Global Information Assurance Certification Paper

Planning, Implementing, and Maintaining a Secure Network Perimeter for GIAC Enterprises

by

Jim Moore

GIAC Certified Firewall Analyst (GCFW) Practical Assignment Version 4.1

DRAFT COPY FOR REVIEW ONLY

Submitted April 6, 2005

Contents

List of Figures

List of Tables

Part I

Wireless Networking: Security Implications for GIAC Enterprises Network

Serious security issues have plagued 802.11 wireless LANs since their introduction in the late 1990s. Numerous security analyses have pointed out the inherently insecure nature of LAN communications through the air and the woefully inadequate protections provided by Wired Equivalent Privacy (WEP), the protocol defined to protect WLAN traffic from eavesdropping, spoofing, and connection hijacking in the original 802.11 standard. Since the issue of the original standard, the IEEE and industry groups have made several attempts to improve the security of WLAN communications, including the 802.11X standard, which defined an improved method for authenticating wireless nodes, WPA (Wi-Fi Protected Access), an interim standard improving on WEP produced by an industry group, which can be applied to existing hardware via firmware upgrades, and 802.11i, a new standard recently approved by the IEEE, which specifies a more robust security architecture for WLAN communications. Under the original standard, WLANs were considered too insecure for a broad number of security-sensitive applications. If one or more wireless devices had to be attached to a private network, it was frequently recommended that that they be isolated from the wired network by a firewall and forced to use a VPN to make a connection to the wired network. With the new standard in place, to what extent and in what circumstances do the security measures recommended for 802.11 networks still make sense? How could WLANs be implemented securely in the GIAC Enterprises network using the features added by the 802.11i standard?

To answer these questions, it would be useful to review the security characteristics of the original 802.11 standard and each of its improvements to establish a basis of comparison for 802.11i security. Then we can consider the remaining security weaknesses of 802.11i and the practicality of exploiting those weaknesses. Finally, we can consider other practical obstacles to implementing full-strength 802.11i security that may qualify a decision about whether other security measures are required.

1 Security Problems with 802.11

1.1 Inherent Risks of Wireless LAN Communications

Wireless communications introduces a new set of complications into the process of defining and defending the boundaries of trust in a networked environment. Wireless devices conforming to the 802.11 standard broadcast signals; any other conforming device within range can pick them up.[1] Range varies depending on the physical medium used. Infrared signals can carry up to about 20 meters,[2] 802.11b signals

can carry up to about 500 meters in ideal conditions.[1] In the United States, 802.11b is in wide use. In urban settings it is trivially easy for anonymous individuals to intercept wireless LAN communications from nearby offices, buildings or streets.

In the most common wireless LAN configuration, known as an Extended Service Set, one or more wireless access points broadcast ("beacon frame") or at least readily transmit to an inquiring device the "Service Set Identifier" (SSID) needed to connect to the wireless LAN.[2] In the original 802.11 standard, access points are configured to accept by default "open system authentication," which basically means anyone can connect to the device and anybody can listen in.[3] Most wireless access points have been set up with the factory default configuration, as the annual Worldwide Wardriving contest maps have abundantly demonstrated.[4] No sane administrator of a private network would open up his organization's network to the public, but ordinary users in many organizations, possibly unaware of the security risks, have set up "rogue" access points reachable by devices outside the organization.[5]

1.2 Inadequacies of WEP

A sane network administrator would want to restrict access to his organization's wireless LAN to authorized devices and prevent unauthorized individuals from being able to intercept messages transmitted from participating wireless devices. The original 802.11 standard provided for authentication and privacy through "Shared Key" authentication and "Wired Equivalent Privacy" (WEP). The shared key is a code configured on each authorized wireless device. When a device wants to "associate" with an access point, the access point uses the code to generate a challenge and sends it to the device, which uses the same code to return the challenge in a WEP-encrypted packet. If the access point can decrypt the packet and the decrypted challenge matches the challenge originally sent, the device is allowed to associate. WEP is then used to encrypt all subsequent data traffic between the access point and the device. WEP employs a 40-bit shared secret key (optionally increased to 104-bit by many vendors' implementations)[1] and the RC4 encryption algorithm to encrypt and decrypt messages between wireless devices. The secret key is distributed to each participating station by a mechanism unspecified by the standard.[2]

Unfortunately, the security provisions in the original 802.11 standard fail to provide adequate authentication or privacy. Even if an administrator were to implement all the recommended provisions for maximizing the security of his organization's wireless LAN(s) available under the 802.11 standard,[3][1] the inherent weaknesses of WEP would expose it to numerous vulnerabilities. First, there is no provision in the standard for a device to authenticate an access point. An attacker with a modicum of knowledge of the wireless LAN could set up a rogue access point with security features turned off and attract victim devices to associate with it under the false pretense that traffic is secured. Second, MAC address spoofing is trivial under 802.11 and most control and management frames are sent in the clear, making it easy for an attacker to masquerade as an access point and interfere with the associations between devices and legitimate access points. Third, since access points send the authentication challenges in the clear, an attacker who intercepts them can use offline brute-force methods to

crack the shared key.[5] Fourth, the WEP IV (initialization vector) field, at 24 bits, is too small. On a busy network, a constantly-changing IV will eventually roll over, allowing a persistent attacker monitoring the network to recover the RC4 key stream. From there, he can use offline methods to crack the WEP key. In addition, there are no provisions in the standard for changing the IV. A vendor's implementation could have all stations changing the IV in a similar manner, leading to identical key streams occurring nearly simultaneously. Even worse, it is possible that a particular station may use a constant IV. The IV is part of the RC4 encryption key. Knowledge of part of the encryption key, combined with a known weakness in the RC4 key scheduling algorithm, makes it possible for an attacker to analyze WEP-encrypted traffic and recover the encryption key.[1] Vendors have reduced the risk associated with this problem by implementing key management schemes that reduce the lifetime of the WEP key to around 5-15 minutes. A frequently-changing WEP key greatly reduces the likelihood that identical key streams will be used over the lifetime of a WEP-encrypted WLAN.[5] Unfortunately, weaknesses in the way IVs are generated do not rule out the possibility. Even if an attacker is unable to crack a WEP key on time to mount an active attack using WEP, he can still decrypt all the traffic using that key. Finally, WEP makes no provision for cryptographic integrity-checking. The integrity of a packet is determined under 802.11 by computing a CRC value and comparing that to a CRC value sent with the packet. There are published methods for altering packets that will result in the identical CRC value. In addition, it is possible to use partial knowledge of the contents of an encrypted packet to generate packets with altered IP, TCP or UDP headers using bit-flipping that will be accepted by an access point and then forwarded to a system under the attacker's control! Some of these attacks require modifications to the firmware of wireless devices, but competent, determined attackers could accomplish that[6].

There are several ways to mitigate these weaknesses within the framework of the original 802.11 standard, but many students of the issue recommend employing additional security measures. These include isolating the wireless network from the rest of an organization's network by address segmentation, firewalling the wired network off from the wireless segments, and requiring all wireless devices to use a VPN based on IPSec or SSL/TLS to communicate with other devices.[3][5][1] The problem with these recommendations is that they are not practical in some applications and do not eliminate all problems. For example, they do not eliminate the problem of communications between nearby wireless devices. If the wireless devices need to communicate securely with each other, either each will have to be forced to route its packets through a VPN concentrator, which could have a serious impact on network performance, or each device will have to be configured to establish a VPN directly with its wireless neighbors. Not all wireless devices are capable of this, and for those that are configuring VPNs is not a trivial exercise.

1.3 Inadequacies of 802.1X

The IEEE recognized the security inadequacies of 802.11 early, and started working groups to address them. The first new standard, 802.1X, published in 2001, addressed

the problem of authentication.[7] It established a mechanism for authenticating wireless devices and their users to another device. The authenticating device could be a wireless access point, hub, router, or even another wireless device. It adapted the Extensible Authentication Protocol (EAP) standard published by the IETF[8] to transport over ISO layer 2 (EAP over LAN). As its name suggests, EAP provides a means for a device communicating with another device over a point-to-point link to authenticate using any conforming method. The 802.1X standard specifies EAP mechanisms employing an "authentication server" that may or may not be part of the authenticating device. The authenticating device ("authenticator") and its peer ("supplicant") exchange authentication messages using EAP. These messages include initialization of the authentication exchange, negotiation of an authentication mechanism, exchange of authentication tokens and authorization information, and deauthentication. The authenticator also acts as an intermediary between the supplicant and the authentication server translating packets from EAP to whatever protocol it uses to communicate with the authentication server and vice versa. Until authentication completes successfully, the authenticator will not accept packets from the supplicant except those needed to complete the authentication exchange. The standard recognizes that in a shared media environment, such as a wireless LAN, EAP mechanisms must be chosen that enable the exchange of authentication information between the supplicant and authenticator and between the authenticator and authentication server securely, typically via encryption. Authentication protocols meeting this requirement include Kerberos,[9] Diameter,[10] and RADIUS using IPSec.[11][12]

The authentication exchange described above is strictly one-way. The "supplicant" does not authenticate the "authenticator." The original standard required that in an IBSS, in which each wireless device communicates directly with every other wireless device in an "ad hoc" network, each wireless device would act both as a "supplicant" and an "authenticator." In this setting, all devices would mutually authenticate. In an ESS, however, only the access point acts as an "authenticator." The "supplicant" device has no way to authenticate access points. Furthermore, at least some of the management packets exchanged between the "supplicant" and the "authenticator" travel in the clear and unauthenticated. These flaws in the design of 802.1X led to the elaboration of "man-in-the-middle" and session-hijacking attacks against 802.1X in an ESS context.[13] Subsequent responses generally conceded the authors' claims against the 802.1X standard, unless enhanced by per-packet encryption using dynically-negotiated keys and a higher-level protocol that performs mutual authentication. Responses written by members of the wireless industry tended to claim that their product's enhancements to the 802.1X standard would defeat the attacks.[14][15]

1.4 Inadequacies of WPA

In fact, vendors had been quite busy implementing security enhancements to their 802.11 devices, mostly without standardization. The industry evntually issued an interim standard of its own, Wi-Fi Protected Access, which defined a protocol for dynamic key management and data packet authentication and encryption called Temporal Key Integrity Protocol (TKIP). The purpose of the standard was to mitigate the

weaknesses of the 802.11 standard without requiring the replacement of deployed hardware. TKIP added an encrypted Message Integrity Code (MIC) to messages to reduce the effectiveness of attacks that relied on MAC address spoofing or packet modification. It also increased the size of the Initialization Vector (IV), added a key mixing function, implemented replay attack prevention measures, defined a rekeying mechanism, and described a series of counter-measures designed to reduce the impact of attempted exploits against the protocol to Denial of Service. TKIP is included in the 802.11i standard as an interim security protocol for networks making the transition from hardware that does not conform to the requirements of 802.11i to hardware that does. It only enhances the security of ESS networks and still relies on the RC4 encryption algorithm specified for WEP.[16][1] The 802.11i standard states that TKIP is a trade-off between security and compatibility with older hardware. It is still vulnerable to active attacks. As a countermeasure, the protocol specifies that reception of a packet with an invalid MIC should be treated as an active attack. After receiving two such packets within 60 seconds, the receiving device will deauthenticate itself and, in the case of an access point or in an IBSS, all stations it has currently authenticated and refuse to accept or send any packets except 802.1X authentication messages for 60 seconds.[18]

This trade-off makes it possible to launch an extended Denial of Service attack against a TKIP-protected WLAN simply by sending 2 invalid TKIP messages to the access point every 60 seconds.[19] Other researchers have verified TKIP's susceptibility to DoS attacks based on spoofed deauthentication or disassociation packets.[20] By itself, thse attacks pose no threat to the privacy of data carried over the WLAN, but in some situations the DoS itself could lead to serious problems for the victim organization. In addition, WLANs making use of WPA with pre-shared keys rather than 802.1X authentication are vulnerable to off-line dictionary attacks against the pre-shared key. Any system capable of intercepting traffic between the access point and any other station in the ESS could perform this off-line attack, and once in possession of the key, compromise the security of the entire WLAN. The vulnerability can be mitigated by using a random pre-shared key of 20 characters or more in length.[21] The use of DoS could be the first step in such an attack, since it would provoke a large number of authentication attempts which could be collected and subjected to analysis. An attack tool based on this vulnerability was subsequently published on the internet with an accompanying technical discussion.[22] Overall, WPA, when implemented with properly-configured 802.1X authentication and without legacy support for WEP-only devices, still represents a huge improvement over any combination of prior non-proprietary security features available for wireless LANs.

[1] The Wi-Fi Alliance's documentation on WPA can be described as marketing literature. It does not mention the limitations of TKIP detailed in the 802.11i standard. At least one document makes the undocumented claim, "Cryptographers have reviewed Wi-Fi Protected Access and have verified that it meets its claims to close all known WEP vulnerabilities and provides an effective deterrent against known attacks."[17] The WPA specification is still available on their Website only for a fee of $25.00, while the full 802.11i standard is now freely available to the public on the IEEE website.

2 Assessment of 802.11i

2.1 Overview of the Security Architecture

The complete IEEE 802.11i standard describes an overhauled 802.11 security architecture called a "Robust Security Network." This architecture can be implemented in an IBSS or ESS. It consists of a set of peer-to-peer security associations (analogous to the security associations established in IPSec) negotiated between wireless devices based on pre-configured security requirements and collectively labelled a "Robust Security Network Association" (RSNA). In an IBSS these security associations are established between any pair of wireless devices wishing to communicate with each other. In an ESS, they are established between the access point and any wireless devices wishing to participate in the wireless network. Each RSNA includes up to 4 specific security associations consisting of keys and policies. The Pairwise Master Key Security Association (PMKSA) includes the long-term key to be used for generation of the transient keys. The Pairwise Transient Key Security Association (PTKSA) includes the key to be used to encrypt unicast packets exchanged between the two peers. The Group Transient Key Security Association (GTKSA) includes the key to be used to encrypt multicast or broadcast packets destined for the members of the wireless network. The STAKey Security Assoction (STAKeySA) includes the key to be used to encrypt traffic sent from one non-AP wireless device to another non-AP device participating in the same ESS.

RSNA-capable devices identify one another by an additional RSN element in beacon frames and (re)association messages. This element includes a list of cipher suites the device is willing to use to communicate securely with other devices. In addition, in an ESS access points can insist on the use of a specific cipher suite during association. Before establishing any of these security associations, wireless peers must authenticate each other using either pre-shared keys or an 802.11X authentication mechanism that implements mutual authentication such as EAP-TLS. Upon authentication, the peers establish the PMKSA, using either the pre-shared key or keying information exchanged during 802.11X authentication as the Pairwise Master Key. The peers then engage in a "4-way handshake" that establishes the PTKSA and GTKSA. If the non-AP peer wishes to establish direct communications with another non-AP device on the wireless LAN, it then initiates a "STAKey handshake" with the access point. Whenever a wireless device (re)(dis)associates, (de)authenticates, or simply moves out of range of an access point with which it has an active security association, any existing PTKSA and GTKSA are deleted. Likewise, the access point deletes the PTKSA for any device that has entered one of these states. The PMKSA, on the other hand, can remain in force indefinitely. PMKSAs can be cached by a device as they are established between different peers. Implementations may use whatever means are available to preserve the cached PMKSAs across system reboots or other interruptions of communication with the wireless network. The device can be configured to specify a pre-defined maximum lifetime for its PMKSAs. Once the lifetime expires, the device must re-authenticate if it is using 802.1X authentication, or the user may be prompted to re-enter a passphrase to activate a pre-shared key that will supply the

key material for the PMKSA. Otherwise, the PMKSA may last as long as the Pairwise Master Key used by the peers does not change.

The 802.11i standard specifies the Counter with CBC-MAC cipher suite[23] using the AES-128[24] encryption algorithm for data and key management, packet encryption and authentication. The only other option available in a Robust Security Network is TKIP, and this must only be used in contexts in which compatibility with non-RSNA-enabled devices is required. The Counter with CBC-MAC cipher suite (CCM) combines a block cipher with message authentication. In 802.11i it takes 4 inputs, an encryption key, a nonce that must be unique across all encryption operations using the same encryption key, a plaintext message block, and part of the MAC header including the source and destination MAC addresses.[18] It computes a message authentication code using over the combined MAC header and message text. Then it encrypts the message authentication code and message text by generating key stream blocks using AES-128 over the nonce and encryption key and XORing blocks of the message text with the key stream. The encrypted message authentication code is appended to the encrypted message text to form the cipher text.[23]

2.2 Potential Weaknesses

As of this writing, the 802.11i standard is not even a year old, and because implementing conformant RSNAs requires new hardware, has not been widely deployed. Still, there is data on potential security weaknesses of the standard, some of it supplied by the standard itself. In particular, the authors include a list of "Assumptions and Constraints" for aspects of a WLAN not part of the standard that must be met in order for the RSNA to possess the security characteristics outlined in the standard. This list is intended to help implementors prevent poor design of features outside the scope of the 802.11i standard from compromising the security of their products and to help consumers assess the real-world security consequences of deploying a particular 802.11i configuration. Notable items include the need to use an EAP method that ensures strong mutual authentication for 802.1X authentication, the need to provide a secure channel between access points and centralized authentication servers to protect keys and authentication tokens passing along the wired LAN, and the limitations of using pre-shared keys for the Pairwise Master Key. Regarding the last item, a malicious insider (someone who has control of another device furnished with the same pre-shared key) can determine the Pairwise Transient Key for any two other stations by examining the first two exchanges of the 4-way handshake. From there, eavesdropping or a man-in-the-middle attack is possible.[18] The exploitation of other potential weaknesses depends on the specifics of the 802.11i implementation and the configuration choices of the 802.11 network administrators.

It is heartening to note that the core privacy and authentication protocol chosen for conforming 802.11i devices, CCMP, has been proven to have robust security properties. An attacker with access to a stream of packets processed by CCMP is highly unlikely to be able to collect two identical ciphertexts (which can be used to attempt to crack the encryption key) or to learn enough about the message authentication code to forge a valid packet,[25] provided the same encryption key is used for no more than

2^{61} encryption operations.[23]. It is possible for an attacker to mount a precomputation attack against CCMP with 128-bit encryption keys when the same nonce value is likely to occur across encryption sessions using different keys and the first 16 bytes of the plaintext message are known. The attacker generates a table of all keys and the first of the resulting key stream blocks generated using that nonce. He then captures packets using that nonce and compares the first encrypted block to those in the table. After about 2^{64}messages the attacker should be able to find a matching block and identifiy the encryption key. This attack can be defeated by using a larger key or by combining additional data with a sequentially increasing nonce value to form the nonce.[23] The 802.11i specification employs 128-bit keys but also requires that the MAC address of the sending station be included in the nonce value. As a result, the series of all possible nonce values is unique to each communicating device. The attacker would have to generate a separate table for each device in the target packet stream and would have to observe 2^{64}messages from at least one of the devices before discovering the encryption key. This additional restriction makes precomputation attacks against CCMP highly unlikely to succeed.[26]

The wedding of 802.11i privacy using CCMP with 802.1X authentication closes nearly all the known holes in the 802.11 architecture. The major remaining problem is Denial of Service. DoS attacks of several types can still succeed against a full-blown 802.11i network. In addition to the TKIP DoS attack mentioned previously1.4, 802.11i networks are subject to DoS attacks using forged deauthentication or disassociation frames, sending any of several forged EAPOL messages to the 802.1X supplicant or authenticator, forging a packet with an incorrect RSN Information Element in message 3 of the 4-way handshake, or sending out numerous forged 4-way handshake message 1 packets to a supplicant.He and Mitchell [26] The impact of these attacks is mitigated somewhat by the requirement that the attacker operate a device on the LAN. Since the device is in the vicinity of the rest of the network, in most cases it could be located fairly quickly. Still, many networks cannot function adequately with even brief interruptions of service to devices on the LAN.

2.3 Implementation Issues

It appears to this author at least that 802.11i WLANs can provide adequate security to an organization that must protect data from unauthorized access, provided certain conditions are met. First, the WLAN must implement the RSNA architecture using CCMP. Legacy devices only capable of using WEP should not be allowed to use the WLAN. This eliminates not only the vulnerabilites of WEP but also those of WPA and TKIP. Second, the WLAN must be configured as an ESS and employ 802.1X authentication using centralized authentication servers running RADIUS, Diameter, or Kerberos. Third, the WLAN must be restricted to devices participating in a shared network of trust, such as a Windows Domain or a PKI infrastructure. If all these conditions are met, there is no need in most cases to firewall off the WLAN from the rest of the organization's network or force WLAN devices to tunnel traffic through a VPN to the wired network.

On the other hand, there are numerous cases in which not all of these conditions can be met. The most obvious case is that of the road warrior who uses a wireless device to access the organization's network from the outside. In many cases the road warrior's device will associate with a 3rd-party access point. In these cases, no guarantees about the security of the device's traffic in the air or across intervening wired networks can be made. In this case it makes sense to firewall off the road warrior's device and force it to establish a VPN tunnel. Some organizations have divisions with different security requirements or have policies which prevent divisions from sharing certain types of information with each other. In these cases, if the organization wishes to deploy WLANs shared by members of different divisions, the network will have to be configured to restrict unauthorized devices higher up the protocol stack, including the use of internal VPNs and intra-divisional network firewalls.

Part II

GIAC Enterprises Perimeter Security Architecture

3 Defining the Information Technology Capabilities Required by GIAC Enterprises in order to Accomplish its Business Objectives

3.1 Historical Assumptions

The description of GIAC Enterprises given in the guidelines for Part 1 of this Practical is too general for determining the precise relationship the company has or intends to have with its customers, suppliers, partners, and employees. In order to develop a realistic network design, it is necessary to make further assumptions about the nature of GIAC Enterprises' business, as was done by previous analysts.[27] These assumptions put flesh to the business needs of the company. The network design will be shaped by the business needs that arise from these assumptions, and it must be judged by how well it satisfies those needs.

In the real world, the analyst rarely has such freedom to shape the circumstances for which he is designing a solution. There is little doubt that if IT Security professionals really had the power to dictate to organizations what their genuine business needs were, there would be far fewer security breaches – and far fewer successful organizations! Likewise, I must not make assumptions about the character of GIAC Enterprises that end up serving simply to make it easier for me to design a satisfying network security infrastructure. With this caveat, let's take a closer look at the company.

GIAC Enterprises is a small publishing company that specializes in the global distri-

bution of fortune cookie sayings. While it is based in the United States and maintains there its corporate headquarters and publishing offices in Brooklyn, NY, it partners with authors and printers around the world to provide buyers with fortune cookie sayings that are in their customers' native language and appropriate to their customers' culture.

The company was founded in the 1960s, grew rapidly in the United States and began expanding overseas, largely by cooperative agreements with foreign businesses, in the 1970s. Management tended to be cautious about adopting new technologies. This policy appeared to serve them well until the 1990s.

As the decade wore on, it became clear that companies more quickly adopting internet technologies were eating their lunch. Fueled by the efficiencies of web-based ordering, their competitors were dropping prices and luring customers away. Management reluctantly decided to establish an English-language web site for product sales in 1998. They hired a web development company to develop and maintain a website. The web development company customized a business-to-business shopping-cart solution for them. They contracted with their ISP to host the website.

That decision led to increased sales. Management was sufficiently impressed to roll out web-based sales worldwide over the following year. By 2001 the company had a well-established international on-line sales presence. Meanwhile, sales costs dropped as many older customers converted to web-based ordering. While management was happy with these results, they were also feeling pressure to increase the breadth of their reliance on internet technologies.

They were not able to develop as broad a base of authors in emerging markets as they desired because of continuing problems with communications. They had also begun to lose authors to companies that were using the web to maintain relationships with authors. Management decided to survey all their existing suppliers, partners, and customers to find out what services they would like the company to provide and how they would like to receive them. The results indicated that business relationships should improve if the company supplemented its personal contacts with its suppliers, partners, and customers with web-based access to more detailed product information, coming projects, and collaboration tools. This year, the company hired a chief technology officer and assigned him the task of bringing in these capabilities, overseeing training of staff, suppliers, partners, and customers in using the new features, and providing and maintaining the infrastructure necessary to make it happen. He in turn hired me to help design and implement a network security architecture to support the new operations.

