQL root account is password protected. • Attempt to access the mysql database without using a password. ○ **mysql –u root mysql** (-and-) ○ **mysql –u root mysql –p** *Source(s):* "MySQL Manual: General Security Guidelines".URL: http://www.mysql.com/doc/G/e/General_security.html (15 June 2002).	The command should result in an error message being generated: ERROR 1045: Access denied for user: 'root@localhost' (Using password: NO) The "-p" command should prompt for a password.	O	A default installation of MySQL leaves the root account for MySQL unprotected. The root account for mysql has full access to all databases so it is imperative that a password is assigned. The mysql database contains a field called 'users' which contains account information and access rights for all mysql databases on the system.
2. Verify version of MySQL being used is current. • Access MySQL with the root account and check version number. ○ **mysql –V** (-or-) ○ **mysql –u root mysql –p** ○ **status;** *Source(s):* "MySQL: The Command-line Tool". URL:http://www.mysql.com/doc/m/y/mysql.html (15 June 2002).	At time of writing the most current version of MySQL is: 3.23.51. Issuing the '-V' command should reveal version 3.23.51 is being used on the management console.	O	Because MySQL is an open source product, it can be prone to constant tweaking and development by its author. Older versions of MySQL have been known to suffer from performance issues and the occasional security vulnerability.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
3. Verify only required databases for Snort usage are present on the management console. • Access mysql session with the mysql root account and check which databases are present on the management console. ○ **show databases;** *Source(s):* "MySQL: SHOW syntax". URL: http://www.mysql.com/doc/S/H/SHOW.html (15 June 2002).	The following databases should be the ONLY ones present on the management console: • mysql (system database) • snort (/etc/snort/snort.conf) • snort_log (/etc/snort/snort.conf)	O	Unless there is an absolute need for a particular database, it should not be available to would-be attackers. MySQL ships with default 'test' database that could allow an attacker to gain entry to the system if not properly configured. Databases that are not in use aren't always monitored for abuse so the prudent measure would be to delete any databases not used for the intrusion detection system.
4. Verify only required snort user accounts are contained in the mysql database. • Access mysql session with the mysql root account and check which users (mysql accounts) are present on the management console. ○ **select * from user;** *Source(s):* "MySQL: Selecting All Data". URL: http://www.mysql.com/doc/S/e/Selecting_all.html (15 June 2002).	The 'user' field of the mysql database will list accounts and their access rights. Only 'root' and 'snort' should be listed.	O	The goal is to minimize the risk of potential abuse. Database users should be restricted to only those required for the intrusion detection system.
5. Verify rights (grants) for user 'snort' are sufficient and do not exceed what is needed for normal operation. • Access mysql session with the mysql root account and show the rights for user 'snort@localhost'.	The 'snort@localhost' rights should be set to the following for snort and snort_log. • CREATE • INSERT	O	A database user account with more rights than needed for the successful operation of the intrusion detection system could be exploited by an attacker and escalates their

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
○ **show grants for snort@localhost;** *Source(s):* "MySQL: Show Grants". URL: http://www.mysql.com/doc/S/H/SHOW_GRANTS.html (15 June 2002). *Source(s):* Poppi, Sandro. "Snort-Setup for Statistics HOWTO". 23 February 2002. URL:http://www.tldp.org/HOWTO/Snort-Statistics-HOWTO/configuration.html#MYSQL-CONFIG. (31 May 2002).	• SELECT • DELETE • UPDATE NOTE: The 'grants' are also shown when entering a 'status' command but the formatting of the output can be difficult to read. The 'show grants' command is another option to verify access rights.		privilege. The snort user account only needs create, insert, select, delete and update for the snort and snort_log database only.
6. Confirm the snort database is logging information received from the intrusion detection probe(s). • Access mysql session with the snort account to test for proper logging. ○ **echo "SELECT count(*) FROM event" | mysql snort –u snort –p** *Source(s):* Danyliw, Roman. "ACID: Installation and Configuration". URL: http://www.andrew.cmu.edu/~rdanyliw/snort/acid_config.html (16 June 2002).	The SELECT statement used in the test should reveal the number of events currently stored in the database. This number should be greater than zero if the probe(s) are sending events to the management station.	O	Intrusion attempts will only be displayed on the ACID management console if they are being logged properly in the MySQL database. It is vital to the successful operation of the intrusion detection system for all data entries to be properly written to the MySQL database.
7. Verify MySQL data being sent from the intrusion detection probe(s) to the management console is encrypted. • Use 'tcpdump' on the management console to sniff traffic coming from the intrusion detection probe(s). ○ tcpdump –l –i eth0 –w test.drw src or dst port 3306 | strings • Use sniffer (e.g ethereal) from to capture	No data should be shown using the 'tcpdump' command. Analyzing the packet capture with Ethereal should NOT reveal any non-SSH traffic between the intrusion detection probe(s) and the management console.	O	All traffic between the IDS probe(s) and the management console should be encrypted to minimize the chance of an attacker locating the machines used for the intrusion detection system. There is also a chance an attacker could 'sniff' the communication between the probe/management console to

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
network traffic between the probe(s) and the management console. ○ Open the file that was captured and check to see if any non-SSH traffic is shown between the probe(s) and the management console. *Source(s):* "MySQL Manual: General Security Guidelines".URL: http://www.mysql.com/doc/G/e/General_security.html (15 June 2002). *Source(s):* Personal Experience			determine what vulnerabilities exist for a particular network segment without ever having to launch his own attacks/scans.
G. APACHE – MANAGEMENT CONSOLE			
Objective(s): To ensure basic security settings for the Apache webserver exist and meet the security needs for the organization.			
1. Verify the version of Apache being used is current. • Check the version of the httpd (apache) daemon being used by the management console. ○ **httpd –v** *Source(s):* "Apache HTTPD Server Project". URL: http://httpd.apache.org/ (16 June 2002).	The main page for the Apache website lists the most current version available. At the time the website was accessed the current version was 1.3.24.	S	Apache is open source and subject to constant modification by the development team. As performance issues and security vulnerabilities are discovered, patches and/or new versions are released. A recent example is the 'chunking' vulnerability that was released.
2. Verify the general configuration settings for the Apache web server. • Check /usr/local/apache/conf/httpd.conf for the general server settings. ○ **cat**	Check the configuration file for: • server type = standalone • user/group = nobody • port = 80 • server admin = aliased email	S	Because Apache is extremely flexible, the configuration file can be especially complex. A novice administrator could inadvertently open a security hole by entering incorrect

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
/usr/local/apache/conf/httpd.conf (-or-) ○ **more /usr/local/apache/conf/httpd.conf** *Source(s):* "Apache Directives". URL: http://httpd.apache.org/docs/mod/directives.html (16 June 2002).	account (e.g. webmaster@giac.org and not joeuser@giac.org) • server root = /usr/local/apache/ NOTE: The above settings are recommended settings and may require a subjective analysis with the administrator of the server. Apache is extremely flexible in terms of configuration possibilities. This step is mostly needed to ensure that due attention was given to the general server settings.		configuration parameters.
3. Verify security configuration settings for the Apache web server. • Check /usr/local/apache/conf/httpd.conf for the presence of general security settings. ○ **cat /usr/local/apache/conf/httpd.conf** (-or-) ○ **more /usr/local/apache/conf/httpd.conf** *Source(s):* "Apache Directives". URL: http://httpd.apache.org/docs/mod/directives.html (16 June 2002).	The following settings should be present in the server configuration file: • log level = notice • allow override = authconfig • Allow from = 192.168.1.100*** • Deny from = all * ** IP address listed is the only machine on the network granted access to the management console.	O	Apache can be configured to limit access from remote hosts. Because of the sensitive nature that will be served from the management console running ACID and Apache, it is important to limit access only to those hosts that need to view the data. As an additional layer of defense, a userid/password combination can also be configured as explained below.
4. Verify .htaccess file is configured for any secure directories (e.g. ACID directory) on the web server. *Source(s):* "Using User Authentication". URL: http://www.apacheweek.com/features/userauth (16 June 2002).	Contents of .htaccess must contain: • AuthType = basic • AuthName = "<descriptive text>" • AuthUserFile = path to user file • Require = valid-user	S/O	As explained above, the web server will be serving sensitive data to intrusion analysts. This data needs to be restricted to only those that need implicit access. The .htaccess file can

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
	NOTE: While the four elements must be included in the .htaccess file, there is some room for modification with AuthName, AuthUserFile and the Require setting. As long as the elements can be verified and work then the step can be considered compliant.		be used as an added layer (ACL) of defense.
5. Verify ACID is the latest version. • Check latest version number at the ACID website and compare it to the version seen running on the management console. ○ Access http://www.cert.org/kb/acid/ and scroll down until you find the latest version number. ○ Access the management console and note the version number listed at the bottom left of the main ACID page. *Source(s):* "Analysis Console for Intrusion Databases". URL: http://www.cert.org/kb/acid/ (16 June 2002).	At the time of this writing, the latest version of ACID was: **0.9.6b21**	O	Because ACID is still in the development phase, it is undergoing a lot of changes. It is a wise idea to keep up to date on the changes being made by CERT. As holes/vulnerabilities are discovered they are patched and available in the newest release.
6. Review settings in the acid_conf.php file to verify the database and email alert action settings. • Check the /usr/local/apache/htdocs/sec/acid/acid_conf.php file for the proper settings. ○ **cat /usr/local/apache/htdocs/sec/acid/**	The database settings should coincide with the settings defined in /etc/snort/snort.conf. The email alert action settings should reflect the following: • Email from: Descriptive ID for Intrusion Detection. (e.g. "IDS Alert <ids>")	S/O	With the deluge of email/pages that most administrators have to deal with, it is very important to ensure the message being sent contains descriptive text that accurately displays the importance of the alert being received. There is a possibility that a received message could

