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 permited without express written permission.

GCFW Practical:
Event Correlation Intrusion Detection Systems

© SANS Institute 2005, Author retains full rights.

Corbin Del Carlo
SANS Online Mentor Program
GCFW Practical (version 4.1)
December 25, 2004

© SANS Institute 2005 Author retains full rights.

Table of Contents

© SANS Institute 2005, Author retains full rights.

© SANS Institute 2005 Author retains full rights.

Abstract

Traditional Intrusion Detection Systems (IDS) have several impediments to their successful operation. Network based intrusion detection products often fail to detect all attacks as they can be defeated by encryption, fragmentation, obfuscation, or just by the shear number of attacks. Host-based systems lack the ability to see what is going on elsewhere in the network, and often are unable to detect network based attacks[1]. To provide complete protection, our defenses need to be informed about all the functions and abilities of our network at all times. Several systems have attempted to bring these abilities to market but most have failed to incorporate all the necessary components to be complete solutions. During this paper we will discuss the creation of a system that will be a central repository for all security related items, this coupled with an informed analysis engine so that it can make informed decisions about network security issues.

GCFW Assignment 1 - Future State of Security Technology

Centralized Intrusion Detection (Event Correlation)

Traditional Intrusion Detection Systems (IDS) have several impediments to their successful operation. Network based intrusion detection products often fail to detect all attacks as they can be defeated by encryption, fragmentation, obfuscation, or just by the shear number of attacks. Host-based systems lack the ability to see what is going on elsewhere in the network, and often are unable to detect network based attacks[2]. To provide complete protection, our defenses need to be informed about all the functions and abilities of our network at all times. Several systems have attempted to bring these abilities to market but most have failed to incorporate all the necessary components to be complete solutions. During this paper we will discuss the creation of a system that will be a central repository for all security related items, this coupled with an informed analysis engine so that it can make informed decisions about network security issues.

Current intrusion detection products have a very limited view of the network as a whole. This is due in part because of our use of defense in depth. Each device is aware of the attacks directly around it but is unaware of attacks happening outside its range of vision or at other layers. This is not to say that defense in depth is a bad thing. Defense in depth provides us with many levels of protection to ensure that even if one security system fails, the system is not necessarily compromised. But, in order to make the intrusion detection system more effective we need to expand its vision to include other layers of our defenses. The additional information provided by other layers will make our network harder to penetrate because the devices at each layer will know about other questionable traffic.

However, providing more information to the intrusion detection system itself does not necessarily help. Most intrusion detection sensors are already quite taxed just attempting to provide enough resources to properly deal with the real time processing of the information within their vision. To effectively manage all this data we will require a separate processor for all IDS messages. We will configure each IDS agent and each system to send alerts and event logs to a central repository. This transactional database will collect events from each of our IDS sensors/agents as well as important network devices such as border routers, firewalls, load balancers, proxy systems, and application

© SANS Institute 2005 Author retains full rights.

and authentication servers. The syslog protocol originally developed for Unix and described in RFC 3164 provides a standard message delivery framework for all types of systems. This framework is cross platform and will serve as the best way to deliver messages from our sensors/agents/hosts to the central repository. It is not perfect, as the message formats from systems developed by different vendors will not be in a standard format. This will have to be accounted for in our log interpretation system. The Internet Engineering Task Force (IETF) has founded the Intrusion Detection Exchange Format Working Group (IDWG) to develop a standard message format for IDS systems.[3] A draft of the Intrusion Detection Exchange Protocol (IDXP) has been published[4] but has yet to be widely adopted. This protocol will reduce the work on our collection system but does not affect the parts of our tracking that are not intrusion detection systems. Several alerts and significant information can be attained from just the system logs of our different hosts. These events are not accounted for in the IDWG work.

The central repository will collect all the relevant information and move it to a relational database. Due to the number of alerts that are assumed to be produced by these various systems, it will be important for the repository to be able to distinguish those messages that are necessary from those that are not necessary. If we assume that we have messages from a network based IDS, a host based IDS, email server, Web server, Web application server, a Cisco Firewall, and a Cisco Content Based Access Controls (CBAC) border router, even a site with a minimal utilization could potentially create a large number of events. The volume of messages from these systems would be overwhelming to attempt to track all messages in our database. To account for this our central repository will filter messages that are not necessary or that do not contribute to our overall security environment. It would be preferential to filter these messages before they are sent, but since some syslog daemons do not provide this functionality, our repository will have to do it itself. The messages filtered at the repository will be information that is not relevant to our security environment.

The messages that we need to filter will be very specific to our environment and will require an amount of customization. Several normal functions can often trigger alerts by intrusion detection systems. For example many network discovery tools conduct a ping sweep of the network to determine uptime or to discover new devices that should be managed. This sweep is conducted at a usual time and by a usual host and should not cause an alarm to be generated. So this type of information should be prevented from being stored in the repository.

Our system needs to know about the structure of the network. Most IDS do not have an understanding of the severity of an attack based on the placement of the device within the structure of the network. For example, our IDS (if placed outside the firewall) is going to see unauthorized traffic attempts to exploit Web vulnerabilities on systems that are not running a Web server and might not even have the port allowed though the firewall. These attempts do not generate enough risk to merit the alert that would be generated by the attackers actions, but this information could prove very important at a later time. If the attacker moved to a valid Web server or the attacker remained persistent by extending the number and types of systems that they are attacking then an alert should be triggered. Our central repository will have a basic understanding of how the network is configured. This will help it determine which systems are most important and what services they are running. Often times companies may have some public IP addresses

© SANS Institute 2005 Author retains full rights.

that are not assigned to any host. Since these addresses are not in use they could be misused. However, because our system understands the structure of the network, unused IP addresses would generate an alert if used in ways inconsistent with their location in the network. In addition to providing enhanced security, having an understanding of how the network is configured would have the added benefit of being able to provide uptime and performance metrics. Since our system will have the ability to alert personnel and will be monitoring logs from our systems, if the Web server were to shutdown we can have it generate an alert to our network administrator to get it up and running again quickly.

Another difficulty that will be presented to this system will be that not all IDS will identify an attacker the same way.[5] This will be essential to tracking our attacker. We will have to normalize all messages with the IP address of the source system being standardized as the identifier of a particular attacker. While this information could be forged or "spoofed" in an attack and could even be made random to simulate multiple attackers, it is the only way we have to determine the source of the attacks against our systems. Since the attacker will have to use at least one address that is legitimate if he/she wants to receive any results of their attacks, we will depend on this except in the case of denial of service attacks. We will not depend on Source IP address in the case of denial of service attacks, as often it is the case that the source address is "spoofed" to ensure that the computer originating the denial of service attack is not itself overwhelmed with response traffic. In any other type of attack, the IP address resolved to the system will be our identifier for that system. To identify an attacker using a denial of service attack we will be forced to assume that if there are multiple identical denial of service signatures that they are originating from the same host or the same controlling entity. Actions taken by that system will be monitored by our analysis engine and generate alerts accordingly. There are two drawbacks to this identification mechanism. The first is the time that could be required to resolve hostnames to IP addresses. Hopefully this will be minimized but will be handled by the central repository to ensure that items in the database have the same identifier. The second drawback is that if the network is using Network Address Translation (NAT) the attacker's IP address could change with each attack. Typically since we will be dealing with source addresses, the NAT issue should be limited. If the attacking network is behind a firewall we will still be informed of the firewall's IP address which is not perfect but will at least lead us to the attackers network. Other issues with NAT could arise in determining the target of the attack. Our external IDS sensors are going to return the external IP address of the target, but internal IDS or host IDS will likely return the internal IP address of the target. Our central repository will have to be aware of any static NAT, and make changes to the messages as they are written to the database.