The new CTO included the following requirements in my assignment. First, the design must assume that as many functions as possible of the company's web presence be located in-house. The CTO was determined to eliminate dependence on third-parties for the security of the company's assets. Second, the design should use open-source tools, unless commercially-available, closed-source tools were significantly superior or open-source tools are not available for a particular function. This decision was based primarily on budgetary constraints for software acquisition and training (NOT because open-source software is inherently more likely to be free of security weaknesses! [28, 29]). As it turns out, the CTO's former company had used many

open-source security tools, and he had managed to bring a few of his key staff along with him to GIAC Enterprises. He wanted to exploit their competencies as quickly and cheaply as possible.

Third, the design must allow for the use of the existing ecommerce solution, which includes the following components: A web/application server cluster provides access for customers to product information via HTTP and secure online ordering and order-status information via HTTPS secured with SSL v.3 using an RSA server certificate signed by Verisign's Certificate Authority and a negotiated encryption algorithm of minimum 128-bit key length. Also, an Oracle database backend cluster located in-house provides access to customer account information to the web/application servers and detailed reporting information to internal customer support and accounting personnel using an internal application server via an encrypted channel on TCP ports 1521-1526 using TLS v. 1.0 with 2-way authentication using RSA client and server certificates signed by Verisign's Certificate Authority and encryption using the triple DES cipher algorithm with a 168-bit key and SHA-1 MAC. In addition, the Oracle database server cluster must be able to make connections to Acme Payment Systems remote payment processor over the internet (IP address 1.100.100.1) on port 4999 using TLS v. 1.0 with 2-way authentication using RSA client and server certificates signed by Verisign's Certificate Authority and encryption using the triple DES cipher algorithm with a 168-bit key and SHA-1 or MD5 MAC.

In addition to these baseline requirements, I had to make provision for additional access by suppliers, partners, and employees as follows:

3.2 Capabilities needed for suppliers of GIAC Enterprises.

GIAC Enterprises' suppliers can be broken into two broad categories: authors of fortune cookie sayings and vendors who provide products needed by the employees of GIAC Enterprises to conduct the various aspects of her business. Authors need access to specifications for new projects, legal and contractual information regarding copyright, ownership of submissions, review policy, plagiarism, pay scales, and other contranct terms, a repository in which they can place and retrieve works in process that are under editorial review, and channels for communication with editors and, in limited cases, with each other, about their work. The CTO already had an answer to my question about exactly where in the organization's network infrastructure authors would go to get this access; he had contracted with a development firm to build a web portal into the existing Oracle application server. This portal required authors to authenticate with a user name and password to gain access to most features of the site, and granted access to various features of the portal based on policies set for their identity by the editorial staff. The portal included a collaboration space, in which authors and editors could brainstorm together, and a restricted-access repository for submissions in progress and contracts. Communications with the portal from the internet were secured with 2-way authentication using RSA certificates signed by Verisign's Certificate Authority and a negotiated encryption algorithm of minimum key length of 128 bits. I had to come up with a secure method for editors and authors to exchange contracts and other sensitive information over the internet in the event the

web portal was not available.

Other suppliers required far less access to GIAC Enterprises. They needed to be able to communicate with GIAC Enterprises employees via email and wanted the employees to be able to get to their websites for product information, purchasing, and educational opportunities. Occasionally, they would be exchanging email that required encryption.

3.3 Capabilities needed for partners of GIAC Enterprises.

Most of GIAC Enterprises' partners were printers with long-term contracts for printing and ensuring delivery of fortune cookie sayings to customers. Many of these firms were based in other countries. Most of the rest of the the firm's partners were overseas as well. Most often, they provided translation services. Frequently, they were also asked to vet projects destined for an overseas customer for cultural appropriateness.

They needed to be able to exchange email with GIAC Enterprises' editorial staff, sometimes in encrypted form. They also needed access to the portal for specifications of ongoing or upcoming projects and to exchange work in progress. Their access to the portal was secured in the same way as that for authors.

3.4 Capabilities needed for GIAC Enterprises' employees located in Brooklyn.

The employees based at GIAC Enterprises' headquarters and publishing offices include their corporate officers, editorial staff, and several support departments: accounting, legal, human resources, purchasing, sales, and IT. In general, the employees require access to the web and must be able to send email to various parties on the internet. In many cases, they will need the ability to engage in secure transactions via SSL. Individuals or departments that require additional access will be specfied below.

Editorial staff need the ability to send encrypted email to authors as a backup to secure access to the company's web portal. There exists the danger of proprietary information leaking out of the company in undetectable form via secure email. There does not seem to be a realistic method of preventing this from happening, and even if there were a way to prevent employees from sending encrypted email, there are too many other relatively easy ways for them to sneak proprietary information out the door. For that reason, this kind of access will be allowed and the risk mitigated by issuing clear policies on proper use and training editorial staff on how to implement those policies in their communications with authors.

Corporate and accounting staff need access to a variety of reports and financial data stored in the Oracle database originally used as part of the company's ecommerce solution. It was decided, however, to consolidate a number of internal databases storing financial data onto the same server. The company purchased a second application server to be used in-house only. This application server would limit direct access to the database, allow for even more granular access control, and isolate sensitive data from the publicly-available application server used by partners,

suppliers, and customers. Of course, it also greatly simplified access to the financial data.

IT staff need FTP access to various sites on the internet in order to download needed software packages and updates. While it is entirely possible to do this via HTTP, FTP is faster, and many of these packages will be quite large. The CTO has already indicated that only IT staff should be allowed to download software packages and updates, including Java programs. If employees need access to software available on the web, it IT's responsibility to fetch it, check it for problems (licensing issues, viruses and worms, security vulnerabilities) and make it available to employees on the company's intranet. IT staff also require access to the newsgroup server at the company's ISP in order to keep up with technical issues discussed in some newsgroups.

3.5 Capabilities needed for GIAC Enterprises' employees located at remote sites.

For the most part, the employees of GIAC Enterprises do not need access to data stored inside the company's network except when they are at work. The exceptions to this rule include corporate officers, some of the editorial staff, remote sales staff, and IT staff. The first three groups need access to email, the web portal, and the internal application server. The internal application server gives corporate officers quick access to confidential financials, performance data and strategy. Editors and salespeople use the internal application server to maintain confidential information on customers. The company's CRM solution, for example, is hosted on the internal application server. IT staff needs comprehensive access to the network to perform remote diagnostics and troubleshooting.

Access to email had already been available via Outlook for Web Access, provided by the company's Microsoft Exchange server. The company was now to migrate all its groupware functions to Oracle Collaboration Suite and use the web-interface provided by Oracle on the web/application server to give remote users access to email and other groupware functions. Remote access to the email site would be granted to users via HTTPS secured with SSL v.3 using 2-way authentication with RSA client and server certificates and a negotiated encryption algorithm of minimum 128-bit key length. The server certificate will be signed by Verisign's Certificate Authority and the client certificates will be signed by GIAC Enterprises' own certificate authority. Traveling employees will connect to the portal site in the same way. The internal application server cannot be accessed directly from the internet. For remote access to this system, remote users will have to establish an IPSec VPN connection to the company network from their remote site.

IT staff require by far the most far-ranging remote access to the company's network as a consequence of the CTO's decision not to staff IT on-site 24x7. After a study of the frequency of problems encountered during the use of the company's ecommerce solution and the number of such problems requiring on-site intervention by IT staff, and several discussions with IT managers who oversaw the use of portal software similar in design and purpose to what the company was about to put into production,

he concluded that the company could provide a sufficient level of service to its site's users without incurring the expense of keeping staff on-site 24 hours a day, 365 days a year.

Instead, he would have IT staff rotate on-call duty from home and have the company's security and network-monitoring tools forward alerts to the on-call staff. In order for this to work, on-call staff would have to have a secure, reliable connection to the company's network and sufficient access to its network infrastructure to perform maintenance and repair. This will take place via an IPSec VPN between each of the IT staff's homes and the company network.

As a backup in case the company's link to its ISP goes down or the VPN gateway or internet firewall fail, a Remote Access Server taking dialup connections will also be provided. The dialup lines are normally only needed on off-hours, and then only when IT staff are unable to reach target hosts via their VPN connection.

It should be noted in this connection that in no case does a remote employee need an end-to-end IPSec tunnel. Confidentiality inside the company's network can be achieved by other means, such as using HTTPS or SSH. The CTO instructed me that in the absence of a clear business need end-to-end tunnels will not be allowed. He preferred to limit the amount of unidentifiable traffic flowing in and out of the company's network.

4 Netork Security Architecture Providing Capabilities Needed by GIAC Enterprises

4.1 Overview

The network security architecture recommended for GIAC Enterprises is meant to accomplish the following objectives: 1.) Provide all the capabilities required by GIAC Enterprises; 2.) Prevent anyone from extending or exploiting those capabilities to achieve unauthorized access to any of GIAC Enterprises IT assets; 3.) Keep implementation and maintenance costs as low as is consistent with meeting objectives 1) and 2) quickly. The security architecture will achieve its objectives by employing a number of complementary security features, including a multi-layered perimeter, network intrusion detection and, for mission-critical, exposed, or otherwise sensitive hosts, host-based intrusion detection, hardening of exposed hosts, segmentation of the company's internal network, enterprise-wide anti-virus detection and removal, centralized system logging and near real-time response to possible security breaches. Costs will be contained by employing free or inexpensive open-source software and recycled hardware for many key components. Centralized administration tools will simplify maintenance and monitoring.

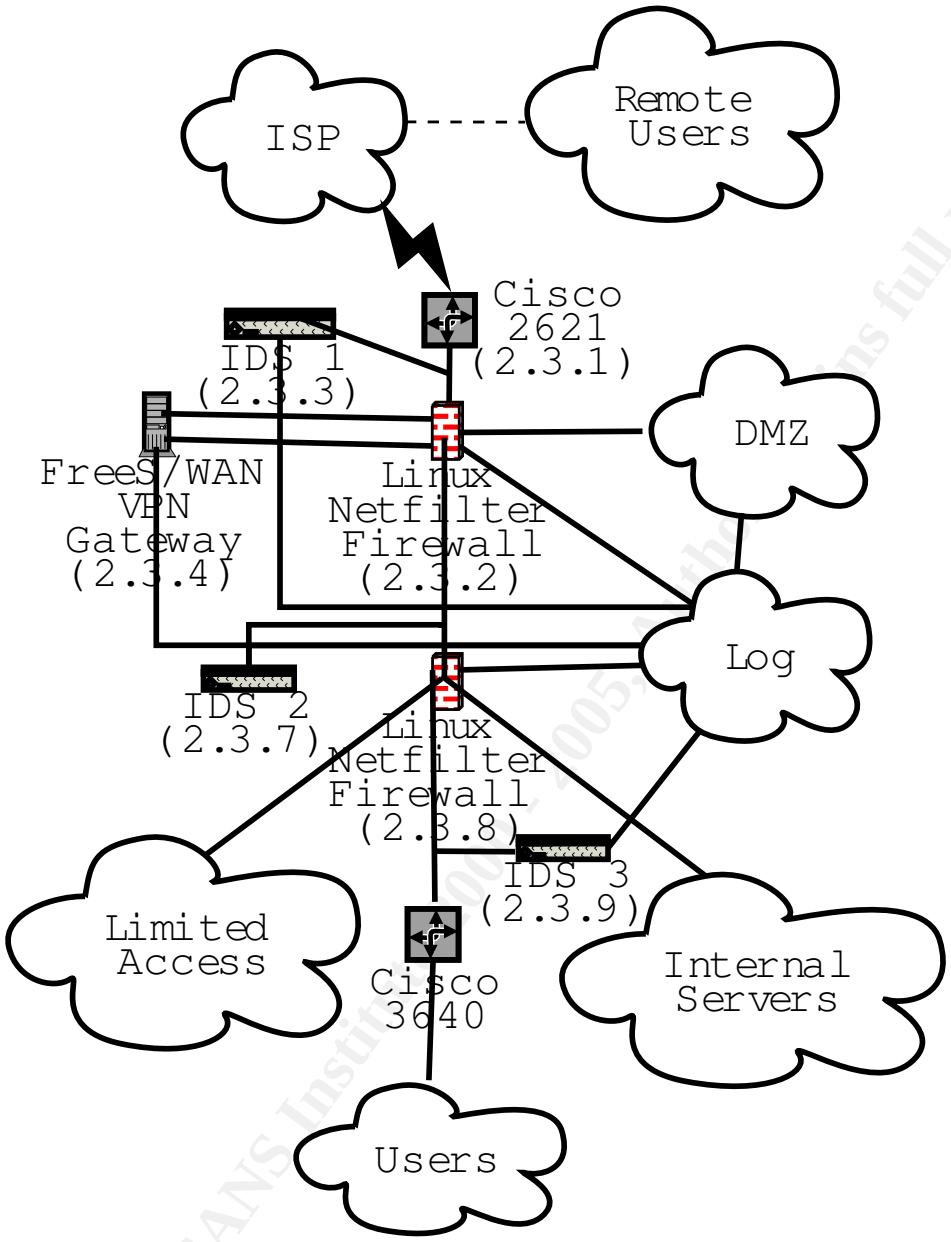


Figure 1: GIAC Enterprises Network Design Overview

4.2 IP Addressing Scheme

Heretofore, GIAC Enterprises' internal network was based on RFC 1918 addressing using class C subnets or smaller in the 192.168/16 address space. I decided to abandon this scheme entirely in favor of one based on the 10/8 address space. The obvious but less important reason for going with the 10/8 address space is that it increases the total number of available addresses by a factor of 256. The company is currently nowhere near large enough to use up the 65536 addresses available in the 192.168/16 space; converting to the 10/8 space simply makes it a complete non-issue.

The more important reason is interoperability. Many home networks and businesses use one or more class C subnets of the 192.168/16 space as their private addressing. By converting to the 10/8 space, the company more easily avoids collisions with private addresses in the home networks of VPN clients. Each VPN client will be assigned an IP address in the 10/8 space, and in no case will a VPN client be permitted to route packets from his/her home network or anywhere else into GIAC Enterprises' network, or vice versa. This is not so easily accomplished if the IP address assigned to a VPN client interface happened to be in a subnet already allocated to a home network. Routing issues between the client and the company network could be alleviated by subnetting the part of the home network assigned to the VPN client.

Another problem would remain, however. The firewall will include access rules for VPN subnets. The brevity and efficiency of these rules depends on being able to predict the subnetting for entire blocks of addresses assigned to certain classes of VPN clients. If exceptions have to be made to prevent address clashes with certain home networks, it will complicate the firewall ruleset. If, in the future the company has a need to set up a network-network VPN, perhaps with a partner, there is far more flexibility in the 10/8 space for finding a network number that will allow the two sides to talk to one another without asking either one to change their current addressing.

4.3 Components

4.3.1 Border Router

The purpose of the border router is to supervise the connection between GIAC Enterprises and the internet. Not only does it route IP traffic in and out of the company's network, but it also blocks IP traffic with invalid source addresses and prevents certain other kinds of potentially dangerous traffic, including certain kinds of ICMP and traceroutes, from entering the company's network. The border router is placed on the company's end of the T1 connecting it to its Internet Service Provider. All other devices connected to the internet in GIAC Enterprises forward their traffic through this router and all network traffic entering GIAC Enterprises network must pass through this router first. Since this is the first device incoming traffic encounters, it is the ideal device to block certain kinds of suspicious traffic which its limited access-control facilities can detect. In particular, it is able to block traffic with unrouteable[30] and otherwise reserved[31] source IP addresses. It is also ideal for blocking traceroutes, which might

Description	Network Block	Comment
Public Networks		
Public WAN Network	1.1.1.104/30	Assigned to GIAC Enterprises by ISP. Border Router <-> Internet Firewall
VPN Gateway Public Network	1.1.1.240/30	Assigned to GIAC Enterprises by ISP. Internet Firewall <-> VPN Gateway
Public Servers	1.1.2.240/28	Assigned to GIAC Enterprises by ISP.
Internal Network Device Link Networks		
Firewall-Firewall Network	10.1.1.0/30	Internet Firewall <-> Internal Firewall network
VPN Gateway Private Network	10.1.1.4/30	VPN <-> Internet Firewall RFC 1918 network
Internal Firewall-Users Network	10.1.1.8/30	Internal Firewall <-> Cisco 3640 Router
Private Networks		
Restricted-Access Network	10.1.3/24	Used to isolate sensitive traffic used mostly for infrastructure support from main subnets
DMZ	10.1.4/24	
Internal Server Network	10.1.5/24	
Logging Network	10.1.6/24	Mostly dedicated to logging to lighten load on main subnets.
Corporate User Network	10.10.2/24	
Editorial User Network	10.10.3/24	
Support User Network	10.10.128/17	Supernet for various support departments. Devices are logically subdivided by department into natural subnets. Used in firewall rules and for later segmentation.
IT Network	10.10.128/24	Used by network security devices to grant special access to IT personnel.
Sales Network	10.10.129/24	Used by network security devices to limit access by sales personnel.
Editorial VPN Client Network	10.253.2/24	System running VPN client software for a remote editorial user gets address in this subnet.
Sales VPN Client Network	10.253.3/24	System running VPN client software for a remote sales user gets address in this subnet.
IT Staff VPN Client Network	10.253.4/24	System running VPN client

otherwise be able to discover invaluable information about the structure of the network defenses protecting the publicly accessible mail gateway and web/application server. It can also reduce the impact of DOS attacks using rate limiting on certain types of otherwise permitted traffic. The border router used at GIAC Enterprises is a Cisco 2621 running Cisco IOS Release 12.2(15)T2, with the IP Plus feature set.

4.3.2 Internet Firewall

The purpose of the internet firewall is to limit the types of IP traffic allowed to pass between the GIAC Enterprises network and the internet. It sits just inside the border router and performs "stateful" filtering to prevent forwarding of unallowed traffic that the router's access-control mechanisms are too simple to detect. It also translates unrouteable (RFC 1918) addresses used in the GIAC Enterprises network into public IP addresses that can be passed along the internet. Since all traffic in and out of the network passes through this system, including unencrypted traffic from the VPN gateway, it is able to comprehensively limit the types of traffic flowing in and out. It is configured to block many types of attacks and evasive scanning techniques based on malformed IP packets, including fragmentation and unusual combinations of TCP flags. This system runs Redhat Linux 9.0 with netfilter v. 1.2.7a.

4.3.3 Network Intrusion Detection System 1

This unit sits just inside the border router. It captures and anaylyzes all inbound and outbound traffic for signs of suspicious activity, sends alerts when such activity is detected, blocks certain types of activity, and logs a record of the subsequent conversation between the source and destination hosts. It supplements the internet firewall by performing protocol analysis and blocking suspicious TCP activity. While it is possible to have the intrusion detection system modify the firewall rulebase dynamically in response to suspicious activity, this function is not implemented in the GIAC Enterprises network. This system runs Snort, v. 2.0.1 on Redhat Linux 9.0.

4.3.4 VPN Gateway

The VPN Gateway enables key GIAC Enterprises employees who need more comprehensive access to the company network than what is available through the web portal to establish a secure channel for communications across the internet. While this function could be performed by the internet firewall itself, it was offloaded to a separate device to ease the load on the firewall and to allow for the use of network address translation on packets coming from systems inside GIAC Enterprises network.

The VPN gateway has two interfaces connected to the internet firewall, an internal and a publicly-addressable interface. Outbound VPN packets pass through the firewall, NAT is performed on them, and they are forwarded to the VPN gateway's internal interface for handling. The VPN gateway wraps the packets in AH or ESP headers and passes them back to the firewall on the publicly-addressable interface. The firewall checks the information in the new packet headers and forwards them if they are acceptable. Inbound VPN packets are checked by the firewall first and passed to

the VPN gateway if allowed. The VPN gateway unwraps and unencrypts the packets and passes them to the firewall through its internal interface. The firewall performs any needed NAT on the packets, checks them to see if they are allowed, and if so, forwards them to a system in GIAC Enterprises' network.

The gateway will be configured to allow for redundant controls on access to the company network by remote clients. The VPN gateway itself will use the distinguished name of the client's X.509 certificate to determine to which areas of the company network the client is granted access . Each user will also be assigned a static IP address to be associated with a virtual interface on the client machine. The IP address will be pulled from a subnet to which all addresses for that access class belong. The internet firewall will include rules that specify to which areas of the company network that subnet is granted access. The VPN gateway and firewall access controls will be coordinated by configuration settings on the gateway that bind a user's distinguished name to the subnet belonging to his/her access class.

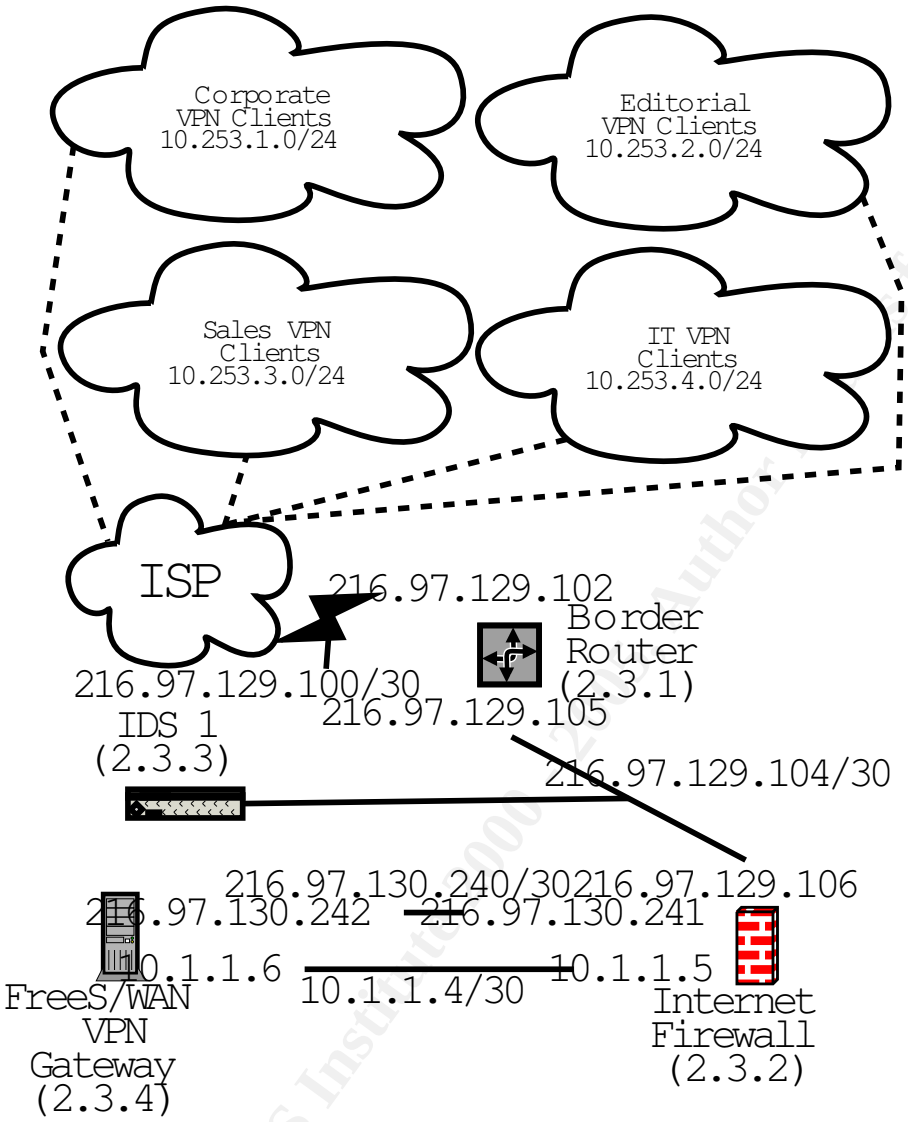
The VPN gateway system runs FreeS/WAN v. 2.01 with the X.509 patch on a Redhat Linux 9.0 system, modified kernel version 2.4.20. The X.509 patch allows the VPN gateway to use PKI for authentication which also enables it to interoperate with Windows 2000/XP clients. A patch to FreeS/WAN which allows for NAT traversal was considered unready for production use, but will be studied closely for possible inclusion in future installations.

