

5 - Network Security

Jordi Nin

nin@ac.upc.edu

Department of Computer Architecture (DAC)
Universitat Politècnica de Catalunya (UPC)
Computer Security (SI)

Contents

- ① Perimetral Security: Firewalls
- ② Intrusion Detection Systems
- ③ Point to Point Security

Contents

① Perimetral Security: Firewalls

Introduction

Firewall Topologies

Filtering Rules

Application Level Filtering

Questions

② Intrusion Detection Systems

③ Point to Point Security

Definition

A **firewall** is a part of a computer system or network designed to block unauthorized access while permitting authorized communications. It is a device or set of devices that is configured to **permit** or **deny** network transmissions based on a set of rules and other criteria.

When do we need a firewall?



Any time we need to connect a secure network to an insecure network

Preliminary Definitions

- **Firewall**: used to refer to the security policy and security strategies
- **Firewall system**: set of hardware and software implementing a firewall
- **Bastion Host**: a secure host exposed to an insecure network
- **Packet**: basic Internet communication unit (datagram)
- **Dual-homed host**: a computer with two network interfaces
- **Network perimeter or Demilitarized Zone (DMZ)**: A network added between the insecure network and the secure network that we need to protect

Types of Firewall

- **Packet filter:** Packet filtering inspects each packet passing through the network and accepts or rejects it based on user-defined rules
- **Circuit-level firewall:** Applies security mechanisms when a *TCP* or *UDP* connection is established. Once the connection has been made, packets can flow between the hosts without further checking
- **Application gateway:** Applies security mechanisms to specific applications, such as *FTP* and *Telnet* servers
- **Proxy server:** Intercepts all messages entering and leaving the network acting as an intermediary between clients and servers. The proxy server hides the true network addresses

Why shouldn't we implement security in the hosts?

There are many reasons, as for example...

- administrate the security in many points is more difficult than in a single one
- hosts execute large amounts of programs, *i.e.* the risk increases
- network monitoring becomes easier
- internal network structure is hidden
- ...

What can a firewall do?

- It provides a single point of defense, allowing a controlled and audited access to services provided
- It reinforces the own system's security
- It implements a security policy to access the secure network
- It can monitor incoming / outgoing traffic
- It can limit the exposure to an insecure network
- It may become the point where security decisions are made since all traffic goes through it

What cannot a firewall do?

- It cannot protect the network against malicious attacks from inside the secure network
- It cannot protect the network against traffic not going through it
- It cannot protect the network against the bugs of authorized services
- Any application data going through it has the potential of causing problems (*i.e.* Trojans)
- If security policy is not deny by default, it cannot protect the network against new attacks

Contents

① Perimetral Security: Firewalls

Introduction

Firewall Topologies

Filtering Rules

Application Level Filtering

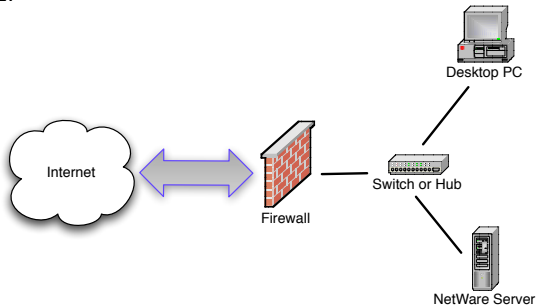
Questions

② Intrusion Detection Systems

③ Point to Point Security

A Simple Dual-Homed Firewall

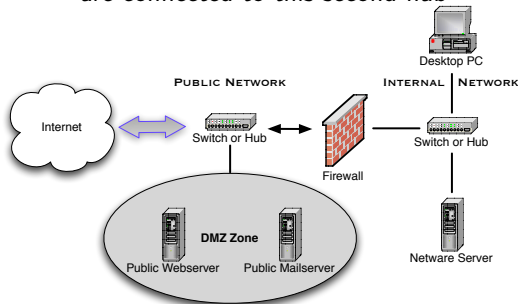
The **dual-homed firewall** is one of the simplest ways to use a firewall. The Internet comes into the firewall directly via a dial-up modem. You can't have a DMZ.



The firewall takes care of passing packets that go through its filtering rules between the internal network and the Internet, and vice versa. The two "homes" refer to the two networks that the firewall is part of - one interface connected to the outside network, and the other one connected to the inside network

A Two-Legged Network with a full exposed DMZ

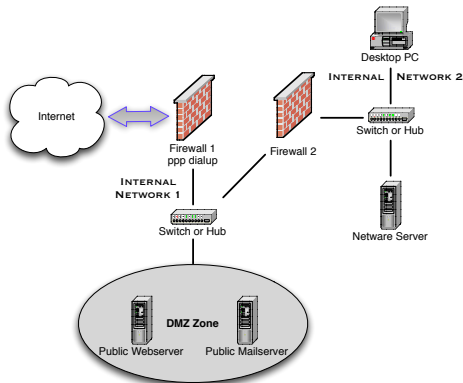
The router (Internet access) is connected to a hub. Servers that want direct access to the outside world (unfiltered by the firewall) and one of the firewall's net adapters connect also to this hub. The other firewall's net adapter connects to the internal hub. PCs that need to be protected are connected to this second hub



- **Advantages:** The firewall needs only two network cards. This simplifies the configuration of the firewall
- **Drawbacks:** DMZ network is totally exposed to the Internet

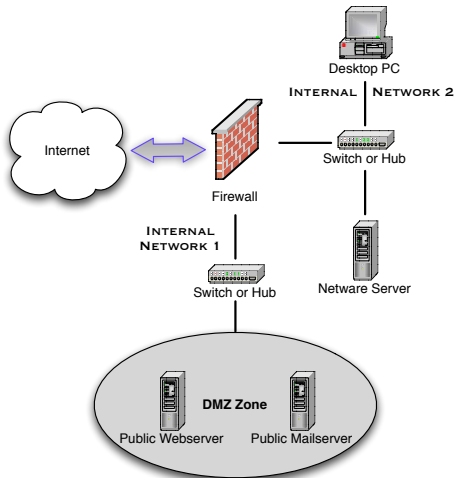
Restricted DMZ via Dialup Firewall

To protect the DMZ network one solution is to build a second router/firewall. This is useful if PPP is used. One machine is the exterior router/ firewall (1). It is responsible for creating the PPP connection and controls the access to the DMZ zone. Firewall 2 is a standard dual-homed host and its job is to protect the internal network



The Three-legged firewall

We need to add one network card in the firewall for the DMZ



- **Advantages:**
DMZ IP masquerade is possible, only one public IP address is needed
- **Drawbacks:** one extra net card → additional complexity

Contents

① Perimetral Security: Firewalls

Introduction

Firewall Topologies

Filtering Rules

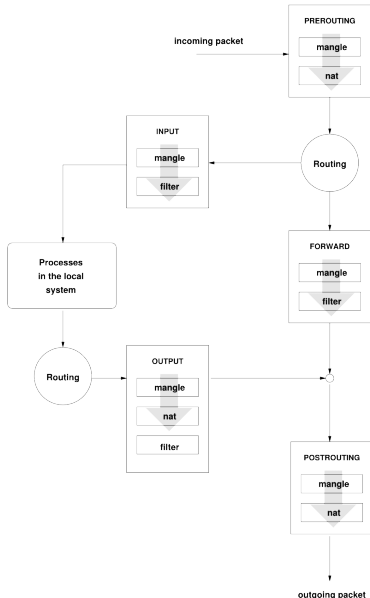
Application Level Filtering

Questions

② Intrusion Detection Systems

③ Point to Point Security

IPTables overview



Chains

- prerouting
- input
- output
- forward
- postrouting

Tables

- mangle
- nat
- filter

IPTables tables description

Table	Chain	Chain Function
Filter	FORWARD	Filters packets to servers accessible by another NIC on the firewall
	INPUT	Filters packets destined to the firewall
	OUTPUT	Filters packets originating from the firewall
NAT	PREROUTING	Address translation occurs before routing. Facilitates the transformation of the destination IP to be compatible with the firewall's routing table (DNAT)
	POSTROUTING	Address translation occurs after routing. There is no need to modify the destination IP. Used with NAT of the source IP address using either one-to-one or many-to-one (SNAT)
	OUTPUT	Network address translation for packets generated by the firewall
Mangle	ALL	Modification of the TCP packet quality of service bits before routing occurs

Basic Security policies

- **Allow** access to a service unless it is explicitly denied
 - More comfortable for users
 - Easier to administer
 - Less secure → it can't prevent unknown attacks or bugs
- **Deny** access to a service unless it is explicitly allowed
 - More secure since it is very difficult to know which services are secure and which are not
 - More restrictive and less comfortable for users

iptables – *P chain* {*ACCEPT*|*DROP*}

Filtering Rules

Filtering rules are a set of rules for filtering/allowing certain network traffic containing a certain port number, protocol type, ...

Possible filtering criteria

- Origin/destination address (or network)
- Origin/destination port numbers (well-known or private)
- Protocol type (IP/TCP/UDP/ICMP)
- Connection establishment

Filtering Rule Format

iptables - [*t table*] *action* [*options*] - *j type*

- *action*: -{A | D | I} chain n → add, delete, insert
- *options*:
 - -*i*: input interface
 - -*o*: output interface
 - -*p*: com. protocol → {IP | TCP | UDP | ICMP}
 - -*s*: source IP → {NETID+WILDCARD|HOST+IP|ANY}
 - -*d*: dest. IP → {NETID+WILDCARD|HOST+IP|ANY}
 - -*sport*: source port → [port number:port number]
 - -*dport*: dest. port → [port number:port number]
 - -*state*: connection state → {NEW,ESTABLISHED}
- *type*
 - { ACCEPT | DROP | SNAT | DNAT }

Wildcards

A wildcard mask is a 32 bit mask. It points out the IP address bits that have to be checked. The 0 mask bits indicate that the corresponding IP address bits have to be checked and 1 otherwise.

Example

- 145.34.5.6 0.0.0.0 → host 145.34.5.6
- 145.34.5.6 255.255.255.255 → ANY
- 145.34.5.6 0.0.0.255 → 145.34.5.0/24

Filtering Rules & Net Interfaces



Example

Internal hosts only access to WWW service and nothing else

- Rule set 1:
 - `iptables -t filter -A FORWARD -p TCP -i eth1 -o eth0 -dport 80 -j ACCEPT`
 - `iptables -t filter -A FORWARD -i eth1 -o eth0 -j DROP`
 - `iptables -P FORWARD ACCEPT`

Filtering Rules & Net Interfaces



Example

Internal hosts only access to WWW service and nothing else

- Rule set 1:
 - `iptables -t filter -A FORWARD -p TCP -i eth1 -o eth0 -dport 80 -j ACCEPT`
 - `iptables -t filter -A FORWARD -i eth1 -o eth0 -j DROP`
 - `iptables -P FORWARD ACCEPT`
- Rule set 2:
 - `iptables -t filter -A FORWARD -p TCP -i eth0 -o eth1 -sport 80 -j ACCEPT`
 - `iptables -t filter -A FORWARD -i eth0 -o eth1 -j DROP`
 - `iptables -P POSTROUTING ACCEPT`

Rule Order

Rules are **only checked** until a packet matches!