Once the information has been standardized and written to our central repository we will need to analyze the information to make use of it. There will be two methods for this. One method will be an ad hoc reporting system for obtaining results after the fact to provide baseline information and statistical anomalies. The second method will be a close to real time event analysis engine. As information is written to our central repository, the analysis engine will be continually analyzing all events for the threat they present, both alone and in aggregate. The analysis engine will have attack signatures as well as heuristic capabilities and will have to be designed to allow for significant customization. The analysis engine is in essence an expert system for finding an attacker

© SANS Institute 2005 Author retains full rights.

and filtering out "noise" so that the security administration personnel are only notified of attacks that require their immediate action. Included at Exhibit A is a flow diagram for the different parts of our system.

The largest single benefit this system will provide is that it will have the capability of correlating events from several systems to get a clear picture of an attack. A simple example of this benefit would be if an attack signature was spotted by our IDS and reported to our database of a blaster worm hit on a particular host. Followed by the central repository receiving a message from the host of a crash of the RPC service. We would have a good idea that the Blaster worm was successful and consequentially generate an alert. In addition, this could provide us with knowledge of firewall misconfigurations before they become too large of an issue. Firewall changes are usually made with a good deal of review to ensure that they are not going to cause problems. However, if the administrator makes a typo or enables rules that they did not mean to enable, the systems security could be compromised. Our system would be able to monitor attack traffic at multiple levels of the network's defense allowing us to spot traffic patterns that should not be allowed. As in our example, RPC traffic is not typically necessary through the firewall, but by seeing the attack and the related crash of the RPC system we know that the attacker somehow got the packets past our firewall. This should allow us to proactively attempt to close these holes in our configuration before they are significantly exploited.

Another problem with traditional IDS is, it is often the case that if a signature on the IDS matches too often or becomes part of an automated attack, administrators will begin to ignore the alert.[6] For instance, in a typical environment, we would want our IDS to alert us of a Web-based attack such as the one used by Code Red. However, automated attacks such as Code Red and its variants that exploited multiple vulnerabilities created so many signature matches and alerts that the typical security administrator began ignoring these types of attacks because responding to each was impractical. Our system would be better equipped to deal with this type of attack. It could see the multiple exploits in close time proximity and from the same host and generate a single alert of "Code Red or variant attack." This will reduce the number of alerts and can allow the administrator to delegate this specific attack type as something that should not be alerted but is logged. By logging these attacks we can do weekly/monthly reporting based on standard deviations. This way we can look for statistically significant changes in attack patterns. Over time this will also give us the ability to know when we have a significant threat or just normal background noise.

In order for our system to be successful we need to establish the rules by which all these events will put together into a defined attack pattern. To avoid the pitfalls of previous IDS systems, we will need the correlation system to be selective to ensure that the system is not over zealous in its notifications. On the other hand it cannot fail to notify of an attack. Since we are provided so much more information and our vision is extended to all the levels of our defense we will be provided with the information to make much more informed decisions. The decision tree for this system will use a security goal list defined by the user as proposed by Goldman et al[7]. In this system our security administrator must develop or implement security goals for the system. For example, goals such as ROOT_PRIVILEGE_USE, or HOST_AVALIBILITY will have to be customized by the security administrator, as each network is different. Each of the

© SANS Institute 2005 Author retains full rights.

defined goals will be assigned to a particular host (i.e. were not very worried if we see an attempted Linux exploit on a Windows system). The security goals will then be compared against the events that are pulled from the central repository. If the events compromise our security goals the system will take the appropriate action configured to either alert the security administrator, block the attack, or both. If the event could contribute to the compromise of our goals then the subsequent events will be monitored for further action that would result in the compromise of a security goal.

Each event signature will be assigned to the goals that it effects. This will create a sort of super structure to our IDS. The individual events will make up the majority of baseline traffic. Each of these will then be applied to the target goals for each system. A compromise of the goals will generate an alert to the security administrator. Because of this, our system will still require a significant amount of human management. Generally, this system will provide significant advantages over traditional IDS, but will realistically cause an increase in workload for security administration. I believe this to be the case because of how most IDS are currently administered. Currently, it is a typical practice for a security administrator to spend the first month "tuning" the IDS to their environment. This involves setting up the system to ignore just about anything that is considered normal traffic. In most cases, this becomes whatever items generate the most output to the security administrator. Once these have been removed from the alert list, day to day maintenance of a typical IDS is not very significant, but large portions of the IDS's functionality have been removed as well. Our new advanced IDS will require more effort from the security administrator, as there is significantly more information that is necessary to operate the device properly. Every time a system IP address is changed, a new service is installed, firewall configuration is changed, a new exploit is discovered, etc. we will be required to update the configuration of the device. The benefit associated with our extra efforts will be an IDS that is more accurate and has less false positives and false negatives. Most current IDS require that the security administrator resolve issues with false positives and false negatives manually. Because of this most issues regarding false positives and false negatives result in the issue being ignored. False negatives (where an attack is not identified by the IDS) are typically not discovered until an attacker is successful in compromising the network security. False Positives (where normal traffic is identified as an attack) typically become noise that is either ignored or generates significant wasted effort attempting to track down the attack or traffic generating the signature match. Since our system is designed to automate the removal of as many of these as possible a major affect will be that the security administrator should be much more efficient and be able to spend more of their time investigating legitimate threats to their organization's information security program.

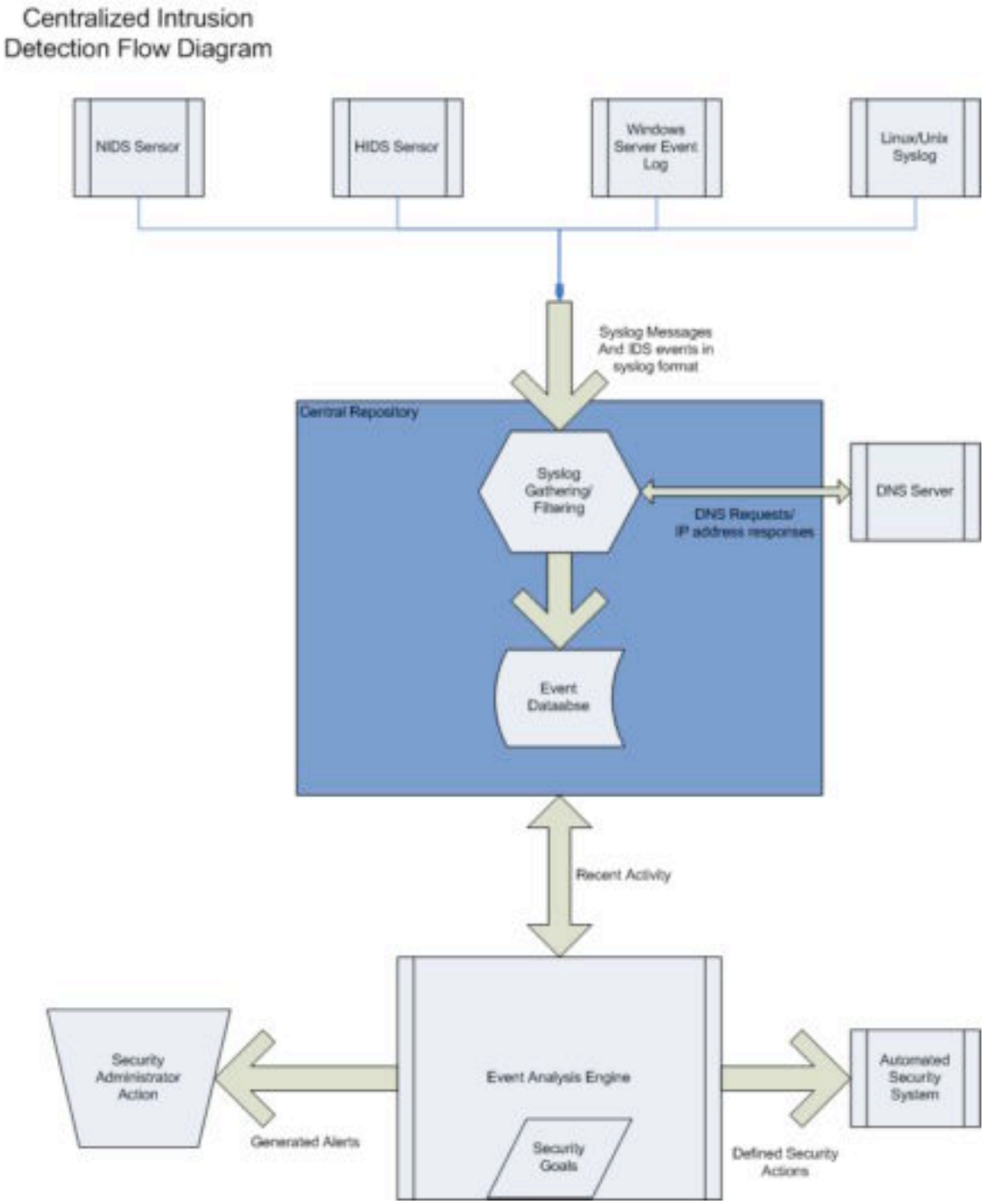
The widespread adoption of centralized logging repositories will provide security administrators with a more complete tool kit to protect their network. The technology provides the potential to eliminate time wasted on any false positive or false negative alert. This will in time allow the security administrator more time to examine other areas of network security and provide a more complete protection environment. Currently, the IDS has struggled to gain a footing. Organizations have purchased and downloaded IDS products. But most are not in regular use or are ignored. When correlation technology is ready, it will make intrusion detection systems a "must have" security device.

