Global Information Assurance Certification Paper

GIAC Training and Certification

SANS Sydney
April 2001

Level Two

Firewalls, Perimeter Protection and VPNs

Version 1.5a

Con Tisci

Assignment 1 - Security Architecture (25 Points)

Business Requirements

GIAC sell bulk, online fortune cookie sayings to th e public. To achieve this they use a WEB server to catalogue and sell these fortunes. GIAC will use an FTP server to deliver the bulk fortunes. These fortunes will be compressed and encrypted. The WEB server uses an authentication database server in a secu re environment to authenticate users.

The secure environment contains all of GIAC's transaction servers that will connect to financial institutions for account payments. The secure network also contains the offline fortune stores.

GIAC has a partnership agreement to provide a VPN to its partners to access its offline fortune store to be able to translate them. Some partners have requested dedicated lines to their servers. The fortune stores can FTP encrypted fortune packages to the online FTP server where customers can collect and decrypt them.

Suppliers can deliver bulk fortunes by VPN access directly to the store. Remote sale and corporate staff will also have VPN access.

GIAC have provided us with a Checkpoint firewall that was purchased for a previou s project. The rest of the infrastructure is to be purchased new.

The solution must be cost effective and flexible enough to expand easily if required.

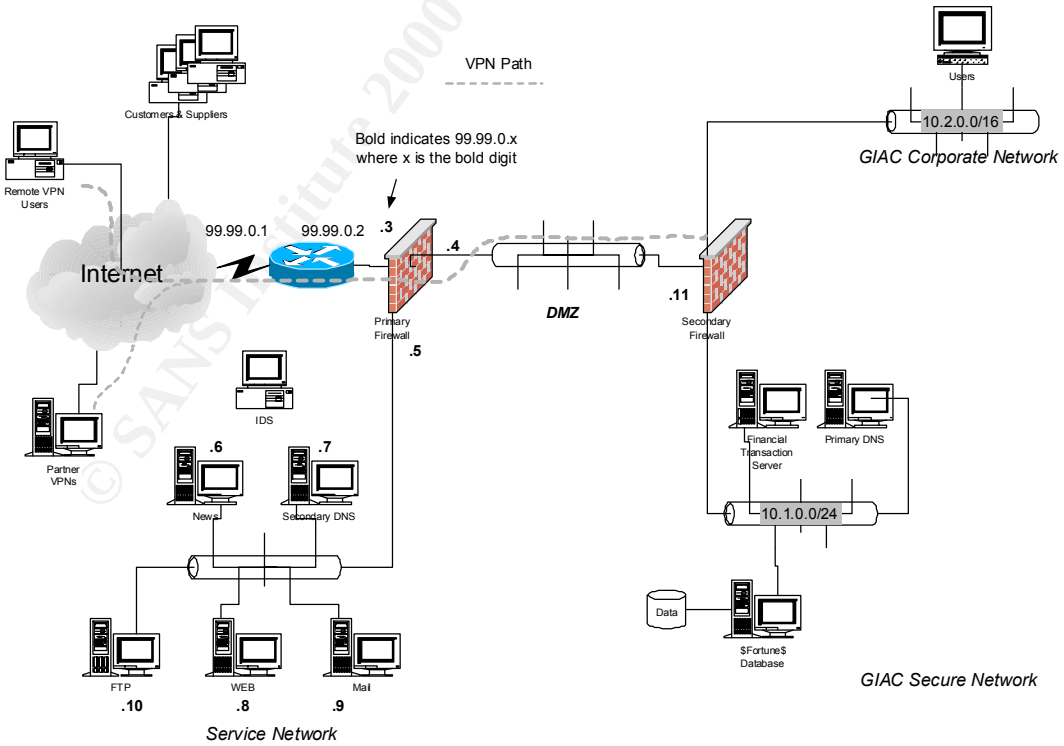


Figure 1 : Overview

Figure 1. shows an overview of the GIAC infrastructure.
Overall I have tried to make the structure of the network as simple as possible. Simple is good because there is less likelihood of things going bad if there are less things that can go wrong. Also, paranoia is good, but we don't get so paranoid that at the end of the day only one host with one port is able to communicate through our secure infrastructure. Its all about the business needs and what risks the business is prepared to take.

The Border router

From the internet we have as our first line of defence our border router. At this point we will require a router that can scale reasonably well, as well as being cost effective. In the event our traffic profile should suddenly jump up we can place another 3640 as a hot standby and load balance between the two.

The choice here is a **Cisco 3640 series router with IOS 12.1** with the basic IP IOS code.

The border router will provide the basic ruleset for entry and exit of the network. Here we can filter out most of the "noise" packets that inevitably will hit our router. We use the router to do most of this grunt work as it really is just copying packets from one interface to another. That's what routers do best and we should aim not to force the router to make too many decisions unless we really have to. * Here we will apply a default DENY ALL policy inbound and permit what we need. Extended access lists will be used here rather than Reflexive access lists for speed.

The "deny all" policy will effectively block out anything that we don't allow. This means that it should block all the most common exploits that rely on the existence of well-known ports running on servers eg. Netbios. By only allowing ports that we know of we hopefully protect ourselves by exploits known and unknown.

* Gotcha Note : Remember that we if have an explicit deny all policy, anything that is not defined will not get through, that includes control traffic like ICMP and router updates. Don't forget this.

Primary Firewall

This is the first stage of isolation between our inner network and our outer network. This firewall will provide the primary means of filtering traffic between our inner networks. This firewall has three (3) interfaces that connect the outer network to our two inner networks. The choice here is a **Cisco 520 PIX firewall**, version 5.3. As Cisco put it (http://www.cisco.com/go/pix). , **"Cisco Secure PIX 520** is intended for large enterprise organizations and complex, high-end traffic environments. It also has a throughput of up to 370 Mbps with the ability to handle 250,000 simultaneous sessions.", so it should scale reasonably if required.

The primary firewall handles three interfaces. The first interface accepts traffic from the border router, the other two interfaces handle traffic for our two internal networks.

The Service Network.

The service network holds all the services we want to provide to the outside world. These include WEB, Mail, News and FTP. The service network is the least protected of our networks. This is because the business needs to have these sorts of services easily available and with as few restrictions as possible because of the very nature of their operation.

We will to some extent expect these to be probed and possibly exploited. In the event they are exploited we want to contain the damage as much as possible, hence their location on a separate network. Although this is something we would rather not happen we should be prepared to accept the fact that these servers will be the primary targets in an attack and we can only do so much to protect them.

The DMZ

The de-militarised zone (DMZ) is the link between our two firewalls.

We could have used the primary firewall to provide the same functionality at this point but we have split the firewall functionally he re for several reasons.

1. The primary firewall will apply some stateful filtering rules and will most likely have a lot of traffic going through it. Limiting its functionally to tracking connections and applying more sophisticated filtering will keep it bus y enough.
2. In the event that the primary firewall is disabled or breached we can continue to communicate with the rest of the business and our partners.
3. The secondary firewall provides an additional obstacle if an intruder usefully exploits our primary fire wall, particularly if the secondary firewall is of a different type. We have doubled our intruder's workload.