These two rule sets are completely different:

Rule set 1

- `iptables -t filter -A INPUT -p ICMP -j DROP`
- `iptables -t filter -A INPUT -p IP -j ACCEPT`

Rule set 2

- `iptables -t filter -A INPUT -p IP -j ACCEPT`
- `iptables -t filter -A INPUT -p ICMP -j DROP`

Rule Order

Rules are **only checked** until a packet matches!

These two rule sets are completely different:

Rule set 1

- iptables -t filter -A INPUT -p ICMP -j DROP
- iptables -t filter -A INPUT -p IP -j ACCEPT

Rule set 2

- iptables -t filter -A INPUT -p IP -j ACCEPT
- iptables -t filter -A INPUT -p ICMP -j DROP

The first rule set **rejects** all the ICMP packets while they are **accepted** with the second set (IP includes ICMP)

Hiding Network Internals

Firewalls hide the structure of internal networks, but how?

Hiding Network Internals

Firewalls hide the structure of internal networks, but how?



NAT (Network Address Translation)



NAT is the process of modifying network address information in IP packet headers while going through a traffic routing device for the purpose of **remapping** one IP address space into another

Types of NAT

NAT is out of the scope of this subject... just a small reminder

- **static NAT**: direct mapping between the internal IPs and public IPs. Internal hosts can be accessed from the Internet

- eth0 is the internet interface

- eth1 is the private network interface

```
iptables -t nat -A POSTROUTING -s internal IP  
-o eth0 -j SNAT --to-source public IP
```

```
iptables -t nat -A PREROUTING -d public IP -i  
eth0 -j DNAT --to-destination internal IP
```

Types of NAT

NAT is out of the scope of this subject... just a small reminder

- **dynamic NAT:** a set of global addresses are dynamically assigned. An Internal host has a different IP each time it accesses the Internet

```
# - eth0 is the internet interface
```

```
# - eth1 is the private network interface
```

```
iptables -A POSTROUTING -t nat -o eth0 -s  
InternalNetwork -j SNAT --to-source  
147.83.34.0-147.83.34.255
```

Types of NAT

NAT is out of the scope of this subject... just a small reminder

- **PAT:** All internal hosts share the same internal global IP, ports are modified to avoid collisions

- eth0 is the internet interface

- eth1 is the private network interface

- Forward the internet traffic

```
iptables -A POSTROUTING -t nat -o eth0 -s  
InternalNetwork -j MASQUERADE
```

Contents

① Perimetral Security: Firewalls

Introduction

Firewall Topologies

Filtering Rules

Application Level Filtering

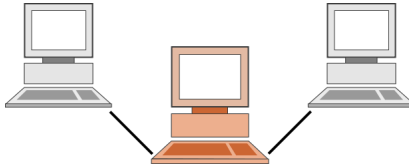
Questions

② Intrusion Detection Systems

③ Point to Point Security

Description

a **proxy server** is a server that acts as an intermediary for requests from clients seeking resources from other servers



General procedure

- 1 A client connects to the proxy server, requesting a service (a file or web page) available from a different server
- 2 The proxy server evaluates the request according to its filtering rules
- 3 If the request is validated by the filter, the proxy provides the resource by connecting to the relevant server and requesting the service on behalf of the client

Possible Applications

- To keep machines behind it anonymous
- To speed up access to resources (using caching)
- To apply access policy to network services or content, e.g. to block undesired sites
- To log / audit usage
- To scan transmitted content for malware before delivery
- To scan outbound content, e.g. for data leak protection
- To circumvent regional restrictions

Proxy Types and Functions I

- **Caching proxy servers** accelerate service requests by retrieving content saved from a previous request. They keep local copies of frequently requested resources
- **Web proxy servers** serve as a web cache. Most proxy programs provide a way to deny access to URLs specified in a blacklist (content filtering). Some web proxies also reformat web pages for a specific purpose or audience, such as for cell phones and PDAs
- **Anonymous proxy servers** attempt to anonymize web traffic
 - *Open proxy* (without access control): the web server receives requests from the anonymizing proxy server, and thus does not receive information about the end user's address. (note that, the requests are not anonymous to the anonymizing proxy server)
 - *Close proxy* (with access control): authorized users must log on to gain access to the web. The proxy administrator (a company) can thereby track usage to individuals

Proxy Types and Functions II

- **Intercepting (transparent) proxy servers** combine a proxy server with a gateway or router (with NAT capabilities). Connections made by client browsers through the gateway are diverted to the proxy without client-side configuration (or knowledge). They are commonly used in businesses to prevent avoidance of acceptable use policy, and to ease administrative burden.
- **Reverse proxy** is a server installed in the neighborhood of one or more web (application) servers. All traffic coming from the Internet and with a destination to the 'client' servers goes through the proxy server. There are several reasons for installing reverse proxy servers:
 - *Encryption / SSL acceleration*: Different final server clients share the same ssl key
 - *Load balancing*: connections are distributed among several servers
 - *Security*: it is an additional layer of defense and it can protect against some OS and WebServer specific attacks. However, it does not provide any protection to attacks against the web application or service itself
 - *Additional services*: Data compression, caching of static content

Contents

① Perimetral Security: Firewalls

Introduction

Firewall Topologies

Filtering Rules

Application Level Filtering

Questions

② Intrusion Detection Systems

③ Point to Point Security

Question I

What can a packet filter firewall do?

- ① it can control the incoming / outgoing traffic
- ② it can control the behavior of the applications
- ③ it can filter the incoming emails by checking their content
- ④ it can decide which connections are allowed

Question I

What can a packet filter firewall do?

- ① it can control the incoming / outgoing traffic
- ② it can control the behavior of the applications
- ③ it can filter the incoming emails by checking their content
- ④ it can decide which connections are allowed

Question II

Which of the following statements about firewalls and proxies are true:

- ① Firewalls speed up TCP connections
- ② Proxies only manage the access control and log capabilities of a computer network
- ③ Proxies audit the content of the packets whilst firewalls only check the packet headers
- ④ Proxies are commonly use to anonymize internet connections
- ⑤ Firewalls are more secure than proxies

Question II

Which of the following statements about firewalls and proxies are true:

- ① Firewalls speed up TCP connections
- ② Proxies only manage the access control and log capabilities of a computer network
- ③ Proxies audit the content of the packets whilst firewalls only check the packet headers
- ④ Proxies are commonly use to anonymize internet connections
- ⑤ Firewalls are more secure than proxies

Question III

Proxy servers goals include ...

- ① to control the web connections of internal clients
- ② to establish load balancing policies
- ③ to hide network internals using NAT
- ④ to offer services like data encryption or compression
- ⑤ to allow for anonymous communications

Question III

Proxy servers goals include ...

- ① to control the web connections of internal clients
- ② to establish load balancing policies
- ③ to hide network internals using NAT
- ④ to offer services like data encryption or compression
- ⑤ to allow for anonymous communications

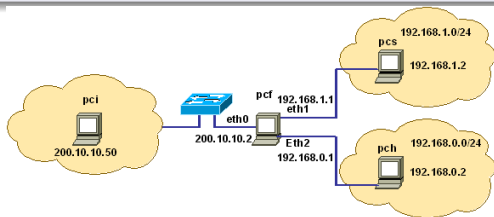
Question IV

Say which statements are true...

Firewall iptables rules (default filtering policy is deny all)

```
iptables -t filter -A FORWARD -i eth2 -o eth1 -j ACCEPT
```

```
iptables -t filter -A FORWARD -i eth1 -o eth2 -j ACCEPT
```



- ① pcs is reachable from pci
- ② pcf is reachable from pch
- ③ pcs is reachable from pch
- ④ pch is reachable from pcs

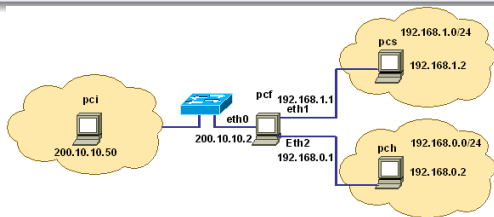
Question IV

Say which statements are true...

Firewall iptables rules (default filtering policy is deny all)

```
iptables -t filter -A FORWARD -i eth2 -o eth1 -j ACCEPT
```

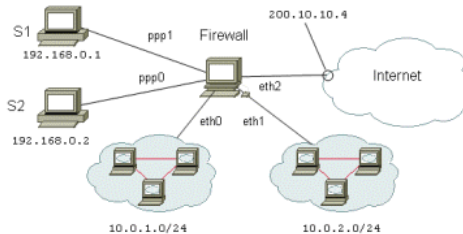
```
iptables -t filter -A FORWARD -i eth1 -o eth2 -j ACCEPT
```



- ① pcs is reachable from pci
- ② pcf is reachable from pch
- ③ pcs is reachable from pch
- ④ pch is reachable from pcs

Question V

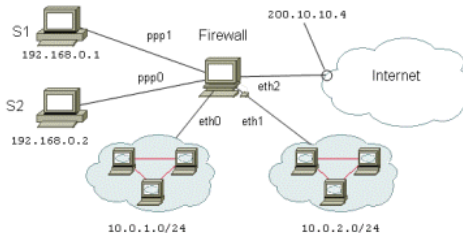
Given the following network, we want any host in the inside network 10.0.1.0/24 to be able to access the inside network 10.0.2.0/24 but not vice versa. Choose the correct filtering rules that we should introduce into the firewall, taking into account that the default filtering policy is deny all



- ① `iptables -t filter -A FORWARD -i eth0 -o eth1 -state NEW -j ACCEPT`
- ② `iptables -t filter -A FORWARD -i eth0 -o eth1 -state ESTABLISHED -j ACCEPT`
- ③ `iptables -t filter -A FORWARD -i eth1 -o eth0 -state NEW -j ACCEPT`
- ④ `iptables -t filter -A FORWARD -i eth1 -o eth0 -state ESTABLISHED -j ACCEPT`

Question V

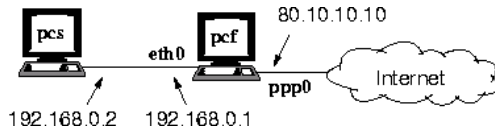
Given the following network, we want any host in the inside network 10.0.1.0/24 to be able to access the inside network 10.0.2.0/24 but not vice versa. Choose the correct filtering rules that we should introduce into the firewall, taking into account that the default filtering policy is deny all



- ① `iptables -t filter -A FORWARD -i eth0 -o eth1 -state NEW -j ACCEPT`
- ② `iptables -t filter -A FORWARD -i eth0 -o eth1 -state ESTABLISHED -j ACCEPT`
- ③ `iptables -t filter -A FORWARD -i eth1 -o eth0 -state NEW -j ACCEPT`
- ④ `iptables -t filter -A FORWARD -i eth1 -o eth0 -state ESTABLISHED -j ACCEPT`