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
acid_conf.php (-or-) ○ **more /usr/local/apache/htdocs/sec/acid/acid_conf.php** *Source(s):* "Analysis Console for Intrusion Databases". URL: http://www.cert.org/kb/acid/ (16 June 2002).	• Subject: Descriptive text that adequately explains the nature of the message being sent. (e.g. "IDS Incident Report") • Message: Descriptive text outlining the nature of the message and relevant contact information for any questions. (e.g. "This message is being sent on behalf of the Incident Response/Analysis Center. Your immediate attention is required. Please contact the IRAC at: xxx.xxx.xxxx") • Email mode: Should be set to 0 to force alert information to be embedded in the text of the message and not sent as an attachment.		be ignored or overlooked by the administrator if the From, Subject and body of the message do not contain descriptive/valid data.
7. Verify the ACID management console can perform necessary tasks. • Check the 'Email Alert' feature. ○ Access the ACID console via a web browser. Select an alert and use the drop-down box to email an alert. • Check the 'Archive' feature. ○ Access the ACID console via a web browser. Select an alert and attempt to archive it (move and/or copy) to	All features checked should be operational. Email should be received by the address(s) entered in the data field. Alerts should be able to be archived and/or deleted. Once the action is selected the console will display a 'SUCCESS' or 'FAILURE'.	O	The functionality afforded by a central management console is rendered useless if the basic services like the ability to send email, archive, delete, etc are not working.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
the archive database. • Check the 'Delete' feature. ○ Access the ACID console via a web browser. Select an alert(s) and attempt to delete it from the database. *Source(s):* "Analysis Console for Intrusion Databases". URL: http://www.cert.org/kb/acid/ (16 June 2002).			
H. COMBINED PROBE(S) AND MANAGEMENT CONSOLE ASSESSMENT			
1. Perform a portscan and verify the scan is detected by the probe(s) and accurately displayed on the ACID management console. • Use Nmap to conduct a scan of the IP address segment being monitored by the probe(s). ○ **nmap –sS –p 1-65000 –v –O 192.168.1.0/24** *Source(s):* Personal Experience	The ACID management console should indicate a port scan from the address being using Nmap.	O	Since the IDS probe is configured to detect a portscan, we need to verify that even stealth scans such as the one listed are detected and logged. Portscans are often the precursor to a more damaging type of attack.
2. Perform a vulnerability assessment on the IDS probe(s) and the management console to ensure there are no vulnerabilities that could be exploited by a remote user. • Use Nessus to conduct a vulnerability assessment of the probe(s) and management console. As an added bonus	The test should not reveal any holes and info field and port field should only contain information regarding port(s) 22 and 80.	O	The goal of an intrusion detection system is to "DETECT' intrusions and not serve as a target for attacks. Therefore, it is necessary to perform a vulnerability assessment on the probe(s) and management console to

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
test, use the IDS evasion technique option within Nessus to see if the test can bypass the IDS probe. ○ Access a server running Nessus via the Nessus client to perform testing. Actual client/server configuration is beyond the scope of this document. *Source(s):* Deraison, Renaud. "Nessus Demonstration". The Nessus Project. URL: http://www.nessus.org/demo/index.html (23 May 2002).			ensure they can't be exploited by attacks.
3. Ensure only those hosts granted access to the management console have access. • Attempt to access the management console from a host that is not defined in the httpd.conf as having web access. 1. Use a web browser from an authorized host (192.168.1.100) and an unauthorized host (192.168.1.200) to access the web server. 2. Web browser coming from authorized host (192.168.1.100) will be prompted for UserID/Password when attempting to access the ACID console. *Source(s):* Personal Experience	Provided with the proper IP address and user credentials, the auditor should be granted access to the ACID management console. Any access from a host NOT specifically listed as having access should result in an error message being displayed in the web browser.	O	The risk for this step is explained in previous sections but the goal is to minimize the exposure to sensitive data.
4. Confirm alerts are being displayed on the management console and real-time alerting functions are operational.	Alerts should be displayed on the ACID management console and real-time notification should be operational.	O	Attackers commonly use the listed tools to perform scans/assessments of a particular network before

Audit Steps: Objective, Testing and Reference	Compliance/Expected Results	Type	Risk
• Use any combination of the following tools to 'attack' the network segment being monitored. Detailed description of how to use the tools is beyond the scope of this document and assumes auditor familiarity with the tools listed. ➢ Nmap (http://www.nmap.org/) ➢ SNScan (http://www.foundstone.com/) ➢ Nessus (http://www.nessus.org/) ➢ BO Pinger (http://www.foundstone.com/) ➢ Snot (http://www.sec33.com/sniph/) *Source(s):* Personal Experience and Snort Users Listserv			launching an attack. As such, the intrusion detection system must alert to the presence of these devices being used on the network being monitored. SNOT has the ability to playback Snort rules to trigger a variety of attacks. However, this only verifies the syntax of the signature and not necessarily a real attack.
5. Determine if packets are being dropped by the IDS probe(s). • Generate internal statistics for Snort and look for the presence of dropped packets. • **ps ax |grep snort** (note PID for Snort) • **kill –SIGUSR1 <pid for snort>** • **more /var/log/messages** (look for the line that indicates the number of packets dropped) *Source(s):* Poppi, Sandro. "Snort-Setup for Statistics HOWTO". 23 February 2002. http://www.tldp.org/HOWTO/Snort-Statistics-HOWTO/configuration.html#SNORT-	The snort internal statistics should reflect zero (0) dropped packets.	O	If a probe is dropping packets this is an indicator the processor/memory can't handle the network traffic it is receiving. It increases the risk of the IDS probe missing an attack and never getting displayed on the ACID management console.

AUDIT STEPS: OBJECTIVE, TESTING AND REFERENCE	COMPLIANCE/EXPECTED RESULTS	TYPE	RISK
CONFIG (31 May 2002).			
End Of Audit Steps			

Assignment 3 – Conduct the Audit

The ten items shown below are the areas I feel are most critical to the success of the audit. Some of the tests had to be conducted on both the IDS probe and the management console. Those items are noted as such and the results will be shown for each device. It should be noted that in order to demonstrate visual proof of the real-time alerting features, I asked the network engineer to provide me with an email account that would receive the exact messages sent to the pager(s) configured in the SWATCH program for text-based alerting. In other words, it would be difficult to provide screen shots of text from a pager so I included screenshots of the actual message as it appeared in email format.

Checklist Item C2: PASS

Objective: Verify latest Operating System patches have been installed.

The 'up2date' command was used because both the IDS probe and the management console have been registered with Red Hat. This command doesn't take nearly as long as a package query and the results are the same.

Management Console

```
[root@mgmt1 /]# up2date -l

Retrieving list of all available packages...
########################################

Removing installed packages from list of updates...
########################################

Removing packages marked to skip from list...
########################################

Getting headers for available packages...
########################################

Removing packages with files marked to skip from list...
########################################

Getting headers for skipped packages...
########################################
The following Packages were marked to be skipped by your configuration:

Name                          Version       Rel  Reason
-------------------------------------------------------------------------
kernel                          2.4.9         34   Pkg name/pattern
kernel-headers                2.4.9         34   Pkg name/pattern
```

[root@mgmt1 /]#

IDS Probe

[root@probe1 /]# up2date -l

Retrieving list of all available packages...
##

Removing installed packages from list of updates...
##

Removing packages marked to skip from list...
##

Getting headers for skipped packages...
##
The following Packages were marked to be skipped by your configuration:

Name	Version	Rel	Reason
kernel	2.4.9	34	Pkg name/pattern
kernel-headers	2.4.9	34	Pkg name/pattern

[root@probe1 /]#

Checklist Item C5: PASS

Objective: Verify only the needed daemons are active and listening on their respective port(s).

I used both the 'netstat' and 'lsof' command to get a solid representation of what daemons are running and tunnels established.

Management Console

```
[root@mgmt1 /]# netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:mysql                 *:*                     LISTEN
tcp        0      0 *:http                  *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 192.168.1.102:ssh       192.168.1.103:1025
      ESTABLISHED
[root@mgmt1 /]#

[root@mgmt1 /]# lsof -i +M
COMMAND   PID    USER   FD   TYPE DEVICE SIZE NODE NAME
```

```
sshd          916   root    3u      IPv4   1277          TCP *:ssh (LISTEN)
mysqld        1280  root    3u      IPv4   1597          TCP *:mysql (LISTEN)
sshd          1398  root    4u      IPv4   1709          TCP 192.168.1.102:ssh-
>192.168.1.103:1026 (ESTABLISHED)
httpd         1401  root    16u     IPv4   1723          TCP *:http (LISTEN)
httpd         1402  root    16u     IPv4   1723          TCP *:http (LISTEN)
httpd         1403  root    16u     IPv4   1723          TCP *:http (LISTEN)
httpd         1404  root    16u     IPv4   1723          TCP *:http (LISTEN)
[root@mgmt1 /]#
```

IDS Probe

```
[root@probe1 root]# netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q  Local Address          Foreign Address        State
tcp       0      0       localhost.localdo:mysql  *:*                    LISTEN
tcp       0      0       *:ssh                    *:*                     LISTEN
tcp       0      0       192.168.1.103:1025       192.168.1.102:ssh      ESTABLISHED
[root@probe1 root]#
```

```
[root@probe1 root]# lsof -i +M
COMMAND  PID   USER   FD   TYPE  DEVICE SIZE       NODE NAME
sshd             579   root   3u      IPv4   782              TCP *:ssh (LISTEN)
ssh           831   root   3u      IPv4   1011             TCP 192.168.1.103:1025-
>192.168.1.102:ssh (ESTABLISHED)
ssh           831   root   4u      IPv4   1012             TCP
localhost.localdomain:mysql (LISTEN)
[root@probe1 root]#
```

Checklist Item C6: PASS

Objective: Verify only needed ports are open on probe(s) and management console.

An Nmap scan was performed from a Linux workstation (192.168.1.101).

Management Console/IDS Probe

```
[root@mail /]# nmap -sS -p 1-65000 -v 192.168.1.[102-103]

Starting nmap V. 2.54BETA36 ( www.insecure.org/nmap/ )
Host  (192.168.1.102) appears to be up ... good.
Initiating SYN Stealth Scan against  (192.168.1.102)
Adding open port 22/tcp
Adding open port 3306/tcp
Adding open port 80/tcp
The SYN Stealth Scan took 2776 seconds to scan 65000 ports.
Interesting ports on  (192.168.1.102):
(The 64996 ports scanned but not shown below are in state: filtered)
Port       State       Service
```

```
22/tcp      open        ssh
80/tcp      open        http
3306/tcp    open        mysql

Host  (192.168.1.103) appears to be up ... good.
Initiating SYN Stealth Scan against  (192.168.1.103)
Adding open port 22/tcp
The SYN Stealth Scan took 12 seconds to scan 65000 ports.
Interesting ports on  (192.168.1.103):
(The 64999 ports scanned but not shown below are in state: closed)
Port        State       Service
22/tcp      open        ssh

Nmap run completed -- 2 IP addresses (2 hosts up) scanned in 2790 seconds
[root@mail /]#
```

Checklist Item C14: PASS

Objective: Verify the system time/date for each device being audited is correct.

