Global Information Assurance Certification Paper

Broad Scope Audit of a

Filtering Router in an Academic Setting

Luis Grangeia

SANS GSNA Assignment (Version 1.1)

12 December, 2001

Abstract

Universities are and have been the common birthplace for new technologies, including most of the protocols used for communicating in the internet. Their networks are home for every kind of user, ranging from the simple student browsing the web, to researchers using enormous amounts of bandwidth. Like in the "outside world", a war takes place every day on these networks, and it is a fierce one. It is very likely that the majority of hackers and crackers are mostly students, so academic networks are usually their first target, making the job of administering these networks a very interesting and motivating challenge.

Despite the academic nature of these networks, it is always necessary to impose some level of network security, by 'sanitizing' its traffic at some point. That's where our filtering router enters the picture.

Objectives

This work is an applied research project that attempts to build on existing methodologies to improve the current state of practice for security auditing a particular system.

The first part of this assignment will research the current state of practice for the auditing of the type of system described below. An analysis will be made on some existing audit resources and some existing problems will be enumerated. Attempting solve those problems, a skeleton for an audit methodology will be drafted. After that, the author will create a sample checklist for auditing the system that assists the auditor to conform to that methodology. While thoroughness is in the top list of requirements for the resulting checklist, the prime objective in the writing of this paper is to present a checklist that will cover a very broad scope of security issues, as to raise awareness in management and even system administration/operation sectors.

System description

The system to be audited is a newly installed filtering router in the perimeter of an academic network, a x86 machine running the 2.4.x version of the Linux kernel system. The network design that serves as the gateway to the internet is as shown below in figure 1. Here are some facts and considerations about the system and its surrounding network architecture:

- The campus is served by a 7.5 Mbit/s ISDN connection to the internet
- The border Cisco router is a very old model with only 10Mbit/s half duplex Ethernet ports, and that can originate a network bottleneck[1].
- The machine serving as a filtering router has an Athlon 1.2 Ghz processor, 512 Mbytes of RAM and a 20 Gigabyte hard drive.
- The machine is equipped with two Ethernet NIC's; one connects to the internal network at 100 Mbit/s full duplex, the other connects to the border router at 10 Mbit/s half duplex.
- The operating system installed on the filtering router is SuSe Linux[2] version 7.2
- The packet filtering engine used in the filtering router is from the 2.4.x kernels, netfilter/iptables[3].

---

[1] In Ethernet networks, 10Mbit/s half duplex can become the practical equivalent of 5Mbit/s full duplex when there is the same amount of data to be transmitted on each side. Even worse, if there is a major amount of outbound traffic (ie. 8Mbit/s), inbound traffic can suffer serious bottlenecks here (inbound traffic will be limited to 2 Mbit/s, instead of the normal 7.5Mbit/s). This should be reported to management ASAP.

[2] http://www.suse.com

[3] http://netfilter.samba.org

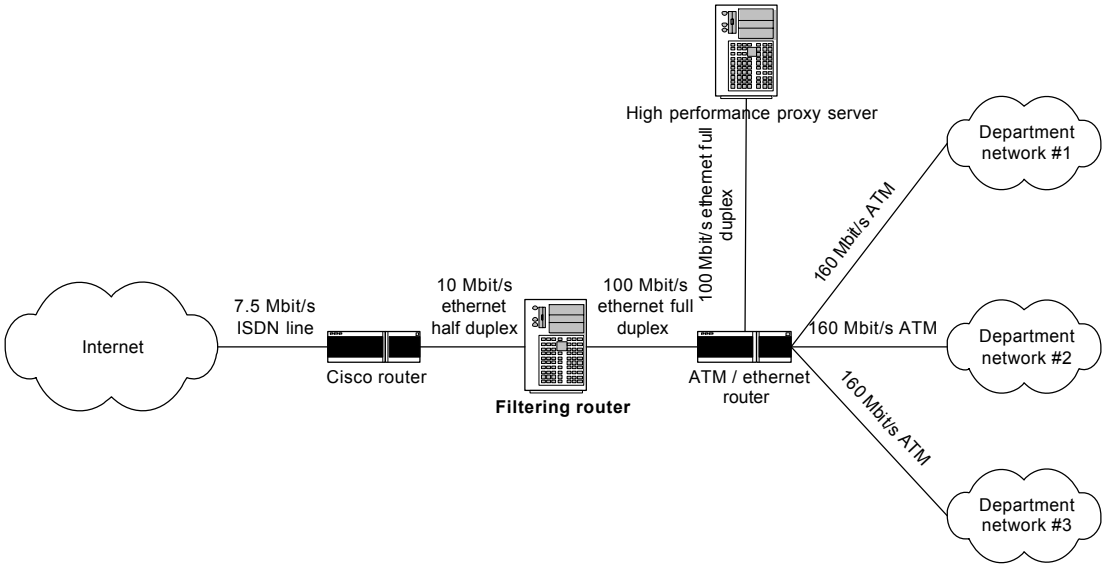


Figure 1 - network design

Framework

Before any research is conducted, even if there is a clear objective on what to research, it is essential to define a structure, or how research material will be presented. To divide the research into sections, some framework must be given beforehand, and that will be the general audit process[4]:

- **Audit Planning** – This is where research will actually be performed, and will prepare the following phases. Research will be distributed into three main tasks:
    - **Review pertinent background information** – This is a very important part of research, as it will give the auditor a good feel of the organization's setting and its particular security needs. The auditor should not only research the particular organization itself, but look at similar organizations and solutions available.
    - **Research policies** – With the knowledge acquired above, the auditor can now look at the organization's security policy documentation and evaluate it for correctness and conformity to best practices for this type of organization.
    - **Prepare the audit program** – Having a good feel of the organization's needs and its security policy, the auditor should now have a good idea of what to test, and should be able to draft a checklist for auditing the selected system or systems.
- **Entrance Conference** – This is the last step before fieldwork. The auditor meets with IS group leaders to let them know what is going to be audited and ask if there is any specific area to be audited so that changes to checklist can be done and prepared before fieldwork. It is vital that the scope of the audit is delineated clearly in this meeting, so that the audit is focused on the needs of management.
- **Fieldwork** – Here the auditor visits the IS systems and performs the steps listed in the audit program. He will verify that the system is performing has expected and that its behavior is as indicated in the organization's security policy documentation.

[4] The "general audit process" as taken and adapted from SANS course material by Randy Marchany

- **The Audit Report** – Here the auditor will look at the data gathered from fieldwork and draft a report containing:
    - What was done
    - Results of the actions performed
    - The criteria used to evaluate those results
    - Recommendations when deemed necessary
    - All the audit checklists used to collect the data

After these steps are performed the auditor should meet with IS group leaders and clear out any misunderstandings in the draft report, that is then finalized and sent to upper management.

The author will now proceed with research work structured as this framework suggests.

Audit planning

Overview

Lance Spitzner writes in his paper on auditing firewalls[5] that "Our first step in auditing our firewall is defining what we expect". We can easily generalize this sentence for every kind of systems, not just firewalls. So what should the auditor expect? Being an outside element to the organization, the auditor will most often start completely unaware of the organization's security practices or policies. So before he starts evaluating the policies and security practices for that specific organization, the auditor should position himself in a knowledgeable position of the organization's environment; he needs to look at the global picture and a research a base set of best practices for that specific environment, which in this case is an academic setting.

Background research – Academic setting

Contrary to the author's expectations, there is quite some good material on university security on the internet. It appears that the security community has realized what most academics have not yet grasped: universities are the target of choice for most attackers. Douglas P. Brown has an excellent paragraph on his article "University Security":

---

[5] Spitzner Lance. *"Auditing Your Firewall Setup"*

> Networks comprised of heterogeneous hosts with fast Internet connections make universities desirable targets to a wide variety of attackers. Members of university communities are often not concerned with security because they assume that hackers attack systems to obtain confidential information. These academics have not realized that many attacks are instead quests for disk space or processor time and that the information stored on a server is sometimes irrelevant to the attacker. The resulting lack of system security at universities has allowed attackers to quickly make universities the preferred staging areas for distributed denial of service attacks.[6]

Securing an academic network is not an easy task. In fact, corporate standards of security will probably never be achieved, mainly because of these problems:

- Universities are learning centers and therefore require an open network environment with little to no network traffic filtering.
- Threats are not localized and can originate from all sides, internal and external.
- An academic network rarely has central management and operation. It is often consisted of several independent department networks connected together.
- Human resources are limited and cooperation between departments for a common and centralized set of security practices is rarely peaceful.

This sort of problems is generally not faced by typical corporate networks. Let's look at a short comparative description of the typical environment in each of these two different environments:

Academic Network	Corporate Network
• Almost every node in the network is visible to the internet • Network security is non-obtrusive and permits almost all traffic to circulate • All hosts security must be very tight because of lax network security. • Most internal hosts cannot be trusted (lab workstations, student laptops).	• Internal network closed from the internet • Network security limits all types of traffic except what is allowed by the security policy • Hosts in the external network are secure, but internal hosts are usually relying on network security • The hosts on the internal network tend to be trusted

[6] Brown, Douglas P. *"University Security"*.

Table 1 - typical characteristics of academic and corporate networks

It should be noted that, in the case of a corporate network, having internal hosts rely on network security is not a good or desired aspect. Multi-layered security is a well documented principle, but in practice, when a firewall is in place, internal host security is often overlooked. In the IDS FAQ by the SANS institute[7], the layered approach is compared to the things one does at home in a winter storm: "It's this utilization of separate things in the household that results in an overall approach that gives us that warm and fuzzy feeling in a winter storm". One can never rely on just one device for security. The same document warns: "The most common misconception is that a firewall will secure your computer facilities and additional steps don't need to be taken".

One could compare both types of networks, corporate and academic, to a fortress and a shopping center, respectively; a fortress has very strong defenses set in the perimeter, while its level of protection in the inside is usually very low. On the other hand, a shopping center by definition lets almost everyone in; its level of protection is usually low to nonexistent at the perimeter, but the stores inside must have greater protection if they don't want to be stolen.

So if universities have that much of an open network environment, why would they even need a filtering router? Why don't they route all traffic in and out and rely on host security alone? Using the comparison between academic networks to shopping centers, let's picture these scenarios:

- One person enters the shopping center and yells that there is a product that is selling much cheaper at another place. This creates a large flow of people from the inside of the building to the outside, making everyday shopping a slow and irritating process.
    - This could be translated into computer terms as a *smurf* attack, as described in the respective CERT advisory[8].
- A shop is attracting a lot of people to the center, flooding the entrance of our shopping center, making entering and exiting a very difficult task for people in general.
    - This configures as a heavy network traffic configuration in computing terms, and may not be caused by an intentional attack; nevertheless it constitutes a security (availability) problem.
- A very bad looking person enters the center, and it looks like he is going to try and steal one of the stores, but no one realizes his intentions until the store is hit. Only

[7] The SANS Institute. *"Intrusion Detection FAQ"*

[8] CERT. *"CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attack"*

then the store clerk realizes the attempt and may or may not manage to stop the robbery.

- The absence of some form of network intrusion detection system in any (academic or corporate) network is a serious problem. This kind of "standard" attack could be easily detected by matching the packet contents to a database of common attacks, and most of them could just as easily be stopped by a network packet filter.

Having realized the challenges that face an academic setting, the auditor should now have a general idea of what should be the security controls and purpose of the system that will be audited. We will call the subject of our audit a 'filtering router', and not a firewall, despite that the system will be performing some typical firewall functions. This will be done to prevent mistaking it with a corporate-type firewall, as this is a typical assumption when referring to a firewall system.

In most cases, there is a higher level of trust on the internal network protected by a firewall. In NIST's security policy guide[9] a firewall is defined as "a safeguard one can use to control access between a trusted network and a less trusted one". But why would we trust an academic network, that is open to the public in general and where the trust level of connected hosts cannot be established on all segments? Since we are talking about an open network, its trust level is very close to that of the external internet. There is, however, a more general definition that applies to our system, in Curtin and Ranum's firewall FAQ[10]: "A firewall is a system or group of systems that enforces an access control policy between two networks". We are in fact enforcing an access policy on the network traffic passing through our system, as it will be shown below. Also, calling our system a firewall would induce the reader to think of the inside network as a closed network and that the access policy exists only to protect the inside network. This is clearly not the case, in a university setting.

Security policy research

Having realized the importance of perimeter protection even in this type of environment, it is necessary to create an organized document that will state "the set of laws, rules and practices that regulate how an organization manages, protects, and distributes information" as defined by Stockstale[11]. We are referring to a system's *security policy*. It's essential that any system operates as expected by everyone, and that implemented security mechanisms have a well documented purpose.

---

[9] Nistgov. "*Internet Security Policy: A Technical Guide"*. Chapter - 6. Internet Firewall Policy

[10] Curtin, Matt. Ranum, Marcus. *"Internet Firewalls: Frequently Asked Questions"*

[11] Stocksdale, Greg. *"NSA Glossary of Terms Used in Security and Intrusion Detection"*

However, in this particular assignment, such document was found not to exist in paper, and so it was necessary to assert the purpose of our filtering router and its security controls. Without the existence of a written security policy, the auditor usually does this by interviewing system administrators and management. This particular filtering router was supposed to work as shown:

- Protect the inner network from *smurf* attacks
- Do not allow that the inner network is used to launch *smurf* attacks
- Do not allow spoofed and otherwise illegal traffic pass through the router
- Log the most typical attack attempts with the *snort* network intrusion detection system
- Redirect HTTP traffic transparently to a proxy server with minimal impact for internal network users
- Make network traffic statistics available in real time to operating staff
- Minimize the visibility of the router, making it only available to staff for monitoring and patching/configuration updates.
- If not specified above, all traffic should pass unmolested through the router

This list is far from perfect, but it is our starting point. Evaluating this list of proposed uses for our filtering router is also part of the audit procedure.

Preparation of the Audit Program

Overview

First of all it is necessary to look at our system and understand that it is not made from a single component. Every information system is composed of many different subsystems, and a truly thorough audit should include the evaluation and verification of the security controls in all of them; from the simple checking of backdoors of a computer's BIOS or the radiation emitted by the console's monitor; to checking configuration files of the higher software layers for errors, or evaluating security awareness and training of the system operating staff.

In the real world such an audit is, however, rarely performed, be it due to time constraints, lack of management awareness of the scope of their systems, or simply to poorly trained auditors. It is the author's opinion that this type of audit should be performed at least once on every system connected to the internet. With this in mind, this document will attempt to refer the main areas of the system and research mostly on issues that seem to need improvement.

Since our system shares the functions of routers and firewalls, research will have to focus both these systems. Also it is necessary to specifically check for information on systems using netfilter/iptables.

What needs improving?

A preliminary search on the internet found a lot of work on installing and auditing routers and firewalls. Most of the material available, however, was found to be very corporate-centric, there was no middle point; or the internal network was completely closed to outside users, or it was wide open. Surely most of the protections employed in an academic environment should be employed in corporations, but the opposite does not always apply. Some very good material was found that centered on this particular environment, such as Elizabeth Mackenzie's paper, *"Perimeter Filtering in a University Setting"*. It contains several very good recommendations for system administrators and configuration options, but lacks an auditor's perspective. The configurations, while made for Cisco routers, can be easily adapted to any other packet filtering system.

While there is some work on Linux firewall and router audits, there isn't much material on auditing an iptables configuration. There are some good tutorials on how to setup a simple host firewall using iptables, like Joshua Drake's paper[12], or Oskar Andreasson's extensive tutorial[13], but there was no written method for auditing an iptables rule set.

Also, the author believes that most existing audit methodologies are somewhat static and allow little or no space for improvement; it is the author's opinion that these methodologies should be 'retouched', so that it is possible for the auditor to contribute more actively to the general audit procedure. At the same time that the auditor performs a series of tests against a particular information system, he should also be evaluating his guidelines and verify that they are accurate and helpful.

As an example, the author will now look into an existing checklist, analyzing it and adapting it to his current needs.

Management Analytics Firewall checklist

Management Analytics Firewall checklist[14] is very broad and focuses on a large number of aspects, while remaining general enough for use in most firewall implementations. It is divided into these categories (See appendix A for the full checklist):

- Basic Firewall Control Principles
- Management Issues
- Policy Issues
- Standards/Procedure Issues
- Documentation Issues
- Audit Issues
- Technical Safeguards
- Incident Response Issues
- Testing
- Physical Security
- Personnel Issues
- Legal Issues
- Awareness Issues
- Training and Education
- Organizational Suitability

Each of these categories contains a series of items that can be marked true or false, and a score is awarded after the completion of each category. This checklist presents several good and not so good aspects:

---

[12] Drake, Joshua. *"10 minutes to an iptables-based Linux firewall"*. URL: http://www.linuxworld.com/site-stories/2001/0920.ipchains.html

[13] Andreasson, Oskar. *"iptables Tutorial"*. V.1.1.3. URL: http://people.unix-fu.org/andreasson/iptables-tutorial/iptables-tutorial.html

[14] Management Analytics and Others. *"Management Analytics Firewall Checklist"*.

The Good:

- It is extremely broad in scope and covers many aspects of firewall security, including management, procedure, and policy issues.
- It is general enough to be applied on most firewall systems.
- Provides a crude scoring mechanism that can be useful to the auditor to have an overall idea of the system's security level.

