Global Information Assurance Certification Paper

Firewalls, perimeter protection and vpns practical assignment version 1.6

by

Jim Hendrick

Submitted in partial fulfillment of the requirements for

Firewall Analyst (GCFW) Certification

GIAC

November 15, 2001

SANS GIAC

Assignment

GCFW Practical Assignment version 1.6 (Revised august 13, 2001)

by Jim Hendrick

This assignment consists of four parts, which I include here verbatim from the GIAC web site:

Assignment 1 - Security Architecture (15 points)

Design a security architecture for GIAC Enterprises, an e-business which deals with the online sale of fortune cookie sayings. Your architecture must include the following components:

1. Filtering routers;
2. Firewalls;
3. VPNs to business partners;
4. Secure remote access; and
5. Internal firewalls

Your architecrure must consider access requirements (and restrictions) for:

- Customers (the companies that purchase bulk online fortunes);
- Suppliers (the authors of fortune cookie sayings that connect to supply fortunes);
- Partners (the international partners that translate and resell fortunes).

Include a diagram or set of diagrams that shows the layout of GIAC Enterprises' network and the location of each component listed above. Provide the specific brand and version of each perimeter defense component, the security function or role it carries out, and how the placement of each component on the network allows it to fulfill this role.

Assignment 2 - Security Policy (35 points)

Part 1 – Define Your Security Policy (25 points)

Based on the security architecture that you defined in Assignment 1, provide a security policy for AT LEST the following three components:

- Border Router
- Primary Firewall
- VPN

You may also wish to include one or more internal firewalls to implement defense in depth or to separate business functions.

By 'security policy' we mean the specific Access Control List (ACL), firewall ruleset, IPSec policy, etc. (as appropriate) for the specific components used in your architecture. For each component, be sure to consider internal business operations, customers, suppliers and partners. Keep in mind that your are an E-Business with customers, suppliers, and partners – you MAY NOT simply block everything!

You must include the complete policy (ACLs, ruleset, IPSec policy) in your paper. It is not enough to simply state "I would include ingress and egress filtering…" etc. The policies may be included in an Appendix if doing so will help the "flow" of the paper.

(Special note VPNs: since IPSec VPNs are still a bit flaky when it comes to implementation, that component will be graded more loosely than the border router and primary firewall. However, be sure to define whether split-horizon is implemented, key exchange parameters, the choice of AH or ESP and why. PPP-based VPNs are also fully acceptable as long as they are well defined.)

Part 2 – Security Policy Tutorial (10 points)

Select **one** of the three security policies defined above and write a tutorial on how to implement the policy. Use screen shots, network traffic traces, firewall log information, and/or URLs to find further information as appropriate. Be certain to include the following:

1. A general explanation of the syntax of format of the ACL, filter or rule for your

device.
2. A general description of each of the parts of the ACL, filter or rule.
3. An general explanation of how to apply a given ACL, filter or rule.
4. For each ACL, filter or rule in your security policy, describe:

    - The service or protocol addressed by the rule, and the reason this service might be considered a vulnerability.
    - Any relevant information about the behavior of the service or protocol on the network.
    - If the **order** of the rules is important, include an explanation of why certain rules must come before (or after) other rules.

5. Select three sample rules from your policy and explain how you would test each rule to make sure it has been applied and is working properly.

Be certain to point out any tips, tricks or potential problems ("gotchas").

Assignment 3 - Audit your Security Architecture (25 points)

You have been asked to conduct a technical audit of the **primary firewall** (described in Assignments 1 and 2) for GIAC Enterprises. In order to conduct this audit, you will need to:

1. Plan the audit. Describe the technical approach you recommend to assess the firewall. Be certain to include considerations such as what shift or day you would do the assessment. Estimate costs and level of effort. Identify risks and considerations.
2. Conduct the audit. Using the approach you described, validate that the primary firewall is actually implementing GIAC Enterprises' security policy. Be certain to state exactly how you do this, including the tools and commands used. Include screen shots in your report if possible.
3. Evaluate the audit. Based on your assessment (and referring to data from your assessment), analyze the perimeter defense and make recommendations for improvements or alternate architectures. Diagrams are strongly recommended for this part of the assignment.

Note: DO NOT simply submit the output of nmap or a similar tool here. It is fine to use any assessment tool your choose, but you must annotate/explain the output.

Assignment 4 - Design under Fire (25 points)

The purpose of this exercise is to help you think about threats to your network and therefore develop a more robust design. Keep in mind that the next certification group will be attacking your architecture!

Select a network design from any previously posted GCFW practical (http://www.sans.org/giactc/gcfw.htm) and paste the graphic into your submission. Be certain to list the URL of the practical you are using. Design the following three attacks against the architecture:

1. An attack against the firewall itself. Research and describe at least **three** vulnerabilities that have been found for the type of firewall chosen for the design. Choose **one** of the vulnerabilities, design an attack based on the vulnerability, and explain the results of running that attack against the firewall.
2. A denial of service attack. Subject the design to a theoretical attack from 50 compromised cable modem/DSL systems using TCP SYN, UDP or ICMP floods. Describe the countermeasures that can be put in place to mitigate the attack that you choose.
3. An attack plan to compromise an internal system through the perimeter system. Select a target, explain your reasons for choosing that target, and describe the process to compromise the target.

In designing your attacks, keep the following in mind:

- The attack should be **realistic**. The purpose of this exercise is for the student to clearly demonstrate that they understand that firewall and perimeter systems are not magic "silver bullets" immune to all attacks.
- The attack should be **reasonable**. The firewall does not necessarily have to be impenetrable (perfectly configured with all of the up-to-the-minute patches installed). However, you should not assume that it is an unpatched, out-of-the-box firewall installed on an unpatched out-of-the-box OS. (Remember, you designed GIAC Enterprises' firewall; would you install a system like that?)
- You **must** supply documentation (e. g. , a URL to the security bulletin, bugtraq archive, or exploit code used) for any vulnerability you use in your attack.
- The attack does not necessarily have to succeed (though a successful attack is often the more interesting approach). If, given the perimeter and network

configuration you have described above, the attack would fail, you can describe this result as well.

Table of Contents

Acknowledgments

I would like to thank my wife Lynda for her incredible patience with me while I worked on this practical. In addition to “the project” as it came to be known consuming all my free time (meaning I could be counted on to do even less dishes and dump runs than usual ;-) she found my coming to bed at 2 or 3 AM night after night made me even more of a joy to be with when I needed to be woken up at 5 to get ready for work. And every time she simply wanted to check her email, she found that the “family PC” had been left locked when I had finished work for the night.

I also wish to express thanks to the folks at my “real job” for allowing me the opportunity to juggle my schedule a bit to spend more time on this assignment when “crunch time” came around.

Assignment 1

GIAC Security architecture

To design the security architecture for GIAC Enterprises, certain assumptions needed to be made about the business. Particularly, there is no mention in the assignment about the size of GIAC. I proceeded in the general format as if I were performing this for a "real" company, including providing some base assumptions I made in order to better frame my solution.

I also wish to remind (nee. Beseech) the reader to remember the bumper sticker that became popular in the 70's "QUESTION AUTHORITY" in taking this or any other security document as a sure-fire plan to protect themselves or their employers' from being cracked. I would hasten to add that there are many excellent sources of information on the subject, and that these *GIAC practicals* are one of the best resources I have ever seen (I reference some of the best at the end of this paper). However, there is no guarantee that the information presented here (or elsewhere) will be suitable for a different environment. Please take this information as a starting point, but validate everything you read yourself. There is no substitute for having a set of machines to "play with" to try some of the things you find here. Whether this document is accepted towards my certification or not, the work it caused me to do in testing and trying many new things on my own systems, actually generating the packets and seeing which ones "got through" has been one of the best learning experiences I have ever had. (Not to mention a heck of a lot of fun!!!)

Planning process: Business and technical information gathering.

In creating any systems architecture, (whether specifically designed for security or simply integrating a new application) it is necessary to understand the actual needs of the company. No matter how good it is, if it isn't what the customer wanted… ("We're sorry you feel that way, but we did want a block of flats, nice though the abattoir is."[1]). To do this, a series of meetings would be conducted with various levels of the GIAC staff. These meetings would give us an understanding of the scope of work. In general, the procedure would begin with

[1] "The Architect Sketch", Monty Python's Flying Circus http://www.montypython.net/scripts/architec.php3

the senior management team and then proceed to meet with the rest of the team GIAC has available to help understand their ideas for this project. My goal for these meetings would be to:

- Identify and document relevant business and technical assumptions. (e.g. What are their needs and how much are they willing/interested in spending? What are their expectations from hiring me to do this architecture? (i.e. what do **they** expect me to deliver) Do they have any designs or partial designs they would like considered? What sort of in-house resources exist? What role will their staff be playing during and after the architecture is delivered?)

- Identify the potential risks to the business and prepare a risk assessment that would be reviewed with the client.

- Design the architecture to address the requirements and mitigate against the identified risks.

- Get GIAC to buy in to the architecture. Make any "fine tunings" needed that do not compromise the business requirements.

To simulate that process, I will make a number of assumptions along the way to facilitate the assignment. This will allow me to conduct a greatly simplified version of a "risk assessment" and demonstrate how to use it to define aspects of the design as would actually be done. Key to this approach is to involve the client with the decision process so they will understand and feel part of the decisions that are being made. Making a successful implementation much more likely.

Assumptions on completing the Practical:

In addition to stating the "business assumptions" about GIAC Enterprises, I want to describe the resources I have available as the student to design and implement the technical components to this assignment (specifically the security architecture, tutorial and security audit). These also drove some of the technical selections I made in this work, and I attempted to make the "business assumptions" match my resource availability in a somewhat realistic fashion. Where possible, I formatted the document so that the security requirements were first abstracted from the pragmatic realities, and then the technical specifics detailed (with consideration noted in areas where they affect or limit the solution). In the "Audit" section, I present alternatives that would improve the design, but were not available to me personally.

In fulfilling the requirements for this assignment, I had the use of a small number of Windows systems to play the roles of various "Internet" and "Intranet" machines when developing and testing rules. My ability to modify them was somewhat limited, as they have other "real" roles to play and needed to be kept (basically) in their original configuration (although one of them is dual-boot with an older version of Linux). In addition, I have a dedicated RedHat Linux system with three NICs that I used to implement the primary firewall to fulfill the "Security Policy Tutorial" (Assignment 2 Part 2) and to allow me to conduct the "Audit" section (Assignment 3).

While having access to a networking lab with lots of commercial routers or firewalls would have been great, I chose an architecture around PC hardware and believe this gave me a better practical learning experience than simply implementing something using a commercial interface. I had (still have) limited experience with Linux, but a fairly broad background in a number of UNIX variants. I felt this would allow me to complete the practical assignment and gain significant experience with implementing firewalls. Hopefully, this approach will also provide some useful information to the security community at large (that you don't need a huge budget or lots of Commercial $oftware to do this.)

Additional caveat. I do not have access to any Cisco gear, so the router configuration will be left in general policy terms. I derived all example IOS rules and ACLs from documentation and other on-line references (which were credited where they occur). I could not include any screen shots or personally test IOS code.

So, Let's Begin!

Here are some "Business assumptions" that will make this seam more or less reasonable.

Company Backgrounder:

GIAC Enterprises is a provider of fortune cookie sayings. Their business model is based on the fact that their product is essentially "intellectual property" that can easily be distributed electronically. Rather than preparing and distributing physical copies of their "fortunes", market research has determined that by providing this information over the Internet to their customers (the companies that bake and sell fortune cookies), GIAC can gain several key business advantages:

- access to a wider market than with traditional distribution channels, and gain a

significant portion of market share

- contract with suppliers more cheaply by allowing the authors of fortunes to work at home, saving GIAC the cost of providing office space, equipment, etc.

- the ability to provide the same fortunes in multiple markets around the world by partnering with companies that provide some combination of translation and distribution services

In this way, GIAC hopes to leverage technology to improve dramatically over traditional profit margins and react quickly to changes in demand for their product, providing a profitable business opportunity.

This small but growing business was founded by a group of (fairly technical) creative folks with very little initial capital. Shy of taking "angel funding" from venture capitalists, they have decided to test the waters themselves by developing their business infrastructure using inexpensive components during their initial growth period. To assist with this, they have recruited a small number of IT staff that are somewhat familiar with Linux and have decided to base most of their initial infrastructure on it.

Due to the long-life of the value of their product (they intend to develop a growing database of fortune cookie sayings) they realize that information security is paramount. Therefore, they have decided to implement a network infrastructure that will allow them to pursue their business while (hopefully) protecting their most valuable assets.

Your mission, should you decide to accept it, is to design and guide the implementation of their network architecture. If discovered, the restaurant manager and the chef will disavow any knowledge of your actions

(not to mention, giving you a really bad plate of egg-foo-young).

They also realize that they may incur rapid growth and have commissioned the design for their network security architecture with this flexibility in mind. Their requirements are for a security policy to be developed that allows maximum flexibility going forward. Once these initial policy requirements are developed, several architectural designs will be considered. Because of their financial limitations, when a technology choice is made, it will be noted what conditions

might actually drive this in reality (and what alternatives there are available if those conditions change).

Security Architecture; GIAC Business Requirements

These requirements would be generated from a series of interviews and round-table discussions with GIAC staff. For the purposes of this assignment, based on the GIAC business model, two basic needs must be provided by the architecture:

1. An internal network where the corporate assets will be located, including the database of fortunes.

2. Internet connections to customers, suppliers and partners.

Internal Services:

The GIAC team intends to begin business with a relatively "open" policy for their employees. They wish to allow unrestricted access to the Internet. When pressed on specifically which services would be required, and warned about the risks with completely open access, they agreed to limit access (initially) to simply HTTP and HTTPS. They did however require that the architecture allow for addition of other services if they were deemed useful.

The desktop platform choices are to be left up to the individual departments, and it is understood already that there will be a mix of Windows (in the Finance and Sales departments at least) and Macintosh (in Marketing). Development (the authors) will also probably make fairly heavy use of Windows, but there is a reasonable contingent of Linux users as well. Their IT department is struggling with the support issues this brings, but at this time has no plans to force a standard desktop.

Basic User Services:

- Email – GIAC will need email for employees as well as to and from the Internet.

- Intranet Web services – Internal servers need to be supported so that access to GIAC corporate information can be provided (by each department publishing their own content if desired)

- Internet Access (open Web access for employees initially, but needs may vary by department policy)

Other Utility Services:

- Time synchronization – NTP servers will be provided internally as well as to machines on the service networks.

- Name services (DNS) – We recommend that GIAC implement a split-DNS name service. All external machines will be listed only in external nameservers. All internal machines will be listed only in internal nameservers. Internal requests will be allowed to be forwarded to the external nameservers for resolution, but no external queries of the internal servers will be allowed. (To be completely correct, there will be the need for service net machines to communicate with some internal servers for mail, name resolution, time and database functions. These servers will be listed in static hosts files and not appear in DNS)

Support for the following departmental business functions:

Finance

Finance will need to maintain the corporate ledger and have access to information from all departments on all aspects of budget planning and tracking. They provide accounts payable and receivable and especially need access to all sales information and generate payroll on a regular basis. They will be (separately) undertaking a project to identify and implement this business system. Their requirements are that the network architecture will provide security for this system, and allow it to interact with other business partners and suppliers as necessary.

Marketing

Marketing will be creating programs for advertising and promoting sales. Initially, they will also be chartered with doing business development and seeking out partnerships, although there is a plan to split this off into a separate "Business Development" department as the company grows. They create and maintain the content for the GIAC Internet web site, and create multimedia designs for advertising campaigns. Their requirements for the network architecture are that they have full ability to update and control the GIAC web sites and access partners' networks as business relationships develop.

Sales

Sales needs to execute marketing plans, take orders, provide fulfillment of product. They will need full ability to control and manage any e-commerce services, and to access financial data for order tracking and fulfillment. Remote sales staff will need access to the network from anywhere on the Internet.

Human Resources

Provides traditional HR services to the company. Maintains an employee database, a resume tracking system, and training services.

Product Development

Internal development consists of authoring, editing and translation services. It must have access to the production database as well as allow access for remote developers.

Legal

As GIAC needs to deal with oversees partners, they will need to make use of legal services. Currently, they are contracting an external firm “on retainer” to provide this function, and no specific network needs are anticipated with the exception of secure transmission of contracts.

Internet Services

From the requirements gathering meetings, it is clear that Internet services can be divided into three main service areas that must be provided by the architecture:

Customer Services

This is the main sales distribution channel for GIAC Enterprises. All connections to customer sites must be secure and reliable. Services will primarily be HTTPS based with authentication required for download of purchased files. The need for the files themselves to be encrypted is still potentially “on the table”.

It would be recommended (although outside the scope of this assignment) that something like PGP be used wherever any durable information assets is transmitted or stored. This is made much more acceptable with the availability of “self decrypting archive” file types, so the recipient would not need to be PGP enabled to take at least partial advantage of this technology. PGP also integrates

with email, and would provide for the requirement that “Legal Contracts” be transmitted securely.

Supplier Services

All remote employees will require access to services as if they were on the internal network. All services are to be available: email, access to the internal web site(s), access to the company database, HR services, etc.

Partner Services

Secure connections must provide partners with access to authorized sections of the production database as well as to other business services (subsets of financial, sales & marketing resources).

Public Internet Services

In addition to these three services, GIAC will provide a basic set of Public Services that are available to the Internet community (i.e. are not restricted to customers). These will include the primary website “www.giacfortunes.com”, their email gateway “mail.giacfortunes.com” and domain name services at “dns.giacfortunes.com”.

Security Architecture: Risk Analysis

Now that the basic business needs (to be provided by the network) have been identified, the next step is to determine what the architecture should protect *against*. This will drive the next major decisions.

Although the process of conducting a risk analysis is outside the scope of this document, the basic method used here is a fairly standard one. The degree to which this is done is a matter for the particular situation. In some cases, a quick review by a single person with rough guesses is fine. In others, the estimates themselves will require separate research over longer periods of time. In this case, it would consist of meetings with all major GIAC department heads in “round table” sessions to address the following items:

- Identify as many of the potential risks as possible (from the network or to network resources).
- For each one, place a cost on the loss associated with the risk. (In this case, they will be “low medium and high” estimates, but as long as they are correct relative to one another, it is sufficient for the analysis)
- Determine a likelihood that each risk will occur. (again, here we will use “low

medium and high")

- Now "calculate" an expected value (In this simplistic case, we do not show this data In other more detailed analysis, it is the product of the loss times the likelihood of that loss.)
- Determine a method (or methods) to mitigate against each risk.
- Now assign a cost for each of the options for guarding against each risk as well.

Using this data, the network design will be developed to minimize the "most likely and most costly" risks as well as to make sure that what resources are available can be used where they can do the most good. This information will help make design choices that take advantage of any potential areas of overlap. For example, there may be two risks, with expected cost of loss per year of $1000 and $2000 respectively. In each case, the same means of protection may guard against that risk, but the cost may be more than $2000 per year. However, since the same means provides protection in two places, as long as the total cost is less than $3000/yr, doing this analysis will actually show the cost to be justified.

At this point in conducting the analysis, (and for the purpose of this exercise) since GIAC is a small start-up, we will use "common sense" to identify the general categories of risk to the main business functions and whether the likelihoods are "Low Medium or High". This will give us some early guidance on design and policy choices. Once the architecture has been defined, further detail can be added to address the implementation-specific risks.

Table 1: Risk Analysis Matrix

Risk	How Likely	How Costly	How to Mitigate	Cost to Mitigate
DOS to Customer Link (i.e. Internet distribution link is attacked)	Med	High	Configuration of link router & firewall Implement redundant links & servers	Low High (2X or more that of single link)
Intrusion: Customer Link (i.e. gain control of the router/firewall/servers)	Med	High	Choice & configuration of equipment Stay current on patches & known vulnerabilities Implement IDS	Low – med Medium (time investment) Low – (using free IDS)
DOS: Supplier Link (i.e. attack on remote access VPN)	Low	Low	Choice & configuration of equipment Implement redundant remote access paths.	Low High (2X)
Intrusion: Supplier Link (i.e. attack through supplier link or attack on VPN system itself)	Low	High	Harden remote access system Implement IDS Limit internal resource availability to legitimate remote clients	Low Low Low – medium (can hurt remote productivity)
DOS: Partner Link	Low	High	IDS on partner VPNs Redundant partner links	Low High
Intrusion: Partner Link (i.e. break into GIAC net via. Partner VPN net)	Low	High	Internal Firewall protecting internal resources. Internal network IDS in place to detect improper access attempts from partner networks. Authenticate at eachapplication (don't trust by IP address.	Med – Cost of an added firewall. Low Low – Med (administrative cost to manage accounts)
DOS: Public Link (i.e. attack on "free" services)	High	Med	Configuration of router , firewall & public servers. Implement redundant link & servers.	Low High
Intrusion: Public Link	High	High	Primary firewall. Internal IDS.	Low

Once this risk matrix has been created (clearly, this one is very simplistic) it will be used to guide the design process. Please keep in mind that the decisions are up to the client as to how they wish to balance the business risks against cost of mitigation. My job before I run off and design a network in a vacuum (and your job if you are doing this for real and reading this practical for some tips, assuming of course that it passes :-) is to present the risks and options to the client and help them make appropriate decisions.

We would help the client (GIAC) through the process by helping walk them through the options.

For example, even though the potential for a "break in" through a VPN link to one of GIAC's "partners" or "suppliers" is low, the potential damage is high, since it would grant the attacker access to the internal GIAC network. Weighing this against the comparatively low potential damage from a DOS to the "Public" services (or even the customer services), and although the likelihood of the public link being attacked is far greater, GIAC should spend it's (limited) funds on protecting against the VPN Intrusion attacks first. (Author's note: Later in the practical, the decision of commercial VPN gear vs. some of the cheaper alternatives ties back to this analysis.)