Secondary Firewall

The secondary firewall separates our backend networks. Whilst we need to communicate with our partners we still don't necessar ily trust them with our systems. This firewall provides our VPN functionality as well as our Network Address Translation to the corporate network. The choice here is **Checkpoint Firewall-1 version 4.0** running on an SUN Enterprise 450 server.

We will use the "hiding" IP address function to Firewall -1 to hide all the private addresses behind one public address. Firewall -1 handles the connections from there.

Switches

Switches are the backbone of our infrastructure. Switches are targets, they have their weaknesses just as any other device in our network. Because of this the firewalls should sit on their own separate switch domains. Ideally each network should sit on its own switch but cost considerations should be taken into account and VLANs used instead. The following from Reference 2. should be observed.

- Ports without any need to trunk, should have any trunk settings set to off, as opposed to auto. This prevents a host from becoming a trunk port and receiving all traffic that would normally reside on a trunk port.
- Make sure that trunk ports use a virtual LAN (VLAN) number not used anywhere else in the switch. This prevents packets tagged with the same VLAN as the trunk

port from reaching another VLAN without crossing a Layer 3 device. For more information, refer to the following URL : http://www.sans.org/newlook/resources/IDFAQ/vlan.htm

- Set all unused ports on a switch to a VLAN that has no Layer 3 connectivity. Better yet, disable any port that is not needed. This prevents hackers from plugging in to unused ports and communicating with the rest of the network.
- Avoid using VLANs as the sole method of securing access between two subnets. The capability for human error, combined with understanding that VLANs and VLAN tagging protocols were not designed with security in mind, makes their use in sensitive environments inadvisable.

Assignment 2 - Security Policy (25 Points)

The Border router

Some of the things we should do here are : -

- Disable telnet on the front facing interface.
  - This stops any one from attaching to the router and trying to g ain access to passwords. Out of band management is secure but awkward. If we define telnet to be available on the inner interfaces we will need to apply access lists and define timeouts.
- Disable source routing.
  - This stops anyone from overriding how the r outer would normally route a packet.
- Restrict SNMP access to the router.
  - and ensure community names are set to other than defaults.
- Control access to the router using Tacacs+ or Radius.
  - Maintain a centralised and easy to change database of users.
- Turn off uneeded services.
  - This prevents some of the more common and easy attacks.
- Turn on logging at appropriate levels (log what **you** are interested in)
  - You should know what is being thrown at you.
- Authenticate routing updates.
  - Don't let any old hacker f ool around with your routes.

Its important that we take extra care in ensuring our router is secure. Routers are potentially hackers' best friends.

As indicate before we have placed an implicit "deny all" rule

We then apply the Access Control Lists to d efine what we will allow into our networks.

The order in which these rules are entered is important. If we know that there will be more http traffic than ftp traffic then we place this ACL first. This is because as soon as an ACL match is found the router will stop processing rules.

Here is a note from http://www.networkcomputing.com/907/907ws13.html by Peter Morrissey.

> **Performance Issues** Access list filters exact a toll on router perform ance. Some of the performance-enhancing features Cisco has built into its routers will not work when access lists are used. As a result, features such as fast switching, autonomous switching, distributed switching and optimal switching will not be utilized , forcing many of the packets to be process -switched. This can burden your router's main CPU. You will want to keep an eye on the router's CPU utilization using the "show process CPU," command and watch for packets dropped at the interface. ….
> A more obv ious issue is that the longer your access list, the more work your router will have to perform every time a packet has to be processed. The size of the list probably will not hurt you quite as much as the performance penalties we just described, but you sh ould try to locate the most -likely matches at the top of your lists.

One way to ensure that most of the incoming packets are matched on the first line of the list is to put in a rule that allows all TCP established or ACK (acknowledged) packets. Established packets are those that are the result of an already established session. Generally, the majority of network packets are established. Because it's very unlikely that these packets can be harmful, even if they are spoofed, you may want to consider letting them all in right off the bat. You might want to try your access list with and without this rule, and observe changes in the CPU utilization. The following statement at the top of your access list allows all established packets:

```
access-list 109 permit  tcp any any established
```

Note : Lines beginning with an exclamation mark "!" are comments

Border router access lists

```
! **** Cisco 3640 Border Router ***
!
! -- begin access list  -
!
! We first need to filter out the bogus addresses
! These are all private  IP ranges and don't belong on the internet
!
access-list 109 deny ip 127.0.0.0 0.255.255.255 any log
access-list 109 deny ip 10.0.0.0 0.255.255.255 any log
access-list 109 deny ip 172.16.0.0 0.240.255.255 any log
access-list 109 deny ip 192.168.0.0 0.0.255 .255 any log
!
! We should not see our own addresses coming back in either
!
access-list 109 deny ip 99.99.0.0 0.0.255.255 any log
!
! This next ACL is optional  - as discussed above
! Note : A smart hacker will use ACK scans to get by this next rule,
! since any device I place on the service network will be visible
! anyway I don't really care. I will be relying on the primary and
! secondary firewall to block out things like this.
!
! access-list 109 permit tcp any any gt 1023 established
!
! Allow WWW,  Mail and News to those servers only
! - tcp 443 allows SSL https communications
!
access-list 109 permit tcp any host 99.99.0.4 eq www
access-list 109 permit tcp any host 99.99.0.4 eq 443
access-list 109 permit tcp any host 99.99.0.5 eq smtp
access-list 109 permit tcp any host 99.99.0.6 eq nntp
!
! Allow PASV ftp to the FTP server only
!
access-list 109 permit tcp any host 99.99.0.8 eq ftp
!
! This next rule allows traffic already established from out network
! outbound.
access-list 109 permit tcp any 99.99 .0.0 0.0.255.255 gt 1023 established
!
! Allow DNS queries but not zone transfers
!
access-list 109 permit udp any host 99.99.0.7 eq domain
!
! Allow VPN connections to our gateway firewall
```

```
!
! We only need to allow access to the VPN gateway.
!
access-list 109 permit udp any eq 500 host 99.99.0.11 eq 500
access-list 109 permit 50 any host 99.99.0.11
access-list 109 permit 51 any host 99.99.0.11
!
!
! Allow pings, we can block the rest at the secondary FW
access-list 109 permit icmp any 99.99.0.0 0.0.255.255   echo
access-list 109 permit icmp any 99.99.0.0 0.0.255.255 echo  -reply
access-list 109 permit icmp any 99.99.0.0 0.0.255.255 packet  -too-big
access-list 109 permit icmp any 99.99.0.0 0.0.255.255 time  -exceeded
access-list 109 permit icmp any 99.99.0.0 0.0.25  5.255 traceroute
access-list 109 permit icmp any 99.99.0.0 0.0.255.255 unreachable
! Finally log anything that violates our list
access-list 109 deny ip any any log
! - end access list  -
```

To test the above rules we can use nmap and scan the whole network. We also throw in the invalid IP addresses and check the logs for them. The returned scan should show the open ports and the rest should show up as filtered. A command to do this would be

```
nmap -sS 99.99.0.0/24  -D 10.0.0.1, 127.0.0.3, 172.16.1.2, 192.168.1  .2, ME
```

The parameters are as follows from left to right.

```
nmap :              The executable program name.
-sS :               Perform a SYN only scan, half open connection (its
faster).
99.99.0.0/24 :      This is the service network, the only network we should
see.
-D :         The list that follows are decoy addresses that are inserted
into the source IP, ME is my real IP address so that some results get
back to me.
```

Primary firewall

Its crucial to understand how the PIX firewall works. A good set of examples can be found at
http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v53/config/examples.htm#xtocid484

Interfaces are assigned security levels. Traffic going fr om a lower security level to a higher security level is denied unless allowed by access lists and ends in an implicit deny all rule.
** Traffic going from a higher level to a lower level is allowed unless restricted by access lists and ends in an implicit permit all rule.