The Not So Good:

- It contains no instructions on how to verify the facts. While it can be easy to verify that "a complete internal audit is done of each firewall at least once every 6 months", it is very subjective to verify that "the systems administration interface makes managing the firewall straight forward". What is straight forward for the auditor could not be for the system administrator.
- It does not contain more technical aspects of firewall auditing, such as rule set verification.
- It does not mention anything about logging mechanisms or intrusion detection systems to be used on the firewall system.
- While useful for limited purposes, its scoring system is not sufficient for a complete evaluation of the audit.
- Because very extensive, it is easy for an inexperienced auditor to get lost in formalities and miss the most important aspects of the audit procedure.

In conclusion, this checklist is a good reminder for the auditor for most of the issues that he must verify. However, these issues are covered in a non-technical way, and this won't help the inexperienced auditor because the way to verify them is not given. It is not very organized either, and its size can be cumbersome.

Proposed checklist

The following checklist is intended to have a broad scope while maintaining a moderate size for practicability. It is mostly a adaptation of existing material that will be referred to whenever necessary. The objective of this checklist is to create a global view of a system's state. This way, following audits can be more focused in the sensitive issues uncovered by this checklist.

Instead of constructing here a single checklist of tests and verifications, a list of test areas will be presented, and for each of these areas a series of tests will be given. This way,

improvements can be easily added in each category without having to change others.

Environmental and physical security

The most basic but sometimes overlooked aspect of security. There is an email found on the internet from Bryn Robert Dole[15] that extensively covers most aspects of environmental and physical security and a few other off-topic security related stuff. First we will only look for environmental safety items, so we will reformat the list, clarify and add some points, and make it shorter without omitting anything important:

Environmental safety checklist (note: every test has an objective *yes* or *no* result):

Fire threats	
Smoke and heat detectors are in place and working	
The room where the equipment is installed is protected by fire containment doors	
Fire extinguishers near the system and in working order	
No smoking is allowed near the computer equipment	
Removable media (backup tapes, floppies) is protected by fireproof containers	
Lightning and electricity threats	
The computer equipment's electrical plug is properly grounded	
The computer equipment is connected to an Uninterruptible Power Supply (UPS)	
The computer room has an anti-static carpet	
Air environment threats	
The room is equipped with a temperature control (AC) system	
The room is equipped with a humidity control (AC) system	
The humidity/temperature (AC) system is separate for the equipment room	
There are alarms in place for humidity/temperature control equipment failures	
Air filters are in place in the equipment room	

[15] See http://www.cerias.purdue.edu/homes/spaf/CS690E/mail/msg00107.html

As the reader may note, an effort was made so that these tests were all objective in measurement and their verification procedure is implicitly present. They are in fact binary, meaning that the statement is true or false, there is no middle point, and the steps required to verify that the system is in spec are implicitly given.

We will now cover the active physical security issues. These are threats that require a malicious person to perform; they just won't happen without someone provoking them. Since we are talking about a filtering router on a campus environment, the risks in this area are somewhat low, however they exist. This list is also taken and adapted from Bryn Dole's email.

Physical security checklist:

	Intruder threats
There is staff actively controlling access to the computer room (subjective) Is someone always controlling badges or ID's of persons leaving/entering the room? Are those verifications adequate? Interview the person(s) involved in this area.	
When the facilities are abandoned after business hours, the protection level is adequate (subjective) This depends on the type of environment. Are we talking about a military facility? Or an academic facility? Visit the facilities off business hours. Interview surveillance personnel, if any. Verify that the minimum passive security measures are in place (the room is locked and the key is safely kept).	
Sensitive trash is safely disposed of (subjective) Again, what is sensitive trash? What is safely disposal? It depends on the type of information and environment. Interview system administrators, ask where are old backup tapes and paper documents are being disposed of, and if they could still contain sensitive information. Look for yourself in the field, if you alone could find any apparently important information lying around at arms reach.	
Media connectors and cabling (network and telephone) are inaccessible by unauthorized personnel (subjective) In some cases this is not a problem. In academic environments, a lot of people mess with network connectors in workstations and it is not a problem. However, critical network cabling such has an Ethernet cable to a firewall should never be tampered.	

The system's console security level is adequate (subjective) There are a lot of objective measurements for this test, but the auditor's experience and judgment is necessary to evaluate them in the global picture. Things to look for: Is the console password protected? If so, check if the password isn't in a piece of paper below the keyboard (very common place). Also, does the system allow booting from floppy disk, or cdrom? Is the BIOS information protected with a password? Does it have a factory backdoor?	

So the plot begins to thicken. While it is straight forward to verify that a system is well protected against environmental threats, when a malicious intruder enters the picture, some experience on the auditor's part is required to evaluate security.

To end this section, and since these checklists are not meant to be thorough in all scenarios, some extra resources to assist in audit environmental and physical security issues will be presented:

Introduction to TEMPEST (Cassi Goodman):

http://www.sans.org/infosecFAQ/encryption/TEMPEST.htm

Why Bother About BIOS Security? (Robert Allgeuer)

http://www.sans.org/infosecFAQ/authentic/BIOS.htm

Physical and Environmental Security Guidelines (Central Queensland University)

http://www.cqu.edu.au/documents/compsec/guidelines/cqu_sec40.html

Personnel security

Next we will attempt to evaluate personnel training and awareness. Still looking at Bryn Dole's mail, we will also look for issues in Management Analytics checklist [6] that could help us evaluate security on our particular system:

Personnel checklist:

The organization has a policy of punishing perpetrators to the fullest extent possible and employees are aware of this policy (subjective) Interview employees and management. Is there such a policy? How well is it implemented? Are system administrators reporting detected attacks to the authorities?	

Personnel and others within the organization who help defend systems from malicious threats are highly rewarded for their efforts. (subjective) What kind of rewards? In an academic environment, where students are administering the system, the learning and experience obtained by the opportunity to be able to help protect an important information system is reward enough. Is it? Interview the system administrators and see if the morale is high.	
Operating personnel is well trained in respect to security and aware of the risks (subjective) Does personnel visit security related sites frequently? Are they aware of possible security problems in their environment? A corporate network has different problems than an academic network, mainly due to the latter's exposure to the internet. Is operating personnel aware that the threats could also come from the inside? Do they think like an attacker and regularly test their own systems for vulnerabilities? Are they aware of "social engineering" attacks?	

Again, we see that issues directly related to personnel, such as surveillance guards, operating staff, etc. are always subjective. As well noted by Dan Strom in his assignment paper[16], "personnel interviews are subjective. The information received, although truthful, will be affected by the experiences of the interviewee".

Some resources will now be presented that address the issues related to interviewing personnel, and looking for personnel specific issues:

SANS Reading Room: Security Awareness:

http://www.sans.org/infosecFAQ/aware/aware_list.htm

SANS Reading Room: Social Engineering:

http://www.sans.org/infosecFAQ/social/social_list.htm

Information Security and Personnel Practices (Edward H. Freeman):

http://secinf.net/info/misc/handbook/687-693.html

Contingency issues

In this area, we should assume the worst and verify that system and staff are prepared for the worst case scenario. There are a lot of good references in this area also. The author will refrain to the essential in this area:

---

[16] Strom, Dan. *"Auditing the Netscreen-5 Firewall Used as a VPN Gateway"*

In the case of the system being shut down or malfunctioning, there are automated procedures to ensure the continuation of service Verification: if possible, shut down the system and try to reach the internal network from the internet. Note: This type of verification will probably not be welcomed by users and system administrators. It is however the most direct and objective way to verify if the system constitutes a single point of failure.	
If the above is false, verify that manual restoration of service is rapidly achieved (subjective) To a sufficiently large company, being disconnected from the internet can cost several thousand dollars a minute. So the auditor must take into consideration the type of organization.	
The correct operation of the system does not depend on external, less secure systems (subjective) If taken literally, this test will always fail, unless the system is self-sufficient. A computer system may be the most secure in the world, but if someone cuts the power supply to the computer room, it will go down, no matter how big is the UPS. What we are looking for is dependence on less secure and highly complex systems. An example is a firewall doing transparent proxy to a proxy server that is very badly configured. No matter how secure the firewall is, if the proxy goes down, all users will not be able to browse the web. It is the auditor's responsibility to evaluate how dependant is the system and recommend what can be done to decrease that dependency.	

Contingency planning and auditing resources:

Contingency Planning & Management Online:

http://www.contingencyplanning.com

The Oversight of Physical Security and Contingency Planning (Andy Krupa):

http://www.sans.org/infosecFAQ/recovery/oversight.htm

Operating system

This area is very well documented. The Linux operating system has several good security resources on the internet. For instance, Mary Laude has written an excellent checklist[17] for auditing RedHat 7.0 installations. On a more general note, there is Roger Retallack's

[17] Laude, Mary. *"Auditing RedHat Linux 7.0"*. URL: http://www.sans.org/y2k/practical/Mary_Laude_GSNA.zip

paper[18] on securing Linux Installations, also very educative.

While auditing creating a checklist for auditing the Linux operating system goes beyond the scope of this article, the author would like to point out some important issues, and some that are somewhat specific of Linux firewalls and this type of system is the object of the audit.

The checklist below is, in the author's opinion, the bare minimum that should be checked for this type of system. All tests are objective unless noted. Since we are reaching a more technical area, the suggested method for verifying these tests is stated below each one:

Operating system issues:

The latest stable version of the Linux kernel is installed and running Verification: enter the command `uname –a` in the system's console. The resulting output should match the latest stable version of the kernel, as shown in http://www.kernel.org. Note: it is important to be running stable software in mission critical systems, and the kernel is the most important and critical component of an operating system. As such, the most stable version is preferred to the one that has most features. Version 2.4.x has the features that are used in the firewall (iptables).	
The system's clock is synchronized to an external and reliable source Verification: A Network Time Protocol client/server program such as xntpd is installed and running correctly. Look for it in the running processes by typing `ps aux` in the console. Note: This issue is frequently overlooked, but when it is necessary to cross-check log entries with different sources it will prove very useful for system administrators.	
The operating system's files are being regularly checked by some form of file integrity checker software (subjective). Verification: The most used file integrity checker for Linux/Unix systems is tripwire. Unfortunately, there is no direct way to look for the program directly in the audited system, has the program may not even be installed locally on the machine for security reasons. The most objective way to verify this is to ask the system administrator there are digital signatures of the operating system's files (including configuration files) for regular integrity checking. One common file integrity technique is using rpm, but will not work for configuration files. If no file integrity checking technique is being used, this test fails.	

[18] Retallack, Roger. *"Securing Linux Installations"*. URL: http://www.sans.org/infosecFAQ/linux/sec_install.htm

The processes running in the machine are only the ones directly or indirectly involved in the functions that the system is assigned to Verification: enter the command `ps aux` in the system's console while logged in as root. For each process in the list, verify that it is needed directly or indirectly for the system to operate as expected. Use the `pstree` command as well to verify which processes are starting others and obtaining a more clear view of the system's process list. Also look at the system startup scripts to see if any other process is being started at system's startup. If possible, as a double-check, restart the system yourself and verify that only the needed processes are started. Note: This test is objective, but requires intimate knowledge of the policy and of the operating system's internals. Some processes may be needed to perform functions not mentioned directly in the policy.	
Only the essential programs and commands required to maintain the operation of the system according to policy are installed Verification: in an rpm based Linux system, execute `rpm –aq` and check for unneeded packages in the resulting list. Preferably, look for unneeded commands and programs in the system's directory tree. Note: the idea of this test is to verify that there isn't any unneeded program that, when ran, may be a source of vulnerabilities. Also, in case of a local user account being compromised, the attacker will take longer to do any harm to the system or network, if no compilation environment is installed, for instance. This test also requires deep knowledge of the Linux operating environment.	
All programs installed run with the principle of least privilege Verification: Use the command `ps aux` in the system's console. For each of the processes running, verify that the process has only the privilege that it needs. For instance, a web server doesn't need to be run as root, because after it is running, it usually only needs to read and serve files to the internet that are marked world readable in the file system. Also look for set user id or set group id commands, by entering the command `find / -perm +6000`. For each of the files listed, verify that the command really needs to run with different privileges than the calling user. As an example, in a system where only root is allowed to change passwords, the command `/usr/bin/passwd` does not need to be set user id root.	

All the TCP and UDP ports open on the system are only the strictly necessary for correct operation and maintenance. Verification: Use the command 'netstat –protocol=inet -l –n -p' as root in the system's console. This will show every "internet family" daemons listening on the machine and their respective PID/program name. An example interpretation of the output is given in the second part of this assignment. Also, for completeness, an external UDP and TCP port scan should be done using nmap on both network interfaces of the system. This can be accomplished by using nmap and scan from the outside internet and from the internal network. A complete example of this procedure is also shown in the second part of this assignment. The test fails if there is at least one open port that belongs to a service not being used, or unessential for correct operation and maintenance.	

To be able to maintain the checklist general enough to be used on all Linux installations, some details for the verification methods must be left for the auditor to research. The author believes that, together with analysis of the audit performed in the second part of this document, the reader will find these guidelines useful.

This checklist does not excuse the auditor of performing the additional tests that he finds necessary in this area. To that end, some tools and resources for auditing and securing Linux are now presented:

Linux Security:

http://www.linuxsecurity.com/

The Tripwire file integrity checker:

http://www.tripwire.com/

SANS Reading Room: Linux Issues:

http://www.sans.org/infosecFAQ/linux/linux_list.htm

Firewall and iptables specific issues

Here we will refer to possible issues of iptables configuration and installation. Prior to running the field tests, the auditor should ask management and system administrators what is the desired behavior of the firewall. If the auditor deems necessary, he will then recommend alterations to the expected behavior so that a more secure environment is achieved. After an agreement on what should be the desired firewall behavior he will then test and verify that the firewall is behaving like expected. This 'agreement' should be conducted in the entrance conference, in the second part of this assignment.

Firewall and iptables issues:

The iptables rule set conforms with the requirements in the security policy Verification: issue the command iptables --list -nv as root in the system's console. The resulting output should be the firewall rule listing. Also, because this particular system is doing network address translation, the NAT tables should also be inspected, with the command iptables --list –nv –t nat. Verify that every rule as a specific purpose stated in the security policy, and that there is no statement in the security policy without a corresponding rule. See the resources on the subject below, since this area can be quite tricky. The test passes if the filtering policy is covered in full by the rule set and there are no rules that have no corresponding entry in the filtering policy. See part II of this assignment for an example of how to conduct this test.	
Inbound and outbound TCP scan results verify that the firewall is actively filtering TCP traffic as specified in the security policy document. Verification: run TCP scans with the nmap port scanner on both sides of the network while tcpdump is listening on the other side and capturing TCP traffic. This test fails if any unexpected behavior (ie. undocumented in the security policy) occurs in network traffic, and passes otherwise. See part II of this assignment for an example of how to conduct this test.	
Inbound and outbound UDP scan results verify that the firewall is actively filtering UDP traffic as specified in the security policy document. Verification: It was observed that nmap's UDP port scan was not adequate for this particular test. It sent packets slowly and most of them out of order. It wouldn't stop sending packets after scanning ports 1-1024, and would start over again in reverse order, probably to account for UDP's non-reliability issue. Taken from nmap's man pages: > Unfortunately UDP scanning is sometimes painfully slow since most hosts implement a suggestion in RFC 1812 (section 4.3.2.8) of limiting the ICMP error message rate. For example, the Linux kernel (in net/ipv4/icmp.h) limits destination unreachable message generation to 80 per 4 seconds, with a 1/4 second penalty if that is exceeded. Solaris has much more strict limits (about 2 messages per second) and thus takes even longer to scan. nmap detects this rate limiting and slows down accordingly, rather than flood the network with useless packets that will be ignored by the target machine. This may be necessary for a good UDP port scan to occur, but it will not help the analysis of tcpdump results on the other side. There is no interest in scanning ports from a host; we are only verifying which packets traverse our firewall. As such, it is advisable to use a simple script with hping[19] to send a large number of UDP packets and verify what is being filtered using tcpdump. This test fails if any unexpected behavior (ie. undocumented in the security policy) occurs in network traffic, and passes otherwise. See part II of this assignment for an example of how to conduct this test.	

[19] See http://www.hping.org

Network testing tools verify that the system is filtering incoming/outgoing spoofed packets (packets from the internet pretending to originate from the internal network and vice-versa) Verification: Use the hping utility (hping -a <spoofed_address> <target>) to spoof packets and observe if they arrive at the target host. First, from a host located in the outside network, try to send a packet with a source address of a host located in the internal network to another host located in the internal network. Try different addresses, like the source address being the same as the target address (this also called a *land*[20] attack and may crash the target machine). Do the opposite to see if outbound spoofing is also permitted (or not). The test fails if spoofed packets successfully traverse the filtering router.	
Network testing tools verify that the system is filtering packets coming to/from illegal and broadcast addresses Verification: This test is very similar to the last, as it also can be done using hping and tcpdump. Simply change the spoofed address to an illegal or broadcast address and verify that the packets are not arriving at the target. Both directions (inward and outward) must be tested. The addresses are: • Any address starting or terminating in .0 or .255 • Any private address (from classes 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, or the loopback address range 127.0.0.0/8) The test fails if any of these packets is allowed thru the filtering router.	
The logging mechanisms in place allow the system administrator to be able to easily detect attacks and heavy traffic patterns (subjective) What can be easily detected by one person looking at the system's logs can pass undetected by an untrained individual looking at the same data; It's up to the auditor to test the system administrators' capabilities and controls in place, by aggressively attacking the system and network, and evaluating the resulting reactions.	