We can now see where there is the potential for a single type of protection (or a particular design) to guard against multiple potential risks (thereby getting the most "bang for the buck"). Even in this simplified form, the risk analysis has identified several areas where the risk can be mitigated by implementing some sort of IDS. Since the cost to do this is quite low, (the academic release of "Tripwire"[2] as a host based system coupled with "snort[3]" for network intrusion detection will give a very effective "early warning" capability for very low cost. This can also be enhanced by the choice of network topology (more on this later).

This exercise also serves to highlight the fact that GIAC internal network attacks also pose a great risk. We were given the business requirement to provide access for remote suppliers, customers and partners. However, the risk matrix makes this extremely obvious (and might help to sell a particular design at some other business where upper management may not be as technically savvy as the GIAC folks are). This drives the design toward a more segmented internal network that truly uses "defense in depth" as an architectural principle.

---

[2] Information available at: http://www.tripwire.com/

[3] Information available at http://www.snort.org/

Using this risk analysis technique of breaking down the problem is also a very simple yet effective way to lead a team toward a solution that everyone will more readily accept than if you (as an individual or as part of what is perceived to be a team of "outsiders") presented it as totally your own idea. It is fairly effective at helping weigh alternatives, finally arriving at the design which best balances the business needs with the potential technical solutions.

This tool is also excellent to keep around to be used to revisit the design at a later time (when perhaps the financial state is more solid or the business needs are more mature). The same design choices made initially may no longer be required, or the features of the original design may be re-worked to provide broader services while continuing to meet the security needs. (Consider multiple partners in a more successful GIAC. The "free" host based and network IDSs may become harder to manage with 50 systems instead of 10 and the extra cost of a commercial version of "Tripwire" or something like the "Dragon[4]" products for network based IDS with better configuration and central management may be justified.

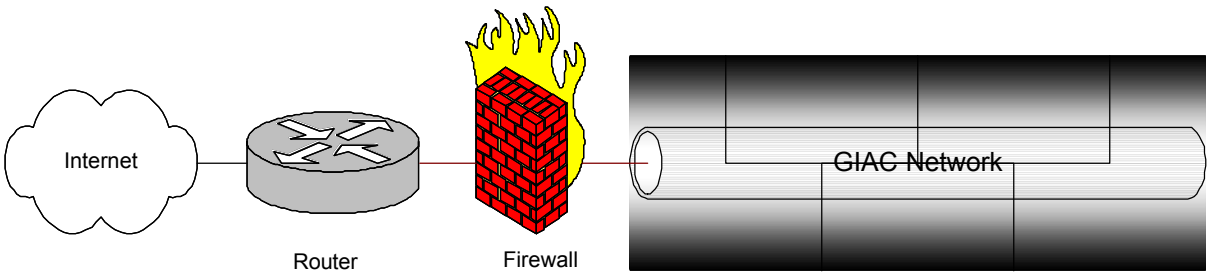
Architecture Options:

In determining which architecture path to go forward on, several basic designs would be presented to GIAC and the major advantages and disadvantages explained at a high level. This is mostly to assist the client in understanding the reason for the recommendation that will be made. These presentations would be very high-level "chalk-talks" and not go into detail about specific technologies or implementation details. This example only addresses one set of services, that of the "customer services" and does not go into detail about the others (Partners, or Suppliers).

Options for Customer Services

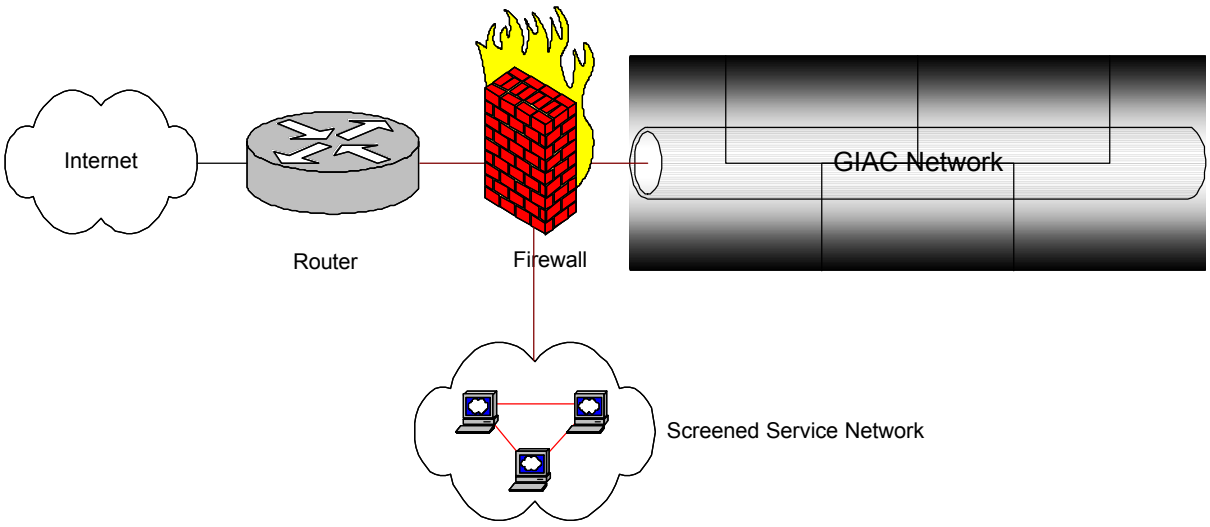
Simple screened network

[4] Information available at: http://www.enterasys.com/ids/



UNACCEPTABLE – placement of services outside of firewall poses significant risk to those services, placement of services inside the firewall opens GIAC private network to risk.

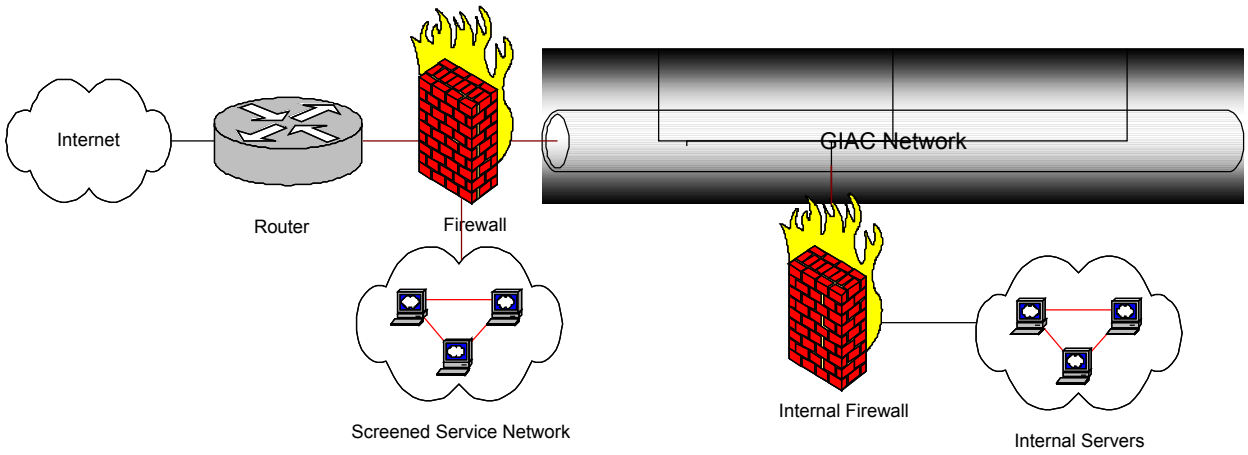
Screened service subnet



UNACCEPTABLE – Although services would be protected and GIAC network would be protected, the main disadvantage is that it treats all internal resources identically. In the GIAC business requirements, they have already stated that they wish their employees to have very open access to the Internet. This means that either:

- All their critical resources live on the same network as their employees (which has great risk due to the likelihood that the employee machines may become compromised at some point).
- Or all the critical resources must live on the service net. This is also very dangerous, because there is a risk that the firewall may not be perfect in protecting these resources. Compromise of this single point of failure would be catastrophic.

Screened service subnet with internal firewall(s)



ACCEPTABLE – Services will be protected, GIAC network will be protected, critical resources will be much better able to be secured. Failure of the external firewall or compromise of internal employee machines will not (immediately) result in the critical servers being dangerously vulnerable.

Options for remote supplier access (VPN):

VPN using SSH tunneling:
One option which would provide a very low-cost means to access specific services over an encrypted tunnel would be to implement SSH tunneling. Although it would provide access to simple services (e.g. mail) it would be potentially difficult to administer for non-technical remote workers.

UNACCEPTABLE – Implementing this for partners and suppliers would be very complicated and high-maintenance. It is unlikely that partner companies would have the technical staff to manage this on their end.

Other Free VPN software

There are several free VPN solutions available including Linux FreeS/WAN[5] implementation of IKE and IPsec.

UNACCEPTABLE – Similar to VPN tunneling over SSH, this would be more complicated and not fulfill the requirements of most potential business partners.

MS VPN using native support:

For remote supplier access this would be a possibility since it is natively supported on newer Windows platforms. There are clearly risks to implementing MSCHAP. Version 1 had a number of security bugs <<list references>> and version 2 still limits the strength of the encryption on the individual password. This would also be prohibitive for non-Windows remote access. It would also be unlikely to meet partner access requirements in very many cases.

UNACCEPTABLE – Due to security risks and platform limitations.

Commercial VPN HW/SW
There are several good offerings in this area which (although stretching the budget of this startup situation) provide good remote access VPNs. Several companies offer products (Cisco VPN Client, CheckPoint SecureRemote and the Enterasys Aurorean Client to name three) in this area.

ACCEPTABLE – Due to the support provided by commercial vendors, it will simplify the management of remote access suppliers to go with one of these.

Managed "ISP provided" Service
Although not appropriate for this practical, a small business with tight budget that still needs to provide supported VPNs for remote individual as well as site-to-site connections should investigate the offerings in this area. There are several advantages, including not having to invest in expensive HW/SW, not needing to provide 24x7 technical support, and the ability to "pick and choose" a VPN provider based on experience and resources in the target geographical area.

UNACCEPTABLE – Violates requirements of the practical assignment.

Options for partner access (VPN):
Leased lines
Dropping a dedicated link from network to network and filtering traffic on either end with a firewall may in some cases still be a reasonably secure and cost

[5] Information about this project is at http://www.xs4all.nl/~freeswan/

effective solution for tight integration with close partners. (It can support much better latency and guaranteed availability than over shared-access media like the Internet.)

UNACCEPTABLE – Requirements specify a VPN based solution. In fact, even traffic over leased lines should be encrypted if security **really** matters. Once data leaves any device under your control, you cannot be certain of the confidentiality/authenticity.

HTTPS Extranet site:
Depending on what services are needed, there are several options here that could be provided using a dedicated SSL web site and no VPN at all. We assume that partners will need to simply exchange files and financial transactions, use internal e-mail and access GIAC internal web sites, etc. All this is quite possible using a simple SSL web site with no VPN at all and either mirroring the internal resources to this site or providing an encrypted reverse-proxy.

UNACCEPTABLE – Violates requirements for VPN solution.

Internet site-to-site VPN

Implementing site-to-site VPNs using local HW/SW seems to be the only option. As in the individual remote access option, there are a number of commercial offerings (Cisco, CheckPoint, Enterasys) that will work here. If this is recommended, the decision should be made based on input from any existing or potential partners, as well as specific product features and cost.

ACCEPTABLE – The solution meets technical requirements and the added HW/SW cost will be balanced by the support provided by the vendor.

Recommended Architecture Design:

Now we have reviewed the potential designs with the client, and are ready to present them with our recommendations.

Lessons from the Risk Analysis

In reviewing the risk analysis, it appears that the greatest business risk is that of intrusion where an attacker gains access to the GIAC fortune database. This data

must also be readily accessible to Suppliers (adding new sayings, editing databases of existing ones), Partners (downloading finished sayings for distribution, downloading sayings for translation and distribution, uploading translated or submitted sayings) and Customers (purchasing the finished cookie sayings). Therefore it is recommended that these databases be placed in the highest security zone behind multiple firewalls. All access from either remote suppliers or business partners will be managed using carefully configured VPNs and monitored using both host based and network IDS.

The second largest threat is from DOS attacks against either the Customer or Partner services, although the highest "likelihood" of attack is to the public services (simply due to the open nature of these servers, i.e. they are *supposed* to attract lots of people).

Network Architecture: Multiple Links, Segmented Network, Multiple Firewalls

To meet both of these needs, it is recommended that GIAC obtain multiple links to different ISPs. One for each of the three main sets of services: Public, Customer and Partner/Supplier. Note that when selecting ISPs for this purpose, take care to think about the physical route that is used by the ISP. It does little good to have separate links that end up sharing an upstream path (whether it is smaller ISPs that both buy bandwidth from the same upstream provider or larger ISPs whose lines share the same physical path (i.e. the one right in front of the same backhoe?). This will:

- make DOS attacks far less effective since the targets themselves are distributed. Even though a coordinated effort against all the ISPs at once is certainly possible, it is highly unlikely that a fortunecookie company would be the target of that type of attack.

- reduce the impact of any outage or vulnerability/attack affecting a single ISP. Separate from intentional disruption, an accident causing loss of service at any single ISP would only affect the services it directly provided. Although not included in the initial design, it would be possible to connect the border routers using a high-availability protocol (HSRP or VRRP for example) so that automatic re-routing of traffic would occur.

- isolate traffic so that high load on any single link does not affect other services. This addresses more the "bursty" nature of Internet traffic, isolating the affects of heavy public web hits (caused by a large lottery jackpot inducing millions of people to come to ww.giacfortunes.com for the free "lucky lottery numbers" page) or by a large amount of traffic from GIAC employees (around the

superbowl?)

- allow each set of services to be managed more easily, reducing the complexity of each firewall. By placing each type of service behind a separate firewall, it will greatly simplify the rules and reduce the likelihood of an incorrect configuration. This is not to be taken lightly since even though we know the initial install will be thoroughly tested, the long-term test of the architecture also includes maintainability. Having simpler configurations will make the administrators' lives much easier.

- allow GIAC greater flexibility and leverage in negotiating contracts with each ISP. If they know they are not the only vendor you are dealing with (especially if they know you will have connections through other providers) they are much more likely to treat you well if they want your continued business.

- allow GIAC to selectively source services where it makes sense (e.g. if one ISP provides better managed VPN hosting than another, this service can be moved with little impact on others)

The three links will be:

1. A "Public Link" – An attacker would most likely attempt to "hit" the public services but even a successful attack would not affect partner, customer or supplier access.

   Services provided will include:

   - www.giacfortunes.com - the public web site where company information, sales and marketing materials, career information and any other free services will be located here.

   - mail.giacfortunes.com – normal email in and out of the company will pass through this link.

   - dns.giacfortunes.com – the primary "public half" of split DNS will reside here. Since there is a very small set of external addresses that will ever need to be made public, no large transfers will be allowed (i.e. no TCP). A DNS secondary will be hosted offsite.

   - ntp.giacfortunes.com – the time service will reside on this link as well. Initially, it will be co-resident on the DNS server, and all other servers on

this net will “chime” from it and “peer” with it so that in the case of loss of external service, they will keep one another internally consistent. The external connections should be authenticated, although this is not a “must have”.

2. A “Customer Link” - An attack on the customer link is next most probable, since the address of this server may well be provided (i.e. a link to it probably exists on the public server).

   Services provided will include:

   - commerce.giacfortunes.com – This will be the HTTPS secured web site where all customer transactions will occur. It does not need to allow any direct access to the internal databases (We recommend placing a local database on this network that contains the necessary information to run the sales applications. This will periodically update to the main database inside GIAC.)

3. A “VPN link” - the services for both Partners and Suppliers will reside on a dedicated link.

   Services provided will include:

   - Dedicated links between GIAC and its partner companies. The nice thing about VPN standards is there are so many to choose from. The most reasonable thing to do here is to pick a market leader to provide dedicated VPN hardware and require that business partners have compatible gear. Another option would be to work with a managed VPN service provider (ISP) who drops one end of their encrypted tunnel in both GIAC and GIAC’s partners’ networks.

   - Private Applications for Partners – There is a great deal of content and service that can be provided simply by placing the “presentation layer” of a multi tiered client-server application on this link, reducing the need for further access into the GIAC network from external companies.

   - Secure remote access for suppliers. There are two main options here as well.

     - Provide VPN access directly. Obtain the hardware and software to provide the required services (email, access to internal web

servers, access to create and edit fortunes, possibly file and print resources). This could be done using Cisco or CheckPoint VPN client software and a dedicated hardware platform at GIAC.

- Provide access through a managed VPN service. This has the advantage of lessening the maintenance effort for GIAC directly. It has the disadvantages of being more expensive in recurring costs, and potentially being a security problem, since GIAC would not have control over who else was sharing the links.

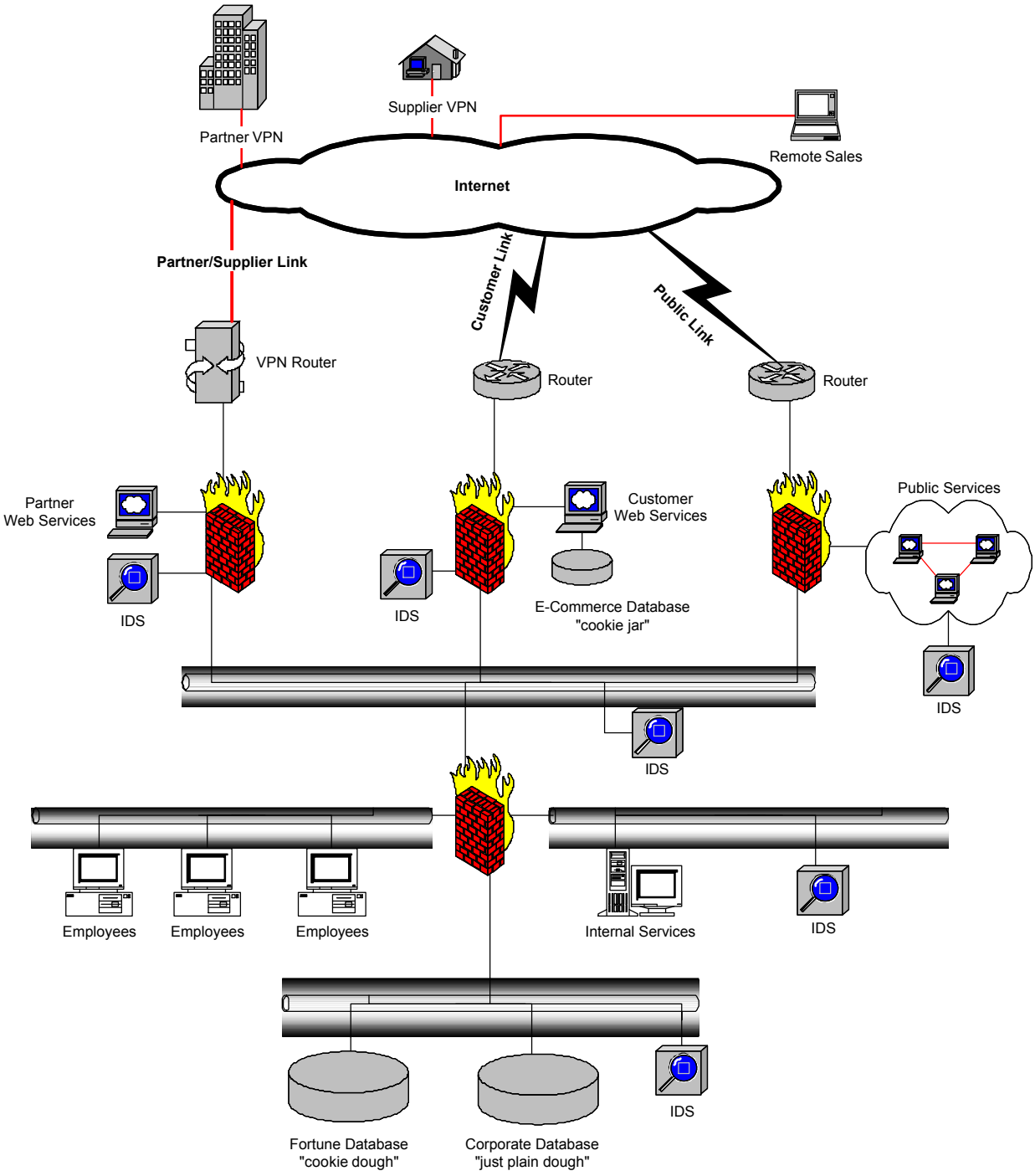
In all cases, the "inside" of this VPN link will feed into a dedicated firewall at GIAC. This has the advantage of providing full filtering of all VPN connections, whether partner or supplier. Since the firewall will reside on the de-crypted side of the VPN box, we are much less vulnerable to attacks that would be hidden if we only had access to the encrypted stream(s).

In addition, since there are no public services provided on this link, it's existence does not need to be publicly known. The domain name may be something unrelated to GIAC (as would be the case with a managed service). Therefore the likelihood of someone choosing this circuit for attack is lowered.

Network based IDS will also reside on each link. Since information security is critical to GIAC, each link will be closely monitored for signs of intrusion. Snort will be configured on a stand-alone PC such that any connections seen outside what is appropriate for the network will be torn-down[6] (i.e. a RST will be sent to both ends of the connection) and those attacks likely blocked.

In keeping with the requirements that this be done as inexpensively as possible, and leveraging the (limited) Linux experience of the GIAC staff, we recommend using standard PCs to implement the majority of firewalls and servers in this design. The total cost for roughly two dozen servers is estimated to be approximately $75,000 for nicely equipped dual CPU units with 1GB RAM and dual 18GB SCSI disks each. The cost of the commercial VPN solution alone will probably be more than this.

[6] "Active Response" or "Flexible Response" allows Snort to take action when specific types of activity is detected.



GIAC Enterprises: Network Architecture

ASSIGNMENT 2: Security Policy

part 1: define your security policy

Security Policy

Defining a security policy includes addressing multiple areas. For the purposes of this assignment, I will skim over many other aspects, and focus on the specifics of the network security policy. Other aspects (how a user policy, admin policy, legal policy, HR policy, incident response policy) would all be addressed with the client, but are outside the scope of this document.