Management Console

```
[root@mgmt1 root]# rdate -p time-a.nist.gov
[time-a.nist.gov]      Tue Jul  2 15:06:52 2002
[root@mgmt1 root]# date
Tue Jul  2 15:06:54 EDT 2002
[root@mgmt1 root]#
```

IDS Probe

```
[root@probe1 root]# rdate -p time-a.nist.gov
[time-a.nist.gov]      Tue Jul  2 15:10:41 2002
[root@probe1 root]# date
Tue Jul  2 15:10:42 EDT 2002
[root@probe1 root]#
```

Checklist Item D1: FAIL

Objective: Verify that OpenSSH is being used and that it is the latest version.

It should be noted that the network engineer that accompanied me during the audit knew this was going to be a deficiency. He stated the version of OpenSSH being used would be updated as the system transitioned from the lab network into production. The engineer disagreed that it should be included in the report but after I explained it **had** to be included to preserve the integrity of the audit, he agreed and we moved onto the next audit objective.

Management Console

```
[root@mgmt1 root]# ssh -V
OpenSSH_3.1p1, SSH protocols 1.5/2.0, OpenSSL 0x0090602f
```

[root@mgmt1 root]#

IDS Probe

[root@probe1 root]# ssh -V
OpenSSH_3.1p1, SSH protocols 1.5/2.0, OpenSSL 0x0090602f
[root@probe1 root]#

Checklist Item E2: PASS

Objective: Verify the latest 'stable' release of Snort is installed.

IDS Probe

[root@probe1 root]# snort -V

-*> Snort! <*-
Version 1.8.6 (Build 105)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
[root@probe1 root]#

Checklist Item E3: PASS

Objective: Determine if a stealth interface is being used.

IDS Probe

[root@probe1 root]# ifconfig -a
eth0 Link encap:Ethernet HWaddr 00:60:08:AD:C6:82
inet addr:192.168.1.103 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:70409 errors:0 dropped:0 overruns:0 frame:0
TX packets:66867 errors:0 dropped:0 overruns:0 carrier:0
collisions:506 txqueuelen:100
RX bytes:4989193 (4.7 Mb) TX bytes:4140816 (3.9 Mb)
Interrupt:10 Base address:0x6400

eth1 Link encap:Ethernet HWaddr 00:60:08:AD:C6:4B
UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
RX packets:405448 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:25824748 (24.6 Mb) TX bytes:0 (0.0 b)
Interrupt:9 Base address:0x6500

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:14 errors:0 dropped:0 overruns:0 frame:0
TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0

```
        RX bytes:1276 (1.2 Kb)  TX bytes:1276 (1.2 Kb)

[root@probe1 root]#
```

Because the Snort initialization script is so long, it is not included here as proof the step was performed. The script **did** indicate that eth1 was the interface being used to monitor the network segment. In lieu of this step not being included, I have shown the results of a 'ps ax' command, which clearly shows Snort running on the eth1 interface.

```
[root@probe1 root]# ps ax|grep snort
 1425 ?        S      0:02 /usr/sbin/snort -d -D -i eth1 -l -l /var/log/snort -O
 1433 pts/1    S      0:00 grep snort
[root@probe1 root]#
```

Checklist Item H3: PASS

Objective: Ensure only those hosts granted access to the management console have access.

A screen capture tool was used to capture the login process from 2 different workstations located at 192.168.1.100 and 192.168.1.104. The workstation at 192.168.1.100 was presented with an authentication screen to login to the management console while the workstation at 192.168.1.104 was presented with a 403 Forbidden error.

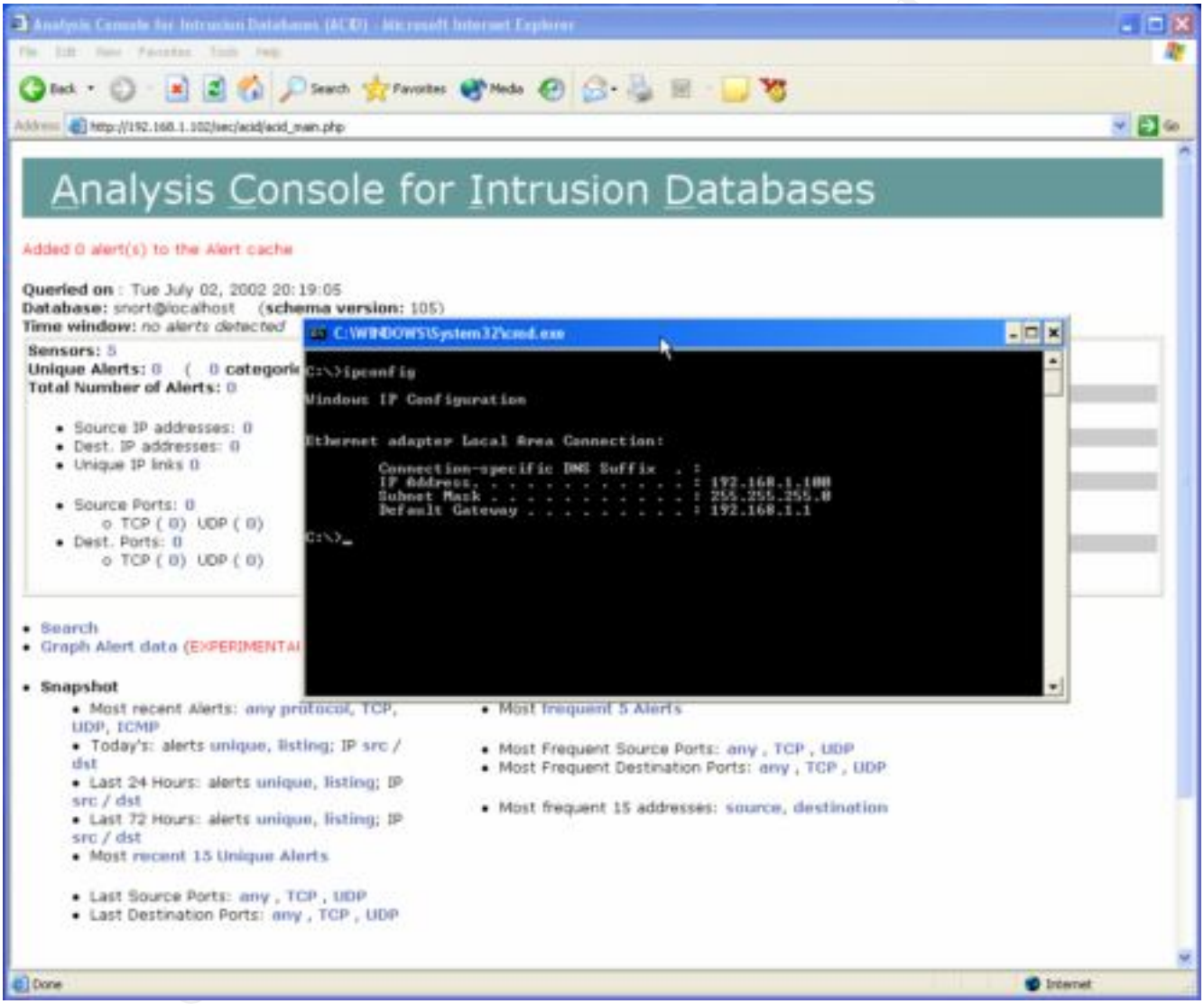
Accessing the Management Console from 192.168.1.100

When I attempted to access the management console from the workstation located at IP address 192.168.1.100, I was presented with a popup window asking for a User Name and Password.

Figure 2 - Windows Pop-Up Box

Once the User Name and Password was correctly entered, I was able to access the ACID management console. Figure 3 depicts the workstation IP address information and the main ACID console.

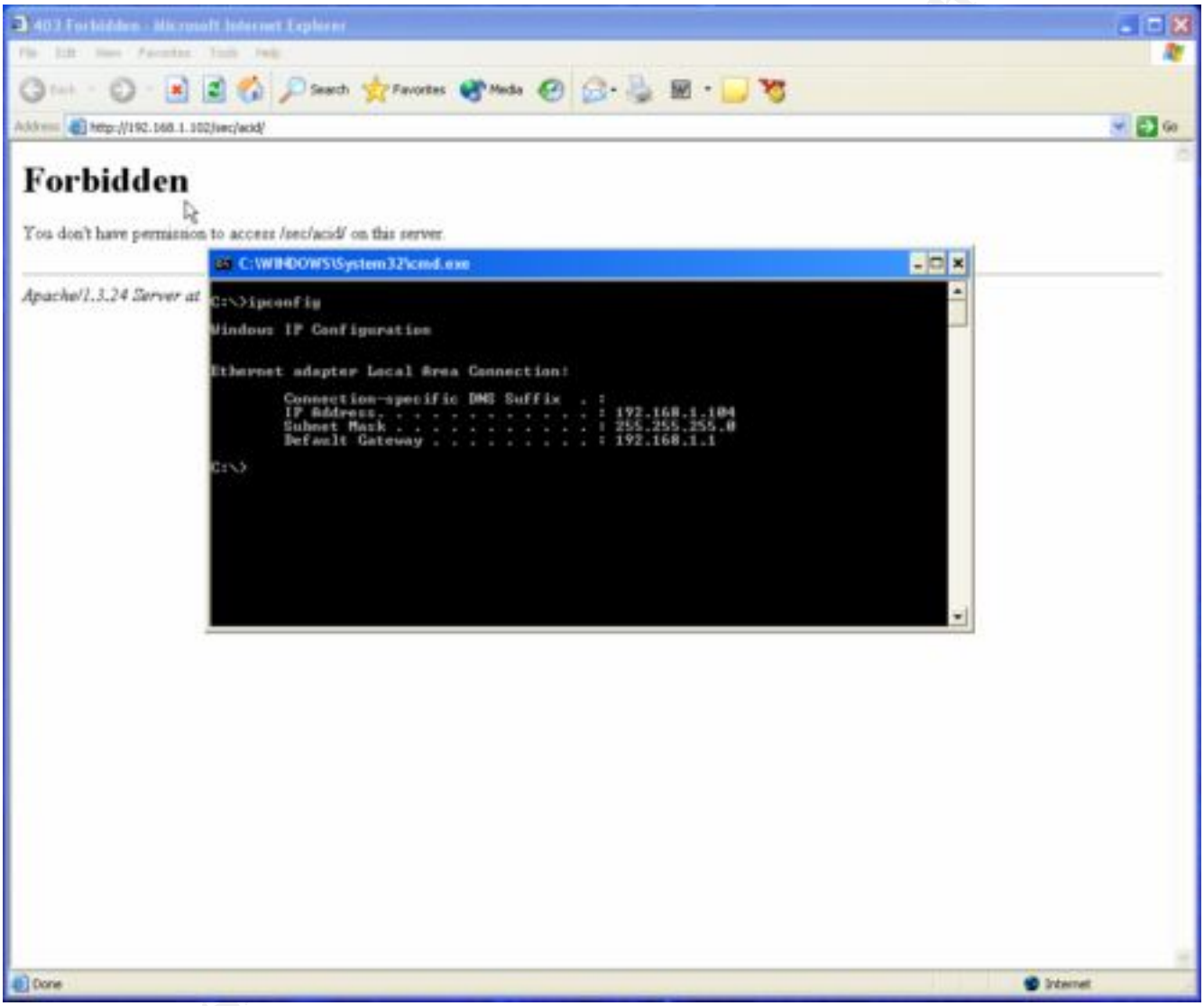
Figure 3 - Successful Login to ACID



Accessing the Management Console from 192.168.1.104

When I attempted to access the ACID management console from the workstation located at IP address 192.168.1.104, I was immediately presented with a 403 Forbidden error, indicating I was not allowed access from this host. Figure 4 illustrates the error message received.