© SANS Institute 2005 Author retains full rights.

© SANS Institute 2005, Author retains full rights.

© SANS Institute 2005 Author retains full rights.

Exhibit A



© SANS Institute 2005 Author retains full rights.

GCFW Assignment 2 – GIAC Enterprises

Network Design Considerations

In developing the network design for GIAC Enterprises we have several different systems that must be merged together. We need to be able to interface with six types of personnel (general public, GIAC remote users, partners, suppliers, customers, and local GIAC employees). General access to the public should be the most restricted. We have determined that the only access we need to give the general public is access to the Web page for marketing (port 80), access to our email server for delivery of email (port 25), and encrypted access (SSL) to the login page to place an order (Port 443). Any further access would be unnecessary and could potentially be a hazard to the organization. The second group is customers or persons that order product from us. We have designed an encrypted (SSL) Web front end (Port 443) for customers to place orders and check the status of placed orders online. This information should be restricted to the person that placed the order. Before or during the order process we will develop a username and password for each customer. All orders will be tied to a single username/password so that only the customer can view their order. The Third group is suppliers. Suppliers provide us with a text file of fortune cookie sayings that we then use in our product. They connect to our network via a virtual private network (VPN) once a month (Protocol 51 Authentication Header, protocol 50 Encapsulated data protocol, and IKE UDP 500). They have access to a single FTP server (FTP Port 20 data and 21 control) that they authenticate to and upload the new sayings for the month to the FTP server. Our personnel then retrieve the file and delete the original once it is properly imported into the database. Only specific GIAC Enterprises employees and the supplier have access to this directory as the fortunes are considered our company's intellectual property and the company or our suppliers could be financial affected by the loss of this confidential data. Our fourth group is our business partners. Different partners provide different services to our organization. Some take the fortunes and translate them for resale in non-English speaking markets. Others take our existing fortunes, layout, and format then resell them to other fortune cookie makers. For these customers we have developed a similar but more advanced Web interface. The users must be set up ahead of time and are authenticated to the Web interface and all transactions are encrypted via SSL (TCP port 443). Customers select the output specification and then the output is generated as an XML or flat text file that is transferred to the client via their Web browser or displayed on screen and copied and pasted into the application of the customer's choice. Our fifth type of user is people with remote access to the network. Remote access has proven difficult to secure in recent times, so access will be limited to as few persons as possible. Our sales force is very mobile and will be unable to do their jobs without occasional remote access. In addition the tech support staff will be called on occasionally to fix items while not on site. For this purpose we will allow access for the technician from home. This access will be granted via a two factor authenticated VPN. (Protocol 51 Authentication Header, protocol 50 Encapsulated data protocol, and IKE UDP 500). Once authenticated these users will have access to the necessary systems on the network. The last users are the internal network users the employees of GIAC Enterprises. This group will have access to the internal local area network when at the office (IP traffic only inside) and will have limited access to Internet. Access to external systems or the

© SANS Institute 2005 Author retains full rights.

Internet will be limited to Web traffic which will be controlled by a Proxy server to provide a layer or protection against malicious HTML code.

Network Design Rational

The design developed for GIAC Enterprises is designed to exemplify the concept of Defense in Depth. Defense in depth incorporates implementing many different security levels. Each level provides an additional layer of protection so that in the event that an external facing level fails the next layer in will still fail-safe and prevent the unauthorized access. The traditional example of this is the medieval castle. Each castle had several different layers to protect the king. It was typically put at the top of a hill so that the enemy could be easily seen (level 1). Then there was the outer wall (level 2), which protected the town. If that wall were to be breached the castle had a moat (level 3) and another wall (level 4). If that were to be breached then the boiling oil (level 5) was used to slow the breach, etc. Defense in depth works the same way in the virtual world. We establish multiple layers to protect our system at several different points to ensure that the least necessary access is granted.

Since all external connections to the network come in though the Internet the Internet Service Provider serves as our first line of defense. Denial of service (DoS) attacks are difficult and expensive to defend against. A company such as GIAC enterprises does not have the cash to pay the ongoing fees associated with redundant Internet connection. To provide adequate bandwidth for our customers, suppliers, and partners our Service Level Agreement with the ISP should include a 99.99% uptime guarantee. It will include provisions to ensure that the ISP has the resources available to us for deflecting a Denial of Service or Distributed Denial of Service attack and to extend us additional bandwidth if necessary. Additionally we intend to utilize our ISP for External Domain Name Services (DNS). DNS can be very difficult to secure and since our external DNS entries will be static most of the time there is little need to take on the additional risk of managing the DNS servers ourselves. To provide an additional layer of protection we will monitor the external DNS entries to ensure the returned IP addresses are correct to ensure the availability of our systems to people outside our organization. Having an ISP supported DNS for external users will allow us to implement a split DNS which separates the DNS entries for internal hosts from the DNS entries of the external hosts. All systems behind the firewall will be using the internal DNS for their DNS traffic. This is a mildly inefficient design as often Web servers and email server need to do reverse DNS lookups and in our particular case in order to do this the traffic will have to pass internally to the DNS server and then the DNS server will forward the request on to the Internet, but this affords us extra protections that make the cost worthwhile. For instance since our Internet DNS servers are hosted by our Internet provider an attacker could attempt to spoof a reverse DNS lookup for him/her self to distribute SPAM or hide their true location. One way to do this is to attempt to send a DNS response faster than the DNS server. But an attacker is going to expect the DNS request to have to return to the Web/Email server that he is attacking. In our case this would not be true, as he would have to return it to our internal DNS, which will pass the result onto the Web/Email server that requested it. Since this Internal DNS server is not published anywhere and not directly accessible from the Internet the attacker will have a hard time finding the IP address that he must return his faked query to.

© SANS Institute 2005 Author retains full rights.

The border router serves as our second line of defense. The border router is designed to protect against basic attacks and to be scalable. Access Control Lists (ACL) on the border router will be designed to block all Request for Comments (RFC) 1918 addresses. Use of the Cisco "AutoSecure" command available in IOS 12.3[8] and above enables several security features of IOS that are typically disabled by default. For instance, it adds the IP addresses in the IANA reserved ranges as well as the RFC 1918 IP addresses to the default ACL applied to the outside interface. In addition the command will configure Ingress and Egress filters so these types of bogus traffic will be blocked. This should give us a reasonable confidence that any traffic that reaches the firewall can be tracked to a valid host. Logging on this device will be selective. Traffic from spoofed IP addresses will not be able to be tracked to their source and often will not allow the attacker to get a valid response. Since the traffic will be dropped flooding our logs with invalid traffic will just make it difficult to see legitimate attacks. This design provides us the additional benefit of being scalable. If the need arises for more bandwidth or a redundant connection a second Internet connection can be brought in from a different ISP to provide disparate redundant paths with additional bandwidth. Remote access to this device will only be enabled via Secure Shell. This will be protected via a username and password. This will encrypt all traffic and provide us with an ability to connect to the device remotely for technical support issues.