When developing any security policy, a holistic approach must be taken. This seemingly obvious statement is fairly often missed in some fairly surprising situations. I believe there are several reasons for this; poor or no initial thought to security needs, the (sadly incorrect) assumption that "no one will bother me, I have nothing of interest or nothing to hide", the (also flawed) assumption that "we can't afford security" because of the assumption that it either is extremely expensive and requires a huge dollar commitment or the equally flawed assumption that it will "get in the way and prevent us from doing our jobs"[7]. These assumptions are usually either unstated or stated "unofficially" when services are being built, and by the time the need for security is accepted, it is either very difficult to re-engineer the entire set of systems and networks (supporting the nay-sayers position that "we can't afford it" or "it will get in the way") or in some less fortunate situations, it is simply too late. (When you find yourself standing in front of a rack of compromised servers, and the only way out is to pull the plug to the Internet and rebuild them all. By the way, this is entirely the wrong time to get the president of the company to remember that talk you tried to have with him/her to explain that even though they were "right across the street from the fire department" that it wasn't what you meant by "You need a firewall."[8])

[7] One of my pet peeves is the often repeated statement that "Security is Inversely Proportional to Convenience" or something similar. My feeling is that this only reflects poor design and laziness on the part of those developing the security (but hey, I love this stuff).

Now, to correctly design a security policy that addresses all the aspects from human engineering to disaster recovery, secure software design and system administration is entirely outside the scope of this assignment, although it should be addressed in the contract with GIAC in some form. If it cannot be, you had better get a disclaimer in writing, since while "bit rot" is an urban myth, "lazy administrator rot" and "scope creep" are absolutely real and that Shiny New System that was so tightly configured when it was installed tends to look more like the screen door on an old summer camp after a few years (you know, the one with the holes that would let in mosquitoes the size of hummingbirds?)

But, before I go on to the actual network component of the security architecture, let me re-state some Basic Truths ™ about security.[9]

> "**Axiom 1 (Murphy)** A*ll programs are buggy.*
>
> **Theorem 1 (Law of Large Programs)** *Large programs are even buggier than their size would indicate.*
>
> *Proof*: By inspection.
>
> **Corollary 1.1** *A security-relevant program has security bugs.*
>
> **Theorem 2** *If you do not run a program, it does not matter whether or not it is buggy.*
>
> *Proof*: As in all logical systems, **(false ==> true) == true**.
>
> **Corollary 2.1** *If you do not run a program, it does not matter if it has security holes*.
>
> **Theorem 3** *Exposed machines should run as few programs as possible; the ones that are run should be as small as possible.*
>
> *Proof:* Follows directly from Corollaries 1.1 and 2.1.
>
> **Corollary 3.1 (Fundamental Theorem of Firewalls**) *Most hosts cannot*

[8] This actually did happen.

[9] Taken from William R. Cheswick and Steven M. Bellovin *Firewalls and Internet Security, Repelling the Wiley Hacker,* Addison-Wesley, 1994.

meet our requirements: they run too many programs that are too large. Therefore, the only solution is to isolate them behind a firewall if you wish to run any programs at all."

I hope the reader understands my indulgence there, I have never seen it stated better. My point is simply to emphasize the need for defense in depth and for good security practices in general when selecting software or designing systems. The short of it is:

- Run only what you absolutely need to and choose the simplest programs that meet your requirements.
- Separate functions onto different systems where possible.
- Run these (hardened) systems behind a firewall.

Now, on with the show.

Since the recommended design includes multiple links, I will address first a set of global policies that will apply across all links, and then go into detail on each link specific to its services.

Global Security Policies: Principles

When designing the policies to implement any security architecture, there are a number of guiding principles that should be kept in mind. The general framework for the next few sections is similar to that from Chapter 3 of "Building Internet Firewalls"[10]:

Least Privilege

Provide each layer or service with only sufficient privileges to accomplish their roles. Do not grant excess rights or abilities simply for convenience.

- Build layers of trust from the inside out not vice versa. Outer layers are more likely to be compromised first. (e.g. Don't accept an admin session from the external router to the firewall. Don't "trust" a session coming from the firewall into an internal server. Build hardened and monitored administrative systems on the internal network and have the firewalls and routers managed from

[10] D. Brent Chapman and Elizabeth D. Zwicky. *Building Internet FIREWALLS,* O'Reilly & Associates, Inc. 1995

them.)

- Consider this principle when configuring "mundane" services like backups or SW updates. Don't open a hole on the firewall to allow backups without carefully considering the ramifications. Could you connect a local tape drive rather than opening a network service? If you choose to perform backups over the net, consider the security ramifications when choosing/configuring the product. Prefer having the system to be updated to "pull" the updates from a trusted host rather than using "push" technology to drive updates from a central source.[11]

- Application privileges, suid/sgid & sudo. Many applications "assume" they can/will be run as root (unix) or Administrator (Windows). This is one of the most fertile grounds for crackers to exploit. It is definitely worth the time to look at ways to reduce the power granted to any application or "taken" by an application through the "suid" or "sgid" permissions[12]. This goes for administrators as well. How many people need that account? On unix, one of my personal favorite-all-time security tools is "sudo"[13] which allows very fine grained access control over "rootly powers". Actually, it works for any userid that needs to be shared. (Ever want to have more than one DBA have access to the "oracle" account without sharing the password?)

- Consider what a server needs and don't install (or disable) what it doesn't. If the DNS server will never route email, remove sendmail. If the webserver doesn't run CGI programs, remove the directory of them. If a machine will not build code, remove the compilers and make utilities. If administrators will be the only ones running special tools, chown and chmod them (again, my unix is showing) or change the Access List (Windows) so that no one else can read/execute them.

Defense in Depth

"Don't put all your eggs in one basket." Is still true. Don't depend on a single mechanism when designing or implementing security. Too many times poor decisions are made because "We're secure, we're behind a firewall." Design each layer of services with "what happens when the other layers fail" in mind. Take

[11] It is possible to do encrypted backups using a combination of SSH and "rsync". See http://www.stearns.org/rsync-backup/ for details.

[12] Run a "find / -type f \(–perm -004000 –o –perm –002000 \) –print" and see what you find!

[13] The Sudo Main Page is at: http://www.courtesan.com/sudo/index.html

how all layers work together into account when implementing them and determining usage and administration policies and procedures.

Design in Choke Points

Although this seems in opposition to "Don't put all your eggs in one basket." It is a sound strategic practice to force attackers to use a narrow channel, which is more easily controlled/monitored. This must be balanced against creating single points of failure, but the key is how to "build a basket you can guard carefully". Consider risk vs. cost here. It may be less risky (for link or hardware failure) to implement multiple redundant links, however if it is significantly more complicated, it is more prone to error than using simple components (which are less prone to failure) and keeping a spare around (making sure to keep its configuration up to date or be able to restore it quickly from backups).

Weakest Link

Analyze the design to identify and rate the vulnerabilities. Identify the weakest link(s) and take measures to make the strength proportional to the risk.

- Hosts generally have no "choke point" – They are wholly accessible through their network interface. We place GIACs behind external firewalls and will run individual firewalls on each server.[14] This way, we have some finer grained control over who even gets past the NIC. In addition, the operating system is often able to restrict points of attack (e.g. xinetd and tcp-wrapper[15] for unix can filter sets of applications.) Some applications can too (e.g. Oracle can be configured to restrict which IP addresses get to talk to it's "listener" process via the "protocols.ora" file) Look for this functionality when choosing applications.

- Hosts have many "weak links" – consider this when mapping services (applications) to servers. In some cases it may be better to use separate servers. In some cases not. Review the "track record" of any applications you are considering. If it's been subject to vulnerabilities in the past, it probably will continue to be in the future. One classic example is the Unix "sendmail". While it is possible to run a secure sendmail site, it is certainly a chore to keep up with patches. We would recommend using "postfix"[16] as a sendmail replacement on GIAC's external email servers. Note: Even if the desktop users request Exchange internally, postfix can be used on the external (service net) mail

[14] We will use Mason to do this and detail it later in the paper. See http://www.stearns.org/mason/ for general information on this tool.

[15] Tcp_wrapper was written by Wietse Venema in the early 1990s to address this. It's author maintains that tradition with many other tools and papers. Please visit http://www.porcupine.org/wietse/

transport server.

- Use filters/firewalls, host and network IDS and host hardening to manage risk. I recommend that GIAC use Tripwire on every external (and internal) server and Snort on every network that contains servers (running Snort on the employee network would be rather pointless, unless you believe there is reason to suspect evil traffic exists there.) Further details on these are coming up.

Fail Safe Stance

Denied unless specifically permitted. This is a case of "what you don't know can certainly hurt you" and "if you don't run it, it doesn't matter how buggy it is"[17]. This applies to all areas of the GIAC network both internal and external.

The VISA Cardholder Information Security Program

VISA has instituted a certification program for vendors doing business on the Internet. As part of this, they have published a list of "Logical Program Requirements" that is also worth a read whether or not your company has an interest in becoming "CISP Compliant".
Their program has a website at:
http://usa.visa.com/business/merchants/cisp_index.html and the link to the current document (Version 5.5 as of 2000) that specifies these requirements is: http://usa.visa.com/media/business/cisp55.pdf Worth investigating for GIAC (or any company doing business on the Internet).

The List is reproduced here for you until you can read the entire document :-)

1. Install and maintain a working network firewall to protect data accessible via the Internet.
2. Keep security patches up-to-date.
3. Encrypt stored data.
4. Encrypt data sent across open networks.
5. Use and regularly update anti-virus software.
6. Restrict access to data by business "need to know."
7. Assign a unique ID to each person with computer access to data.
8. Don't use vendor-supplied defaults for system passwords and other security parameters.

[16] Here is the Postfix home page: http://www.postfix.org/

[17] Taken from "Firewalls and Internet Security, Repelling the Wiley Hacker", William R. Cheswick and Steven M. Bellovin 1994.

9. Track access to data by unique ID.
10. Regularly test security systems and processes.

Two additional requirements pertain to administrative and physical security issues:

11. Maintain a policy that addresses information security for employees and contractors.
12. Restrict physical access to cardholder information.

Specifically address all "known" attacks.
As part of our security architecture, we will also provide GIAC with recommendations on how to maintain their systems. We will begin by listing several services for information on guarding against attacks. The GIAC IT team should pay direct attention to any attacks that apply to services they are running even though the servers are behind a firewall. Most of this is "host based" security, but I believe it is appropriate to at least list the references here.

- The SANS "Top-20" list of vulnerabilities http://www.sans.org/top20.htm

- SecurityFocus maintains a set of mailing lists that include vulnerabilities, incidents, tools, and many other topics at: http://www.securityfocus.com/cgi-bin/forums.pl

- RedHat also maintains services including the "Red Hat Network" where you may register systems (a fee service) and receive email notification of newly released versions of software to address known vulnerabilities. There is even a tool "up2date" that is very useful to keep your system at current levels. They are at http://www.redhat.com and the Red Hat Network is https://rhn.redhat.com/

- Not detailed in this paper, but necessary for ongoing maintenance would be a part of the administration/operation section of the GIAC security policy manual on how to use such information.

    - Who and how often will these check these resources?

    - What types of attacks will be looked for based on the specific services and brands of equipment used to implement them?

    - What procedures will be used to react to this information?

- What types of review will this information be subject to prior to implementing a fix/work-around?

- When is something critical enough to be responded to without review?

- As business grows, consider using external services like "TruSecure" that have the resources to monitor many sources for new attacks and vulnerabilities "in the wild" and know their clients architectures so they can provide an "early warning system" outside GIAC. In addition to some valuable free resources available at http://www.trusecure.com/, they maintain a staff dedicated to following the latest threats and cross reference them with their customers' potential vulnerabilities. Their service includes audits as well as automated alerts and work-arounds to threats that are usually available to their customers long before the threat becomes publicly known. Quite expensive, but depending on the in-house resources available and the cost of the potential risks, it may be worth consideration as GIAC grows.

Implement monitoring as part of the overall architecture
The best architecture includes ongoing processes (both automatic and manual) to check for signs of trouble.

Host based IDS: Tripwire
As an extension to the "traditional" monitoring that takes place via syslog, I strongly recommend the use of a host based intrusion detection system as the "inner layer" of security. Tripwire is the one I have the most experience with and recommend for GIAC. The premise here is that these tools record an image (or signature) of the system in a known good condition. It is then run at regular intervals to indicate any sign of tampering. Although it is designed to detect changes "after the fact", it is nearly foolproof if the original signatures (and the application) are kept on read-only media. The trick here is to configure it so as to not send too many "false positives" when files or directories change due to normal operation. There are excellent resources on its use, which I will place in the references section. Although there is a commercial version, which offers much greater ease of manageability, and claims to protect switches and routers[18], in keeping with the requirement that this architecture be very low cost, the Linux version[19] is specified in the initial design.

As part of the overall architecture, a strategy for the use of Tripwire will be

[18] Commercial versions are available through http://www.tripwire.com/

[19] Available as part of the RedHat installation or at http://sourceforge.net/projects/tripwire

developed. This will include a series of tests with different levels of detail included in the configuration to determine the optimum frequency and configuration that may be run without hindering performance. The more often it can be run, the better. The goal is to catch an attack in progress, before much damage can be done. Minimally, running tripwire to monitor the critical system files that would be initial targets for attack such as /etc/password, /etc/shadow, /etc/masonrc, /var/lib/mason/baserules and any other files that control operation of the applications on each server.

Key servers within GIAC that will require Tripwire are:

- All externally facing servers on the Public, Customer and Partner/Supplier links.
- All internal servers on the Services network. Although these are less critical and have less sensitive information on them, they require regular monitoring.
- All internal servers on the Database network. These are the corporate "family jewels" and must be protected to the greatest degree possible.

Network based IDS: Snort
Network Intrusion Detection Systems (whether freely available like "snort" or commercial applications like Dragon) are designed to monitor the network for signs of intrusion. In essence they are configured to watch for either a set of traffic they "accept" and ignore, and log/alert all else, or they are configured for specific patterns (Trojans, DoS attacks, etc.) and take some action when these are seen.

We specify the free utility "snort" for the GIAC architecture. One very useful feature of snort is it's ability to actively defend against attack[20]. It can be configured to send any type of packet desired on seeing a particular traffic pattern. Specifically, it will be configured to send a RST to either or both end(s) of what it deems an improper connection (the innermost IDS may not be able to send to the attacking system if it is outside the firewalls, the ones on the service nets will.). This plays a big role in the GIAC architecture, and systems will be placed in key locations in the network. Each of the following locations will have Snort based IDSs configured to protect the servers at their location. Each of them should be configured with a local modem to allow them to dial out to an on-call pager without depending on network connectivity.

[20] "Active Response" or "Flexible Response" is the feature that allows Snort to take action when some kind of activity is detected.

- The “Public Services” network. This will defend the web, mail and dns servers, as well as the internal networks from attacks originating from the web, mail or dns servers.

- The E-commerce service network. This system will monitor and guard the customer facing sales system. It will need to be configured to handle a fair traffic load, but less than the other systems.

- Behind the (three) services link firewalls on the “internal DMZ”. This system will be faced with the important task of monitoring for any unauthorized traffic and tearing down the connections. Since it resides between four firewalls at the one single point where all traffic between the internal network and both the service nets and the Internet, it must be a fast machine (perhaps multiple machines). Active Response will be configured to tear down connections for any inward or outward bound traffic between systems where there should be none (e.g. if anything on the “customer services network” starts trying to connect to anything on the “internal services network”, or if anything except a specific list of servers try to access anything on the “internal database network”) It’s rules will be tailored by it’s performance, and many of them will be duplicated in more detail on the internal IDSs.

- On the GIAC internal database network. This is the final (active) line of defense on the network itself. It will be configured so that any traffic outside of approved servers connecting to the specific services on the database machines is torn-down immediately and an urgent message sent to the on-call staff (via pager through a dedicated phone line).

- On the VPN link “service network” where any applications for partners or suppliers will be hosted.

- On the internal GIAC server network. This is where the internal web sites for employee information, internal product and marketing pages, and other department and corporate materials will be kept.

System log monitoring.
The proper configuration of system log facilities can be an excellent layer in the Defense in Depth strategy. This can be done either with simple scripts and native system logs or with a complex enterprise wide management application. As with other tools however, improper configuration can leave you with either a false sense of security when in fact you are either not logging enough, or are “drowning” in a sea of trivial alerts. As with many other critical aspects of an

overall security architecture, this topic is worthy of much more detail and time than this forum permits. I will however, list what I consider to be a few important points to guide the readers' thought process.

The GIAC architecture relies more on snort and less on syslog for "real-time" alerting, which allows them to not maintain a central syslog server (reducing the holes required in the firewalls). Instead, the backup servers will perform this log collection (since we need to do backups anyway).

- Consider what your requirements for using the logs are.
    - Do you expect that they will integrate with an automatic event escalation system and process? Will there be staff available to receive these alerts and be able to take action? Do you have reliable means for this escalation to take place? If you believe you have automatic escalation, but an attacker (or simply a system or network outage) can disable this alerting mechanism, you have a false sense of security.
    - Do you need it to be a detailed transaction "audit trail" of all events that take place on every server?
    - Do you need the logs to be useful in any potential legal action?
    - Do you need it to monitor for other events outside the traditional security realm? Things such as performance, resource availability, application monitoring?

- Consider what resources you have available to setup and maintain the logging framework.
    - Do you expect to have one central log repository that collects from every server? This has it's own set of hidden costs, since providing a network path to this host presents additional risk.
    - Do you have sufficient bandwidth, CPU cycles, disk space to maintain a central facility?
    - Do you have sufficient human cycles to monitor and maintain the logs? Aside from the logs that actually generate alerts, will someone be

assigned to screen them for things that may be happening "under the radar"? How often do you expect this to take place? Daily? Weekly?

Design in "flexibility"
Hopefully this is handled primarily in the architecture. Any good architecture will already consider the technology of the day partially obsolete, and have built in several areas where change can occur without a major redesign. This must be supported by incorporating the means to stay current into the policies. By this I mean do not assume that a set of policies that implement good (or even excellent) security today will be good (or even acceptable) tomorrow. Security vulnerabilities are discovered continually and the policy must have built in the mechanisms to stay reasonably current. This should be addressed directly as part of someone's job description!

GIAC Network Design (basic principles):
In designing the network for GIAC, I used the "divide and conquer" approach. This was simplified by the decision to use all private internal addresses and perform IP masquerading for all internal traffic that is destined for the Internet. The design allows subnet masks to break the internal network into easily controlled regions for different purposes. Where possible, those networks are aggregated into larger groups that can be represented with a broader subnet mask. This allows routing rules (and firewall policy) to take advantage of this and use the broader scope mask for wide policy decisions, and the narrower mask for more specific rules. Coupled with the extensive use of DHCP for the internal employee sections of the network, it allows workers and administrators alike to simply move machines from one area to another, and get the appropriate policy applied automatically[21]. This also simplifies laptop configuration, since more and more people have full-time Internet access at home with addresses provided via DHCP. NOTE: Please do not misinterpret that we are basing the internal security policy on IP addressing. All corporate resources will be required to provide authentication according to the nature of the resource. The addressing scheme merely simplifies management of the network space.

Internal GIAC addressing: Subnet breakdown:

In designing the internal address space, it was decided to use the class A "10.0.0.0" as all internal GIAC hosts. This had several advantages:

[21] DHCP permanently assigned addresses based on MAC may be used where it is not practical for machines to keep changing addresses (servers within departments or laptops where access to restricted resources may use source address as part of the filtering scheme).

- It gives the external firewalls & routers simple control over this traffic.
- It blocks the entire internal space from external probing fairly effectively.
- It simplifies the routing and firewall rules by allowing aggregation of sets of subnets for macro rules, while allowing for the possibility of refining those rules later.

As an example of the simplification made possible, consider the internal firewall. It's job is to separate the internal address space into four main sections:

1. The IDS net (10.0.0.0/10) which serves as a kind of internal DMZ behind the external connections. This mitigates the added risk created by multiple connections to the Internet by providing a central point for monitoring and control. This network will host only the interfaces from the firewalls and an Internal IDS (snort).
2. The Internal "Services" network (10.64.0.0/10) which provides commonly available utility services. Creating a separate network for this allows a finer grained access control to these services and mitigates the risk from break-in.
    - Web – all internally "officially hosted" web servers.
    - DNS & NTP – the internal half of the "split dns" and an internal time server
    - Email – initially a POP & SMTP server, perhaps an Exchange server with SMTP gateway as demand grows.

3. The Internal "Database" network (10.128.0.0/10) where the most sensitive resources are kept. Placing them on a separate network allows for very tight access control and makes placing an IDS on this network very effective.
4. The "Employee" network (10.192.0/10) which is the less restrictive network. Using a single large address space within the "class A" gives two key advantages:
    - Allow the firewalls to treat them as a single group by using a "macro" subnet.
    - Allow the firewalls to separate between them by using smaller subnets based on departmental need. This is not a strong protection (since we assume the employees will be able to change the IP addresses of their machines) but does allow for yet another layer of control that can be implemented very simply.

    Note that the Employee network is further able to be broken down by department using the 32 subnets in the "upper half" of the class A from 10.192.0.0/14 through 10.252.0.0/14 (Note: each of these will contain 262142 possible addresses. Should allow for a fair amount of growth at GIAC…) In consideration that each of these networks may need to be further split (for example by "local" or "remote" employees in each

department or perhaps allowing close business partners to masquerade within GIAC using these ranges) we "skip" subnets so that these other ranges will be available in the future.

a. Authors: 10.192.0.0/14
   The 10.196.0.0/14 is unused, allowing for this to be allocated for remote authors and still treat all authors the same by using the 10.192.0.0/13 subnet mask which encompasses both ranges from 10.192.0.1-10.195.255.254 **and** 10.196.0.1-10.199.255.254)
b. HR: 10.200.0.0/14
   The 10.204.0.0/14 is unused, allowing for this to be allocated for remote HR workers and still treating all HR the same by using the 10.200.0.0/13 subnet mask which encompasses both ranges from 10.200.0.1-10.203.255.254 **and** 10.204.0.1-10.207.255.254)
c. Finance: 10.208.0.0/14
   The 10.212.0.0/14 is unused, allowing for this to be allocated for remote HR workers and still treating all HR the same by using the 10.208.0.0/13 subnet mask which encompasses both ranges from 10.208.0.1-10.211.255.254 **and** 10.212.0.1-10.215.255.254)
d. Marketing: 10.216.0.0/14
   The 10.220.0.0/14 is unused, allowing for this to be allocated for remote Marketing workers and still treat all Marketing the same by using the 10.216.0.0/13 subnet mask which encompasses both ranges from 10.216.0.1-10.219.255.254 **and** 10.220.0.1-10.223.255.254)
e. Sales: 10.224.0.0/14
   The 10.228.0.0/14 is unused, allowing for this to be allocated for remote Sales workers and still treating all Sales the same by using the 10.224.0.0/13 subnet mask which encompasses both ranges from 10.224.0.1-10.227.255.254 **and** 10.228.0.1-10.231.255.254)

This scheme still allows for other departments to be created in the future using the space from 10.232.0.1 through 10.255.255.254.