4.3.5 VPN Clients

Since the company already owns a fair number of laptops with Windows 2000 or XP installed, the IPSec sub-system that comes standard in a Windows 2000 and XP Professional workstation installation will be the standard VPN client software used by the company. The IPSec sub-system will be configured to operate in client-only mode and use X.509 certificates signed by GIAC Enterprises own certificate authority for authentication.

4.3.6 Email Gateway

The email gateway handles email traffic going in and out of GIAC Enterprises. It is designed to trap SPAM and viruses or trojans embedded in email and strip them out of the message flow. It also rewrites headers on outbound email to hide sensitive information about the internal structure of GIAC Enterprises network and blocks 3rd-party relaying. It also runs a DNS service, both to provide DNS to the mail service for quick name resolution, and to act as a proxy for internal DNS servers. The company's forward and reverse DNS records are maintained by the company's ISP, so this gateway DNS server does not actually serve any DNS records to the internet. All inbound access to the email gateway's DNS service is blocked. The gateway DNS server performs recursive queries on behalf of internal DNS servers and forwards them its cached information. It also acts as a slave server for the internal DNS domain and provides direct DNS service to the perimeter hosts that need it. The DNS service on



(a) VPN Gateway Subnets

Figure 2: VPN Gateway Subnets

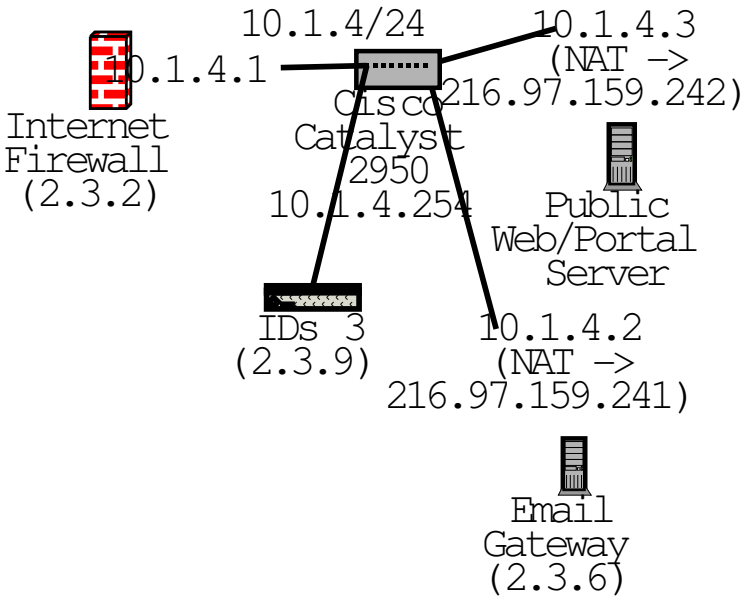


Figure 3: DMZ

this host is configured to allow queries only from perimeter hosts and internal nameservers. The internal DNS master server is configured to allow zone transfers from the email gateway and other internal DNS servers. The gateway runs MailScanner 4.22-5 to integrate anti-virus and anti-SPAM checks with the mail delivery system, Spamassassin 2.55 for SPAM detection, Sophos Anti-Virus for virus-checking and removal, Sendmail 12.8 with the latest security patches for mail transport and BIND v.9 for the DNS service on a Redhat Linux 9.0 system.

4.3.7 Network Intrusion Detection System 2

This intrusion detection system sits between the internet firewall and an internal firewall to monitor traffic from the internet and VPN gateway after network address translation has been performed (See 4.3.4). It also monitors outbound traffic before network address translation is performed. It is meant to perform several functions. First, it verifies the success of IDS 1 in detecting and blocking suspicious inbound activity. Second, it traps suspicious traffic passing through the VPN gateway, since it is unencrypted when it passes through this IDS, unless of course it is encrypted email, SSL or SSH traffic. Third, it verifies the sucess of the user network IDS and proxy server at detecting and trapping dangerous HTTP or FTP requests and will stop them if detected. This system's software configuration is identical to that of IDS 1.

4.3.8 Internal Firewall

The internal firewall offers a second line of defense against traffic inbound from the internet. It also protects sensitive areas of GIAC Enterprises' internal network from other internal systems and users. All server systems and network devices in the GIAC

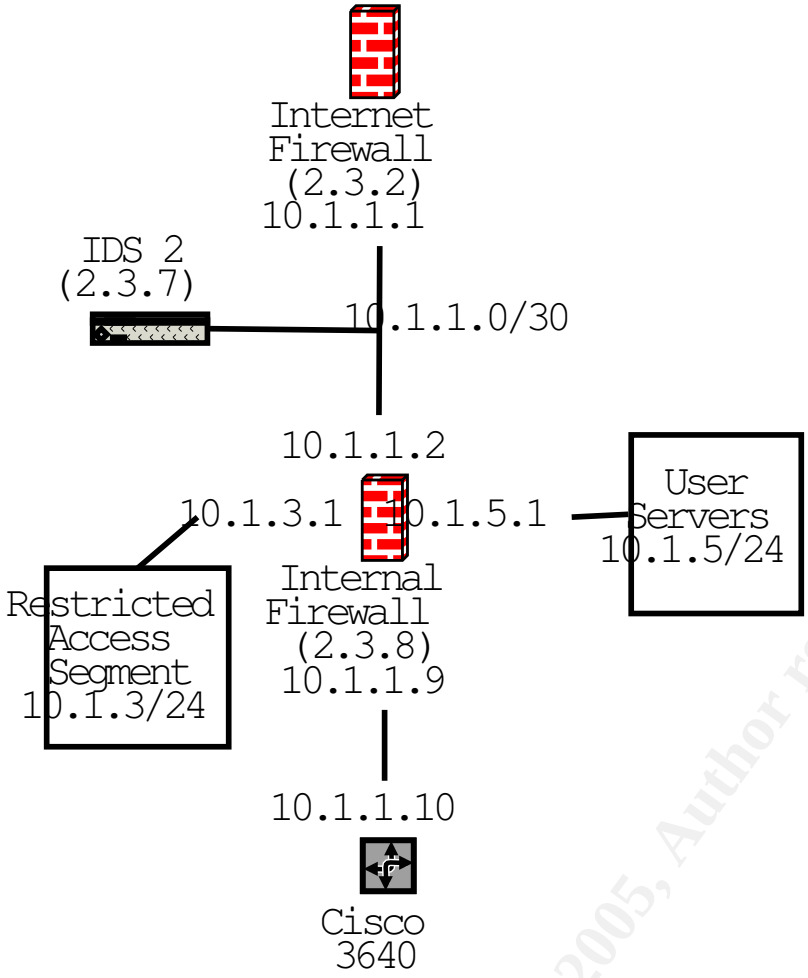


Figure 4: Internal Firewall

Enterprises network log to a remote syslog server. This logging is carried over a back-channel LAN segment through the internal firewall to the remote syslog server. Communications between the application servers and the Oracle database server are also carried over this back-channel segment. The internal firewall allows limited access to these servers for crucial maintenance without exposing them unnecessarily. The company already owns a Cisco 3640 router that was used to forward traffic between segments of their LAN. An IOS upgrade to include Cisco's firewall feature set could have enabled the router to perform the functions of the internal firewall fairly effectively, but The router's available ethernet interfaces were already taken up with user LAN segments. Had the switches in the user portion of the LAN been capable of handling vlan trunking, this problem could have been alleviated. Unfortunately, they were not. As a result, the addition of a restricted-access segment and a segment dedicated to internal servers necessitated either the introduction of another device or a costly hardware upgrade. The internal firewall runs Redhat Linux 9.0 with netfilter v. 1.2.7a.

4.3.9 Network Intrusion Detection System 3

This system inspects traffic traversing the DMZ (See 4.3.6). It is connected to one of the Gigabit ethernet ports on the Cisco 2950 switch to which all devices on that segment connect. Cisco SPAN is configured to copy all traffic to that port; the IDS monitors all incoming, outgoing, and intra-segment traffic. It is configured to alert on suspicious requests addressed to any of the servers and any unexpected activity originating from any of the servers. It runs Snort, v. 2.0.1 on Redhat Linux 9.0.

4.3.10 Network Intrusion Detection System 4

This system inspects traffic traversing the internal server LAN segment. It is connected to one of the Gigabit ethernet ports on the Cisco 2950 switch to which all devices on that segment connect. Cisco SPAN is configured to copy all traffic to that port; the IDS monitors all incoming, outgoing, and intra-segment traffic. It is configured to alert on suspicious requests addressed to any of the servers and any unexpected activity originating from any of the servers. It runs Snort, v. 2.0.1 on Redhat Linux 9.0.

4.3.11 Network Intrusion Detection System 5

This system inspects traffic traversing the restricted access LAN segment. It is connected to one of the Gigabit ethernet ports on the Cisco 2950 switch to which all devices on that segment connect. Cisco SPAN is configured to copy all traffic to that port; the IDS monitors all incoming, outgoing, and intra-segment traffic. It is configured to alert on any unencrypted communication with the Oracle service or any communication not originating from one of the application servers or the network management workstation, either of which is a sure sign of trouble. It also will alert on attempts to get unencrypted web traffic from any of the systems in this segment. The syslog server, backup server, and network management workstation all run web services that allow remote access to their management and reporting capabilities, but these connections are all SSL-enabled to protect the network management data from being visible inside GIAC Enterprises' network to any but IT staff. This system runs Snort, v. 2.0.1 on Redhat Linux 9.0.

4.3.12 Web Proxy Server

The web proxy server takes http, https, and ftp requests and fetches the requested information on behalf of the connecting client (See 4.3.10). In the process, it offers access control by requiring users to authenticate and limiting access to internet resources by matching the identity of the user with a list of ACLs based on the IP address/DNS hostname, URL, mime-type, request method, and a variety of other criteria. The proxy server is configured to block access to downloads of certain types of files, such as Windows executables and script files (except from Microsoft's automatic

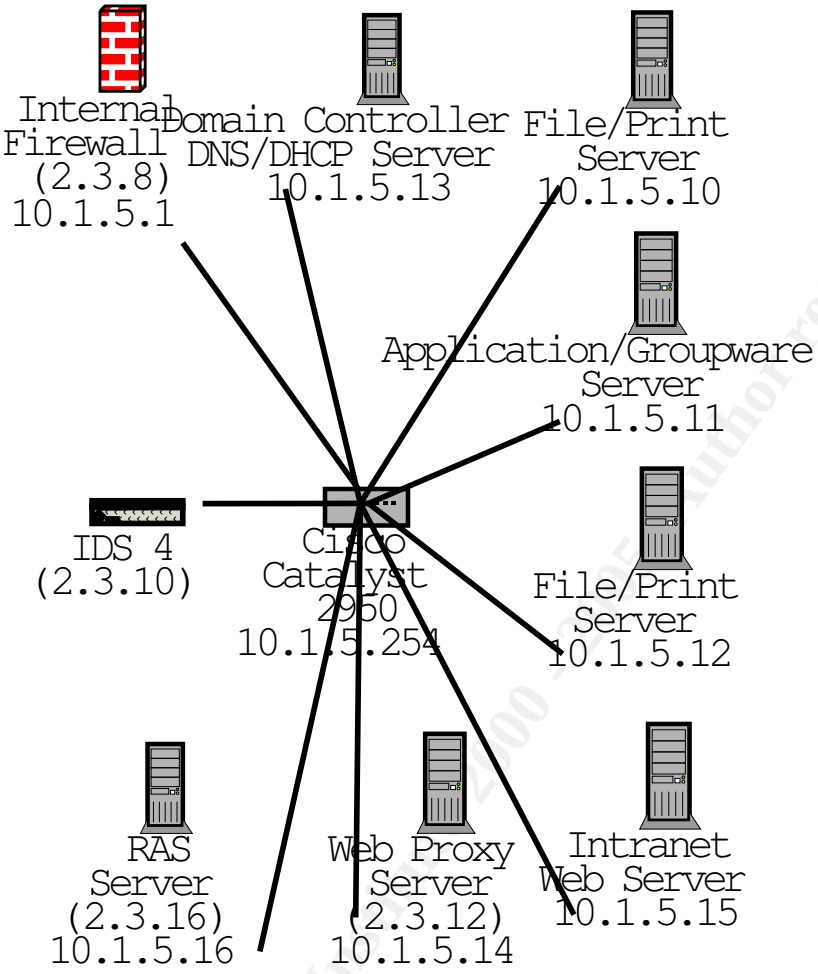


Figure 5: Internal Servers Network

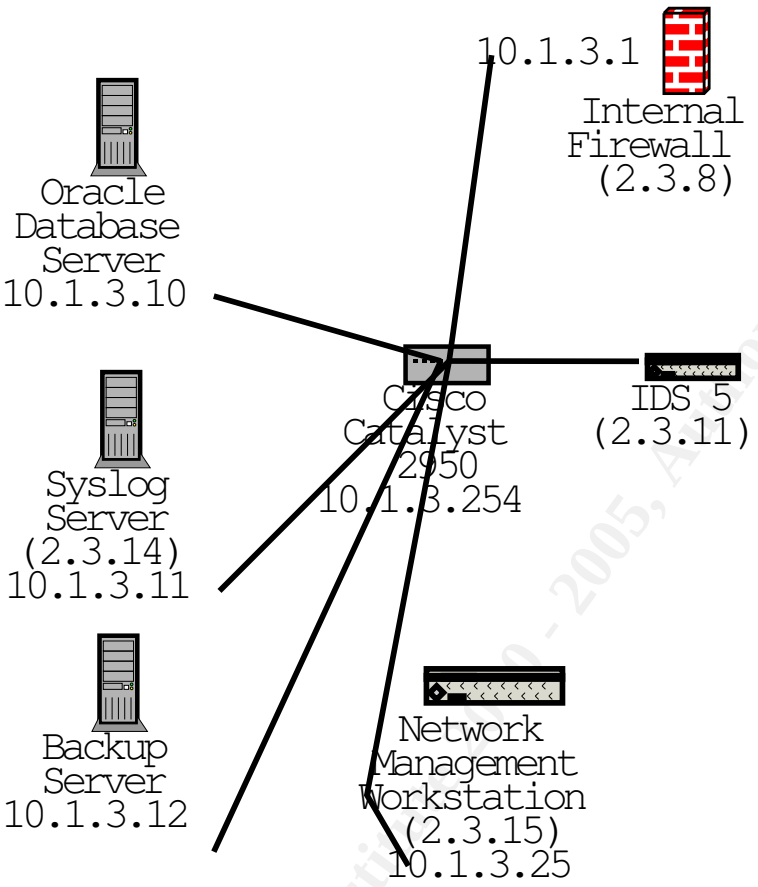


Figure 6: Restricted-Access Network

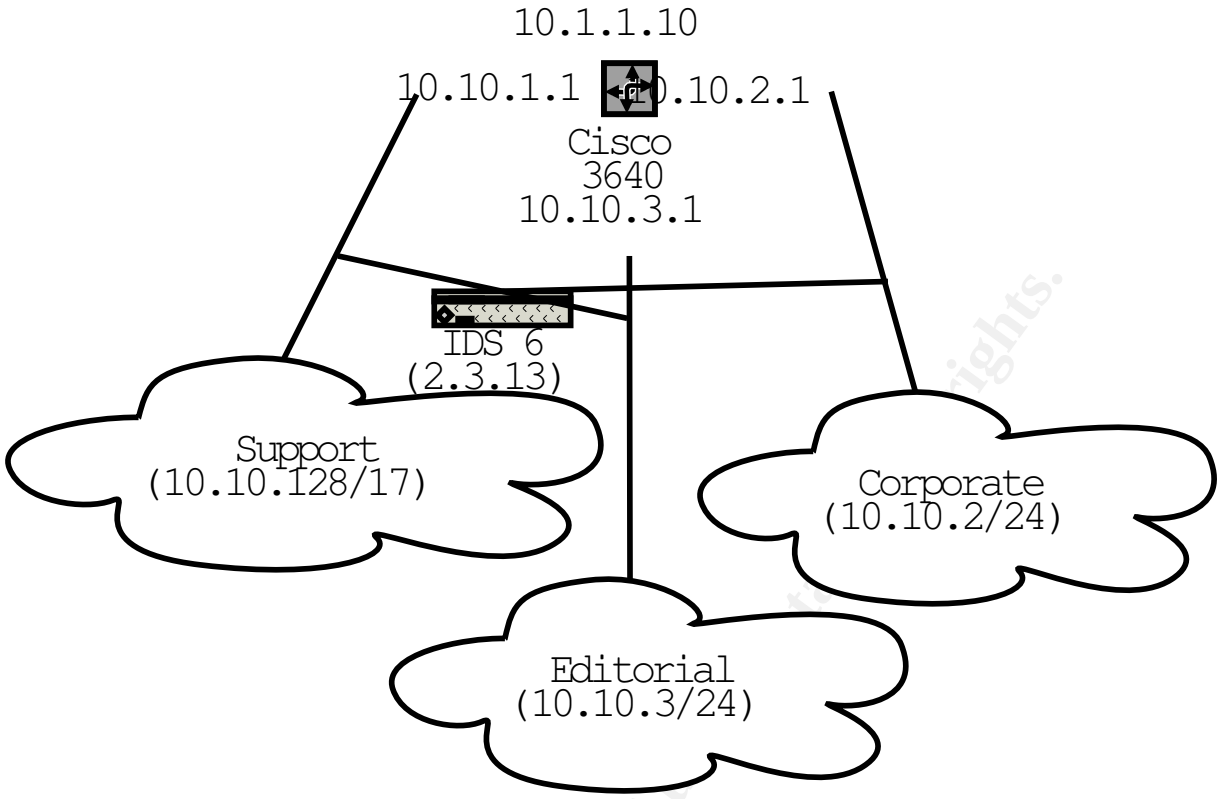


Figure 7: User Subnets

updates site), MP3s, java archives, and all very large files. It will also filter out pornography to a limited degree. The server runs Squid 2.5.3 for proxying and caching and DansGuard 2.6.1-3 for web content filtering on a Redhat 9.0 Linux system.

4.3.13 Network Intrusion Detection System 6

This system inspects traffic entering and leaving the user LAN segments. Ideally, the IDS would be monitoring all traffic on each segment, including intra-segment traffic. Unfortunately, the systems in these segments are attached to Cisco catalyst 2820 switches, which do not allow for SPAN ports. At some point in the future, the company will invest in a switch upgrade. Until that time, it was decided to attach a hub to the port leading out of each switch to the Cisco 3640 router and run the monitoring interface for each segment into the hub to capture traffic entering and leaving the segment. For the time being, this rather busy IDS will alert on suspicious requests originating from the user segments and dangerous responses coming back. It will also alert on suspicious incoming activity. The system runs Snort, v. 2.0.1 on Redhat Linux 9.0 with modifications to the system init scripts and snort rules to allow for independent configuration and control of Snort for each sensor interface.

4.3.14 Syslog Server

This system captures logs sent to it from every server and network device on the GIAC Enterprises network (see 4.3.11). It uses syslog-ng, v. 1.6 to customize logging sources and destinations so that system logs are automatically deposited in a SQL database as well as in standard log files. It uses swatch 3.0.8 to analyze logs for security violations and alert IT staff via pages, emails, and Windows desktop pop-ups when detected. The Analysis Console for Intrusion Databases, v. 0.9.6b23 and php-syslog-ng, v. 1.4 are used to provide web-based tools for searching and reporting on security incidents. IT staff connect to the web-based interfaces using HTTPS with 2-way authentication using RSA certificates signed by the Verisign Certificate Authority and a negotiated encryption algorithm of minimum 128-bit key length. The system runs Redhat Linux 9.0

4.3.15 Network Management Workstation

This system acts as the management console for configuration of network security. From this system, updates to router and firewall configurations are permitted

4.4 Enterprise-wide Features

4.4.1 Patch Management

To ease the administrative burden of staying up-to-date with security fixes on GIAC Enterprises' servers and workstations, all qualified systems will use a vendor's patch management distribution system. Systems that are not qualified will be slated for replacement as soon as possible. In the meantime, non-qualified systems will be kept up-to-date via scripted software installation with available patches. Both Redhat Network[32] and Windows Update[33] make available an automatic update feature with their patch management solutions. The Windows Update feature in Windows 2000/XP Professional and Windows 2000/2003 Server allows administrators to schedule automatic updates. All Windows 2000/XP workstations and Windows 2000/2003 servers will be scheduled to download and install available updates once per week. Windows servers and workstations will be scheduled for updating early every Sunday morning. Redhat Network's automatic update feature does not yet allow for scheduled updates. This is easily managed, however with a small shell script run from cron on a weekly basis. This script will be run once per week early Sunday morning. In addition, a staggered schedule of reboots will be set up through the Redhat Network so that any updates are applied to daemons or the kernel as soon as possible after updates are installed. The schedule will stagger the reboot of web server and database cluster members so that at least one member of each cluster is available at all times.

To prevent large numbers of systems from attempting to carry out downloads of updates over the internet, local update distribution will be used. Windows Software Updates Service[34] will be deployed on the same Windows 2000 server that deploys Sophos Anti-Virus updates. Redhat Network's Satellite Server software[35] will be

installed on the web proxy server. All other systems in the company will be configured to contact one of these two systems to fetch available updates.

Part III
GIAC Enterprises Perimeter Security Policy and Implementation

5 Internet Firewall Security Policy

5.1 Overview

The internet firewall's security policy is designed to limit access from the internet to a small set of TCP/IP hosts and services that have been available to the public or to selected parties outside GIAC Enterprises network, limit access to the internet from within GIAC Enterprises network to TCP/IP services approved for use by GIAC Enterprises employees, frustrate various techniques employed by network enumeration and exploit tools, perform network address tanslation on the RFC 1918 IP addresses used by systems in GIAC Enterprises network, limit access to the border router and the firewall itself, and maintain a detailed log of all network activity supervised by the firewall. We will examine the firewall settings that accomplish each of these tasks in detail below. See B on page 69 for the complete configuration.