Question VI

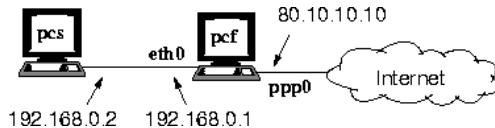
Assume that 80.10.10.10 is the only available public address. This address has been assigned to the port `ppp0` of `pcf` (see the figure). In `pcs` there is a web server (port 80). Say which of the following NAT options would configure the only web server of `pcs` to be reachable from the Internet using the address 80.10.10.10



- ① static NAT redirecting all the network traffic from `pcf` to `pcs`
- ② PAT redirecting only the network traffic of the TCP port number 80 from `pcf` to `pcs`
- ③ it is not possible to redirect the `www` traffic to `pcs`, `pdf` should be replaced by a proxy server
- ④ It is not possible to have access to `pcs` with this address, because it has been already assigned to `pcf`

Question VI

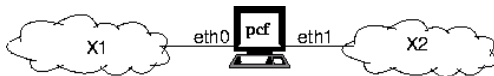
Assume that 80.10.10.10 is the only available public address. This address has been assigned to the port ppp0 of pcf (see the figure). In pcs there is a web server (port 80). Say which of the following NAT options would configure the only web server of pcs to be reachable from the Internet using the address 80.10.10.10



- ① static NAT redirecting all the network traffic from pcf to pcs
- ② PAT redirecting only the network traffic of the TCP port number 80 from pcf to pcs
- ③ it is not possible to redirect the www traffic to pcs, pcf should be replaced by a proxy server
- ④ It is not possible to have access to pcs with this address, because it has been already assigned to pcf

Question VII

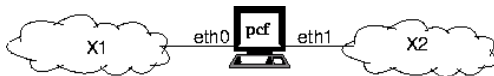
In the network of the figure we wish that the hosts from the network X1 can reach the servers from the network X2, but not otherwise. Say which of the following configurations would achieve this goal



- ① `iptables -t filter -A FORWARD -i eth0 -j ACCEPT`
`iptables -P FORWARD DROP`
- ② `iptables -t filter -A FORWARD -o eth1 -j ACCEPT`
`iptables -t filter -A FORWARD -o eth0 -state ESTABLISHED -j ACCEPT`
`iptables -P FORWARD DROP`
- ③ `iptables -t filter -A FORWARD -o eth1 -j DROP`
- ④ `iptables -t filter -A FORWARD -i eth0 -o eth1 -j ACCEPT`
`iptables -t filter -A FORWARD -i eth1 -o eth0 -j DROP`

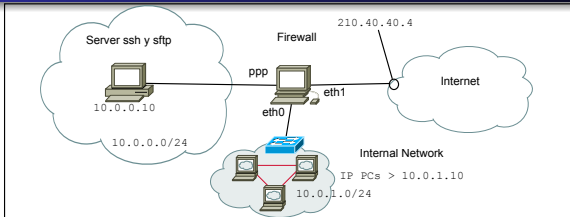
Question VII

In the network of the figure we wish that the hosts from the network X1 can reach the servers from the network X2, but not otherwise. Say which of the following configurations would achieve this goal



- ① `iptables -t filter -A FORWARD -i eth0 -j ACCEPT`
`iptables -P FORWARD DROP`
- ② `iptables -t filter -A FORWARD -o eth1 -j ACCEPT`
`iptables -t filter -A FORWARD -o eth0 -state ESTABLISHED -j ACCEPT`
`iptables -P FORWARD DROP`
- ③ `iptables -t filter -A FORWARD -o eth1 -j DROP`
- ④ `iptables -t filter -A FORWARD -i eth0 -o eth1 -j ACCEPT`
`iptables -t filter -A FORWARD -i eth1 -o eth0 -j DROP`

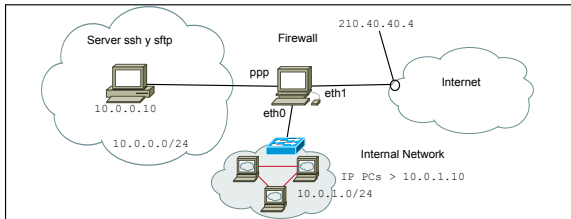
Problem I



Suppose that the internal network and the Internet are already configured. Define the NAT polices and filtering rules for the following scenarios:

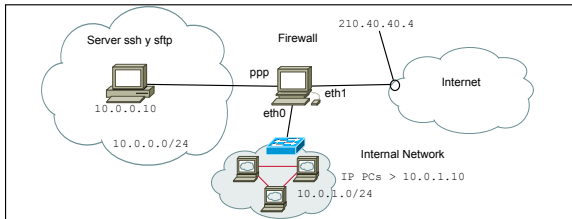
- 1 As the internal network `10.0.0.0/24` belongs to a private range. Configure the NAT service in the firewall in such a way that the server `10.0.0.10` is accessible from the Internet
- 2 Configure the firewall in such a way that any pc from the internal network can access to the server (to all possible services) but not vice versa and internal hosts have access to the WWW services
- 3 Configure the firewall in such a way that users from the Internet can only access the server to use the ssh and sftp services

Problem I



- 1 As the internal network 10.0.0.0/24 belongs to a private range. Configure the NAT service in the firewall in such a way that the server 10.0.0.10 is accessible from the Internet

Problem I



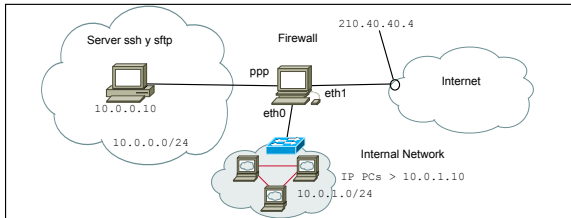
- 1 As the internal network 10.0.0.0/24 belongs to a private range. Configure the NAT service in the firewall in such a way that the server 10.0.0.10 is accessible from the Internet

In this case the use of static NAT is the simplest solution

```
iptables -t nat -A POSTROUTING -s 10.0.0.10 -o eth1 -j SNAT --to-source 210.40.40.4
```

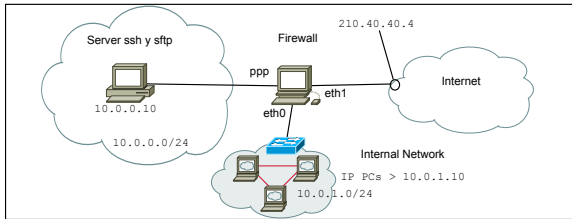
```
iptables -t nat -A PREROUTING -d 210.40.40.4 -i eth1 -j DNAT
--to-destination 10.0.0.10
```

Problem I



- 2 Configure the firewall in such a way that any pc from the internal network can access to the server (to all possible services) but not vice versa and internal hosts have access to the WWW services

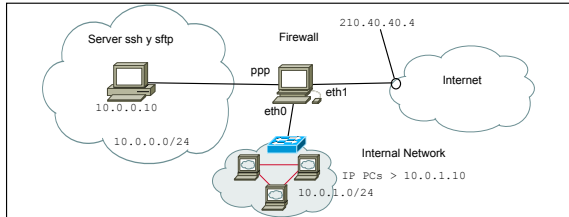
Problem I



- ② Configure the firewall in such a way that any pc from the internal network can access to the server (to all possible services) but not vice versa and internal hosts have access to the WWW services

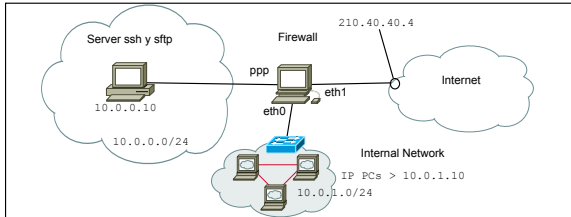
```
iptables -a FORWARD -i eth0 -o ppp -d host 10.0.0.10 -j ACCEPT
iptables -a FORWARD -i ppp -o eth0 -s host 10.0.0.10 -state established
-j ACCEPT
iptables -a FORWARD -i eth0 -o eth1 -s 10.0.1.0 0.0.0.255 -dport 80 -j
ACCEPT
iptables -a FORWARD -i eth1 -o eth0 -d 10.0.1.0 0.0.0.255 -sport 80
-state established -j ACCEPT
iptables -P FORWARD DROP
```

Problem I



- ③ Configure the firewall in such a way that users from the Internet can only access the server to use the ssh and sftp services

Problem I



- ③ Configure the firewall in such a way that users from the Internet can only access the server to use the ssh and sftp services

```
iptables -a FORWARD -i eth1 -o ppp -d host 10.0.0.10 -dport 22 -j ACCEPT
iptables -a FORWARD -i ppp -o eth1 -s host 10.0.0.10 -sport 22 -state
established -j ACCEPT
```

Contents

① Perimetral Security: Firewalls

② Intrusion Detection Systems

Introduction

Classification by Functionality

Classification by Architecture

Snort

Questions

③ Point to Point Security

Intrusion Definition

Intrusion is the act of thrusting in, or of entering into a place or state without invitation, right, or welcome



When we speak of intrusion detection, we are referring to the act of detecting an unauthorized intrusion by a *computer* on a *network*. This unauthorized access, or intrusion, is an attempt to compromise, or otherwise do harm, to other network devices

IDS Definition

An **Intrusion Detection System** (IDS) is the high-tech equivalent of a burglar alarm. A burglar alarm is configured to monitor access points, hostile activities, and known intruders.



The simplest way to define an IDS is to describe it as a specialized tool that knows how to read and interpret the contents of log files from routers, firewalls, servers, and other network devices

IDS store a database of known attack signatures and compare patterns of activity or traffic they see in the logs they are monitoring against those signatures to recognize when a **close** match between a signature and current or recent behavior occurs

IDS vs Antivirus

IDSs trigger **alarms** or take various kinds of **automatic actions** ranging from shutting down Internet links or servers to launching backtraces, and make other active attempts to identify attackers and collect evidence of their activities



By analogy, an IDS does for a **network** what an antivirus does for **files**: it inspects the contents of network traffic to look for and deflect possible attacks, as an antivirus searches the contents of incoming files, e-mail attachments, and so forth to look for virus signatures or for possible malicious actions

IDS vs. Firewalls

Doesn' t My Firewall Serve as an IDS?



Not really, a firewall can be configured to detect certain types of intrusions, such as an attempt to access a certain TCP port, and trigger an alert if it happens. However, without a deep packet inspection this is not enough.