The third line of defense would be our network intrusion detection system (NIDS). We had decided on a SNORT based system with the Guardian add-on to allow our network to react to specific attack signatures. The SNORT system was selected due to the high cost associated with the commercial systems that provided the prevention features of the Guardian add-on. We have also selected mostly major vendors for the majority of these devices (Cisco and Microsoft). This will reduce the risk that the Administrator will have a skill set that is difficult to replace if they were to leave the company. Conversely, the use of a Linux based system for Intrusion detection will allow us some diversity in our protection environment, ensuring a single Windows or Linux vulnerability would not extinct our network. We have placed this device outside the Firewall to get an idea of all the attack traffic that is destined for our network. The Guardian add on will provide us with the ability to block basic network attacks. We have designed the IDS to play to the strengths of the Network IDS. Since NIDS are less likely to detect application exploits and can have their fragmentation buffers overrun by a sophisticated attacker these types of attacks will not be checked for on our NIDS. The NIDS will be designed to look for holistic attack patterns. For instance port scans that cover many different systems, or say a probe for a specific service on multiple systems. ICMP based attacks would also be ideal to check for on our external NIDS. The majority of other types of attacks will be detected by other systems in our defenses. Automated blocking of attacks will be very cautious. We do not wish to have service disruptions due to our IDS so known IP ranges will be configured to be on a never block list. This will not prevent the IDS from alerting us of the attack, but if it comes from one of our known business partners or suppliers we would allow it by default and implement a manual block only when serious unresolved issues were discovered. All our NIDS systems will be multihomed. Each will have two network interface cards. The primary card will be in promiscuous mode, meaning it will be completely passive, have no IP address and be near impossible to attack. The second interface will be plugged into our internal network.

© SANS Institute 2005 Author retains full rights.

This will protect the IDS devices and allow them to quickly report events to our central alerting facility.

Our next layer of protection is the firewall itself. We have determined to implement a Cisco 525 PIX firewall in a fully redundant implementation. The Pix 525 is flexible enough to give us several different interfaces to allow us the ability to physically separate disparate networks. We determined that four different Demilitarized Zones (DMZ) would be necessary and then an outside and inside interface. The DMZ's will be separated by the firewall and will have varying security levels. Interfaces with a higher security level can pass traffic to interfaces with lower security levels in the default configuration. But lower security levels can only pass traffic to higher security interfaces when specific exceptions are made.

The least secure interface will be the outside interface with a security of 0. The outside interface will have static address translations to hosts that require access from the Internet. Hosts that will be mapped to the external interface would be Email server, Public Web Server, Public E-Commerce reverse proxy, Supplier/Partner VPN gateway, and Remote User VPN gateway. The Following ports/protocols will be allowed to pass inbound from the outside interface:

Host	Traffic allowed	From Hosts
Email Server (DMZ2)	TCP Port 25 (SMTP)	Any Internet Host
Public Static Web Server (DMZ2)	TCP Port 80 (HTTP)	Any Internet Host
Public E-Commerce Server (reverse Proxy) (DMZ 1)	TCP Port 80 (HTTP Redirect to SSL) TCP Port 443 (SSL)	Any Internet Host
Supplier/Partner VPN (DMZ 4)	Protocol 51 Authentication Header Protocol 50 Encapsulated data protocol UDP Port 500 (IKE) UDP Port 10000 (IPSec NAT Translation)	From IP ranges supplied by Partners and suppliers
Remote User VPN (Internal Network)	Protocol 51 Authentication Header Protocol 50 Encapsulated data protocol UDP Port 500 (IKE) UDP Port 10000 (IPSec NAT Translation)	Any Internet Host

We will assign the DMZ1 interface, which will contain the Web application front-end for our customer ordering system a security of 20. This network will have two systems, one that will function as a reverse proxy and Application firewall for our Web based customer-ordering system. The other will host the web application. Only the reverse proxy will be accessible to systems from a lower or equal security level. The actual Web application will not be directly accessible from any system but the reverse

© SANS Institute 2005 Author retains full rights.

proxy. The application will pass all data on to the Live database. To provide an additional layer the reverse proxy will be functioning as Web command sanitizer and a Host-based Intrusion Detection System (HIDS). Since the reverse proxy is running Linux we will use Snare for Linux as our HIDS on the reverse proxy. The reverse proxy system is implemented to protect against the potentially weak Web application. The reverse proxy will provide a layer of protection against SQL injection, or other Web application attacks that would be allowed through the firewall. Alerts from this system will be passed to our internal syslog server, which will handle all alerting functions (detailed below).

Host	Traffic allowed	From Hosts
Internal Syslog system (Inside)	UDP Port 514 syslog	Web Application (DMZ1) and Reverse Proxy (DMZ1) systems

The DMZ2 that will contain the public Web and Email servers. This interface will have security of 20 as well. We decided to separate these systems, as all access in this segment will be public (unauthenticated). Segmenting this traffic will give us the ability to monitor for public intrusions. All the material in this segment is also very static in nature. Once the Email server and Web Server are setup minimal changes will be needed to the content. Other than access from the Internal network or the Internet no other segment will be granted access to this segment. All of these systems will be monitored to ensure uptime and that the Web page is not changed or defaced. In addition this interface will have a NIDS monitoring for attacks that may have breached the firewall. Some attacks are designed to circumvent firewalls and other times a we intentioned employee may open up bigger holes in his/her firewall that he/she intended, to account for this the NIDS on this interface will alert staff on duty of these attacks so that proper action can be taken. The Public Email server will not store any Email. Email will be forwarded on to the internal server but this system will allow us to have system to provide SPAM and Anti-virus protection to all messages bound for the internal Email server. It will also allow us to prevent an external attacker from impersonating an internal employee Email address. This system will give us the ability to filter any messages like that before our users see them.

Host	Traffic allowed	From Hosts
Internal Mail Server	TCP Port 25 (SMTP)	Email Gateway (DMZ2)
Internal Syslog system (Inside)	UDP Port 514 syslog	Email Gateway (DMZ2) Public Web Server (DMZ2)

DMZ3 will consist of the Backed "Live" database server, which will store all the data from the front-end ordering system. We call this the "Live" database, as all production systems will query this database for information. To provide an additional layer of data protection a backup "production" database will be maintained on the

© SANS Institute 2005 Author retains full rights.

internal network. This database will contain a complete backup of information in the "Live" database to ensure that if an attacker were to manage to corrupt the information in the "Live" database the data could be restored quickly from the Production database. Traffic will be migrated from the "Live" database to the "Production" database after it has passed the daily validation check. The last additional layer of protection in this DMZ will be the implementation of a NIDS on this subnet. A HIDS would be preferable but the transaction load on the "Live" database makes it unlikely that the system would be able to efficiently process both the transaction data and the HIDS signatures. The DMZ3 interface will receive a security level of 70. This will prevent access from all interfaces (except the inside interface) to this interface and will require specific rules to pass traffic to DMZ3. Traffic that will be allowed to access this segment is detailed below:

Host	Traffic allowed	From Hosts
Backend "Live" Database	TCP port 1433 (SQL)	Web Front-end Customer Ordering (10.1.1.5 DMZ1)
Backend "Live" Database	TCP port 1433 (SQL)	Supplier/Partner Web Front-end (10.4.1.5 DMZ 4)

The last DMZ, DMZ4 will contain the Supplier/Partner interface systems. It will also be the termination point of the Suppliers and Partners VPN connections to the Network. The partner supplier network will receive a security level of 20. Suppliers and partners will need more access than the general public, but this access will be explicitly granted to the DMZ they enter the network from to keep them separate from the rest of the organization. Partners and suppliers will be limited by the VPN appliance to only directly access the Web application system over SSL. The Application server will be the only system that can communicate with the "Live" database.