5.2 Border Router and Firewall Access Rules

Technical Support personnel may obtain virtual terminal access to the firewall via SSH from company headquarters or a remote location, as specified in the global policy ruleset below. As mentioned above, the border router does not run a SSH server. Telnet access is granted only from the firewall to the ethernet interface facing the firewall. Access is granted to the firewall via its non-public interfaces implicitly as part of the GIAC Enterprises private address space. The first rule includes access via SSH or PCAnywhere. Since nearly every company system runs only one of these, one could write a separate rule for each type of access. Unfortunately, there is no easy way to separate out the two cases; each rule would have a long list of specific IP addresses to which it applies. Technical Support would have the onerous job of maintaining this long list whenever a particular device is added, removed, or changes its remote access configuration. The class C subnet 10.10.128.0/24 does not represent a physical or virtual network segment; planning for the day when that can be a reality, the IP addresses of all systems belonging to Technical Support were kept in this range. The 10.253.4/24 and 10.254.1/24 subnets represent Technical Support VPN and dialin access networks.

```
#
```

```
# Rule 19(global)
#
# Tech Support gets remote admin access to all
internal hosts via ssh or PCAnywhere
#
$IPTABLES -N Cid3F4B9237.0
$IPTABLES -A INPUT  -d 10.0.0.0/8  -m state --state
NEW  -j Cid3F4B9237.0
$IPTABLES -A INPUT  -d 1.1.1.105  -m state --state
NEW  -j Cid3F4B9237.0
$IPTABLES -N Cid3F4B9237.1
$IPTABLES -A Cid3F4B9237.0 -p tcp  -m multiport
--destination-ports 5631,22  -m state --state NEW
-j Cid3F4B9237.1
$IPTABLES -N RULE_19
$IPTABLES -A Cid3F4B9237.1  -s 10.253.4.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A Cid3F4B9237.1  -s 10.254.1.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A Cid3F4B9237.1  -s 10.10.128.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.253.4.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.253.4.0/24  -d
1.1.1.105  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.254.1.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.254.1.0/24  -d
1.1.1.105  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.10.128.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.10.128.0/24  -d
1.1.1.105  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -N Cid3F4B9237.2
$IPTABLES -A FORWARD  -d 10.0.0.0/8  -m state
--state NEW  -j Cid3F4B9237.2
$IPTABLES -A FORWARD  -d 1.1.1.105  -m state
--state NEW  -j Cid3F4B9237.2
$IPTABLES -N Cid3F4B9237.3
$IPTABLES -A Cid3F4B9237.2 -p tcp  -m multiport
--destination-ports 5631,22  -m state --state NEW
-j Cid3F4B9237.3
```

```
$IPTABLES -A Cid3F4B9237.3  -s 10.253.4.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A Cid3F4B9237.3  -s 10.254.1.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A Cid3F4B9237.3  -s 10.10.128.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.253.4.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.253.4.0/24  -d
1.1.1.105  --destination-port 5632  -m state --state
NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.254.1.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.254.1.0/24  -d
1.1.1.105  --destination-port 5632  -m state --state
NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.10.128.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.10.128.0/24  -d
1.1.1.105  --destination-port 5632  -m state --state
NEW  -j RULE_19
$IPTABLES -A RULE_19   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 19 -- ACCEPT "
$IPTABLES -A RULE_19  -j ACCEPT
#
# Rule 21(global)
#
# Staff who can SSH to firewall are allowed to telnet from
firewall to border router for remote administration.
#
$IPTABLES -N RULE_21
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d 1.1.1.105
--destination-port 23  -m state --state NEW  -j RULE_21
$IPTABLES -A RULE_21   -m limit --limit 6/minute -j LOG
--log-level 6 --log-prefix "RULE 21 -- ACCEPT "
$IPTABLES -A RULE_21  -j ACCEPT
```

5.3 Inbound Access Rules

Generally, responses to outbound requests are allowed into the network. The range of responses is largely limited by restrictions on outbound requests. Additionally, most types of inbound ICMP are blocked regardless of whether they come in response to outbound requests. Only echo replies and unreachables (ICMP type 3) are allowed in, and only in response to an outbound request. Fragmented packets are blocked, except for fragmented AH or ESP packets bound for the public IP address of the VPN

gateway. This exception is granted to allow for oversize packets resulting from IPSec encapsulation in cases where pMTU discovery fails between the IPSec gateways. Below are relevant snippets from the firewall configuration script, with extra comments added to clarify:

```
# Rules accepting packets that are part of established connections precede
# all other rules
$IPTABLES -t drop -A DROPPING -j LOG  --log-level 6
--log-prefix "RULE %N -- %A "
$IPTABLES -A INPUT   -m state --state
ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A OUTPUT  -m state --state
ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -m state --state
ESTABLISHED,RELATED -j ACCEPT
...
#
#
# Rule 1(eth0)
#
# Drop most types of inbound ICMP.  Exceptions are
unreachables and echo replies.  These are covered by
the generic rule allowing ESTABLISHED and RELATED
traffic back into the network.  We don't
specifically mention echo requests because we are
going to allow them to the VPN gateway public
interface, and they won't make it there if they are
blocked here.
#
$IPTABLES -N eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
4/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
5/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
5/1   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
5/2   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
5/3   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
13/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
14/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
15/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
16/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
```

```
9/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
10/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
37/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
38/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
4/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
5/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
5/1   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
5/2   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
5/3   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
13/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
14/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
15/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
16/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
9/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
10/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
37/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
38/0   -j eth0_In_RULE_1
$IPTABLES -A eth0_In_RULE_1    -m limit --limit
6/minute -j LOG  --log-level 4 --log-prefix "BAD
ICMP -- DENY "
$IPTABLES -A eth0_In_RULE_1   -j DROP
...
#
#
# Rule 2(eth1)
#
# Allow internet hosts to ping the VPN gateway.  We
allow this for testing VPN connectivity issues.
Rate limiting applies to the ping requests.  We
allow replies to go back, but the types allowed are
limited by the internet firewall.
#
$IPTABLES -N eth1_In_RULE_2
```

```
$IPTABLES -A FORWARD  -i eth1 -p icmp  -d 1.1.1.242
--icmp-type 8/0   -m state --state NEW  -j
eth1_In_RULE_2
$IPTABLES -A eth1_In_RULE_2    -m limit --limit
3/second --limit-burst 5 -j LOG  --log-level 6
--log-prefix "RULE 2 -- ACCEPT "
$IPTABLES -A eth1_In_RULE_2    -m limit --limit
3/second --limit-burst 5 -j ACCEPT
$IPTABLES -N eth1_Out_RULE_2
$IPTABLES -A OUTPUT  -o eth1 -p icmp  -d 1.1.1.242
--icmp-type 8/0   -m state --state NEW  -j
eth1_Out_RULE_2
$IPTABLES -A FORWARD  -o eth1 -p icmp  -d 1.1.1.242
--icmp-type 8/0   -m state --state NEW  -j
eth1_Out_RULE_2
$IPTABLES -A eth1_Out_RULE_2    -m limit --limit
3/second --limit-burst 5 -j LOG  --log-level 6
--log-prefix "RULE 2 -- ACCEPT "
$IPTABLES -A eth1_Out_RULE_2    -m limit --limit
3/second --limit-burst 5 -j ACCEPT
$IPTABLES -N eth1_In_RULE_2
```

Here are the rules allowing IPSec packets, possibly fragmented, to and from the VPN gateway interface and blocking all others:

```
#
#
# Rule 0(eth1)
#
# Allow internet hosts to connect to VPN
gateway's public interface for IPSec.
#
$IPTABLES -N Cid3F4B2699.0
$IPTABLES -A FORWARD  -i eth1 -p udp  -d 1.1.1.242
 --destination-port 500  -m state --state NEW  -j
Cid3F4B2699.0
$IPTABLES -A FORWARD  -i eth1 -p 50  -d 1.1.1.242
-f   -m state --state NEW  -j Cid3F4B2699.0
$IPTABLES -A FORWARD  -i eth1 -p 51  -d 1.1.1.242
-f   -m state --state NEW  -j Cid3F4B2699.0
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 1.1.1.241
-j RETURN
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 1.1.1.106
-j RETURN
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 10.0.0.0/8
-j RETURN
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 1.1.1.242
-j RETURN
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 1.1.2.242
```

```
-j RETURN
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 1.1.2.241
-j RETURN
$IPTABLES -N eth1_In_RULE_0_3
$IPTABLES -A Cid3F4B2699.0  -i eth1  -m state
--state NEW  -j eth1_In_RULE_0_3
$IPTABLES -A eth1_In_RULE_0_3    -m limit --limit
6/minute -j LOG  --log-level 6 --log-prefix "IPSEC
-- ACCEPT"
$IPTABLES -A eth1_In_RULE_0_3   -j ACCEPT
$IPTABLES -N Cid3F4B2699.1
$IPTABLES -A OUTPUT  -o eth1 -p udp  -d 1.1.1.242
 --destination-port 500  -m state --state NEW  -j
Cid3F4B2699.1
$IPTABLES -A OUTPUT  -o eth1 -p 50  -d 1.1.1.242
-f   -m state --state NEW  -j Cid3F4B2699.1
$IPTABLES -A OUTPUT  -o eth1 -p 51  -d 1.1.1.242  -f
  -m state --state NEW
-j Cid3F4B2699.1
$IPTABLES -A FORWARD  -o eth1 -p udp  -d 1.1.1.242
--destination-port 500  -m
state --state NEW  -j Cid3F4B2699.1
$IPTABLES -A FORWARD  -o eth1 -p 50  -d 1.1.1.242
-f   -m state --state NEW  -j Cid3F4B2699.1
$IPTABLES -A FORWARD  -o eth1 -p 51  -d 1.1.1.242
-f   -m state --state NEW  -j Cid3F4B2699.1
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 1.1.1.241
-j RETURN
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 1.1.1.106
-j RETURN
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 10.0.0.0/8
-j RETURN
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 1.1.1.242
-j RETURN
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 1.1.2.242
-j RETURN
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 1.1.2.241
-j RETURN
$IPTABLES -N eth1_Out_RULE_0_3
$IPTABLES -A Cid3F4B2699.1  -o eth1  -m state
--state NEW  -j eth1_Out_RULE_0_3
$IPTABLES -A eth1_Out_RULE_0_3    -m limit --limit
6/minute -j LOG  --log-level 6 --log-prefix "IPSEC
-- ACCEPT"
$IPTABLES -A eth1_Out_RULE_0_3   -j ACCEPT
#
#
# Rule 1(eth1)
#
```

```
# Allow the public interface of the VPN gateway to
connect with other internet hosts for IPSec.
Currently, the gateway does not actually do this,
except to initiate keying negotiations.
#
$IPTABLES -N Cid3F4C0423.0
$IPTABLES -A INPUT  -i eth1 -p udp  -s 1.1.1.242
--destination-port 500  -m state --state NEW  -j
Cid3F4C0423.0
$IPTABLES -A INPUT  -i eth1 -p 50  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.0
$IPTABLES -A INPUT  -i eth1 -p 51  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.0
$IPTABLES -A FORWARD  -i eth1 -p udp  -s 1.1.1.242
--destination-port 500  -m state --state NEW  -j
Cid3F4C0423.0
$IPTABLES -A FORWARD  -i eth1 -p 50  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.0
$IPTABLES -A FORWARD  -i eth1 -p 51  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.0
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 1.1.1.106
-j RETURN
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 1.1.1.241
-j RETURN
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 1.1.2.241
-j RETURN
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 1.1.2.242
-j RETURN
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 1.1.1.242
-j RETURN
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 10.0.0.0/8
-j RETURN
$IPTABLES -N eth1_In_RULE_1_3
$IPTABLES -A Cid3F4C0423.0  -i eth1  -m state
--state NEW  -j eth1_In_RULE_1_3
$IPTABLES -A eth1_In_RULE_1_3    -m limit --limit
6/minute -j LOG  --log-level 6 --log-prefix "IPSEC
-- ACCEPT "
$IPTABLES -A eth1_In_RULE_1_3   -j ACCEPT
$IPTABLES -N Cid3F4C0423.1
$IPTABLES -A FORWARD  -o eth1 -p udp  -s 1.1.1.242
--destination-port 500  -m state --state NEW  -j
Cid3F4C0423.1
$IPTABLES -A FORWARD  -o eth1 -p 50  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.1
$IPTABLES -A FORWARD  -o eth1 -p 51  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.1
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 1.1.1.106
-j RETURN
```

```
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 1.1.1.241
-j RETURN
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 1.1.2.241
-j RETURN
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 1.1.2.242
-j RETURN
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 1.1.1.242
-j RETURN
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 10.0.0.0/8
-j RETURN
$IPTABLES -N eth1_Out_RULE_1_3
$IPTABLES -A Cid3F4C0423.1  -o eth1  -m state
--state NEW  -j eth1_Out_RULE_1_3
$IPTABLES -A eth1_Out_RULE_1_3    -m limit --limit
6/minute -j LOG  --log-level 6 --log-prefix "IPSEC
-- ACCEPT "
$IPTABLES -A eth1_Out_RULE_1_3   -j ACCEPT
...
#
# Rule 0(global)
#
# block fragments.  Fragmented packets that are part
of an AH or ESP packet to the VPN gateway are
excepted.  (See eth1 rules).  They are allowed there
because the extra IPSec headers may make the packet
too large for our pipes.
#
$IPTABLES -N RULE_0
$IPTABLES -A OUTPUT -p ip  -f   -j RULE_0
$IPTABLES -A INPUT -p ip  -f   -j RULE_0
$IPTABLES -A FORWARD -p ip  -f   -j RULE_0
$IPTABLES -A RULE_0   -m limit --limit 6/minute -j
LOG  --log-level 1 --log-prefix "FRAG -- DENY"
$IPTABLES -A RULE_0  -j DROP
```

A large subset of TCP scanning techniques, including ACK, FIN, null and xmas tree scans, is blocked. The rules do not try to detect SYN (half-open) scans, since these cannot easily be differentiated from legitimate connection attempts by Netfilter until the prober sends a RST packet. By then, it is already too late. We leave it to the IDS1 system to detect and respond to SYN scans. While it is likely that a very stealthy SYN probe could fly under the IDS, most will get caught. The firewall responds to detected probes by dropping the offending packets. The decision to react this way is a trade-off. On the positive side, dropping the scan packets prevents the attacker from receiving anything from the security devices themselves. Were we to send TCP RSTs or ICMP unreachables in response to the probes, the attacker could use passive fingerprinting to guess the OS and possibly the firewalling software we are running on the security devices. In the case of ICMP unreachables, we would also be giving him the public IP address of the firewall. Even in the case of a TCP RST, the response packet will

contain information (TTL and TCP Window size are two examples) that could tip the attacker to the presence and nature of the firewall. This assumes that the attacker has packets from the target host with which to make comparisons, but obtaining these is a trivial exercise. On the negative side, this makes all ports appear open – or filtered – to the probing system.[36] A savvy attacker would recognize that a security device is filtering responses to his probes. A newbie may just go ahead and launch his favorite attacks on the target.

What would the prober learn about the network by attacking this design? First, he could learn that SYN scans are treated differently from other types of scans. A simple TCP connect scan – until it is blocked by the IDS – would show that only ports for which we allow inbound connections are open. The other scans would show all ports open. A comparison of results would tell the prober that one or more security devices have intervened. He is likely to figure out that TCP connections are blocked except to a few ports and that the filtering devices are configured to drop all TCP scan packets of certain types. The prober would eventually figure this out no matter what kind of response we designed. At least this way, the prober gets no information that will help him attack the security devices themselves. Here is the relevant ruleset with additional comments:

```
#
# Rule 1(global)
#
# Drop Stealth scans.  Normally, open ports drop bad packets,
while closed ports send a TCP RST. This blocks the destination
response and makes all ports appear open.
#
$IPTABLES -N RULE_1
$IPTABLES -A OUTPUT -p tcp  --tcp-flags ALL NONE  -j RULE_1
$IPTABLES -A OUTPUT -p tcp  --tcp-flags URG,PSH,FIN URG,PSH,FIN  -j RULE_1
$IPTABLES -A OUTPUT --protocol tcp --tcp-flags SYN,ACK,RST,FIN FIN
-m state --state NEW  -j RULE_1
$IPTABLES -A OUTPUT --protocol tcp --tcp-flags ACK ACK -m state
--state NEW  -j RULE_1
$IPTABLES -A INPUT -p tcp  --tcp-flags ALL NONE  -j RULE_1
$IPTABLES -A INPUT -p tcp  --tcp-flags ALL URG,ACK,PSH,RST,SYN,FIN
-j RULE_1
$IPTABLES -A INPUT --protocol tcp --tcp-flags SYN,ACK,RST,FIN FIN
-m state --state NEW  -j RULE_1
$IPTABLES -A INPUT --protocol tcp --tcp-flags ACK ACK -m state
--state NEW  -j RULE_1
$IPTABLES -A FORWARD -p tcp  --tcp-flags ALL NONE  -j RULE_1
$IPTABLES -A FORWARD -p tcp  --tcp-flags URG,PSH,FIN URG,PSH,FIN  -j RULE_1
$IPTABLES -A FORWARD --protocol tcp --tcp-flags SYN,ACK,RST,FIN FIN
-m state --state NEW  -j RULE_1
$IPTABLES -A FORWARD --protocol tcp --tcp-flags ACK ACK -m state
--state NEW  -j RULE_1
$IPTABLES -A RULE_1   -m limit --limit 6/minute -j LOG  --log-level 4
--log-prefix "TCP Scan -- DENY "
```

```
$IPTABLES -A RULE_1  -j DROP
```

Inbound connections to the email gateway's SMTP services and web/portal server's web services are allowed generally. VPN clients are also allowed inbound connections from the VPN gateway's private interface to a number of services on the internal server network. Technical Support personnel are also allowed access to all internal systems via SSH or PCAnywhere, as noted above 5.2 on page 32, and to the telnet service on some legacy Cisco switches. Here are the relevant snippets from the firewall configuration script, with additional comments:

```
#
# Rule 2(global)
#
# Unrestricted web access to web/portal server
#
$IPTABLES -N RULE_2
$IPTABLES -A OUTPUT -p tcp  -m multiport  -d
10.1.4.3  --destination-ports 80,443  -m state
--state NEW  -j RULE_2
$IPTABLES -A FORWARD -p tcp  -m multiport  -d
10.1.4.3  --destination-ports 80,443  -m state
--state NEW  -j RULE_2
$IPTABLES -A RULE_2   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 2 --
ACCEPT "
$IPTABLES -A RULE_2  -j ACCEPT
#
# Rule 3(global)
#
# Unrestricted SMTP access to email gateway
#
$IPTABLES -N RULE_3
$IPTABLES -A OUTPUT -p tcp  -m multiport  -d
10.1.4.2  --destination-ports 25,465  -m state
--state NEW  -j RULE_3
$IPTABLES -A FORWARD -p tcp  -m multiport  -d
10.1.4.2  --destination-ports 25,465  -m state
--state NEW  -j RULE_3
$IPTABLES -A RULE_3   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 3 -- ACCEPT "
$IPTABLES -A RULE_3  -j ACCEPT
...
#
# Rule 13(global)
#
# All VPN users get access to internal servers for
various services
#
$IPTABLES -N RULE_13
```

```
$IPTABLES -A FORWARD -p tcp  -m multiport  -s
10.253.0.0/16  -d 10.1.5.0/24  --destination-ports
139,135,42,445,88,389,636,3268,3269,53,25,465,143,
993,80  -m state --state NEW  -j RULE_13
$IPTABLES -A FORWARD -p tcp  -m multiport  -s
10.253.0.0/16  -d 10.1.5.0/24  --destination-ports
443,21,20,3128  -m state --state NEW  -j RULE_13
$IPTABLES -A FORWARD -p udp  -m multiport  -s
10.253.0.0/16  -d 10.1.5.0/24  --destination-ports
138,137,53,88  -m state --state NEW  -j RULE_13
$IPTABLES -A RULE_13   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 13 -- ACCEPT "
$IPTABLES -A RULE_13  -j ACCEPT
...
#
# Rule 19(global)
#
# Tech Support gets remote admin access to all
internal hosts and the border router via ssh or
PCAnywhere
#
$IPTABLES -N Cid3F4B9237.0
$IPTABLES -A INPUT  -d 10.0.0.0/8  -m state --state
NEW  -j Cid3F4B9237.0
$IPTABLES -A INPUT  -d 1.1.1.105  -m state --state
NEW  -j Cid3F4B9237.0
$IPTABLES -N Cid3F4B9237.1
$IPTABLES -A Cid3F4B9237.0 -p tcp  -m multiport
--destination-ports 5631,22  -m state --state NEW
-j Cid3F4B9237.1
$IPTABLES -N RULE_19
$IPTABLES -A Cid3F4B9237.1  -s 10.253.4.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A Cid3F4B9237.1  -s 10.254.1.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A Cid3F4B9237.1  -s 10.10.128.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.253.4.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.253.4.0/24  -d
1.1.1.105  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.254.1.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.254.1.0/24  -d
1.1.1.105  --destination-port 5632  -m state
```

```
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.10.128.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.10.128.0/24  -d
1.1.1.105  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -N Cid3F4B9237.2
$IPTABLES -A FORWARD  -d 10.0.0.0/8  -m state
--state NEW  -j Cid3F4B9237.2
$IPTABLES -A FORWARD  -d 1.1.1.105  -m state
--state NEW  -j Cid3F4B9237.2
$IPTABLES -N Cid3F4B9237.3
$IPTABLES -A Cid3F4B9237.2 -p tcp  -m multiport
--destination-ports 5631,22  -m state --state NEW
-j Cid3F4B9237.3
$IPTABLES -A Cid3F4B9237.3  -s 10.253.4.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A Cid3F4B9237.3  -s 10.254.1.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A Cid3F4B9237.3  -s 10.10.128.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.253.4.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.253.4.0/24  -d
1.1.1.105  --destination-port 5632  -m state --state
NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.254.1.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.254.1.0/24  -d
1.1.1.105  --destination-port 5632  -m state --state
NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.10.128.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.10.128.0/24  -d
1.1.1.105  --destination-port 5632  -m state --state
NEW  -j RULE_19
$IPTABLES -A RULE_19   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 19 -- ACCEPT "
$IPTABLES -A RULE_19  -j ACCEPT
```

All the systems on the network perimeter are allowed to send SNMP traps or informs to a network management station inside GIAC Enterprises network. DMZ hosts are otherwise forbidden from making any connections to the firewall or internal hosts. The

border router and firewall are also allowed to send logs to the central syslog server. Finally, inbound ident connections to the email gateway are specifically rejected with an ICMP port-unreachable packet so that SMTP connections from gateways that use ident queries don't hang. Here are the relevant snippets from the firewall configuration script with additional comments:

```
#
# Rule 9(global)
#
# Allow SNMP traps/informs back to Management station
#
$IPTABLES -N RULE_9
$IPTABLES -A OUTPUT -p tcp  -s 10.1.1.1  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A OUTPUT -p udp  -s 10.1.1.1  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 10.1.1.6  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 1.1.1.105  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.3  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.2  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.254  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p udp  -s 10.1.1.6  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p udp  -s 1.1.1.105  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p udp  -s 10.1.4.3  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p udp  -s 10.1.4.2  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p udp  -s 10.1.4.254  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
```

```
$IPTABLES -A RULE_9   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 9 -- ACCEPT "
$IPTABLES -A RULE_9  -j ACCEPT
#
# Rule 10(global)
#
# blocking access to internal net
# and firewall from DMZ.
# Rules that permit access to servers
# on DMZ should be added above.
#
$IPTABLES -N Cid3F4B2716.0
$IPTABLES -A INPUT  -s 10.1.6.6  -j Cid3F4B2716.0
$IPTABLES -A INPUT  -s 10.1.4.3  -j Cid3F4B2716.0
$IPTABLES -A INPUT  -s 10.1.4.2  -j Cid3F4B2716.0
$IPTABLES -A INPUT  -s 10.1.6.5  -j Cid3F4B2716.0
$IPTABLES -N RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 1.1.1.106  -j RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 1.1.1.241  -j RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 10.1.1.5  -j RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 10.1.4.1  -j RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 10.1.1.1  -j RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 10.1.6.2  -j RULE_10
$IPTABLES -A INPUT  -s 10.1.6.6  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A INPUT  -s 10.1.4.3  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A INPUT  -s 10.1.4.2  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A INPUT  -s 10.1.6.5  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A FORWARD  -s 10.1.6.6  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A FORWARD  -s 10.1.4.3  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A FORWARD  -s 10.1.4.2  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A FORWARD  -s 10.1.6.5  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A RULE_10   -m limit --limit 6/minute -j
LOG  --log-level 4 --log-prefix "RULE 10 -- DENY "
$IPTABLES -A RULE_10  -j DROP

...
#
# Rule 16(global)
#
# Firewall and Border Router may send logging
messages to central syslog server.  Firewall's logs
are routed out its logging interface.  The border
router has no logging interface, so we allow logs to
```

```
come from its first ethernet interface
#
$IPTABLES -N RULE_16
$IPTABLES -A OUTPUT -p tcp  -s 10.1.6.2  -d
10.1.3.11  --destination-port 514  -m state --state
NEW  -j RULE_16
$IPTABLES -A OUTPUT -p udp  -s 10.1.6.2  -d
10.1.3.11  --destination-port 514  -m state --state
NEW  -j RULE_16
$IPTABLES -A FORWARD -p tcp  -s 1.1.1.105  -d
10.1.3.11  --destination-port 514  -m state --state
NEW  -j RULE_16
$IPTABLES -A FORWARD -p udp  -s 1.1.1.105  -d
10.1.3.11  --destination-port 514  -m state --state
NEW  -j RULE_16
$IPTABLES -A RULE_16   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 16 -- ACCEPT "
$IPTABLES -A RULE_16  -j ACCEPT
#
# Rule 17(global)
#
# Reject ident queries to the email gateway with a
port unreachable, so smtp conversations don't hang.
#
$IPTABLES -N RULE_17
$IPTABLES -A OUTPUT -p tcp  -d 10.1.4.2
--destination-port 113  -j RULE_17
$IPTABLES -A FORWARD -p tcp  -d 10.1.4.2
--destination-port 113  -j RULE_17
$IPTABLES -A RULE_17   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 17 -- REJECT "
$IPTABLES -A RULE_17  -j REJECT  --reject-with
icmp-port-unreachable
```