Deep Packet inspection

Deep Packet Inspection (DPI) is a form of computer network packet filtering that examines the data part (and possibly also the header) of a packet as it passes an inspection point, searching for protocol non-compliance, viruses, spam, intrusions or predefined criteria to decide if the packet can pass or if it needs to be routed to a different destination, or for the purpose of collecting statistical information

Main Problems

- Performance → CPU intensive
- Complexity → rules definition become more difficult

Contents

① Perimetral Security: Firewalls

② Intrusion Detection Systems

Introduction

Classification by Functionality

Classification by Architecture

Snort

Questions

③ Point to Point Security

Signature-based IDS

- Similarly to antivirus, signature-based IDS detect attacks by matching the network activity against a database of known attacks
- Signature-based IDS have a database of attack signatures (e.g. GET /etc/passwd)
- If a rule matches, an alert is triggered → simple and effective

Problem → new attacks cannot be detected

Anomaly-Based IDS

- Anomaly-based IDS build a model of “normal” system behaviour, when a deviation from the model is detected an alert is sent
- Anomaly-based IDS classify network activities as normal or anomalous

But how do you define a normal activity?

- **AI techniques:** Neural networks, pattern recognition, machine learning, fuzzy sets, ...
- **Mathematical Techniques:** Functional equations, statistical analysis, ...

Contents

① Perimetral Security: Firewalls

② Intrusion Detection Systems

Introduction

Classification by Functionality

Classification by Architecture

Snort

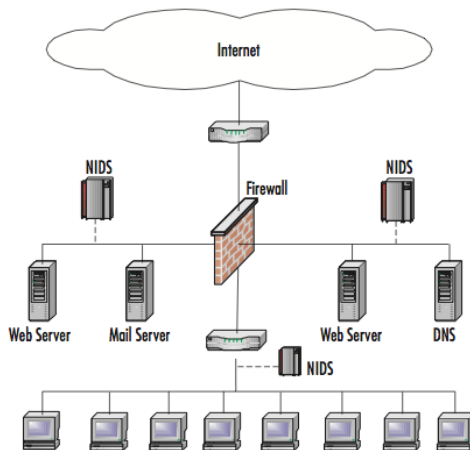
Questions

③ Point to Point Security

Network IDS

- NIDS monitors network traffic for particular network segments or devices and analyzes network, transport, and application protocols to identify suspicious activity
- At least one of the network interfaces on this machine works in promiscuous mode, capturing and analyzing all the frames that pass through him for patterns indicative of an attack

NIDS architecture



NIDS components: Sensors

- A NIDS sensor **monitors** and **analyzes** network activity on one or more network segments
- The network interface cards that will be performing monitoring are placed into **promiscuous mode**
- Sensors can be hardware-based or software-based
- Sensors can be deployed in two modes, **inline** (monitored traffic must pass through it) or **passive** (it monitors a copy of the actual network traffic; no traffic passes through the sensor)

Why passive sensors?

- Sometimes network traffic cannot be analyzed in real time
- Switches segment the network traffic → Spanning port sees a copy of the traffic
- IDS Load Balancer → It is a device that aggregates and directs network traffic to a sensor systems (DPI analysis)

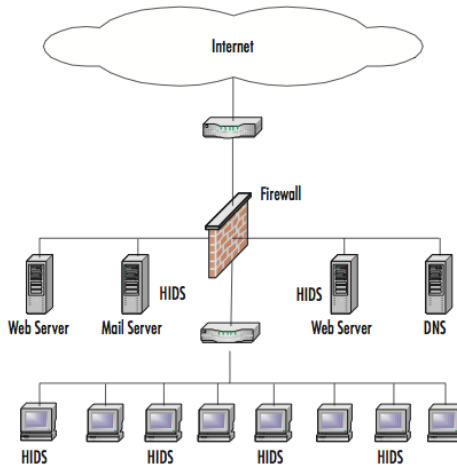
NIDS Security capabilities

- **Information gathering:** NIDS can collect information about hosts and network activity to identify hosts, operating systems, applications or network characteristics
- **Logging:** NIDS perform extensive logging of data related to detected events. Such can be used to confirm the validity of alerts, to investigate incidents, etc
- **Detection:** NIDS use a combination of signature-based detection and anomaly-based detection to perform in-depth analysis of common protocols
- **Prevention:** Once an alert is triggered other similar connections can be aborted

Host IDS

- HIDS monitor the characteristics of a single host and the events occurring within to find suspicious activity
- HIDS inspect Network traffic for the host (non-promiscuous), system logs, running processes, file accesses and modifications, system and application configuration changes, ...
- HIDS have agents installed on the hosts of interest or dedicated applications. Each agent monitors activity and transmits data to management servers
- Main functions: System Integrity Verifiers (SIV), Log File Monitors (LFM) and Honeypots

HIDS Architecture



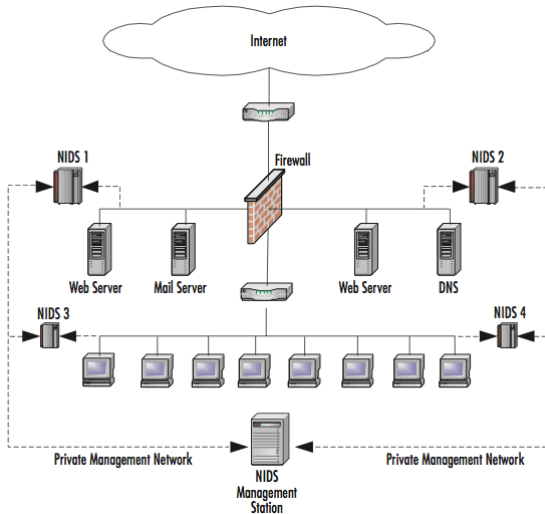
HIDS Security capabilities

- **Logging:** HIDS perform extensive logging of data related to detected events (e.g. TCP timestamp + file modification)
- **Detection:** Code Analysis (e.g. Buffer overflow detection, System call monitoring), Network Traffic Analysis and Filter, File integrity checking, ...
- **Prevention:** File system monitoring, incoming network traffic filter, ...

Distributed IDS

- NIDS detection sensors are remotely located and report to a centralized management station
- Attack logs are periodically or continuously uploaded to the management station and can be stored in a central database
- New attack signatures can be downloaded to the sensors on an as-needed basis
- The rules for each sensor can be tailored to meet its individual needs. Alerts can be forwarded to a messaging system located on the management station and used to notify the IDS administrator
- In a DIDS, the individual sensors can be NIDS, HIDS, or a combination of both

DIDS Architecture



Contents

① Perimetral Security: Firewalls

② Intrusion Detection Systems

Introduction

Classification by Functionality

Classification by Architecture

Snort

Questions

③ Point to Point Security

What Is Snort?

The name Snort came from the fact that the application is a “sniffer and more”

In short, Snort is a packet sniffer/packet logger/network IDS.

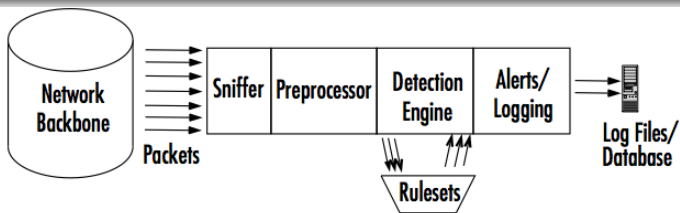
Main properties

- Snort is a NIDS and it uses signature-based analysis
- Works on multiple OSs
- Uses a hexdump payload dump
- Displays all the different network packets the same way

Additional Software needed

- MySQL, Postgres, or Oracle (SQL databases)
- smbclient if using WinPopup messages
- Apache or another Web server
- PHP or Perl, if you have plug-ins that require them
- SSH for remote access (or Terminal Server with Windows)
- Apache with SSL capabilities for monitoring (or IIS for Windows)

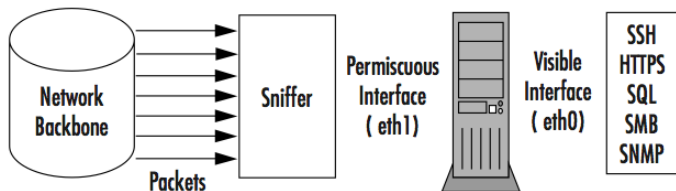
Snort Components & Architecture



Snort's architecture is similar to a mechanical coin sorter:

- ① It takes all the coins (packets from the network)
- ② Then, it sends them through a chute to determine if they are coins, and how they should roll (the preprocessor)
- ③ Next, it sorts the coins according to the coin type. This is for storage from 1 cent to 2 euros (IDS detection engine)
- ④ Finally, it is the administrator's task to decide what to do with the coins, usually you will roll them and store them (logging and database storage)

Packet Sniffer

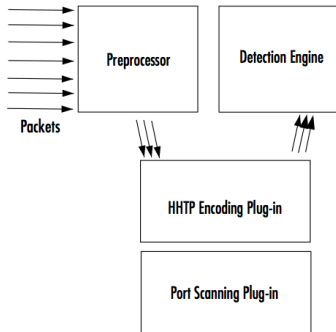


Packet sniffers have various uses:

- Network analysis and troubleshooting
- Performance analysis and benchmarking
- Eavesdropping for clear-text passwords and other interesting tidbits of data

Encrypting your network traffic can prevent people from being able to sniff your packets into something readable. Like any network tool, packet sniffers can be used for good and evil

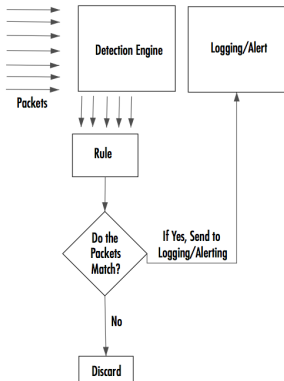
Preprocessor



- It takes the raw packets and checks them against certain plug-ins. These plug-ins check for a certain type of behavior from the packet
- Once the packet is determined to have a particular type of “behavior”, it’s then sent to the detection engine

Plug-ins can be enabled and disabled as they are needed at the preprocessor level

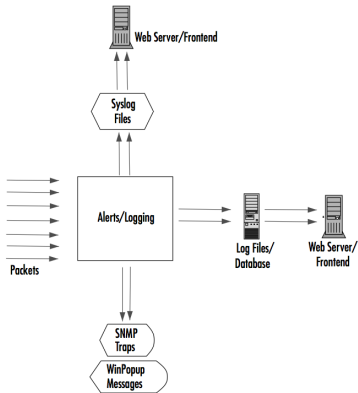
Detection Engine



- It takes the data that comes from the preprocessor and its plug-ins, and that data is checked through a set of rules
- If the rules match the data in the packet, then they are sent to the alert processor

- **The rule header.** It is basically the action to take (log or alert), type of network packet (TCP, UDP, ICMP, ...), source and destination IP addresses, and ports
- **The rule option.** It is the content in the packet that should make the packet match the rule

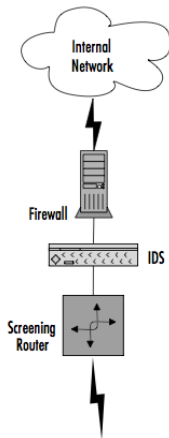
Alerting Component



- After the Snort data goes through the detection engine, it needs to go out somewhere
- If the data matches a rule in the detection engine, then an alert is triggered

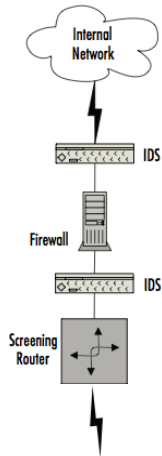
Alerts can be sent to a log file, through a network connection, through UNIX sockets or Windows Popup (SMB), or SNMP traps. The alerts can also be stored in an SQL database such as MySQL or Postgres. ...