The last interface will be the internal interface, which will have a security level of 100. Internal employees to gain access to servers in the various DMZ's or the Internet will use this interface. Outbound restrictions will be applied to this interface to protect the systems on lower security interfaces from internal users that do not need access to these systems.

Each DMZ has it's own depths of defense so these will be addressed separately. DMZ1 which contains the customer Web front end and reverse proxy has multiple levels of defense. First each system is addressed with a RFC 1918 private address. Since these are blocked at the border router spoofing from the Internet is well protected against. Only the reverse proxy system is accessible from the Internet and only on port 443 the necessary port for SSL. Once users connect to the system they will be prompted for a username and password. Unless authenticated the user will not have access to the application. SQUID's reverse proxy feature gives us the ability to setup application level firewall rules for our consumer Web application. This should give us sufficient protection against SQL injection or other Web application vulnerabilities. To provide an additional level of protection we have implemented SNARE a Linux host-based intrusion detection system (HIDS) on the reverse proxy system. This will detect any strange requests or potential attacks and alert the support staff so that corrective action can be

© SANS Institute 2005 Author retains full rights.

taken. It will provide automated protection of assets as well for certain types of known attacks by immediately ending the transaction. Once the reverse proxy approves of the command and it has been analyzed by the HIDS it is passed to the Windows 2003 Web application server. This system then takes the command processes it and makes any necessary SQL queries or calls. All connections from the reverse proxy to the SQL database will be encrypted with SSL. An additional layer is provided to us by having additional diversity as the reverse proxy runs SQUID and Linux, while the actual Web application server runs Windows and IIS. The application authentication will be passed with the command giving us the ability to restrict access to specific instances and sections of each SQL database. Most customers will have a default level of access, which will only give them access to the product descriptions, quantities, and ability to place an order. The last line of defense is that critical information in the database will be stored encrypted. Information like customer passwords, Customer information, prices, Credit Card numbers, etc would all be encrypted.

DMZ 2, which contains the public Web server and Email Gateway, will have slightly less protection, as the potential risk for financial loss is lower. These systems will addressed with RFC 1918 addresses to provide their first layer of protection. However they will be statically mapped to public IP addresses so they can accomplish their functions. Only the necessary ports of 80 (HTTP) on the Web server and 25 (SMTP) on the Email server will be accessible from the Internet. This will prevent access to all unnecessary ports providing a layer of protection to the network interfaces of our systems. Both systems will be protected by a SNORT network intrusion detection system monitoring for intrusions that have got past the initial NIDS on the external interface and will be alerting IS personnel of critical intrusions. This NIDS will be configured to monitor for application attacks against both the web server and the mail server. This will provide us with a layer of protection for all traffic allowed to pass through the firewall. Since the Web page will not be changing often we will be monitoring the page by downloading the front page and conducting a MD5 hash and comparing against stored hashes from known good components. This will ensure that the page has not been defaced. The SMTP gateway will be protected with an Antivirus and Anti-spam gateway to minimize the risk from malicious emails being forwarded on to our internal users. Of course both systems would be protected with Antivirus software to ensure they are protected from known malicious traffic.

DMZ 3 will contain the "Live" database. As before an initial layer of protection will be provided from the RFC 1918 address and the firewall configuration will only allow the necessary ports. Due to the large processing requirements on the database we do not wish to slow it down by implementing a Host based Intrusion detection on the Database server. So this DMZ will have another SNORT Intrusion Detection system to watch for attacks against the database itself. The database server will have multiple instances that will separate the multiple applications (supplier, partner, and customer orders). Each instance will only be available to the DMZ that runs their parent application. So, the customer order instance will only have access rules allowing SQL traffic to DMZ1. This will provide a level of protection against customers attempting to gain partner or supplier type access. Application authentication will be passed from the Web front-ends to the database. This will provide us the ability to restrict access to data

© SANS Institute 2005 Author retains full rights.

in the database to specific users, preventing malicious users from having the permissions to access the data of other customers.

DMZ 4 is the end point for the Partner/Supplier DMZ. Only the VPN concentrator will be accessible from the Internet. This device will only have the necessary ports and protocols to operate an IPSec VPN available to the Internet. Due to the limited number of partners and suppliers and the limited resources of our organization we will use a RSA Public/Private Key combination. Each key combination will be unique to that partner/supplier to ensure accountability for the tunnel. Each tunnel will only have access to the one system that is located in the DMZ to prevent our partners or suppliers from having access to more things than necessary. This system will act as a Web front-end to the applications necessary for GIAC enterprises business partners. This web front-end will have username and password protection for the application. This will provide the application level access to the system. The suppliers will have access to the FTP service on the server. They will upload files to a directory assigned to their company. Access to the FTP server will be controlled by a unique username and password for each supplier. Once they attach, they will only have read and create permissions to their directory to ensure that the supplier can provide a file but can not remove any files, nor gain unnecessary access to competing suppliers information. Partners will be given a username and password to the Web based application that provides access to all necessary functions for partner applications. The application will provide partners with the ability to order cookies in a bulk fashion at a discount price for resale. In addition partners will be able to purchase fortunes, and design layouts for international sales.

The last segment is the internal network. The internal network consists of all employees of GIAC enterprises and their international offices. Only one system on the internal network will be available from the Internet, the VPN gateway. This will provide remote access for home users and users from the companies international offices. Since device is the highest risk host to the organization we will provide additional layers on protection around this device. The first layer being that only the necessary ports will be available on the Internet. The second layer will be a secure authentication mechanism. As before, this will be an IPSec VPN, but this time we will use a Pre-Shared Key instead of a RSA Public/Private Key. We do not have the resources to manage a PKI implementation and it was determined to be more secure and cheaper to have remote users use RSA SecurID tokens to gain access to the network remotely. This will provide a two-factor authentication of username/password and randomly generated token provided PIN. The VPN will have split tunneling disabled to ensure that any user connected will have all traffic go through the VPN. The only other traffic allowed inbound for the Internal network will be to relay SMTP messages to the internal exchange mail server. The internal "Production" database will have a daily replica of all information in the "Live" database. The "Production" database will pull information from the "Live" database every night at midnight automatically, unless prevented by management. The on site security officer will get a daily transaction summary sheet from the "Live" database everyday. This will give basic statistical data of the transactions on the database for that day (number of transactions per customer, type of transactions, total transactions, etc). This information will be compared to a baseline before it can be transferred to the "Production" database. Any transactions outside of the baseline will be

© SANS Institute 2005 Author retains full rights.

verified with the account manager before the end of day. This will provide us the ability to ensure that a large number of false transactions have not been generated by the hacked account of a client or partner and provide our last line of defense against data corruption in the application. Internal users will gather the supplier FTP files in a timely manner pull them into a system behind the firewall, verify the file is in the correct format, import the data into the "Production" database, and then verify that the import worked successfully. This will be replicated to the "Live" database overnight.

Internal traffic outbound will be restricted to necessary ports and protocols to prevent internal users from abusing the companies Internet access. All outbound Web traffic will be directed though a SQUID proxy server. This server will provide protection by preventing access to malicious content as well as restrict access to sites with questionable content. We have decided to utilize a block list provided by the Websense Corporation to block sites in categories that are not necessary for our employees' daily functions. Additionally we will restrict internal users from attempting to contact external DNS servers. All DNS requests will be forwarded to our internal DNS server. Lastly, each system will be equipped with an anti-virus product and an anti-spyware product. These products will provide an additional layer of protection for our internal users from code that could be used to avoid our protections.

© SANS Institute 2005, Author retains full rights.

© SANS Institute 2005 Author retains full rights.