GIAC Network Security Policies (least specific to most)

Global Policies for External Routers:

Basically, the external routers should be used as a "coarse filter" for the most obvious "incorrect" or "explicitly denied" traffic. Their job is not to implement a

mirror of the firewall rules, but to make the firewall's job easier and provide an external layer of protection to the network.

At GIAC, the default policy will be "denied unless explicitly required", so the allowed traffic rules will differ from link to link (the services and address ranges allowed through will differ). However, there are some global rules that are worth mentioning here:

1. harden the router itself. Commands vary by brand[22], but in general:

    a. change the banner to "WARNING: Unauthorized Access Prohibited" or something like this that the GIAC legal folks approve of. (In fact, every host or service should be configured to give away as little information as possible in it's initial connection dialogue.)

    b. restrict access to the router to specific interfaces, hosts and protocols required Note: allow time synchronization for the routers, as for all the gear in this architecture. Makes the logs make more sense.

    c. restrict responses from the router itself. No need to respond to potential probes by sending icmp messages that give away information or responding to traceroute scans (TTL field set to 1)

    d. block SNMP unless your ISP requires it (and then don't allow write access and choose something besides "public" for the read-only community string)

    e. Be aware of how your router allows administrative access. Some passwords are stored in cleartext. Some routers allow admin access from all interfaces by default. Read your documentation.

2. Block any packets that should never appear. Pay attention to both source and destination addresses when looking for these "leaks"[23]

    a. IP ranges should not be permitted either in or out of GIAC external routers with "private" RFC1918 source or destination addresses. These include 0.0.0.0/8, 127.0.0.0/8, 192.0.2/24, 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 and 169.254.0.0/16. In fact only GIAC public IP addresses should be allowed out at all.

[22] Hardening Cisco IOS whitepaper at http://www.cisco.com/warp/public/707/21.html

[23] This is identified at http://www.sans.org/top20.htm - G5.1

b. Internal GIAC addresses should never appear on the outside interface as the source address. This may be a sign of a spoofed attack trying to circumvent filters.

3. Drop packets with source routing specified, or any IP options field values set. There are several source routing types and reasons we should block them:[24] They can be recognized as packets with an IP header greater than 20 bytes and having the following IP options field values
    a. Strict source routing – 0x89 – specifies (up to 9) intermediate routers that the packet must pass through, beginning at the source.
    b. Loose source routing – 0x83 - specifies some hosts the packet must hit along the route.
    c. Record route option – 0x07 – collect the addresses of the routers the packet passed through.
    d. IP timestamp option – 0x44 – collect and return timestamps from the routers seen.

    Both (1) and (2) allow an attacker to specify a router (presumably one under their control) that must be involved in the route. (3 and 4) allows them to gain information about the route "normally" taken. All of these give away potentially useful information. In practice, the IP options field is (nearly) never used, so we can assume that anything seen here is not something we want in our network.

4. If the packet has passed all the above, allow traffic to or from services that are appropriate for this network as follows:
    a. The packet has a legal destination address (not just GIAC, but one of the specific servers or ranges that this particular router should serve).
    b. The packet is to a legal destination port (this will be a list of specific server address/port allow rules) or at the very least, a list of valid ports if there is a valid reason that specific IP addresses would not be practical here (e,g, on the VPN link, the rules apply to more IP addresses than on the E-Commerce link). The intent is not to re-invent a firewall with a router, but to make a sound, solid ruleset that blocks the majority of the junk.

5. Drop everything else. Note: we don't specify logging all dropped packets here for a couple of reasons:
    a. This may be an ISP's router (especially when GIAC is starting up) and the cost to have them perform custom logging and monitoring may be prohibitive.
    b. More importantly, this is designed to be a filter, not an IDS. We will get more

[24] "TCP/IP for Firewalls and Intrusion Detection" SANS Institute"

specific at the firewall and beyond.

Global Policies for Firewalls

In conjunction with the filtering routers, firewalls will be the next line of defense in GIAC's architecture. The perimeter firewalls need not re-implement all the routing rules (although if performance permits, this would be an additional safety net), but will still adopt a "DENY unless specifically ACCEPTED" stance. Their main difference is that they will assume that most obviously "incorrect" traffic has been blocked, and so if they see any, they will log it and send an alert.

1. Harden the firewall itself.

    a. Block any initiating service requests to any of the firewall addresses except for known administrative systems (specify a list and require ssh). Should not allow access from the Internet interfaces. (for the GIAC Public link, since we do masquerading for outbound traffic, we cannot simply say block all traffic to the firewall.)

    b. Block any outbound traffic from the firewall that we do not expect. This may vary somewhat by link.

    c. Allow administrative access over SSH, time sync to the appropriate servers, and DNS to our servers from the firewall (useful for log synchronization and analysis).

2. Allow appropriate traffic to access services provided by that firewall. This is a repeat of rules implemented on the accompanying router. This gives us extra protection in case one ruleset is either implemented incorrectly (Murphy wins one once in a while) or the router becomes compromised. As on the router, the rules should be very specific. If there are three separate servers doing DNS, SMTP and HTTP, list them individually as "SMTP to mail.giacfortunes.com allow" etc. Do not simply allow DNS, SMTP and HTTP onto this network.

3. Deny any traffic that seems to be an attack, log it and send an alert. Examples are:

    a. Inbound telnet to any server. GIAC administers their machines from inside using SSH.

    b. Inbound SMTP to the Web server. (any request for service to the "wrong"

machine)

c. Outbound FTP from the DNS server (it may be partially compromised, and the attacker is trying to retrieve his root-kit).

d. Mal-formed packets (incorrect combinations of flags, etc. should be dropped) They may be an attempt at crashing or cracking a service through a known or unknown vulnerability. Since different OSs treat these (six) flags differently, it is hard to determine which ones to specifically watch for (monitor those vulnerability lists). By default, we know we need SYN, ACK, SYN/ACK, FIN, FIN/ACK, RST are all legal combinations in normal TCP traffic. Anything outside of this is suspect and could be trouble. However, the only ones we will block (initially) are:

   i. packets with both SYN and FIN or RST (illegal combos)

   ii. packets with no flags at all set. (NULL scans to see what the system will do)

4. Deny any other traffic not specifically allowed and log it. This should be a small amount of traffic if the router is doing its job. We don't throw an alert here, because any "known" attack signatures should go into the previous ruleset. This section is intended to help tune the router, and also allow analysis of abnormal traffic changes that might be signs of trouble. For example, if the number of attempts to talk to DNS on the firewall interface (where we don't run DNS) suddenly increases over a short period, it may be a sign that there is a new DNS vulnerability. This is a catchall ruleset.

Global VPN Policies

On the VPN link, we need to be both extremely careful and also more liberal with what traffic is allowed. Since the traffic is assumed to be from a trusted source, we are required to let it into the internal GIAC LAN. However we take two major precautions here:

1. We authenticate strongly here using something of no lasting value (like S/Key one-time passwords). If GIAC budget permits, we recommend using a two-factor authentication (something you know and something you have) like a SecurID card. The specific technology is not as important as the fact that the authenticators are not reusable, so to prevent replay attacks.

2. We control the IP addresses that come inside and use this in conjunction with

the firewall authentication policies to provide restrictions on what there is access to internally. (Recall, we have an internal firewall that protects critical resources).

As for all external access, since access can come from anywhere, the only addresses we block are the illegal ones (mentioned in the previous section on global configuration parameters). In addition, (also mentioned previously) the VPN router will be configured to drop and alert any known attacks (depending on available local paging services). The VPN firewall itself will provide two main services:

1. Allow individual entities to have controlled access to internal services (remote access for suppliers/employees). VPNs will be configured with very tight authentication and logging for supplier access. Services will be a limited set of what is available within the GIAC perimeter. Note: The use of a Linux firewall would make validating the authentication much more difficult, since that will take place on the VPN router. We must assume the router has done it's job and use the firewall for IP based filtering to allow employees access to the internal applications. Note however, that being inside the GIAC network gives a user very limited access. All internal applications are themselves access controlled; so further authentication will be performed at their end.

2. Site to site VPNs for access between business partners' networks. These will be configured with very strict rules allowing each partner to only access what is provided for them. Authentication will need to be customized for each link, but since it is a "mostly on" link, it will need to be box-to-box authentication. If there are "user services" that are provided for the partners' employees, they should be able to be provided via the partner applications web site rather than letting them directly access internal systems. (Recall that since internal systems do their own authentication, so granting external users accounts still allows fine grained access control).

   Site to site VPNs will also be configured to be paranoid about what is let out. A similar set of access controls for GIAC employees and applications passing out through the VPN will need to be developed and maintained according to specific partnerships.

Specific Example: Public Link Policies

Border Router

The Public Link border router will be configured to provide

- All the policies listed in the "Global Policies" section.

- Allow access to all “public” services:
    - http to www.giacfortunes.com and return traffic
    - smtp to and from mail.giacfortunes.com
    - DNS to dns.giacfortunes.com (lookups to and from this server only). We don’t allow zone transfers, so only UDP will be enabled. Note: since the number of external GIAC hosts is small, we can probably deny TCP DNS traffic. However, watch this for signs of trouble (it may need to be enabled at some point in the future.) Make sure the DNS server is configured to control zone transfers in any case.
    - NTP connections to ntp.giacfortunes.com
- Allow outbound access to HTTP and HTTPS from internal systems (they will be masqueraded as the firewall external address). Allow reply traffic.

Primary Firewall

The firewall will provide

- All global access rules (spoofing, bad packets, etc.)
- External access to www.giacfortunes.com over HTTP.
- External email access to mail.giacfortunes.com over smtp.
- Provide the external half of split-DNS with dns.giacfortunes.com. Allow forwarding for all external lookups from this server and allow access to this server from the firewall itself plus the internal DNS servers within the GIAC private network.
- Masqueraded Internet access for all local GIAC employees. Initially only HTTP and HTTPS will be allowed. Note: future installation of a proxy application (like Squid[25]) should be considered.

[25] The Squid Proxy main page http://www.squid-cache.org/

- Allow the other servers (DNS, SMTP and NTP) to synchronize with their mirror images on the internal GIAC network.

- Allow the firewall itself to receive SSH from a restricted list of internal systems (their IT staff).

- Allow the firewall to pass SSH to the router and machines on the service net for managing these devices. (Note: This will meet Marketing's requirements to manage the web server. They will be given SSH file transfer ability.)

Customer Link Policy:
E-commerce Border Router
This router will be configured to provide access only to the "E-commerce" servers. We need to run an HTTPS web server here. In addition to this, we need to send confirmation email to customers. But the number of servers is limited, so the routing rules should be fairly straightforward.

E-commerce Firewall
This is also pretty straightforward. Limit to specifically what services are provided by which servers. Even if the rules are nearly duplicates to the firewall, since the routers and firewalls are different architectures, there is less chance of a cracker finding a vulnerability that will get through both devices.

- Allow HTTPS to and from this server.

- Allow SMTP outbound from this server. We will receive email on the main MTA on the public link. That way we don't need to allow SMTP inbound.

- Allow NTP to internal servers.

- Allow SSH from internal IT management systems.

- Allow connection with the main database application (possibly SQL*Net, possibly another protocol to the application server).

VPN Link Policy:
The VPN Link will implement the following policy:

- Access to the VPN router from any legal Internet address. (standard block illegal traffic rules from the general configuration section)

- Allow authentication traffic from the router to the internal server (assuming GIAC can afford a SecurID setup, this will be the ACE server)

- Use Encapsulating Security Payload (ESP) for all VPN traffic.

- Use either AH and ESP or ESP alone for partner links (depending on their addressing scheme, if they perform NAT from their end we cannot use AH)

- Individual remote access sessions shall only use ESP (and not AH) to allow them to function through NAT, which is becoming quite common for home networks.

- The Security Associations for these connections will depend upon the use of ESP or ESP & AH.

    - Hosts (Suppliers and remote sales staff) will support both transport and tunnel mode

    - Partners will support only tunnel mode (unless they only require individual hosts have access).

- Split horizon will not be implemented for supplier access. This carries with it some risk to a compromised system giving a back-door into GIAC, however all internal services do their own authentication and are behind another set of firewalls, monitored by IDSs. The risk is considered low. In addition, if the remote system has been compromised, split horizon does not really stop an attacker. (A trojan that only activated once the VPN came up, or a virus from email or file sharing would not be blocked)

- Since I do not include the VPN component in the tutorial section, here are some important items specific to the Cisco 3000 VPN concentrator:

    - The security advisory field notes page is http://www.cisco.com/warp/public/770/52.html

    - The ISP's router will need to pass a number of ports to allow the Cisco VPN concentrator to work http://www.cisco.com/warp/public/471/vpn_3000_faq.shtml)

PPTP Control Connection	6 (TCP)	1023	1723
PPTP Tunnel Encapsulation	47 (GRE)	N/A	N/A
ISAKMP/IPSEC Key Management	17 (UDP)	500	500
IPSEC Tunnel Encapsulation	50 (ESP)	N/A	N/A
IPSEC NAT Transparency	17 (UDP)	10000 (default)	10000 (default)

part 2: Security Policy Tutorial

Public Services Link: Primary Firewall Implementation

This section will describe how to implement the firewall policy on the public link. In preparing this part of the practical assignment, I created a testbed network using three Windows machines (two laptops and an older desktop) to simulate the Internet, Service Network and Internal network surrounding the firewall. For the firewall itself, I used the following hardware/software configuration:

Hardware Software configuration
Firewall HW:

- DELL PowerEdge 1400SC
- (1) 933MHz
- 256MB
- 18GB 10k SCSI
- (1) 10/100 Intel EtherPro (onboard) NIC, (2) 3Com 3C980 10/100 NICs.

Firewall SW:
Note: detailed references are listed as a separate section of the paper.

- RedHat Linux 7.1
- Ipchains 1.3.10
- Mason 0.13.0.92

This system would be reasonable for even a production implementation at a startup. The only changes that I would recommend would be the addition of a second CPU, upgrading the RAM to 1GB and adding additional SCSI disks for

log collection, bringing the total to around $3000.

Installing the firewall software:

I will not address the details of the installation of RedHat Linux, other than to refer the reader to several excellent resources [26]and mention some key points. As is always proper practice, the installation should be a "ground-up" install including partitioning the hard disk and take place with the system physically disconnected from the network. The installation should begin by defining which services are required for installation and include a systematic "hardening" which at very least consists of the following basic steps:

1. Evaluate your physical security and potentially make BIOS changes.

2. Perform the installation, including only the minimum components necessary to run the system.

3. Check and disable all unnecessary network services after installation.

4. Conduct an initial run of "tripwire", copying the database off-line to write-protected media.

5. Connect the NICs to a "safe" network for firewall configuration.

6. Install any updates to resolve known security problems with the version of software installed.

Note: The initial "tripwire" baseline must be conducted prior to the system being connected to any potentially untrustworthy network, or having any software loaded which is not from a trusted source. Subsequent snapshots can be taken at any intermediate steps desired. A final baseline will be created following the firewall configuration, but having this initial one copied to read-only media will insure that we can always refer back to see what has been changed.

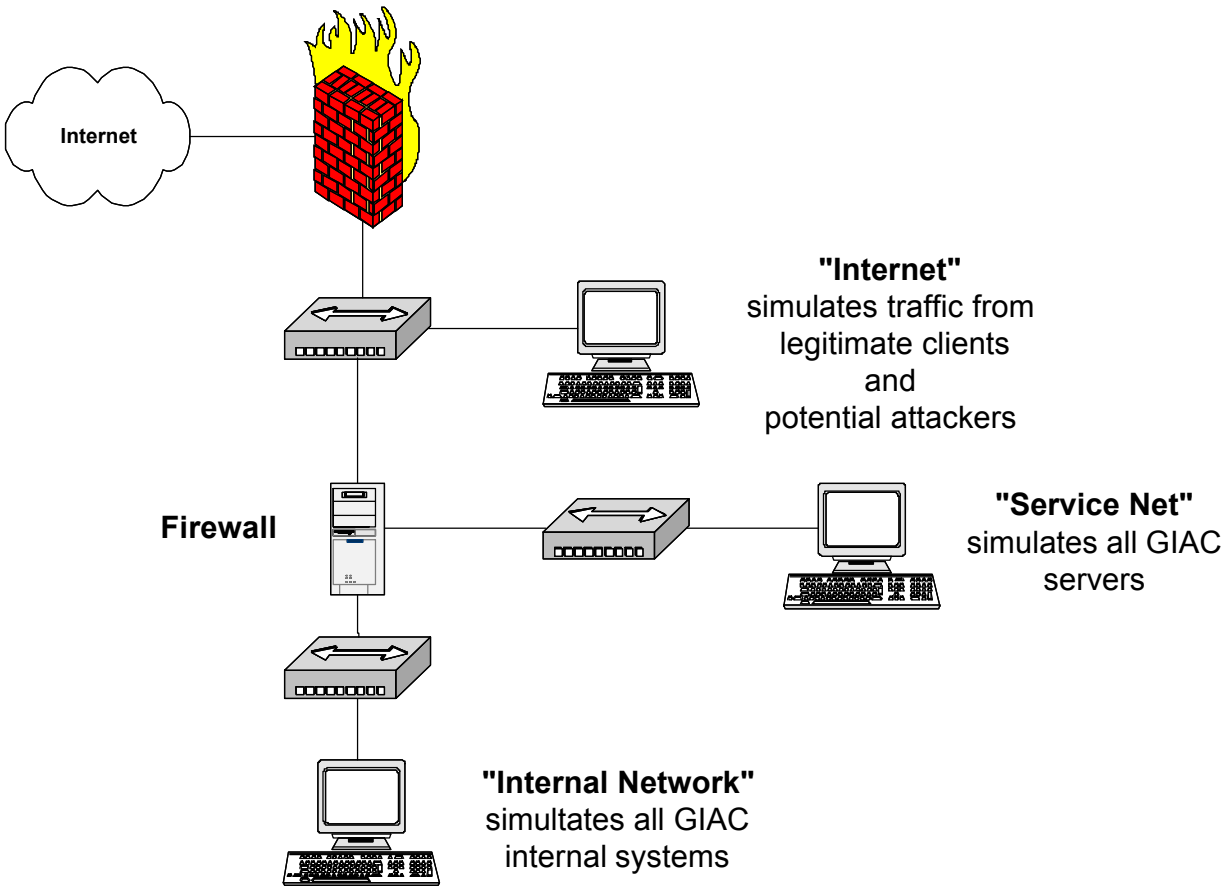
Setup a Development and Testing environment:
Once the system is at this level, (basic hardening, patches loaded, tripwire baseline established) it is time to actually build and test the firewall ruleset. This

[26] "Securing Linux Step-by-Step" Version 1.0, 1999-2000. The SANS Institute,

"Trinity OS": http://www.ecst.csuchico.edu/~dranch/LINUX/index-linux.html#trinityos

and "Bastille Linux" at http://www.bastille-linux.org/

requires the system to be connected to a test network, which is either not connected to the Internet, or is at least itself behind a firewall. The goal here is to allow verification of the firewall configuration without exposing the system to crackers. The testbed network used for this practical is here:



Network Testbed

The firewall configuration itself consists of several steps. They will be detailed specifically, but the general process is as follows:

1. configure the network interfaces of all hosts with appropriate parameters (IP addresses, etc.)

2. verify connectivity with no firewall running

3. Implement a set of rules – That is, configure the firewall rules for a particular service.

4. Perform testing on that unit – Specifically test the services intended to be either provided or blocked. Note any defects.

5. Analyze and correct the defects, repeating steps 4 & 5 until you are satisfied with that block of rules.

6. Add the next logical block of rules, iteratively performing steps 3, 4 and 5 until the complete configuration is in place.

7. Perform comprehensive testing of all services and the intended blocked ports.

8. Analyze and correct any system defects, repeating steps 7 & 8 until you are finished.

9. Take a final tripwire baseline (once again, creating a copy on read-only media) before moving this equipment into production.

Perform the Firewall Configuration

The firewall will be a Linux system with three NICs running the ipchains firewall. The decision to use ipchains vs. the newer iptables was due to the experience level of the on-site staff at GIAC. They are not Linux experts, nor are they firewall administrators, and they (and I) felt that the amount of support available for ipchains in the form of good documentation, newsgroups, and the availability of Linux administrators with that experience made it the best choice for the initial rollout[27]. There is a proposal to revisit this design with the staff gaining the appropriate level of experience with iptables and re-implementing the firewalls

[27] Author's note: This reflects my own situation. I was only able to obtain my firewall system roughly four weeks ago, and up until that time, had no experience with building Linux firewalls or routers, and felt there was more documentation available for ipchains than iptables. I intend to re-build this machine and begin work

using this as a followon to this project. Undoubtedly, within the next year or two most Linux firewalls will be based on iptables. The fact that stateful inspection is not available with ipchains places a heavier burden on the IDSs, and they will need to be maintained with the latest appropriate signatures, however the level of security provided by the overall design will be sufficient.

(Author's note: I look forward to someone testing this ipchains design in their "Design Under Fire" section in the coming months!)