Some snort Architectures



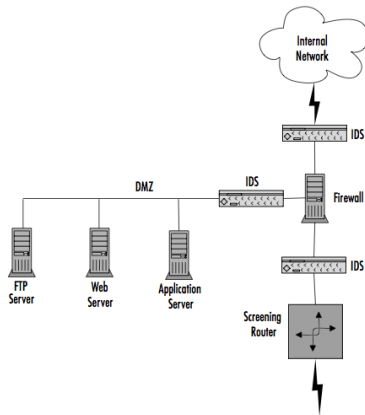
An IDS Network Architecture with a Screening Router

Some snort Architectures



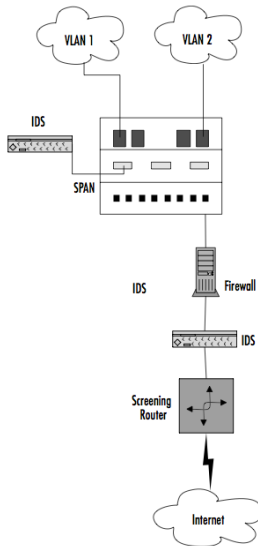
A Firewalled Network with Snort Systems

Some snort Architectures



A Firewalled Network with a DMZ and Snort

Some snort Architectures



A Switched Network with Snort Systems

Defining rules: Variables

Snort provides users the ability to define custom variables to use them within the rule sets. Defining variables is straightforward, as they use a one-to-one substitution method

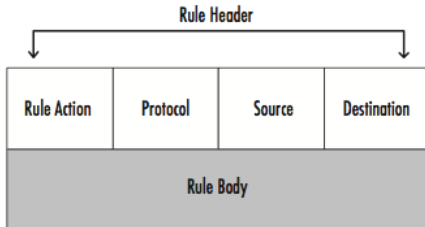


```
var <desired_variable_name> <variable_value>
```

Example

```
var DNS_SERVER 10.1.1.2
var INTERNAL_NET 10.20.0.0/16
var INTERNAL_NETS [10.1.0.0/16, 10.2.1.0/24]
```

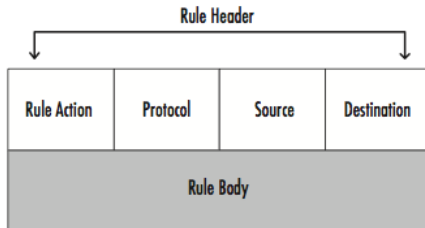
Defining rules: Rule structure



Rule action

- **Pass.** It simply ignores the packet
- **Log.** It allows you to log the packet
- **Alert.** It logs the packet, and then alerts the user
- **Dynamic.** It remains dormant until an Activate rule triggers it "on", then it acts like a Log action rule
- **Activate.** It generates an alert and then starts the specified dynamic rule

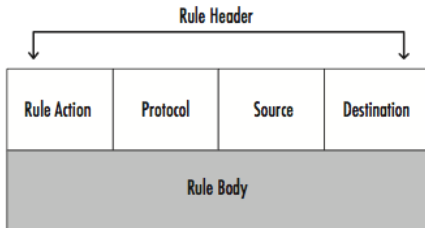
Defining rules: Rule structure



Protocols

- Supported protocols: ICMP, TCP, UDP and IP
- Additional modules for: 802.11, HTTP, ARP, etc.
- You can only specify **one** protocol per rule!

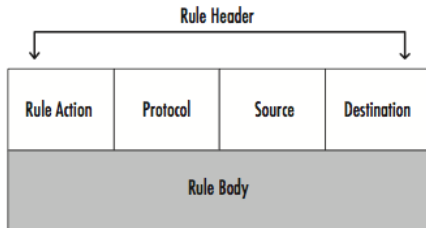
Defining rules: Rule structure



source & destination addresses

- There are two available options for including system addresses in the rule: using individual IP addresses, or Classless Inter Domain Routing addresses (CIDR) → 10.2.0.0/16
- Port range: initial:final → 21:23 or :21 or !80 or any

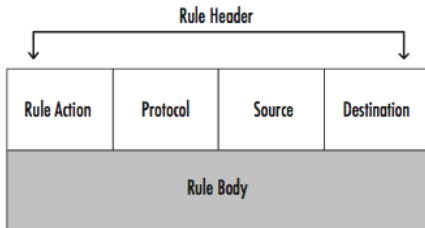
Defining rules: Rule structure



Direction Operators

- operator \rightarrow . It tells the Snort engine that the source information for the rule is on the left side of the arrow, and the destination side of the rule is on the right side
- operator \leftrightarrow . It is the bidirectional operator

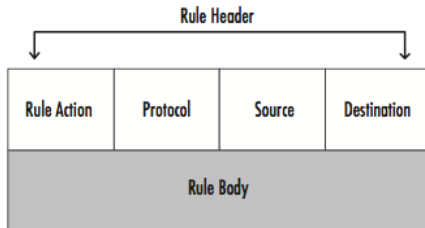
Defining rules: Rule structure



The Rule Body

- The rule options are separated by semicolons within the main body of the Snort rule:
alert tcp any any -> any 80 (**msg:"OINK!";**)
- Rule Content → The NeverEnding Story
 - ASCII Content: alert tcp any any -> any any
(content:'malicious string /etc/passwd';
msg:'Searching for ASCII Garbage!');

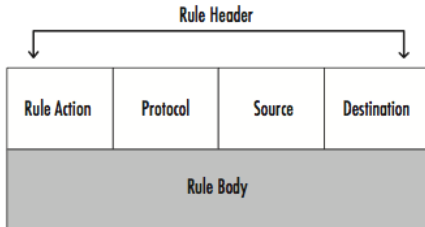
Defining rules: Rule structure



The Rule Body

- The rule options are separated by semicolons within the main body of the Snort rule:
alert tcp any any -> any 80 (**msg: "OINK!";**)
- Rule Content → The NeverEnding Story
 - Binary Content (|): alert tcp any any -> any any
(content: '|0000 0101 EFFF|';
msg: 'Searching for Garbage!');

Defining rules: Rule structure



The Rule Body

- The rule options are separated by semicolons within the main body of the Snort rule:
alert tcp any any -> any 80 (**msg: "OINK!";**)
- Rule Content → The NeverEnding Story
 - options: TCP/UDP flags, offset (initial searching point), nocase, session, URI, IP options (don't fragment), TCP options (sequence number) ...

Contents

① Perimetral Security: Firewalls

② Intrusion Detection Systems

Introduction

Classification by Functionality

Classification by Architecture

Snort

Questions

③ Point to Point Security

Question I

Fill the msg option value with the goal of the following rule

```
alert tcp any any -> any 2001 (msg: "?????";)
```

Question 1

Fill the msg option value with the goal of the following rule

```
alert tcp any any -> any 2001 (msg: "?????";)
```

Alert anyone who tries to access Port 2001

Question II

Fill the msg option value with the goal of the following rule

```
alert tcp $HOME_NET 22 -> $EXTERNAL_NET any  
(msg:"????"; flags: S; tag: session, 300, packets;)
```

Question II

Fill the msg option value with the goal of the following rule

```
alert tcp $HOME_NET 22 -> $EXTERNAL_NET any  
(msg:"????"; flags: S; tag: session, 300, packets;)
```

Alert SSH login from untrusted networks

Question III

Fill the msg option value with the goal of the following rule

```
alert tcp any any -> any any (msg: "???"; content: "Microsoft Word"; offset = 116 ; depth = 14; content: "MSWord-Doc"; nocase ; distance = 14 ; within = 9 ;)
```

Question III

Fill the msg option value with the goal of the following rule

```
alert tcp any any -> any any (msg: "???"; content: "Microsoft Word"; offset = 116 ; depth = 14; content: "MSWord-Doc"; nocase ; distance = 14 ; within = 9 ;)
```

Alert MS Word documents

Question IV

Fill the msg option value with the goal of the following rule

```
alert tcp $HOME_NET any -> $EXTERNAL_NET  
$HTTP_PORTS (msg:"????"; flow:to_server,established;  
content:"eBay.com"; nocase;)
```

Question IV

Fill the msg option value with the goal of the following rule

```
alert tcp $HOME_NET any -> $EXTERNAL_NET  
$HTTP_PORTS (msg:"????"; flow:to_server,established;  
content:"eBay.com"; nocase;)
```

Alert all the HTTP connections to www.eBay.com

Question V

Fill the msg option value with the goal of the following rule

```
alert tcp any any -> 147.83.2.135 80 ( content: "POST";  
nocase; msg: "???"; threshold: type both, track by_src, count  
30, seconds 60; )
```

Question V

Fill the msg option value with the goal of the following rule

```
alert tcp any any -> 147.83.2.135 80 ( content: "POST";  
nocase; msg: "???"; threshold: type both, track by_src, count  
30, seconds 60; )
```

Webmail Brute Force Attempt or Spam Attack, this rule Alert anyone that does an HTTP POST more than 30 times within 60 seconds at the www.upc.edu webpage

Question VI

Fill the msg option value with the goal of the following rule

```
var DNS_SERVERS [10.1.1.1, 10.2.1.1, 10.3.1.1, 10.4.1.1]
```

```
alert udp ![$DNS_SERVERS,$SMTP_SERVERS] - >  
$DNS_SERVERS 53 (msg:"????")
```

Question VI

Fill the msg option value with the goal of the following rule

```
var DNS_SERVERS [10.1.1.1, 10.2.1.1, 10.3.1.1, 10.4.1.1]  
  
alert udp ![$DNS_SERVERS,$SMTP_SERVERS] - >  
$DNS_SERVERS 53 (msg:"????")
```

Unauthorized DNS query. Users can only query the DNS servers stored in DNS_SERVERS variable

Question VII

Fill the msg option value with the goal of the following rule

```
var FTP_SERVERS [10.1.4.4/32,10.1.3.7/32]

pass tcp $EXTERNAL_NET any <> $FTP_SERVERS 21

alert tcp $FTP_SERVERS any -> any !21
(msg:"????"; flow:established; classtype:bad-unknown; )
```

Question VII

Fill the msg option value with the goal of the following rule

```
var FTP_SERVERS [10.1.4.4/32,10.1.3.7/32]

pass tcp $EXTERNAL_NET any <> $FTP_SERVERS 21

alert tcp $FTP_SERVERS any -> any !21
(msg:"????"; flow:established; classtype:bad-unknown; )
```

Odd port use - ftp server. Ftp servers can only work using the port 21

Question VIII

Fill the msg option value with the goal of the following rule

```
alert udp [192.168.1.0/24,192.168.2.0/24] any - >
![192.168.1.0/24,192.168.2.0/24] any (msg:"????";
classtype:policy-violation;)
```

```
alert tcp ![192.168.1.0/24,192.168.2.0/24] any - >
[192.168.1.0/24,192.168.2.0/24] any (msg:"????";
classtype:policy-violation;)
```

Question VIII

Fill the msg option value with the goal of the following rule

```
alert udp [192.168.1.0/24,192.168.2.0/24] any - >  
![192.168.1.0/24,192.168.2.0/24] any (msg:"????";  
classtype:policy-violation;)
```

```
alert tcp ![192.168.1.0/24,192.168.2.0/24] any - >  
[192.168.1.0/24,192.168.2.0/24] any (msg:"????";  
classtype:policy-violation;)
```

UPD Traffic from Whitelist Network to Non-Whitelist Network
Destination and TCP Traffic To Whitelist Network from
Non-Whitelist Network Destination.