Exhibit B
Network Diagram

GCFW Assignment 3 - Firewall Configuration/Rule base

Legend

Format	**Meaning**
!--- text	PIX Comment
`nameif ethernet0`	PIX Command
General text	Author explanation

```
PIX Version 6.3
!--- Identify the interface names
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz1 security20
nameif ethernet3 dmz2 security20
nameif ethernet4 dmz3 security70
nameif ethernet5 dmz4 security20
nameif ethernet6 ethernet6 security100
nameif ethernet7 Failover security99
hostname GIAC-PIX

!---fixup protocols enable basic application level firewall  protections on these protocols such as Mailguard
fixup protocol ftp 21
fixup protocol http 80
fixup protocol smtp 25

!---Each name makes the configuration more readable as the access lists will state names instead of IP addresses.
!--- Web Server requires public access
names 12.24.13.15 Static_Public_Web_Server

!--- Customer ordering Web Server requires public access
```

© SANS Institute 2005, Author retains full rights.

Exhibit B
Network Diagram

```
names 12.24.13.14 App_Public_Web_Server

!--- Customer ordering Web Server requires public access internal IP
names 10.1.1.6 Int_Public_App_Web_Server

!--- Servers that interface with the Live Database
names 10.1.1.5 Web_Application_Server_DMZ1

!--- Servers that interface with the Live Database
names 10.4.1.5 Web_Application_Server_DMZ4

!--- SMTP Relay Server
names 12.24.13.16 Email_Gateway

!--- Primary Internal Mail Server
names 10.10.1.10 Exchange_Server

!--- Live database server that interfaces with external applications
names 10.3.1.6 Live_Database

!--- Internal Production Database that contains "clean" data
names 10.10.1.5 Production_Database

!--- Syslog server that stores all logging information from systems in the DMZ
names 10.10.1.7 Syslog_host

!--- Partner VPN Gateway
names 12.24.13.20 Partner_VPN_GW

!--- Partner VPN Gateway(Internal IP)
names 12.24.13.20 Partner_VPN_ref
```

© SANS Institute 2005, Author retains full rights.

Exhibit B
Network Diagram

```
!--- Employee VPN Gateway
names 12.24.13.21 Employee_VPN_GW

!--- Internal DNS
names 10.10.1.20 DNS_Server

!--- Internal Proxy Server
names 10.10.1.25 Web_Proxy_Server

!--- Internal Proxy Server outside IP
names 12.24.13.22 Web_Proxy_Server_ref

!---These object groups describe necessary ports or services that are needed and assign friendly names to them.
object-group service MSSQL_ports tcp
  description Ports necessary for SQL Operation
  port-object eq 1433
object-group service VPN_ports udp
  description Ports necessary for VPN Operation
  port-object eq 500
  port-object eq 10000
object-group service Syslog_ports udp
  description Ports necessary for accessing the Syslog host
  port-object eq 514
object-group service DNS_ports udp
  description Ports necessary for Accessing Internal DNS Server
  port-object eq 53
```

In our selection of the order of the firewall rules that we were going to implement we intend to capitalize on some features of the PIX firewall. First that each access-list will only be applied to a single interface in a single direction. This will reduce the confusion of the

© SANS Institute 2005, Author retains full rights.

function of rules in the rule base but will also improve performance since each interface will only have a small number of rules to process. Second since the rule base will be short we are prioritizing the rules only on the interface that we are applying them to. For simplicities sake we are going to discuss the interfaces in order (outside, DMZ1, DMZ2, DMZ3, DMZ4, inside). Since the access-list is divided by interface as traffic goes in the appropriate interface the firewall will move in the configuration immediately to the access-list that is appropriate for that interface. This would make the order of the interfaces in this document irrelevant in the performance of the firewall. This is not to say that the order of the rules in a particular interface's rule base is unimportant. It just does not affect the processing of traffic on other interfaces.

!--- Access-List to allow traffic from the Internet to the publicly available Web servers.

```
access-list outside_in permit tcp any Static_Public_Web_Server eq http
access-list outside_in permit tcp any App_Public_Web_Server eq https
access-list outside_in permit tcp any App_Public_Web_Server eq http
```

This collection of three access-lists will be the first rules processed by traffic inbound on the external interface. We expect that the most traffic will come to the public Web servers. Most people will be directed to the front page of our static Web site so the first access-list permits access to the opening page of our Web site. The Second Access list provides access to the SSL secured Web Store. This allows our customers to place orders and manage their accounts. We are sure that the second most traffic will come to this site. The Third access list is to allow access to unencrypted Web port on the customer Web application server. The server will be configured to force the users to use SSL but in the event that the user types an http instead of an https we wish to ensure they are forwarded to the proper location.

!--- Access-List to allow traffic from the Internet to the VPN ports on the VPN Gateways.

```
access-list outside_in permit udp any Employee_VPN_GW eq object-group VPN_ports
access-list outside_in permit udp any Partner_VPN_GW eq object-group VPN_ports
```

The Next two access-lists in our firewall configuration will provide us with access from the Internet to the two VPN devices we have running. Our initial thought was that the VPN's would create more Firewall rule hits than the Web traffic, but the vast majority of the VPN traffic is connected quickly and then uses established connections, which are not part of the firewall rule set. Once the connection is established it is stored in the Cisco PIX the connection table and so it does not generate a separate access of the firewall rule base. Since we only need to worry about initial connections we determined there would be more initial connections to the Web devices than to the VPN. It should be noted that the access list only opens up the port for IKE. Lower in the configuration we have

added the "sysopt connection permit-ipsec" command. This command allows IPSec traffic (AH and ESP) to pass through the PIX firewall. Since we are using this command we do not need to provide further access-lists for the VPN.

!--- Access-List to allow traffic to the SMTP server.
```
access-list outside_in permit tcp any Email_Gateway eq smtp
```

The Last rule to be applied to our external interface is the rule that will allow a connection to the SMTP email server. Unless an extreme situation is encountered the traffic from Email should be relatively low considering the number of users that we have. The use of Cisco's Mailguard feature will restrict the SMTP server to the following SMTP commands: HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT.

!--- Access-List to allow traffic to the syslog server.
```
access-list Dmz1_in permit udp Int_Public_App_Web_Server Syslog_host eq object-group
Syslog_ports
access-list Dmz1_in permit udp Web_application_server_DMZ1 Syslog_host eq object-group
Syslog_ports
```

The First rule applied to the DMZ1 interface is configured to allow the syslog messages from our DMZ1 systems to the internal syslog server. As most events will generate a syslog message and because we will be running a Host intrusion detection system on the reverse proxy, which will generate several messages, we consider this to be the most probable traffic pattern to be matched and so we have made it the first rule on this interface. Syslog traffic can be high volume and we considered two other possibilities. One being to leave the syslog traffic on the local hosts or a syslog host in the DMZ but we determined this was not a viable solution as a compromise of the host could lead to a compromise of the logs and make it difficult to determine the extent to which and intrusion occurred. It also limited our ability to have a centralized repository of information. The second option was to have separate Network Interface cards for each host the connected separately to the syslog server. We determined that this added a large amount of management to properly protect the syslog host. So it was determined to pass the syslog traffic through the firewall, if performance becomes a problem we will make adjustments to the design.

!--- Access-List to allow MSSQL traffic to the Live database.
```
access-list Dmz1_in permit tcp Web_application_server_DMZ1 Live_database eq object-group
MSSQL_ports
```

© SANS Institute 2005, Author retains full rights.

Our next rule is configured to allow the MSSQL traffic from our DMZ1 Web application systems to the internal Live MSSQL database server.

!--- Access-List to allow DNS traffic to the DNS Server.

```
access-list Dmz1_in permit udp Int_Public_App_Web_Server DNS_Server eq object-group
DNS_ports
access-list Dmz1_in permit udp Web_application_server_DMZ1 DNS_Server eq object-group
DNS_ports
```

The last rule on this interface will be to allow DNS requests to our internal DNS server. DNS is designed to cache information and typically only does a single query to determine an IP address for a hostname. Once connected further DNS requests are only for new connections.