> ** Gotcha Note : The permit all means that if you don't match it, it goes through, so remember, if you want to apply some egress rules put a "deny any any" rule at the end.

```
! **** Cisco Secure PIX 520 Primary Firewall ****
!
! -- begin interface spec  --
!
nameif ethernet0 outside security0
```

```
nameif ethernet1 service security50
nameif ethernet2 inside security100
!
interface ethernet0 100basetx
interface ethernet1 100basetx
interface ethernet2 100basetx
!
ip address outside 99.99.0.3 255.2  55.0.0
ip address inside 99.99.0.4 255.255.0.0
ip address service 99.99.0.5 255.255.0.0
!
! -- end interface spec  --
!
! -- begin access lists  --

! We now define and apply ACLs to the interfaces inbound
access-group acl_from_outside in interface outside
access-group acl_from_service in interface service
access-group acl_from_inside in interface inside
!
! Name some important locations
name 99.99.0.2 border_router
name 99.99.0.6 service_news
name 99.99.0.7 service_dns
name 99.99.0.8 service_www
name 99.99.0 .9 service_mail
name 99.99.0.10 service_ftp
! name 99.99.0.11 inner_firewall
name 99.99.0.11 nat_GIAC_net
!
! -- begin PIX access list  -
!
! ** What we allow in from the internet to our service network **
!
access-list acl_from_outside permit tcp any host   service_www eq www
access-list acl_from_outside permit tcp any host service_www eq 443
access-list acl_from_outside permit tcp any host service_mail eq smtp
access-list acl_from_outside permit tcp any host service_news eq nntp
! Only query to the service  DNS.
access-list acl_from_outside permit udp any host service_dns eq domain
access-list acl_from_outside permit tcp any host service_ftp eq ftp
! VPN access to the gateway only
access-list acl_from_outside permit udp any host nat_GIAC_net eq 500
access-list acl_from_outside permit 50 any host nat_GIAC_net
access-list acl_from_outside permit 51 any host nat_GIAC_net
!
! There is an implicit deny all at the end so we don't need
! anything else.
!
! Gotcha Note : ACLs from the inside are permitted by default
! We lock down this ruleset with a deny any any at the end
! We decide on what we will allow out first.
!
! ** What we allow out to the internet from our corporate network **
!
! We scan on this server for viruses. The Mail, News and
! Domain Name Server  are the only machines allowed to access
! the external networks as well as the private networks.
!
access-list acl_from_inside permit ip nat_GIAC_net any
! We are allowing everything out
```

```
access-list acl_from_inside deny ip any any
!
! ** What we allow from  our service network out to the internet. **
!
! Q. Do we really need our WWW servers issuing http, ftp etc requests?
! A. No, if this is happening, we may have a compromised server.
! So we put an explicit "deny all" at the end and let out what we need.
access-list acl_from_service permit tcp host service_news any eq nntp
access-list acl_from_service permit tcp host service_mail any eq smtp
access-list acl_from_service permit udp host service_dns any eq 53
!
! Close off the service ACL
access-list acl_from _service deny ip any any
```

VPN - Secondary Firewall

Here, Checkpoint Firewall -1 has been chosen as the secondary firewall and VPN gateway.

Remote users will use the SecurRemote client supplied by CheckPoint to access the GIAC corporate network using ESP tu nnelling.

Gotcha Note : We now have devices connecting to one of our trusted networks. We cannot guarantee the integrity of remote user machines (unless we follow a strict regime of building them). If one of these machines is compromised we have opened ou r network to attack.

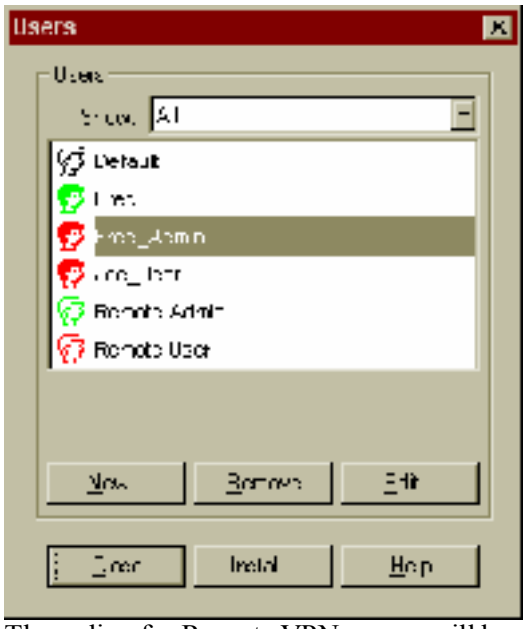
As part of the corporate security policy it will be stipulated that a personal firewall should be used at the client end. A good choice here would be ZoneAlarm.

ZoneAlarm setup is quite simple. Install and run. Establish a connection with the VPN and ZoneAlarm will alert with a dialog box. Check the "remember this connection" checkbox. That's it.

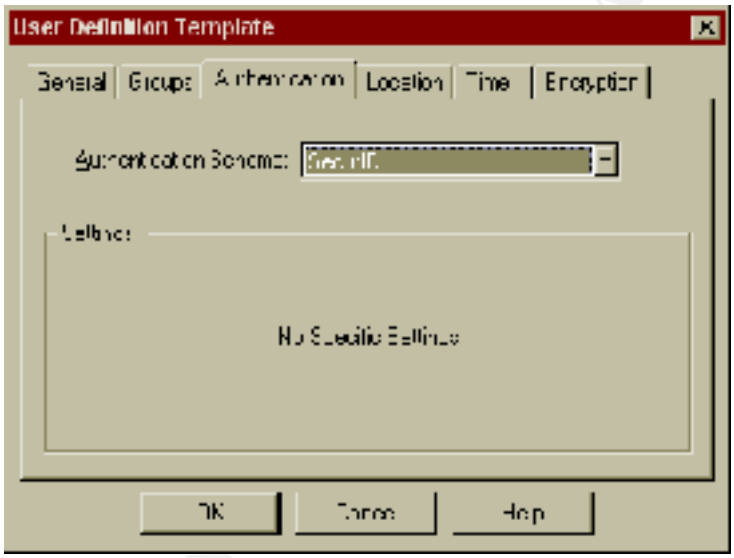
We want to ensure that communications are secure up to the gateway. This way we are still protected if an intruder has breached the service network and placed a sniffer somewhere outside the gateway. Additionally we can ensure data confidentiality even from our own IDS if required. We will choose ESP to do this.

Firewall-1 makes it very easy to set up and maintain VPNs and can work with seve ral encryption standards. I have chosen IPSec since it is a standard. We have available to us Checkpoint Firewall-1 Version 4.0. Firewall-1 has its own proprietary encryption schemes also.