Network traffic is only allowed among the authorized networks

Contents

① Perimetral Security: Firewalls

② Intrusion Detection Systems

③ Point to Point Security

Introduction

VPN Architectures

IPSec Fundamentals

Practical Examples

Questions

Contents

① Perimetral Security: Firewalls

② Intrusion Detection Systems

③ Point to Point Security

Introduction

VPN Architectures

IPSec Fundamentals

Practical Examples

Questions

Network Layer Security

Initially computer networks were used by academic researchers for mail (or information in general) exchange



Security was not very important



Nowadays, this is **not** the case, millions of users use the Internet to access to their bank services, to buy products or services, etc ...

Then, we need to protect network communications at the layer that is responsible for routing packets across networks


Network Layers Description

TCP/IP is widely used throughout the world to provide network communications. TCP/IP communications are composed of four layers that work together

- **Application layer** sends and receives data for particular applications, such as DNS, HTTP and SMTP
- **Transport Layer** provides connection-oriented (TCP) or connectionless (UDP) services for transporting application layer services between networks
- **Network Layer** routes packets across networks. Internet Protocol (IP) is the fundamental protocol
- **Data Link Layer** handles communications on the physical network components. The best-known data link layer protocol is Ethernet

The Need for Network Layer Security

Security controls exist for network communications at each layer of the TCP/IP model. The goal in each layer is

- **Application layer** Separate controls must be established for each application. For example, Pretty Good Privacy (PGP) is commonly used to encrypt e-mail messages (SMTP) 
- **Transport Layer** Controls at this layer can be used to protect the data in a single communication session between two hosts. For example (TLS / SSL) protocols secure HTTP traffic
- **Network Layer** Controls at this layer apply to all applications and are not application-specific. For example, IPSec secures all network communications between two hosts without modifying applications
- **Data Link Layer** Controls are applied to all communications on a specific physical link, such as a dedicated circuit between two buildings or a dial-up modem connection to an ISP

Internet Protocol Security (RFCs 4301 & 4309)

IPSec is the most commonly used network layer security control. IPSec is a framework of open standards for ensuring private communications over IP networks. Depending on its implementation, it can provide any combination of the following types of protection:

- **Confidentiality.** IPSec can ensure that data cannot be read by unauthorized parties. This is accomplished by encrypting data using a cryptographic algorithm and a secret key
- **Integrity.** IPSec can determine if data has been changed during transit. Data integrity can be assured by generating a message authentication code (MAC) value, a cryptographic data checksum
- **Peer Authentication.** Each IPSec endpoint confirms the identity of the other IPSec endpoint, ensuring that the network traffic and data is being sent from the expected host

Internet Protocol Security (RFCs 4301 & 4309)

IPSec is the most commonly used network layer security control. IPSec is a framework of open standards for ensuring private communications over IP networks. Depending on its implementation, it can provide any combination of the following types of protection:

- **Replay Protection.** The same data is not delivered multiple times, and data is not delivered out of order. However, IPSec does not ensure that data is delivered in the exact order in which it is sent
- **Traffic Analysis Protection.** A person monitoring the traffic does not know which parties are communicating, how often communications are occurring, or how much data is being exchanged. However, the number of packets can be counted.
- **Access Control.** IPSec endpoints can perform filtering to ensure that only authorized IPSec users can access particular network resources

Virtual Private Networking (VPN)

The most common use of IPsec implementations is providing Virtual Private Networking (VPN) services

A **VPN** is a virtual network, built on top of existing physical networks, that can provide a secure communication mechanism for data and IP information transmitted between networks



As VPNs can be used over the Internet, they facilitate the secure transfer of sensitive data across public networks

VPN Advantages

- VPNs are often less expensive than alternatives such as dedicated private communications lines (e.g. X.25 lines)
- VPNs can also provide flexible solutions, such as securing communications between remote telecommuters (remote worker) and the organization's servers, regardless of where the telecommuters are placed
- A VPN can even be established within a single network to protect particularly sensitive communications from other parties on the same network (e.g. servers administration)

VPN & Cryptography

VPNs can use both private and public key cryptography

- **Private key** cryptography uses the same key for both encryption and decryption, (e.g. DES, 3DES, AES, ...). It is used for protecting the actual data because of its relative efficiency
- **Public key** cryptography uses separate keys for encryption and decryption, or to digitally sign and verify a signature (e.g. RSA, ...). It is used to authenticate the identities of both parties

Contents

① Perimetral Security: Firewalls

② Intrusion Detection Systems

③ Point to Point Security

Introduction

VPN Achitectures

IPSec Fundamentals

Practical Examples

Questions

Gateway-to-Gateway Architecture

IPsec-based VPNs are often used to provide secure network communications between **two networks** by deploying a VPN gateway onto each network and establishing a VPN connection between them.

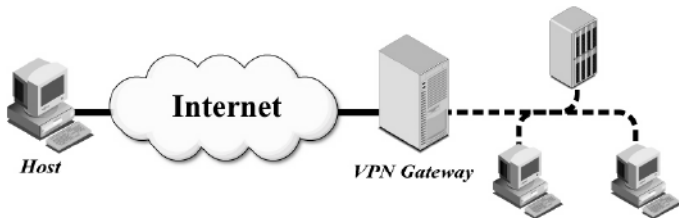


Usually, to facilitate VPN connections, one of the VPN gateways issues a request to the other to establish an IPsec connection. Routing on each network is configured so that as hosts on one network need to communicate with hosts on the other network, their network traffic is automatically routed through the IPsec connection

This is the easiest VPN model to implement, in terms of user and host management

Host-to-Gateway Architecture

This model is used to provide [secure remote access](#). The organization deploys a VPN gateway onto their network; each remote access user then establishes a VPN connection between his/her host and the VPN gateway

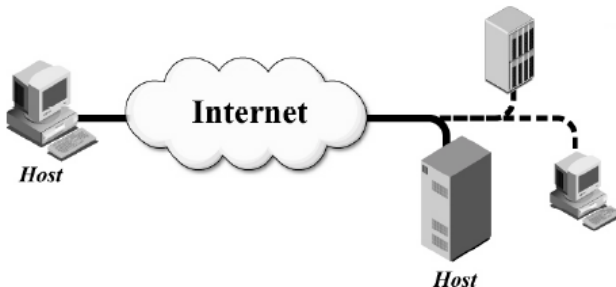


IPsec connections are created as needed for each individual VPN user. The user is typically asked by the VPN gateway to authenticate before the connection can be established

The host-to-gateway model is somewhat complex to implement and maintain in terms of user and host

Host-to-Host Architecture

This is the least commonly used VPN architecture. It is typically used for special purpose needs, such as system administrators performing **remote management** of a single server.



This model is the only one that provides protection for data throughout its transit. This can be a problem, because packet firewalls, IDS, and other devices cannot be placed to inspect the decrypted data, which effectively circumvents certain layers of security

Model Comparison

Feature	G-to-G	H-to-G	H-to-H
Provides protection between client and local gateway	No	N/A	N/A
Provides protection between VPN endpoints	Yes	Yes	Yes
Protection between remote gateway and remote server (behind gateway)	No	No	N/A
Transparent to users	Yes	No	No
Transparent to servers	Yes	Yes	No

Contents

① Perimetral Security: Firewalls

② Intrusion Detection Systems

③ Point to Point Security

Introduction

VPN Architectures

IPSec Fundamentals

Practical Examples

Questions

IPSec Components

IPsec is a **collection of protocols** that assist in protecting communications over IP networks working together in various combinations to provide protection for communications

Main protocols

- **Authentication Header (AH)**. It provides integrity for packets headers
- **Encapsulating Security Payload (ESP)**. It provides authentication and encryption services
- **Internet Key Exchange (IKE)**. It negotiates, creates and manages security associations (AS)

Authentication Header (AH)

AH provides **integrity protection for packet headers and data**, as well as user authentication. It can optionally provide replay protection and access protection. AH **cannot encrypt** any portion of packets.

Its functionality is debatable since ESP also provides authentication. However, AH is still of value because AH can authenticate portions of packets that ESP cannot

AH Modes

AH provides integrity protection for the **entire packet**, regardless of which mode is used.

- *Tunnel mode*: AH creates a new IP header for each packet. It is used in gateway VPNs

New IP Header	AH Header	Original IP Header	Transport and Application Protocol Headers and Data
Authenticated (Integrity Protection)			

- *Transport mode*: AH does not create a new IP header. It is used in host-to-host VPNs

IP Header	AH Header	Transport and Application Protocol Headers and Data
Authenticated (Integrity Protection)		

Integrity Protection Process

The first step of integrity protection is to **create a hash** by using a **keyed** hash algorithm → a message authentication code (MAC)



keyed hash algorithms create a hash based on both a **message** and a **secret key** shared by the two endpoints. The hash is added to the packet, and it is sent to the recipient.



The recipient can then **regenerate** the hash using the shared key and confirm that the two hashes match, providing integrity protection for the packet

Dynamic Header Fields

Certain IP header fields, such as **time to live (TTL)** and the **IP header checksum**, are dynamic and may change during routine communications



If the hash is calculated on **all** the original IP header values, the recalculated hash will be different.