Resources:

Internet Firewalls: Frequently Asked Questions:

http://www.interhack.net/pubs/fwfaq/

The netfilter/iptables FAQ (Harald Welte):

http://netfilter.samba.org/netfilter-faq.html

---

[20] See http://www.cert.org/advisories/CA-1997-28.html

Rusty's Remarkably Unreliable Guides:

http://netfilter.samba.org/unreliable-guides/

Audit procedure

The Entrance Conference

This first step is where the auditor gets written permission for the audit and discusses security policy issues. Since there was no such document, it was agreed that this was the time to create one. A very crude security policy document was agreed on, so that the audit could proceed. Most of this material is taken from the SANS Firewall checklist [12]:

- Silently drop any type of traffic coming from/to broadcast and unroutable addresses[21], like 255.255.255.255 and 0.0.0.0, passing through the firewall (not just ICMP packets).

- All packets coming from/to internal addresses[22] (such as 192.168.0.0) or the loopback address (127.0.0.1) should be silently dropped, if coming from external network interfaces (not the loopback interface).

- Block IP spoofing attempts from both sides of the network; packets coming from the external network pretending to originate from the internal network and vice-versa should be silently dropped.

- HTTP requests coming from the internal network should be transparently routed to a proxy server located on the same machine running the firewall.

- TCP packets traversing the system in any direction destined to ports 87, 111, 512, 513, 514, 515 or 540 should be actively rejected.

- UDP packets traversing the system in any direction should pass, unless destined to ports 69, 111, 161, 162 or 2049, which belong to tftp, sunrpc, snmp, snmptrap and nfs, respectively. These packets should be silently dropped.

- Minimize the visibility of the machine, by silently dropping any packets addressed to it that do not come from the system administrator's systems located in the internal network.

---

[21] RFC 1812. *"Requirements for IP Version 4 Routers"*. Section 4.2.3.1 *"IP Broadcast Addresses"*. URL: http://www.ietf.org/rfc/rfc1812.txt

[22] RFC 1918. *"Address Allocation for Private Internets"*. URL: http://www.ietf.org/rfc/rfc1918.txt

- If not specified above, all traffic should pass through the machine.
- The network services that the machine should be running are:
    - SSH or another remote secure administration service.
    - A proxy server for the transparent routing of web requests.
    - Any other service strictly required for managing and monitoring the system, such as routing services, file integrity checkers, network statistics services, intrusion detection systems, etc.

After coming to terms with the desired behavior of the system, considerations on the scope of the audit were also addressed, and management manifested the need for a broad scope, as no audit had been conducted before, and management's knowledge of the system from a security standpoint was scarce or even nonexistent.

The Audit

We will now go over all the items in every checklist in order and determine if the tests pass or fail.

For each objective test that requires some work to perform, the auditor will recreate all steps performed to verify that the system conformed to spec (or not).

For each subjective test, the auditor will write down all of the criteria that he used in passing or failing the test, and record all the material (interviews, descriptions, etc) that were used to ponder his decision.

Whenever he deems necessary, the auditor should include a short recommendation on how to improve certain aspects of the system being audited.

At the end each checklist, the auditor should write down his general impression of the area being audited, give out a short evaluation of the checklist, and write down any tests that he felt necessary to perform but were not in the checklist. This will allow the continuous improvement of the checklist and the general audit process.

Environmental safety checklist (note: every test has an objective *yes* or *no* result):

Fire threats

Smoke and heat detectors are in place and working Auditor: There were smoke and heat detectors in place in the room were the system was installed, but they were not tested manually by the auditor. Surveillance staff said they were tested when installed, so the auditor assumes they were working.	PASS
The room where the equipment is installed is protected by fire containment doors Auditor: No fire containment doors exist in the room. The building, however, is built with stone materials and fire propagation should not be rapid.	FAIL
Fire extinguishers near the system and in working order Auditor: Again, fire extinguishers were found and seemed to be in working order but were not manually tested by the auditor.	PASS
No smoking is allowed near the computer equipment Auditor: A 'no smoking policy' is enforced in the main server room. Some system administrators break that policy, but rarely and carefully.	PASS
Removable media (backup tapes, floppies) is protected by fireproof containers Auditor: There is no backup policy for the system being audited, so no removable media exists. The lack of an audit policy is a serious risk and will be evaluated later in the audit.	FAIL
Lightning and electricity threats	
The computer equipment's electrical plug is properly grounded	PASS
The computer equipment is connected to an Uninterruptible Power Supply (UPS) Auditor: The entire server room is connected to an UPS, however the batteries are faulty and need to be replaced (this was reported by a system administrator about one month before the audit).	FAIL

The computer room has an anti-static carpet Auditor: The computer room is equipped with anti-static carpet and a false floor where cabling is laid through. Below the false floor a layer of anti-static paint is applied over the cement. The auditor directly verified the presence of the anti-static carpet, and was informed about the painting by a system administrator.	PASS
Air environment threats	
The room is equipped with a temperature control (AC) system	PASS
The room is equipped with a humidity control (AC) system	PASS
The humidity/temperature (AC) system is separate for the equipment room Auditor: There are three different Air Conditioner systems in place in the computer room, and all three are dedicated to ensure that all equipment functions properly	PASS
There are alarms in place for humidity/temperature control equipment failures Auditor: The presence of malfunction alarms in the AC systems could not be directly verified. Also, no one that was interviewed was aware of such alarms, so they are assumed not to exist. (Note: this fail is minimized by the presence of three different AC systems and smoke and heat detectors. It remains an issue not to be disregarded, however).	FAIL
Air filters are in place in the equipment room Auditor: It was directly verified that the installed AC systems all had working air filters.	PASS

Auditor's notes: The environmental conditions were deemed adequate for the system. A major problem remains the fact that the Uninterruptible Power Supply system is not in working order and presents a serious risk.

This checklist was found thorough enough for the requisites of this audit, and no other issues needed to be investigated in this area.

The lack of any type of removable media identifies a problem that will have to be further researched (lack of backup policy), but not in this area.

Intruder threats	
There is staff actively controlling access to the computer room (subjective) Auditor: There is always a member of operating staff in the room every working day from 10:00 am to 10:00 pm, unless there is someone on vacation. During that period, only someone identified as a member of the operating staff is allowed inside. There are exceptions, but only when strictly necessary, such as the auditor's case, but the auditor was always accompanied by a member of operations. This was deemed appropriate for the type of organization, and this test has passed.	**PASS**
When the facilities are abandoned after business hours, the protection level is adequate (subjective) Auditor: a visit to the facilities late at night shows that there is always at least one security guard near the building in where the computer room is situated. There are no electronic surveillance measures such as motion sensors or security cameras inside the building. However, the computer room is protected by a secure lock and only the head of security personnel has the key, along with another five members of the operating staff. This was told to the auditor directly by the head of security, and verified by a system operator. For an academic environment, the protection level of the computer room is deemed barely adequate, so the test passes. However, at least one electronic surveillance mechanism (an alarm or camera system) is advised to avoid unnecessary risks.	**PASS**
Sensitive trash is safely disposed of (subjective) Auditor: a visit to the offices of operating staff did not show a general practice of safely disposal of sensitive information. There were exceptions and some operators even had and used shredding machines, but the auditor believes that the general practice was not adequate. While no sensitive information about the system being audited was found, several student's user accounts for the campus main server where found in the offices of operating staff, which gives the auditor a negative impression, so this test fails.	**FAIL**
Media connectors and cabling (network and telephone) are inaccessible by unauthorized personnel (subjective) Auditor: This system had a critical role in maintaining network operations in the organization, and so the media connectors leading to it should be protected. The system was connected to a border router and to an ATM switch for the internal network. Both of these machines were in the same secure perimeter as the system being audited, so the media connectors were as secure as the machine itself. As so, the auditor considers that this test passes.	**PASS**

The system's console security level is adequate (subjective)	FAIL

Auditor: The console of the system being tested had little to no security measures implemented. While there was no user password lying around in paper and there was no open user account (and according to operators, they log off every time they use the console), the kernel boot loader that was installed, `lilo`, had no password set. Also, a system administrator interviewed by the auditor informed that the BIOS was found to be unprotected by a password so that booting from a floppy disk or password was trivial.

Manually rebooting the system and attempt to gain super-user privileges using these vulnerabilities was not possible by the auditor because the system was operating and could not be shut down. However, a non intrusive way to check for a password in the `lilo` configuration would be typing the following command as root in the system's console:

```
cat /etc/lilo.conf
```

This was the recorded output:

```
# LILO configuration file
# Start LILO global Section
# If you want to prevent console users to boot with
# init=/bin/bash, restrict usage of boot params by setting
# a passwd and using the option
#  restricted.
#password=bootpwd
#restricted
append="vga=0x0303"
boot=/dev/hda
#compact        # faster, but won't work on all systems.
vga = normal    # force sane state
message=/boot/message
menu-scheme=Wg:kw:Wg:Wg
read-only
prompt
timeout=50
# End LILO global Section
#
image = /boot/vmlinuz
  root = /dev/hda3
  label = Linux
  initrd = /boot/initrd
# end of lilo config file
```

As the reader may note, the password line is commented out. So it is possible to gain root privileges by rebooting the system and type the following in the lilo prompt:

```
Linux init=/bin/bash
```

This was found to be an unnecessary risk. The test was failed by the auditor.

Auditor's notes: Overall, the system is fairly secure in respect to physical intruder threats. Recommendations include acquiring at least one electronic surveillance system to protect the physical perimeter, and increasing the security of the system's console, by protecting the BIOS with a password and following the recommendations on the `lilo.conf` file, regarding passwords.

Regarding completeness, the auditor doesn't have any more tests to add, in this particular audit. However, when doing future audits to this system, this checklist should contain

more binary tests that serve as a verification for the problems found, such like:

- The lilo boot loader is protected by a password protecting against local unauthorized users gaining access to the system (objective)
- The BIOS of the computer is protected by a password and does not allow booting from external media (such as a cdrom or floppy) (objective)
- There is at least one electronic surveillance measure such as a camera or alarm protecting access to the computer room. (objective)

This way, in future audits, these potential problems will rapidly be tested for correctness and the auditor's expertise in finding them through subjective testing will not be required.

Contingency issues:

In the case of the system being shut down or malfunctioning, there are automated procedures to ensure the continuation of service Auditor: This test failed disastrously. While the actual simulation of malfunction was not performed (it wasn't needed), the fact that the system is not redundant is enough to realize the issue. There are no automated procedures in the system or outside of it to allow continuation of service. This system is, in effect, a single point of failure for the entire campus's network. The auditor recommends the administration to allow budget for the installation of some form of redundancy to this system.	**FAIL**
If the above is false, verify that manual restoration of service is rapidly achieved (subjective) Auditor: Interviews with system administrators show that there is no plan for a manual restoration of service. This is a matter that was not even discussed or taken seriously. So if the system goes down off business hours, the down time is unpredictable. In the event of a hard drive failure, the system must be installed again manually, because there is no backup policy in effect. This is a serious issue that must be rapidly looked into. The auditor recommends the urgent elaboration and implementation of a backup policy for the system, and that a replica of the system (or parts of it, such as a second hard drive) is created for the rapid restoration of service.	**FAIL**

The correct operation of the system does not depend on external, less secure systems (subjective) Auditor: All the systems that this system is dependant on are in the same secure perimeter. The issue of the UPS not working is a separate issue that was already addressed, so it won't be taken into account here. Interviews have shown that there was an issue but it was taken care of; HTTP traffic was being redirected to a proxy server outside the secure perimeter; when that server was down, the whole campus was deprived from browsing the web; however, a proxy server was started in the local system, that now uses the outer proxy has a web cache parent. This way a certain degree of independence can be established, and the failure of external systems will not cause a malfunction in the local system.	**PASS**

Auditor's notes: While the system is fairly independent of outside failures, there is absolutely no contingency plan deployed. This is a serious problem that should be looked at urgently. The auditor suggests that system administrators and management look at the resources given in part I of this assignment and rapidly deploy a contingency solution for the system.

When that plan is in effect, the auditor recommends the conduction of a second audit to evaluate those contingency plans.

Operating system issues:

The latest stable version of the Linux kernel is installed and running Auditor: the result of the command uname –a is as follows: `Linux gatekeeper 2.4.12 #1 Fri Oct 12 19:09:26 WEST 2001 i686 unknown` At this date (4 Nov. 2001), the page at http://www.kernel.org is announcing the latest stable version of the Linux kernel to be version 2.4.13, instead of the installed 2.4.12. While this is a minor version update, some security issues might have been found and solved in this latest version. This test fails. The auditor recommends that the latest version is installed as soon as possible. A quick look at the latest version's Changelog[23] shows that the modifications were mostly driver updates, but there are fixes for some issues that could cause the system to crash or run unreliably.	**FAIL**

[23] See http://www.kernel.org/pub/linux/kernel/v2.4/ChangeLog-2.4.13

The system's clock is synchronized to an external and reliable source
 	
Auditor: The result of the command ps aux shows xntpd running:
 	
`# ps aux | grep xntpd`
resulting output:
`root 17432 0.0 0.7 3952 3944 ? SL Oct16 0:00 /usr/sbin/xntpd`
`root 17433 0.0 0.7 3952 3944 ? SL Oct16 0:00 /usr/sbin/xntpd`
`root 17434 0.0 0.7 3952 3944 ? SL Oct16 0:01 /usr/sbin/xntpd`
`luis 29026 0.0 0.1 1540 568 pts/2 S 00:37 0:00 grep xntpd`
 	
There are three processes of the `xntpd` daemon running. The clock is in fact being synchronized to a remote time server used by most machines in the campus, as it can be observed in the configuration files. So the test passes.
 	
While beyond the scope of this test, the auditor recommends reading "A Security Analysis of the NTP Protocol", a paper by Matt Bishop[24].
	PASS

The operating system's files are being regularly checked by some form of file integrity checker software. Auditor: A system administrator interviewed by the auditor, informed that there was no file integrity checking controls in place. He was one of the involved persons in installing the operating system, so the information is deemed worthy. The interviewee admitted the risk and manifested the will of implementing such controls. However, the test has failed for now.	**FAIL**

[24] Bishop, Matt *"A Security Analysis of the NTP Protocol"*. URL: http://nob.cs.ucdavis.edu/~bishop/papers/Pdf/ntpsec.pdf

[25] See http://www.postfix.org/docs.html

The processes running in the machine are only the ones directly or indirectly involved in the functions that the system is assigned to Auditor: The commands `ps aux` and `pstree` were executed and its full output is shown in Appendix B. The following was found of worthy of note by the auditor: • The `master` process (PID 379) is used by the postfix mail transfer agent. According to its documentation[25], this process must be running for the correct delivery of mail by local accounts. This was found necessary, because the system sends automated mails from the `root` user with status information of the machine to system administrators. • The `nscd` process is the name server cache daemon. It is used by the system to cache domain name server replies. It was found necessary. • `squid, snort, snmpd, ospfd, bgpd, xntpd` and `rtmon` were all services needed for the correct operation of the machine, according to system administration. • `gpm` and `screen` were not needed for the correct operation of the system. One was the mouse daemon for the console; the other was a program that allowed a user terminal session left open earlier to be restored. However this was not found to be a risk factor by the auditor. • All other processes are services needed to the correct operation of the system. Despite `gpm` and a `screen` session running and it is recommended that these processes are terminated, the auditor passes this test with a fairly high degree of confidence.	**PASS**
Only the essential programs and commands required to maintain the operation of the system according to policy are installed Auditor: the auditor has executed the `rpm -aq` command on the system and the full output can be observed in Appendix B. A careful analysis of the list shows that a development environment is set on the machine; It has a C/C++ compiler (cpp-2.95.3-52, gcc-2.95.3-52, glibc-devel-2.2.2-38), which can be of help to an attacker that gains local access to the machine. While these may be useful for system administrators, it is advised that these packages are removed. Interviews with the system administrators made the auditor believe that this problem will be addressed. However, this test failed at this time.	**FAIL**

[25] See http://www.postfix.org/docs.html

All programs installed run with the principle of least privilege Auditor: Looking at the output of the `ps aux` command, the auditor couldn't find any process that was running with unnecessary privileges. However, the listing provided by the command `find / -perm +6000` shown in full in Appendix B, shows that some programs can be run with unneeded privileges according to the machine's policy; It is recommended that at least the following programs are stripped from their elevated run-time privileges: mount, umount, chfn, chsh, crontab, at, passwd, gpasswd. Until then, the auditor fails this test.	**FAIL**

All the TCP and UDP ports open on the system are only the strictly necessary for correct operation and maintenance.