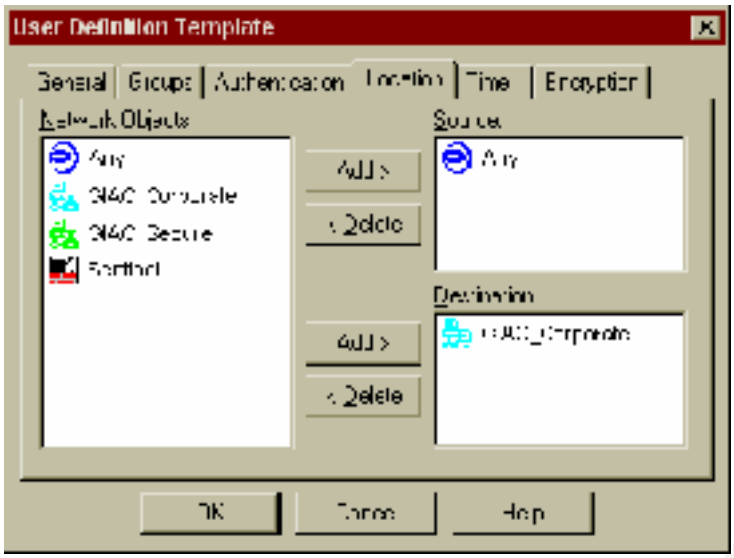
We will set up VPN access for our remote users and administrators . First we define a template for our remote users. This ensures we have consistency across users. We create a Remote users template and a Users and Admins group.



The policy for Remote VPN access will be as follows.
Authentication Method : Secure ID Token

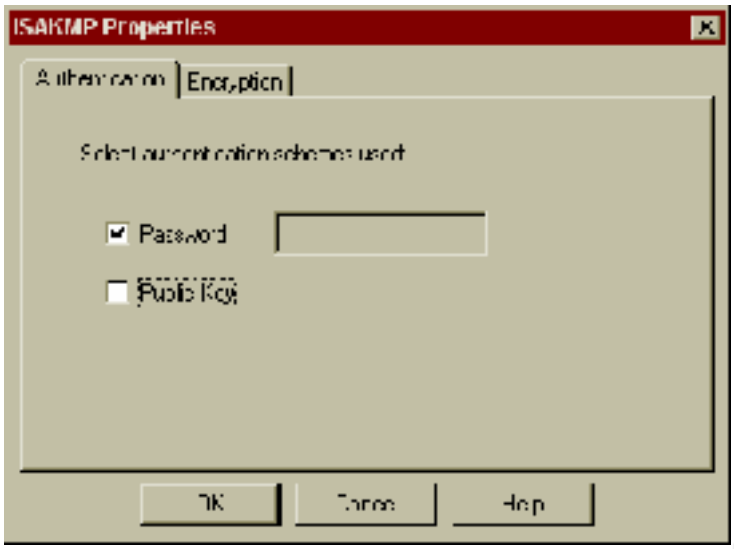


We will restrict normal users to the GIAC corporate network.

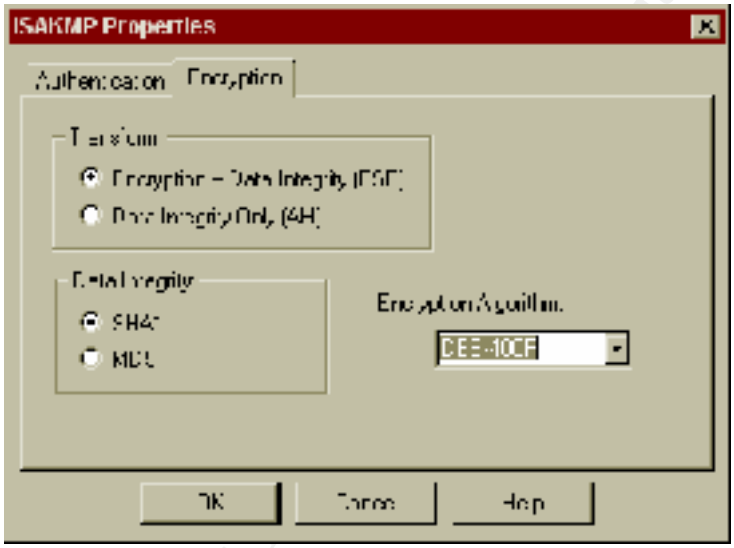


There are no time restrictions.
We will use ISAKMP/OAKLEY encryption.

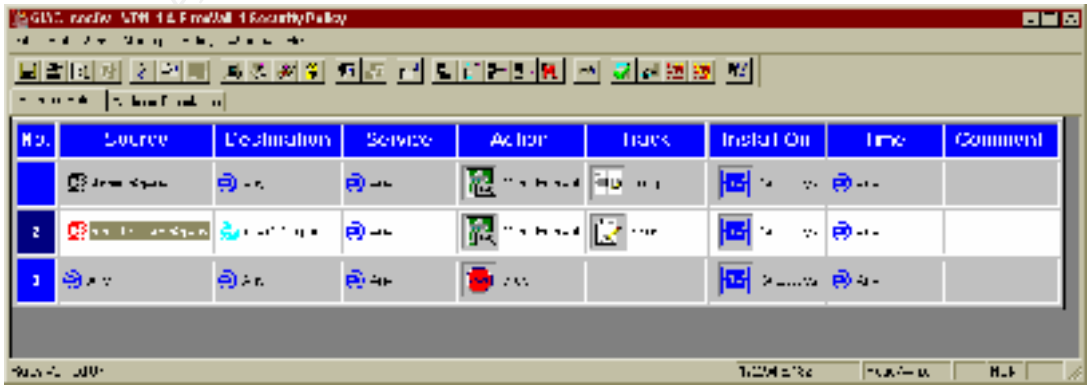
We will use a password only for now.



We will ESP with SHA1 and DES -40CP. This version of Firewall -1 only has this available. Firewall-1 supports 3DES and this will be used on the final production firewall.



We now build up the policy in the policy editor required to allow our users access.



We have given our "Remote Users" group VPN access to the GIAC corpo rate network only. Our security policy forbids access to the secure network by normal users. We have an "Admin" group that is allowed everywhere to manage the network.

We similarly create a Partner_VPN template and add an additional Partner VPN user afte r rule 2.

We complete the firewall setup with the following rules with the following defined objects.

Object	**Description**
Net_Srv	The service network
Net_Out	The internet
Net_Crp	The GIAC corporate network
Net_Sec	The GIAC secure network
XXXX_Crp	A specific server on the GIAC network (XXXX=Accts, Maint etc)
XXXX_Sec	A specific server on the secure network (XXXX=Accts, Maint etc)
XXXX_Srv	A specific server on the services network (XXXX=Mail, News etc)

No	**Source**	**Destination**	**Service**	**Action**	**Track**	**Install On**	**Time**	**Comment**
4	XXXX_Crp	Ftp_Sec	ftp	Allow	Long	Gateways		
5	XXXX_Crp	Mail_Sec	smtp	Allow	Long	Gateways		
6	Net_Crp	DNS_Sec	domain-udp	Allow	Short	Gateways		
7	Any	Net_Sec	Any	Drop	Long	Gateways		
8	DNS_Sec	DNS_Srv	domain-udp	Allow	Short	Gateways		
9	Net_Sec	Ftp_Srv	ftp	Allow	Short	Gateways		
10	Net_Sec	Mail_Srv	smtp	Allow	Short	Gateways		
11	Net_Sec	News_Srv	nntp	Allow	Short	Gateways		
12	Net_Sec	Any	Any	Drop	Long	Gateways		
13	Net_Crp	Any	http	Allow	Short	Gateways		

Rules 4 to 5 allows specific nominated hosts access to the secure network, we may need to allow our accounts staff access to the financial transaction servers etc.
Rule 6 allows corporate users access to our secure DNS s erver. We have split DNS.
Rule 7 drops everything we have not allowed for.