In order to make it easier for the GIAC staff to build and maintain the GIAC firewalls, I am using an automated tool to help generate the specific rulesets. Mason[28] is a utility that essentially automates the majority of the creation process. It allows you to build the firewall rules dynamically, rather than crafting rules according to researched protocol specifications and then testing them, simply telling the "mason-gui-text" utility to "Begin Learning" and then running the service you wish to allow (or deny). Mason will create the appropriate rules based on traffic it detects.

A word to the reader. Mason is only a tool. It cannot substitute for good design or thorough testing. Please make sure you understand what the rules are specifying and tighten or loosen appropriately.

With that, here are the steps used to build this firewall:

Configure network interfaces on the firewall:
Note: For the purposes of this practical, consider all 192.168.0.0/16 addresses to be "legal public GIAC IP space" and "10.0.0.0/8" to be the private address space used internally at GIAC. This will be detailed later.

In my testbed network, the "DMZ" is created behind my home firewall which provides DHCP and connection to the Internet. To simplify access to "real" external services for testing, I used this for the Internet facing interface of the firewall:

```
/etc/sysconfig/network-scripts/ifcfg-eth0:
# The GIAC Public DMZ interface will use a static IP.
# on the testbed, it is a DHCP address on the 192.168.0.0/28
DEVICE=eth0
```

---

with iptables after a short but much needed break to catch up on my sleep…

[28] Lists of references for "mason" are included in the References section at the end of this paper. General information can be found at http://mason.stearns.org/

```
BOOTPROTO=dhcp
ONBOOT=yes
BROADCAST=192.168.0.31
NETWORK=192.168.0.0
NETMASK=255.255.255.224
```

The Service Net is configured on eth1 next.

```
/etc/sysconfig/network-scripts/ifcfg-eth1:
# the Public Service Net will be in the 192.168.0.32/28 network
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=none
BROADCAST=192.168.0.47
NETWORK=192.168.0.32
NETMASK=255.255.255.240
IPADDR=192.168.0.33
USERCTL=no
```

Now we configure the Internal GIAC network interface

```
/etc/sysconfig/network-scripts/ifcfg-eth2:
# the GIAC Internal Net will be in the 10.10.0.0/24 network (where the IDS is)
# We configure it using the entire Class A subnet mask and broadcast address
# so that no matter what IP I am testing, it will route OK.
DEVICE=eth2
ONBOOT=yes
BOOTPROTO=none
BROADCAST=10.255.255.255
NETWORK=10.0.0.0
NETMASK=255.0.0.0
IPADDR=10.10.0.1
USERCTL=no
```

Now since this is being tested with laptops (which spend their spare time as actual work machines at my paying job) I have the firewall providing DHCP service so that I don't need to manually reconfigure them both twice a day. Clearly, the real GIAC network firewall would not be a DHCP server… Hope this is OK. I will document what would be different in the **real** GIAC configuration vs. my basement , (er, that is, "The GIAC Project Section of the JRH Enterprises Networking Lab") Here is the DHCP configuration file (GIAC would not need one) This configures the firewall to be:

A DHCP client from my Internet Gateway/firewall (to my broadband provider).

A DHCP server to the Service net machine and the Internal GIAC networks.

And yes, this made me enable bootp in the firewall rules.

```
/etc/dhcpd.conf
# eth0 the Internet facing "DMZ". Don't serve anything.
Subnet 192.168.0.0 netmask 255.255.255.224 {
}
# eth1 the Service net. For the testbed only, serve
# appropriate addresses. These would be static at GIAC
# also domain-name-server is on the service net
subnet 192.168.0.32 netmask 255.255.255.240 {
        range 192.168.0.35;
        option routers 192.168.0.33;
        option subnet-mask 255.255.255.240;
        option broadcast-address 192.168.0.47;
        option domain-name-servers 192.168.0.35;
        default-lease-time 3600;
        max-lease-time 3600;
}
# eth2 the Internal (IDS) net. For the testbed, serve appropriate
# addresses. These would be served by another machine at GIAC
# also domain-name-servers & routers would be different
# the DNS would be an internal caching server that got it's external from
# the one on the service net.
# the IDS net will be 10.10.0.0/24
# with the department desktops subnetted in the 10.[1..n].0.0/24
# range. In this example, we are serving 10.1.0.0/24 (Authors)
subnet 10.0.0.0 netmask 255.0.0.0 {
        range 10.1.0.2 10.1.0.254;
        option routers 10.10.0.1;
        option subnet-mask 255.0.0.0;
        option broadcast-address 10.255.255.255;
        option domain-name-servers 192.168.0.35;
        default-lease-time 3600;
        max-lease-time 3600;
}
```

Creating the Firewall rules:

The following example shows how Mason will be used to implement rulesets. Recall, the architecture specifies "split DNS" using a single machine on the Service Net "dns.giacfortunes.com" which will be the primary nameserver for all

publicly available GIAC addresses. It will respond authoritatively for this domain, forwarding all other queries outside to a designated server that will be another GIAC secondary. Within the GIAC Internal network, another (set of) machine(s) will provide all internal names, and be configured to ask “dns.giacfortunes.com” for external resolution.

Using mason to create these rules, we will simply setup the internal, service and external machines to do these queries and sit back and watch:

1. Configure my “Internet” server (a Windows NT PC at 192.168.0.4) to provide the services that GIAC would obtain externally: Give it a route to the GIAC Services Network:

   ```
   route add 192.168.0.32 mask 255.255.255.240 192.168.0.3
   ```

2. Run a caching only DNS server on this machine. I used the “Simple DNS Plus” package. (see References section). Set it to forward queries to 192.168.0.1 (my “real” Internet DNS server i.e. my firewall)

3. Configure the “Services Net” machine (also an NT box at 192.168.0.35) with “Simple DNS Plus” as well, similarly having it forward queries to the Internet server (192.168.0.4)

Now the fun stuff. Creating the firewall.

I start with the following configuration for “mason”. Set the default policies in the appropriate sections of /etc/masonrc. Specifically, we tell mason to allow anything to pass, building rules that allow each new session and output the rules in “ipchains” format.

Note: We will change the defaults to “DENY” after the rules are stable.

```
#-------------------------------------------------------------
# Essential settings - please set these.
#-------------------------------------------------------------
#A quote enclosed, space separated list of interfaces that change
#IP address from time to time.  Leave as "" if all addresses stay constant.
```

```
#See DYNIFMODE if you want to fine tune how Mason handles these.
#Default: no dynamic interfaces, all have static addresses.
#DYNIF="ppp0"
DYNIF="eth0"



#What policy should mason use for upcoming rules?
#There is no default for this field.  You must choose one of
#the following.
NEWRULEPOLICY="accept"
#NEWRULEPOLICY="reject"
#NEWRULEPOLICY="deny"

#What should the default policy for your firewall be?
#There is no default for this field.  You must choose one of
#the following.
DEFAULTPOLICY="accept"
#DEFAULTPOLICY="reject"
#DEFAULTPOLICY="deny"
#What should the default policy for your system be when the
#firewall is flushed?
#There is no default for this field.  You must choose one of
#the following.
FLUSHEDPOLICY="accept"
#FLUSHEDPOLICY="reject"
#FLUSHEDPOLICY="deny"
```

.

Also of note is the fact that mason can be used to output rules in several formats:

```
# "ipchains" = echo ipchains command to STDOUT, "ipfwadm" = echo
# ipfwadm command to STDOUT, "none" = don't echo either.
# Use "cisco" if you want Mason to spit out Cisco IOS access-list rules.
# Autodetected if not set at all.
# This is what you change if you want a different format in the
# output rule file.
# Default: Whatever this kernel supports.
ECHOCOMMAND="ipchains"
```

Note: the masonrc has many pre-defined sections which allow you to quickly define policy. Some of them may be quite useful, others you may wish to leave the default and edit the “/var/lib/mason/baserules” file by hand. I list some of my favorites:

```
BLOCKEDHOSTS=
```

Lists of hosts which are “blacklisted” presumably for being bad. This is probably one to grow over time.

The file also contains a list of entries for known services, many of which have well-known problems. It makes it extremely simple to add/remove these filters. Here is a (very brief) snippet from that section to show the format:

```
#NOINCOMING="${NOINCOMING} 31337/tcp"                          #BIND Shell
Backdoor
#NOINCOMING="${NOINCOMING} 31337:31338/udp"          #Back Orifice/Deep Back
Orifice Backdoor
#NOINCOMING="${NOINCOMING} 31339/udp"                          #NetSpy Backdoor
```

By uncommenting these lines, you gradually build up the appropriate variable list for processing later. Below this section is one where you may specify the degree to which the firewall itself should try to "disappear":

```
NOINCOMING="${NOINCOMING} 0/tcp 0/udp 7/tcp 7/udp 8/icmp 15/tcp
33434:33524/udp"
NOOUTGOING="${NOOUTGOING} 0/icmp 3.0/icmp 3.1/icmp 3.2/icmp 3.3/icmp
3.5/icmp 3.6/icmp 3.7/icmp 3.8/icmp 3.9/icmp 3.10/icmp 3.11/icmp 3.12/icmp
3.13/icmp 3.14/icmp 3.15/icmp 9/icmp 11.0/icmp 11/icmp 18/icmp"
```

Note that the icmp set still allows 3.4/icmp "fragmentation needed" and "DF" both set. Others are blocked.

Before simply running mason, I will (briefly) explain how ipchains works. There are much better tutorials on this software, but I feel it necessary to provide at least some overview to help the unfamiliar reader follow along.

Ipchains basics
In its simplest form, ipchains can be thought of as having three main types of rules (chains).

1. Input rules – these are applied to every packet that comes from the outside to any network interface.

2. Forward rules – these are applied if the packet is to leave the firewall on a different interface than it came in on. The interface specified is the one it is going towards.

3. Output rules – are applied to every packet that leaves from an interface.

The other important files (running ipchains under mason) to be aware of are:

```
/var/lib/mason/baserules
```

which holds the "merged" rules that you specify as they are being built. It is

empty initially. And the place mason writes any rules it creates on the fly:

```
/var/lib/mason/newrules
```

These apply during the running of "mason-gui-text" but are flushed before it exits, leaving you with only what is in the masonrc plus baserules.

Now we proceed to let mason build a set of rules to allow DNS traffic in the appropriate directions:

- allow DNS from the name server on the service net to query out and receive replies
- allow the internal GIAC DNS server to query the one on our service net and receive replies
- allow the firewall to query the name server on the service net and receive replies

Each of these parts is tested separately, with the rules generated by mason examined after the test is done. Specifically:

1. Start with an empty "newrules" file and run
   `sudo`[29] `/usr/bin/mason-gui-text`
2. Type "bl" for "begin learning".
3. Run "nslookup" from the name server on the service net (dns.giacfortunes.com)
4. Hit "enter" in mason-gui-text and "en" for "edit new rules"
   - Examine the rules created to allow this test.
   - Add a comment "tag" for the exact rules you want to go into "baserules". This is simply that. A "comment" in the file that will allow you to select which rules you wish to merge (there is often a service name in the comment field already that will work just fine)
   - Save these changes (you are editing the newrules file still).
5. Exit the editor and type "mr" for "merge rules".
   - Select "some" rules to be merged
   - Give it your "tag" and say "yes" to merge these rules into "baserules"
6. Type "eb" to "edit baserules"
   - Move these rules where you want them (mason will append them to the bottom)
   - Replace the specific address ranges with the variables defined that

[29] I highly recommend always using "sudo" rather than logging in or "su"ing to root. See References section for more details.

represent the “real” addresses to be used by GIAC. (Part of that “it’s only a tool” thing… In many cases, these will be more restrictive than mason itself will create, while allowing for flexibility later).
    - Make any documentation changes in the baserules file to make it clear what this ruleset does.
    - Save and exit this editor session.
7. Quit mason-gui-text. (Now we will verify the new DNS rules are correct)
8. Remove the newrules file. (We will create another one for the next section of rules)
9. Run mason-gui-text (but don’t tell it to “begin learning” yet)
10. Test you rules, the service should still run correctly. If they don’t go back and re-examine the file. (if you get confused, you can always remove the rules with that “tag” from baserules and start over again!

We continue in this way to do the other two DNS rules. Starting mason with it’s new baserules, running a service, tweaking the rules as they are learned, and then merging them into the baserules file for the next test. Now we have a section in the baserules file that looks something like this (NOTE: To improve formatting, I have removed the beginning of each rule which was: “/sbin/ipchains” and moved the rules so that they are in “input -> forward -> output” order (which is the way the packets will be processed).

Recall: On the firewall:

- eth0 is Internet
- eth1 is Service net
- eth2 is Internal GIAC hosts

```
#!# variables for special addresses to make modifying these rules easier
#!# NOTE: IN THE TESTBED, THERE ARE ONLY THREE OTHER MACHINES FOR INTERNET, SERVICE NET AND
INTERNAL
#!# the addressing scheme would allow for aliases for the different internal subnets here to
allow central
#!# control over whether "finance" gets to surf the net or if "authors" get to use ftp, etc.
without using
#!# real authentication. This "open" policy may well need to change, but for now, we'll all be
good boys and girls.
export eth0ADDR="192.168.0.3"           # firewall Internet side (DMZ)
export eth1ADDR="192.168.0.33"          # firewall service net side
export eth2ADDR="10.0.0.1"              # firewall Intranet side (IDS net)
export extNTP="192.168.0.4"             # Internet NTP server
export extDNS="192.168.0.4"             # Internet DNS server
export extMAIL="192.168.0.4"            # Internet MAIL server
export extANY="0.0.0.0/0"               # This is really “any”.
```

```
export svcDNS="192.168.0.35"            # service net DNS server
export svcMAIL="192.168.0.35"           # service net MAIL (SMTP and POP) server
export svcNTP="192.168.0.35"            # service net NTP server
export svcWEB="192.168.0.35"            # service HTTP server www.giacfortunes.com
#!# tighten these down for production
export intDNS="10.0.0.0/8"              # internal net DNS server
export intMAIL="10.0.0.0/8"             # internal net MAIL (SMTP and POP) server
export intNTP="10.0.0.0/8"              # internal net NTP server
export intEMPL="10.192.0.0/10"          # internal employees all networks
export intSSH="10.0.0.0/8"              # TEMP SSH allow for TESTBED NETWORK
export intTEMP="10.0.0.0/8"             # TEMP for TESTBED NETWORK
export svcTEMP="192.168.0.32/24"        # TEMP for TESTBED NETWORK
export pubRTR="192.168.0.1"             # Temp for the TESTBED
export pubSVC="192.168.0.32/24"         # temp for the testbed
```

Note: The IP spoofing and other specific blocking rules need to go before these ACCEPT sections since we use 0.0.0.0/0 for ${extANY} Otherwise we would let them in.

```
#!# Rules added 11/3/2001 to block RFC IPs on  input (except of course the ones we are using
:-)
#!# also the output rules for the same

 -A input          -i eth0   -s 0.0.0.0/8       -j DENY
 -A input          -i eth0   -s 127.0.0.0/8     -j DENY
 -A input          -i eth0   -s 10.0.0.0/8      -j DENY
 -A input          -i eth0   -s 172.16.0.0/12   -j DENY
 -A input          -I eth0   -s 192.0.2.0/24    -j DENY
 -A input          -I eth0   -s 169.254.0.0/16  -j DENY
 -A input          -i eth0   -s 192.168.0.16/28 -j DENY
 -A input          -i eth0   -s 192.168.0.32/28 -j DENY   # NOTE: THESE WILL ALL
 -A input          -i eth0   -s 192.168.0.48/28 -j DENY   # GET DENIED IN THE REAL
 -A input          -i eth0   -s 192.168.0.64/26 -j DENY   # GIAC NETWORK 192.168.0.0/16

 -A output                   -i eth0   -s 0.0.0.0/8       -j DENY
 -A output                   -i eth0   -s 127.0.0.0/8     -j DENY
 -A output                   -i eth0   -s 10.0.0.0/8      -j DENY
 -A output                   -i eth0   -s 172.16.0.0/12   -j DENY
 -A output                   -I eth0   -s 192.0.2.0/24    -j DENY
 -A output                   -I eth0   -s 169.254.0.0/16  -j DENY
 -A output                   -i eth0   -s 192.168.0.16/28 -j DENY
 -A output                   -i eth0   -s 192.168.0.32/28 -j DENY   # NOTE: THESE WILL ALL
 -A output                   -i eth0   -s 192.168.0.48/28 -j DENY   # GET DENIED IN THE REAL
 -A output                   -i eth0   -s 192.168.0.64/26 -j DENY   # GIAC NETWORK 192.168.0.0/16


#!# DNS section. Allow this first to improve overall performance.
#Rules merged from the new rule file: allow internal hosts to talk to dns.giacfortunes.com and
receive the response
 -A input    -i eth2  -p udp          -s ${intDNS} 1024:65535 -d ${svcDNS} domain -j ACCEPT
      # domain/udp (I)
 -A forward  -i eth1  -p udp          -s ${intDNS} 1024:65535 -d ${svcDNS} domain -j ACCEPT
      # domain/udp (F)
 -A output   -i eth1  -p udp          -s ${intDNS} 1024:65535 -d ${svcDNS} domain -j ACCEPT
      # domain/udp (O)
 -A input    -i eth1  -p udp          -s ${svcDNS} domain -d ${intDNS} 1024:65535 -j ACCEPT
      # domain/udp (I)
 -A forward  -i eth2  -p udp          -s ${svcDNS} domain -d ${intDNS} 1024:65535 -j ACCEPT
      # domain/udp (F)
 -A output   -i eth2  -p udp          -s ${svcDNS} domain -d ${intDNS} 1024:65535 -j ACCEPT
```

```
      # domain/udp (O)

#Rules merged from the new rule file: allow dns.giacfortunes.com access to its external server
and receive the response
 -A input    -i eth1  -p udp           -s ${svcDNS} 1024:65535 -d ${extDNS} domain -j ACCEPT
  # domain/udp (I)
 -A forward  -i eth0  -p udp           -s ${svcDNS} 1024:65535 -d ${extDNS} domain -j ACCEPT
  # domain/udp (F)
 -A output   -i eth0  -p udp           -s ${svcDNS} 1024:65535 -d ${extDNS} domain -j ACCEPT
  # domain/udp (O)
 -A input    -i eth0  -p udp           -s ${extDNS} domain -d ${svcDNS} 1024:65535 -j ACCEPT
  # domain/udp (I)
 -A forward  -i eth1  -p udp           -s ${extDNS} domain -d ${svcDNS} 1024:65535 -j ACCEPT
  # domain/udp (F)
 -A output   -i eth1  -p udp           -s ${extDNS} domain -d ${svcDNS} 1024:65535 -j ACCEPT
  # domain/udp (O)

#Rules merged from the new rule file: allow external machines to query our DNS server and get
responses
 -A input    -i eth0  -p udp           -s ${extANY} 1024:65535 -d ${svcDNS} domain -j ACCEPT
       # domain/udp (I)
 -A forward  -i eth1  -p udp           -s ${extANY} 1024:65535 -d ${svcDNS} domain -j ACCEPT
       # domain/udp (F)
 -A output   -i eth1  -p udp           -s ${extANY} 1024:65535 -d ${svcDNS} domain -j ACCEPT
       # domain/udp (O)
 -A input    -i eth1  -p udp           -s ${svcDNS} domain -d ${extANY} 1024:65535 -j ACCEPT
       # domain/udp (I)
 -A forward  -i eth0  -p udp           -s ${svcDNS} domain -d ${extANY} 1024:65535 -j ACCEPT
       # domain/udp (F)
 -A output   -i eth0  -p udp           -s ${svcDNS} domain -d ${extANY} 1024:65535 -j ACCEPT
       # domain/udp (O)

#Rules merged from the new rule file: allow the firewall to do DNS queries and get replies
 -A output   -i eth1  -p udp            -s ${eth1ADDR} 1024:65535 -d ${svcDNS} domain -j
ACCEPT                                 # domain/udp (O)
 -A input    -i eth1  -p udp            -s ${svcDNS} 1024:65535 -d ${eth1ADDR} domain -j
ACCEPT                                 # domain/udp (I)
```

In this way, we are able to build and verify the rules that we want, testing them as we go. The method also will be repeated for SMTP and HTTP to the service net, implementing the following policy:

- allow HTTP from the Internet to reach www.giacfortunes.com on the service net and receive replies
- allow the internal GIAC systems to reach www.giacfortunes.com and receive replies
- allow SMTP from our email relay system(s) on the Internet to mail.giacfortunes.com and back
- allow SMTP from the GIAC Internal mail server to reach mail.giacfortunes.com and back

```
#Rules merged from the new rule file: allow Internet access to www.giacfortunes.com
```

```
 -A input     -i eth0   -p tcp             -s ${extANY} 1024:65535 -d ${svcWEB} http -j ACCEPT
# http/tcp (I)
 -A forward   -i eth1   -p tcp             -s ${extANY} 1024:65535 -d ${svcWEB} http -j ACCEPT
# http/tcp (F)
 -A output    -i eth1   -p tcp             -s ${extANY} 1024:65535 -d ${svcWEB} http -j ACCEPT
# http/tcp (O)
 -A input     -i eth1   -p tcp       ! -y -s ${svcWEB} http -d ${extANY} 1024:65535 -j ACCEPT
# http/tcp (I)
 -A forward   -i eth0   -p tcp       ! -y -s ${svcWEB} http -d ${extANY} 1024:65535 -j ACCEPT
# http/tcp (F)
 -A output    -i eth0   -p tcp       ! -y -s ${svcWEB} http -d ${extANY} 1024:65535 -j ACCEPT
# http/tcp (O)


#Rules merged from the new rule file: allow internal employee access to www.giacfortunes.com

 -A input     -i eth2   -p tcp             -s ${intEMPL} 1024:65535 -d ${svcWEB} http -j ACCEPT
# http/tcp (I)
 -A forward   -i eth1   -p tcp             -s ${intEMPL} 1024:65535 -d ${svcWEB} http -j ACCEPT
# http/tcp (F)
 -A output    -i eth1   -p tcp             -s ${intEMPL} 1024:65535 -d ${svcWEB} http -j ACCEPT
# http/tcp (O)
 -A input     -i eth1   -p tcp       ! -y -s ${svcWEB} http -d ${intEMPL} 1024:65535 -j ACCEPT
# http/tcp (I)
 -A forward   -i eth2   -p tcp       ! -y -s ${svcWEB} http -d ${intEMPL} 1024:65535 -j ACCEPT
# http/tcp (F)
 -A output    -i eth2   -p tcp       ! -y -s ${svcWEB} http -d ${intEMPL} 1024:65535 -j ACCEPT
# http/tcp (O)

# allow Internet access to mail.giacfortunes.com
# (need to modify the server to prevent us being used as an smtp relay
 -A forward   -i eth0   -p tcp       ! -y -s ${svcMAIL} smtp -d 192.168.0.0/24 1024:65535 -j
ACCEPT                          #jrhok smtp/tcp (F)
 -A forward   -i eth1   -p tcp             -s ${extANY} 1024:65535 -d ${svcMAIL} smtp -j ACCEPT
#jrhok smtp/tcp (F)
 -A input     -i eth0   -p tcp             -s ${extANY} 1024:65535 -d ${svcMAIL} smtp -j ACCEPT
#jrhok smtp/tcp (I)
 -A input     -i eth1   -p tcp       ! -y -s ${svcMAIL} smtp -d ${extANY} 1024:65535 -j ACCEPT
#jrhok smtp/tcp (I)
 -A output    -i eth0   -p tcp       ! -y -s ${svcMAIL} smtp -d ${extANY} 1024:65535 -j ACCEPT
#jrhok smtp/tcp (O)
 -A output    -i eth1   -p tcp             -s ${extANY} 1024:65535 -d ${svcMAIL} smtp -j ACCEPT
#jrhok smtp/tcp (O)
```

Now allow time service. This is slightly different, because the firewall wants to play too.