!--- Access-List to allow legitimate email to be passed to the exchange server.

```
access-list Dmz2_in permit tcp Email_Gateway Exchange_Server eq smtp
```

In DMZ 2 we have very little traffic that is going to be destined for any higher security network. Our Email server will be the primary system to interface to communicate out of the network. It will generate SMTP messages that will be forwarded to the internal Exchange server. The SMTP server will be filtering many messages but there will still be a number of connections that will need to be established and email by its nature is typically large volumes of small messages.

!--- Access-List to allow traffic to the syslog server.

```
access-list Dmz2_in permit udp Email_Gateway Syslog_host eq object-group Syslog_ports
access-list Dmz2_in permit udp Static_Public_Web_Server Syslog_host eq object-group
Syslog_ports
```

The next two rules on the DMZ2 interface simply allow the two systems on the DMZ to send syslog messages to the internal syslog host. The syslog communications on this interface should consist of mostly status reporting and system errors, unlike our MSSQL sessions that require an authentication for each session and a basic transaction log as well. This provides us with the rational that the syslog traffic should be less than the actual application traffic.

© SANS Institute 2005, Author retains full rights.

!--- Access-List to allow traffic to the Internal DNS server.

```
access-list Dmz2_in permit udp Email_Gateway DNS-Server eq object-group DNS_ports
access-list Dmz2_in permit udp Static_Public_Web_Server DNS_Server eq object-group
DNS_ports
```

These last two rules on the DMZ2 interface simply allow the two systems on the DMZ to send DNS requests to the internal DNS Server. The amount of DNS requests will be larger than on other interfaces as the Email server and the Web server will be doing some reverse lookups on hosts that utilize these services. But it is still less traffic than would be generated by syslog or the actual Web requests.

!--- Access-List to allow traffic to the syslog server.

```
access-list Dmz3_in permit udp Live_Database Syslog_host eq object-group Syslog_ports
```

The DMZ 3 interface has a security of 70. Because of this it has a higher security than all the interfaces except the inside interface. The static translations below allow the system to communicate with systems on lower security interfaces but the rule to allow our syslog traffic to the syslog host on the internal network must be applied to facilitate this communication.

!--- Access-List to allow traffic to the DNS server.

```
access-list Dmz3_in permit udp Live_Database DNS_Server eq object-group DNS_ports
```

Our last rule for this interface will be to allow the Live_Database to make DNS requests from the Internal DNS server.

!--- Access-List to allow traffic to the syslog server.

```
access-list Dmz4_in permit udp Web_Application_Server_DMZ4 Syslog_host eq object-group
Syslog_ports
access-list Dmz4_in permit udp Partner_VPN_ref Syslog_host eq object-group Syslog_ports
```

DMZ 4 contains the Web application server for our partners and suppliers and is the termination point for their VPN. The VPN device will be configured to only allow communication with the Web application server but if someway the attacker were able to circumvent this precaution they would be isolated on the network with no confidential information, and without any direct access to the Live

© SANS Institute 2003, Author retains full rights.

database. Again syslog traffic will be very substantial to this interface. The application server will be logging transactions, authentications, and availability messages to the internal syslog host. The Partner VPN gateway will be sending authentication requests and session establishment and teardown messages. This will allow us to ensure that our partner/suppliers are not connecting to the network without need or from strange locations.

!--- Access-List to allow legitimate email to be passed to the exchange server.

```
access-list Dmz4_in permit tcp Web_Application_Server_DMZ4 Live_Database eq object-group
MSSQL_ports
```

The next rule on this interface will be to allow the Web application server to connect to the Live database and conduct transactions.

!--- Access-List to allow DNS Requests to be passed to the Internal DNS server.

```
access-list Dmz4_in permit udp Web_Application_Server_DMZ4 DNS_Server eq object-group
DNS_ports
```

The Last rule on this interface will be to allow the Web application server to connect to the internal DNS server for DNS traffic. We want it to use the Internal DNS because it is protected but it should only be making very limited queries to DNS at best and will cache some requests by default so this should have reduced number of hits on our firewall.

!--- Access-List to allow Web browsing from the Proxy Server

```
access-list inside_in permit tcp Web_Proxy_Server any eq www
access-list inside_in permit tcp Web_Proxy_Server any eq 443
```

The Last interface is out internal interface. This interface will be used to control access that internal employees have to the outside world. The First item is to allow the Web proxy to access the Internet. All other traffic will be denied so this will be necessary to allow Web traffic to leave the network.

!--- Access-List to allow Internal Mail server to forward Mail to Mail Gateway

```
access-list inside_in permit tcp Exchange_Server Email_Gateway eq smtp
```

© SANS Institute 2003, Author retains full rights.

After Web traffic the next most active traffic system on the internal network will be the mail server. Because of this we have selected it as the next rule in our access-list.

!--- Access-List to allow Internal DNS Server to gather DNS information
```
access-list inside_in permit tcp DNS_Server any eq 53
access-list inside_in permit udp DNS_Server any eq 53
```

Next will be our DNS traffic. Again this is restricted to only the Internal DNS server. This will decrease the likelihood of a successful DNS spoofing attack against our organization.

!--- Access-List to allow Production database to gather information from Live database.
```
access-list inside_in permit tcp Production_Database Live_Database eq object-group
MSSQL_ports
```

Our production database will only pull information from the Live database once a day. So this is likely to be the rule with the least hits.

!--- Access-List to deny all remaining traffic
```
access-list inside_in deny ip any any
```

This last rule will block anything not explicitly allowed above. It must be placed last to prevent killing all traffic to the firewall

!---Map External IP addresses to hosts available from the Internet (2VPN, 2Web, Email)
```
static (inside, outside) 12.24.13.21 10.10.1.3 netmask 255.255.255.255 0 0
static (dmz4, outside) 12.24.13.20 10.4.1.3 netmask 255.255.255.255 0 0
static (dmz2, outside) 12.24.13.15 10.2.1.5 netmask 255.255.255.255 0 0
static (dmz2, outside) 12.24.13.16 10.2.1.7 netmask 255.255.255.255 0 0
static (dmz1, outside) 12.24.13.14 10.1.1.6 netmask 255.255.255.255 0 0
```

© SANS Institute 2005, Author retains full rights.

Exhibit B
Network Diagram

```
!---Map IP address of Live Data Base to Application DMZ's
static (dmz3, dmz1) 10.1.1.5 10.1.1.5 netmask 255.255.255.255 0 0
static (dmz3, dmz4) 10.4.1.4 10.4.1.4 netmask 255.255.255.255 0 0

!---Map IP address of Email Gateway to Internal Network
static (inside, dmz2) 10.2.1.7 10.2.1.7 netmask 255.255.255.255 0 0

!---Map IP address of Syslog host to all DMZ's.
static (inside, dmz1) 10.10.1.7 10.10.1.7 netmask 255.255.255.255 0 0
static (inside, dmz2) 10.10.1.7 10.10.1.7 netmask 255.255.255.255 0 0
static (inside, dmz3) 10.10.1.7 10.10.1.7 netmask 255.255.255.255 0 0
static (inside, dmz4) 10.10.1.7 10.10.1.7 netmask 255.255.255.255 0 0

!---Map IP address of DNS host to all DMZ's.
static (inside, dmz1) 10.10.1.20 10.10.1.20 netmask 255.255.255.255 0 0
static (inside, dmz2) 10.10.1.20 10.10.1.20 netmask 255.255.255.255 0 0
static (inside, dmz3) 10.10.1.20 10.10.1.20 netmask 255.255.255.255 0 0
static (inside, dmz4) 10.10.1.20 10.10.1.20 netmask 255.255.255.255 0 0

!---Map IP address of Proxy Server to External Network
static (inside, outside) 12.24.13.22 10.10.1.25 netmask 255.255.255.255 0 0

!---Map IP address of Internal DNS  to External Network to receive forwarded queries
static (inside, outside) 12.24.13.23 10.10.1.20 netmask 255.255.255.255 0 0

pager lines 24

!--- Logging sent to the Internal Syslog host.
logging on
logging timestamp
logging buffered informational
```

© SANS Institute 2003, Author retains full rights.