Figure 4 – Failed Login to ACID



Checklist Item H4: FAIL

Objective: Confirm alerts are being displayed on the ACID management console and real-time alerting functions are operational.

I began this part of the testing by accessing the ACID management console and deleting all of the current alerts being displayed. I wanted to start with a clean slate to ensure I was looking at the most current data available after each series of tests. I selected a series of applications that look for various vulnerabilities. By using these

tools, I should trigger a series of alerts that can be verified via the ACID management console and real-time alerting via email.

SNScan 1.4 Results

SNScan is a Windows-based GUI scanner tailored specifically to finding open SNMP services on a given network. The use of this tool should trigger various SNMP alerts.

Figure 5 shows the SNScan GUI configured to perform a scan of the entire 192.168.1.0 subnet.

Figure 5 - SNScan Client Configuration

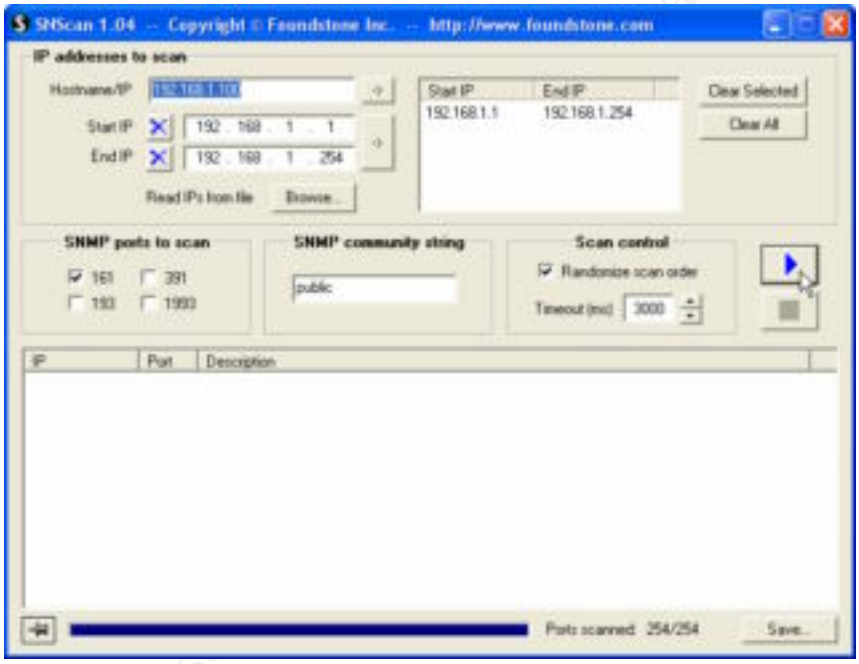


Figure 6 shows the results of the SNMP scan in the ACID management console. It was successful in picking up each targeted scan.

Figure 6 - SNScan Results in ACID

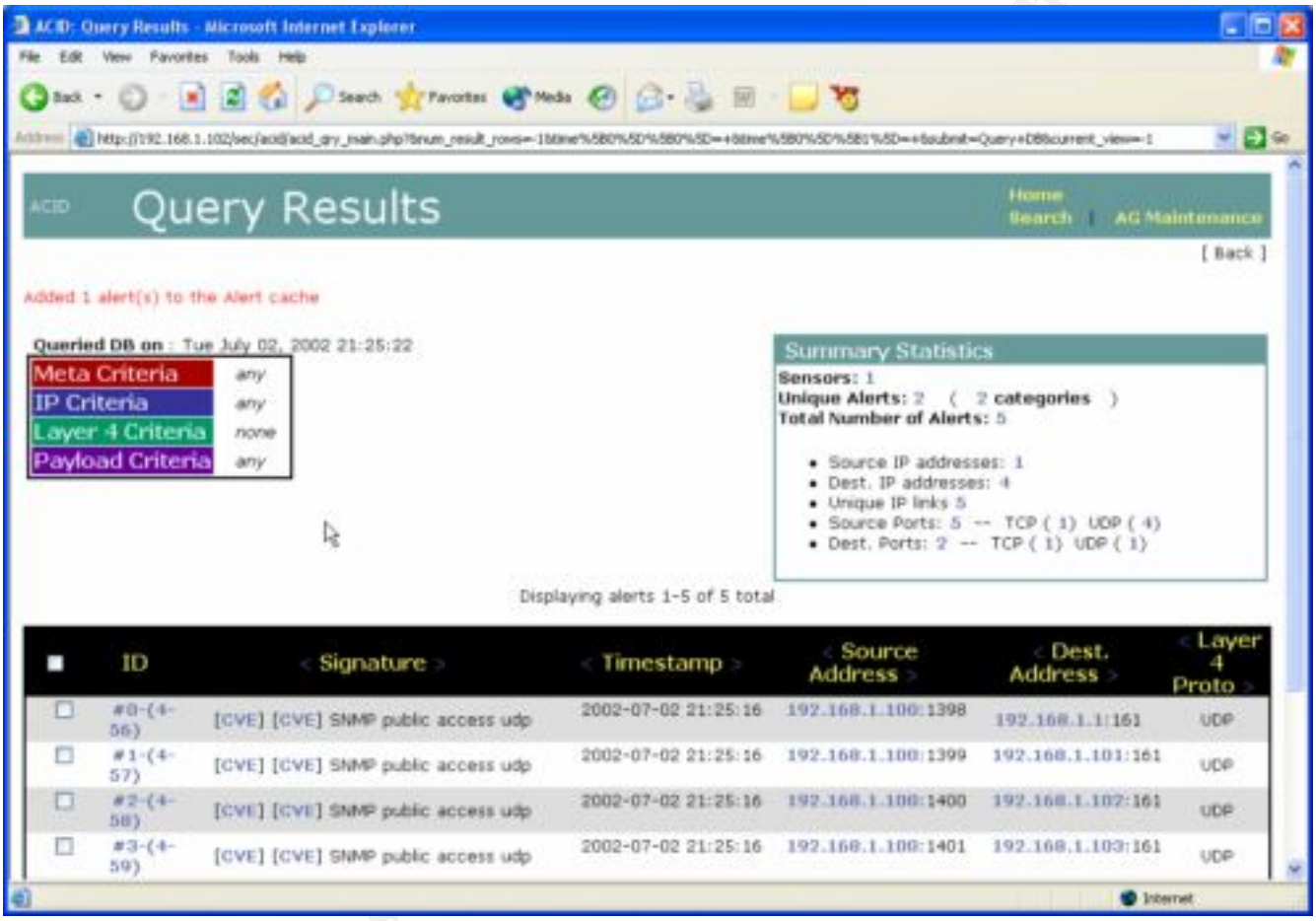
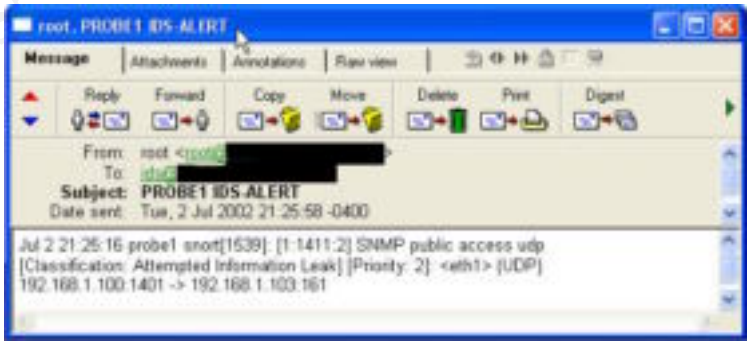


Figure 7 shows the result of real-time alerting as it pertains to the SNScan tool.

Figure 7 - SNScan Real-Time Email



Back Orifice Pinger 2.0 Results

Back Orifice Pinger 2.0 is a windows-based GUI client that can be used to look for hosts that have been compromised by Back Orifice. Its use on a monitored network should trigger Back Orifice alerts.

Figure 8 shows the Back Orifice Pinger client configured to perform a scan of the 192.168.1.0 subnet.

Figure 8 - BO Ping Client Configuration

Figure 9 shows the results of the Back Orifice scan in the ACID management console. It was successful in picking up each targeted scan.