- allow NTP queries from ntp.giacfortunes.com to go to a list of systems on the Internet and back
- allow NTP queries from GIAC Internal time server(s) to go to ntp.giacfortunes.com and back
- allow the firewall to sync time from ntp.giacfortunes.com using its eth1 interface and address

To do this, I needed an NTP time server on all four machines. The firewall has "real" ntp application, but for the Windows machines, I used WinSNTP.

```
#Rules merged from the new rule file: allow service net time server to sync to external source
and receive replies
 -A forward  -i eth0  -p udp            -s ${svcNTP} ntp -d ${extNTP} ntp -j ACCEPT
#jrh ntp/udp (F)
 -A forward  -i eth1  -p udp            -s ${extNTP} ntp -d ${svcNTP} ntp -j ACCEPT
#jrh ntp/udp (F)
 -A input    -i eth0  -p udp            -s ${extNTP} ntp -d ${svcNTP} ntp -j ACCEPT
#jrh ntp/udp (I)
 -A input    -i eth1  -p udp            -s ${svcNTP} ntp -d ${extNTP} ntp -j ACCEPT
#jrh ntp/udp (I)
 -A output   -i eth0  -p udp            -s ${svcNTP} ntp -d ${extNTP} ntp -j ACCEPT
#jrh ntp/udp (O)
 -A output   -i eth1  -p udp            -s ${extNTP} ntp -d ${svcNTP} ntp -j ACCEPT
#jrh ntp/udp (O)


#Rules merged from the new rule file: allow the firewall interface to sync time (this could be
on internal as easily)
 -A input    -i eth1  -p udp            -s ${svcNTP} ntp -d ${eth1ADDR} ntp -j ACCEPT
#jrh ntp/udp (I)
 -A output   -i eth1  -p udp            -s ${eth1ADDR} ntp -d ${svcNTP} ntp -j ACCEPT
#jrh ntp/udp (O)


#Rules merged from the new rule file: allow internal to sync with service net time
#This needs to be changed to be the internal time servers not the entire network
 -A forward  -i eth1  -p udp            -s ${intNTP} ntp -d ${svcNTP} ntp -j ACCEPT
# ntp/udp (F)
 -A forward  -i eth2  -p udp            -s ${svcNTP} ntp -d ${intNTP} ntp -j ACCEPT
# ntp/udp (F)
 -A input    -i eth1  -p udp            -s ${svcNTP} ntp -d ${intNTP} ntp -j ACCEPT
# ntp/udp (I)
 -A input    -i eth2  -p udp            -s ${intNTP} ntp -d ${svcNTP} ntp -j ACCEPT
# ntp/udp (I)
 -A output   -i eth1  -p udp            -s ${intNTP} ntp -d ${svcNTP} ntp -j ACCEPT
# ntp/udp (O)
 -A output   -i eth2  -p udp            -s ${svcNTP} ntp -d ${intNTP} ntp -j ACCEPT
# ntp/udp (O)
```

To implement management of the firewall, a list of systems which will be allowed to connect using SSH is developed and then the following policy implemented:

- allow SSH from the authorized GIAC Internal systems to connect to the firewall IP on eth2

```
#!# now allow ssh from inside to the firewall


 -A input   -i eth2  -p tcp      -s ${intSSH} 1024:65535 -d ${eth2ADDR} ssh -j ACCEPT
```

```
#jrhok# ssh/tcp (I)
 -A output    -i eth2   -p tcp! -y -s ${eth2ADDR} ssh -d ${intSSH} 1024:65535 -j ACCEPT
#jrhok# ssh/tcp (O)

# from the firewall to the router

-A output    -i eth0   -p tcp       -s ${eth0ADDR} 1024:65535 -d ${pubRTR} ssh -j ACCEPT
#jrhok# ssh/tcp (I)
-A input     -i eth0   -p tcp !     -s ${pubRTR} ssh -d ${eth0ADDR} 1024:65535 -j ACCEPT
#jrhok# ssh/tcp (O)

# and from an internal list through the firewall to the service net

-A input     -i eth2   -p tcp             -s ${intSSH} 1024:65535 -d ${pubSVC} ssh -j ACCEPT
# ssh (I)
 -A forward  -i eth1   -p tcp              -s ${intSSH} 1024:65535 -d ${pubSVC} ssh -j ACCEPT
# ssh (F)
 -A output   -i eth1   -p tcp              -s ${intSSH} 1024:65535 -d ${pubSVC} ssh -j ACCEPT
# ssh (O)
 -A input    -i eth1   -p tcp       ! -y -s ${pubSVC} ssh -d ${intSSH} 1024:65535 -j ACCEPT
# ssh (I)
 -A forward  -i eth2   -p tcp       ! -y -s ${pubSVC} ssh -d ${intSSH} 1024:65535 -j ACCEPT
# ssh (F)
```

And to allow GIAC Internal employees Web access to the Internet, we will do something different. We will implement IP masquerading (NAT) from all the GIAC Internal addresses out, but allow the internal machines to see the real IP of the external server. Ipchains does this by doing port translation so that it maps individual machine requests outbound to a unique port, and sends replies to that port to the internal server.

- Allow outbound HTTP from any GIAC Internal address to the Internet but masquerade the source IP. Allow replies to be sent back to the correct client.

```
#!# Allow internal employees (all networks) to access the Internet unrestricted (but masqueraded)
#!# Does not apply to packets toward the service net.
#Rules merged from the new rule file: allow employees external web access.
 -A input    -i eth2  -p tcp          -s ${intEMPL} 1024:65535 --dport http -j ACCEPT
#jrhnew http/tcp (I)
 -A forward  -i eth0                  -s ${intEMPL} ! -d ${svcWEB}          -j MASQ
#Masquerade
 -A output   -i eth0  -p tcp          -s ${eth0ADDR} 61000:65096 --dport http -j ACCEPT
#jrhnew http/tcp (O)
 -A input    -i eth0  -p tcp          ! -y --sport http -d ${eth0ADDR} 61000:65096 -j ACCEPT
#jrhnew http/tcp (I)
 -A output   -i eth2  -p tcp          ! -y --sport http -d ${intEMPL} 1024:65535 -j ACCEPT
#jrhnew http/tcp (O)

# and HTTPS (check this)

-A input    -i eth2  -p tcp           -s ${intEMPL} 1024:65535 --dport https -j ACCEPT
#jrhnew http/tcp (I)
 -A forward  -i eth0                  -s ${intEMPL} ! -d ${svcWEB}          -j MASQ
#Masquerade
```

```
 -A output   -i eth0  -p tcp          -s ${eth0ADDR} 61000:65096 --dport https -j ACCEPT
#jrhnew http/tcp (O)
 -A input    -i eth0  -p tcp          ! -y --sport https -d ${eth0ADDR} 61000:65096 -j ACCEPT
#jrhnew http/tcp (I)
 -A output   -i eth2  -p tcp          ! -y --sport https -d ${intEMPL} 1024:65535 -j ACCEPT
#jrhnew http/tcp (O)
```

When you have implemented all the rules you are aware of (and a few you weren't probably) you should:

Change the defaults in /etc/masonrc to:

```
NEWRULEPOLICY="deny"

DEFAULTPOLICY="deny"

FLUSHEDPOLICY="deny"
```

Remove /var/lib/mason/newrules

Stop and start the firewall (mason-gui-text) and let it run a while, making sure the rules work for all your normal traffic.

Once the firewall is running as you believe it should, re-run tripwire and carefully review the list of changes it identifies. You should only see things you expect. When you are satisfied that the system is ready for operation, re-run tripwire (again, storing a copy of the "new" database on write-protected media off-line) and you will be ready to connect your Shiny New Firewall to the Big Bad Internet.

Create the other firewalls (including individual servers)

The other firewalls should be configured using the same iterative process of:

1. Configure the default parameters in /etc/masonrc to "ACCEPT" newrules.

2. Add any "known" rules in /var/lib/mason/baserules

3. Run mason-gui-text iteratively for each service you wish to run.

4. Test these rules and perform any manual editing as needed.

5. Finish with a final test of the entire ruleset with “DENY” defaults in /etc/masonrc.

Note: This includes the servers themselves (each external and internal server will run mason while it does it’s normal duties (remember to include administrative tasks like backups) and then “set” the rules in place, take a tripwire snapshot, and move on.

.

A S S I G N M E N T 3

Audit your Architecture (primary firewall audit)

Define and Conduct the Audit

Conducting a security audit for anyone must begin with a statement of work, agreed to **in writing** about what you will do and that they cannot later claim damages. This is important whether you are performing the audit as a direct employee or as an external consulting job.

In preparing the audit plan, we propose something like the following to GIAC. Acceptance of the terms of this proposal includes an explicit agreement that:

- GIAC recognizes that the conduct of any security audit carries an inherent risk. That there are technical components of the audit that may cause problems with their systems including but not limited to: slow performance during the audit, the raising of security alerts or alarms (both at GIAC and at their ISPs) and other problems with specific applications or operating systems that may be adversely affected by the tests.

- GIAC agrees that they will in no way hold the auditor liable for any damages, specific or implied, that may arise from the conduct of this audit. Neither will they seek reparation for any lost-time, business, etc.

- GIAC further agrees that they will bear the sole responsibility for any incidental damages or outages caused at other Internet sites, including but not limited to their ISP, customers or members of the public who might be accessing GIAC resources during the audit itself.

(I am not a lawyer, but there needs to be a solid acknowledgement to the risks from the audit, and an agreement not to hold us responsible for any problems. Whether or not the ISP is at risk and should be involved will depend on the specific tests performed and where they are initiated.)

So, please get these needs taken care of up front (somehow).

Initial meeting with GIAC: Business requirements.
Based on an initial meeting with the CEO and CIO, an audit of GIAC's primary firewall has been commissioned. Their concerns and motivations are:

- The initial design was done with keeping cost low as a criterion, and they want a "second opinion" on the validity of both the architecture in general, and of the choices of technology (specifically Linux) to implement that architecture.

- As they grow, they will need to implement stricter security policies on a department-level. They realize (were told by their initial architect) that the "open" policy regarding employee Internet access may need to change. They wish to have an assessment of how this change in business policy will impact their network. They are looking for alternate (technical) solutions, with an impact assessment and approximate (relative) cost.

When asked for details surrounding the need for implementing department-specific Internet access policies, there were no clear requirements, simply a "we think we will need to tighten this down soon". It appeared that they had received some internal complaints about co-workers mis-use (some reports of pornographic images, some simply that people spend too much time on "The Net" at work.)

These concerns are common among growing companies. What starts out as a "we trust our employees" stance soon gives way to less idealistic concerns. The two main areas this concern manifests itself in are normally:

1. Productivity - The concern that people spend too much time "surfing". This can arise in any tough times for the company, or simply because of some employees perception that others are wasting time.

2. Legal and Human Resources issues – Usually based on the need of the company to provide a non-threatening work environment. The concern usually focuses on the display of inappropriate images or sounds that other employees find offensive.

These requirements need to be handled carefully and from the auditor's (our) perspective we need to allow the internal departments (usually mainly Human Resources) to take the lead role. They can identify what their specific needs are, and provide the required documentation.

Plan the Technical Audit
The technical approach to this audit will involve two phases. First we will review

the current systems to determine their compliance with the existing architecture. This will be followed by an analysis of the original business conditions and the current needs to determine any actual architectural or policy change recommendations.

Audit the Security Architecture
In this case, we have copies of the original security architecture design documents and detailed policies, documentation on the current systems as well as the full cooperation of both internal GIAC staff (upper management and IT technical staff) as well as the ISP. (Hey, it could happen ;-)

To determine compliance with the architecture, we will conduct several targeted scans from the Internet (with cooperation from the ISP) to determine toe overall effectiveness of the systems. These will take several forms:

1. Information gathering – using public name services, we will probe to find out as much information as possible about the GIAC infrastructure. This is designed to determine if what is available matches what is supposed to be.

    a. This part has very low risk and probably will not ring any alarms at all. It is simply the use of normal services to identify what is really publicly available.

    b. It can be conducted with no internal assistance at any time day or night. Estimated time, less than a day including documentation.

2. Information probing – now we use tools designed to pull "hidden" knowledge out of the exposed GIAC systems. Using tools that send various traffic types into GIAC at different points, we see what responses we get and use this to infer knowledge about the bits we can't directly see. (We all played "black box" as kids, right?)

    a. This may raise some alarms at either the ISP or in the GIAC network management center. While not specifically "attacks", we will be generating a number of probes that will be outside the normal traffic patterns.

    b. It will require alerting the ISP and GIAC staff (we want to be professionals about this).

    c. These probes can be performed by a single auditor and will take less than a day, including documenting the findings.

3. System Verification – We will use a combination of external systems and internal monitors to ensure that the existing system "functions as designed". This will require coordinated access to internal system logs as well as the ability to place "stealth" traffic monitors at specific places in the topology to determine whether a particular traffic type actually makes it through each layer in the architecture or not.

    a. This must be conducted separately from the previous steps to insure the different scanning efforts do not confuse the results.

    b. There will be specific traffic generated based on the original security architecture, and "scanners" (i.e. nmap on a laptop) and traffic monitors (laptop with either Ethereal or snort) will be on the "inside" and "outside" of the network. This is a system-wide test.

    c. External validation will take place first. This audit will verify that valid traffic is passing in both directions (this should be a trivial effort), then that explicit blocked traffic is indeed being dropped, and further that unexpected traffic (i.e. that which is not specifically blocked, but should be caught in the default deny rule) does not penetrate.

4. Failure Analysis - Once these "system wide" tests occur, test machines will be introduced into sections of the network where they would not ordinarily be present to examine scenarios simulating failure of a particular security component, to determine whether the remaining layers would in fact provide sufficient defense.

    a. This is fairly risky and very time consuming. It should raise many alarms and will definitely cause brief outages. This type of testing must occur during "off-hours" and be coordinated closely with all potentially impacted groups. The level of staff required will vary, but clearly will require people with access to all GIAC gear to monitor logs and potentially restart services.

    b. Specific "fail-safe" plans must be reviewed and be in place prior to conducting this phase. The level of impact before testing is aborted must be understood, and the people with the authority to make that call must be available.

    c. It requires physical access to secure areas of the network and may require introducing extra pieces of network gear. For example, it may be necessary to introduce a small hub between the inside interface of the

border router and the external interface of the firewall to generate traffic that would not ordinarily make it through the router.

d. This is where the "Active Response" of the snort IDSs will be verified. This will be a lengthy effort, probably continuing over at least a week and requiring availability of ISP phone support, internal GIAC IT staff and at least two auditors. (For each network segment on either side of every firewall and router, we need to setup test equipment, run a series of tests and monitor & document the results and then move the equipment)

5. System vulnerability – More a "host security" thing than a glamorous network based scan like nmap, but in order to perform a thorough audit, we need to identify all the critical servers and their applications, and perform some basic research on whether or not they are vulnerable to any known attacks. This involves visiting each system (working with the local system administrator) and checking versions of all the live services. For example, the main DNS server would be checked for known Linux vulnerabilities that have appeared since it was installed, as well as to determine if the version of BIND has been kept up to date. The email server would be examined to verify that whatever mail transport agent (MTA) was installed (sendmail, postfix, etc.) was properly configured. The web server would be checked for it's software level and basic configuration, etc. I include this since whether or not the network is secure, if the services you **do** run are vulnerable, there is no way you can keep intruders out. (The best you could hope for is that once they had exploited the latest BIND vulnerability, that they could not establish a connection to retrieve their root kit before the alarms went off from Tripwire and Snort.)

Analyze how the Security Architecture meets Current Business needs
This part of the audit process will consist of reviewing the original security architecture alongside current business requirements. Specifically, there was a request noted during the initial meetings for recommendations on putting further controls over employee Internet access. These requirements will be examined further to determine whether or not real changes are required. If it is determined that there are new functional requirements, they will be developed as a separate process. In this practical, I will address some potential modifications in the "Evaluate the Audit" section that will focus a bit more on this.

Conduct the Audit
Information Gathering

1. From our external systems, we will perform Whois and DNS queries on the giacfortunes.com domain.

2. From the list of public names and IP addresses returned, we will build a diagram of what the domain structure appears to be including location of all public servers.

    a. We know that www.giacfortunes.com, mail.giacfortunes.com, dns.giacfortunes.com and ntp.giacfortunes.com will all be listed from the Public net, plus the commerce.giacfotrunes.com server from the Customer net.

3. Additional public information can be obtained from the servers themselves. For example, using any number of available tools that recursively traverse web sites, we can potentially find links to related but unpublished sites (perhaps there is a link off www.giacfortunes.com for employees to find the VPN link or other information?). This may or may not prove fruitful, but takes fairly little effort.

Information Probing

From the base set of information provided by our public audit, we will now conduct more active scanning to see what we can determine that way. There are a number of tools available, but the examples I will give are from nmap. What we are looking for here is simply what information we (or a potential attacker) can glean about the giacfortunes.com architecture. Not only are we interested in the output of nmap, we would have an "accomplice" inside GIAC (one of their IT staff) monitor the system logs while we are probing and noting what was seen and what got in "under the radar".

1. Initially, we will try basic TCP and UDP scans against all the public server addresses (note, my testbed network contains only one server at this IP address)

    ```
    nmapnt -vv -sT 192.168.0.35
    ```

    returns nothing, so we try suppressing "ping" and run it again:

    ```
    nmapnt -vv -sT -P0 192.168.0.35
    ```

    This is more interesting. It returns that port 25 and port 80 are open (recall that there is only one server on my testbed network, GIAC would show only 80 open on www and 25 on mail).

    Author's note: nmap also correctly identified the OS as Windows (NT4 or 9x) based on tcp sequence prediction.

We continue this way running through other nmap scans: This one is the "half-open SYN scan" which also looks for open TCP services, although it is more "stealthy". This would be used to determine what degree of logging/alerting was taking place and again we find our standard services:

```
[hendrick@localhost hendrick]$ sudo nmap -vv -sS -P0 192.168.0.35
Password:

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Initiating SYN half-open stealth scan against  (192.168.0.35)
Adding TCP port 25 (state open).
Adding TCP port 80 (state open).
The SYN scan took 341 seconds to scan 1523 ports.
Interesting ports on  (192.168.0.35):
(The 1521 ports scanned but not shown below are in state: filtered)
Port       State      Service
25/tcp     open       smtp
80/tcp     open       http

Nmap run completed -- 1 IP address (1 host up) scanned in 342 seconds
[hendrick@localhost hendrick]$
```

and this run scanning the UDP ports found nothing exciting either. Good.

```
[hendrick@localhost hendrick]$ sudo nmap -vv -sU -P0 192.168.0.35
Password:

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Initiating FIN,NULL, UDP, or Xmas stealth scan against  (192.168.0.35)
The UDP or stealth FIN/NULL/XMAS scan took 1745 seconds to scan 1448
ports.
(no udp responses received -- assuming all ports filtered)
All 1448 scanned ports on  (192.168.0.35) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 1745 seconds
[hendrick@localhost hendrick]$
```

2. Next we will proceed to scan ranges surrounding the public servers, trying to determine if there are other machines that we can find. Since we are masquerading all hosts out through the firewall, it will be simple to locate that machine and add it to our list (192.168.0.3). On my testbed I skip this step (since there are no other hosts than the two I already know about)

System Verification

This will consist of two auditors and at least one internal GIAC IT administrator. The auditors conduct tests of each service and then test blocked services:

- Working from the security policy documentation, validate HTTP, HTTPS, SMTP, DNS, NTP, SSH, databse connectivity, backups, and indivudual and partner VPN connectivity.

- Now, still following the design specs, the audit documents attempts to send traffic that should get blocked:

    - Valid protocols to the wrong servers (e.g. SMTP to the DNS machine)

    - Protocols for non-existent applications (FTP from the Web server)

    - Invalid traffic: (e.g. spoofed addresses in and out, illegal header flags, source routed packets, etc.)