PASS

Auditor: running the command 'netstat --protocol=inet -l –n -p' produced the following result:

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address      Foreign Address   State   PID/Program
name
tcp        0      0 0.0.0.0:2601       0.0.0.0:*         LISTEN  18510/zebra
tcp        0      0 0.0.0.0:2604       0.0.0.0:*         LISTEN  18519/ospfd
tcp        0      0 0.0.0.0:2605       0.0.0.0:*         LISTEN  18528/bgpd
tcp        0      0 0.0.0.0:179        0.0.0.0:*         LISTEN  18528/bgpd
tcp        0      0 0.0.0.0:3128       0.0.0.0:*         LISTEN  648/(squid)
tcp        0      0 0.0.0.0:22         0.0.0.0:*         LISTEN  357/sshd
udp        0      0 0.0.0.0:32776      0.0.0.0:*                 648/(squid)
udp        0      0 0.0.0.0:161        0.0.0.0:*                 327/snmpd
udp        0      0 0.0.0.0:3130       0.0.0.0:*                 648/(squid)
udp        0      0 123.234.134.45:123 0.0.0.0:*                 544/xntpd
udp        0      0 192.168.253.2:123  0.0.0.0:*                 544/xntpd
udp        0      0 127.0.0.1:123      0.0.0.0:*                 544/xntpd
udp        0      0 0.0.0.0:123        0.0.0.0:*                 544/xntpd
raw        0      0 0.0.0.0:89         0.0.0.0:*         7       18519/ospfd
```

The first column is the protocol the daemon is using; in TCP it appears that there are six open ports: 2601, 2604, 2605, 179, 3128 and 22. The first four are routing services that need to be running, according to system administrators. Port 3128 belongs to the local proxy server used for transparent proxying and also needs to be running. Port 22 is the SSH server and is necessary for system administration.

As for UDP, the ports open, the only services using open ports were squid, xntpd and snmpd. Squid uses UDP for communicating with other proxies, so this was found to be correct by the auditor. snmpd was required for generating network statistics and xntpd was required to synchronize the clock to an external clock source.

```
# nmap -sS 192.168.253.2

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.253.2):
(The 1516 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
179/tcp    open        bgp
2601/tcp   open        zebra
2604/tcp   open        ospfd
2605/tcp   open        bgpd
3128/tcp   open        squid-http
Nmap run completed -- 1 IP address (1 host up) scanned in 2 seconds
```

This output is a TCP scan to the inside interface of the filtering router; it is somewhat consistent with netstat's results. Port 80 can be easily explained by the transparent proxy function, it is not an actual open port.

```
# nmap -sS 123.234.134.45

Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
Interesting ports on  (123.234.134.45):
(The 1541 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
179/tcp    open        bgp
2601/tcp   open        zebra
2604/tcp   open        ospfd
2605/tcp   open        bgpd
3128/tcp   open        squid-http
Nmap run completed -- 1 IP address (1 host up) scanned in 94 seconds
```

This is a scan from the outside internet to the outside interface of the filtering router. Everything matches the inside scan except for port 80, which means that the transparent proxy is not functioning on the outside interface, which is correct.

Auditor's notes: Despite failing 4 out of 7 tests, the operating system was found to be moderately secure. The most severe problem found by the auditor is the lack of a file integrity checking control.

This checklist was found to be incomplete by the auditor in this area. For example, it was not verified that the latest version of the critical services was being used. For the trust level to be increased with regard to the operating system, additional steps must be performed. It is advisable to perform a more thorough audit with existing Internet resources on the Linux operating system (see first part of this assignment).

Firewall and iptables issues:

---

[26] RFC 1812. *"Requirements for IP Version 4 Routers"*. Section 4.2.3.1 *"IP Broadcast Addresses"*. URL: http://www.ietf.org/rfc/rfc1812.txt

[27] RFC 1918. *"Address Allocation for Private Internets"*. URL: http://www.ietf.org/rfc/rfc1918.txt

The iptables rule set conforms with the requirements in the security policy	FAIL

Auditor: The execution of the `iptables -list -nv` and `iptables -list -nv -t nat` commands presented a list of firewall rules in effect for this system, shown in full in Appendix C. For convenience, here is the firewall policy that was agreed to be enforced, accompanied by the result of the evaluation:

- Silently drop any type of traffic coming from/to broadcast and unroutable addresses[26], like 255.255.255.255 and 0.0.0.0, passing through the firewall (not just ICMP packets). **Correct** – The first four rules in the FORWARD table cover all of this issue; it took the auditor some to figure out the reversed netmask (0.0.0.255) in the rules, causing all packets going through the firewall with the destination or source IP ending in .255 or .0 addresses to be silently dropped.
- All packets coming from/to internal addresses[27] (such as 192.168.0.0) or the loopback address (127.0.0.1) should be silently dropped, if coming from external network interfaces (not the loopback interface). **Correct –** rules 6 to 13 of the FORWARD chain cover this issue entirely, silently dropping the specified packets.
- Block IP spoofing attempts from both sides of the network; packets coming from the external network pretending to originate from the internal network and vice-versa should be silently dropped. **Missing –** the rules that address this issue were not found by the auditor in the rule set.
- HTTP requests coming from the internal network should be transparently routed to a proxy server located on the same machine running the firewall. **Correct –** TCP packets coming from the internal network destined to port 80 of an external server are matched in rule 5 of the PREROUTING table; after that they are sent to a table called HTTP-OUT, and that table will filter for sites that require a direct connection to the user's browser; if a destination IP is matched, the user will connect directly to the site; otherwise the last rule of the HTTP-OUT table instructs iptables to redirect all traffic to the local system's proxy (port 3128). This description is somewhat vague, and the reader should consult the provided iptables resource list in part I of this assignment.
- Minimize the visibility of the machine, by silently dropping any packets addressed to it that do not come from the system administrator's systems located in the internal network. **Missing –** The default policy in the INPUT table (the table that handles packets destined to the local system) was ACCEPT, meaning that by default it will accept connections from everywhere. This is not the desired policy and should be changed.
- If not specified above, all traffic should pass through the machine. **Correct –** The default policy in the FORWARD (the table that handles packets passing thru the local system) is ACCEPT, which allows all traffic not specified in the rule set to traverse the system freely. **However** there are firewall rules that are not required for the compliance of this policy and so it must be assumed that either this policy is incomplete, or the rules were incorrectly set up.

Evaluating the results of going through the policy, it was verified that there are items in the policy that are incomplete or incorrectly implemented in the firewall rule set. Also, there are rules in the firewall that do not have a corresponding policy.

For those reasons, this test is not passed until the identified issues are taken care of.

http://www.ietf.org/rfc/rfc1812.txt

[27] RFC 1918. *"Address Allocation for Private Internets"*. URL: http://www.ietf.org/rfc/rfc1918.txt

Inbound and outbound TCP scan results verify that the firewall is actively filtering TCP traffic as specified in the security policy document.

P A S S

Auditor: from a host in the **inside** of the network, nmap was ran against a host on the outside internet with the following command:

nmap -P0 -sS -r -g 1 -p 1-1024 <target ip>

The first option, '-P0' tells nmap not to send an ICMP echo request to see if the host is active. There was no need for that, because it was known beforehand that the host was in fact active and running tcpdump.

The next option, '-sS' tells nmap to 'syn scan' the host, that is, only send one SYN packet per destination port and look at the result. This was chosen to cut down on tcpdump output, avoiding full connection start/termination traffic.

The next option, '-r' tells nmap not to randomize the order of the ports to scan, so ports are scanned from lowest to highest. This is done to simplify tcpdump analysis.

The next option, '-g 1' tells nmap to keep the source port fixed at 1; this was also done to simplify tcpdump analysis and to quickly identify probe packets.

The next option, '-p 1-1024' tells nmap to scan destination ports from 1 to 1024. These are priveleged ports, and the auditor chose not to scan higher numbered ports because there was no need to filter those ports at the firewall, so they were not verified.

The full output of the nmap command is as follows:

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (213.22.64.248):
(The 1023 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
Nmap run completed -- 1 IP address (1 host up) scanned in 10 seconds
```

This output does not tell much, only that port 22 and 80 are open in the target host. It does not show filtered ports. However a closer look at the tcpdump log taken from the target host (shown in appendix D.1) shows that packets targeted at ports 87, 111, 512 to 515 and 540 are not arriving at the target. So why doesn't nmap report that these ports are filtered? Because the firewall is replying with a RST pretending to come from the target host, making nmap believe that the target has a closed port.

Also, the auditor knew beforehand that port 80 was not opened in the target host, and a quick check at the tcpdump logs shows that the corresponding SYN packet did not arrive at its destination. This verifies that the configuration for transparent proxying exists.

The next scan is from the **outside** internet to a host in the internal network.

nmap was used with the same parameters, and its output is as follows:

```
Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
Interesting ports on  (123.234.164.4):
(The 1021 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 10 seconds
```

Once again the output from nmap is not helpful, but looking at the tcpdump output, shown in appendix D.2, we see that the same ports are being filtered and actively rejected by the firewall: 87, 111, 512 to 515 and 540.

Both inbound and outbound tests were ran several time to reduce the possibility of erroneous results due to packet loss.

Inbound and outbound UDP scan results verify that the firewall is actively filtering UDP traffic as specified in the security policy document.

PASS

Auditor: hping was used instead of nmap, as mentioned in the original checklist. The actual command to send the packets was the following:

```
perl -e 'for($i=1; $i<=1024; $i++) { \
system ("hping -c 1 -2 -s 1 –k 123.234.164.2 -p $i &"); \
sleep 1; }'
```

This is a simple PERL script that calls hping for ports 1-1024 and then sleeps for one second, to try and keep the packets arriving in order. The parameters for hping were the following:

Option '-c 1' makes sure that only one packet is sent each time hping is executed.

Option '-2' turns on UDP mode.

Option '-s 1 -k' sends the packet with source port 1 and tells hping not to increase it, just to make sure.

123.234.164.2 is the destination IP for the packets.

Option '-p $i' is the destination port for the packets as given by the PERL interpreter.

The tests were ran for inbound and outbound traffic, using an external host targeted at an inside host in the first case and the opposite in the latter. Both these tests were ran for port ranges 1-1024 and 2045-2054 (this last range to verify filtering of the NFS service, as described in the security policy).

Both inbound and outbound tests were run several times to reduce the possibility of erroneous results due to packet loss.

The full output of both tests is shown in appendixes D.3 and D.4 of this document.

By looking at the full output of tcpdump, it became obvious that the same rules were being applied to both inbound and outbound packets, and that the ports being filtered were 69, 111, 161, 162 and 2049, that are usually reserved for tftp, sunrpc, snmp, snmptrap and nfs services, respectively.

This test passes since the behavior encountered was as documented in the security policy draft.

Network testing tools verify that the system is filtering incoming/outgoing spoofed packets (packets from the internet pretending to originate from the internal network and vice-versa) Auditor: from a host in the outside internet, hping was used to send spoofed packets to a host in the internal network: the command was: hping -a 123.234.164.1 123.234.164.2 No flags were set, so hping sent a TCP packet to 123.234.164.2 with no TCP flags on, pretending to come from 123.234.164.1. The output of tcpdump running on host 123.234.164.2 was the following: # tcpdump -nn -i eth0 tcp Kernel filter, protocol ALL, datagram packet socket tcpdump: listening on eth0 05:51:27.184127 123.234.164.1.2603 > 123.234.164.2.0: . 1875648349:1875648349(0) win 512 05:51:27.184174 123.234.164.2.0 > 123.234.164.1.2603: R 0:0(0) ack 1875648349 win 0 (DF) 05:51:28.189425 123.234.164.1.2604 > 123.234.164.2.0: . 397790507:397790507(0) win 512 05:51:28.189453 123.234.164.2.0 > 123.234.164.1.2604: R 0:0(0) ack 397790507 win 0 (DF) 05:51:29.182785 123.234.164.1.2605 > 123.234.164.2.0: . 1773333266:1773333266(0) win 512 05:51:29.182815 123.234.164.2.0 > 123.234.164.1.2605: R 0:0(0) ack 1773333266 win 0 (DF) 6 packets received by filter This clearly shows that the packet is traversing the filtering router and entering the internal network. Since these are unsolicited packets, the target host is simply replying with RST packets to innocent host 123.234.164.1 that did not originate any traffic. This could translate into a major security problem, and is not what was expected and documented in the security policy document. The hping command was also used from host 123.234.164.2 to send spoofed packets to a host in the external internet. The command was: hping -a 123.234.134.46 213.22.64.248 123.234.134.46 is the border router, which means that it is already an IP from the internet, from the filtering router's perspective (see network topology diagram on the first part of this document). This IP was chosen for spoofing because it makes it impossible that other routers with anti-spoofing rules detect this as a spoofed packet. tcpdump output was similar on host 213.22.64.248: # tcpdump -nn -i eth1 tcp tcpdump: listening on eth1 05:54:43.705129 123.234.134.46.2689 > 213.22.64.248.0: . win 512 05:54:43.705231 213.22.64.248.0 > 123.234.134.46.2689: R 0:0(0) ack 386778672 win 0 (DF) 05:54:44.741499 123.234.134.46.2690 > 213.22.64.248.0: . win 512 05:54:44.741582 213.22.64.248.0 > 123.234.134.46.2690: R 0:0(0) ack 440224413 win 0 (DF) 05:54:45.812781 123.234.134.46.2691 > 213.22.64.248.0: . win 512 05:54:45.812869 213.22.64.248.0 > 123.234.134.46.2691: R 0:0(0) ack 2117677689 win 0 (DF) 6 packets received by filter 0 packets dropped by kernel This means that our filtering router is not configured to block spoofed packets, and that this test positively fails.	**F A I L**

Network testing tools verify that the system is filtering packets coming to/from illegal and broadcast addresses Auditor: The hping command was also used in this test. All the commands executed are listed below and none of them produced a packet arrival on inward and outward tests. Outward testing: **hping -a 123.234.164.0 213.22.64.248** **hping -a 123.234.164.255 213.22.64.248** **hping –a 0.136.164.1 213.22.64.248** **hping –a 255.136.164.1 213.22.64.248** **hping –a 127.0.0.1 213.22.64.248** **hping –a 192.168.1.1 213.22.64.248** **hping –a 172.16.1.2 213.22.64.248** **hping –a 10.1.2.3 213.22.64.248** Inward testing was similar except that the destination IP and source of the scans was different; an external host was used to send packets while an external host was running tcpdump to look for packets. The auditor passes this test since none of the commands listed produced any traffic on the hosts with tcpdump. This means that the filtering router is probably doing what it should.	**PASS**
The logging mechanisms in place allow the system administrator to be able to easily detect attacks and heavy traffic patterns (subjective) Auditor: A search in the running process list of the system shows that the snort intrusion detection system is running and logging attacks. However, doing a full ports scan using nmap did not trigger any reactions in the system administration staff. After interviewing them, they denied knowing about the simulated attack and argued that snort was for long-term logging purposes only. The auditor can only fail this test. The academic setting of this system does not permit that attack detection controls are taken lightly. While the auditor understands that it would require a much greater amount of human resources than the currently available to be able to efficiently monitor any attack attempt, this issue will not be resolved unless that happens.	**FAIL**

Auditor's notes: The system was moderately configured, but the failure to inspect intrusion logs is a serious concern, especially because the open nature of this network environment. Also, the lack of spoofing filtering is a significant security risk and should be taken care of immediately.

Evaluation of the audit

Evaluation of the audit procedure shows that this audit was useful in providing a global overview on the current system state, because of the broad scope of issues it addresses. However, and because of that fact, one could never go into too much detail without creating a checklist that would consume an enormous amount of time and human resources to go through.

This kind of broad audit is also useful for management; most of the time management is completely unaware of the broad spectrum of vulnerabilities that affect a system. This type of audit helps to raise awareness and motivation in the higher levels of management so that budget is put where it should be.

Also, the scope of the audit helps to keep system administrators looking in every direction, instead of focusing on only one aspect of security, like a very efficient and secure firewall rule set, but forgetting to lock their critical system's BIOS, or even locking the computer room on their way out. Also, planning and creating policies for these issues is vital for the system administrator, and this audit procedure attempts to raise awareness in that area.

For the auditor, this procedure reminds him that a checklist is not a static document; he should always try and contribute to the improvement of the checklist, by creating objective testing while trying to keep the checklist modular enough so that other auditors could benefit from the material. The next time the auditor will perform the tests to the system, he will benefit from his improvements, and quite possibly, so will the audit.

Since the scope of the audit is broad, thoroughness suffers slightly and that must be considered by management. However, by raising awareness, the auditor expects to be able to conduct more specific audits in the future.

Subjective testing will always be a necessary and very important part of an audit procedure, however; especially when we are talking of a broad scope audit like this one, where many areas are covered. Breaking down a subjective test into smaller objective tests would create a very large checklist and possibly create gaps in the methodology, where some issue will not be covered just because it isn't mentioned in the checklist. As an example: this test sentence "The system's console security level is adequate" is subjective, the auditor must make decisions and employ criteria when deciding about this; but how many objective test sentences would be necessary to replace this test? We would have to create tests for every kind of possible system environment, and that would create a very large and cumbersome checklist. So the auditor's experience is his ultimate weapon, not his checklist. A checklist is only the auditor's next best weapon.

Strict objective testing is impracticable. You can never take the auditor off the audit.