Figure 9 - Back Orifice Ping Results in ACID

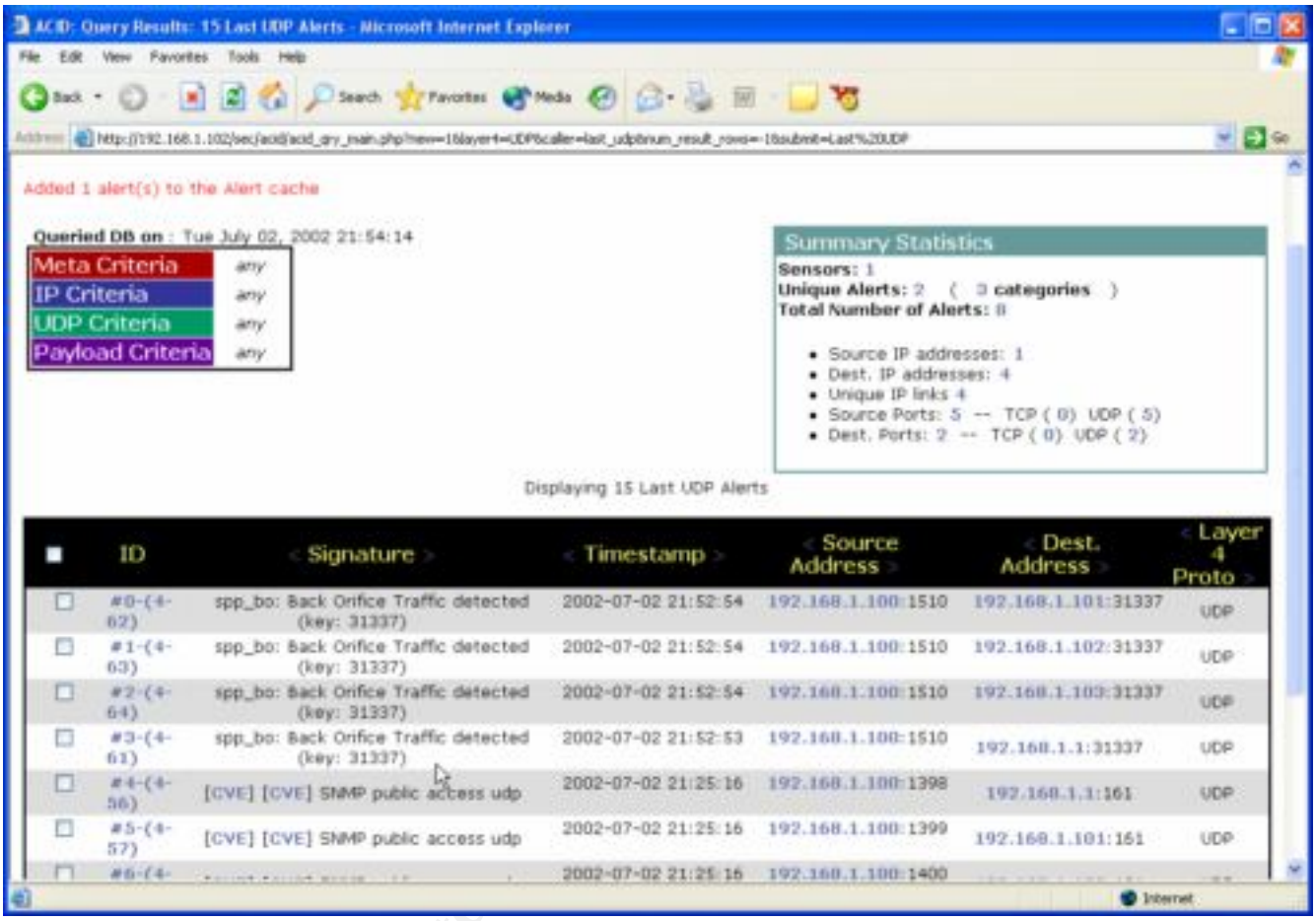
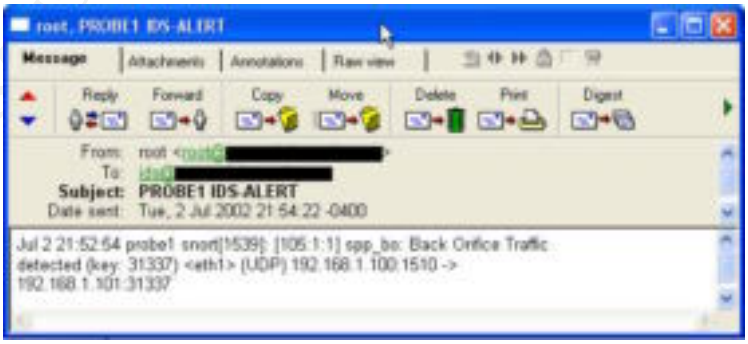


Figure 10 shows the result of real-time alerting as it pertains to the Back Orifice Ping tool.

Figure 10 - Back Orifice Real-Time Email



Snot Results

Using Snot to generate packets that resemble known vulnerabilities caused a surge in alerts to be displayed on the console and a multitude of real-time alerts to be processed. The following is the command used to generate a random set of packets.

```
[root@mail snot-0.92a]# ./snot -r snortrules.txt -s 192.168.1.101 -d 192.168.1.103 -n 5 -l
3
snot V0.92 (alpha) by sniph (sniph00@yahoo.com)

---------------------------------------
Rulefile          : snortrules.txt
Source Address    : 192.168.1.101
Dest Address      : 192.168.1.103
Number of Packets : 5
Delay (max seconds): 3
Payloads          : Random
---------------------------------------

[Parse Rules - Completed parsing 1066 rules - Sending now]

ICMP - "ICMP PING WhatsupGold Windows" - 192.168.1.101 -> 192.168.1.103
Sleeping for 0 seconds
TCP - "TELNET 4Dgifts SGI account attempt" - 192.168.1.101:43554 ->
192.168.1.103:23
Sleeping for 1 seconds
UDP - "DDOS mstream agent pong to handler"  - 192.168.1.101:5234 ->
192.168.1.103:10498
Sleeping for 0 seconds
TCP - "RPC NFS Showmount" - 192.168.1.101:59160 -> 192.168.1.103:32771
Sleeping for 1 seconds
TCP - "NETBIOS SMB IPC$access" - 192.168.1.101:26300 -> 192.168.1.103:139
Sleeping for 1 seconds
[root@mail snot-0.92a]#
```

Figure 11 shows the type of alerts received by the ACID management console. It should be noted that the IDS probe only alerted on 3 out of the 5 packets generated. A further review of the Snort signatures in use by the IDS probe revealed that the 'ksh access' and 'sparc NOOP' were not active at the time of testing.

Figure 11 - Snot Results in ACID

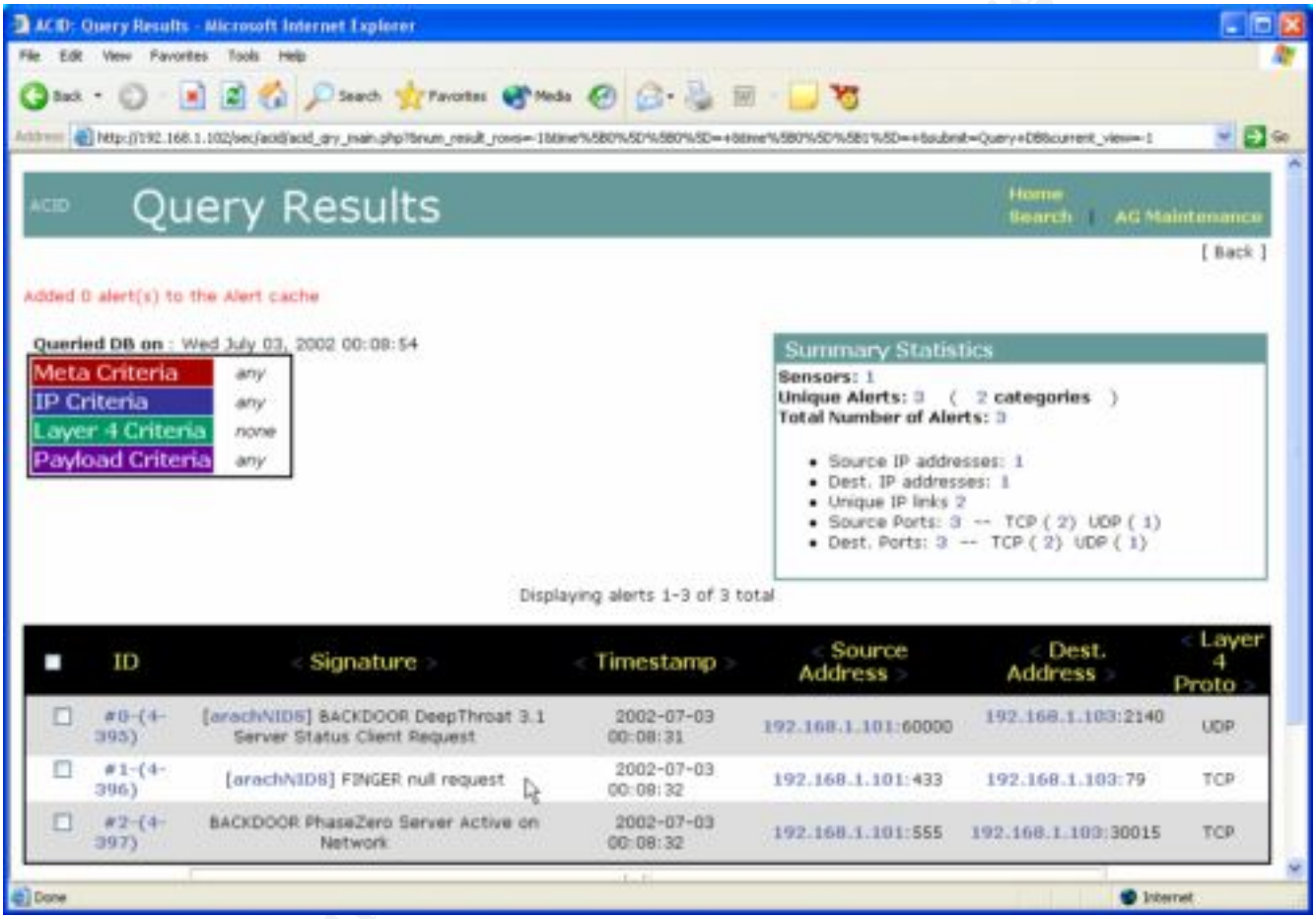
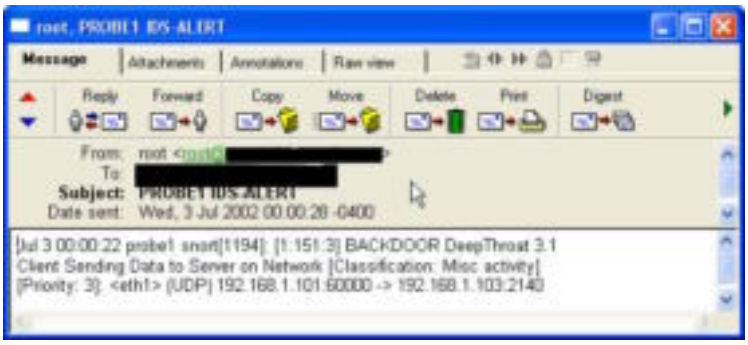


Figure 12 shows the result of real-time alerting as it pertains to the test conducted by Snot.