To avoid this problem, IP header fields that may legitimately change in transit in an unpredictable manner are **excluded** from the integrity protection calculations

NAT Problems

For the same reason as before AH is often **incompatible** with network address translation (NAT) implementations



The IP source and destination address fields **are included** in the AH integrity protection calculations



If these addresses are **altered** by a NAT device, the AH integrity protection calculation made by the destination will **not** match

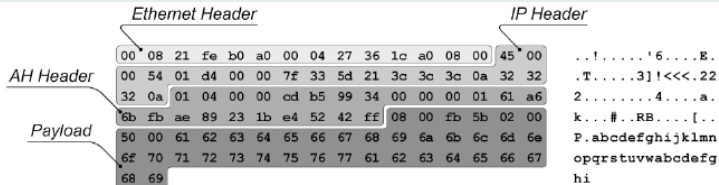
AH Header

Next Header	Payload Length	Reserved
Security Parameters Index		
Sequence Number		
Authentication Information		

- **Next Header.** It contains the IP protocol number for the next packet payload. In tunnel mode, the payload is an IP packet, so the Next Header value is set to 4 for IP-in-IP. In transport mode, the payload is usually a transport-layer protocol, often TCP (6) or UDP (17)
- **Payload Length.** This field contains the length of the payload in 4-byte increments, minus 2
- **Reserved.** This value is reserved for future use, so it should be set to 0
- **Security Parameters Index (SPI).** Each endpoint has an arbitrarily chosen SPI value, which acts as a unique identifier for the connection. The recipient uses the SPI value, along with the destination IP address and (optionally) the IPsec protocol type to determine which Security Association (SA) is being used
- **Sequence Number.** Each packet is assigned a sequential sequence number, and only packets within a sliding window of sequence numbers are accepted. This provides protection against replay attacks
- **Authentication Information.** This field contains the MAC output

How AH works

Example

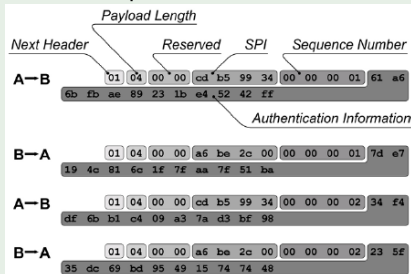


ICMP echo request packet with transport mode
it only contains a [single](#) IP header

How AH works

Example

AH header for the first four packets in an AH session between A and B



- **SPI.** A uses the hex value `cd b5 99 34` for the SPI in its packets, while host B uses the hex value `a6 b3 2c 00` for the SPI in its packets. An AH connection is composed of two one-way connections, each with its own SPI
- **Sequence Number.** Both hosts initially set the sequence number to 1, and both incremented the number to 2 for their second packets
- **Authentication Information.** The authentication (integrity protection) information, a keyed hash based on the bytes in the packet, is different in each packet. It should be different even if only one byte in a hashed section changes

Encapsulating Security Payload (ESP)

ESP is the second core IPsec security protocol. It provides both **encryption** for packet payload data and **authentication** to provide integrity protection (although not for the outermost IP header)



ESP's encryption and authentication can be **disabled**



ESP can be used to provide only encryption; encryption and integrity protection; or only integrity protection

ESP Modes

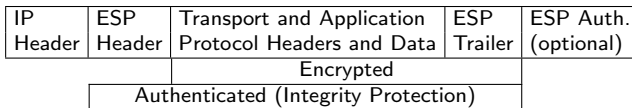
- *Tunnel mode*: It creates a new IP header for each packet. The new IP header lists the endpoints of the ESP tunnel as the source and destination of the packet.

New IP Header	ESP Header	Original IP Header	Transport and Application Protocol Headers and Data	ESP Trailer	ESP Auth. (optional)
		Encrypted			
		Authenticated (Integrity Protection)			

It can encrypt and/or protect the integrity of both the data and the original IP header. Encrypting the data protects it from being accessed or modified by unauthorized parties; encrypting the IP header conceals the nature of the communications, such as the actual source or destination of the packet. If authentication is being used for integrity protection, each packet will have an ESP Authentication section after the ESP trailer

ESP Modes

- *Transport mode*: it uses the original IP header instead of creating a new one.



It can only encrypt and/or protect the integrity of packet payloads and certain ESP components, but not IP headers. As with AH, ESP transport mode is generally only used in host-to-host architectures. Also, transport mode is incompatible with NAT

Encryption Process

ESP uses **symmetric cryptography** to provide encryption for IPsec packets. Both endpoints must use the same key to encrypt and decrypt the packets.

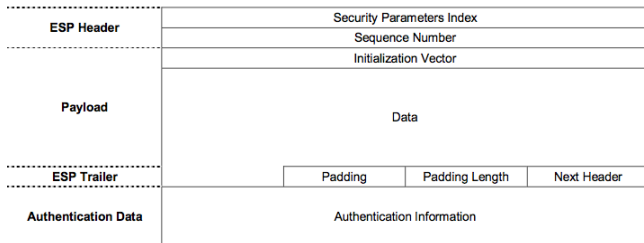


When an endpoint **encrypts** data, it **divides** the data into **small blocks** (e.g. for the AES algorithm, 128 bits each), and then performs multiple sets of cryptographic operations using the data blocks and shared key



When the other endpoint receives the encrypted data, it performs **decryption** using the same key and a similar process

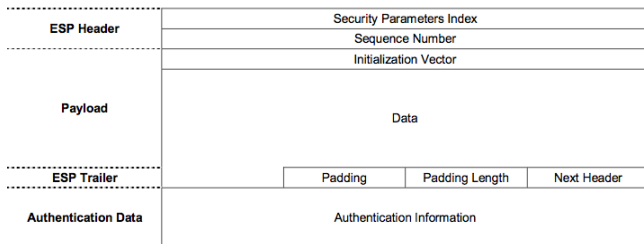
ESP Packet Fields



Each ESP header is composed of two fields:

- **SPI.** Each endpoint of each IPsec connection has an arbitrarily chosen SPI value, which acts as a unique identifier for the connection. The recipient uses the SPI value, along with the destination IP address and (optionally) the IPsec protocol type (in this case, ESP), to determine which SA is being used
- **SequenceNumber.** Each packet is assigned a sequential sequence number, and only packets within a sliding window of sequence numbers are accepted. This provides protection against replay attacks

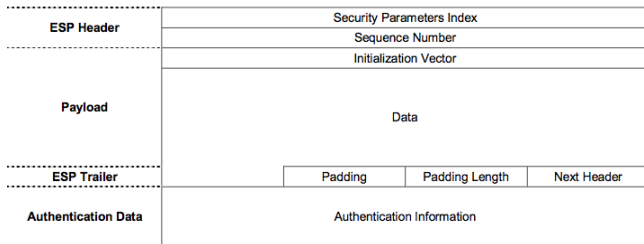
ESP Packet Fields



The next part of the packet is the payload. It is composed of

- **Payload data.** It is encrypted,
- **Initialization vector (IV).** It is not encrypted. The IV is used during encryption. Its value is different in each packet, so if two packets have the same content, the inclusion of the IV will cause the encryption of the two packets to have different results

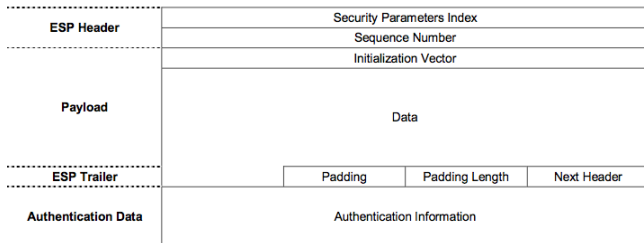
ESP Packet Fields



The third part of the packet is the ESP trailer, which contains

- **Padding.** An ESP packet may optionally contain padding, which is additional bytes of data that make the packet larger and are discarded by the packet's recipient. Because ESP uses block ciphers for encryption, padding may be needed so that the encrypted data is an integral multiple of the block size
- **Padding Length.** This number indicates how long the padding is in bytes
- **Next Header.** In tunnel mode, the payload is an IP packet, so the Next Header value is set to 4 for IP-in-IP. In transport mode, the payload is usually a transport-layer protocol, often TCP (6) or UDP (17)

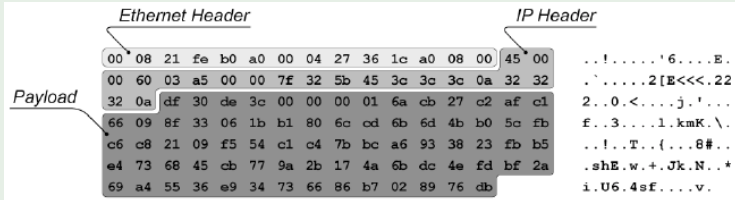
ESP Packet Fields



If ESP integrity protection is enabled, the ESP trailer is followed by an Authentication Information field. As in AH header, it contains the field that contains the MAC output. Unlike AH, the MAC in ESP does not include the outermost IP header in its calculations

How ESP works

Example



It contains five sections: Ethernet header, IP header, ESP header, encrypted data (payload and ESP trailer), and (optionally) authentication information. From the encrypted data, it is not possible to determine if this packet was generated in transport mode or tunnel mode.

How ESP works

Example

ESP header for the first four packets in an AH session between A and B

	<i>SPI</i>				<i>Sequence Number</i>			
A→B	df	30	de	3c	00	00	00	01
B→A	d9	64	ce	53	00	00	00	01
A→B	df	30	de	3c	00	00	00	02
B→A	d9	64	ce	53	00	00	00	02

The SPI and Sequence Number fields work in the same way in ESP that they do in AH.

Internet Key Exchange (IKE)

IKE protocol negotiates, creates, and manages security associations (SA). SA is a generic term for a set of values that define the IPsec features and protections applied to a connection

IKE protocol has two phases:

- **Phase One Exchange** → IPsec endpoints to successfully negotiate a secure channel (*IKE SA*) through which an IPsec SA can be negotiated
- **Phase Two Exchange** → The purpose of phase two is to establish an SA for the actual IPsec connection

Phase One Exchange Modes

- **Main mode.** It negotiates the establishment of the IKE SA through **three pairs** of messages
 - ① In the first pair of messages, each endpoint proposes parameters to be used for the SA
 - ② The second pair of messages perform a key exchange through Diffie-Hellman
 - ③ In the third pair of messages, each endpoint is authenticated to the other

Phase One Exchange Modes

- **Aggressive mode.** It offers a **faster** alternative to main mode. It negotiates the establishment of the IKE SA through **three** messages
 - ① In the first message, endpoint A sends all the protection suite parameters, as well as its portion of the Diffie-Hellman key exchange, a nonce, and its identity
 - ② In the second message, endpoint B sends the protection suite parameters, its portion of the Diffie-Hellman key exchange, a nonce, its identity, and its authentication payload (through digital signature or hash)
 - ③ In the third message, endpoint A sends its authentication payload

Phase One Exchange

In phase one, each endpoint proposes parameters for the SA. The four mandatory parameters are referred to as **protection suite**

- **Encryption Algorithm.** This specifies the algorithm to be used to encrypt data, e.g. DES, 3DES, AES, ...
- **Integrity Protection Algorithm.** This indicates which keyed hash algorithm should be used for integrity protection, e.g. HMAC-MD5, HMAC-SHA-1, ...
- **Diffie-Hellman (DH) Group.** It is used to generate a shared secret for the endpoints in a secure manner.
- **Authentication Method.** There are several possible methods for authenticating the two endpoints (*Pre-shared Keys, Digital Signatures, ...*)