Directions for future work

In general, providing more resources to the checklist and more methodology on how to perform each test will give the technically insecure auditor a good "safety net" if he is caught unprepared. So this checklist presents itself as an eternal 'work in progress', where it is each auditor's duty to contribute with his own knowledge where he believes that it will simplify testing procedures. He should also document all the extra tests that he

performed that were not already in the checklist.

Firewall testing procedures

While planning and conducting the audit for this assignment, it became apparent to the author that the firewall rule set auditing and subsequent network testing was the area where more time was expended and more expertise was required.

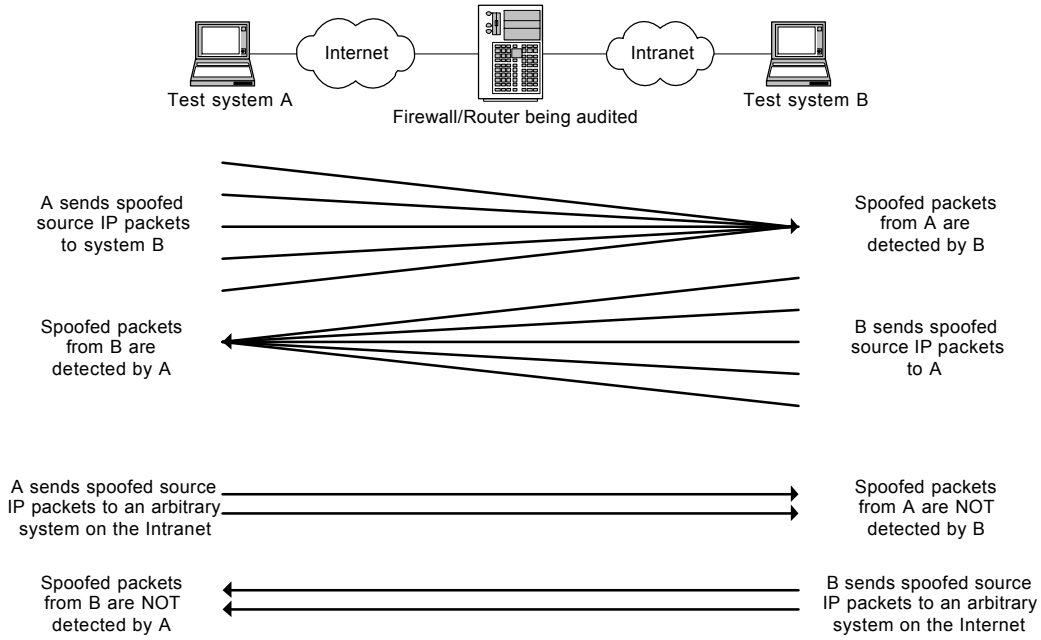
This is perhaps the field where automation seems feasible, but existing tools for this purpose are scarce, and have not yet matured for widespread use. It is interesting to note that the vast majority of automated security testing software, like the Nessus[28] security scanner, is more oriented towards host security than network security.

During the tests performed, the system was still connected and forwarding packets. This could lead to outside inadvertent (or not) tampering. The system should be tested isolated from any other host other than the test platform, so tests results are coherent and undisturbed by other traffic. Also, it is frequent that the systems used for testing are not adjacent to the system being tested. This can lead to erroneous results because network traffic can be filtered or otherwise modified during transit to the test system.

Another shortcoming of the method of testing currently used is that, while it is possible to spoof source IP addresses and verify that they are being forwarded (or not) to the destination by the system, it is not possible to send packets with a destination IP address of a system currently being used and observe the results. This means that complete rule set auditing is simply not possible, in this sort of environment:

[28] http://www.nessus.org

The impossibility of thorough testing on regular environments



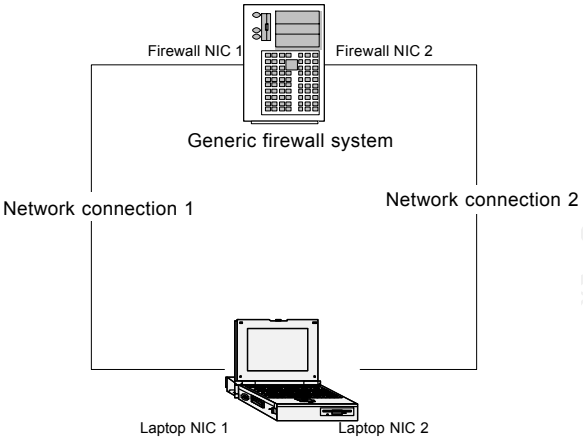
For instance, some hosts in the intranet may be allowed direct access to port 80 on external host **A**; however, this is not the default setting on the firewall, so test system **B** will not pick up that hole. A solution for this problem would be to install network 'sniffers' on both internal and external interfaces and analyze the results from there. However, the problem of external sources tampering with tests still exists.

A different testing infrastructure

A web search done on the Google search engine resulted in some interesting solutions being developed for this problem. Next, the author will propose a different solution for automated firewall testing procedures that attempts to solve the problems addressed above, and at the same time referring to existing material and its potential contribute for this solution.

The author purposes the development of a tool suite for firewall testing that could automate most of the common tests and even analyze simple test results and automatically point directions for future tests where they should be done.

An infrastructure for firewall testing



Test platfom w/ firewall testing tool suite

Using this network configuration, the auditor could simulate any network topology and simulate every type of attack. The advantages of having a single host connected to the firewall system using both network interfaces allows for unlimited possibilities, network-wise:

- Creation of 'virtual' or 'ghost' networks on each side of the firewall, for testing against *smurf* attacks, for instance.
- The ability to test for illegal traffic that could otherwise be filtered or lost in the internet, such has packets with internal destination IP addresses, spoofed or erroneous packets.
- The ability to run resource starvation and stress tests with great precision, since the whole testing environment is under control, and there is no possibility for outside tampering.
- The ability to send **and** receive network packets with no restrictions of source and/or destination IP addresses.
- The ability to send totally illegal and unexpected network traffic (fuzz-testing) to inspect the system's network protocol stack robustness.

The tool would take input in the form of parameters or even scripts for the test modules, simulate attacks on one network interface and watch the results on the other. After the test run is complete, the firewall would be taken back online, while the auditor analyses test results.

For instance, the tool suite could be configured to test for the ability to pass spoofed packets across the firewall. Then it could perform a simple TCP SYN packet scan and look for ports where traffic is allowed over the firewall. With the results from that test, it could then run some resource starvation attack at the firewall engine level, by creating a very large number of bogus connections thru the firewall to a single port, and all of this would be done without human intervention, logging results for later analysis.

Since the firewall will be unusable during the tests, this could present a problem for some organizations. But since the test run would be almost completely automated, the downtime should be very short. Also, the tests could be done on some backup system instead. This shortcoming seems a small price to pay for the ability to run network tests without limitations.

While this sounds complex at first, there are a lot of good tools that could speed up the development of this tool; the author found two scripting languages that could be used for the test suite, CASL[29] and NASL[30].

NASL at first glance seems perfect for the job: it is the scripting language for writing security tests for the Nessus open source security scanner. Unfortunately it has one limitation that rules it out completely: A NASL script will not send any packet to a host other than the target host for the Nessus scan. While this was originally meant for protecting the Nessus user from maliciously crafted NASL scripts, this makes impossible to test firewall rule sets, where it is essential to send packets with different destination IP's. So until this tool is redesigned to meet this requirement, it cannot be used.

CASL however, can perform some of the functions pretended, but has a restrictive non commercial use license. This could mean that additional development for this tool could not be used freely. Also, the company that once released this scripting language to the public seems to have withdrawn all support; the interpreter could not be run on a recent Linux installation, and the release found on the internet was to be used on an old Redhat 5.0 installation.

There still however good sources of inspiration, such as libpcap and libnet, probably used in conjunction with PERL.

The author also found a similar but still conceptually different project by the COAST team at Purdue University that seems to have ceased development, named Underfire. The documentation can still be browsed at http://www.cerias.purdue.edu/homes/firewall/methodology/underfire/index.html.

---

[29] Custom Auditing Scripting Language - http://www.ussrback.com/UNIX/utilities/casl20.tgz

[30] Nessus Attack Scripting Language - http://www.nessus.org/doc/nasl.html

Sources

Nistgov. "*Internet Security Policy: A Technical Guide - 6. Internet Firewall Policy*". URL: http://www-08.nist.gov/isptg/html/ISPTG-6.html (22 Oct. 2001).

Curtin, Matt. Ranum, Marcus. *"Internet Firewalls: Frequently Asked Questions"*. Rev. 10.0. 1 Dec. 2001. URL: http://www.interhack.net/pubs/fwfaq/ (22 Oct. 2001)

The SANS Institute. *"Intrusion Detection FAQ"*. Version 1.52. URL: http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.htm (23 Oct. 2001)

CERT. *"CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attack"*. Rev. Mar. 13, 2000. URL: http://www.cert.org/advisories/CA-1998-01.html (24 Oct. 2001)

Stocksdale, Greg. *"NSA Glossary of Terms Used in Security and Intrusion Detection"*. April 1998. URL: http://www.sans.org/newlook/resources/glossary.htm (24 Oct. 2001)

Management Analytics and Others. *"Management Analytics Firewall Checklist"*. URL: http://all.net/books/audit/Firewall/manal/index.html (29 Oct. 2001)

Spitzner, Lance. *"Auditing Your Firewall Setup"*. 12 Dec. 2000. URL: http://www.enteract.com/~lspitz/audit.html (31 Oct. 2001)

Loye, Ray. *"Security Audit Checklist"*. URL: http://polaris.umuc.edu/~lray/ifsm430/security-audit.htm (2 Nov. 2001)

Strom, Dan. *"Auditing the Netscreen-5 Firewall Used as a VPN Gateway"*. 16 Aug. 2001. URL: http://www.sans.org/y2k/practical/Dan_Strom_GSNA.zip (1 Nov. 2001)

Naidu, Krishni. The SANS Institute Resources. *"Firewall Checklist"*. (1 Nov. 2001).

Mackenzie, Elizabeth. *"Perimeter Filtering in a University Setting"*. 11 Sep. 2000. URL: http://www.sans.org/infosecFAQ/firewall/perimeter_filter.htm (3 Nov. 2001).

Appendix A - Management Analytics Firewall Checklist

Basic Firewall Control Principles

- Control Objectives
  - There are well defined control objectives for the firewall and it is known to and approved by the organization's executives at the highest level.
  - The control objectives for the firewall are attainable through the methods being used in the firewall.
  - Top-level management has the ability to determine the effectiveness of the firewalls in terms of whether or not they are providing the control specified by their control objectives.
  - There is adequate means to ensure that the roles and actions of all parties who interact with firewall are clearly defined, identified, and authenticated at a level commensurate with the sensitivity and criticality of the firewall's function.
- Functional Objectives
  - The firewall is designed to protect inside systems from exploitation by outside threats.
  - The firewall is designed to protect outside systems from exploitation by inside threats.
  - The firewall is designed to protect inside systems from exploitation by inside threats to the extent that that exploitation involves the firewall in any way.
  - The firewall is designed to protect itself and all attached systems against being used by attackers as a launch point for attacking other systems.
  - The firewall is designed so as to limit organizational liability.
  - The firewall is designed to prevent denial of service attacks.
  - The firewall is designed to prevent the corruption of organizational information and systems.
  - The firewall is designed to prevent the leakage of sensitive information.

Management Issues

- Executive Management
  - Management is highly supportive of effective firewall protection.
  - Management supports extensive protection of firewalls and recognized the criticality of protection in these systems.
  - Management recognizes that even with a perfect firewall, internal systems are not completely safe and has taken measures to secure internal systems in addition to the efforts to secure firewalls.

- Firewall operation is controlled by top-level management at an executive level and there are no more than 3 managers between the technical staff working on the firewall and the executive management of the organization.
- People working on the firewall do not report to the people whose access they control.
- Management is properly educated to make management decisions about firewalls.
- Firewall management crosses the boundaries of the organizations that directly or indirectly use the firewall.

- Systems Administration
  - The systems administration interface makes managing the firewall straight forward.
  - The management interface tends to reduce unnecessary complexity while allowing flexible drill-down for detailed understanding and verification of proper operation.
  - Error detection and response is highly automated, however, systems administrators are fully aware of the details of all incidents and the meaning of all indicative messages.
  - The management interface provides information in a manner and fashion that clarifies without hiding information.
  - The management interface provides statistical information and a method for viewing and analyzing data with an external analysis program such as a standard database or spreadsheet.
  - The management interface provides analytical capabilities that allow attacks, defenses, configurations, and user behavior to be readily analyzed.
  - The management interface provides a useful report generation capability.
  - Remote administration is facilitated by the firewall technology and there is a secure method for remote as well as local administration.
  - There is a safe method for making and testing changes to the firewall configuration.
  - The change mechanism includes a comprehensive automated configuration management system with a usable interface.
  - There is a method for easily and quickly backing out of changes.
  - There is an independent method of comparing two configurations for their differences and verifying those differences against change control information.

Policy Issues

- The firewall protection policy defined and approved in approved corporate security policy.
- The firewall protection policy identifies the specific assets that the firewall is intended to protect and those are the proper assets.
- The firewall protection policy specifies that integrity, availability, and confidentiality are all critical factors for the firewall.

- The policy specifies who is responsible for firewalls and provides the responsible individuals with adequate control to carry out the policy.
- The policy specifies a means of settling disputes that does not put undue pressure on the firewall personnel.
- The firewall policy is an official organizational policy and is approved and periodically reviewed by top management.

Standards/Procedure Issues

- There are written and approved standards and procedures for all aspects of firewall operation.
- The standards and procedures adequately characterize and specify the means by which policy elements are to be carried out.
- All employees have or have access to all standards and procedure information and are regularly told about changes as part of their awareness program.
- The standards and procedures specify who is responsible and empowered to do each function required for the proper operation of the firewall.
- Standards and procedures exist for secure distribution and update of all firewall hardware and software.
- Compliance with standards and procedures are verified by internal and external audit on a regular basis.
- Compliance with standards and procedures is documented contemporaneously with the performance of duties.

Documentation Issues

- The auditors, systems administrators, and incident response teams have input and review responsibilities for documentation.
- There are separate firewall documents for users, auditors, systems administrators, incident response teams, managers, and executives, each covering the information required for the function.
- Firewall hardware and software manuals are kept with the firewall.
- User manuals and help cards are kept in the users' work areas and pockets respectively.
- Auditor manuals and checklists are kept by the auditors.
- Systems administration manuals are kept at firewall control points.
- Incident response team manuals are kept in the incident response team locations.
- Manager manuals are kept in the mangers' offices.
- Executive manuals are kept in executive suites or wherever the executives are required to use them as part of their decision making process.
- Manuals are periodically reviewed.
- Manuals required to restore firewall function and content are kept in physical form next to the systems they apply to.

Audit Issues

- Internal Audit
  - A complete internal audit is done of each firewall at least once every 6 months.
  - Each internal audit covers all aspects of firewall protection covered in this (overall) audit checklist.
  - The internal IT audit team is well versed in all aspects of firewall operation.
  - Internal firewall audit reports go to the organization's director of security, members of the organization's security board, and the internal audit manager.
  - Internal firewall audit reports have identified problems and those problems have been addressed before the next audit report.
  - The internal firewall audit team is independent of the people who operate the firewall, the organization in charge of operating the firewall, and audit team members are rewarded for finding flaws in the firewall.
  - The team is regularly updated on this subject by widely acknowledged experts in the field.
  - The firewall audit team has found flaws in every audit to date.
- External Audit
  - A complete external audit is done of each firewall prior to it being put into service.
  - A complete external audit is done of each firewall at least once every operating year.
  - Each external audit covers all aspects of firewall protection covered in this (overall) audit checklist.
  - The external IT audit team is at least as well versed in firewall protection than internal firewall experts.
  - External firewall audit reports go (at least) to the organization's director of security, members of the organization's security board, and the internal audit manager.
  - External firewall audit reports have identified problems in the past and those problems are being addressed.
  - The external firewall audit team is independent of the people who operate the firewall and the organization in charge of operating the firewall.
  - The external audit team actively works on firewall technology and includes widely acknowledged experts in the field.
  - The external firewall audit team has found flaws in every audit to date.

Technical Safeguards

- Technical safeguards include protection from outside attacks, inside attacks, and attacks directed from within the firewall.
- The interaction of technical safeguards is well defined and understood.
- Technical safeguards include automated response to many of the most

common threats.