Exhibit B
Network Diagram

```
logging trap informational
logging history debugging
logging host inside 10.10.1.7

!--- Enabling the appropriate interfaces
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
interface ethernet3 100full
interface ethernet4 100full
interface ethernet5 100full
interface ethernet6 disabled
interface ethernet7 100full

!--- IP addresses assigned to each interface
ip address outside 12.24.13.2 255.255.255.0
ip address inside 10.10.1.1 255.255.0.0
ip address dmz1 10.1.1.1 255.255.255.0
ip address dmz2 10.2.1.1 255.255.255.0
ip address dmz3 10.3.1.1 255.255.255.0
ip address dmz4 10.4.1.1 255.255.255.0
ip address failover 192.168.1.1 255.255.255.0

!--- Log and alert on basic attack signatures recognized by the PIX
ip audit info action alarm
ip audit attack action alarm

!--- These commands apply the Access-lists to the appropriate Interfaces
access-group outside_in in interface outside
access-group DMZ1_in in interface DMZ1
access-group dmz2_in in interface dmz2
```

SANS Institute 2005, Author retains full rights.

Exhibit B
Network Diagram

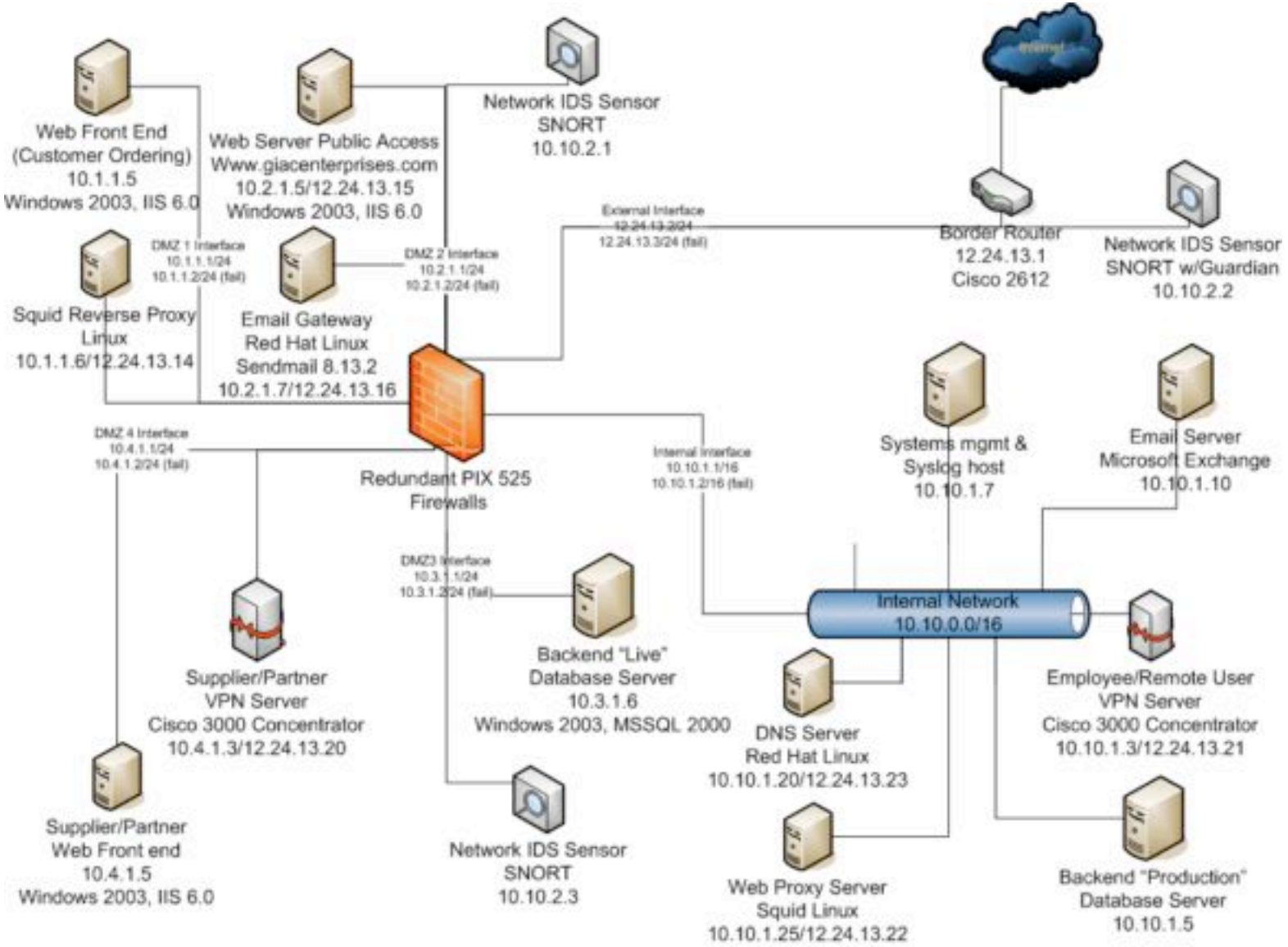
```
access-group dmz3_in in interface dmz3
access-group dmz4_in in interface dmz4
access-group inside_in in interface inside


!---Routing to connected networks is implied and does not need to be stated, however the default route to the internet and the route to
!---  internal company networks must be here to allow communication.
route outside 0.0.0.0 0.0.0.0 12.24.13.1 1
route inside 10.0.0.0 255.0.0.0 10.10.254.254 1

!--- Allows IPSec traffic to pass through the PIX Firewall
!--- and does not require an additional conduit
!--- or access-list statements to permit IPSec traffic.
sysopt connection permit-ipsec
```

© SANS Institute 2005, Author retains full rights.

Exhibit B
Network Diagram



Notes

[1]Del Carlo, Corbin, "Intrusion Detection Evasion, how attackers get past the burglar alarm" SANS Reading Room, December 2003, 25 Sept. 2003 <http://www.sans.org/rr/whitepapers/detection/1284.php>
[2]Del Carlo, Corbin, "Intrusion Detection Evasion, how attackers get past the burglar alarm" SANS Reading Room, December 2003, 25 Sept. 2003 <http://www.sans.org/rr/whitepapers/detection/1284.php>
[3]"Intrusion Detection Exchange Format" 11 Nov. 2004 <http://www.ietf.org/html.charters/idwg-charter.html>
[4] B. Feinstein, G. Matthews, and J. White, "The Intrusion Detection Exchange Protocol (IDXP)" 22 Oct. 2002, 25 Dec. 2004 <http://www.ietf.org/internet-drafts/draft-ietf-idwg-beep-idxp-07.txt>
[5]Herve Debar and Andreas Wespi, "Aggregation and Correlation of Intrusion-Detection Alerts" 25 Dec. 2004, < http://perso.rd.francetelecom.fr/debar/papers/DebWes01.pdf>
[6] Herve Debar and Andreas Wespi, "Aggregation and Correlation of Intrusion-Detection Alerts" 25 Dec. 2004, < http://perso.rd.francetelecom.fr/debar/papers/DebWes01.pdf>
[7] Robert P. Goldman, Walter Heimerdinger, Steven A. Harp, Christopher W. Geib, Vicraj Thomas, and Robert L. Carter, "Information Modeling for Intrusion Report Aggregation" 25 Dec. 2004, <http://www.cc.gatech.edu/~wenke/ids-readings/ Robert_Golderman_Intrusion_Report_Aggregation.pdf>
[8] "Cisco AutoSecure Data Sheet" 25 Dec. 2004 <http:// www.**cisco**.com/warp/public/ cc/pd/iosw/prodlit/cas11_ds.pdf>

© SANS Institute 2005, Author retains full rights.

© SANS Institute 2005 Author retains full rights.