  Here are some examples of the tests and results:

  Sending ICMP echo to the firewall interface (with logging turned on):

```
Nov 15 00:41:38 localhost kernel: Packet log: input REJECT eth2 PROTO=1
10.192.0.1:8 10.0.0.1:0 L=60 S=0x00 I=10361 F=0x0000 T=128 (#133)
Nov 15 00:41:39 localhost kernel: Packet log: input REJECT eth2 PROTO=1
10.192.0.1:8 10.0.0.1:0 L=60 S=0x00 I=10363 F=0x0000 T=128 (#133)
Nov 15 00:41:40 localhost kernel: Packet log: input REJECT eth2 PROTO=1
10.192.0.1:8 10.0.0.1:0 L=60 S=0x00 I=10365 F=0x0000 T=128 (#133)
Nov 15 00:41:41 localhost kernel: Packet log: input REJECT eth2 PROTO=1
10.192.0.1:8 10.0.0.1:0 L=60 S=0x00 I=10367 F=0x0000 T=128 (#133)
```

```
And trying to telnet to the same interface:

Nov 15 00:44:18 localhost kernel: Packet log: input REJECT eth2 PROTO=6
10.192.0.1:1330 10.0.0.1:23 L=48 S=0x00 I=10376 F=0x4000 T=128 SYN (#133)
Nov 15 00:44:20 localhost kernel: Packet log: input REJECT eth2 PROTO=6
10.192.0.1:1330 10.0.0.1:23 L=48 S=0x00 I=10378 F=0x4000 T=128 SYN (#133)
Nov 15 00:44:26 localhost kernel: Packet log: input REJECT eth2 PROTO=6
10.192.0.1:1330 10.0.0.1:23 L=48 S=0x00 I=10380 F=0x4000 T=128 SYN (#133)

Make sure we can't do a DNS query to the firewall:
```

```
Nov 15 00:51:40 localhost kernel: Packet log: input DENY eth1 PROTO=17
192.168.0.35:3006 192.168.0.33:53 L=71 S=0x00 I=2134 F=0x0000 T=128 (#129)
Nov 15 00:51:42 localhost kernel: Packet log: input DENY eth1 PROTO=17
192.168.0.35:3006 192.168.0.33:53 L=71 S=0x00 I=2646 F=0x0000 T=128 (#129)
Nov 15 00:51:46 localhost kernel: Packet log: input DENY eth1 PROTO=17
192.168.0.35:3006 192.168.0.33:53 L=71 S=0x00 I=3414 F=0x0000 T=128 (#129)
```

In this way, we verify each of the rules either positive or negative.

Failure Analysis

Here is where we put things to tests that they would never really experience. First of all, we assume complete knowledge about our systems. Network topology, system and software inventory, even application configuration details that would never (we hope) be available to an attacker. We place probes in places that should never have probes placed there (woo hoooo, that's cold…) to see how the security systems behave. The intention here is to verify that the systems really will behave as they were designed. So if there is an "inner layer" of defenses that should never see use, here is where we test it.

1. Network probing. First, we attempt to probe the firewall itself to see what we can determine about it. What we are looking for is not so much the output of "nmap" itself (although if we do see anything, we will dig further). What we care about is whether or not the firewall (router, IDS) catches the scan in it's logs. Remember, much of this traffic should never be hitting the firewall (it should be caught at the router). The things we are testing are:

    a. If the router is compromised, does the firewall still catch it?

    b. Will the system log it? Generate the proper alert?

    c. If traffic makes it through the firewall that should not, will the IDS sound the alarm (and will it indeed knock down the connection?)

    We know it's a firewall, so we'll try an ACK scan:

    ```
    [hendrick@localhost hendrick]$ sudo nmap -vv -sA -P0 192.168.0.3
    Password:

    Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
    Initiating ACK scan against  (192.168.0.3)
    ```

```
The ACK scan took 2519 seconds to scan 1523 ports.
All 1523 scanned ports on  (192.168.0.3) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 2519 seconds
```

Hmmm, not much there. Now a half-open "SYN" scan (to see if anything sends a RST)

```
[hendrick@localhost hendrick]$ sudo nmap -vv -sS -P0 192.168.0.3
Password:

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Initiating SYN half-open stealth scan against  (192.168.0.3)
The SYN scan took 2919 seconds to scan 1523 ports.
All 1523 scanned ports on  (192.168.0.3) are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 2919 seconds
```

And finally we dig a bit deeper to validate things. Logging on to the firewall, we attempt to generate some traffic in the wrong direction. This is to verify that our "stealth" configuration is working as designed. Specifically, we try to "ping" out the external interface:

```
[hendrick@localhost hendrick]$ ping www.yahoo.com
PING www.yahoo.akadns.net (64.58.76.224) from 192.168.0.3 : 56(84)
bytes of data.
ping: sendto: Operation not permitted
ping: sendto: Operation not permitted
ping: sendto: Operation not permitted

--- www.yahoo.akadns.net ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
```

Now we check DNS to make sure we are actually getting name resolution from the Service net (we should be going to 192.168.0.35 and not our Internet DNS secondary at 0.4)

```
[hendrick@localhost hendrick]$ nslookup
> server 192.168.0.4
Default server: 192.168.0.4
Address: 192.168.0.4#53
> www.redhat.com
;; connection timed out; no servers could be reached
> server 192.168.0.35
Default server: 192.168.0.35
Address: 192.168.0.35#53
> www.redhat.com
Server:        192.168.0.35
Address:       192.168.0.35#53

Non-authoritative answer:
Name:   www.redhat.com
Address: 216.148.218.195
```

```
Name:   www.redhat.com
Address: 216.148.218.197
>
```

Evaluate the Audit

Review the Technical Findings

GIAC provided us with full diagrams and descriptions of the network design they purchased. We reviewed these and the information gathered during the technical audit and find several areas for improvement:

Architecture Review: Original Design Goals

In conducting the audit of GIAC's network architecture, there were no major security flaws uncovered. The design itself implements its original architecture reasonably well. There are a number of layers of security that seem to function together and support each other. The goal of providing basic "defense in depth" while allowing free web and email access to employees and securing corporate data was met. The trade-offs of functionality vs. cost are acknowledged choices and the architecture does what it claims to do.

Design Review: Recommendations for Addressing Changing Business Needs

In order to address the changing business needs of GIAC Enterprises, we wish to identify areas that are at risk and present a number of recommendations for improving the architecture.

1. There is a lack of reduncancy to the Internet - Although there are multiple connections to the Internet, there is no ability configured to fail-over either manually or automatically if one becomes unusable. The circuits already exist, and the border routers are all from a common vendor.

    a. We recommend undertaking a project to implement Cisco's HSRP between the Public and Customer links. This would not be a trivial project, however all the pieces are in place and the potential gains make it a strong recommendation.

2. Single internal firewall - The use of one internal firewall connecting all four main networks poses the risk that failure of this system whether through accident or malice would result in a major loss of communication for GIAC.

a. We recommend undertaking a project to investigate ways to mitigate this risk. Either by splitting the functionality across multiple firewalls or by replacing this one with a pair of firewalls configured for auto-fail-over.

3. Better management of Internet access – This was raised by GIAC. They desire the ability to better manage the productivity and content issues identified.

    a. We recommend replacing the current open access with an authenticated proxy with content filtering capabilities. This will give several benefits:

        i. By requiring employees to login, it will be obvious that they may be held accountable for their actions. This alone may address some of the perceived problems.

        ii. By having all access authenticated, proxy use reports can be developed that make it easy to identify any potential problems and address them through normal management channels.

        iii. The ability to perform content filtering will allow GIAC the ability to manage any inappropriate use.

    Important Caveat: The management of a content filtering proxy comes with a fairly high cost. Simply maintaining the logs, updating the filter lists and managing the desktop support issues will be an added full-time set of tasks. The technical implementation

Audit the Implementation

Although the design was reasonable to meet the goals as GIAC was starting, there were some choices made that are worthy of being revisited at this point.

1. The firewalls themselves are running as traditional packet filters. There are some attacks they are not designed to prevent

    Since ipchains can only look at a packet at a time, setting both SYN and ACK will allow an attacker to probe all open ports (which are all the high order ones between 61000 and 65096). This probably will not result in any serious trouble, since even guessing a port with an open connection the traffic will only be directed at a web browser on an employee machine.

There is probably a decent DDoS here but it would be one that snort could be configured to look for.

a. We recommend implememting snort rules to do the stateful inspection, and if it sees SYN/ACKs inbound without a recent SYN outbound, log and alert (may not need to take active response).

b. We recommend transitioning to iptables from ipchains as soon as possible

c. There has been increased work in other active defense technology such as “Tar Pits” that slow down the process of scanning address spaces and effectively neutralize them. We recommend investigating http://www.hackbusters.net/LaBrea.html for potential implementation outside the firwall.

2. There is very limited central logging or management of the security services.

All systems maintain their own system logs. Correlating events across multiple servers requires running the “backup” scripts to collect the logs.

a. we recommend configuring central logging servers in three places: One behind the Internet facing links. A second on the Internal Services network. The third on the Database network.

The firewall implementation lacks a central console that can manage policy across all systems. Making changes requires visiting each system individually.

b. We recommend monitoring support time and helpdesk calls involving maintaining the multiple firewalls individually. There may be no need to change this, but it is worth understanding what the current level of effort is.

3. Performance of the firewalls could become a limiting factor.

As load increases, the internal firewall especially may well become overaloaded. A single machine it is responsible for all major traffic that crosses the GIAC internal network backbone.

a. Recommend this be split with one fw for employees <-> IDS net and another FW or two for database & servers <-> IDS net. Watch the load

and determine if splitting it would suffice or if a Much Faster System ™ (like a big PIX or a Nokia 440) is needed.

4. How well do we believe the architecture meets their (current) needs?

    a. Need to address internet access with proxies and allow more services than just HTTP/S

    b. Need to investigate other internal needs (how have sales been doing? Is there any planned expansion or big new partnerships on the horizon? Or are things going reasonably well?)

    c. How is IT support doing? Do the staff have time to monitor all the systems, keep up on patches, etc?

A S S I G N M E N T 4

Design Under Fire

Target Selection

I have chosen to research the design presented by Lenny Zeltser in February 2001. His multi-tier client-server architecture was based on Cisco VPNs, routers and PIX firewalls. This architecture caught my attention, since I am currently working on two (real) projects using a very similar three tiered architecture with presentation layer servers running Java servlets, talking to BEA WebLogic applications running Enterprise Java Beans and talking with back-end Oracle databases.

My method for designing the specified attacks is simply this:

Info Gathering

- Public information – looking for related links on their web site(s), etc.
- Public name services (Internic registrations, whois, name services, dig/nslookup)
- Probing their address space.
- Social Engineering (how else did I get this cool network implementation diagram?)

Vulnerability Research

Once having identified the target OS, I run this through Internet searches in a number web sites. I list the recent vulnerabilities later I had to choose from later in this section.

Design and Prepare the Attacks

After selecting some tasty morsels from these sources, I set about to gather the exploit tools, build or install the scripts or programs on my (already compromised) servers, and wait for night to fall…

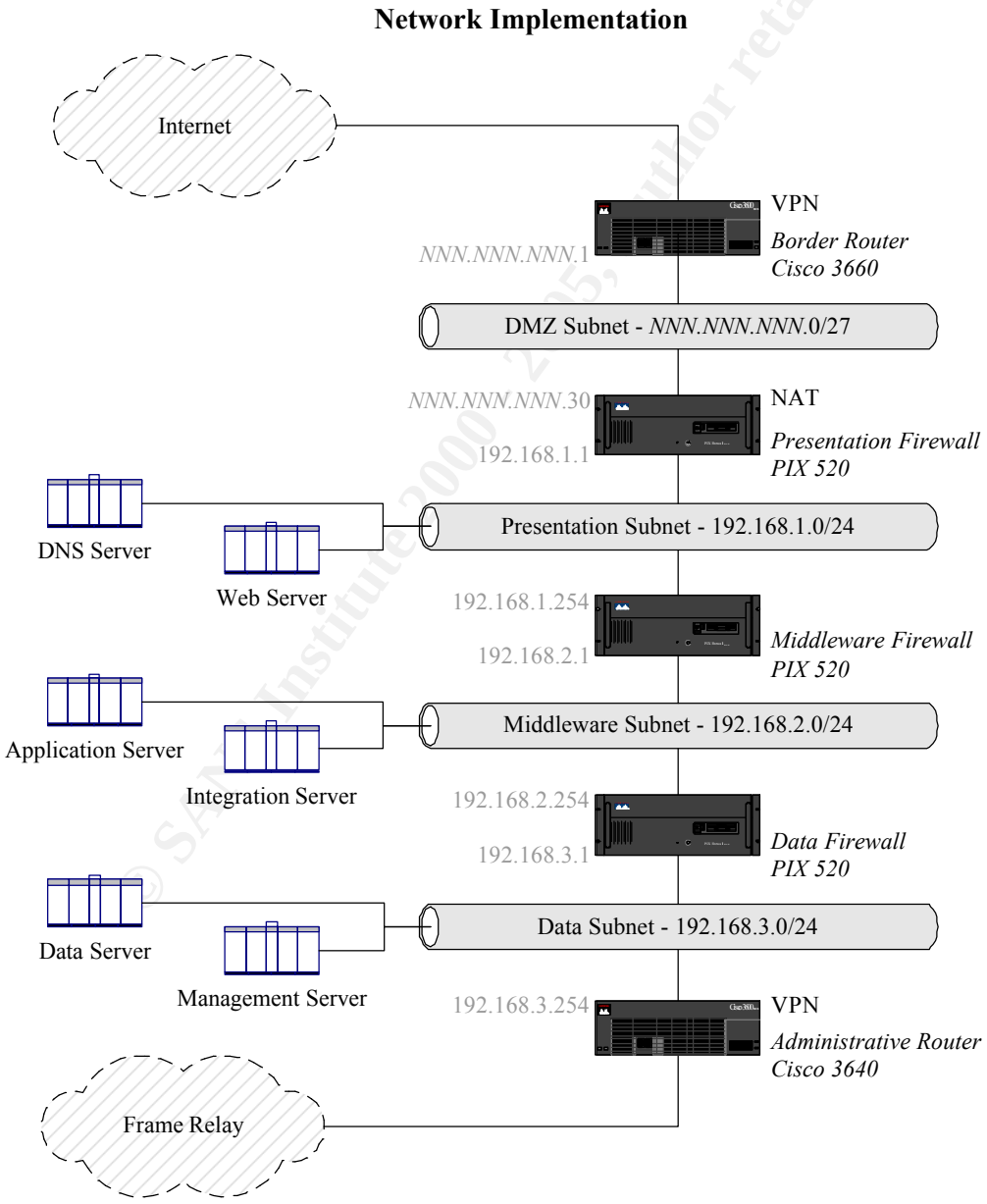


Figure 1.5-3 from Lenny Zeltser GCFW Practical Assignment, Capitol SANS December 2000

Map their network
In this case I have an entire architecture diagram to work from. In reality, crackers usually have much less to go on and need to create a map on their own using normal name services like Whois and nslookup, and tools such as nmap to probe the IP addresses around any servers they can find.

Select the Attacks

Once the targets have been identified, we look for potential vulnerabilities. I found these within a few minutes at the web sites listed below, and chose my attacks from them.

CERT Coordination Center Vulnerability Notes Database

- **10/1/2001: Cisco PIX Firewall Manager stores enable password in plain text**
    - http://www.kb.cert.org/vuls/id/639507
    - A vulnerability exists in the way the Cisco Pix Firewall Manager stores authentication credentials which could allow local attackers to have read access to the enable password
- **2/28/2001: Cisco IOS/X12-X15 has default SNMP read/write string of "cable-docsis"**
    - http://www.kb.cert.org/vuls/id/840665

- There is a vulnerability that permits unauthorized access to several switch and router products manufactured by Cisco Systems. An attacker who gains access to an affected device can read and modify its configuration, creating a denial-of-service condition, an information leak, or both.

- **2/28/2001: Cisco IOS/CatOS exposes read-write SNMP community string via traversal of View-based Access Control MIB (VACM) using read-only community string**

  - http://www.kb.cert.org/vuls/id/645400

- **10/9/2001: Cisco IOS vulnerable to denial of service via Cisco Discovery Protocol**

  - http://www.kb.cert.org/vuls/id/139491

  - The Cisco Internetwork Operating System (IOS) contains a vulnerability in its processing of Cisco Discovery Protocol (CDP) packets. By sending large numbers of crafted CDP packets to an affected device, a nearby remote attacker can consume all available memory resources, causing the device to either crash or stop responding. It is important to note that the CDP protocol operates at the data link layer of the ISO/OSI model, so it cannot be propagated by network and transport layer protocols such as IP and TCP, respectively. As such, attackers will only be able to attack devices on networks they can access directly (ie. without IP routing). However, this also means that many of the strategies commonly used to block malicious traffic (such as port filtering) cannot be used to prevent attackers from reaching an affected host.

Common Vulnerabilities and Exploits

- **2001-10-12: IOS DOS attack**
    - http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0750
    - Cisco IOS 12.1(2)T, 12.1(3)T allow remote attackers to cause a denial of service (reload) via a connection to TCP ports 3100-3999, 5100-5999, 7100-7999 and 10100-10999.
    - NOTE Added from CERT. This is a SYN scan vulnerability
- **2001-08-29:**
    - http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0650
    - Cisco devices IOS 12.0 and earlier allow a remote attacker to cause a crash, or bad route updates, via malformed BGP updates with unrecognized transitive attribute.

Security Focus "Bugtraq" archive

http://www.securityfocus.com/cgi-bin/vulns.pl

- **Jun 27 2001: Cisco IOS HTTP Configuration Arbitrary Administrative Access Vulnerability**
    - http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=2936
    - It is possible to gain full remote administrative access on devices using affected releases of IOS. By using a URL of http://router.address/level/$NUMBER/exec/.... where $NUMBER is an integer between 16 and 99, it is possible for a remote user

to gain full administrative access.

- **Jul 25, 2001: Cisco IOS UDP Denial of Service Vulnerability**
    - http://www.securityfocus.com/bid/3096
    - The problem reportedly occurs when a large number of UDP packets are sent to device running IOS. This causes the system to use all available CPU resources and thus become unresponsive. The device may have to be reset manually if the attack is successful.
- **Jun 14, 2001: Cisco NRP2 Unauthorized Telnet Access Vulnerability**
    - http://www.securityfocus.com/bid/2874
    - The Cisco Node Route Processor 2 card is a module designed to enhance the services of the Cisco 6400 series broadband aggregators. It is distributed by Cisco Systems.

      A problem in the Node Route Processor 2 (NRP2) makes it possible for remote users to gain unauthorized access to vtys. The default configuration of the NRP2 allows access to the vtys of the module when no password has been set. By default configuration, the NRP2 should allow no access until a password has been set.

      This makes it possible for a remote user to gain access to systems behind the NRP2 module, potentially accessing secure systems.
- **Oct 22, 2000: Cisco IOS Extended Access List Failure Vulnerability**
    - http://www.securityfocus.com/bid/1880

- IOS is the firmware used by many Cisco network devices.

  In some versions of IOS 12.x (verified on 12.1(4) and reportedly other versions), certain rules in extended access control lists will not be enforced. This may allow attackers to access vulnerable network services thought to be protected by the access control lists. The reason for this behaviour is not yet known.

- **Apr 6, 2001: Cisco PIX TACACS+ Denial of Service Vulnerability**

  - http://www.securityfocus.com/bid/2551

- **Oct 15, 2000: OpenBSD Pending ARP Request Remote DoS Vulnerability**

  - http://www.securityfocus.com/bid/1759

  - OpenBSD is vulnerable to a remotely exploitable denial of service condition. The problem seems to be a lack of limits on the storage of pending arp requests, and a failure to handle the condition of too many. If an attacker somehow causes a victim machine to send out too many arp requests, it can cause a kernel panic and the target system to halt.

- **Jul 18, 2001 OpenBSD: Multiple Vendor Telnetd Buffer Overflow Vulnerability**

  - http://www.securityfocus.com/bid/3064

  - A boundary condition error exists in telnet daemons derived from the BSD telnet daemon.

    Under certain circumstances, the buffer overflow can occur when a

combination of telnet protocol options are received by the daemon. The function responsible for processing the options prepares a response within a fixed sized buffer, without performing any bounds checking.

This vulnerability is now being actively exploited. A worm is known to be circulating around the Internet

Execute the Attacks

It was a dark and stormy night…

Seriously, after looking over the vulnerabilities above, there appear to be several good candidates. These are the ones I would try first:

Attacks against the Firewall itself:

The border router is a Cisco 3660 running IOS 12.1(3) according to our mole. (even if we did not know this, the attacks are tempting enough to try on any Cisco device)

Attack # 1 – Attempt admin access on the router

The first attempt would be to use this recent attack against the Border Router. The exploit can easily be scripted. Hopefully it would get us through both the border router and the NAT box.. Given the extended access list vulnerability as well, (http://www.securityfocus.com/bid/1880) we may get in.

- http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=2936

Jun 27 2001: Cisco IOS HTTP Configuration Arbitrary Administrative Access Vulnerability

It is possible to gain full remote administrative access on devices using affected releases of IOS. By using a URL of http://router.address/level/$NUMBER/exec/.... where $NUMBER is an integer between 16 and 99, it is possible for a remote user to gain full administrative access.

Attack #2 – Attack against a Presentation Layer host

If we could compromise the web site, we would be much closer to being into where the real goods are. This presentation layer host no doubt has access using JNDI or some other connection mechanism to the applications servers, and then the database.

Many systems run BSD based telnetd's. It may be that these systems do not allow it, but it's a common service (and they are behind two filtering devices, they might be open). Once we are on the firewall, we would need to try to exploit this using the existing worm code.

- http://www.securityfocus.com/bid/3064

- **Jul 18, 2001 OpenBSD: Multiple Vendor Telnetd Buffer Overflow Vulnerability**

  A boundary condition error exists in telnet daemons derived from the BSD telnet daemon.
  Under certain circumstances, the buffer overflow can occur when a combination of telnet protocol options are received by the daemon. The function responsible for processing the options prepares a response within a fixed sized buffer, without performing any bounds checking. This vulnerability is now being actively exploited. A worm is known to be circulating around the Internet.

Attack # 3 – DoS against the site

We'll save this one for last (after we hopefully compromise either the Web or DNS server). Again, simple to script from my 50 zombies. It is either going to bring them down instantly or not, depending on whether they allow access at those high ports.