5.4 Outbound Access Rules

Outbound ICMP is restricted carefully. Echo requests and unreachables are allowed out, the latter only in response to an inbound connection request. Echo replies to pings to the VPN gateway are allowed out as well. These are covered by the generic rules permiting packets belonging to established connections. All other standard types are blocked. This protects the network from traceroute probes and other types of reconnaisance. It also prevents most types of inbound ICMP traffic, because if no outbound request is allowed, no inbound response will be allowed either. Outbound fragments and TCP Scans are blocked in the same way as for inbound attempts. Here are snippets from the firewall configuration script with additional comments:

```
#
# Rule 0(eth0)
#
```

```
# Drop most types of outbound ICMP.  Exceptions are
echo requests and ICMP unreachables.  Many of these
types will be dropped by the implicit default DENY,
but we want to capture them in a special rule.
There are very few cases in which these packets are
likely to appear for legitimate reasons.  We include
source quench in this group because it can be used
for a DOS.
#
$IPTABLES -N eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
11/1   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
11/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
5/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
5/1   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
5/2   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
5/3   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
13/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
15/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
14/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
16/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
9/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
10/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
37/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
38/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
11/1   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
11/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
5/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
5/1   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
5/2   -j eth0_Out_RULE_0
```

```
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
5/3   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
13/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
15/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
14/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
16/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
9/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
10/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
37/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
38/0   -j eth0_Out_RULE_0
$IPTABLES -A eth0_Out_RULE_0    -m limit --limit
6/minute -j LOG  --log-level 4 --log-prefix "Bad
ICMP -- DENY "
$IPTABLES -A eth0_Out_RULE_0   -j DROP
```

The firewall, the email gateway, and the web/portal server also fulfill the role of time hosts for the rest of the company's network. They are allowed access to the NTP ports on nine public NTP servers. Each server synchronizes with three different time servers. All internal hosts are allowed to use the NTP service on these hosts for time synchronization. No other access to internet time servers is granted:

```
#
# Rule 7(global)
#
# All hosts get time from these servers
#
$IPTABLES -N RULE_7
$IPTABLES -A INPUT -p tcp  -s 10.0.0.0/8  -d
1.1.1.106  --destination-port 123  -m state
--state NEW  -j RULE_7
$IPTABLES -A INPUT -p tcp  -s 1.1.1.105  -d
1.1.1.106  --destination-port 123  -m state
--state NEW  -j RULE_7
$IPTABLES -A INPUT -p udp  -s 10.0.0.0/8  -d
1.1.1.106  --destination-port 123  -m state
--state NEW  -j RULE_7
$IPTABLES -A INPUT -p udp  -s 1.1.1.105  -d
1.1.1.106  --destination-port 123  -m state
--state NEW  -j RULE_7
$IPTABLES -N Cid3F4B36BD.0
$IPTABLES -A OUTPUT  -s 10.0.0.0/8  -m state
```

```
--state NEW  -j Cid3F4B36BD.0
$IPTABLES -A OUTPUT  -s 1.1.1.105  -m state --state
NEW  -j Cid3F4B36BD.0
$IPTABLES -A Cid3F4B36BD.0 -p tcp  -d 10.1.4.2
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -A Cid3F4B36BD.0 -p tcp  -d 10.1.4.3
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -N Cid3F4B36BD.1
$IPTABLES -A OUTPUT  -s 10.0.0.0/8  -m state --state
NEW  -j Cid3F4B36BD.1
$IPTABLES -A OUTPUT  -s 1.1.1.105  -m state --state
NEW  -j Cid3F4B36BD.1
$IPTABLES -A Cid3F4B36BD.1 -p udp  -d 10.1.4.2
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -A Cid3F4B36BD.1 -p udp  -d 10.1.4.3
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -N Cid3F4B36BD.2
$IPTABLES -A FORWARD  -s 10.0.0.0/8  -m state
--state NEW  -j Cid3F4B36BD.2
$IPTABLES -A FORWARD  -s 1.1.1.105  -m state
--state NEW  -j Cid3F4B36BD.2
$IPTABLES -A Cid3F4B36BD.2 -p tcp  -d 10.1.4.2
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -A Cid3F4B36BD.2 -p tcp  -d 10.1.4.3
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -N Cid3F4B36BD.3
$IPTABLES -A FORWARD  -s 10.0.0.0/8  -m state
--state NEW  -j Cid3F4B36BD.3
$IPTABLES -A FORWARD  -s 1.1.1.105  -m state
--state NEW  -j Cid3F4B36BD.3
$IPTABLES -A Cid3F4B36BD.3 -p udp  -d 10.1.4.2
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -A Cid3F4B36BD.3 -p udp  -d 10.1.4.3
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -A RULE_7   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 7 -- ACCEPT "
$IPTABLES -A RULE_7  -j ACCEPT
...
#
# Rule 14(global)
```

```
#
# Allow these servers to contact selected NTP servers
#
$IPTABLES -N RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
209.51.161.238  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
132.236.56.250  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
128.59.59.177  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
216.204.156.2  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
198.147.37.140  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
65.211.109.1  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
65.211.109.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
128.105.39.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
128.105.37.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
209.51.161.238  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
132.236.56.250  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
128.59.59.177  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
216.204.156.2  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
198.147.37.140  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
```

```
65.211.109.1  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
65.211.109.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
128.105.39.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
128.105.37.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -N Cid3F4B361B.0
$IPTABLES -A FORWARD  -s 10.1.4.2  -m state --state
NEW  -j Cid3F4B361B.0
$IPTABLES -A FORWARD  -s 10.1.4.3  -m state --state
NEW  -j Cid3F4B361B.0
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 209.51.161.238
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 132.236.56.250
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 128.59.59.177
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 216.204.156.2
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 198.147.37.140
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 65.211.109.1
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 65.211.109.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 128.105.39.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 128.105.37.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -N Cid3F4B361B.1
$IPTABLES -A FORWARD  -s 10.1.4.2  -m state --state
NEW  -j Cid3F4B361B.1
$IPTABLES -A FORWARD  -s 10.1.4.3  -m state --state
NEW  -j Cid3F4B361B.1
```

```
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 209.51.161.238
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 132.236.56.250
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 128.59.59.177
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 216.204.156.2
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 198.147.37.140
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 65.211.109.1
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 65.211.109.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 128.105.39.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 128.105.37.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A RULE_14   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 14 -- ACCEPT "
$IPTABLES -A RULE_14  -j ACCEPT
```

The email gateway is allowed to contact internet mailservers to conduct SMTP transactions. It also runs a caching-only DNS service to improve its mail-handling response time and serves as a forwarder for our internal nameserver so it doesn't have to contact internet nameservers directly. The email gateway DNS service is allowed to contact internet DNS servers for name resolution. The internal nameserver is allowed to contact the email gateway to forward queries it cannot resolve on its own:

```
#
# Rule 4(global)
#
# Email gateway allowed access to forward mail,
except to firewall
#
$IPTABLES -N Cid3F4B29D6.0
$IPTABLES -A INPUT -p tcp  -m multiport  -s 10.1.4.2
 --destination-ports 25,465,53  -m state --state NEW
 -j Cid3F4B29D6.0
```

```
$IPTABLES -A INPUT -p udp  -s 10.1.4.2
--destination-port 53  -m state --state NEW  -j
Cid3F4B29D6.0
$IPTABLES -A FORWARD -p tcp  -m multiport  -s
10.1.4.2  --destination-ports 25,465,53  -m state
--state NEW  -j Cid3F4B29D6.0
$IPTABLES -A FORWARD -p udp  -s 10.1.4.2
--destination-port 53  -m state --state NEW  -j
Cid3F4B29D6.0
$IPTABLES -A Cid3F4B29D6.0  -d 1.1.1.106  -j RETURN
$IPTABLES -A Cid3F4B29D6.0  -d 1.1.1.241  -j RETURN
$IPTABLES -A Cid3F4B29D6.0  -d 10.1.1.5  -j RETURN
$IPTABLES -A Cid3F4B29D6.0  -d 10.1.4.1  -j RETURN
$IPTABLES -A Cid3F4B29D6.0  -d 10.1.1.1  -j RETURN
$IPTABLES -A Cid3F4B29D6.0  -d 10.1.6.2  -j RETURN
$IPTABLES -N RULE_4_3
$IPTABLES -A Cid3F4B29D6.0  -m state --state NEW
-j RULE_4_3
$IPTABLES -A RULE_4_3   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 4 -- ACCEPT "
$IPTABLES -A RULE_4_3  -j ACCEPT
#
# Rule 6(global)
#
# Internal DNS master can forward outside queries to
caching-only DNS server on email gateway.  It can
also forward zone update notifications. Bastion
hosts use email gateway as their primary DNS server
as well.
#
$IPTABLES -N RULE_6
$IPTABLES -A OUTPUT -p tcp  -s 10.1.1.1  -d 10.1.4.2
 --destination-port 53  -m state --state NEW  -j
RULE_6
$IPTABLES -A OUTPUT -p udp  -s 10.1.1.1  -d 10.1.4.2
 --destination-port 53  -m state --state NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.5.13  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.6.15  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.1.6  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 1.1.1.105  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.3  -d
```

```
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.2  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.254  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.5.13  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.6.15  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.1.6  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 1.1.1.105  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.4.3  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.4.2  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.4.254  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A RULE_6   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 6 -- ACCEPT "
$IPTABLES -A RULE_6  -j ACCEPT
```

All internal systems are allowed to perform pings, traceroutes, and whois queries to internet hosts for network troubleshooting, but FTP and web requests must go through the web proxy. No direct connections to internet hosts are allowed. Requests that the web proxy can't cache are simply passed through it:

```
#
# Rule 9(global)
#
# Web Proxy does all direct connection to internet web/ftp servers.  We explicitly
# exclude access to ports on the firewall.
#
$IPTABLES -N Cid3F4B36CD.0
$IPTABLES -A INPUT -p tcp  -m multiport  -s 10.1.5.14  --destination-ports 21,20,80,443  -m st
$IPTABLES -A FORWARD -p tcp  -m multiport  -s 10.1.5.14  --destination-ports 21,20,80,443  -m
$IPTABLES -A Cid3F4B36CD.0  -d 1.1.1.106  -j RETURN
$IPTABLES -A Cid3F4B36CD.0  -d 1.1.1.241  -j RETURN
```

```
$IPTABLES -A Cid3F4B36CD.0  -d 10.1.1.5  -j RETURN
$IPTABLES -A Cid3F4B36CD.0  -d 10.1.4.1  -j RETURN
$IPTABLES -A Cid3F4B36CD.0  -d 10.1.1.1  -j RETURN
$IPTABLES -A Cid3F4B36CD.0  -d 10.1.6.2  -j RETURN
$IPTABLES -N RULE_9_3
$IPTABLES -A Cid3F4B36CD.0  -m state --state NEW  -j RULE_9_3
$IPTABLES -A RULE_9_3   -m limit --limit 6/minute -j LOG  --log-level 6 --log-prefix "RULE 9 --
$IPTABLES -A RULE_9_3  -j ACCEPT
#
# Rule 10(global)
#
# allow troubleshooting anything from internal hosts.  The VPN gateway's
# public interface is included in this list because we may need to do
# troubleshooting of the parent interface to detect source of problems
# with IPSec
#
$IPTABLES -N RULE_10
$IPTABLES -A INPUT -p icmp  -s 10.0.0.0/8  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A INPUT -p icmp  -s 1.1.1.242  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A INPUT -p icmp  -s 10.1.1.6  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A INPUT -p icmp  -s 10.1.6.5  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A INPUT -p tcp  -s 10.0.0.0/8  --source-port 1024:65535  --destination-port 43  -m s
$IPTABLES -A INPUT -p tcp  -s 1.1.1.242  --source-port 1024:65535  --destination-port 43  -m st
$IPTABLES -A INPUT -p tcp  -s 10.1.1.6  --source-port 1024:65535  --destination-port 43  -m sta
$IPTABLES -A INPUT -p tcp  -s 10.1.6.5  --source-port 1024:65535  --destination-port 43  -m sta
$IPTABLES -A INPUT -p udp  -s 10.0.0.0/8  --destination-port 33434:33524  -m state --state NEW
$IPTABLES -A INPUT -p udp  -s 1.1.1.242  --destination-port 33434:33524  -m state --state NEW
$IPTABLES -A INPUT -p udp  -s 10.1.1.6  --destination-port 33434:33524  -m state --state NEW
$IPTABLES -A INPUT -p udp  -s 10.1.6.5  --destination-port 33434:33524  -m state --state NEW
$IPTABLES -A OUTPUT -p icmp  -s 10.0.0.0/8  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A OUTPUT -p icmp  -s 1.1.1.242  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A OUTPUT -p icmp  -s 10.1.1.6  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A OUTPUT -p icmp  -s 10.1.6.5  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A OUTPUT -p tcp  -s 10.0.0.0/8  --source-port 1024:65535  --destination-port 43  -m
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.242  --source-port 1024:65535  --destination-port 43  -m s
$IPTABLES -A OUTPUT -p tcp  -s 10.1.1.6  --source-port 1024:65535  --destination-port 43  -m st
$IPTABLES -A OUTPUT -p tcp  -s 10.1.6.5  --source-port 1024:65535  --destination-port 43  -m st
$IPTABLES -A OUTPUT -p udp  -s 10.0.0.0/8  --destination-port 33434:33524  -m state --state NEW
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.242  --destination-port 33434:33524  -m state --state NEW
$IPTABLES -A OUTPUT -p udp  -s 10.1.1.6  --destination-port 33434:33524  -m state --state NEW
$IPTABLES -A OUTPUT -p udp  -s 10.1.6.5  --destination-port 33434:33524  -m state --state NEW
$IPTABLES -A FORWARD -p icmp  -s 10.0.0.0/8  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A FORWARD -p icmp  -s 1.1.1.242  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A FORWARD -p icmp  -s 10.1.1.6  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A FORWARD -p icmp  -s 10.1.6.5  --icmp-type 8/0   -m state --state NEW  -j RULE_10
$IPTABLES -A FORWARD -p tcp  -s 10.0.0.0/8  --source-port 1024:65535  --destination-port 43  -m
$IPTABLES -A FORWARD -p tcp  -s 1.1.1.242  --source-port 1024:65535  --destination-port 43  -m
$IPTABLES -A FORWARD -p tcp  -s 10.1.1.6  --source-port 1024:65535  --destination-port 43  -m s
$IPTABLES -A FORWARD -p tcp  -s 10.1.6.5  --source-port 1024:65535  --destination-port 43  -m s
$IPTABLES -A FORWARD -p udp  -s 10.0.0.0/8  --destination-port 33434:33524  -m state --state NE
$IPTABLES -A FORWARD -p udp  -s 1.1.1.242  --destination-port 33434:33524  -m state --state NEW
$IPTABLES -A FORWARD -p udp  -s 10.1.1.6  --destination-port 33434:33524  -m state --state NEW
$IPTABLES -A FORWARD -p udp  -s 10.1.6.5  --destination-port 33434:33524  -m state --state NEW
```

```
$IPTABLES -A RULE_10   -m limit --limit 6/minute -j LOG  --log-level 6 --log-prefix "RULE 10 --
$IPTABLES -A RULE_10  -j ACCEPT
```

The Oracle database cluster is allowed to contact a remote payment processor listening on TCP port 4999. These communications are encrypted using SSL. The cluster always initiates this contact:

```
#
# Rule 5(global)
#
# Oracle Database Server allowed to contact
Heartland Payment Systems Secure Processor via
TLS-encrypted channel.
#
$IPTABLES -N RULE_5
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.10  -d
1.100.100.1  --destination-port 4999  -j RULE_5
$IPTABLES -A RULE_5   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 5 -- DENY "
$IPTABLES -A RULE_5  -j DROP
```

Finally, we allow the network management station to perform SNMP queries to inward-facing interfaces on the firewall, border router, and DMZ hosts:

```
#
# Rule 18(global)
#
# Management station allowed to perform SNMP queries
to all network devices
#
$IPTABLES -N RULE_18
$IPTABLES -A INPUT -p tcp  -s 10.1.3.25  -d 10.1.1.1
 --destination-port 161  -m state --state NEW  -j
RULE_18
$IPTABLES -A INPUT -p udp  -s 10.1.3.25  -d 10.1.1.1
 --destination-port 161  -m state --state NEW  -j
RULE_18
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.25  -d
10.1.1.6  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.25  -d
1.1.1.105  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.25  -d
10.1.4.3  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.25  -d
10.1.4.2  --destination-port 161  -m state --state
NEW  -j RULE_18
```

```
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.25  -d
10.1.4.254  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p udp  -s 10.1.3.25  -d
10.1.1.6  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p udp  -s 10.1.3.25  -d
1.1.1.105  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p udp  -s 10.1.3.25  -d
10.1.4.3  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p udp  -s 10.1.3.25  -d
10.1.4.2  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p udp  -s 10.1.3.25  -d
10.1.4.254  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A RULE_18   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 18 -- ACCEPT "
$IPTABLES -A RULE_18  -j ACCEPT
```

5.5 Network Address Translation Rules

The firewall performs network address translation on all outbound packets from private addresses. It selectively performs NAT on inbound packets also. Since the DMZ uses RFC 1918 addressing, NAT has to be performed on all packets from or to the DMZ in order for connections to be sustained. If no NAT is performed on an inbound connection attempt, the packet(s) will be sent back out the firewall's public interface and dropped by the border router's anti-spoofing rules. If NAT is performed, the rest of the firewall ruleset will be applied to the packet with its new destination address. Performing NAT on all packets sent to DMZ hosts improves the accuracy of logging, because denied packets will be logged by the firewall for the real reason they were dropped. Inbound syslog packets from the border router are sent to the syslog port on the firewall's public interface. The destination IP address of these packets is translated to that of the central logging server. Outbound NTP queries from the web/portal server and email gateway and DNS queries and SMTP connections from the email gateway are translated to their public IP addresses. All other outbound connections are translated to the firewall's public IP address:

```
#
#  Rule 0(NAT)
#
# NAT outgoing ntp requests from the web/portal
server's 'public' interface to its public internet
address
$IPTABLES -t nat -A POSTROUTING -o eth+ -p tcp  -s
```

```
10.1.4.3  --destination-port 123 -j SNAT --to-source
1.1.2.242
$IPTABLES -t nat -A POSTROUTING -o eth+ -p udp  -s
10.1.4.3  --destination-port 123 -j SNAT --to-source
1.1.2.242
#
#  Rule 1(NAT)
#
# NAT all allowed outgoing requests from email
gateway's 'public' interface to its public ip address
$IPTABLES -t nat -A POSTROUTING -o eth+ -p tcp -m
multiport  -s 10.1.4.2  --destination-ports
25,465,53,123 -j SNAT --to-source 1.1.2.241
$IPTABLES -t nat -A POSTROUTING -o eth+ -p udp -m
multiport  -s 10.1.4.2  --destination-ports 53,123
-j SNAT --to-source 1.1.2.241
#
#  Rule 2(NAT)
#
# We don't do NAT on internal boxes that try to hit
the public addresses of the firewall or DMZ
servers.  They will be able to reach the firewall
(if allowed) without NAT.  The rules below that
handle access to the DMZ servers from inside the
network.
$IPTABLES -t nat -A POSTROUTING -o eth0  -s
10.0.0.0/8 -j SNAT --to-source 1.1.1.106
#
#  Rule 3(NAT)
#
# NAT all incoming packets to the email gateway to
the private ip address on its 'public' interface
$IPTABLES -t nat -A PREROUTING   -d 1.1.2.241 -j
DNAT --to-destination 10.1.4.2
#
#  Rule 4(NAT)
#
# NAT incoming packets to web/portal server to
private ip address of its 'public' interface
$IPTABLES -t nat -A PREROUTING   -d 1.1.2.242 -j
DNAT --to-destination 10.1.4.3
#
#  Rule 5(NAT)
#
# NAT incoming packets to the UDP syslog port on the
firewall from the border router to the syslog server.
$IPTABLES -t nat -A PREROUTING  -p udp  -s 1.1.1.105
 -d 1.1.1.106 --destination-port 514 -j DNAT
--to-destination 10.1.3.11
```

5.6 Logging Rules

By default, all rules get logged with a limit of at most 1 entry for each unique packet type every 10 seconds at level 'info' with the prefix 'Rule [rule number] – [Action].' All packets that are dropped are logged at level 'warn' or higher. Explicitly denied ICMP packets are logged with the custom prefix 'Bad ICMP – Deny.' Denied fragments are logged at level 'alert' with the custom prefix 'FRAG – Deny.' TCP Scan packets are logged with the custom prefix 'TCP Scan – Reject.' IKE, ESP, and AH packets bound for the VPN gateway are logged with the custom prefix 'IPSEC – Accept.' The selected example snippets from the firewall configuration script below display only the logging rule. See B for the full surrounding context.

```
# Drop most types of outbound ICMP.  Exceptions
are echo requests and ICMP unreachables.  Many of
these types will be dropped by the implicit default
DENY, but we want to capture them in a special rule.
There are very few cases in which these packets are
likely to appear for legitimate reasons.  We include
source quench in this group because it can be used for
a DOS.
#
$IPTABLES -N eth0_Out_RULE_0
...
$IPTABLES -A eth0_Out_RULE_0    -m limit --limit 6/minute -j
LOG  --log-level 4 --log-prefix "Bad ICMP -- DENY "
...
#
# Rule 0(eth1)
#
# Allow internet hosts to connect to VPN gateway's
public interface for IPSec.
#
$IPTABLES -N Cid3F4B2699.0
...
$IPTABLES -A eth1_In_RULE_0_3    -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "IPSEC -- ACCEPT"
...
#
# Rule 0(global)
#
# block fragments.  Fragmented packets that are part of
an AH or ESP packet to the VPN gateway are excepted.  (See
eth1 rules).  They are allowed there because the extra
IPSec headers may make the packet too large for our pipes.
#
$IPTABLES -N RULE_0
...
$IPTABLES -A RULE_0   -m limit --limit 6/minute -j
LOG  --log-level 1 --log-prefix "FRAG -- DENY"
```

```
...
#
# Rule 1(global)
#
# Reject Stealth scans with ICMP Port Unreachable to hide
open ports.  Normally, open ports drop bad packets, while
closed ports send a TCP RST. This blocks the destination
response and makes the port look filtered.
#
$IPTABLES -N RULE_1
...
$IPTABLES -A RULE_1   -m limit --limit 6/minute -j
LOG  --log-level 4 --log-prefix "TCP Scan -- REJECT "
...
#
# Rule 2(global)
#
# Unrestricted web access to web/portal server
#
$IPTABLES -N RULE_2
...
$IPTABLES -A RULE_2   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 2 -- ACCEPT "
...
#
# Rule 21(global)
#
# 'catch all' rule
#
$IPTABLES -N RULE_21
$IPTABLES -A RULE_21   -m limit --limit 6/minute -j
LOG  --log-level 4 --log-prefix "RULE 21 -- DENY "
```

5.7 Order of Rules

The firewall configuration script puts the rules accepting packets that are part of or related to established connections first. This prevents rules that do not use stateful inspection from creating unwanted conflict with rules that do. Immediately following are the rules tied to specific interfaces. Since these rules precede all "global" rules, exceptions to global policies can be placed in an interface-specific ruleset. For example, in order to give unfettered access to IP traffic passed back and forth between the loopback interface and the rest of the local system, we put a stateless rule in the ruleset for the loopback interface and any other restrictions on traffic to or from the firewall are ignored in the case of the loopback interface. This also means that fragmented IKE, AH, and ESP packets heading to the VPN gateway's public interface will pass, even though the very first global rule blocks all fragments.

The first two global rules block fragments and TCP scans. These packets are potentially malicious and dangerous; we put the rules forbidding them first to ensure

that packets that would otherwise be allowed through, say to an open port on the web/portal server, are stopped. Next come rules that give access to and from the publicly accessible servers and the Oracle database cluster. We want these packets to be passed quickly, so we keep the rules near the top. We also put rules granting access from the DMZ hosts to internal systems higher up in the list, followed by a specific block rule for any further access from the DMZ to internal hosts. Then come rules for user access to public services, followed by rules that are not used as frequently, such as access to internet NTP servers by our time servers or Technical Support access to perimeter hosts.