Rules 8 to 12 similarly confine access from the secure network into the service and outside networks to services and hosts that its needs to know about.

Rule 13 only allows web tr affic out from corporate network.

Assignment 3 - Audit Your Security Architecture (25 Points)

We now move on to the auditing phase of this exercise. We now determine if the policies that are in place are in fact performing the functions set out in our pol icy.

Assessment planning

There are many considerations when planning and executing an audit.

Overall, we need to define the scope of the audit. Here we define what the boundaries are. Unless we do this we end up auditing too much and are overwhelmed with information. In this case the scope of the audit is confined to the primary firewall.

Secondly, we need to decide who will do the audit. An external party will have different methodologies and tools to what we would normally use. Its likely that they will find something you had not planned for. However, these approaches are usually very costly. We will assume that we have a limited budget and so will be doing the audit ourselves.

We can split the audit up into two phases, a non -intrusive and an intrusive.

Non intrusive audit

This audit will not impact on the business. We do the following here.

1. Review Network Architecture with all major stakeholders.
    a. Does the network architecture meet the business needs?
    b. Do all stakeholders understand any risks involved?
        i. VPNs extend our private networks out to the partners and remote users, who do we trust with what?
2. Review policies and procedures in place for.
    a. Operating System and Application builds.
    b. User id and Passwords.
    c. Acceptable use.
    d. Incident handling.
    e. Remote access.
    f. External party connection.
    g. Monitoring and alarming.
    h. Disaster recovery.
    i. Physical security.
3. Assemble the target list to audit.
4. Assemble audit items per target.
5. Estimate audit resources required per target and derive costs.

Intrusive audit

The intrusive audit is where we start to throw packets at our targets.
Next we decide on what we should audit on our list from point 4 above based on the business needs. For example, do we need to test for denial of service attacks?

Given this, we determine if there may be any detrimental effect on the systems that we are auditing. Given that some of the policies we have implemented are put in place for anti -

spoofing and denial of service we should endeavour to perform the test out of normal business hours and notify the re levant stakeholders that services may be affected and agree on a time. Additionally some form of indemnity should be discussed prior to performing the test in case damage to systems occur as a result of the tests.

For this part we will require the follow ing :
1. A PC connected to a public internet account, preferably cable or ADSL.
2. A PC connected to the DMZ and one connected to out service network.

The following software tools should be loaded onto the system.

Linux Operating System with kernel 2.2.13 or greater.

The linux operating system provides us with a wealth of tools and services that we can use to probe and test our network, and its all free. Apart from the usual ping, traceroute and other utilities the following should be loaded on the system.

Nmap scanning tool

Nmap is a very powerful and simple tool that can craft IP packets and scan entire networks. The tool can scan a network in a short time and can provide a wealth of information on how particular packets are handled with in the network.

Nessus Vulnerabilty Scanner

Although Nessus provides a scanning tool as part of its standard setup, its most powerful feature is the ability to probe applications for vulnerabilities. Another useful feature of Nessus is the way in which it scans the applica tions. Nessus probes every port it finds. It makes no assumptions about what ports should be running what service. If it finds port 100 open it probes that port for WEB servers, FTP servers, anything that can get e response.

PERL

When it starts to get down to the nitty gritty and you want to start writing some script to crack a port, PERL is usually the hackers tool of choice.

Assessment Implementation

We should bear in mind that we will see the packet filter performing some of its functions and such will not affect the firewall. As we perform our probes and tests we should also inspect our logs and have tcpdump running on our internal PCs to ensure that the border router and primary firewall are doing their job. That is, we should not see any packets get through that are not meant to get through.

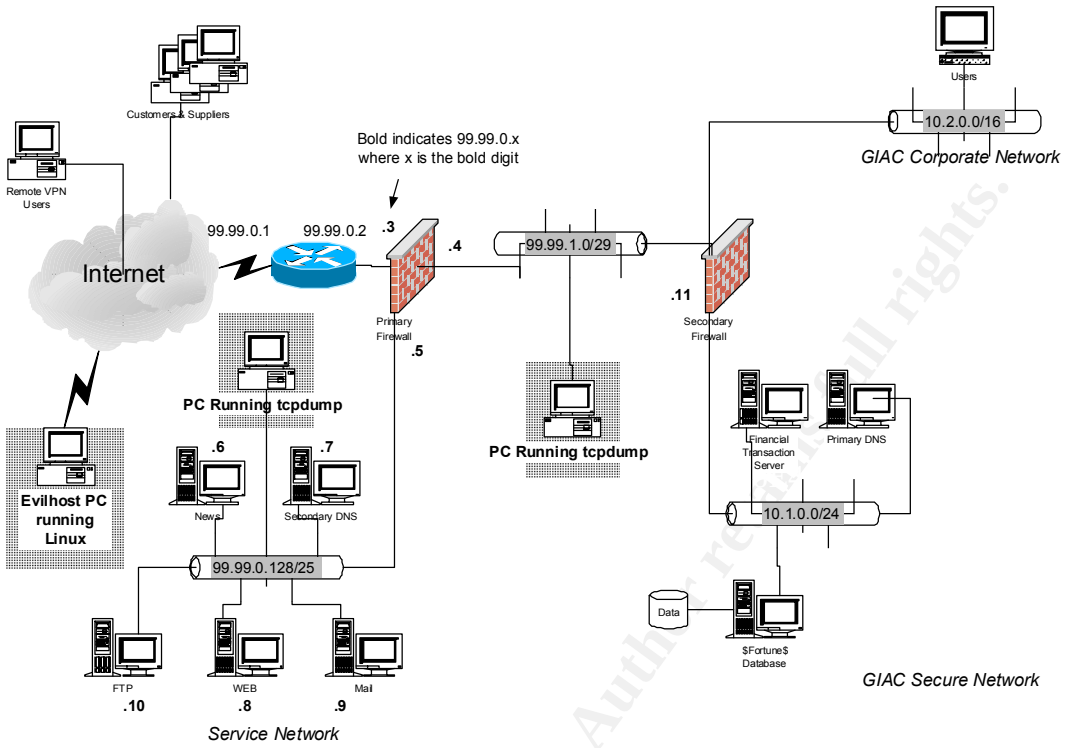


Figure 2 : Placement of audit tools.