Figure 12 - Real-Time Response for Snot



Nmap Stealth Scan Results

Predictably, a normal Nmap scan was immediately detected by the IDS probe. The following command ran in less than 2 seconds.

```
[root@mail snot-0.92a]# nmap -sS 192.168.1.103

Starting nmap V. 2.54BETA36 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.1.103):
(The 1557 ports scanned but not shown below are in state: closed)
Port       State      Service
22/tcp     open       ssh

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
[root@mail snot-0.92a]#
```

I wanted to see how an Nmap scan would work with the –T command. The –T command allows various timing settings to be set. These settings range from very aggressive to truly paranoid. I decided to use the 'Sneaky' time setting to see if it would be detected by the IDS probe. The following command took nearly 6.5 hours to scan a single IP address. This is hardly an efficient scan but some attackers are willing to spend a great deal of time mapping a network before attempting an actual exploit.

```
[root@mail snot-0.92a]# nmap -sS -v -T Sneaky 192.168.1.103

Starting nmap V. 2.54BETA36 ( www.insecure.org/nmap/ )
Host  (192.168.1.103) appears to be up ... good.
Initiating SYN Stealth Scan against  (192.168.1.103)
Adding open port 22/tcp
The SYN Stealth Scan took 23370 seconds to scan 1558 ports.
Interesting ports on  (192.168.1.103):
(The 1557 ports scanned but not shown below are in state: closed)
Port       State      Service
22/tcp     open       ssh

Nmap run completed -- 1 IP address (1 host up) scanned in 23385 seconds
[root@mail snot-0.92a]#
```

As shown in Figure 13, the Nmap scan did not trigger the expected alert of an Nmap scan or portscan. Rather, the IDS probe was triggered by the stealth scan hitting port 8080 which matched the 'Scan Proxy (8080)' signature in the rule set being used. While the scan was still detected, this may have not been the case if the attacker had chosen to conduct his scan with a limited port range (e.g. 22,80).

Figure 13 - Stealth Scan Results in ACID

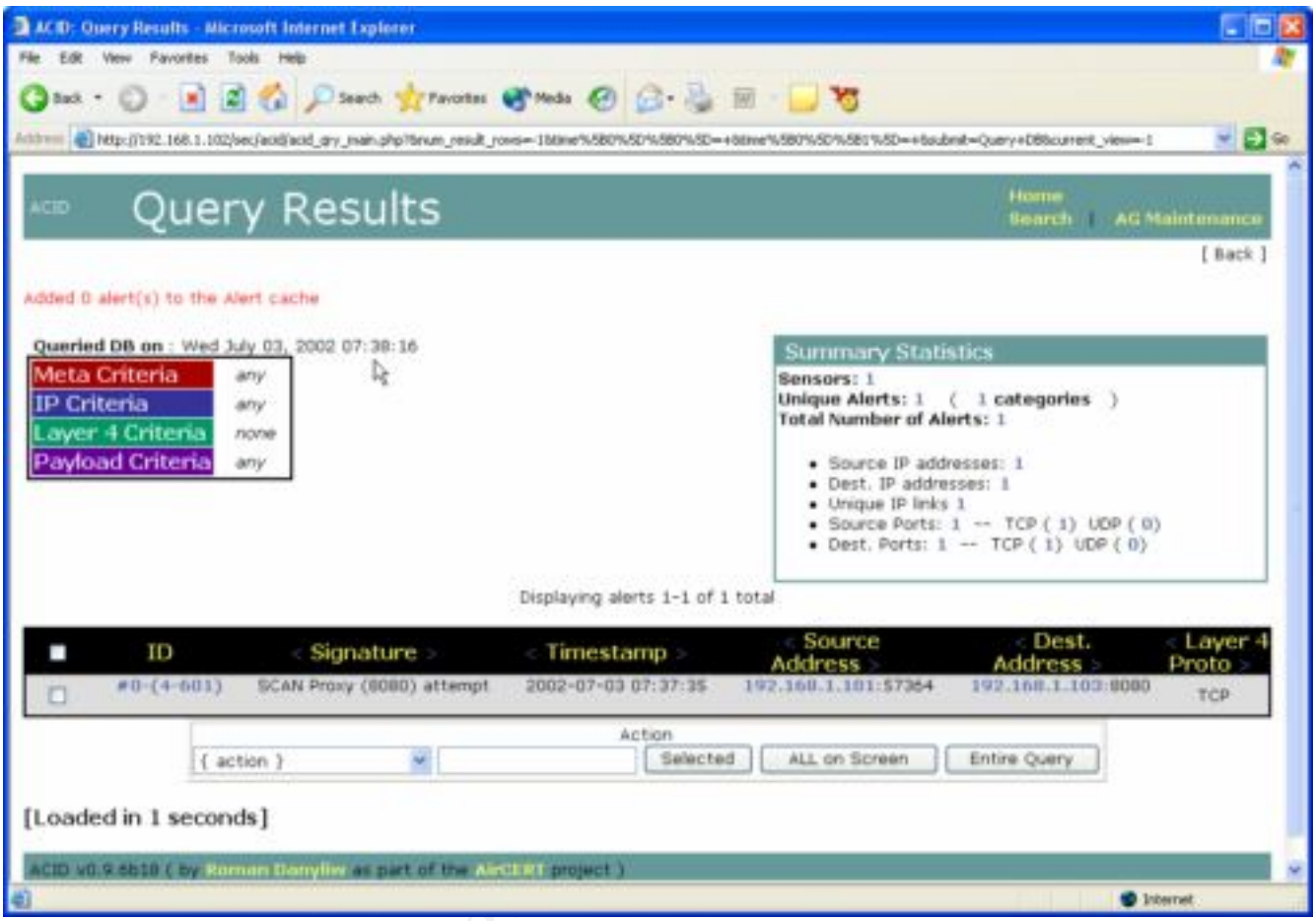
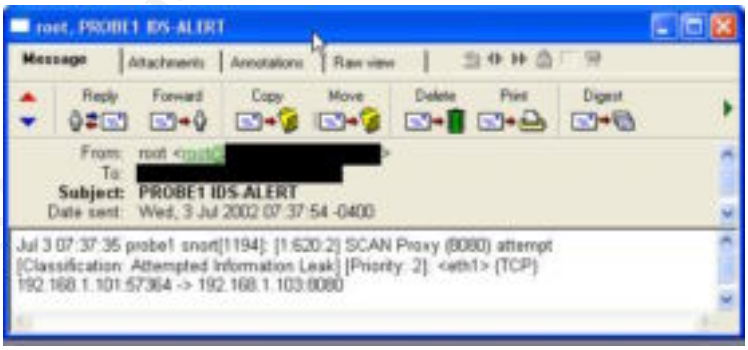


Figure 14 - Stealth Scan Real-Time Results



Checklist Item H5: FAIL

Objective: Determine if packets are being dropped by the IDS probe(s).

The following output reveals the PID for Snort on the IDS probe.

```
[root@probe1 root]# ps ax|grep snort
 1194 ?        S      1:20 /usr/sbin/snort -d -D -i eth1 -l -l /var/log/snort -O
 6899 pts/1    S      0:00 grep snort
[root@probe1 root]#
```

Once the PID is determined, a 'kill' command is entered which will generate internal statistics that contain the number of packets scanned and how many were dropped.

```
[root@probe1 root]# kill -SIGUSR1 1194
[root@probe1 root]#
```

The internal statistics are written to /var/log/messages and the output below shows over 20% of the packets were dropped by the IDS probe.

```
Jul  3 08:04:23 probe1 snort[1194]:
===============================================================
=
Jul  3 08:04:23 probe1 snort[1194]: Snort analyzed 155089 out of 155089 packets,
Jul  3 08:04:23 probe1 snort[1194]: The kernel dropped 31612(20.383%) packets
Jul  3 08:04:23 probe1 snort[1194]: Breakdown by protocol:                Action Stats:
Jul  3 08:04:23 probe1 snort[1194]: TCP: 129862       (83.734%)           ALERTS:
589
Jul  3 08:04:23 probe1 snort[1194]: UDP: 20745      (13.376%)            LOGGED:
481
Jul  3 08:04:23 probe1 snort[1194]:  ICMP: 1325     (0.854%)        PASSED: 0
Jul  3 08:04:23 probe1 snort[1194]:  ARP: 3157      (2.036%)
Jul  3 08:04:23 probe1 snort[1194]:  IPv6: 0          (0.000%)
Jul  3 08:04:23 probe1 snort[1194]:  IPX: 0          (0.000%)
Jul  3 08:04:23 probe1 snort[1194]:  OTHER: 0        (0.000%)
Jul  3 08:04:23 probe1 snort[1194]:  DISCARD: 0      (0.000%)
Jul  3 08:04:23 probe1 snort[1194]:
===============================================================
=
Jul  3 08:04:23 probe1 snort[1194]: Fragmentation Stats:
Jul  3 08:04:23 probe1 snort[1194]: Fragmented IP Packets: 0        (0.000%)
Jul  3 08:04:23 probe1 snort[1194]: Fragment Trackers: 0
Jul  3 08:04:23 probe1 snort[1194]: Rebuilt IP Packets: 0
Jul  3 08:04:23 probe1 snort[1194]: Frag elements used: 0
Jul  3 08:04:23 probe1 snort[1194]: Discarded(incomplete): 0
Jul  3 08:04:23 probe1 snort[1194]: Discarded(timeout): 0
Jul  3 08:04:24 probe1 snort[1194]: Frag2 memory faults: 0
```

```
Jul  3 08:04:24 probe1 snort[1194]:
============================================================
=
Jul  3 08:04:24 probe1 snort[1194]: TCP Stream Reassembly Stats:
Jul  3 08:04:24 probe1 snort[1194]: TCP Packets Used: 129862     (83.734%)
Jul  3 08:04:24 probe1 snort[1194]: Stream Trackers: 5757
Jul  3 08:04:24 probe1 snort[1194]: Stream flushes: 22161
Jul  3 08:04:24 probe1 snort[1194]: Segments used: 91110
Jul  3 08:04:24 probe1 snort[1194]: Stream4 Memory Faults: 0
Jul  3 08:04:24 probe1 snort[1194]:
============================================================
=
```

Evaluate the System

Ultimately, the system(s) audited can be secured with fairly minimal effort and cost on the part of the healthcare organization. The audit noted some deficiencies but not anything that could not be corrected through manual operations or a change in the security policy to reflect the actual network environment being monitored by the intrusion detection system.