Part IV

Appendices

A Border Router Configuration

```
!
! Last configuration change at 13:44:28 EDT Fri Jul 25 2003
! NVRAM config last updated at 09:06:15 EDT Fri Jul 25 2003
!
version 12.2
service timestamps debug uptime
service timestamps log datetime localtime show-timezone
service password-encryption
!
hostname border-router
!
ip cef
no ip http server
no cdp enable
no service tcp-small-servers
no service udp-small-servers
no ip finger
no service finger
no ip bootp server
no boot network
no service config
no ip domain-lookup
service tcp-keepalives-in
!
enable secret 5 $1$Q2vY$d.qenu/qNAujuZPcvOX5D1
username jmoore secret 5 $1$54Ub$ft4fgK45ATQJLCY8FQQgd.
username jmoore privilege 15
username bvogel secret 5 $1$Q4td$vxn8Vdiz2nAnanV7APzvO/
username bvogel privilege 15
aaa new-model
aaa authentication login default local
aaa authorization commands 15 default local
!
banner motd %
********************** WARNING! ***************************
This device is the property of GIAC Enterprises, Incorporated.
Access is restricted to authorized staff members of GIAC Enterprises
Technical Support Department. Unauthorized use will be prosecuted to
the fullest extent of the law. All activity is subject to monitoring
reviewed by GIAC Enterprises IT Security and could be reported to law
enforcement officials to assist in criminal prosecution in the event
of misuse.  If you do not wish to be subject to these conditions,
```

```
disconnect now!
************************************************************
%
!
clock timezone EST -5
clock summer-time EDT recurring 1 Sun Apr 2:00 last Sun Oct 3:00
no ip source-route
!
interface Ethernet0/0
 ip address 1.1.1.105 255.255.255.252
 ip access-group 151 in
 no ip directed-broadcast
 rate-limit input access-group 171 15440 2895 5790
conform-action transmit    exceed-action drop
 rate-limit input access-group 161 154400 28950 57900
conform-action transmit exceed-action drop
!
interface Serial0/0
 description p2p link to ISP
 ip address 1.1.1.102 255.255.255.252
 ip access-group 150 in
 ip access-group 152 out
 no ip directed-broadcast
 no ip proxy-arp
 no ip unreachable
 no ip mask-reply
 no ip redirect
 ip verify unicast reverse-path
 ntp disable
 rate-limit input access-group 170 30880 5790 11580
conform-action transmit    exceed-action drop
 rate-limit input access-group 160 154400 28950 57900
conform-action transmit exceed-action drop
!
interface null0
 no ip unreachables
!
ip classless
ip route 0.0.0.0 0.0.0.0 1.1.1.101
ip route 1.1.1.240 255.255.255.252 1.1.1.106
ip route 1.1.1.159.240 255.255.255.240 1.1.1.106
ip route 0.0.0.0 254.0.0.0 null0
ip route 2.0.0.0 255.0.0.0 null0
ip route 5.0.0.0 255.0.0.0 null0
ip route 7.0.0.0 255.0.0.0 null0
ip route 23.0.0.0 255.0.0.0 null0
ip route 27.0.0.0 255.0.0.0 null0
ip route 36.0.0.0 254.0.0.0 null0
ip route 41.0.0.0 255.0.0.0 null0
ip route 42.0.0.0 255.0.0.0 null0
```

```
ip route 58.0.0.0 254.0.0.0 null0
ip route 70.0.0.0 254.0.0.0 null0
ip route 72.0.0.0 248.0.0.0 null0
ip route 83.0.0.0 255.0.0.0 null0
ip route 84.0.0.0 252.0.0.0 null0
ip route 88.0.0.0 248.0.0.0 null0
ip route 96.0.0.0 224.0.0.0 null0
ip route 169.254.0.0 255.255.0.0 null0
ip route 173.0.0.0 255.0.0.0 null0
ip route 174.0.0.0 254.0.0.0 null0
ip route 176.0.0.0 248.0.0.0 null0
ip route 184.0.0.0 252.0.0.0 null0
ip route 189.0.0.0 255.0.0.0 null0
ip route 190.0.0.0 255.0.0.0 null0
ip route 192.0.2.0 0.0.0.255 null0
ip route 197.0.0.0 255.0.0.0 null0
ip route 198.18.0.0 255.254.0.0 null0
ip route 224.0.0.0 224.0.0.0 null0
ip route 10.0.0.0 255.0.0.0 null0
ip route 192.168.0.0 255.255.0.0 null0
ip route 172.16.0.0 255.16.0.0 null0
!
logging trap notifications
logging source-interface Ethernet0/0
logging 1.1.1.106
access-list 10 permit 1.1.1.106
access-list 10 deny any log
access-list 20 permit 1.1.1.106
access-list 20 permit 1.1.2.241
access-list 20 permit 1.1.2.242
access-list 20 deny any log
access-list 30 permit 1.1.1.106
access-list 30 deny any log
!
access-list 150 deny ip 0.0.0.0 1.255.255.255 any log
access-list 150 deny ip 1.1.1.102 any log
access-list 150 deny ip 1.1.1.104 0.0.0.3 any log
access-list 150 deny ip 1.1.1.240 0.0.0.3 any log
access-list 150 deny ip 1.1.2.240 0.0.0.15 any log
access-list 150 deny ip 2.0.0.0 0.255.255.255 any log
access-list 150 deny ip 5.0.0.0 0.255.255.255 any log
access-list 150 deny ip 7.0.0.0 0.255.255.255 any log
access-list 150 deny ip 23.0.0.0 0.255.255.255 any log
access-list 150 deny ip 27.0.0.0 0.255.255.255 any log
access-list 150 deny ip 36.0.0.0 1.255.255.255 any log
access-list 150 deny ip 41.0.0.0 0.255.255.255 any log
access-list 150 deny ip 42.0.0.0 0.255.255.255 any log
access-list 150 deny ip 58.0.0.0 1.255.255.255 any log
access-list 150 deny ip 70.0.0.0 1.255.255.255 any log
access-list 150 deny ip 72.0.0.0 7.255.255.255 any log
access-list 150 deny ip 83.0.0.0 0.255.255.255 any log
```

```
access-list 150 deny ip 84.0.0.0 3.255.255.255 any log
access-list 150 deny ip 88.0.0.0 7.255.255.255 any log
access-list 150 deny ip 96.0.0.0 31.255.255.255 any log
access-list 150 deny ip 169.254.0.0 0.0.255.255 any log
access-list 150 deny ip 173.0.0.0 0.255.255.255 any log
access-list 150 deny ip 174.0.0.0 1.255.255.255 any log
access-list 150 deny ip 176.0.0.0 7.255.255.255 any log
access-list 150 deny ip 184.0.0.0 3.255.255.255 any log
access-list 150 deny ip 189.0.0.0 0.255.255.255 any log
access-list 150 deny ip 190.0.0.0 0.255.255.255 any log
access-list 150 deny ip 192.0.2.0 0.0.0.255 any log
access-list 150 deny ip 197.0.0.0 0.255.255.255 any log
access-list 150 deny ip 198.18.0.0 0.1.255.255 any log
access-list 150 deny ip 224.0.0.0 31.255.255.255 any log
access-list 150 deny ip 10.0.0.0 0.255.255.255 any log
access-list 150 deny ip 192.168.0.0 0.0.255.255 any log
access-list 150 deny ip 172.16.0.0 0.240.255.255 any log
access-list 150 permit icmp any any echo-reply
access-list 150 permit icmp any any unreachable
access-list 150 permit icmp any any source-quench
access-list 150 permit icmp any any time-exceeded
access-list 150 permit icmp any any parameter-problem
access-list 150 permit 50 any any
access-list 150 permit 51 any any
access-list 150 deny   53 any any log
access-list 150 deny   55 any any log
access-list 150 deny   77 any any log
access-list 150 deny   pim any any log
access-list 150 deny   tcp any host 1.1.1.102 eq telnet log
access-list 150 deny   tcp any host 1.1.1.105 eq telnet log
access-list 150 deny   tcp any host 1.1.1.102 eq 22 log
access-list 150 deny   tcp any host 1.1.1.105 eq 22 log
access-list 150 permit tcp any any
access-list 150 permit udp any any
access-list 150 deny ip any any log
!
access-list 151 deny   53 any any log
access-list 151 deny   55 any any log
access-list 151 deny   77 any any log
access-list 151 deny   pim any any log
access-list 151 permit tcp host 1.1.1.106 host 1.1.1.102 eq 23
access-list 151 permit ip 1.1.1.106 any
access-list 151 permit ip 1.1.1.240 0.0.0.3 any
access-list 151 permit ip 1.1.2.240 0.0.0.15 any
access-list 151 deny ip any any log
!
access-list 152 permit icmp any any echo-request
access-list 152 permit icmp 1.1.1.242 any packet-too-big
access-list 152 deny icmp any any
access-list 152 deny ip 1.1.1.102 any
access-list 152 deny ip 1.1.1.105 any
```

```
access-list 152 permit ip any any
!
access=list 160 permit tcp any 1.1.1.241 syn
access=list 160 permit tcp any 1.1.1.242 syn
!
access-list 161 permit tcp any any syn
!
access-list 170 permit icmp any any echo-reply!
!
access-list 171 permit icmp any any echo-request
!
snmp-server engineID local 00000009020000D058201560
snmp-server view admin .1 included
snmp-server view nata .1 excluded
snmp-server group admin v3 priv read admin write nata notify nata access 30
snmp-server group notify v3 priv read nata write nata notify admin access 30
snmp-server user bigbrother admin v3 encrypted auth md5 halleyberrton
snmp-server user notifier notify v3 encrypted auth md5 d0ntphenceME1n
snmp-server location GIAC Enterprises Data Center
snmp-server contact GIAC Enterprises Technical Support, ext. 7785
snmp-server enable traps snmp
snmp-server enable traps config
snmp-server enable traps entity
snmp-server enable traps envmon
snmp-server enable traps syslog
snmp-server enable traps frame-relay
snmp-server enable traps rtr
snmp-server host 1.1.1.106 version 3 auth notifier
snmp-server enable traps envmon
snmp-server enable traps syslog
snmp-server enable traps frame-relay
snmp-server enable traps rtr
snmp-server host 1.1.1.106 version 3 auth notifier
!
line con 0
 exec-timeout 5 0
 transport input telnet
 aaa authentication login default
 authorization commands 15 defaultline con 0
line vty 0 4
 access-class 10 in
 session-timeout 5
 transport input telnet
 aaa authentication login default
 authorization commands 15 default
line aux 0
 transport input none
!
ntp clock-period 17208264
ntp source Ethernet0/0
```

```
ntp server 1.1.1.241
ntp server 1.1.1.106 prefer
no scheduler allocate
end
```