1. Try to map out the network

We begin the intrusive part of the audit by mapping the way to our target. We choose the WEB server to check the route.

```
traceroute www.giac.com

traceroute to www.giac.com (99.99.0.8), 30 hops max, 38 byte packets
 1  evilhost_023.myisp.com (1.1.1.23)  16.620 ms  11.338 ms  19.804 ms
 2  gigabit.router.com (61.9.193.4)  16.284 ms  10.67  3 ms  12.765 ms
 3  11.18.76.3 (11.18.76.3)  9.908 ms  9.787 ms  25.079 ms
 4  Ethernet.local.net (13.10.71.92)  10.273 ms  17.923 ms  10.633 ms
                                    .
                                    .
                                    .
18  GIAC-gw1.com (99.99.0.1)  195.772 ms  188.509 ms  199.321 ms
19  GIAC-fw.com (99.9 9.0.3)  200.118 ms  184.714 ms  202.655 ms
20  www.giac.com (99.99.0.8)  211.260 ms * *
```

RESULT : We can see our border router GIAC -gw1.com and firewall GIAC - fw.com.

2. Probe the discovered network

We now fire up nmap and scan the entire port range and ad dress range to ensure all ports that should be open are open and all closed ports are closed. This will take an extremely long time, we should have decided on this in our planning phase. Nmap can be set up to ensure that the scans are fairly unobtrusive an d will not adversely affect systems. If time is a factor the number of ports to scan should be set to the nmap default (1523).

We do this tho make sure our ACLs are as they should be.

```
nmap -sS 99.99.0.0/24

Starting nmap V. 2.53 by fyodor@insecure.org (   www.insecure.org/nmap/ )
Interesting ports on www.giac.com (99.99.0.8):
(The 1520 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     open        smtp
80/tcp     open        http
443/tcp    open        https

** we have all the right hosts with the right ports open **

Nmap run completed  -- N IP addresses (N hosts up) scanned in 287 seconds
```

RESULT : We map out all of our hosts in the service network .

The results from above will give us a clear indication of what is allowed and what is not. It also server to discover hosts that are there as well as any hosts with "unusual" open ports.

We now try and access the network beyond what we are supposed to see, the DMZ and the secondary firewall. We can do this by trying another nmap scan but this time we set the scan type to an ACK scan (-sA) and we force the source port to 20 (-g 20), we don't bother pinging (-P0) and target the secondary firewall (99.99.0 .11).

```
nmap -sA -g 21 -P0 99.99.0.11
```

This will hopefully fool the primary firewall into thinking we are an outbound ftp data session as well as trying to get past the firewall with an ACK scan.

Heres an excerpt from the nmap man page describing the ACK s can.

> "ACK scan: This advanced method is usually used to map out firewall rulesets. In particular, it can help determine whether a firewall is stateful or just a simple packet filter that blocks incoming SYN packets. This scan type sends an ACK packet (with random looking acknowledgement/sequence numbers) to the ports specified. If a RST comes back, the ports is classified as "unfiltered". If nothing comes back (or if an ICMP unreachable is returned), the port is classified as "filtered". Note that nmap usually doesn't print "unfiltered" ports, so getting no ports shown in the output is usually a sign that all the probes got through (and returned RSTs). This scan will obviously never show ports in the "open" state."

```
nmap -sA -P0 -g 21 99.99.0.11
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
All 1523 scanned ports on  (99.99.0.11) are: filtered
Nmap run completed  -- 1 IP address (1 host up) scanned in 181 seconds
```

RESULT : We see nothing com e back, the firewall should block everything .
Except for the VPN.

One test of interest is to send packets that attempt to bypass router access lists with fragmented packets. First we send a normal SYN scan, next we fragment them (-f switch in nmap).

```
nmap -sS -p 80,21 www.giac.com

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on www.giac.com (99.99.0.8):
Port       State       Service
21/tcp     filtered    ftp
80/tcp     open        http

Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds

nmap -sS -p 80,21 -f www.giac.com

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on www.giac.com (99.99.0.8):
Port       S tate       Service
21/tcp     filtered    ftp
80/tcp     filtered    http

Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds
```

RESULT : Fragmented packets are treated differently

To complete the au dit we now use Nessus to check over 600 known vulnerabilities. Nessus provides a report in several formats. The resulting report will provide details on fixes that should be applied to any application that requires it. Nessus can check for application vulnerabilities and things such as telnet servers left running on cisco router. A full list of what Nessus can scan for are located at http://www.nessus.org

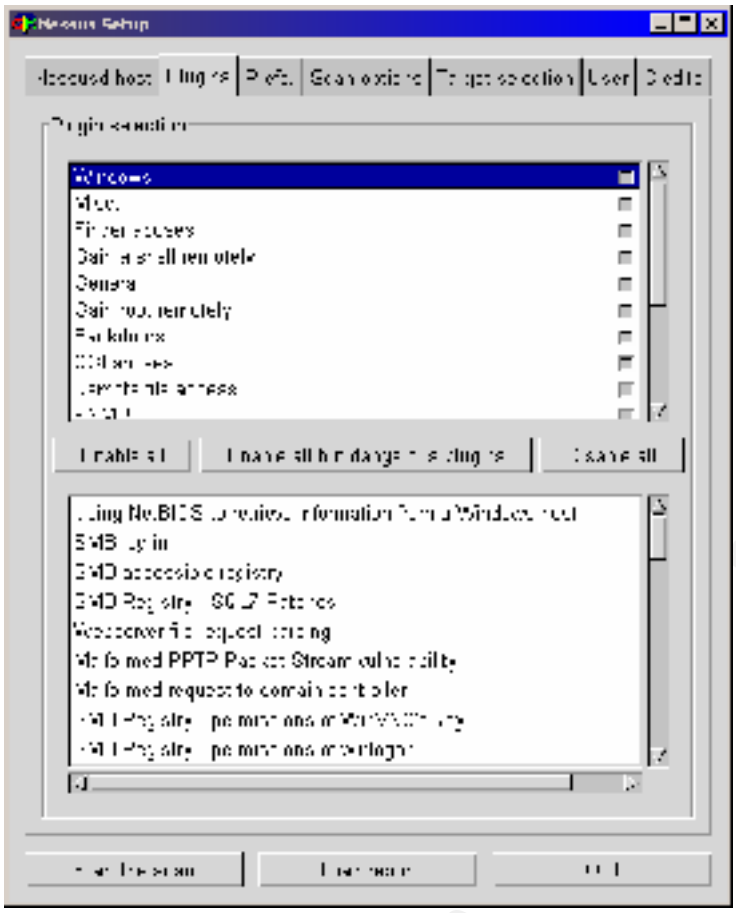


Figure 3 : A Nessus session showing available Windows tests.

Assessment Analysis

We have some results from the audit and can now begin to go over what needs to be done to improve security.

The first result will detail exactly what can be seen from the outside networks. As we can see the border router (GIAC -gw1) and the firewall (GIAC-fw) are clearly identifiable.

The second result shows that we can see the hosts we defined in our access lists along with the appropriate service ports open.

The third result shows our firewall is doing its job.

The fourth result show what happ ens when we try and use unusual packet structures. The result shows us that under normal circumstances the packet is allowed through to hosts and port numbers that are defined. If we now try to fragment the packet to get by the packet filter we see that the router and firewall still block the packet on the illegal port as well as blocking the legitimate port.

Recommendation

We have deliberately opened up our network to specific services. We really can't stop anyone from probing our network to these legitima te ports. We could disable ping and

traceroute, but that just makes it harder for our own network engineers to diagnose problems.

One thing we could possibly do is to delete the DNS records for the router name and the firewall name so that they become jus t an IP address. This makes its harder to determine what is on the perimeter and thus more difficult to identify what is in the perimeter network.