- Technical safeguards provide for interface with automated intrusion detection systems or capabilities.
- The firewall operates on highly secure operating systems.
- The firewall does NOT consist entirely of a screening router.
- The firewall properly separates a DMZ from the inside network and the outside network.
- The firewall does not artificially limit the number of simultaneous sessions that can operate through it, or the limits are such that they are beyond any anticipated performance requirements.
- The firewall is not artificially limited by the state information required to perform its function, or the limits are such that they are beyond any anticipated performance requirements.
- The size of the access control file does not grow to extremes given the complexity of the organization's current or anticipated access control requirements.
- Control of the access control file is adequate to assure that there are no windows of vulnerability as the access control information is changed.
- No denial of service results during changes of access control information.
- When access control information is changed, active sessions which access controls should not permit are terminated.
- None of the attacks that have become widely known in the last months have worked against this firewall.
- There is a systematic method for finding out about and updating the firewall to defend against new attacks.
- IP packet forwarding is turned off.
- Source routing does not operate through the firewall.
- The recent packet fragmentation attack did not work through this firewall.
- The firewall uses redundancy in the form of defense-in-depth to assure that no single attack or configuration error can bypass the firewall's controls.
- All processes operating on all firewall computers at the time of the audit are known to be appropriate and appear to be operating properly based on the process status listing.
- Traceroute through the Internet properly identifies routes including routes that cannot be verified as appropriate.
- Widely used tests run from over the Internet or other similar networks do not reveal any firewall flaws.
- The /etc/services file contains only services in actual use on each machine within the firewall.
- The /etc/inetd.conf file contains only services in actual use on each machine within the firewall.
- Comments are not used to disable services, rather, those service entries are not within the files used to identify those services to the operating system.
- All entries in all access control lists are known to be appropriate and have been

individually verified as part of reviewing this checklist.

- The password file has been examined for widely know inappropriate practices and no inappropriate or questionable entries are included within it.
- Crack has been run against a copy of the password file and none of the passwords were successfully guessed.
- Rsh and Portmapper functions are disabled on all firewall components.
- Regular backups are done of all firewall components.
- Copies of firewall backups are stored both on-site for rapid recovery and off-site for disaster recovery.
- Backups are restored on a regular basis on machines designated for disaster recovery as a test of their proper operation.
- Firewall files are cryptographically checksummed and those checksums are regularly verified.
- Firewall files are stored on read-only media and a system of sound change control is used to make firewall alterations.

Incident Response Issues

- There is a specific group of employees whose task is incident response.
- Incident response team members are highly experienced in information security, in the operation of the firewalls.
- The incident response team is represented on the organization's security committee.
- The incident response team participates in defining firewall protection.
- The incident response team participates in and contributes to the protection awareness program.
- The incident response team is backed up by more highly skilled experts.
- The incident response team tracks and reports on all incidents, both individually and on a statistical basis.
- The incident response team helps to implement protection.
- The incident response team plays a vital role in defining the protection mechanisms used in firewalls.
- The incident response team is intimately familiar with the day-to-day operation of the firewalls.
- The incident response team is responsible for incident follow-up including tracking down the sources of all incidents and following up until incidents are closed.
- The incident response team is available 24 hours a day, 7 days a week.
- There is a well defined disaster recovery plan.
- The disaster recovery plan is regularly tested and works properly every time.
- The disaster recovery plan switches over all functions of each firewall to other firewalls in a timely fashion.

Testing

- Internal testing of firewall effectiveness is performed by firewall administrators an at least a weekly basis.
- Internal testing is performed before and after each significant change in firewall configuration.
- The firewall is periodically tested from both sides using automated tools provided by outside providers on a regular basis.
- Select critical functions of the firewall are tested on a nearly continuous basis.
- Internal auditors do a thorough test of the firewall on every internal audit.
- External auditors do spot checks on every audit.
- Random, blind, and periodic outside testing of the firewall and the entire incident response system dealing with the firewall is done on an ongoing basis.
- Guest testers are periodically invited in to do firewall testing.
- The coverage of some of the tests is known and select firewall functions are fully tested by some of the tests.

Physical Security

- Physical protection of the firewall is comparable to the physical protection of systems used to create, maintain, process, and use the most critical information within the organization.
- Every firewall is protected to the same level of physical security.
- All firewall components including all devices used to manage the firewall are within the same secure perimeter.

Personnel Issues

- All people working on firewalls have been through company clearance procedures.
- Firewall personnel have had background checks and do not have criminal records.
- Firewall personnel are paid at a level commensurate with their experience and expertise.
- Firewall personnel consistently have good on-the-job performance.
- Firewall personnel have passed drug tests on a regular basis.
- Firewall personnel have at least five years of experience within the organization and are highly trusted.
- The organization has a policy of punishing perpetrators to the fullest extent possible and employees are aware of this policy.
- The organization has a policy to help victims of information-based attacks.
- The organization recognizes employees who help identify attacks and rewards them in regular performance evaluations.
- Firewall personnel and others within the organization who help defend systems from malicious threats are highly rewarded for their efforts.
- Working on the firewall is a step toward long term job security, high pay, and increased responsibility.

- Human resources help to coordinate training and awareness programs.
- Human resources have an effective system of informing firewall personnel of all personnel changes.
- Human resources tracks information protection performance with other performance evaluation criteria.

Legal Issues

- Organizational lawyers include experts in the areas of the law relating to information protection.
- Outside specialists are used to augment organizational lawyers.
- The firewalls provide adequate notice to outsiders attempting to exploit them.
- The firewalls provide adequate notice to insiders attempting to exploit them.
- All messages provided by the firewall are approved by organizational lawyers.
- All policies and procedures related to firewall operation are approved by corporate lawyers.
- All incident response activities are approved by corporate legal staff.
- There is a well-defined interface with law enforcement for all actions related to attacks against the firewall.
- Corporate legal staff understands the jurisdictional issues related to all interconnected networks, including the international legal implications of the Internet.
- Legal contracts with vendors supplying external network services include proper liability limitations and responsibilities.
- Contracts with other organizations using the firewall include all necessary and appropriate language required to protect the organizations from liability and from attacks in and from each others' networks.
- Officers are aware of their responsibilities relating to firewall operation.

Awareness Issues

- All employees are taught about the dangers associated with use of outside systems.
- All employees are trained on who to report suspected attacks to.
- Managers have a special awareness program that covers what managers need to know about using firewalls and dealing with employees who use them.
- Executives have a special awareness program that covers what managers need to know about using firewalls and dealing with employees who use them.
- Every employee, manager, and executive is given an awareness briefing regarding the use of firewalls at least once per year.
- Awareness programs include measurements of their effectiveness to assure that they work properly.
- Awareness programs include reports of incidents from within the organization, incidents at other comparable organizations, and the overall situation throughout the world.

- Awareness programs explicitly address the fact that social engineering makes firewalls and other perimeter security ineffective.
- Awareness programs teach all employees how to deal with social engineering threats.
- The awareness program is one of the highlights of employees' life within the organization.

Training and Education

- Users are trained who to call about, when to call about, and how to recognize incidents.
- Users are trained in the use of any tools they will have to use as a part of the incident response process.
- Programmers and analysts are trained about typical programming errors and the impacts of errors and omissions on the organization.
- Programmers and analysts are trained on how to design for integrity and availability.
- Programmers and analysts are trained on the rules of the road for IT employees.
- Managers are trained on how to make prudent protection decisions.
- Managers are trained on how to react to incidents and how to evaluate decisions in these situations.
- Managers are trained on how the protection process works, the role that firewalls play and don't play, and how their part of the organization interacts with the firewall and the people who operate it.
- Incident response teams are trained on all aspects of information protection.
- Adequate measures are made of the effectiveness of training and they show that the training is working as it is supposed to.

Organizational Suitability

- Upper management is supportive of firewall protection efforts.
- There is adequate funding to support all protection measures desired for firewalls.
- Proper policies have been approved by upper management for firewalls.
- Management has supported adequate personnel to properly operate the firewalls.
- Management has facilitated secure firewall operation in an active way.
- No inter-organizational problems are encountered in the placement and operation of firewalls.

© SANS Institute 2000 - 2005 Author retains full rights.

Appendix B – Full command output

Output from the pstree command:

```
$ pstree
init-+-atd
     |-bgpd
     |-cron
     |-gpm
     |-keventd
     |-klogd
     |-kreiserfsd
     |-master-+-pickup
     |        |-qmgr
     |        `-tlsmgr
     |-6*[mingetty]
     |-nscd---nscd---5*[nscd]
     |-ospfd
     |-rtmon
     |-screen---2*[bash]
     |-snmpd
     |-snort
     |-squid---squid---unlinkd
     |-sshd---sshd---bash---pstree
     |-syslogd
     |-xntpd---xntpd---xntpd
     `-zebra
```

Output from the ps aux command:

```
$ ps aux
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   440   64 ?        S    Oct12   0:04 init [3]
root         2  0.0  0.0     0    0 ?        SW   Oct12   0:00 [keventd]
root         3  0.0  0.0     0    0 ?        SWN  Oct12   0:45 [ksoftirqd_CPU0]
root         4  0.0  0.0     0    0 ?        SW   Oct12   3:38 [kswapd]
root         5  0.0  0.0     0    0 ?        SW   Oct12   0:00 [bdflush]
root         6  0.0  0.0     0    0 ?        SW   Oct12   2:51 [kupdated]
root         9  0.0  0.0     0    0 ?        SW   Oct12   0:02 [kreiserfsd]
root       248  0.0  0.0  2236  260 ?        S    Oct12   0:11 /usr/sbin/sshd
root       260  0.0  0.0  1352  228 ?        S    Oct12   0:26 /sbin/syslogd
root       264  0.0  0.0  1868  188 ?        S    Oct12   0:04 /sbin/klogd -c 1
at         286  0.0  0.0  1432  104 ?        S    Oct12   0:00 /usr/sbin/atd
root       379  0.0  0.0  3568  140 ?        S    Oct12   0:00 /usr/lib/postfix/master
root       418  0.0  0.0  1436  116 ?        S    Oct12   0:00 /usr/sbin/cron
root       435  0.0  0.0 11764  352 ?        S    Oct12   0:00 /usr/sbin/nscd
root       452  0.0  0.0 11764  352 ?        S    Oct12   0:00 /usr/sbin/nscd
root       453  0.0  0.0 11764  352 ?        S    Oct12   0:00 /usr/sbin/nscd
root       454  0.0  0.0 11764  352 ?        S    Oct12   0:00 /usr/sbin/nscd
root       455  0.0  0.0 11764  352 ?        S    Oct12   0:00 /usr/sbin/nscd
root       456  0.0  0.0 11764  352 ?        S    Oct12   0:00 /usr/sbin/nscd
root       457  0.0  0.0 11764  352 ?        S    Oct12   0:00 /usr/sbin/nscd
root       459  0.0  0.0  1316   24 ?        S    Oct12   0:00 /usr/sbin/gpm -t ps2 -m /dev/mouse
root       465  0.0  0.0  1268    4 tty3     S    Oct12   0:00 /sbin/mingetty tty3
root       466  0.0  0.0  1268    4 tty4     S    Oct12   0:00 /sbin/mingetty tty4
root       467  0.0  0.0  1268    4 tty5     S    Oct12   0:00 /sbin/mingetty tty5
```

```
root       468  0.0  0.0  1268    4 tty6     S    Oct12   0:00 /sbin/mingetty tty6
root      6226  0.0  0.0  1344   84 ?        S    Oct12   0:00 rtmon file /var/log/rtmon.log
root     17432  0.0  0.7  3952 3944 ?        SL   Oct16   0:00 /usr/sbin/xntpd
root     17433  0.0  0.7  3952 3944 ?        SL   Oct16   0:00 /usr/sbin/xntpd
root     17434  0.0  0.7  3952 3944 ?        SL   Oct16   0:01 /usr/sbin/xntpd
postfix  26558  0.0  0.0  6236  200 ?        S    Oct16   0:03 qmgr -l -t unix -u -c
postfix  26559  0.0  0.0  3588  172 ?        S    Oct16   0:01 tlsmgr -l -t fifo -u
root     14954  3.5  1.6 11876 8320 ?        S    Oct17 953:55 /usr/sbin/snort -d -D -c
/etc/snort/sn
root     19695  0.0  0.0  1268    4 tty1     S    Oct18   0:00 /sbin/mingetty --noclear tty1
root     14635  0.0  0.0  3568  292 ?        S    Oct19   3:07 /usr/sbin/snmpd -c /etc/ucdsnmpd.conf
root       438  0.0  0.0  2272    4 ?        S    Oct24   0:00 SCREEN
root       439  0.0  0.0  2616    4 pts/7    S    Oct24   0:00 /bin/bash
root       451  0.0  0.0  2612    4 pts/9    S    Oct24   0:00 /bin/bash
root     12748  0.0  0.0  1272    4 tty2     S    Oct26   0:00 /sbin/mingetty tty2
root     14023  0.0  0.2  2280 1256 ?        S    Oct31   0:08 /usr/sbin/ospfd -d
root     19975  0.0  0.1  1848  812 ?        S    Nov02   0:00 /usr/sbin/zebra -d
root     20220  0.0  0.3  2628 1612 ?        S    Nov02   0:00 /usr/sbin/bgpd -d
root     21520  0.0  0.2  3836 1188 ?        S    Nov02   0:00 /usr/local/squid/bin/squid -sYD
squid    21522  2.5 22.1 115172 113800 ?     S    Nov02  77:37 (squid) -sYD
squid    21523  0.0  0.0  1256  356 ?        S    Nov02   0:45 (unlinkd)
postfix  28985  0.0  0.2  3520 1132 ?        S    00:21   0:00 pickup -l -t unix -c
root     29014  0.0  0.3  2916 1668 ?        S    00:36   0:00 /usr/sbin/sshd
luis     29015  0.0  0.2  2600 1488 pts/2    S    00:36   0:00 -bash
root     29091  0.0  0.0     0    0 ?        Z    01:15   0:00 [cron <defunct>]
luis     29119  0.0  0.2  2436 1500 pts/2    R    01:15   0:00 ps aux
```

Output from the rpm -qa command:

```
$ ps aux

aaa_base-2001.5.15-2
aaa_dir-2001.5.2-1
aaa_skel-2001.5.21-0
ash-0.2-334
base-2001.5.9-4
bash-2.05-21
bc-1.06-55
bdflush-1.5-335
bzip-1.0.1-48
compress-4.2.4-328
cpio-2.4.2-344
cracklib-2.7-313
cron-3.0.1-339
db-3.1.17-56
devs-2001.5.21-0
diffutils-2.7-72
e2fsprogs-1.19-56
file-3.32-74
fileutils-4.0.35-42
findutils-4.1.6-65
gawk-3.0.6-96
gdbm-1.8.0-282
glibc-2.2.2-38
gppshare-2.95.3-52
grep-2.4-29
groff-1.16.1-93
gzip-1.3-46
kbd-1.05-14
less-358-74
libz-1.1.3-341
lilo-21.6-56
m4-1.4o-26
```

```
man-2.3.17deb3.2-28
mktemp-1.5-192
modutils-2.4.5-12
ncurses-5.2-86
net-tools-1.60-8
netcfg-2001.5.11-0
pam-0.74-36
pam_devperm-2000.12.1-47
pciutils-2.1.8-118
perl-5.6.0-81
ps-2001.5.10-3
readline-2.05-21
reiserfs-3.x.0j-17
rpm-3.0.6-78
sh-utils-2.0-50
shadow-20000902-79
syslogd-1.3.33-239
sysvinit-2.78.4-42
terminfo-5.2-65
textutils-2.0.10-49
timezone-2.2.2-38
util-linux-2.11b-22
vim-5.7-90
yast-1.11.1-0
at-3.1.8-323
gfxboot-1.2-2
gpm-1.18.1-213
hdparm-3.9-25
idedma-1.0-135
isapnp-1.21-180
joe-2.9.5-16
lsof-4.55-28
pico-4.33-42
rawio-2-214
recode-3.5-246
unzip-5.42-14
binutils-2.10.91.0.4-40
cpp-2.95.3-52
gcc-2.95.3-52
glibc-devel-2.2.2-38
hwinfo-2.15-1
libgpp-2.95.3-52
popt-1.6-0
strace-4.2-155
bindutil-8.2.3-140
finger-1.0-26
iproute2-2.2.4-234
iputils-20001110-44
lynx-2.8.3dev9-318
mailx-8.1.1-311
netcat-1.10-319
portmap-5beta-154
procmail-3.15.1-39
tcpdump-3.4a6-325
telnet-1.0-27
xntp-4.0.99f-67
cyrus-sasl-1.5.24-59
gpg-1.0.5-15
openssh-2.9p1-7
openssl-0.9.6a-9
tcl-8.3.3-23
xshared-4.0.3-35
libscr-2.3.2-14
liby2-2.3.9-0
```

```
libycp-2.3.27-4
libyui-2.3.9-7
yast2-agent-any-2.3.11-3
yast2-agent-fdisk-2.3.7-0
yast2-agent-isax-2.3.4-6
yast2-agent-modules-2.3.4-7
yast2-agent-probe-2.3.8-0
yast2-agent-rcconfig-2.3.5-14
yast2-base-2.3.4-14
yast2-config-adsl-2.3.6-2
yast2-config-bootfloppy-2.3.5-5
yast2-config-bootloader-2.3.10-0
yast2-config-environment-1.0.8-0
yast2-config-hwinfo-2.3.4-0
yast2-config-inet-2.3.3-8
yast2-config-network-2.3.21-0
yast2-config-online-update-2.3.8-3
yast2-config-package-manager-2.3.6-0
yast2-config-rcconfig-2.3.3-3
yast2-config-security-2.3.5-13
yast2-config-tune-2.3.3-0
yast2-config-update-2.3.5-2
yast2-config-users-2.3.8-0
yast2-core-clients-2.3.10-8
yast2-core-pkginfo-2.3.18-3
yast2-core-scr-2.3.30-1
yast2-core-translator-2.3.5-14
yast2-core-wfm-2.3.7-7
yast2-db-groups-1.0.20-0
yast2-db-keyboard-2.1.38-0
yast2-db-mouse-2.1.37-0
yast2-db-timezone-2.1.37-0
yast2-lib-sequencer-2.3.5-22
yast2-lib-wizard-2.3.8-9
yast2-module-support-2.3.13-0
yast2-trans-adsl-2.1.38-0
yast2-trans-boot-2.0.40-0
yast2-trans-bootfloppy-1.0.15-0
yast2-trans-hwinfo-2.0.13-0
yast2-trans-inet-2.1.34-0
yast2-trans-inst-bootloader-2.1.47-0
yast2-trans-inst-environment-2.1.47-0
yast2-trans-inst-general-2.1.54-0
yast2-trans-inst-language-2.1.44-0
yast2-trans-inst-packages-2.1.52-0
yast2-trans-inst-partitioning-2.1.54-0
yast2-trans-inst-update-2.1.48-0
yast2-trans-inst-user-2.1.46-0
yast2-trans-inst-x11-2.1.50-0
yast2-trans-menu-2.1.38-0
yast2-trans-network-2.1.55-0
yast2-trans-online-update-2.1.19-0
yast2-trans-package-manager-2.2.17-0
yast2-trans-rcconfig-2.0.41-0
yast2-trans-security-2.1.20-0
yast2-trans-support-1.0.27-0
yast2-trans-tune-2.1.43-0
yast2-trans-update-2.0.41-0
yast2-trans-users-1.0.17-0
yast2-trans-wizard-1.0.31-0
yast2-ui-ncurses-2.3.12-3
yast2-2.3.90-0
k_deflt-2.4.4-15
eazy-2001.4.24-1
```

```
lsb-0.4.1-41
acct-6.3.5-150
dump-0.4b21-34
makedev-2.5.3-129
quota-3.01pre5-4
su1-4.2-308
wipe-1.2.2-52
iptraf-2.3.1-39
ncftp-3.0.2-46
postfix-20010228pl02-8
rsync-2.4.6-85
squid-2.3.STABLE4-51
ucdsnmp-4.2.1-15
zebra-0.90a-45
iptables-1.2.1a-37
nmap-2.53-127
scanlogd-2.2-43
snort-1.7-38
tripwire-1.2-296
cvs-1.11-71
libpcapn-0.4a6-324
ltrace-0.3.10-145
libnet-1.0.2a-34
openldap2-client-2.0.7-70
pcre-3.4-61
make-3.79.1-126
ncurses-devel-5.2-86
screen-3.9.8-104
ntop-1.3.2-95
emacs-info-20.7-71
emacs-20.7-71
emacs-el-20.7-71
emacs-nox-20.7-71
gdbm-devel-1.8.0-308
patch-2.5.4-44
```