B Internet Firewall Configuration

The Netfilter firewall script below was generated by the fwb_ipt program, part of the Firewall Builder multi-platform firewall ruleset manager. Comments embedded in the script were mostly added to comment fields in the fwbuilder GUI; a few, including the file header, were generated by the iptables ruleset compiler, fwb_ipt. The generated script is stored in /etc/sysconfig/firewall.fw and called by init script /etc/rc.d/init.d/firewall. Long lines in the original file have been split to make it easier to read.

```
#!/bin/sh
#
 #  This is automatically generated file. DO NOT
MODIFY !
#
#  Firewall Builder  fwb_ipt v1.0.10-1
#
#  Generated Mon Sep 15 06:35:52 2003 EDT by jmoore
#
#
#
#
log() {
  test -x "$LOGGER" && $LOGGER -p info "$1"
}
va_num=1
add_addr() {
  addr=$1
  nm=$2
  dev=$3
  type=""
  aadd=""
  L=`$IP -4 addr ls $dev | grep "$dev:"`
  if test -n "$L"; then
    OIFS=$IFS
    IFS=" /:,<"
    set $L
    type=$4
    IFS=$OIFS
    L=`$IP -4 addr ls $dev to $addr | grep " inet "`
    if test -n "$L"; then
      OIFS=$IFS
      IFS=" /"
      set $L
      aadd=$2
      IFS=$OIFS
    fi
  fi
  if test -z "$aadd"; then
    if test "$type" = "POINTOPOINT"; then
      $IP -4 addr add $addr dev $dev scope global
label $dev:FWB${va_num}
      va_num=`expr $va_num + 1`
```

```
    fi
    if test "$type" = "BROADCAST"; then
      $IP -4 addr add $addr/$nm dev $dev brd + scope
global label $dev:FWB${va_num}
      va_num=`expr $va_num + 1`
    fi
  fi
}
getaddr() {
  dev=$1
  name=$2
  L=`$IP -4 addr show dev $dev | grep inet`
  test "Z$L" == "Z" && {
    echo "Interface $dev is down, its IP address is
unknown. Can not install firewall policy."
    exit 1
  }
  OIFS=$IFS
  IFS=" /"
  set $L
  eval "$name=$2"
  IFS=$OIFS
}
LSMOD="/sbin/lsmod"
MODPROBE="/sbin/modprobe"
IPTABLES="/sbin/iptables"
IP="/sbin/ip"
LOGGER="/usr/bin/logger"
cd /etc || exit 1
log "Activating firewall script generated Mon Sep 15
06:35:52 2003 EDT by jmoore"
INTERFACES="eth0 eth1 eth2 eth3 eth4 eth5 lo "
for i in $INTERFACES ; do
  $IP link show "$i" > /dev/null 2>&1 || {
    echo Interface $i does not exist
    exit 1
  }
done
echo 1 > /proc/sys/net/ipv4/ip_dynaddr
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/all/log_martians
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout
echo 1800 > /proc/sys/net/ipv4/tcp_keepalive_intvl
echo 1 > /proc/sys/net/ipv4/tcp_window_scaling
echo 1 > /proc/sys/net/ipv4/tcp_sack
echo 1 > /proc/sys/net/ipv4/tcp_fack
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
echo 1 > /proc/sys/net/ipv4/tcp_ecn
echo 1 > /proc/sys/net/ipv4/tcp_timestamps
```

```
$IP -4 neigh flush dev eth0
$IP -4 addr flush dev eth0 label "eth0:FWB*"
$IP -4 neigh flush dev eth1
$IP -4 addr flush dev eth1 label "eth1:FWB*"
$IP -4 neigh flush dev eth2
$IP -4 addr flush dev eth2 label "eth2:FWB*"
$IP -4 neigh flush dev eth3
$IP -4 addr flush dev eth3 label "eth3:FWB*"
$IP -4 neigh flush dev eth4
$IP -4 addr flush dev eth4 label "eth4:FWB*"
$IP -4 neigh flush dev eth5
$IP -4 addr flush dev eth5 label "eth5:FWB*"
add_addr 1.1.1.106 30 eth0
$IP link set eth0 up
add_addr 1.1.1.241 30 eth1
$IP link set eth1 up
add_addr 10.1.1.5 30 eth2
$IP link set eth2 up
add_addr 10.1.4.1 24 eth3
$IP link set eth3 up
add_addr 10.1.1.1 30 eth4
$IP link set eth4 up
add_addr 10.1.6.2 24 eth5
$IP link set eth5 up
add_addr 127.0.0.1 8 lo
$IP link set lo up
$IPTABLES -P OUTPUT  DROP
$IPTABLES -P INPUT   DROP
$IPTABLES -P FORWARD DROP
cat /proc/net/ip_tables_names | while read table; do
  $IPTABLES -t $table -L -n | while read c chain rest; do
      if test "X$c" = "XChain" ; then
        $IPTABLES -t $table -F $chain
      fi
  done
  $IPTABLES -t $table -X
done
MODULE_DIR="/lib/modules/‘uname -r‘/kernel/net/ipv4/netfilter/"
MODULES=‘(cd $MODULE_DIR; ls *_conntrack_*  *_nat_* |
sed 's/\.o.*$//')‘
for module in $(echo $MODULES); do
  if $LSMOD | grep ${module} >/dev/null; then continue; fi
  if [ -e "${MODULE_DIR}/${module}.o" -o -e
"${MODULE_DIR}/${module}.o.gz" ]; then
    $MODPROBE ${module} ||  exit 1
  fi
done
#
#  Rule 0(NAT)
#
# NAT outgoing ntp requests from the web/portal
server's 'public' interface to its public internet
address
```

```
$IPTABLES -t nat -A POSTROUTING -o eth+ -p tcp  -s
10.1.4.3  --destination-port 123 -j SNAT --to-source
1.1.2.242
$IPTABLES -t nat -A POSTROUTING -o eth+ -p udp  -s
10.1.4.3  --destination-port 123 -j SNAT --to-source
1.1.2.242
#
#  Rule 1(NAT)
#
# NAT all allowed outgoing requests from email
gateway's 'public' interface to its public ip address
$IPTABLES -t nat -A POSTROUTING -o eth+ -p tcp -m
multiport  -s 10.1.4.2  --destination-ports
25,465,53,123 -j SNAT --to-source 1.1.2.241
$IPTABLES -t nat -A POSTROUTING -o eth+ -p udp -m
multiport  -s 10.1.4.2  --destination-ports 53,123
-j SNAT --to-source 1.1.2.241
#
#  Rule 2(NAT)
#
# We don't do NAT on internal boxes that try to hit
the public addresses of the firewall or DMZ
servers.  They will be able to reach the firewall
(if allowed) without NAT.  The rules below that
handle access to the DMZ servers from inside the
network.
$IPTABLES -t nat -A POSTROUTING -o eth0  -s
10.0.0.0/8 -j SNAT --to-source 1.1.1.106
#
#  Rule 3(NAT)
#
# NAT all incoming packets to the email gateway to
the private ip address on its 'public' interface
$IPTABLES -t nat -A PREROUTING   -d 1.1.2.241 -j
DNAT --to-destination 10.1.4.2
#
#  Rule 4(NAT)
#
# NAT incoming packets to web/portal server to
private ip address of its 'public' interface
$IPTABLES -t nat -A PREROUTING   -d 1.1.2.242 -j
DNAT --to-destination 10.1.4.3
#
#  Rule 5(NAT)
#
# NAT incoming packets to the UDP syslog port on the
firewall from the border router to the syslog server.
$IPTABLES -t nat -A PREROUTING  -p udp  -s 1.1.1.105
 -d 1.1.1.106 --destination-port 514 -j DNAT
```

```
--to-destination 10.1.3.11
#
#
$IPTABLES -t drop -A DROPPING -j LOG  --log-level 6
--log-prefix "RULE %N -- %A "
$IPTABLES -A INPUT   -m state --state
ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A OUTPUT  -m state --state
ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -m state --state
ESTABLISHED,RELATED -j ACCEPT
#
# Rule 0(eth0)
#
# Drop most types of outbound ICMP.  Exceptions are
echo requests and ICMP unreachables.  Many of these
types will be dropped by the implicit default DENY,
but we want to capture them in a special rule.
There are very few cases in which these packets are
likely to appear for legitimate reasons.  We include
source quench in this group because it can be used
for a DOS.
#
$IPTABLES -N eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
11/1   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
11/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
5/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
5/1   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
5/2   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
5/3   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
13/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
15/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
14/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
16/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
9/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
10/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
```

```
37/0   -j eth0_Out_RULE_0
$IPTABLES -A OUTPUT  -o eth0 -p icmp  --icmp-type
38/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
11/1   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
11/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
5/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
5/1   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
5/2   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
5/3   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
13/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
15/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
14/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
16/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
9/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
10/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
37/0   -j eth0_Out_RULE_0
$IPTABLES -A FORWARD  -o eth0 -p icmp  --icmp-type
38/0   -j eth0_Out_RULE_0
$IPTABLES -A eth0_Out_RULE_0    -m limit --limit
6/minute -j LOG  --log-level 4 --log-prefix "Bad
ICMP -- DENY "
$IPTABLES -A eth0_Out_RULE_0   -j DROP
#
# Rule 1(eth0)
#
# Drop most types of inbound ICMP.  Exceptions are
unreachables and echo replies.  These are covered by
the generic rule allowing ESTABLISHED and RELATED
traffic back into the network.  We don't
specifically mention echo requests because we are
going to allow them to the VPN gateway public
interface, and they won't make it there if they are
blocked here.
#
$IPTABLES -N eth0_In_RULE_1
```

```
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
4/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
5/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
5/1   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
5/2   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
5/3   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
13/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
14/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
15/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
16/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
9/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
10/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
37/0   -j eth0_In_RULE_1
$IPTABLES -A INPUT  -i eth0 -p icmp  --icmp-type
38/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
4/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
5/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
5/1   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
5/2   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
5/3   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
13/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
14/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
15/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
16/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
9/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
10/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
```

```
37/0   -j eth0_In_RULE_1
$IPTABLES -A FORWARD  -i eth0 -p icmp  --icmp-type
38/0   -j eth0_In_RULE_1
$IPTABLES -A eth0_In_RULE_1    -m limit --limit
6/minute -j LOG  --log-level 4 --log-prefix "BAD
ICMP -- DENY "
$IPTABLES -A eth0_In_RULE_1   -j DROP
#
# Rule 0(eth1)
#
# Allow internet hosts to connect to VPN gateway's public interface for IPSec.
#
$IPTABLES -N Cid3F4B2699.0
$IPTABLES -A FORWARD  -i eth1 -p udp  -d 1.1.1.242
 --destination-port 500  -m state --state NEW  -j
Cid3F4B2699.0
$IPTABLES -A FORWARD  -i eth1 -p 50  -d 1.1.1.242
-f   -m state --state NEW  -j Cid3F4B2699.0
$IPTABLES -A FORWARD  -i eth1 -p 51  -d 1.1.1.242
-f   -m state --state NEW  -j Cid3F4B2699.0
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 1.1.1.241
-j RETURN
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 1.1.1.106
-j RETURN
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 10.0.0.0/8
-j RETURN
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 1.1.1.242
-j RETURN
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 1.1.2.242
-j RETURN
$IPTABLES -A Cid3F4B2699.0  -i eth1  -s 1.1.2.241
-j RETURN
$IPTABLES -N eth1_In_RULE_0_3
$IPTABLES -A Cid3F4B2699.0  -i eth1  -m state
--state NEW  -j eth1_In_RULE_0_3
$IPTABLES -A eth1_In_RULE_0_3    -m limit --limit
6/minute -j LOG  --log-level 6 --log-prefix "IPSEC
-- ACCEPT"
$IPTABLES -A eth1_In_RULE_0_3   -j ACCEPT
$IPTABLES -N Cid3F4B2699.1
$IPTABLES -A OUTPUT  -o eth1 -p udp  -d 1.1.1.242
 --destination-port 500  -m state --state NEW  -j
Cid3F4B2699.1
$IPTABLES -A OUTPUT  -o eth1 -p 50  -d 1.1.1.242
-f   -m state --state NEW  -j Cid3F4B2699.1
$IPTABLES -A OUTPUT  -o eth1 -p 51  -d 1.1.1.242  -f
  -m state --state NEW
-j Cid3F4B2699.1
$IPTABLES -A FORWARD  -o eth1 -p udp  -d 1.1.1.242
```

```
--destination-port 500  -m
state --state NEW  -j Cid3F4B2699.1
$IPTABLES -A FORWARD  -o eth1 -p 50  -d 1.1.1.242
-f   -m state --state NEW  -j Cid3F4B2699.1
$IPTABLES -A FORWARD  -o eth1 -p 51  -d 1.1.1.242
-f   -m state --state NEW  -j Cid3F4B2699.1
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 1.1.1.241
-j RETURN
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 1.1.1.106
-j RETURN
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 10.0.0.0/8
-j RETURN
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 1.1.1.242
-j RETURN
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 1.1.2.242
-j RETURN
$IPTABLES -A Cid3F4B2699.1  -o eth1  -s 1.1.2.241
-j RETURN
$IPTABLES -N eth1_Out_RULE_0_3
$IPTABLES -A Cid3F4B2699.1  -o eth1  -m state
--state NEW  -j eth1_Out_RULE_0_3
$IPTABLES -A eth1_Out_RULE_0_3    -m limit --limit
6/minute -j LOG  --log-level 6 --log-prefix "IPSEC
-- ACCEPT"
$IPTABLES -A eth1_Out_RULE_0_3   -j ACCEPT
#
# Rule 1(eth1)
#
# Allow the public interface of the VPN gateway to
connect with other internet hosts for IPSec.
Currently, the gateway does not actually do this,
except to initiate keying negotiations.
#
$IPTABLES -N Cid3F4C0423.0
$IPTABLES -A INPUT  -i eth1 -p udp  -s 1.1.1.242
--destination-port 500  -m state --state NEW  -j
Cid3F4C0423.0
$IPTABLES -A INPUT  -i eth1 -p 50  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.0
$IPTABLES -A INPUT  -i eth1 -p 51  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.0
$IPTABLES -A FORWARD  -i eth1 -p udp  -s 1.1.1.242
--destination-port 500  -m state --state NEW  -j
Cid3F4C0423.0
$IPTABLES -A FORWARD  -i eth1 -p 50  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.0
$IPTABLES -A FORWARD  -i eth1 -p 51  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.0
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 1.1.1.106
```

```
-j RETURN
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 1.1.1.241
-j RETURN
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 1.1.2.241
-j RETURN
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 1.1.2.242
-j RETURN
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 1.1.1.242
-j RETURN
$IPTABLES -A Cid3F4C0423.0  -i eth1  -d 10.0.0.0/8
-j RETURN
$IPTABLES -N eth1_In_RULE_1_3
$IPTABLES -A Cid3F4C0423.0  -i eth1  -m state
--state NEW  -j eth1_In_RULE_1_3
$IPTABLES -A eth1_In_RULE_1_3    -m limit --limit
6/minute -j LOG  --log-level 6 --log-prefix "IPSEC
-- ACCEPT "
$IPTABLES -A eth1_In_RULE_1_3   -j ACCEPT
$IPTABLES -N Cid3F4C0423.1
$IPTABLES -A FORWARD  -o eth1 -p udp  -s 1.1.1.242
--destination-port 500  -m state --state NEW  -j
Cid3F4C0423.1
$IPTABLES -A FORWARD  -o eth1 -p 50  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.1
$IPTABLES -A FORWARD  -o eth1 -p 51  -s 1.1.1.242
-f   -m state --state NEW  -j Cid3F4C0423.1
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 1.1.1.106
-j RETURN
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 1.1.1.241
-j RETURN
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 1.1.2.241
-j RETURN
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 1.1.2.242
-j RETURN
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 1.1.1.242
-j RETURN
$IPTABLES -A Cid3F4C0423.1  -o eth1  -d 10.0.0.0/8
-j RETURN
$IPTABLES -N eth1_Out_RULE_1_3
$IPTABLES -A Cid3F4C0423.1  -o eth1  -m state
--state NEW  -j eth1_Out_RULE_1_3
$IPTABLES -A eth1_Out_RULE_1_3    -m limit --limit
6/minute -j LOG  --log-level 6 --log-prefix "IPSEC
-- ACCEPT "
$IPTABLES -A eth1_Out_RULE_1_3   -j ACCEPT
#
# Rule 2(eth1)
#
# Allow internet hosts to ping the VPN gateway.  We
```

```
allow this for testing VPN connectivity issues.
Rate limiting applies to the ping requests.  We
allow replies to go back, but the types allowed are
limited by the internet firewall.
#
$IPTABLES -N eth1_In_RULE_2
$IPTABLES -A FORWARD  -i eth1 -p icmp  -d 1.1.1.242
--icmp-type 8/0   -m state --state NEW  -j
eth1_In_RULE_2
$IPTABLES -A eth1_In_RULE_2    -m limit --limit
3/second --limit-burst 5 -j LOG  --log-level 6
--log-prefix "RULE 2 -- ACCEPT "
$IPTABLES -A eth1_In_RULE_2    -m limit --limit
3/second --limit-burst 5 -j ACCEPT
$IPTABLES -N eth1_Out_RULE_2
$IPTABLES -A OUTPUT  -o eth1 -p icmp  -d 1.1.1.242
--icmp-type 8/0   -m state --state NEW  -j
eth1_Out_RULE_2
$IPTABLES -A FORWARD  -o eth1 -p icmp  -d 1.1.1.242
--icmp-type 8/0   -m state --state NEW  -j
eth1_Out_RULE_2
$IPTABLES -A eth1_Out_RULE_2    -m limit --limit
3/second --limit-burst 5 -j LOG  --log-level 6
--log-prefix "RULE 2 -- ACCEPT "
$IPTABLES -A eth1_Out_RULE_2    -m limit --limit
3/second --limit-burst 5 -j ACCEPT
#
# Rule 0(lo)
#
# allow everything on loopback
#
$IPTABLES -N lo_In_RULE_0
$IPTABLES -A INPUT  -i lo  -j lo_In_RULE_0
$IPTABLES -A lo_In_RULE_0    -m limit --limit
6/minute -j LOG  --log-level 6 --log-prefix "RULE 0
-- ACCEPT "
$IPTABLES -A lo_In_RULE_0   -j ACCEPT
$IPTABLES -N lo_Out_RULE_0
$IPTABLES -A OUTPUT  -o lo  -j lo_Out_RULE_0
$IPTABLES -A lo_Out_RULE_0    -m limit --limit
6/minute -j LOG  --log-level 6 --log-prefix "RULE 0
-- ACCEPT "
$IPTABLES -A lo_Out_RULE_0   -j ACCEPT
#
# Rule 0(global)
#
# block fragments.  Fragmented packets that are part
of an AH or ESP packet to the VPN gateway are
excepted.  (See eth1 rules).  They are allowed there
```

```
because the extra IPSec headers may make the packet
too large for our pipes.
#
$IPTABLES -N RULE_0
$IPTABLES -A OUTPUT -p ip  -f   -j RULE_0
$IPTABLES -A INPUT -p ip  -f   -j RULE_0
$IPTABLES -A FORWARD -p ip  -f   -j RULE_0
$IPTABLES -A RULE_0   -m limit --limit 6/minute -j
LOG  --log-level 1 --log-prefix "FRAG -- DENY"
$IPTABLES -A RULE_0  -j DROP
#
# Rule 1(global)
#
# Drop Stealth scans.  Normally, open ports drop bad packets,
while closed ports send a TCP RST. This blocks the destination
response and makes all ports appear open.
#
$IPTABLES -N RULE_1
$IPTABLES -A OUTPUT -p tcp  --tcp-flags ALL NONE  -j RULE_1
$IPTABLES -A OUTPUT -p tcp  --tcp-flags URG,PSH,FIN URG,PSH,FIN  -j RULE_1
$IPTABLES -A OUTPUT --protocol tcp --tcp-flags SYN,ACK,RST,FIN FIN
-m state --state NEW  -j RULE_1
$IPTABLES -A OUTPUT --protocol tcp --tcp-flags ACK ACK -m state
--state NEW  -j RULE_1
$IPTABLES -A INPUT -p tcp  --tcp-flags ALL NONE  -j RULE_1
$IPTABLES -A INPUT -p tcp  --tcp-flags ALL URG,ACK,PSH,RST,SYN,FIN
-j RULE_1
$IPTABLES -A INPUT --protocol tcp --tcp-flags SYN,ACK,RST,FIN FIN
-m state --state NEW  -j RULE_1
$IPTABLES -A INPUT --protocol tcp --tcp-flags ACK ACK -m state
--state NEW  -j RULE_1
$IPTABLES -A FORWARD -p tcp  --tcp-flags ALL NONE  -j RULE_1
$IPTABLES -A FORWARD -p tcp  --tcp-flags URG,PSH,FIN URG,PSH,FIN  -j RULE_1
$IPTABLES -A FORWARD --protocol tcp --tcp-flags SYN,ACK,RST,FIN FIN
-m state --state NEW  -j RULE_1
$IPTABLES -A FORWARD --protocol tcp --tcp-flags ACK ACK -m state
--state NEW  -j RULE_1
$IPTABLES -A RULE_1   -m limit --limit 6/minute -j LOG  --log-level 4
--log-prefix "TCP Scan -- DENY "
$IPTABLES -A RULE_1  -j DROP
#
# Rule 2(global)
#
# Unrestricted web access to web/portal server
#
$IPTABLES -N RULE_2
$IPTABLES -A OUTPUT -p tcp  -m multiport  -d
10.1.4.3  --destination-ports 80,443  -m state
--state NEW  -j RULE_2
$IPTABLES -A FORWARD -p tcp  -m multiport  -d
10.1.4.3  --destination-ports 80,443  -m state
```

```
--state NEW  -j RULE_2
$IPTABLES -A RULE_2   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 2 --
ACCEPT "
$IPTABLES -A RULE_2  -j ACCEPT
#
# Rule 3(global)
#
# Unrestricted SMTP access to email gateway
#
$IPTABLES -N RULE_3
$IPTABLES -A OUTPUT -p tcp  -m multiport  -d
10.1.4.2  --destination-ports 25,465  -m state
--state NEW  -j RULE_3
$IPTABLES -A FORWARD -p tcp  -m multiport  -d
10.1.4.2  --destination-ports 25,465  -m state
--state NEW  -j RULE_3
$IPTABLES -A RULE_3   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 3 -- ACCEPT "
$IPTABLES -A RULE_3  -j ACCEPT
#
# Rule 4(global)
#
# Email gateway allowed access to forward mail,
except to firewall
#
$IPTABLES -N Cid3F4B29D6.0
$IPTABLES -A INPUT -p tcp  -m multiport  -s 10.1.4.2
 --destination-ports 25,465,53  -m state --state NEW
 -j Cid3F4B29D6.0
$IPTABLES -A INPUT -p udp  -s 10.1.4.2
--destination-port 53  -m state --state NEW  -j
Cid3F4B29D6.0
$IPTABLES -A FORWARD -p tcp  -m multiport  -s
10.1.4.2  --destination-ports 25,465,53  -m state
--state NEW  -j Cid3F4B29D6.0
$IPTABLES -A FORWARD -p udp  -s 10.1.4.2
--destination-port 53  -m state --state NEW  -j
Cid3F4B29D6.0
$IPTABLES -A Cid3F4B29D6.0  -d 1.1.1.106  -j RETURN
$IPTABLES -A Cid3F4B29D6.0  -d 1.1.1.241  -j RETURN
$IPTABLES -A Cid3F4B29D6.0  -d 10.1.1.5  -j RETURN
$IPTABLES -A Cid3F4B29D6.0  -d 10.1.4.1  -j RETURN
$IPTABLES -A Cid3F4B29D6.0  -d 10.1.1.1  -j RETURN
$IPTABLES -A Cid3F4B29D6.0  -d 10.1.6.2  -j RETURN
$IPTABLES -N RULE_4_3
$IPTABLES -A Cid3F4B29D6.0  -m state --state NEW
-j RULE_4_3
$IPTABLES -A RULE_4_3   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 4 -- ACCEPT "
```

```
$IPTABLES -A RULE_4_3  -j ACCEPT
#
# Rule 5(global)
#
# Oracle Database Server allowed to contact
Heartland Payment Systems Secure Processor via
TLS-encrypted channel.
#
$IPTABLES -N RULE_5
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.10  -d
1.100.100.1  --destination-port 4999  -j RULE_5
$IPTABLES -A RULE_5   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 5 -- DENY "
$IPTABLES -A RULE_5  -j DROP
#
# Rule 6(global)
#
# Internal DNS master can forward outside queries to
caching-only DNS server on email gateway.  It can
also forward zone update notifications. Bastion
hosts use email gateway as their primary DNS server
as well.
#
$IPTABLES -N RULE_6
$IPTABLES -A OUTPUT -p tcp  -s 10.1.1.1  -d 10.1.4.2
 --destination-port 53  -m state --state NEW  -j
RULE_6
$IPTABLES -A OUTPUT -p udp  -s 10.1.1.1  -d 10.1.4.2
 --destination-port 53  -m state --state NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.5.13  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.6.15  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.1.6  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 1.1.1.105  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.3  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.2  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.254  -d
```

```
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.5.13  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.6.15  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.1.6  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 1.1.1.105  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.4.3  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.4.2  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A FORWARD -p udp  -s 10.1.4.254  -d
10.1.4.2  --destination-port 53  -m state --state
NEW  -j RULE_6
$IPTABLES -A RULE_6   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 6 -- ACCEPT "
$IPTABLES -A RULE_6  -j ACCEPT
#
# Rule 7(global)
#
# All hosts get time from these servers
#
$IPTABLES -N RULE_7
$IPTABLES -A INPUT -p tcp  -s 10.0.0.0/8  -d
1.1.1.106  --destination-port 123  -m state
--state NEW  -j RULE_7
$IPTABLES -A INPUT -p tcp  -s 1.1.1.105  -d
1.1.1.106  --destination-port 123  -m state
--state NEW  -j RULE_7
$IPTABLES -A INPUT -p udp  -s 10.0.0.0/8  -d
1.1.1.106  --destination-port 123  -m state
--state NEW  -j RULE_7
$IPTABLES -A INPUT -p udp  -s 1.1.1.105  -d
1.1.1.106  --destination-port 123  -m state
--state NEW  -j RULE_7
$IPTABLES -N Cid3F4B36BD.0
$IPTABLES -A OUTPUT  -s 10.0.0.0/8  -m state
--state NEW  -j Cid3F4B36BD.0
$IPTABLES -A OUTPUT  -s 1.1.1.105  -m state --state
```

```
NEW  -j Cid3F4B36BD.0
$IPTABLES -A Cid3F4B36BD.0 -p tcp  -d 10.1.4.2
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -A Cid3F4B36BD.0 -p tcp  -d 10.1.4.3
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -N Cid3F4B36BD.1
$IPTABLES -A OUTPUT  -s 10.0.0.0/8  -m state --state
NEW  -j Cid3F4B36BD.1
$IPTABLES -A OUTPUT  -s 1.1.1.105  -m state --state
NEW  -j Cid3F4B36BD.1
$IPTABLES -A Cid3F4B36BD.1 -p udp  -d 10.1.4.2
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -A Cid3F4B36BD.1 -p udp  -d 10.1.4.3
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -N Cid3F4B36BD.2
$IPTABLES -A FORWARD  -s 10.0.0.0/8  -m state
--state NEW  -j Cid3F4B36BD.2
$IPTABLES -A FORWARD  -s 1.1.1.105  -m state
--state NEW  -j Cid3F4B36BD.2
$IPTABLES -A Cid3F4B36BD.2 -p tcp  -d 10.1.4.2
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -A Cid3F4B36BD.2 -p tcp  -d 10.1.4.3
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -N Cid3F4B36BD.3
$IPTABLES -A FORWARD  -s 10.0.0.0/8  -m state
--state NEW  -j Cid3F4B36BD.3
$IPTABLES -A FORWARD  -s 1.1.1.105  -m state
--state NEW  -j Cid3F4B36BD.3
$IPTABLES -A Cid3F4B36BD.3 -p udp  -d 10.1.4.2
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -A Cid3F4B36BD.3 -p udp  -d 10.1.4.3
--destination-port 123  -m state --state NEW  -j
RULE_7
$IPTABLES -A RULE_7   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 7 -- ACCEPT "
$IPTABLES -A RULE_7  -j ACCEPT
#
# Rule 8(global)
#
# bastion hosts allowed to use internal DNS master
as secondary DNS.  Email gateway, as slave server to
```

```
internal domain, allowed to contact master for zone
transfers.
#
$IPTABLES -N RULE_8
$IPTABLES -A OUTPUT -p tcp  -s 10.1.1.1  -d
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A OUTPUT -p tcp  -s 10.1.1.1  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A OUTPUT -p udp  -s 10.1.1.1  -d
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A OUTPUT -p udp  -s 10.1.1.1  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p tcp  -s 10.1.1.6  -d
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p tcp  -s 10.1.1.6  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p tcp  -s 1.1.1.105  -d
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p tcp  -s 1.1.1.105  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.3  -d
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.3  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.2  -d
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.2  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.254  -d
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.254  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p udp  -s 10.1.1.6  -d
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p udp  -s 10.1.1.6  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p udp  -s 1.1.1.105  -d
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p udp  -s 1.1.1.105  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p udp  -s 10.1.4.3  -d
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p udp  -s 10.1.4.3  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p udp  -s 10.1.4.2  -d
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p udp  -s 10.1.4.2  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p udp  -s 10.1.4.254  -d
```

```
10.1.5.13  --destination-port 53  -j RULE_8
$IPTABLES -A FORWARD -p udp  -s 10.1.4.254  -d
10.1.6.15  --destination-port 53  -j RULE_8
$IPTABLES -A RULE_8   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 8 -- DENY "
$IPTABLES -A RULE_8  -j DROP
#
# Rule 9(global)
#
# Allow SNMP traps/informs back to Management station
#
$IPTABLES -N RULE_9
$IPTABLES -A OUTPUT -p tcp  -s 10.1.1.1  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A OUTPUT -p udp  -s 10.1.1.1  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 10.1.1.6  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 1.1.1.105  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.3  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.2  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 10.1.4.254  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p udp  -s 10.1.1.6  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p udp  -s 1.1.1.105  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p udp  -s 10.1.4.3  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p udp  -s 10.1.4.2  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
$IPTABLES -A FORWARD -p udp  -s 10.1.4.254  -d
10.1.3.25  --destination-port 162  -m state --state
NEW  -j RULE_9
```

```
$IPTABLES -A RULE_9   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 9 -- ACCEPT "
$IPTABLES -A RULE_9  -j ACCEPT
#
# Rule 10(global)
#
# blocking access to internal net
# and firewall from DMZ.
# Rules that permit access to servers
# on DMZ should be added above.
#
$IPTABLES -N Cid3F4B2716.0
$IPTABLES -A INPUT  -s 10.1.6.6  -j Cid3F4B2716.0
$IPTABLES -A INPUT  -s 10.1.4.3  -j Cid3F4B2716.0
$IPTABLES -A INPUT  -s 10.1.4.2  -j Cid3F4B2716.0
$IPTABLES -A INPUT  -s 10.1.6.5  -j Cid3F4B2716.0
$IPTABLES -N RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 1.1.1.106  -j RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 1.1.1.241  -j RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 10.1.1.5  -j RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 10.1.4.1  -j RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 10.1.1.1  -j RULE_10
$IPTABLES -A Cid3F4B2716.0  -d 10.1.6.2  -j RULE_10
$IPTABLES -A INPUT  -s 10.1.6.6  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A INPUT  -s 10.1.4.3  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A INPUT  -s 10.1.4.2  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A INPUT  -s 10.1.6.5  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A FORWARD  -s 10.1.6.6  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A FORWARD  -s 10.1.4.3  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A FORWARD  -s 10.1.4.2  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A FORWARD  -s 10.1.6.5  -d 10.0.0.0/8  -j
RULE_10
$IPTABLES -A RULE_10   -m limit --limit 6/minute -j
LOG  --log-level 4 --log-prefix "RULE 10 -- DENY "
$IPTABLES -A RULE_10  -j DROP
#
# Rule 11(global)
#
# Web Proxy does all direct connection to internet
web/ftp servers
#
$IPTABLES -N Cid3F4B36CD.0
$IPTABLES -A INPUT -p tcp  -m multiport  -s
10.1.5.14  --destination-ports 21,20,80,443  -m
```

```
state --state NEW  -j Cid3F4B36CD.0
$IPTABLES -A FORWARD -p tcp  -m multiport  -s
10.1.5.14  --destination-ports 21,20,80,443  -m
state --state NEW  -j Cid3F4B36CD.0
$IPTABLES -A Cid3F4B36CD.0  -d 1.1.1.106  -j RETURN
$IPTABLES -A Cid3F4B36CD.0  -d 1.1.1.241  -j RETURN
$IPTABLES -A Cid3F4B36CD.0  -d 10.1.1.5  -j RETURN
$IPTABLES -A Cid3F4B36CD.0  -d 10.1.4.1  -j RETURN
$IPTABLES -A Cid3F4B36CD.0  -d 10.1.1.1  -j RETURN
$IPTABLES -A Cid3F4B36CD.0  -d 10.1.6.2  -j RETURN
$IPTABLES -N RULE_11_3
$IPTABLES -A Cid3F4B36CD.0  -m state --state NEW  -j
RULE_11_3
$IPTABLES -A RULE_11_3   -m limit --limit 6/minute
-j LOG  --log-level 6 --log-prefix "RULE 11 -- ACCEPT "
$IPTABLES -A RULE_11_3  -j ACCEPT
#
# Rule 12(global)
#
# allow troubleshooting anything from internal hosts
#
$IPTABLES -N RULE_12
$IPTABLES -A INPUT -p icmp  -s 10.0.0.0/8
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A INPUT -p icmp  -s 1.1.1.242
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A INPUT -p icmp  -s 10.1.1.6
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A INPUT -p icmp  -s 10.1.6.4
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A INPUT -p tcp  -s 10.0.0.0/8
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
$IPTABLES -A INPUT -p tcp  -s 1.1.1.242
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
$IPTABLES -A INPUT -p tcp  -s 10.1.1.6
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
$IPTABLES -A INPUT -p tcp  -s 10.1.6.4
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
$IPTABLES -A INPUT -p udp  -s 10.0.0.0/8
--destination-port 33434:33524  -m state --state
NEW  -j RULE_12
$IPTABLES -A INPUT -p udp  -s 1.1.1.242
--destination-port 33434:33524  -m state --state
NEW  -j RULE_12
$IPTABLES -A INPUT -p udp  -s 10.1.1.6
```

```
--destination-port 33434:33524  -m state --state
NEW  -j RULE_12
$IPTABLES -A INPUT -p udp  -s 10.1.6.4
--destination-port 33434:33524  -m state --state
NEW  -j RULE_12
$IPTABLES -A OUTPUT -p icmp  -s 10.0.0.0/8
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A OUTPUT -p icmp  -s 1.1.1.242
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A OUTPUT -p icmp  -s 10.1.1.6
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A OUTPUT -p icmp  -s 10.1.6.4
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A OUTPUT -p tcp  -s 10.0.0.0/8
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.242
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
$IPTABLES -A OUTPUT -p tcp  -s 10.1.1.6
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
$IPTABLES -A OUTPUT -p tcp  -s 10.1.6.4
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
$IPTABLES -A OUTPUT -p udp  -s 10.0.0.0/8
--destination-port 33434:33524  -m state --state
NEW -j RULE_12
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.242
--destination-port 33434:33524  -m state --state
NEW  -j RULE_12
$IPTABLES -A OUTPUT -p udp  -s 10.1.1.6
--destination-port 33434:33524  -m state --state
NEW  -j RULE_12
$IPTABLES -A OUTPUT -p udp  -s 10.1.6.4
--destination-port 33434:33524  -m state --state
NEW  -j RULE_12
$IPTABLES -A FORWARD -p icmp  -s 10.0.0.0/8
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A FORWARD -p icmp  -s 1.1.1.242
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A FORWARD -p icmp  -s 10.1.1.6
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A FORWARD -p icmp  -s 10.1.6.4
--icmp-type 8/0   -m state --state NEW  -j RULE_12
$IPTABLES -A FORWARD -p tcp  -s 10.0.0.0/8
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
```

```
$IPTABLES -A FORWARD -p tcp  -s 1.1.1.242
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
$IPTABLES -A FORWARD -p tcp  -s 10.1.1.6
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
$IPTABLES -A FORWARD -p tcp  -s 10.1.6.4
--source-port 1024:65535  --destination-port 43  -m
state --state NEW  -j RULE_12
$IPTABLES -A FORWARD -p udp  -s 10.0.0.0/8
--destination-port 33434:33524  -m state --state
NEW  -j RULE_12
$IPTABLES -A FORWARD -p udp  -s 1.1.1.242
--destination-port 33434:33524  -m state --state
NEW  -j RULE_12
$IPTABLES -A FORWARD -p udp  -s 10.1.1.6
--destination-port 33434:33524  -m state --state
NEW  -j RULE_12
$IPTABLES -A FORWARD -p udp  -s 10.1.6.4
--destination-port 33434:33524  -m state --state
NEW  -j RULE_12
$IPTABLES -A RULE_12   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 12 -- ACCEPT "
$IPTABLES -A RULE_12  -j ACCEPT
#
# Rule 13(global)
#
# All VPN users get access to internal servers for
various services
#
$IPTABLES -N RULE_13
$IPTABLES -A FORWARD -p tcp  -m multiport  -s
10.253.0.0/16  -d 10.1.5.0/24  --destination-ports
139,135,42,445,88,389,636,3268,3269,53,25,465,143,
993,80  -m state --state NEW  -j RULE_13
$IPTABLES -A FORWARD -p tcp  -m multiport  -s
10.253.0.0/16  -d 10.1.5.0/24  --destination-ports
443,21,20,3128  -m state --state NEW  -j RULE_13
$IPTABLES -A FORWARD -p udp  -m multiport  -s
10.253.0.0/16  -d 10.1.5.0/24  --destination-ports
138,137,53,88  -m state --state NEW  -j RULE_13
$IPTABLES -A RULE_13   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 13 -- ACCEPT "
$IPTABLES -A RULE_13  -j ACCEPT
#
# Rule 14(global)
#
# Allow these servers to contact selected NTP servers
#
```

```
$IPTABLES -N RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
209.51.161.238  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
132.236.56.250  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
128.59.59.177  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
216.204.156.2  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
198.147.37.140  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
65.211.109.1  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
65.211.109.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
128.105.39.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d
128.105.37.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
209.51.161.238  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
132.236.56.250  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
128.59.59.177  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
216.204.156.2  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
198.147.37.140  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
65.211.109.1  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
```

```
65.211.109.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
128.105.39.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -A OUTPUT -p udp  -s 1.1.1.106  -d
128.105.37.11  --destination-port 123  -m state
--state NEW  -j RULE_14
$IPTABLES -N Cid3F4B361B.0
$IPTABLES -A FORWARD  -s 10.1.4.2  -m state --state
NEW  -j Cid3F4B361B.0
$IPTABLES -A FORWARD  -s 10.1.4.3  -m state --state
NEW  -j Cid3F4B361B.0
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 209.51.161.238
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 132.236.56.250
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 128.59.59.177
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 216.204.156.2
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 198.147.37.140
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 65.211.109.1
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 65.211.109.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 128.105.39.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.0 -p tcp  -d 128.105.37.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -N Cid3F4B361B.1
$IPTABLES -A FORWARD  -s 10.1.4.2  -m state --state
NEW  -j Cid3F4B361B.1
$IPTABLES -A FORWARD  -s 10.1.4.3  -m state --state
NEW  -j Cid3F4B361B.1
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 209.51.161.238
 --destination-port 123  -m state --state NEW  -j
RULE_14
```

```
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 132.236.56.250
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 128.59.59.177
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 216.204.156.2
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 198.147.37.140
 --destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 65.211.109.1
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 65.211.109.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 128.105.39.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A Cid3F4B361B.1 -p udp  -d 128.105.37.11
--destination-port 123  -m state --state NEW  -j
RULE_14
$IPTABLES -A RULE_14   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 14 -- ACCEPT "
$IPTABLES -A RULE_14  -j ACCEPT
#
# Rule 15(global)
#
# Tech Support access to ISP's news server
#
$IPTABLES -N RULE_15
$IPTABLES -A FORWARD -p tcp  -s 10.10.128.0/24  -d
216.168.3.44  --destination-port 119  -m state
--state NEW  -j RULE_15
$IPTABLES -A RULE_15   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 15 -- ACCEPT "
$IPTABLES -A RULE_15  -j ACCEPT
#
# Rule 16(global)
#
# Firewall and Border Router may send logging
messages to central syslog server.  Firewall's logs
are routed out its logging interface.  The border
router has no logging interface, so we allow logs to
come from its first ethernet interface
#
$IPTABLES -N RULE_16
```

```
$IPTABLES -A OUTPUT -p tcp  -s 10.1.6.2  -d
10.1.3.11  --destination-port 514  -m state --state
NEW  -j RULE_16
$IPTABLES -A OUTPUT -p udp  -s 10.1.6.2  -d
10.1.3.11  --destination-port 514  -m state --state
NEW  -j RULE_16
$IPTABLES -A FORWARD -p tcp  -s 1.1.1.105  -d
10.1.3.11  --destination-port 514  -m state --state
NEW  -j RULE_16
$IPTABLES -A FORWARD -p udp  -s 1.1.1.105  -d
10.1.3.11  --destination-port 514  -m state --state
NEW  -j RULE_16
$IPTABLES -A RULE_16   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 16 -- ACCEPT "
$IPTABLES -A RULE_16  -j ACCEPT
#
# Rule 17(global)
#
# Reject ident queries to the email gateway with a
port unreachable, so smtp conversations don't hang.
#
$IPTABLES -N RULE_17
$IPTABLES -A OUTPUT -p tcp  -d 10.1.4.2
--destination-port 113  -j RULE_17
$IPTABLES -A FORWARD -p tcp  -d 10.1.4.2
--destination-port 113  -j RULE_17
$IPTABLES -A RULE_17   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 17 -- REJECT "
$IPTABLES -A RULE_17  -j REJECT  --reject-with
icmp-port-unreachable
#
# Rule 18(global)
#
# Management station allowed to perform SNMP queries
to all network devices
#
$IPTABLES -N RULE_18
$IPTABLES -A INPUT -p tcp  -s 10.1.3.25  -d 10.1.1.1
 --destination-port 161  -m state --state NEW  -j
RULE_18
$IPTABLES -A INPUT -p udp  -s 10.1.3.25  -d 10.1.1.1
 --destination-port 161  -m state --state NEW  -j
RULE_18
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.25  -d
10.1.1.6  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.25  -d
1.1.1.105  --destination-port 161  -m state --state
NEW  -j RULE_18
```

```
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.25  -d
10.1.4.3  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.25  -d
10.1.4.2  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p tcp  -s 10.1.3.25  -d
10.1.4.254  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p udp  -s 10.1.3.25  -d
10.1.1.6  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p udp  -s 10.1.3.25  -d
1.1.1.105  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p udp  -s 10.1.3.25  -d
10.1.4.3  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p udp  -s 10.1.3.25  -d
10.1.4.2  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A FORWARD -p udp  -s 10.1.3.25  -d
10.1.4.254  --destination-port 161  -m state --state
NEW  -j RULE_18
$IPTABLES -A RULE_18   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 18 -- ACCEPT "
$IPTABLES -A RULE_18  -j ACCEPT
#
# Rule 19(global)
#
# Tech Support gets remote admin access to all
internal hosts and the border router via ssh or
PCAnywhere
#
$IPTABLES -N Cid3F4B9237.0
$IPTABLES -A INPUT  -d 10.0.0.0/8  -m state --state
NEW  -j Cid3F4B9237.0
$IPTABLES -A INPUT  -d 1.1.1.105  -m state --state
NEW  -j Cid3F4B9237.0
$IPTABLES -N Cid3F4B9237.1
$IPTABLES -A Cid3F4B9237.0 -p tcp  -m multiport
--destination-ports 5631,22  -m state --state NEW
-j Cid3F4B9237.1
$IPTABLES -N RULE_19
$IPTABLES -A Cid3F4B9237.1  -s 10.253.4.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A Cid3F4B9237.1  -s 10.254.1.0/24  -m
state --state NEW  -j RULE_19
```

```
$IPTABLES -A Cid3F4B9237.1  -s 10.10.128.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.253.4.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.253.4.0/24  -d
1.1.1.105  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.254.1.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.254.1.0/24  -d
1.1.1.105  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.10.128.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A INPUT -p udp  -s 10.10.128.0/24  -d
1.1.1.105  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -N Cid3F4B9237.2
$IPTABLES -A FORWARD  -d 10.0.0.0/8  -m state
--state NEW  -j Cid3F4B9237.2
$IPTABLES -A FORWARD  -d 1.1.1.105  -m state
--state NEW  -j Cid3F4B9237.2
$IPTABLES -N Cid3F4B9237.3
$IPTABLES -A Cid3F4B9237.2 -p tcp  -m multiport
--destination-ports 5631,22  -m state --state NEW
-j Cid3F4B9237.3
$IPTABLES -A Cid3F4B9237.3  -s 10.253.4.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A Cid3F4B9237.3  -s 10.254.1.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A Cid3F4B9237.3  -s 10.10.128.0/24  -m
state --state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.253.4.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.253.4.0/24  -d
1.1.1.105  --destination-port 5632  -m state --state
NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.254.1.0/24  -d
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.254.1.0/24  -d
1.1.1.105  --destination-port 5632  -m state --state
NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.10.128.0/24  -d
```

```
10.0.0.0/8  --destination-port 5632  -m state
--state NEW  -j RULE_19
$IPTABLES -A FORWARD -p udp  -s 10.10.128.0/24  -d
1.1.1.105  --destination-port 5632  -m state --state
NEW  -j RULE_19
$IPTABLES -A RULE_19   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 19 -- ACCEPT "
$IPTABLES -A RULE_19  -j ACCEPT
#
# Rule 20(global)
#
# Tech Support gets remote admin access to legacy
network devices via telnet (internal only).
#
$IPTABLES -N Cid3F525A2D.0
$IPTABLES -A FORWARD  -s 10.253.4.0/24  -m state
--state NEW  -j Cid3F525A2D.0
$IPTABLES -A FORWARD  -s 10.254.1.0/24  -m state
--state NEW  -j Cid3F525A2D.0
$IPTABLES -A FORWARD  -s 10.10.128.0/24  -m state
--state NEW  -j Cid3F525A2D.0
$IPTABLES -N RULE_20
$IPTABLES -A Cid3F525A2D.0 -p tcp  -d 10.10.128.254
--destination-port 23  -m state --state NEW  -j
RULE_20
$IPTABLES -A Cid3F525A2D.0 -p tcp  -d 10.10.128.253
--destination-port 23  -m state --state NEW  -j
RULE_20
$IPTABLES -A Cid3F525A2D.0 -p tcp  -d 10.10.2.254
--destination-port 23  -m state --state NEW  -j
RULE_20
$IPTABLES -A Cid3F525A2D.0 -p tcp  -d 10.10.3.254
--destination-port 23  -m state --state NEW  -j
RULE_20
$IPTABLES -A Cid3F525A2D.0 -p tcp  -d 10.10.128.251
--destination-port 23  -m state --state NEW  -j
RULE_20
$IPTABLES -A Cid3F525A2D.0 -p tcp  -d 10.10.128.252
--destination-port 23  -m state --state NEW  -j
RULE_20
$IPTABLES -A Cid3F525A2D.0 -p tcp  -d 10.10.128.250
--destination-port 23  -m state --state NEW  -j
RULE_20
$IPTABLES -A Cid3F525A2D.0 -p tcp  -d 10.10.128.249
--destination-port 23  -m state --state NEW  -j
RULE_20
$IPTABLES -A Cid3F525A2D.0 -p tcp  -d 10.10.2.253
--destination-port 23  -m state --state NEW  -j
```

```
RULE_20
$IPTABLES -A Cid3F525A2D.0 -p tcp  -d 10.10.3.253
--destination-port 23  -m state --state NEW  -j
RULE_20
$IPTABLES -A Cid3F525A2D.0 -p tcp  -d 10.10.3.252
--destination-port 23  -m state --state NEW  -j
RULE_20
$IPTABLES -A RULE_20   -m limit --limit 6/minute -j
LOG  --log-level 6 --log-prefix "RULE 20 -- ACCEPT "
$IPTABLES -A RULE_20  -j ACCEPT
#
# Rule 21(global)
#
# Staff who can SSH to firewall are allowed to telnet from
firewall to border router for remote administration.
#
$IPTABLES -N RULE_21
$IPTABLES -A OUTPUT -p tcp  -s 1.1.1.106  -d 1.1.1.105
--destination-port 23  -m state --state NEW  -j RULE_21
$IPTABLES -A RULE_21   -m limit --limit 6/minute -j LOG
--log-level 6 --log-prefix "RULE 21 -- ACCEPT "
$IPTABLES -A RULE_21  -j ACCEPT
#
# Rule 22(global)
#
# 'catch all' rule
#
$IPTABLES -N RULE_22
$IPTABLES -A OUTPUT  -j RULE_22
$IPTABLES -A INPUT  -j RULE_22
$IPTABLES -A FORWARD  -j RULE_22
$IPTABLES -A RULE_22   -m limit --limit 6/minute -j LOG
--log-level 4 --log-prefix "RULE 22 -- DENY "
$IPTABLES -A RULE_22  -j DROP
#
#
echo 1 > /proc/sys/net/ipv4/ip_forward
```