Scans and probes are now par for the course on the internet. However, we can provide a means of knowing when this is done by installing an Intrusion Detection System that will alert us if these scans and probes become excessive.

We should place the IDS sensors in strategic locations in our network.

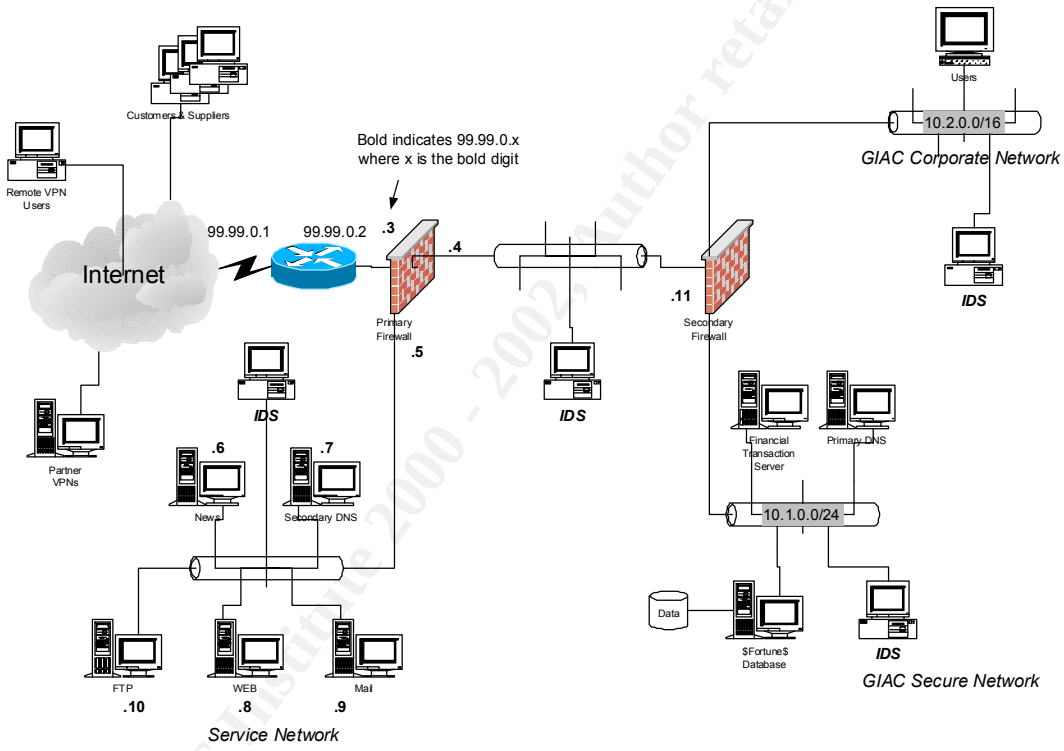


Figure 4 : Network with IDS placem ents

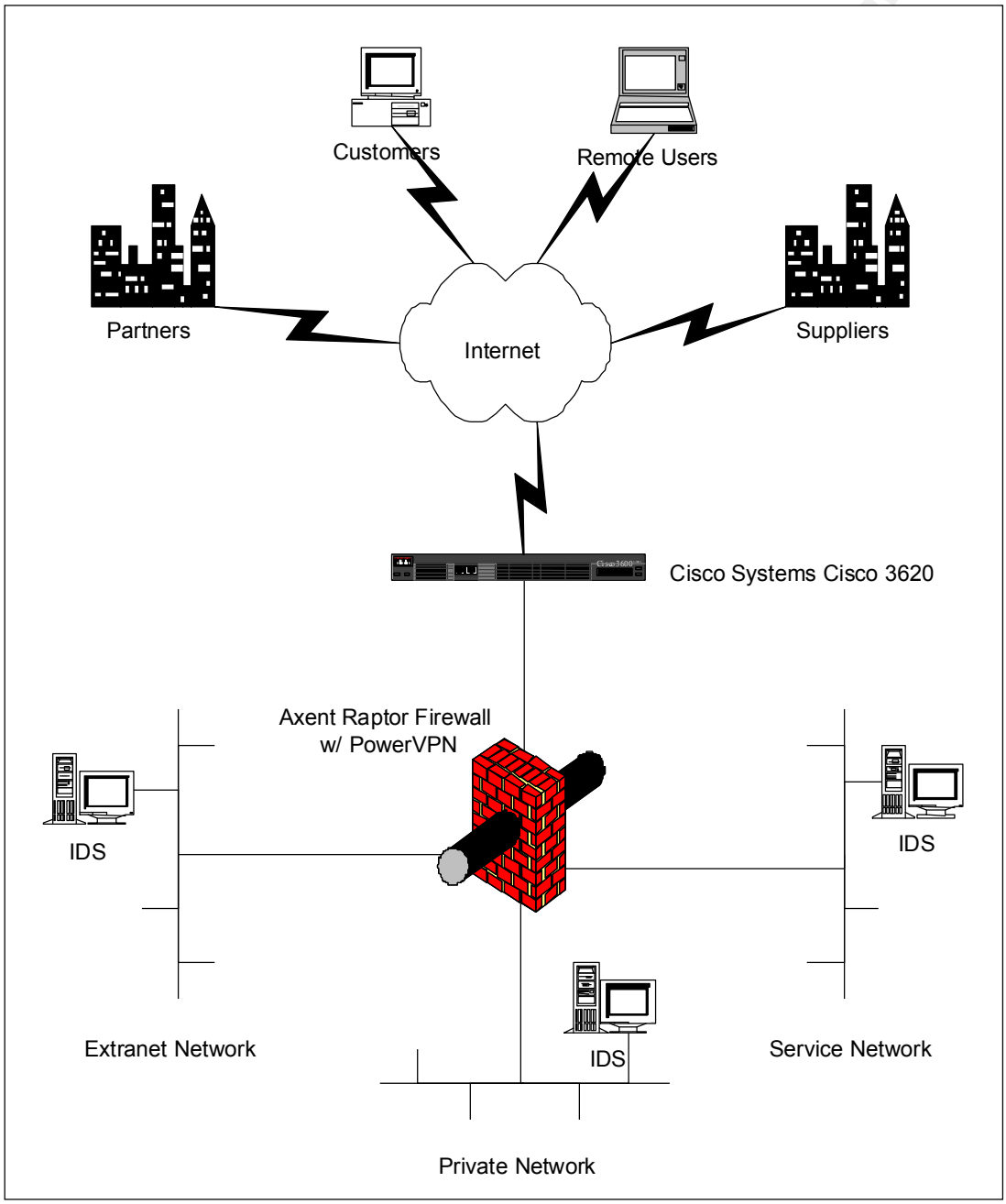
Conclusion

Our network seems to be quite tight in what is allowed to connect to what. We provide security in depth by containing the damage that *could* be done if one or two of the network elements were compromised.

We limit the capability of a hacke r being able to jump from system to system much as possible without affecting the normal business functions.

Assignment 4 - Design Under Fire (25 Points)

I have chosen Dennis Webb's design to attack. To Dennis's credit, I had found this particular vulnerability after he had designed the solution. Here we have an Axent Raptor firewall v6.5 behind a Cisco router.



We know the network architecture so we won't go and "ping" and "traceroute" to see what's there.