The audit took several hours longer than expected due to the sheer scope of the IDS design and the research needed on my part to ensure I was auditing the proper control objectives. Future assessments of this technology should not take as long because the checklist has been created and documentation has been created to allow both the auditor and the organization being audited to have a detailed plan of attack. Research time for the next audit will be limited to discovering changes in IDS design methodology, current versions and if any new devices or networks have been added to the monitoring program. Because an audit is just a 'snapshot' for that moment in time, it is very important to ensure that future work takes the changed environment into consideration. A prime example is the recent OpenSSH and Apache vulnerabilities. At the time I created the checklist there were no recent vulnerabilities to be concerned with for these two applications. However, shortly before I was able to schedule time to conduct the audit, the vulnerabilities for both of these applications were receiving widespread media attention.

Other than the noted problems with application versions, the biggest area of concern I had was the IDS probe's ability to recognize and alert on various types of attacks. The IDS probe actually locked up twice during the audit and the network engineer was not able to determine the cause of this problem. In the end, the failure was attributed to the older hardware being used on the probe (Pentium 120 with 64 MB of RAM) and the lack of fine-tuning applied to the Snort configuration files. Another area of concern was the rule sets being used to monitor for various signs of intrusions or electronic tampering. It was not readily discernible if the rule sets being used were sufficient to cover all of the devices on the network being monitored. In addition, I was not able to determine if too many signatures were being used that would lead to degraded performance. For

example, did the IDS probe need to be configured to monitor every Microsoft IIS vulnerability or the most recent ones?

The most obvious recommendation would be to ensure that hardware is adequate to support the network when this system is moved into the production network. Consideration was given to the fact this audit was conducted in a lab environment that could not realistically portray the type of traffic that would need to be monitored on a live environment. The second recommendation I would make is to carefully evaluate the signatures being used by the IDS probe(s). Of course, the organization must be fully aware of the hardware and software platforms on their network and weigh the risk associated with each before being able to determine which signatures can safely be deleted from the Snort rule sets.

Evaluate the Audit

Because the audit encompassed multiple systems, it was extremely time consuming and required careful coordination with the healthcare organization to ensure enough resources in terms of time and engineer availability. The change control process in place at the healthcare organization required me to conduct the audit after normal business hours, which meant I spent a couple of days working from midnight to 9am to comply with their process. This was one aspect of the audit that I did not expect and it meant having to delay the audit by a couple of weeks to not impact production network coverage. It might be possible to segment the audit into more manageable sections rather than "slam" the engineer with a multi-hour intensive audit. This would entail changing the audit program to reflect logical separation of duties. For example, the audit program could be broken down into a few distinct areas such as Operating System, Access Control, Alert Reporting and Management Issues.

The strongest areas of the audit program had to do with the Red Hat OS and the MySQL configuration. Both the engineer and myself felt very comfortable with the steps in these sections and I, in particular, felt they addressed the security risks appropriately. I also feel the Apache section was sufficient for the design of this particular intrusion detection system. In a network where the Apache server would not be controlled with access control lists or on a private network, more attention would have to be paid to ensure secure access and communication between the IDS probe and the central management console.

False positives were not adequately addressed in this audit. It was not evident until the final phase of testing that I did not have a good way to ensure the alerts being triggered were, in fact, real events worthy of further investigation. This is partly due to the system being on an isolated network for testing and also the fact I was not using 'real' attack tools or exploits to test the system. For future audits, I need to spend more time researching this aspect to ensure I am developing appropriate measures. I do feel the audit was sufficient for the lab network but I would not feel comfortable conducting this phase in a live production network.

Real-time alerting, while functional, was rendered almost unusable because of the amount of alerts being generated. During the course of the audit, I received well over 300 real-time alerts to a test email account. Can you imagine an intrusion analyst receiving over 300 pages in a 48-hour period? The end result is that pages would ultimately be ignored, overlooked or the feature simply disabled. Future work on the audit plan will include a closer inspection of SWATCH and its associated configuration files.

Finally, the audit did not cover the rotation of logs contained in the MySQL database. The process of archiving in ACID is manual and currently done by the intrusion analyst on a daily basis but this is probably not practical for a production network with multiple IDS probes. Research needs to be conducted by the organization to see if there is a way to migrate files from one database to another automatically. Another deficiency in the audit plan was the failure to look at the archive retention policy, backup methods and maintenance of the IDS probe(s) and the central management console.

All of the deficiencies in the audit plan can be corrected and did not distract greatly from the overall success of the audit. Because this audit was conducted on a lab network, I am given fair warning to develop steps to cover those areas that slipped under the radar in this assessment.

Assignment Four – Follow Up

Executive Summary

This audit reviewed the control procedures and processes related to an intrusion detection system design and configuration. The audit was conducted through interviews with the network engineer responsible for the support and maintenance of the IDS configuration, software/hardware configuration, supporting documentation and other substantive testing as considered necessary.

The design asked to be certified for the production network consists of several open source tools built in a distributed client/server model. The software audited consisted of:

- Red Hat Linux
- OpenSSH
- Snort
- Apache Web Server
- ACID
- SWATCH
- MySQL
- Sendmail

The key summary items listed below are covered in detail later in this report:

- Configure the operating system to require a minimum password length of 8 characters that meets the security policy of the healthcare organization.

- Upgrade the version of OpenSSH installed on the IDS probe and management console to eliminate potential security vulnerabilities.
- Upgrade the version of Apache Web Server installed in the management console to eliminate potential security vulnerabilities.
- Develop a plan that will ensure the signatures being used by the Snort rulebase are current and consistent with the security needs of the healthcare organization.
- Upgrade the hardware being used for the IDS probe to more accurately resemble production level equipment of the healthcare organization.

During the course of the audit several items were considered deficient and will need to be corrected before the design of the intrusion detection system can be certified for usage on the healthcare organization's production network. Many of the noted deficiencies are relatively minor and will require minimal resources in terms of personnel and money to correct the items of note. It is expected that all areas of this report will be addressed in a timely fashion and there is no doubt the design and configuration of the intrusion detection system will meet all required guidelines with minimal effort.

Findings

Red Hat Linux 7.2 Operating System

Observation:

Red Hat Linux 7.2 is the standard operating system for any Linux devices deployed at the healthcare organization. The audit was conducted knowing that version 7.2 is the only supported version that is to be used on the healthcare organization's network.

Background/Risk:

Older versions of various operating systems are prone to vulnerabilities. There is a risk of Red Hat 7.2 to contain vulnerabilities that could be exploited by an attacker. However, given the fact that the healthcare organization does a very good job in keeping the OS patched and stays abreast of recent events, the risk is greatly reduced. As Red Hat 7.2 gets closer to 'end-of-life' it will no longer be supported.

Recommendation:

While not a specific audit point, it should be noted that Red Hat version 7.3 was released in May 2002. It may be worthwhile to investigate the feasibility of migrating to the latest version available from Red Hat. Red Hat 7.3 offers increased functionality and stability that the healthcare organization may find beneficial as they continue to migrate applications to the Linux platform. At this point, there is no compelling reason to migrate to the latest version but it should be considered.

Cost:

Cost is dependent on a number of variables ranging from the number of machines currently on older releases, the support contract current being used and any training that may be required to ensure proper usage, configuration and support of the newer release of the operating system. Because Red Hat 7.3 is 'free' there are still costs that

need to be considered before making the choice to deploy the new version across the enterprise.

Compensating Controls:

Security awareness in the form of staying abreast of recent vulnerabilities and ensuring systems are patched to the most current release is sufficient for the time being.

Observation:

The IDS probe being audited was not properly configured to conform to minimum password quality standards as set forth in the healthcare organization's security policy. As stated in the policy, the minimum password length is 8 characters with a rotation basis of 90 days. The IDS probe is configured for a minimum password length of 6 characters.

A check of the /etc/login.defs file showed the following configuration:

```
[root@probe1 /]# more /etc/login.defs
# *REQUIRED*
#   Directory where mailboxes reside, _or_ name of file, relative to the
#   home directory.  If you _do_ define both, MAIL_DIR takes precedence.
#   QMAIL_DIR is for Qmail
#
#QMAIL_DIR      Maildir
MAIL_DIR        /var/spool/mail
#MAIL_FILE      .mail

# Password aging controls:
#
#       PASS_MAX_DAYS           Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password
changes.
#       PASS_MIN_LEN    Minimum acceptable password length.
#       PASS_WARN_AGE  Number of days warning given before a password expires.
#
PASS_MAX_DAYS           90
PASS_MIN_DAYS           0
PASS_MIN_LEN            6
PASS_WARN_AGE           7

#
--More--(51%)
```

Additionally, a test User ID was created with a password containing only six characters.

Background/Risk:

Passwords are easily cracked or guessed. Passwords are often deployed as the first and last line of defense so it is critical to the security of a system to enforce a strong password policy.

Recommendation:

The configuration for the IDS probe (probe1) at 192.168.1.103 should be changed to reflect the security policy of the healthcare organization. In this case, the minimum password length should be changed to 8 characters and tested by attempting to create a User ID with less than the specified minimum length.

The timeframe for the audit did not allow for a password strength assessment. Therefore, it is my recommendation that this be conducted by the healthcare organization at their earliest convenience.

Cost:

The cost for making this change is minimal as it only affects one server. Any existing passwords should be changed when the configuration file is modified to reflect the new minimum length.

OpenSSH Configuration for IDS Probe and Management Console

Observation:

The version of OpenSSH being used on the IDS probe and management console is 3.1p1 which may contain a serious vulnerability if not properly configured. The network engineer that was present for the audit explained that because this IDS system resided on an isolated network and was not being used on a production network that he would wait to upgrade to the latest version of OpenSSH.

The following command shows the version being used:

For the IDS probe:

```
[root@probe1 /]# ssh -V
OpenSSH_3.1p1, SSH protocols 1.5/2.0, OpenSSL 0x0090602f
[root@probe1 /]#
```

For the management console:

```
[root@mgmt1 /]# ssh -V
OpenSSH_3.1p1, SSH protocols 1.5/2.0, OpenSSL 0x0090602f
[root@mgmt1 /]#
```

Background/Risk:

According to an announcement on 26 June 2002, earlier versions of OpenSSH (including the version currently running on the intrusion detection system) are

vulnerable to privilege escalation, meaning an attacker can gain administrative or root level access by exploiting a flaw in the OpenSSH code base.

Recommendation:
Version 3.4 of OpenSSH should be installed on the IDS probe(s) and management console prior to this system being moved into a production environment. It should be noted that all production systems at the healthcare organization have already been upgraded to the most recent release and went through an "emergency" change control process at the time the vulnerability was announced.

Cost:
Because the healthcare organization already has a process for upgrading to the latest version of OpenSSH, cost should be limited to the time it takes to upgrade the IDS probe(s) and management console. Based on the Change Control records each device being upgraded takes approximately 30-45 minutes.