Output from the find / -perm +6000 command

```
$ find / -perm +6000 2> /dev/null
/bin/su
/bin/ping
/bin/mount
/bin/ping6
/bin/umount
/usr/bin/at
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/crontab
/usr/bin/chage
/usr/bin/mandb
/usr/bin/expiry
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/gpasswd
/usr/lib/pt_chown
/usr/sbin/pam_auth
/usr/sbin/postdrop
/sbin/unix_chkpwd
```

Appendix C – iptables rule listing

Output from the "iptables –list –nv" command (IP addresses were changed):

```
# iptables --list -nv
Chain INPUT (policy ACCEPT 72747163 packets, 39216154083 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       udp  --  *      *       123.234.102.164      0.0.0.0/0           udp dpt:3130

Chain FORWARD (policy ACCEPT 519728956 packets, 330583803023 bytes)
 pkts bytes target     prot opt in     out     source               destination
 1210 66098 DROP       all  --  *      *       0.0.0.255/0.0.0.255  0.0.0.0/0
 278K   44M DROP       all  --  *      *       0.0.0.0/0            0.0.0.255/0.0.0.255
 2132  112K DROP       all  --  *      *       0.0.0.0/0.0.0.255    0.0.0.0/0
1267K  216M DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0.0.0.255
   90  5040 DROP       icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmp type 5
    0     0 DROP       all  --  *      *       0.0.0.0/0            127.0.0.0/8
    0     0 DROP       all  --  *      *       127.0.0.0/8          0.0.0.0/0
43428 2187K DROP       all  --  *      *       0.0.0.0/0            10.0.0.0/8
  805 59644 DROP       all  --  *      *       10.0.0.0/8           0.0.0.0/0
  379 23057 DROP       all  --  *      *       0.0.0.0/0            172.16.0.0/12
10658  993K DROP       all  --  *      *       172.16.0.0/12        0.0.0.0/0
 105K 7475K DROP       all  --  *      *       0.0.0.0/0            192.168.0.0/16
 617K   35M DROP       all  --  *      *       192.168.0.0/16       0.0.0.0/0
  234 12018 DROP       all  --  *      *       0.0.0.0/0            123.234.130.77
  536 36030 DROP       all  --  *      *       0.0.0.0/0            123.234.140.200
  213 10360 DROP       all  --  *      *       0.0.0.0/0            123.234.142.131
 1595 76560 DROP       all  --  *      *       123.234.142.131      0.0.0.0/0
  239 12802 DROP       all  --  *      *       0.0.0.0/0            123.234.152.143
  259 12428 DROP       all  --  *      *       0.0.0.0/0            123.234.140.28
 499M  320G TCP        tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
  23M 2916M UDP        udp  --  *      *       0.0.0.0/0            0.0.0.0/0
   36  1860 DROP       all  --  *      *       0.0.0.0/0            123.234.130.12
    4   192 DROP       all  --  *      *       123.234.130.12       0.0.0.0/0

Chain OUTPUT (policy ACCEPT 76629959 packets, 40959254227 bytes)
 pkts bytes target     prot opt in     out     source               destination
 151K   13M DROP       icmp --  *      *       0.0.0.0/0            0.0.0.0/0           icmp type 11

Chain SMTP (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 REJECT     tcp  --  eth1   *       0.0.0.0/0            123.234.129.89      reject-with tcp-reset
  811 40104 REJECT     tcp  --  eth1   *       0.0.0.0/0            123.234.167.2       reject-with tcp-reset
   50  2284 REJECT     tcp  --  eth1   *       0.0.0.0/0            123.234.132.3       reject-with tcp-reset

Chain TCP (1 references)
 pkts bytes target     prot opt in     out     source               destination
  142  6852 REJECT     tcp  --  eth1   *       0.0.0.0/0            123.234.132.160     reject-with tcp-reset
 1075 61632 REJECT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpts:512:515
reject-with tcp-reset
26097 1565K REJECT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports
87,540,111 reject-with tcp-reset
  11M   10G SMTP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:25

Chain UDP (1 references)
 pkts bytes target     prot opt in     out     source               destination
11433  869K ACCEPT     udp  --  *      *       123.234.134.46       0.0.0.0/0
12399  943K ACCEPT     udp  --  *      *       0.0.0.0/0            123.234.134.46
    0     0 ACCEPT     udp  --  *      *       123.234.134.58       0.0.0.0/0
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            123.234.134.58
4295K  420M ACCEPT     udp  --  *      *       123.234.128.1        0.0.0.0/0
    0     0 ACCEPT     udp  --  *      *       123.234.134.82       123.234.128.1
42395 2429K DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           multiport dports
```

69,111,161,162,2049

Output from the "iptables –list –nv –t nat" command (IP addresses were changed):

```
# iptables --list -nv -t nat
Chain PREROUTING (policy ACCEPT 12748082 packets, 960913455 bytes)
 pkts bytes target     prot opt in     out     source               destination
 128K 6136K DROP       tcp  --  eth0   *       123.234.130.105      0.0.0.0/0           tcp dpt:80
    0     0 DROP       tcp  --  eth0   *       123.234.167.99       0.0.0.0/0           tcp dpt:80
    0     0 DROP       tcp  --  eth0   *       123.234.167.98       0.0.0.0/0           tcp dpt:80
  16M  759M DROP       tcp  --  eth0   *       123.234.199.136      0.0.0.0/0           tcp dpt:80
  188  8268 ACCEPT     tcp  --  *      *       123.234.164.3        0.0.0.0/0           tcp dpt:80
3724K  181M HTTP-OUT   tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80

Chain POSTROUTING (policy ACCEPT 11499924 packets, 706038826 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 SNAT       tcp  --  *      inesc   192.168.253.2        0.0.0.0/0           to:123.234.134.45
  195 11700 SNAT       tcp  --  *      inesc   192.168.250.2        0.0.0.0/0           to:123.234.134.45

Chain OUTPUT (policy ACCEPT 825368 packets, 49725546 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain HTTP-OUT (1 references)
 pkts bytes target     prot opt in     out     source               destination
   48  4309 RETURN     tcp  --  eth0   *       0.0.0.0/0            193.131.119.86      tcp dpt:80
   96  7113 RETURN     tcp  --  eth0   *       0.0.0.0/0            216.174.50.68       tcp dpt:80
   10   600 RETURN     tcp  --  eth0   *       0.0.0.0/0            149.28.1.15         tcp dpt:80
   24  1668 RETURN     tcp  --  eth0   *       0.0.0.0/0            193.54.241.194      tcp dpt:80
    2    88 RETURN     tcp  --  eth0   *       0.0.0.0/0            209.183.218.100     tcp dpt:80
   30  1384 RETURN     tcp  --  eth0   *       0.0.0.0/0            216.205.17.52       tcp dpt:80
  131  5792 RETURN     tcp  --  eth0   *       0.0.0.0/0            192.87.90.182       tcp dpt:80
   26  1560 RETURN     tcp  --  eth0   *       0.0.0.0/0            137.122.49.242      tcp dpt:80
    0     0 RETURN     tcp  --  eth0   *       0.0.0.0/0            207.228.228.116     tcp dpt:80
    0     0 RETURN     tcp  --  eth0   *       0.0.0.0/0            216.63.246.103      tcp dpt:80
   19  1132 RETURN     tcp  --  eth0   *       0.0.0.0/0            141.66.176.201      tcp dpt:80
    0     0 RETURN     tcp  --  eth0   *       0.0.0.0/0            62.49.248.162       tcp dpt:80
    0     0 RETURN     tcp  --  eth0   *       0.0.0.0/0            206.183.24.41       tcp dpt:80
   83  9741 RETURN     tcp  --  eth0   *       0.0.0.0/0            193.63.84.7         tcp dpt:80
  129  6424 RETURN     tcp  --  eth0   *       0.0.0.0/0            208.215.179.82      tcp dpt:80
    9   404 RETURN     tcp  --  eth0   *       0.0.0.0/0            209.58.150.38       tcp dpt:80
    0     0 RETURN     tcp  --  eth0   *       0.0.0.0/0            129.79.147.66       tcp dpt:80
    3   128 RETURN     tcp  --  eth0   *       0.0.0.0/0            202.248.51.150      tcp dpt:80
   25  1112 RETURN     tcp  --  eth0   *       0.0.0.0/0            192.11.226.2        tcp dpt:80
   14   616 RETURN     tcp  --  eth0   *       0.0.0.0/0            130.88.203.157      tcp dpt:80
  150 11185 RETURN     tcp  --  eth0   *       0.0.0.0/0            198.81.209.3        tcp dpt:80
    0     0 RETURN     tcp  --  eth0   *       0.0.0.0/0            208.5.159.4         tcp dpt:80
 1595 77272 RETURN     tcp  --  eth0   *       0.0.0.0/0            194.65.112.25       tcp dpt:80
    0     0 RETURN     tcp  --  eth0   *       0.0.0.0/0            64.55.87.3          tcp dpt:80
    0     0 RETURN     tcp  --  eth0   *       0.0.0.0/0            147.65.1.17         tcp dpt:80
  197 10955 RETURN     tcp  --  eth0   *       0.0.0.0/0            129.70.14.12        tcp dpt:80
    0     0 RETURN     tcp  --  eth0   *       0.0.0.0/0            128.42.5.70         tcp dpt:80
   17  1878 RETURN     tcp  --  eth0   *       0.0.0.0/0            130.79.4.90         tcp dpt:80
   97  4268 RETURN     tcp  --  eth0   *       0.0.0.0/0            192.68.254.190      tcp dpt:80
  498 23896 RETURN     tcp  --  eth0   *       123.234.165.175      0.0.0.0/0           tcp dpt:80
3721K  180M REDIRECT   tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80 redir
ports 3128
```

Appendix D – Inbound and outbound traffic analysis with tcpdump

1 – Outbound TCP traffic (syn scan to an outside machine, port range from 1-1024)

(Due to very large output, only important and example entries are shown)

output from: tcpdump -n -i eth1 src net 123.234

```
05:23:48.835362 123.234.164.2.1 > 213.22.64.248.1: S 2284321954:2284321954(0) win 1024
05:23:48.835463 123.234.164.2.1 > 213.22.64.248.2: S 2284321954:2284321954(0) win 1024
05:23:48.835570 123.234.164.2.1 > 213.22.64.248.3: S 2284321954:2284321954(0) win 1024
05:23:48.835733 123.234.164.2.1 > 213.22.64.248.4: S 2284321954:2284321954(0) win 1024
05:23:48.837877 123.234.164.2.1 > 213.22.64.248.5: S 2284321954:2284321954(0) win 1024
05:23:48.837977 123.234.164.2.1 > 213.22.64.248.6: S 2284321954:2284321954(0) win 1024
05:23:48.838084 123.234.164.2.1 > 213.22.64.248.7: S 2284321954:2284321954(0) win 1024
05:23:48.838262 123.234.164.2.1 > 213.22.64.248.8: S 2284321954:2284321954(0) win 1024
05:23:48.838290 123.234.164.2.1 > 213.22.64.248.9: S 2284321954:2284321954(0) win 1024
...
<snip>
...
05:23:49.316394 123.234.164.2.1 > 213.22.64.248.79: S 2284321954:2284321954(0) win 1024
05:23:49.317060 123.234.164.2.1 > 213.22.64.248.81: S 2284321954:2284321954(0) win 1024
05:23:49.317153 123.234.164.2.1 > 213.22.64.248.82: S 2284321954:2284321954(0) win 1024
05:23:49.493158 123.234.164.2.1 > 213.22.64.248.83: S 2284321954:2284321954(0) win 1024
05:23:49.493765 123.234.164.2.1 > 213.22.64.248.84: S 2284321954:2284321954(0) win 1024
05:23:49.493851 123.234.164.2.1 > 213.22.64.248.85: S 2284321954:2284321954(0) win 1024
05:23:49.493921 123.234.164.2.1 > 213.22.64.248.86: S 2284321954:2284321954(0) win 1024
05:23:49.494013 123.234.164.2.1 > 213.22.64.248.88: S 2284321954:2284321954(0) win 1024
05:23:49.494067 123.234.164.2.1 > 213.22.64.248.89: S 2284321954:2284321954(0) win 1024
05:23:49.494224 123.234.164.2.1 > 213.22.64.248.90: S 2284321954:2284321954(0) win 1024
...
<snip>
...
05:23:49.497119 123.234.164.2.1 > 213.22.64.248.107: S 2284321954:2284321954(0) win 1024
05:23:49.497276 123.234.164.2.1 > 213.22.64.248.108: S 2284321954:2284321954(0) win 1024
05:23:49.500555 123.234.164.2.1 > 213.22.64.248.109: S 2284321954:2284321954(0) win 1024
05:23:49.743576 123.234.164.2.1 > 213.22.64.248.110: S 2284321954:2284321954(0) win 1024
05:23:49.744034 123.234.164.2.1 > 213.22.64.248.112: S 2284321954:2284321954(0) win 1024
05:23:49.744122 123.234.164.2.1 > 213.22.64.248.113: S 2284321954:2284321954(0) win 1024
05:23:49.744192 123.234.164.2.1 > 213.22.64.248.114: S 2284321954:2284321954(0) win 1024
...
<snip>
...
05:23:49.765713 123.234.164.2.1 > 213.22.64.248.134: S 2284321954:2284321954(0) win 1024
05:23:49.765809 123.234.164.2.1 > 213.22.64.248.135: S 2284321954:2284321954(0) win 1024
05:23:49.765967 123.234.164.2.1 > 213.22.64.248.136: S 2284321954:2284321954(0) win 1024
05:23:49.767855 123.234.164.2.1 > 213.22.64.248.137: S 2284321954:2284321954(0) win 1024
05:23:49.768341 123.234.164.2.1 > 213.22.64.248.138: S 2284321954:2284321954(0) win 1024
05:23:49.768746 123.234.164.2.1 > 213.22.64.248.139: S 2284321954:2284321954(0) win 1024
05:23:49.769024 123.234.164.2.1 > 213.22.64.248.140: S 2284321954:2284321954(0) win 1024
05:23:49.959850 123.234.164.2.1 > 213.22.64.248.141: S 2284321954:2284321954(0) win 1024
...
<snip>
...
05:23:53.034239 123.234.164.2.1 > 213.22.64.248.509: S 2284321954:2284321954(0) win 1024
05:23:53.036001 123.234.164.2.1 > 213.22.64.248.510: S 2284321954:2284321954(0) win 1024
05:23:53.036103 123.234.164.2.1 > 213.22.64.248.511: S 2284321954:2284321954(0) win 1024
05:23:53.036200 123.234.164.2.1 > 213.22.64.248.516: S 2284321954:2284321954(0) win 1024
05:23:53.036251 123.234.164.2.1 > 213.22.64.248.517: S 2284321954:2284321954(0) win 1024
...
<snip>
...
```

```
05:23:54.087961 123.234.164.2.1 > 213.22.64.248.536: S 2284321954:2284321954(0) win 1024
05:23:54.088033 123.234.164.2.1 > 213.22.64.248.537: S 2284321954:2284321954(0) win 1024
05:23:54.088123 123.234.164.2.1 > 213.22.64.248.538: S 2284321954:2284321954(0) win 1024
05:23:54.088175 123.234.164.2.1 > 213.22.64.248.539: S 2284321954:2284321954(0) win 1024
05:23:54.088333 123.234.164.2.1 > 213.22.64.248.541: S 2284321954:2284321954(0) win 1024
05:23:54.088491 123.234.164.2.1 > 213.22.64.248.542: S 2284321954:2284321954(0) win 1024
05:23:54.088648 123.234.164.2.1 > 213.22.64.248.543: S 2284321954:2284321954(0) win 1024
...
<snip>
...
05:23:57.522068 123.234.164.2.1 > 213.22.64.248.1022: S 2284321954:2284321954(0) win 1024
05:23:57.523206 123.234.164.2.1 > 213.22.64.248.1023: S 2284321954:2284321954(0) win 1024
05:23:57.524030 123.234.164.2.1 > 213.22.64.248.1024: S 2284321954:2284321954(0) win 1024
```