Attacking the application

The Cheap shots - Denial of Service (DOS)

Since the prime targets WEB, FTP etc are behind a router and firewall I doubt that flooding those services with SYNs will be very effective, as both Cisco and Raptor will have mechanisms in place to protect against these.

If we want to do a DOS we need to find out what is running on the services network and hopefully find an exploit that will down the service with a few well -crafted packets.

Fortunately we have open access to the relevant ports. Lets try th e WEB server first. For this we use a nifty PERL script from rainforest puppy – whisker.pl. Whisker is an extremely fast and lightweight CGI scanner. Nessus can do similar scans, however I found whisker to be fast and simple.

```
./whisker.pl  -h www.giac.com
-- whisker / v1.4.0+SSL / rain forest puppy / www.wiretrip.net   --

= - = - = - = - = - =
= Host: www.giac.com
= Server: Microsoft -IIS/5.0

+ 200 OK: GET /SiteServer/Publishing/viewcode.asp
+ 200 OK: HEAD /_vti_inf.html
+ 200 OK: HEAD /_vti_bin/shtml.dll
+ 200 OK: HEAD /_vti_bin/shtml.exe
+ 200 OK: HEAD /search/
```

Ok, so we now know it's a Microsoft IIS server Version 5 (and interestingly enough we have also found some files that should not be there. The presence of these files indicates that vulnerabilitie s exist to gain some form of privileged access to the server. Lets forget these for now, this part of the exercise is just to find out what server we have at the other end, this scan just demonstrates what we could find, not necessarily what's there.).

We now need to find an exploit that can bring down the server. I find that SecurityFocus http://www.securityfocus.com has an excellent database. There is one, very recent vulnerability for this version of IIS.

bugtraq id	2483
class	Failure to Handle Exceptional Conditions
cve	CVE-MAP-NOMATCH
remote	Yes
local	No
published	March 16, 2001
updated	March 19, 2001
vulnerable	Microsoft IIS 5.0
	+ Microsoft Windows 2000
not vulnerable	

WebDAV contains a flaw in the handling of unusually long requests, submitting a valid yet unusually long WebDAV 'search' request could restart the IIS services and possibly cause the server to stop responding.

The exploit (PERL script) was provided by Georgi Guninski <guninski@guninski.com>:

So by simply executing the script from our evil linux box we may be able to down the WEB server for a significant amount of time.

We can search again for similar attacks on every other service, FTP, SMTP, News etc.

Attacking the firewall

First thing we do is to see if the firewall itself has any vulnerability.

From http://www.securityfocus.com we find the following.

The Description.

bugtraq id	2517
class	Origin Validation Error
cve	CVE-MAP-NOMATCH
remote	Yes
local	No
published	March 24, 2001
updated	March 29, 2001
vulnerable	Axent Raptor 6.5
	- Sun Solaris 7.0
	- Sun Solaris 2.6
	- Microsoft Windows NT 4.0 - HP HP-UX 11.11
	- HP HP-UX 11.0
	- Digital (Compaq) TRU64/DIGITAL UNIX 5.0
	- Digital (Compaq) TRU64/DIGITAL UNIX 4.0g
not vulnerable	

The Discussion

Raptor Firewall is a product distributed and maintained by Axent Technologies, Inc. Raptor is an Enterprise -level firewall, providing a mixture of features and performance.

A problem in the software package could allow intruders access to private web resources. By using the nearest interface of the firewall as a proxy, it is possible to access a system connected to the other interface of the firewall within TCP ports 79 -99, and 200-65535. The firewall will only permit connections to the other side on ports in this range, excluding port 80, and using HTTP. This affects firewall rules that permit HTTP traffic.

Therefore, it is possible for a malicious user to access internal web assets, and potentially gain access to sensitive information. It is also possible for an internal user to gain access to external web resources through the firewall, providing the resources are not running on the default port 80.

The Exploit

Attacker configures browser to use IP address of Raptor firewall as HTTP Proxy, then begins probing internal network.

The Solution

The following workarounds are possible:

1. Use httpd.noproxy in the affected rule.

2. Downgrade to version 6.0.2

Additionally, patches are available:

Axent Raptor 6.5:

Axent hotfix SG6500-20000920-00 and SG6500-20001121-00
ftp://ftp.axent.com/pub/RaptorFirewall/Patches/6.50/Internal/http-int.zip

Credit

This vulnerability was discovered by Benny Amorsen <benny_amorsen@hp.com> and Christian E. Lysel <chlys@wmdata.com> on August 29, 2000, and was announced via Bugtraq on March 24, 2001.

(Raptor can be detected via a simple port scan. If the Raptor Firewall is being managed remotely, Ports 416, 417 and 418 will be open, along with any services that are running. This is probably enough to identify the Raptor.)

This gives us a way to do some reconnaissance on any network we want from a WEB browser. The firewall will happily proxy all our requests through to any network. Unfortunately port 80 (the common port for http access) is not available, however it gives us an opportunity to penetrate the firewall.

The danger of a serious breach is even more probable if we find there are services in the private network that do not run on port 80. For example, some Compaq Servers run with Compaq Insight Manager that runs a WEB server on port 2301. Some earlier versions of these WEB servers were vulnerable to directory traversal exploits that allowed any WEB user to access the root directory.

To do this we craft our request like so : http://somehost.that.may.be .there:2301

With the proliferation of Network appliances that run their own WEB servers we rapidly increase the probability that a way in will be found. If we are able to find a server to exploit we can then load a Trojan and use port 443 to tunnel back out to us. We can use port 443 because it will most likely not be proxied through the firewall.

We could try loading a Trojan by using a set of tools from http://packetstorm.securify.com known as "unitools.tgz". Unitools exploit a well known bug in Microsoft IIS servers that use Unicode characters to force the server to traverse directories. A more detailed description can be found at : -
http://www.securityfocus.com/vdb/bottom.html?section=exploit&vid=1806.

A comment from the readme file gives some insight as to its use. For example, "Works like this - two files (upload.asp and upload.inc - have it in the same dir as the PERL script) are build in the webroot (or anywhere else) using echo and some conversion strings. These files allows you to upload any file by simply surfing with a browser to the server.' It then goes on and explains the procedure. Further on it says, "This procedure is nice for servers that are very tightly firewalled; servers that are not allowed to FTP, RCP or TFTP to the Internet."

Conclusion

The network can be improved with some tightening of rules and a patch to the Firewall. If this is done we would still have quite a good secure design without much effort at all..

References

Print resources :

1. Cisco Press, Macmillan Technical Publishing, "Designing Network Security". Merike Kaeo , 1999

2. Cisco White Paper, "Cisco Safe: A Security Blueprint for Enterprise Networks". Sean Convery and Bernie Trudel., 2000

3. Sans Course Notes, "Advanced Perimeter Protection and Defense", Chris Brenton, 2000.

Online resources :

4. http://www.networkcomputing.com/907/907ws13.html , Peter Morrissey.
5. www.cisco.com : Router and firewall documents
6. www.securityfocus.com : Security and vulnerability database
7. www.checkpoint.com : Firewall documentation and whitepapers
8. packetstorm.securify.com : Security and exploit database
9. http://www.wiretrip.net/rfp / : Security site (Rain forest Puppy – whisker home)