Compensating Controls:
According to the announcement from OpenSSH, it is possible to prevent the privilege escalation by disabling ChallengeResponseAuthentication in the ssh_config file and disabling PAMAuthenticationViaKbdInt in the sshd_config file.[9] This should be considered a short-term solution only.

Snort Configuration for IDS Probe

Observation:
There is currently no method for ensuring the signatures used by Snort are kept up to date. The following output reveals a difference in the rule set when compared to the most recent update available for the Snort web site:

```
[root@probe1 rules]# more snort.conf |grep Id
# $Id: snort.conf,v 1.77.2.19 2002/06/29 13:32:48 chrisgreen Exp $
[root@probe1 rules]# more /etc/snort/snort.conf |grep Id
# $Id: snort.conf,v 1.77.2.7 2002/03/02 05:33:01 cazz Exp $
[root@probe1 rules]#
```

The version should be 1.77.2.19 rather than the 1.77.2.7 being used on the IDS probe.

Background/Risk:
New signatures are released on regular basis (sometimes daily) as new vulnerabilities are announced. It is critical the success of an IDS deployment to be able to detect even the most current of vulnerabilities. A recent example is the OpenSSH vulnerability.

[9] "Internet Security Systems Security Advisory – OpenSSH". 26 June 2002. URL: http://www.openssh.com/txt/iss.adv (8 July 2002).

Without updated signatures, the IDS probe would not be able to alert to any attacker probing for exploitable versions of the SSH service.

Recommendation:
A regular check of the snort.org web site should be done by the network engineer responsible for maintaining the intrusion detection system. New signatures are available on a seemingly daily basis and need to be integrated into the current IDS configuration to ensure the most complete coverage possible for the monitored networks.

There are automated scripts available to aid in keeping Snort rules current. Oinkmaster (http://www.algonet.se/~nitzer/oinkmaster/) is a good example of the usage of the Perl scripting language to automatically check for new updates and move them to a pre-configured directory. The script can be set up as a nightly function or run manually from the command-line.

Cost:
Keeping the rule base current can be a time consuming and arduous process. Because this needs to be maintained on a daily basis, an estimated 25-30 minutes per day is needed for a network engineer to check for current updates. This time could be lessened with the usage of CRON to automate the updating of the rules. However, manual configuration may still be needed.

Apache Web Server – Management console

Observation:
Although the version of Apache being used was considered current at the time the audit checklist was developed, a security vulnerability was recently announced that should be addressed. According to the Apache web site (http://httpd.apache.org/), the most current version of the web server is 1.3.26. The version being used on the management console is 1.3.24 as shown in the output below:

```
[root@mgmt1 bin]# httpd -v
Server version: Apache/1.3.24 (Unix)
Server built:   May  7 2002 18:14:40
[root@mgmt1 bin]#
```

Background/Risk:
On 20 June 2002, the Apache Software Foundation announced a potentially serious vulnerability that could allow an attacker to gain administrative or root level access to any server running the Apache web service.[10] This announcement was made due to the fact a security researcher by the name of "Gobbles" released code on 19 June 2002 to several security related mailing lists that would take advantage of a vulnerability in OpenBSD versions of Apache.[11] It is noted that this exploit is only known to affect

[10] "Apache Security Bulletin". URL: http://httpd.apache.org/info/security_bulletin_20020620.txt. (23 June 2002).
[11] Gobbles. "Remote Apache 1.3.x Exploit". URL: http://online.securityfocus.com/archive/1/277830 (24 June 2002).

OpenBSD at this time but as is often the case, it is just a matter of time before a Linux version of this exploit will become freely available.

Because of the access controls being used and the fact the web server resides on a restricted/trusted network, there is a very slight risk involved with running an older version of the code. However, the potential exists for this web server to be located on a network without the safety of ACL's so the risk deserves more than a quick glance.

Recommendation:
The version of the Apache software being used on the management console should be upgraded to version 1.3.26 prior to moving into a production network environment. At a minimum, all recommended patches should be applied as soon as possible to mitigate the vulnerability exposure.

Cost:
Based on an estimate given by the network engineer responsible for the IDS support and maintenance, time is needed to explore the upgrade process and test the configuration. An initial estimate of 15 hours was given but this should be considered a raw estimate only.

Compensating Controls:
There are a few compensating controls already in place to include severely restricted access control lists and proper server placement on the trusted network. Applying all recommended patches will help mitigate the effect of this potential exploit until the web server can be upgraded.

Combined Probe(s) and Management Console Assessment

Observation:
Deploying various 'hacker' tools on the test network did not yield the expected results. In particular, packets generated by Snot to simulate various attacks were not completely detected by Snort. Also, various Nmap scans were virtually unnoticed by the Snort engine due to various timing sequences used to attempt to evade the intrusion detection system. In addition, tests revealed that over 20% of the packets on the test network were dropped by the IDS probe.

Background/Risk:
Attackers commonly deploy the tools used in the audit to map a network before attempting an actual exploit. Sometimes, attacks are automated (e.g. Code Red) and will scan entire subnets looking for devices that can be exploited. It is imperative that these types of probes and/or attacks trigger alerts in the intrusion detection system. Otherwise, the use of IDS is just an exercise in futility and the system is rendered useless.

Given the fact this audit was conducted on an isolate network, there is cause for concern because the Snort engine was not able to detect possible signs of intrusion on

a ‘quiet’ network. It stands to reason that these effects would be amplified on a production network with a live connection to the public network.

Recommendation:
A careful analysis of the signatures being used by the Snort engine is needed to ensure all possible scenarios are covered in the event of an actual intrusion or electronic tampering. Consideration should be given to the types of devices residing on the monitored network (e.g. Microsoft IIS, Sendmail server, FTP, etc) to ensure sufficient coverage. Further, signatures must be updated on a regular basis to stay current with new vulnerabilities.

It is highly possible the reason for so many dropped packets is because of the hardware configuration of the IDS probe. The probe used for lab testing was an older machine with a very slow processor (120 MHz) and insufficient memory (64 MB) to accommodate the amount of traffic on the network during testing. It is understood that this was a test environment and that updated hardware would be installed for a production environment. However, the test did reveal that newer hardware (Pentium 300 MHz and at least 128 MB of RAM) will be needed to guarantee the most complete coverage on a live network. The most obvious recommendation is to upgrade the hardware being used not only on the live network but also in the lab to enable the lab to lend some value to the testing process.

Cost:
Based on conversation with the network engineer responsible for the testing of the IDS configuration, a “bare-bones” system that is capable of handling live network traffic cost between 1300-1500 dollars. This price was not verified and is best left in the hands of the person responsible for “spec’ing” systems for the healthcare organization. The network engineer estimated it would take a “few hours” to configure a new probe for deployment on a production network.

Compensating Controls:
It may be possible to mitigate packet loss by disabling all but the essential signatures. That is, use only the signatures needed to provide protection for the devices on the monitored network while eliminating those that could be considered overhead. This approach runs the risk of missing a potential intrusion but if the network engineer has a solid understanding of each device on the network; it stands to reason that many of the signatures could be safely deleted from the rulebase.

Conclusion
With the exception of the previously identified observations, the IDS configuration appears reasonable in meeting the healthcare organization’s business and security needs. The inclusion of the aforementioned recommendations will enhance the overall security posture of the intrusion detection system and lend itself to future deployment in the healthcare organizations production network. Those items marked deficient during the course of the audit and outlined in this report will need corrective action before the system can be certified for production usage.

References

- "Apache Security Bulletin". URL: http://httpd.apache.org/info/security_bulletin_20020620.txt. (23 June 2002).
- "CIS Level-1 Benchmark and Scoring Tool for Linux". URL: http://www.cisecurity.org/bench_linux.html (15 June 2002).
- "Internet Security Systems Security Advisory – OpenSSH". 26 June 2002. URL: http://www.openssh.com/txt/iss.adv (1 July 2002).
- "login.defs". URL: http://docs.csoft.net/cgi-bin/man.cgi?section=5&topic=login.defs (23 May 2002).
- "MySQL Manual: General Security Guidelines".URL: http://www.mysql.com/doc/G/e/General_security.html (15 June 2002).
- "MySQL: Selecting All Data". URL: http://www.mysql.com/doc/S/e/Selecting_all.html (15 June 2002).
- "MySQL: The Command-line Tool". URL:http://www.mysql.com/doc/m/y/mysql.html (15 June 2002).
- "Nmap Network Security Scanner Man Page". URL: http://www.nmap.org/nmap/nmap_manpage.html (22 May 2002).
- "rdate manual page". URL: http://linuxcommand.org/man_pages/rdate1.html (23 May 2002).
- "Technical Security Services to Guard Data Integrity, Confidentiality and Availability". Proposed HIPAA Security Regulations. URL: http://www.hipaadvisory.com/regs/securityandelectronicsign/technicalsecur.htm (23 May 2002).
- Andrews, James. "Time for Linux". URL: http://www.linuxplanet.com/linuxplanet/tutorials/215/1/ (23 May 2002).
- Danyliw, Roman. "ACID: Installation and Configuration". URL: http://www.andrew.cmu.edu/~rdanyliw/snort/acid_config.html (16 June 2002).
- Deraison, Renaud. "Nessus Demonstration". The Nessus Project. URL: http://www.nessus.org/demo/index.html (23 May 2002).
- Herzog, Pete. "Open-Source Security Testing Methodology Manual". 26 February 2002. URL: http://www.ideahamster.org/download.html (23 May 2002).
- Laude, Mary. "Auditing Red Hat Linux 7.0" 23 July 2001. URL: http://www.giac.org/practical/Mary_Laude_GSNA.zip (23 May 2002).
- Mourani, Gerhard. Securing and Optimizing Linux: The Ultimate Solution. Montreal: Open Network Architecture, Inc, 2001.
- Poppi, Sandro. "Snort-Setup for Statistics HOWTO". 23 February 2002. URL: http://www.tldp.org/HOWTO/Snort-Statistics-HOWTO/ (31 May 2002).
- Ratliff, Richard L. Internal Auditing: Principles and Techniques. Altamonte Springs: The Institute of Internal Auditors, 1996.

- Roesch, Marty. “Snort Users Manual”. URL: http://www.snort.org/docs/writing_rules/ (23 May 2002).
- Shipley, Greg. “Intrusion Detection, Take Two”. 15 November 1999. URL: http://www.networkcomputing.com/1023/1023f19.html (23 May 2002).
- Spitzner, Lance. “Armoring Linux”. 19 September 2000. URL: http://www.enteract.com/~lspitz/linux.html (23 May 2002).