C VPN Gateway Configuration

C.1 /etc/ipsec.conf

```
config setup
  # use the default route interface [internet-bound]
  # as the IPSec interface also
  interfaces=%defaultroute
  # use strict crl policy.  We want terminated
  # employees to be locked out immediately
  strictcrlpolicy=yes
  # load all the connection descriptions marked for
  # auto loading.
  plutoload=%search
  plutostart=%search
  uniqueids=yes
  # this is the default, but make it explicit for
  # clarity's sake.  We don't allow any packet not
  # matching a connection description to pass through
  # the gateway!
  packetdefault=drop
conn %default
  # not strictly needed, except for clarity
  type=tunnel
  # use RSA based authentication with certificates
  # for key generation (ISAKMP)
  authby=rsasig
  rightrsasigkey=%cert
  keyingtries=10
  # use ESP for IPSec authentication
  auth=esp
  # my side is left - the freeswan security gateway
  left=1.1.1.242
  leftnexthop=1.1.1.241
  leftid=@secure.giac-enterprises.com
  leftcert=gateway.pem
  # check source address of unencrypted packets on arrival
  disablearrivalcheck=no
  # load connection definitions   automatically
  auto=add

# The following three entries are functionally
# redundant right now.  In the future, there may
# be genuine differences in the type of access
# granted these groups.  Hence the separate
# definitions.
```

```
# Also note that large blocks of the 10.253/16
# network space remain.  This allows us to add
# far more remote clients and treat each one
# differently by assigning it to a particular
# subnet
conn corporate
  right=%any
  rightid="C=US, O=GIAC Enterprises, OU=Corporate, CN=*"
  rightsubnetwithin=10.253.1.0/24
  leftsubnet=10.1.5.0/24
  leftupdown=[script + arguments needed to set up dynamic firewalling for corporate users]
conn editorial
  right=%any
  rightid="C=US, O=GIAC Enterprises, OU=Editorial, CN=*"
  rightsubnetwithin=10.253.2.0/24
  leftsubnet=10.1.5.0/24
conn sales
  right=%any
  rightid="C=US, O=GIAC Enterprises, OU=Sales, CN=*"
  rightsubnetwithin=10.253.3.0/24
  leftsubnet=10.1.5.0/24
conn IT
  right=%any
  rightid="C=US, O=GIAC Enterprises, OU=Technical Support, CN=*"
  rightsubnetwithin=10.253.4.0/24
  leftsubnet=10.0.0.0/8
conn IT-gateway
  right=%any
  rightsubnetwithin=10.253.4.0/24
  rightid="C=US, O=GIAC Enterprises, OU=Technical Support, CN=*"
  leftsubnet=10.1.1.6/32

conn restricted
  type=reject
  right=%any
  leftsubnet=10.0.0.0/8
conn internet
  type=passthrough
  right=%any
  rightsubnetwithin=10.253.0.0/16
  rightid="C=US, O=GIAC Enterprises, OU=*, CN=*"
  leftsubnet=0.0.0.0/0
conn private
  auto=ignore

conn block
  auto=ignore
```

```
conn private-or-clear
  auto=ignore

conn clear-or-private
  auto=ignore

conn clear
  auto=ignore

conn packetdefault
  auto=ignore

conn internet-only
right=%any
rightid="CS=US, O=GIAC Enterprises"
rightsubnetwithin=10.0.0.0/8
leftsubnet=0.0.0.0/0
```

C.2 /etc/ipsec.secrets

```
# gateway's private key file
# Stored unlocked!  Protect the file and
# directory from being read by anyone but
# root! with 700 perms on dir and 600 perms
# on key file!
: RSA private/gateway.key
```

C.3 /etc/ipsec.d/cacerts

C.4 /etc/ipsec.d/crls

C.5 Crontable entries

```
# update my cert, CA certs, CRLs, and keys
[insert command to fetch Verisign crl and store]
30 12 * * * ipsec auto --rereadall
```

D Redhat Network Scheduled Updates Script

This script is copyright 2003 by Rick Shank of Sourceminders, Inc. Long lines in the original have been broken up for readability.

```
#!/bin/bash

/usr/sbin/up2date -u > /tmp/up2date.log

SCAN=‘grep "Retrieved." /tmp/up2date.log‘

if [ "$SCAN" != "" ]
then
  mail -s "‘hostname --short‘ package update"
admin@thebank.com < /tmp/up2date.log >/dev/null 2> /dev/null
fi

ERROR=‘grep "RPM package conflict error." /tmp/up2date.log‘

if [ "$ERROR" != "" ]
then
  mail -s "Error running up2date on ‘hostname --short‘"
admin@thebank.com < /tmp/up2date.log >/dev/null 2> /dev/null
fi
```

References

[1] Tom Karygiannis and Les Owens. Wireless network security: 802.11, bluetooth, and handheld devices. Technical Report 800-48, National Institute of Standards and Technology, 2002. URL csrc.nist.gov/publications/nistpubs/800-48/NIST$_S P_8 00 - 48.pdf$.

[2] *IEEE Standard for Information technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, New York, NY, 1999. Institute of Electrical and Electronics Engineers, Inc. URL http://standards.ieee.org/getieee802/download/802.11-1999.pdf.

[3] Susan Kennedy. Best practices for wireless network security. *Computer World*, November 2003. URL http://www.computerworld.com/printthis/2003/0,4814,86951,00.html.

[4] *The WWWD4 Map Page*, July 2004. worldwidewardrive.org, wigle.net. URL https://wigle.net/gps/gps/GPSDB/onlinemap/?eventid=1.

[5] Matthew Gast. The top seven security problems of 802.11 wireless. Technical report, AirMagnet, Inc., 2004. URL www.airmagnet.com./.../AirMagnet.Security.WhitePaper25.pdf.

[6] Nikita Borisov, Ian Goldberg, and David Wagner. Security of the wep algorithm. Technical report, Unversity of California, Berkeley, January 2001. URL http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html.

[7] *IEEE Standard for Local and Metropolitan Area Networks - Port-Based Network Access Control*, New Yoek, NY, July 2001. Institute of Electrical and Electronics Engineers, Inc. URL http://standards.ieee.org/getieee802/download/802.1X-2001.pdf.

[8] Larry J. Blunk and John R. Vollbrecht. Ppp extensible authentication protocol (eap). Standard, Internet Engineering Task Force, March 1998. URL http://www.ietf.org/rfc/rfc2284.txt?number=2284.

[9] John Koh.l and B. Clifford Newman. The kerberos network authentication service (v5). Proposed standard, Internet

Engineering Task Force, September 1993. URL http://www.ietf.org/rfc/rfc1510.txt?number=1510.

[10] Pat Calhoun, John Joughney, Erik Guttman, Glen Zorn, and Jari Arkko. Diameter base protocol. Proposed standard, Internet Engineering Task Force, September 2003. URL http://www.ietf.org/rfc/rfc3588.txt?number=3588.

[11] Carl Rigney, Steve Willens, Allan Rubens, and William Simpson. Remote authentication dial in user service (radius). Draft standard, Internet Engineering Task Force, June 2000. URL http://www.ietf.org/rfc/rfc2865.txt?number=2865.

[12] Bernard Aboba and Pat Calhoun. Radius (remote authentication dial in user service) support for extensible authentication protocol (eap). Informational, Internet Engineering Task Force, September 2003. URL http://www.ietf.org/rfc/rfc3579.txt?number=3579.

[13] Arunesh Mishra and William A. Arbaugh. An initial security analysis of the ieee 802.1x standard. Research paper, University of Maryland, February 2002. URL http://www.cs.umd.edu/ñaa/1x.pdf.

[14] Comments on âan initial security analysis of the ieee 802.1x standardâ. Technical report, Funk Software, Inc., March 2002. URL http://www.funk.com/radius/Solns/umdresp$_{w}p.asp.$

[15] Cisco aironet response to university of maryland's paper, "an initial security analysis of the ieee 802.1x standard". Product bulletin, Cisco Systems, Inc., August 2002. URL http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/1680$_{p}p.htm.$

[16] Karen Hanley. Overview: Wi-fi protected access. Technical report, Wi-Fi Allance. URL http://www.wi-fi.org/OpenSection/pdf/Wi-Fi$_{P}rotected_{A}ccess_{O}verview.pdf f.$

[17] Wi-fi protected access: Strong, standards-based, interoperable security for todayâs wi-fi networks. Whitepaper, Wi-Fi Alliance, April 2003. URL http://www.wi-fi.org/OpenSection/pdf/Whitepaper$_{W}i-Fi_{S}ecurity4-29-03.pdf.$

[18] *IEEE Standard 802.11i-2004 IEEE Standard for Information technology- Telecommunications and information exchange*

between systems- Local and metropolitan area networks- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 6: Medium Access Control (MAC) Security Enhancements, New York, NY, July 2004. Institute of Electrical and Electronics Engineers, Inc. URL http://ieeexplore.ieee.org/xpl/tocresult.jsp?isNumber=29229.

[19] Adam Stone. The 'michael' vulnerability. *Wi-Fi Planet*, December 2002. URL http://www.wi-fiplanet.com/columns/article.php/1556321.

[20] Sandro Grech and Jani Nikkanen. Attacks on wi-fi protected access. Workshop presentation, November 2004. URL http://www.tml.hut.fi/Nordsec2004/Presentations/grech.pdf.

[21] Robert Moskowitz. Weakness in passphrase choice in wpa interface. Technical report, TruSecure Corporation, November 2003. URL http://wifinetnews.com/archives/002452.html. Unofficially-circulated copy.

[22] Takehiro Takahashi. Wpa passive dictionary attack overview. Technical report, tinypeap.com, November 2004. URL http://www.tinypeap.com/docs/WPA$_P$$assive_D$$ictionary_A$$ttack_O$$verview.pdf$.

[23] Doug Whiting, Russel Housley, and Niels Ferguson. Counter with cbc-mac (ccm). Informational, The Internet Engineering Task Force, September 2003. URL http://www.ietf.org/rfc/rfc3610.txt?number=3610.

[24] *Federal Information Processing Standard Publication 197: Specification for the Advanced Encryption Standard (AES)*, November 2001. National Institute of Standards and Technology. URL http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[25] Jacob Jonsson. On the security of ctr + cbc-mac. Technical report, National Institute of Standards and Technologies, 2002. URL csrc.nist.gov/CryptoToolkit/ modes/proposedmodes/ccm/ccm-ad1.pdf.

[26] Changhua He and John C. Mitchell. Security analysis and improvements for ieee 802.11i. Internet Society, February 2005. URL http://www.isoc.org/isoc/conferences/ndss/05/proceedings/papers/NDSS05-1107

[27] Susan Delaney. *SANS GCFW Practical Assignment, v. 1.9: A Look into the World of External Network Perimeter Security*. SANS Institute, 2003. URL http://www.giac.org/practical/GCFW/Susan$_D elaney_G CFW.pdf$.

[28] John Viega. The myth of open source security. 2000. URL http://itmanagement.earthweb.com/secu/article.php/11076$_6$21851$_2$.

[29] David Obasanjo. The myth of open source security revisited. 2002. URL http://www.developer.com/open/article.php/983621.

[30] Yakov Rekhter, Robert Moskowitz, Daniel Karrenberg, Geert Jan de Groot, and Eliot Lear. Address allocation for private internets. Request for Comments 1918, Internet Engineering Task Force, 1996. URL http://www.ietf.org/rfc/rfc1918.txt.

[31] Internet Assigned Numbers Authority. Special-use ipv4 addresses. Request for Comments 3330, Internet Engineering Task Force, 2002. URL http://www.ietf.org/rfc/rfc3330.txt.

[32] Redhat network offerings. Product description, Redhat, Inc., 2003. URL http://www.redhat.com/software/rhn/offerings/.

[33] How to: Schedule automatic updates in windows xp, windows 2000, or windows server 2003. Technical report, Microsoft, Inc., 2003. URL http://support.microsoft.com/default.aspx?scid=kb;en-us;327838.

[34] Software update services deployment white paper. White paper, Microsoft, Inc., 2003. URL http://www.microsoft.com/windows2000/windowsupdate/sus/susdeployment.asp.

[35] About rhn. Technical report, Redhat, Inc., 2003. URL http://rhn.redhat.com/help/about.pxt.

[36] Michael Rash. Paranoid penguin: Detecting suspect traffic. Tech Note 91, November 2001. URL http://www.linuxjournal.com/article.php?sid=4876.