Phase Two Exchange

The purpose of phase two is to establish an SA for the actual IPsec connection (*IPSec SA*). IPSec SA connection is created using three messages:

- In the first message, endpoint A sends keys, nonces, and IPsec SA parameter suggestions. The nonces are an anti-replay measure
- In the second message, endpoint B sends keys, nonces, and IPsec SA parameter selections, plus a hash for authentication
- In the third message, endpoint A sends a hash for authentication

Contents

① Perimetral Security: Firewalls

② Intrusion Detection Systems

③ Point to Point Security

Introduction

VPN Architectures

IPSec Fundamentals

Practical Examples

Questions

ESP in a Gw-to-Gw Architecture

The goal is to establish an IPsec connection that provides **encryption** and **non-complete integrity** protection services between endpoints (gateways) A and B. Initially, we have to create an **IKE SA**, as follows:

- ① Endpoint A **creates** and **sends** a regular (**non-IPsec**) packet that has a destination address of endpoint B
- ② Network A routes the packet to gateway A
- ③ Gateway A receives the packet and performs **NAT**, altering the packet's source IP address
- ④ Gateway A **initiates** an IKE SA negotiation with Gateway B using either main mode or aggressive mode. At the end of the negotiation, the IKE SA **is created**

ESP in a Gw-to-Gw Architecture

The next step is to **create** the IPsec SA, as follows:

- ⑤ GW A uses the parameters set in the IKE SA to initiate an IPsec SA negotiation with GW B. ESP **tunnel mode** is used
- ⑥ Once the two IPsec SAs are created, gateway A **finishes** processing the packet sent by endpoint A in the step 1:
 - ① GW A **modifies** the packet in accordance with the SA parameters: A new IP packet header is added (source IP: GW A and destination IP: GW B), encrypting the data and adding the authentication information
 - ② Gateway A then sends the packet to Gateway B
 - ③ GW B receives the packet and uses the SPI value in the **unencrypted** ESP header to determine the SA parameters. GW B processes and validates the packet: remove the additional IP header, check the integrity and decrypt the original payload
 - ④ GW B sends the packet to endpoint B

AH and ESP in a Gw-to-Gw Architecture

The goal is to establish an IPsec connection that provides encryption and **complete** integrity protection (including headers) services between endpoints (gateways) A and B

Steps 1-4 are identical to the previous example

- ⑤ Gateway A uses the parameter set in the IKE SA to initiate an IPsec SA negotiation with gateway B for the AH service. The IKE SA provides protection for the negotiation of the AH **tunnel** mode
- ⑥ Step 5 is repeated to negotiate the SAs for the ESP service

Contents

① Perimetral Security: Firewalls

② Intrusion Detection Systems

③ Point to Point Security

Introduction

VPN Architectures

IPSec Fundamentals

Practical Examples

Questions

Question I

Say which of the following sentences are true

- ① IPSec works on the application layer
- ② IPSec works on the transport layer
- ③ IPSec works on the network layer
- ④ IPSec works on the data link layer

Question I

Say which of the following sentences are true

- ① IPSec works on the application layer
- ② IPSec works on the transport layer
- ③ IPSec works on the network layer
- ④ IPSec works on the data link layer

Question II

Say which of the following uses of VPNs are correct

- ① G-to-G architecture for remote workers
- ② H-to-H architecture for system administration issues
- ③ H-to-G architecture for establishing secure building connections
- ④ G-to-G architecture for establishing secure building connections
- ⑤ H-to-G architecture for secure dial-up Internet connections

Question II

Say which of the following uses of VPNs are correct

- ① G-to-G architecture for remote workers
- ② H-to-H architecture for system administration issues
- ③ H-to-G architecture for establishing secure building connections
- ④ G-to-G architecture for establishing secure building connections
- ⑤ H-to-G architecture for secure dial-up Internet connections

Question III

AH protocol aims to cover the following issues ...

- ① Data encryption
- ② Integrity protection for packet data
- ③ Integrity protection for packet headers and data
- ④ Integrity protection only for packet headers, ESP also protects the data
- ⑤ In tunnel mode only takes care of the internal packet header

Question III

AH protocol aims to cover the following issues ...

- ① Data encryption
- ② Integrity protection for packet data
- ③ Integrity protection for packet headers and data
- ④ Integrity protection only for packet headers, ESP also protects the data
- ⑤ In tunnel mode only takes care of the internal packet header

Question IV

Message authentication codes (MACs) used in AH ...

- ① only consider the packet information for computing the output
- ② consider the packet information and a key for computing the output
- ③ key is only used to compute the output not for verifying it
- ④ it is impossible to obtain the same output without knowing the key
- ⑤ it is not necessary both VPN endpoints share the same key

Question IV

Message authentication codes (MACs) used in AH ...

- ① only consider the packet information for computing the output
- ② consider the packet information and a key for computing the output
- ③ key is only used to compute the output not for verifying it
- ④ it is impossible to obtain the same output without knowing the key
- ⑤ it is not necessary both VPN endpoints share the same key

Question V

About AH modes ...

- ① transport mode adds a new IP header to the packet
- ② tunnel mode only authenticates the original IP header
- ③ transport mode does not add a new IP header
- ④ transport mode is compatible with NAT
- ⑤ tunnel mode secures the packet from the source and destination hosts

Question V

About AH modes ...

- ① transport mode adds a new IP header to the packet
- ② tunnel mode only authenticates the original IP header
- ③ transport mode does not add a new IP header
- ④ transport mode is compatible with NAT
- ⑤ tunnel mode secures the packet from the source and destination hosts

Question VI

Say which of the following sentences for the ESP protocol are true

- ① it offers encryption for packet payload data
- ② it provides a complete header authentication in tunnel mode
- ③ transport mode does not authenticate packets
- ④ ESP authentication field is always used
- ⑤ ESP trailer field is mandatory

Question VI

Say which of the following sentences for the ESP protocol are true

- ① it offers encryption for packet payload data
- ② it provides a complete header authentication in tunnel mode
- ③ transport mode does not authenticate packets
- ④ ESP authentication field is always used
- ⑤ ESP trailer field is mandatory

Question VII

IKE protocol negotiates, creates, and manages SA. IKE has two phases. Say which of the following sentences for the IKE phases are true

- ① The purpose of phase one is to establish an IPSec SA
- ② The goal of phase one is to negotiate a secure channel for IPSec parameter definition
- ③ Aggressive mode offers less functionality than main mode
- ④ Aggressive mode offers a faster alternative to the main mode
- ⑤ In aggressive mode Diffie-Hellman key exchange is not performed
- ⑥ In aggressive mode Diffie-Hellman key exchange messages are mixed with other protocol messages

Question VII

IKE protocol negotiates, creates, and manages SA. IKE has two phases. Say which of the following sentences for the IKE phases are true

- ① The purpose of phase one is to establish an IPSec SA
- ② The goal of phase one is to negotiate a secure channel for IPSec parameter definition
- ③ Aggressive mode offers less functionality than main mode
- ④ Aggressive mode offers a faster alternative to the main mode
- ⑤ In aggressive mode Diffie-Hellman key exchange is not performed
- ⑥ In aggressive mode Diffie-Hellman key exchange messages are mixed with other protocol messages

Problem I: Romeo and Juliet Chat

Romeo and Juliet are two Italian teenagers living in Verona. They belong to the two principal families of the city. They fall in love but Juliet's fathers try to persuade Juliet to accept a marriage of convenience with Count Paris.

To avoid the marriage Juliet buys a drug to a friar that will put her into a death-like coma for several hours. The Friar promises to send a messenger to inform Romeo of the plan, so that he can rejoin her when she awakens. On the night before the convenience wedding, she takes the drug and, when discovered apparently dead, she is laid in the family crypt.

The messenger, however, does not reach Romeo and, instead, he learns of Juliet's apparent death. Heartbroken, Romeo buys poison from an apothecary and goes to the Capulet crypt. He drinks the poison. Juliet then awakens and, finding Romeo dead, stabs herself with his dagger.

Problem I: Romeo and Juliet Chat

So sad... Please, help Juliet to establish a secure VPN with Romeo to be sure he receives her message privately.

Romeo and Juliet have two laptops but they share their Internet connections with their parents. As Juliet's father is really interested in the marriage of convenience of Juliet, he buys a complete set of network inspection. Therefore, Juliet's home network is insecure. In addition, Romeo has no access to the router of his family and he cannot modify the routing parameters, however as their families are rich their laptops have a public IP address.

Questions:

- 1 Say which VPN architecture and IPSec options are the most convenient for Romeo and Juliet
- 2 Write the protocol to deploy the VPN defined before

Problem I: Romeo and Juliet Chat

- 1 Say which VPN architecture and IPSec options are the most convenient for Romeo and Juliet

A Host-to-host Architecture with ESP and AH using a transport mode seems to be the best choice. Why?

- ESP ensures data cannot be read
- AH ensures authentication
- As both laptops have public IP addresses NAT is not necessary
- Host-to-host architecture ensures that packets can cross any insecure local network bypassing the network inspection bought by Juliet's father.
- It's not necessary for Romeo to modify his router

Problem I: Romeo and Juliet Chat

- 2 Write the protocol to deploy the VPN defined before

The first step in establishing the connection is to create an IKE SA, as follows:

- 1 Endpoint A (Juliet) creates a regular (non-IPsec) packet that has a destination address of endpoint B (Romeo). When endpoint A attempts to send this packet, its IPsec client software determines that ESP and AH should be applied to the packet.
- 2 Endpoint A initiates an IKE SA negotiation with endpoint B using either main mode or aggressive mode. At the end of the negotiation, the IKE SA is created

Problem I: Romeo and Juliet Chat

- 2 Write the protocol to deploy the VPN defined before

The next step in establishing the ESP and AH connection is to create IPsec SAs, as follows:

- 1 Endpoint A uses the parameters set in the IKE SA to initiate an IPsec SA negotiation with endpoint B for the AH service. The IKE SA provides protection for the negotiation. The parameters specify that AH transport mode will be used. At the end of the negotiation, a pair of unidirectional SAs is created for the tunnel
- 2 Previous step is repeated to negotiate the SAs for the ESP service

Problem I: Romeo and Juliet Chat

- 2 Write the protocol to deploy the VPN defined before

Once the four IPsec SAs have been created, endpoint A can finish processing the initial packet:

- 1 Endpoint A modifies the packet so it is protected in accordance with the SA parameters. ESP is applied first, then AH, which provides integrity for ESP packet portions
- 2 Endpoint A then sends the packet to endpoint B.
- 3 Endpoint B receives the packet and uses the AH header SPI value to determine which SA should be applied.

Endpoint B processes and validates the packet, in terms of AH. Next, Endpoint B uses the value in the unencrypted ESP header SPI field to determine which SA should be applied next. Then, endpoint B processes and validates the packet, in terms of ESP

5 - Network Security

Jordi Nin

nin@ac.upc.edu

Department of Computer Architecture (DAC)
Universitat Politècnica de Catalunya (UPC)
Computer Security (SI)