- http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0750

- **Cisco IOS 12.1(2)T, 12.1(3)T allow remote attackers to cause a denial of service** (reload) via a connection to TCP ports 3100-3999, 5100-5999, 7100-7999 and 10100-10999. NOTE Added from CERT. This is a SYN scan vulnerability

This one would be fairly easy to guard against, and if they have been monitoring the vulnerability alerts, they would have blocked these ports and rendered the attack ineffective.

R E F E R E N C E S

Specific software used

All software listed is available on the Internet. Where appropriate, I give specific URLs and excerpts from the package documentation (including appropriate credits). If not otherwise specified, the software is for Linux and available from the RedHat site or directly on the distribution CDs.

Operating System

RedHat Linux 7.1 (2.4.9-6 kernel)

The Operating System on which to base all servers including the firewalls. (although as of the writing, 7.2 has been released). This was selected for cost and functionality. Since GIAC is in "startup mode", it cannot afford the high (some would say exorbitant) pricing for "COTS" software (so named because you need to lie down when you see the licensing cost for most of these "common off the shelf" components.) It also cannot afford to risk it's intellectual property, so the cost factor was balanced with the state of the RedHat distribution and support structure. Operations procedures must be adhered to in order to keep up with critical vulnerabilities, and RedHat (currently) has one of the better (IMHO) support structures.

Firewall Software

ipchains 1.3.10

The base firewall software. It has been in use long enough for any critical design flaws to have been discovered, and there are many sources of excellent documentation. Probably the most useful I found was http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html

Ipchains is part of the RedHat distribution.

mason 0.13.0.92

Mason was used to assist with firewall configuration. This software "auto generates" a set of rules based on an initial default policy and actual use. This was specified in order to simplify configuration as well as to allow further hardening for critical servers (i.e. all GIAC servers run "private" firewalls that provide additional protection for themselves. Mason will allow the system administrators to easily keep their configurations up-to-date.)

The main source of documentation (and source code) is at http://mason.stearns.org , but there are many sites.

Here is one that maintains a set of pre-made rules for various firewall configurations:

http://rama.poly.edu/~ejcli/rsghome/projects/linux/mason/

Mason was shipped as part of the RedHat distribution.

Intrusion Detection Systems
Host Based IDS
tripwire-2.3.0

This has been the "gold standard" for host based intrusion detection systems since it's creation by Gene Kim and Gene Spafford in the mid 1990s. It uses strong cryptography to build a set of signatures for critical system resources (basically any or every file or directory on the system). If any modifications occur, tripwire will be able to detect this event. The trick is to (1) select a set of files that do not change (it is amazing how many files change as part of "normal" operation) and (2) actually have tripwire "trigger" something when it sees a change (like paging someone or perhaps even halting the system!) Somewhat high-maintenance, but there is nothing else that even comes close.

Tripwire is part of the RedHat distribution.

Network IDS
snort 1.7-3
Although there are a number of "packet sniffers" available, this one (like tripwire) sets the standard. It is very fully featured and allows configuration to monitor for many known attack signatures and perform custom actions for each. One of the most useful is the ability to force a connection to terminate (send a RST to the offending host). This is implemented behind the firewalls at several levels of the

network topology, and builds on the “defense in depth” strategy.

Snort is part of the RedHat distribution.

Other Utilities

In building the testbed for the primary firewall, I attempted to make it as realistic as possible and actually ran services representative of a minimal set required for GIAC to go “on-line”. This also allowed me to use “mason” to assist with the ruleset creation, since in order to generate the rules, it needed “real” traffic. My criteria for selecting this software was simple:

- It must run on either Windows NT or Windows 2000, since that was the available set of machines I had to implement the testbed network.
- It must be either free or available as a free limited-time evaluation or trial version.

That said, I make no endorsement of any of the utilities below other than they did operate correctly in my testbed environment for the short time I needed them. Had I been able to use Linux machines totally for the testbed, my selections would have been different. Your mileage may vary.

Services

apache_1.3.22-win32-x86

This web server was used to run the “www.giacfortunes.com” site. It is widely available, and I got my copy from:

http://www.apache.org/dist/httpd/binaries/win32/

Merak Mail Server Version 4.10.040

This was used to actually pass SMTP and POP3 email. While serious overkill for the testbed, this trial download version was obtained from:

http://www.icewarp.com/

Simple DNS Plus Version 3.20.02

This Windows name server was used to simulate dns.giacfortunes.com and the corresponding internal and Internet servers on all three legs of the primary firewall. It was configured to do caching and forwarding only. The 14 day evaluation copy came from:

http://www.jhsoft.com/

WinSNTP Version 3.0

This Windows implementation of an NTP protocol time server allows you to specify a single server (name or IP address) for synchronization. It was used to provide the external, service net and internal services for my testbed network.

The 30 day trial download came from:

http://www.coetanian.com/

Packet Utilities

To perform some of the nitty-gritty tests that cried out for some decent "swiss army IP wood-chipper" functionality, I used the following software:

nc1.1

Netcat generates and listens to arbitraty TCP and UDP connections. It was used to perform testing on the firewall (before I got some of the trial DNS, SMTP and NTP software).

I got the NT distribution from: http://www.atstake.com/research/tools/nc11nt.zip

nmap 2.53

Nmap is the standard utility for network exploration and security scanning. I used it to verify the firewall rules by generating many different scans.

The Unix version is available at: http://www.insecure.org/

and there is an NT port at:
http://www.eeye.com/html/databases/software/nmapnt/nmapntsp1.zip

Etheral Version 0.8.19

Ethereal is a free network protocol analyzer. While there are many tools that do this (including snort and tcpdump), ethereal provides a nice GUI interface on NT. I used this extensively to monitor what was going on on various parts of the network. It requires the pcap libraries to run in packet capture mode.

Available from: http://www.ethereal.com/ and http://netgroup-mirror.ethereal.com/winpcap/ although the main site for Winpcap is: http://netgroup-serv.polito.it/winpcap/

IPv4 Network Calculator

I used this to calculate the subnets, masks, etc. when defining the GIAC address space. Recall that I made use of the ${intEMPL} for all internal GIAC and the ${extANY} for "legal" addresses assigned to GIAC by their ISPs.

This calculator is online at:

http://www.telusplanet.net/public/sparkman/netcalc.htm

The following are tables generated from this tool that I used to help me visualize and test the various network address spaces:

List of 16 networks for the 192.168.0.0 network with the subnet mask 255.255.255.240 (or /28)

Network	Purpose	Hosts		Broadcast Address
		from	to	
192.168.0.0	Public external	192.168.0.1	192.168.0.14	192.168.0.15
192.168.0.16	Public DMZ	192.168.0.17	192.168.0.30	192.168.0.31
192.168.0.32	Public Service Net	192.168.0.33	192.168.0.46	192.168.0.47
192.168.0.48	Customer External	192.168.0.49	192.168.0.62	192.168.0.63
192.168.0.64	Customer DMZ	192.168.0.65	192.168.0.78	192.168.0.79
192.168.0.80	Customer Service Net	192.168.0.81	192.168.0.94	192.168.0.95

192.168.0.96	Partner external	192.168.0.97	192.168.0.110	192.168.0.111
192.168.0.112	Partner DMZ	192.168.0.113	192.168.0.126	192.168.0.127
192.168.0.128	Partner	192.168.0.129	192.168.0.142	192.168.0.143
192.168.0.144	unused	192.168.0.145	192.168.0.158	192.168.0.159
192.168.0.160	unused	192.168.0.161	192.168.0.174	192.168.0.175
192.168.0.176		192.168.0.177	192.168.0.190	192.168.0.191
192.168.0.192		192.168.0.193	192.168.0.206	192.168.0.207
192.168.0.208		192.168.0.209	192.168.0.222	192.168.0.223
192.168.0.224		192.168.0.225	192.168.0.238	192.168.0.239
192.168.0.240		192.168.0.241	192.168.0.254	192.168.0.255

List of 8 networks

for the 192.168.0.0 network with the subnet mask 255.255.255.224 or /27

Network	Hosts		Broadcast Address
	from	to	
192.168.0.0	192.168.0.1	192.168.0.30	192.168.0.31
192.168.0.32	192.168.0.33	192.168.0.62	192.168.0.63

192.168.0.64	192.168.0.65	192.168.0.94	192.168.0.95
192.168.0.96	192.168.0.97	192.168.0.126	192.168.0.127
192.168.0.128	192.168.0.129	192.168.0.158	192.168.0.159
192.168.0.160	192.168.0.161	192.168.0.190	192.168.0.191
192.168.0.192	192.168.0.193	192.168.0.222	192.168.0.223
192.168.0.224	192.168.0.225	192.168.0.254	192.168.0.255

List of 4 networks

for the 192.168.0.0 network with the subnet mask 255.255.255.192 or /26

Network	Purpose	Hosts		Broadcast Address
		from	to	
192.168.0.0		192.168.0.1	192.168.0.62	192.168.0.63
192.168.0.64		192.168.0.65	192.168.0.126	192.168.0.127
192.168.0.128		192.168.0.129	192.168.0.190	192.168.0.191
192.168.0.192		192.168.0.193	192.168.0.254	192.168.0.255

List of 2 networks

for the 192.168.0.0 network with the subnet mask 255.255.255.128 or /25

Network	Hosts		Broadcast Address
	from	to	
192.168.0.0	192.168.0.1	192.168.0.126	192.168.0.127
192.168.0.128	192.168.0.129	192.168.0.254	192.168.0.255

And here are the similar tables for the 10.0.0.0 range. Recall we use ranges from the “64 networks” division internally for various departments both local and partner counterparts and aggregate up from there with the smaller masks (Recall: 10.192.0.0/10 is all employees, 10.128.0.0/10 is the internal Database network, 10.64.0.0/10 is “Internal Services” and 10.0.0.0/10 is the “IDS” network behind the firewalls.) The “2, 8, 16 and 32” network tables are listed for completeness.

List of 2 networks

for the 10.0.0.0 network with the subnet mask 255.128.0.0 (or /9)

Network	Hosts		Broadcast Address
	from	to	
10.0.0.0	10.0.0.1	10.127.255.254	10.127.255.255
10.128.0.0	10.128.0.1	10.255.255.254	10.255.255.255

List of 4 networks

for the 10.0.0.0 network with the subnet mask 255.192.0.0 (or /10)

Network	Hosts		Broadcast Address
	from	to	
10.0.0.0	10.0.0.1	10.63.255.254	10.63.255.255
10.64.0.0	10.64.0.1	10.127.255.254	10.127.255.255
10.128.0.0	10.128.0.1	10.191.255.254	10.191.255.255
10.192.0.0	10.192.0.1	10.255.255.254	10.255.255.255

List of 8 networks

for the 10.0.0.0 network with the subnet mask 255.224.0.0 (or /11)

Network	Hosts		Broadcast Address
	from	to	
10.0.0.0	10.0.0.1	10.31.255.254	10.31.255.255
10.32.0.0	10.32.0.1	10.63.255.254	10.63.255.255
10.64.0.0	10.64.0.1	10.95.255.254	10.95.255.255
10.96.0.0	10.96.0.1	10.127.255.254	10.127.255.255
10.128.0.0	10.128.0.1	10.159.255.254	10.159.255.255
10.160.0.0	10.160.0.1	10.191.255.254	10.191.255.255
10.192.0.0	10.192.0.1	10.223.255.254	10.223.255.255
10.224.0.0	10.224.0.1	10.255.255.254	10.255.255.255

List of 16 networks

for the 10.0.0.0 network with the subnet mask 255.240.0.0 (or /12)

Network	Hosts		Broadcast Address
	from	to	
10.0.0.0	10.0.0.1	10.15.255.254	10.15.255.255
10.16.0.0	10.16.0.1	10.31.255.254	10.31.255.255
10.32.0.0	10.32.0.1	10.47.255.254	10.47.255.255

10.48.0.0	10.48.0.1	10.63.255.254	10.63.255.255
10.64.0.0	10.64.0.1	10.79.255.254	10.79.255.255
10.80.0.0	10.80.0.1	10.95.255.254	10.95.255.255
10.96.0.0	10.96.0.1	10.111.255.254	10.111.255.255
10.112.0.0	10.112.0.1	10.127.255.254	10.127.255.255
10.128.0.0	10.128.0.1	10.143.255.254	10.143.255.255
10.144.0.0	10.144.0.1	10.159.255.254	10.159.255.255
10.160.0.0	10.160.0.1	10.175.255.254	10.175.255.255
10.176.0.0	10.176.0.1	10.191.255.254	10.191.255.255
10.192.0.0	10.192.0.1	10.207.255.254	10.207.255.255
10.208.0.0	10.208.0.1	10.223.255.254	10.223.255.255
10.224.0.0	10.224.0.1	10.239.255.254	10.239.255.255
10.240.0.0	10.240.0.1	10.255.255.254	10.255.255.255

List of 32 networks

for the 10.0.0.0 network with the subnet mask 255.248.0.0 (or /13)

Network	Hosts		Broadcast Address
	from	to	
10.0.0.0	10.0.0.1	10.7.255.254	10.7.255.255
10.8.0.0	10.8.0.1	10.15.255.254	10.15.255.255
10.16.0.0	10.16.0.1	10.23.255.254	10.23.255.255
10.24.0.0	10.24.0.1	10.31.255.254	10.31.255.255

10.32.0.0	10.32.0.1	10.39.255.254	10.39.255.255
10.40.0.0	10.40.0.1	10.47.255.254	10.47.255.255
10.48.0.0	10.48.0.1	10.55.255.254	10.55.255.255
10.56.0.0	10.56.0.1	10.63.255.254	10.63.255.255
10.64.0.0	10.64.0.1	10.71.255.254	10.71.255.255
10.72.0.0	10.72.0.1	10.79.255.254	10.79.255.255
10.80.0.0	10.80.0.1	10.87.255.254	10.87.255.255
10.88.0.0	10.88.0.1	10.95.255.254	10.95.255.255
10.96.0.0	10.96.0.1	10.103.255.254	10.103.255.255
10.104.0.0	10.104.0.1	10.111.255.254	10.111.255.255
10.112.0.0	10.112.0.1	10.119.255.254	10.119.255.255
10.120.0.0	10.120.0.1	10.127.255.254	10.127.255.255
10.128.0.0	10.128.0.1	10.135.255.254	10.135.255.255
10.136.0.0	10.136.0.1	10.143.255.254	10.143.255.255
10.144.0.0	10.144.0.1	10.151.255.254	10.151.255.255
10.152.0.0	10.152.0.1	10.159.255.254	10.159.255.255
10.160.0.0	10.160.0.1	10.167.255.254	10.167.255.255
10.168.0.0	10.168.0.1	10.175.255.254	10.175.255.255
10.176.0.0	10.176.0.1	10.183.255.254	10.183.255.255

10.184.0.0	10.184.0.1	10.191.255.254	10.191.255.255
10.192.0.0	10.192.0.1	10.199.255.254	10.199.255.255
10.200.0.0	10.200.0.1	10.207.255.254	10.207.255.255
10.208.0.0	10.208.0.1	10.215.255.254	10.215.255.255
10.216.0.0	10.216.0.1	10.223.255.254	10.223.255.255
10.224.0.0	10.224.0.1	10.231.255.254	10.231.255.255
10.232.0.0	10.232.0.1	10.239.255.254	10.239.255.255
10.240.0.0	10.240.0.1	10.247.255.254	10.247.255.255
10.248.0.0	10.248.0.1	10.255.255.254	10.255.255.255

List of 64 networks

for the 10.0.0.0 network with the subnet mask 255.252.0.0 (or /14)

Network	Hosts		Broadcast Address
	from	to	
10.0.0.0	10.0.0.1	10.3.255.254	10.3.255.255
10.4.0.0	10.4.0.1	10.7.255.254	10.7.255.255
10.8.0.0	10.8.0.1	10.11.255.254	10.11.255.255
10.12.0.0	10.12.0.1	10.15.255.254	10.15.255.255
10.16.0.0	10.16.0.1	10.19.255.254	10.19.255.255
10.20.0.0	10.20.0.1	10.23.255.254	10.23.255.255
10.24.0.0	10.24.0.1	10.27.255.254	10.27.255.255

10.28.0.0	10.28.0.1	10.31.255.254	10.31.255.255
10.32.0.0	10.32.0.1	10.35.255.254	10.35.255.255
10.36.0.0	10.36.0.1	10.39.255.254	10.39.255.255
10.40.0.0	10.40.0.1	10.43.255.254	10.43.255.255
10.44.0.0	10.44.0.1	10.47.255.254	10.47.255.255
10.48.0.0	10.48.0.1	10.51.255.254	10.51.255.255
10.52.0.0	10.52.0.1	10.55.255.254	10.55.255.255
10.56.0.0	10.56.0.1	10.59.255.254	10.59.255.255
10.60.0.0	10.60.0.1	10.63.255.254	10.63.255.255
10.64.0.0	10.64.0.1	10.67.255.254	10.67.255.255
10.68.0.0	10.68.0.1	10.71.255.254	10.71.255.255
10.72.0.0	10.72.0.1	10.75.255.254	10.75.255.255
10.76.0.0	10.76.0.1	10.79.255.254	10.79.255.255
10.80.0.0	10.80.0.1	10.83.255.254	10.83.255.255
10.84.0.0	10.84.0.1	10.87.255.254	10.87.255.255
10.88.0.0	10.88.0.1	10.91.255.254	10.91.255.255
10.92.0.0	10.92.0.1	10.95.255.254	10.95.255.255
10.96.0.0	10.96.0.1	10.99.255.254	10.99.255.255
10.100.0.0	10.100.0.1	10.103.255.254	10.103.255.255

10.104.0.0	10.104.0.1	10.107.255.254	10.107.255.255
10.108.0.0	10.108.0.1	10.111.255.254	10.111.255.255
10.112.0.0	10.112.0.1	10.115.255.254	10.115.255.255
10.116.0.0	10.116.0.1	10.119.255.254	10.119.255.255
10.120.0.0	10.120.0.1	10.123.255.254	10.123.255.255
10.124.0.0	10.124.0.1	10.127.255.254	10.127.255.255
10.128.0.0	10.128.0.1	10.131.255.254	10.131.255.255
10.132.0.0	10.132.0.1	10.135.255.254	10.135.255.255
10.136.0.0	10.136.0.1	10.139.255.254	10.139.255.255
10.140.0.0	10.140.0.1	10.143.255.254	10.143.255.255
10.144.0.0	10.144.0.1	10.147.255.254	10.147.255.255
10.148.0.0	10.148.0.1	10.151.255.254	10.151.255.255
10.152.0.0	10.152.0.1	10.155.255.254	10.155.255.255
10.156.0.0	10.156.0.1	10.159.255.254	10.159.255.255
10.160.0.0	10.160.0.1	10.163.255.254	10.163.255.255
10.164.0.0	10.164.0.1	10.167.255.254	10.167.255.255
10.168.0.0	10.168.0.1	10.171.255.254	10.171.255.255
10.172.0.0	10.172.0.1	10.175.255.254	10.175.255.255
10.176.0.0	10.176.0.1	10.179.255.254	10.179.255.255

10.180.0.0	10.180.0.1	10.183.255.254	10.183.255.255
10.184.0.0	10.184.0.1	10.187.255.254	10.187.255.255
10.188.0.0	10.188.0.1	10.191.255.254	10.191.255.255
10.192.0.0	10.192.0.1	10.195.255.254	10.195.255.255
10.196.0.0	10.196.0.1	10.199.255.254	10.199.255.255
10.200.0.0	10.200.0.1	10.203.255.254	10.203.255.255
10.204.0.0	10.204.0.1	10.207.255.254	10.207.255.255
10.208.0.0	10.208.0.1	10.211.255.254	10.211.255.255
10.212.0.0	10.212.0.1	10.215.255.254	10.215.255.255
10.216.0.0	10.216.0.1	10.219.255.254	10.219.255.255
10.220.0.0	10.220.0.1	10.223.255.254	10.223.255.255
10.224.0.0	10.224.0.1	10.227.255.254	10.227.255.255
10.228.0.0	10.228.0.1	10.231.255.254	10.231.255.255
10.232.0.0	10.232.0.1	10.235.255.254	10.235.255.255
10.236.0.0	10.236.0.1	10.239.255.254	10.239.255.255
10.240.0.0	10.240.0.1	10.243.255.254	10.243.255.255
10.244.0.0	10.244.0.1	10.247.255.254	10.247.255.255
10.248.0.0	10.248.0.1	10.251.255.254	10.251.255.255
10.252.0.0	10.252.0.1	10.255.255.254	10.255.255.255

bibliography

Brotzman & Ranch, *Securing Linux Step-by-Step version1.0*, The SANS Institute, 2000

Champan & Zwicky: *Building Internet Firewalls,* O'Reilly & Associates, Inc. 1995

Cheswick & Bellovin: *Firewalls and Internet Security*, Addison-Wesley, 1994

Mancill, *Linux Routers A Primer for Network Administrators*, Prentice Hall, 2001

Roesch, *Intrusion Detection – Snort Style*, SANS Institute, 3.3PM/3.4

Stevens: *TCP/IP Illustrated, Volume 1 The Protocols*, Addison-Wesley, 1994

Yuan & Strayer: *Virtual Private Networks, Technologies and Solutions*, Addison-Wesley, 2001

I also read several of the Honors GIAC GCFW Practicals during preparation of this work:

Sam Campbell: http://www.sans.org/y2k/practical/Sam_Campbell_GCFW.zip

Edward Luck http://www.sans.org/y2k/practical/Edward_Luck_GCFW.zip

Michael Vars http://www.sans.org/y2k/practical/Michael_Vars_GCFW.zip

Brian Kelley http://www.sans.org/y2k/practical/Brian_Kelley_GCFW.zip

Daniel Martin http://www.sans.org/y2k/practical/Daniel_Martin_GCFW.zip

Angela Orebaugh http://www.sans.org/y2k/practical/Angela_Orebaugh_GCFW.zip

Jeff Stelzner http://www.sans.org/y2k/practical/Jeff_Stelzner_GCFW.zip

Lenny Zeltser http://www.sans.org/y2k/practical/Lenny_Zeltser_GCFW.zip

Alan Moe http://www.sans.org/y2k/practical/Alan_Moe_GCFW.zip