2 – Inbound TCP traffic (syn scan to an inside machine, port range from 1-1024)

(Due to very large output, only important and example entries are shown)

output from: tcpdump -n -i eth0 src **213.22.64.248**

```
06:04:26.065545 213.22.64.248.1 > 123.234.164.4.1: S 3825619951:3825619951(0) win 2048
06:04:26.073770 213.22.64.248.1 > 123.234.164.4.2: S 3825619951:3825619951(0) win 2048
06:04:26.079676 213.22.64.248.1 > 123.234.164.4.3: S 3825619951:3825619951(0) win 2048
06:04:26.083806 213.22.64.248.1 > 123.234.164.4.4: S 3825619951:3825619951(0) win 2048
06:04:26.096600 213.22.64.248.1 > 123.234.164.4.5: S 3825619951:3825619951(0) win 2048
06:04:26.097272 213.22.64.248.1 > 123.234.164.4.6: S 3825619951:3825619951(0) win 2048
06:04:26.101684 213.22.64.248.1 > 123.234.164.4.7: S 3825619951:3825619951(0) win 2048
06:04:26.107684 213.22.64.248.1 > 123.234.164.4.8: S 3825619951:3825619951(0) win 2048
06:04:26.113653 213.22.64.248.1 > 123.234.164.4.9: S 3825619951:3825619951(0) win 2048
...
<snip>
...
06:04:28.187368 213.22.64.248.1 > 123.234.164.4.80: S 3825619951:3825619951(0) win 2048
06:04:28.199927 213.22.64.248.1 > 123.234.164.4.81: S 3825619951:3825619951(0) win 2048
06:04:28.204721 213.22.64.248.1 > 123.234.164.4.82: S 3825619951:3825619951(0) win 2048
06:04:28.627530 213.22.64.248.1 > 123.234.164.4.83: S 3825619951:3825619951(0) win 2048
06:04:28.632936 213.22.64.248.1 > 123.234.164.4.84: S 3825619951:3825619951(0) win 2048
06:04:28.645562 213.22.64.248.1 > 123.234.164.4.85: S 3825619951:3825619951(0) win 2048
06:04:28.646944 213.22.64.248.1 > 123.234.164.4.86: S 3825619951:3825619951(0) win 2048
06:04:28.659640 213.22.64.248.1 > 123.234.164.4.88: S 3825619951:3825619951(0) win 2048
06:04:28.665522 213.22.64.248.1 > 123.234.164.4.89: S 3825619951:3825619951(0) win 2048
06:04:28.672746 213.22.64.248.1 > 123.234.164.4.90: S 3825619951:3825619951(0) win 2048
...
<snip>
...
06:04:28.772912 213.22.64.248.1 > 123.234.164.4.107: S 3825619951:3825619951(0) win 2048
06:04:28.779490 213.22.64.248.1 > 123.234.164.4.108: S 3825619951:3825619951(0) win 2048
06:04:28.785256 213.22.64.248.1 > 123.234.164.4.109: S 3825619951:3825619951(0) win 2048
06:04:29.187573 213.22.64.248.1 > 123.234.164.4.110: S 3825619951:3825619951(0) win 2048
06:04:29.197947 213.22.64.248.1 > 123.234.164.4.112: S 3825619951:3825619951(0) win 2048
06:04:29.203505 213.22.64.248.1 > 123.234.164.4.113: S 3825619951:3825619951(0) win 2048
06:04:29.212059 213.22.64.248.1 > 123.234.164.4.114: S 3825619951:3825619951(0) win 2048
...
<snip>
...
06:04:29.336826 213.22.64.248.1 > 123.234.164.4.134: S 3825619951:3825619951(0) win 2048
06:04:29.339774 213.22.64.248.1 > 123.234.164.4.135: S 3825619951:3825619951(0) win 2048
06:04:29.344219 213.22.64.248.1 > 123.234.164.4.136: S 3825619951:3825619951(0) win 2048
06:04:29.351000 213.22.64.248.1 > 123.234.164.4.137: S 3825619951:3825619951(0) win 2048
06:04:29.356918 213.22.64.248.1 > 123.234.164.4.138: S 3825619951:3825619951(0) win 2048
06:04:29.362909 213.22.64.248.1 > 123.234.164.4.139: S 3825619951:3825619951(0) win 2048
06:04:29.372312 213.22.64.248.1 > 123.234.164.4.140: S 3825619951:3825619951(0) win 2048
06:04:29.838604 213.22.64.248.1 > 123.234.164.4.141: S 3825619951:3825619951(0) win 2048
...
```

```
<snip>
...
06:04:34.795730 213.22.64.248.1 > 123.234.164.4.509: S 3825619951:3825619951(0) win 2048
06:04:34.799635 213.22.64.248.1 > 123.234.164.4.510: S 3825619951:3825619951(0) win 2048
06:04:34.805618 213.22.64.248.1 > 123.234.164.4.511: S 3825619951:3825619951(0) win 2048
06:04:34.836116 213.22.64.248.1 > 123.234.164.4.516: S 3825619951:3825619951(0) win 2048
06:04:34.840373 213.22.64.248.1 > 123.234.164.4.517: S 3825619951:3825619951(0) win 2048
06:04:34.846396 213.22.64.248.1 > 123.234.164.4.518: S 3825619951:3825619951(0) win 2048
...
<snip>
...
06:04:35.282472 213.22.64.248.1 > 123.234.164.4.536: S 3825619951:3825619951(0) win 2048
06:04:35.288900 213.22.64.248.1 > 123.234.164.4.537: S 3825619951:3825619951(0) win 2048
06:04:35.294430 213.22.64.248.1 > 123.234.164.4.538: S 3825619951:3825619951(0) win 2048
06:04:35.301515 213.22.64.248.1 > 123.234.164.4.539: S 3825619951:3825619951(0) win 2048
06:04:35.316766 213.22.64.248.1 > 123.234.164.4.541: S 3825619951:3825619951(0) win 2048
06:04:35.323586 213.22.64.248.1 > 123.234.164.4.542: S 3825619951:3825619951(0) win 2048
06:04:35.328902 213.22.64.248.1 > 123.234.164.4.543: S 3825619951:3825619951(0) win 2048
...
<snip>
...
06:04:43.378437 213.22.64.248.1 > 123.234.164.4.1022: S 3825619951:3825619951(0) win 2048
06:04:43.385417 213.22.64.248.1 > 123.234.164.4.1023: S 3825619951:3825619951(0) win 2048
06:04:43.391494 213.22.64.248.1 > 123.234.164.4.1024: S 3825619951:3825619951(0) win 2048
```

3 – Outbound UDP traffic (ranges 1-1024 and 2045-2054)

(Due to very large output, only important and example entries are shown)

```
output from: tcpdump -n -i eth1 src net 123.234

04:48:00.375635 123.234.164.2.1 > 213.22.64.248.1:  udp 0
04:48:01.375983 123.234.164.2.1 > 213.22.64.248.2:  udp 0
04:48:02.226772 123.234.164.2.1 > 213.22.64.248.3:  udp 0
04:48:05.296141 123.234.164.2.1 > 213.22.64.248.6:  udp 0
04:48:06.442914 123.234.164.2.1 > 213.22.64.248.7:  udp 0
...
<snip>
...
04:49:05.764861 123.234.164.2.1 > 213.22.64.248.65:  udp 0
04:49:06.793693 123.234.164.2.1 > 213.22.64.248.66:  udp 0
04:49:08.749417 123.234.164.2.1 > 213.22.64.248.68:  udp 0
04:49:10.827952 123.234.164.2.1 > 213.22.64.248.70:  udp 0
04:49:12.065025 123.234.164.2.1 > 213.22.64.248.71:  udp 0
04:49:12.701402 123.234.164.2.1 > 213.22.64.248.72:  udp 0
...
<snip>
...
04:49:49.469329 123.234.164.2.1 > 213.22.64.248.108:  udp 0
04:49:50.652928 123.234.164.2.1 > 213.22.64.248.109:  udp 0
04:49:51.594023 123.234.164.2.1 > 213.22.64.248.110:  udp 0
04:49:53.587131 123.234.164.2.1 > 213.22.64.248.112:  udp 0
04:49:54.446424 123.234.164.2.1 > 213.22.64.248.113:  udp 0
04:49:56.573175 123.234.164.2.1 > 213.22.64.248.115:  udp 0
...
<snip>
...
04:50:37.285656 123.234.164.2.1 > 213.22.64.248.155:  udp 0
04:50:38.759582 123.234.164.2.1 > 213.22.64.248.156:  udp 0
04:50:39.406789 123.234.164.2.1 > 213.22.64.248.157:  udp 0
04:50:40.601858 123.234.164.2.1 > 213.22.64.248.158:  udp 0
04:50:42.204152 123.234.164.2.1 > 213.22.64.248.159:  udp 0
04:50:44.218583 123.234.164.2.1 > 213.22.64.248.160:  udp 0
04:50:45.625481 123.234.164.2.1 > 213.22.64.248.163:  udp 0
```

```
04:50:46.881838 123.234.164.2.1 > 213.22.64.248.164:  udp 0
04:50:47.509928 123.234.164.2.1 > 213.22.64.248.165:  udp 0
04:50:48.643033 123.234.164.2.1 > 213.22.64.248.166:  udp 0
04:50:49.418813 123.234.164.2.1 > 213.22.64.248.167:  udp 0
04:50:50.625118 123.234.164.2.1 > 213.22.64.248.168:  udp 0
...
<snip>
...
05:05:18.388312 123.234.164.2.1 > 213.22.64.248.1018:  udp 0
05:05:19.515413 123.234.164.2.1 > 213.22.64.248.1019:  udp 0
05:05:20.635243 123.234.164.2.1 > 213.22.64.248.1020:  udp 0
05:05:21.715170 123.234.164.2.1 > 213.22.64.248.1021:  udp 0
05:05:22.829881 123.234.164.2.1 > 213.22.64.248.1022:  udp 0
05:05:23.892234 123.234.164.2.1 > 213.22.64.248.1023:  udp 0
05:05:24.556698 123.234.164.2.1 > 213.22.64.248.1024:  udp 0

05:30:20.374198 123.234.164.2.1 > 213.22.64.248.2045:  udp 0
05:30:45.751750 123.234.164.2.1 > 213.22.64.248.2046:  udp 0
05:30:49.589312 123.234.164.2.1 > 213.22.64.248.2047:  udp 0
05:30:52.080452 123.234.164.2.1 > 213.22.64.248.2048:  udp 0
05:31:19.204145 123.234.164.2.1 > 213.22.64.248.2050:  udp 0
05:31:23.003676 123.234.164.2.1 > 213.22.64.248.2051:  udp 0
05:31:32.041850 123.234.164.2.1 > 213.22.64.248.2052:  udp 0
05:31:35.645896 123.234.164.2.1 > 213.22.64.248.2053:  udp 0
05:31:39.246828 123.234.164.2.1 > 213.22.64.248.2054:  udp 0
05:31:42.151872 123.234.164.2.1 > 213.22.64.248.2055:  udp 0
```

4 – Inbound UDP traffic (ranges 1-1024 and 2045-2054)

(Due to very large output, only important and example entries are shown)

output from: tcpdump -n -i eth0 src **213.22.64.248**

```
07:42:36.252816 213.22.64.248.1 > 123.234.164.2.1: udp 0
07:42:37.287511 213.22.64.248.1 > 123.234.164.2.2: udp 0
07:42:38.300543 213.22.64.248.1 > 123.234.164.2.3: udp 0
07:42:39.316586 213.22.64.248.1 > 123.234.164.2.4: udp 0
07:42:40.335844 213.22.64.248.1 > 123.234.164.2.5: udp 0
07:42:41.355466 213.22.64.248.1 > 123.234.164.2.6: udp 0
07:42:42.369089 213.22.64.248.1 > 123.234.164.2.7: udp 0
07:42:43.407071 213.22.64.248.1 > 123.234.164.2.8: udp 0
07:42:44.416785 213.22.64.248.1 > 123.234.164.2.9: udp 0
...
<snip>
...
07:43:42.559886 213.22.64.248.1 > 123.234.164.2.66: udp 0
07:43:43.579626 213.22.64.248.1 > 123.234.164.2.67: udp 0
07:43:44.600330 213.22.64.248.1 > 123.234.164.2.68: udp 0
07:43:46.640137 213.22.64.248.1 > 123.234.164.2.70: udp 0
07:43:47.664082 213.22.64.248.1 > 123.234.164.2.71: udp 0
07:43:48.684299 213.22.64.248.1 > 123.234.164.2.72: udp 0
...
<snip>
...
07:44:25.403415 213.22.64.248.1 > 123.234.164.2.108: udp 0
07:44:26.424280 213.22.64.248.1 > 123.234.164.2.109: udp 0
07:44:27.443324 213.22.64.248.1 > 123.234.164.2.110: udp 0
07:44:29.490233 213.22.64.248.1 > 123.234.164.2.112: udp 0
07:44:30.504611 213.22.64.248.1 > 123.234.164.2.113: udp 0
07:44:31.536569 213.22.64.248.1 > 123.234.164.2.114: udp 0
...
<snip>
...
07:45:13.339719 213.22.64.248.1 > 123.234.164.2.155: udp 0
```

```
07:45:14.361976 213.22.64.248.1 > 123.234.164.2.156: udp 0
07:45:15.380785 213.22.64.248.1 > 123.234.164.2.157: udp 0
07:45:16.410411 213.22.64.248.1 > 123.234.164.2.158: udp 0
07:45:17.429513 213.22.64.248.1 > 123.234.164.2.159: udp 0
07:45:18.453235 213.22.64.248.1 > 123.234.164.2.160: udp 0
07:45:21.509487 213.22.64.248.1 > 123.234.164.2.163: udp 0
07:45:22.537454 213.22.64.248.1 > 123.234.164.2.164: udp 0
07:45:23.552597 213.22.64.248.1 > 123.234.164.2.165: udp 0
07:45:24.577636 213.22.64.248.1 > 123.234.164.2.166: udp 0
07:45:25.587747 213.22.64.248.1 > 123.234.164.2.167: udp 0
07:45:26.606270 213.22.64.248.1 > 123.234.164.2.168: udp 0
...
<snip>
...
07:59:55.693226 213.22.64.248.1 > 123.234.164.2.1020: udp 0
07:59:56.714424 213.22.64.248.1 > 123.234.164.2.1021: udp 0
07:59:57.745078 213.22.64.248.1 > 123.234.164.2.1022: udp 0
07:59:58.750899 213.22.64.248.1 > 123.234.164.2.1023: udp 0
07:59:59.779894 213.22.64.248.1 > 123.234.164.2.1024: udp 0

04:57:20.037379 213.22.64.248.1 > 123.234.164.2.2045: udp 0
04:57:34.019256 213.22.64.248.1 > 123.234.164.2.2046: udp 0
04:57:51.200641 213.22.64.248.1 > 123.234.164.2.2047: udp 0
04:58:05.153070 213.22.64.248.1 > 123.234.164.2.2048: udp 0
04:58:34.569307 213.22.64.248.1 > 123.234.164.2.2050: udp 0
04:58:52.711604 213.22.64.248.1 > 123.234.164.2.2051: udp 0
04:59:05.443407 213.22.64.248.1 > 123.234.164.2.2052: udp 0
04:59:18.286084 213.22.64.248.1 > 123.234.164.2.2053: udp 0
04:59:38.059711 213.22.64.248.1 > 123.234.164.2.2054: udp 0
```