Lecture 3
Firewalls and IPS

Print version of the lecture in *CPS633 Computer Security*

presented on Week of September 16, 2019

by Ali Miri ©2019from Department of Computer Science at Ryerson University

3.1

Learning Objectives (page 289)

- Explain the role of firewalls as part of a computer and network security strategy.
- List the key characteristics of firewalls, and Discuss the various basing options for firewalls.
- Understand the relative merits of various choices for firewall location and configurations.
- Distinguish between firewalls and intrusion prevention systems.

This set of slides closely follows Chapter 9 of the textbook. 3.2

Contents

1 Preliminaries **1**
1.1 Requirements . . . 2
1.2 Characteristics . . . 3
1.3 Design Goals . . . 3

2 Types **3**
2.1 Packet Filtering . . . 3
2.2 Stateful Inspection . . . 5
2.3 Application Proxy . . . 9
2.4 Circuit-Level Proxy . . . 10
2.5 Host-based/Personal . . . 11

3 Locations and Configurations **11**
3.1 DMZ . . . 11
3.2 VPN . . . 12
3.3 Distributed FWs . . . 12

4 IPS **15**
4.1 HIPS . . . 16
4.2 NIPS . . . 16
4.3 Hybrid . . . 17
4.4 Suricata . . . 17
4.5 UTM . . . 17

3.3

1 Preliminaries

A General Model for Computer Security

Countermeasures:

From FIPS 'Framework for Improving Critical Infrastructure Cybersecurity', Version 1.1 (April 16, 2018) available at https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf

3.4

Basic Idea

Q: How to protect systems against network-based threats?

Q: Are these threats posed by outside or inside attackers?

Typical decisions:

- Allow
- Reject and notify
- Discard and not notify

Can you consider at least one scenario in which each of the last two policy options can be useful/needed?

3.5

1.1 Requirements

Basic requirements

Basic, common requirements we may want in (any) applications

- powerful/capable solution
- convenient
- inexpensive
- quick to install, and easy to use

Here, our focus is on *firewalls*, and other types of *Intrusion Prevention Systems*

3.6

An Old(?) Solution

Can you think one pro and con for this (type) of solution? 3.7

Some key points:

- policy-based
  - where it is coming from or going to ← `traffic`
  - what it is for ← `application`
  - who is it from ← `user`
  - what does it carry ← `content`

3.8

1.2 Characteristics

Filtering Characteristics

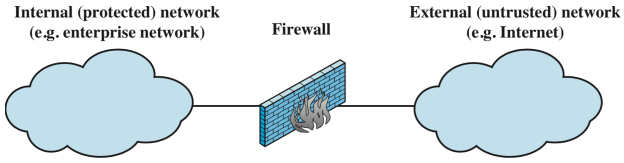
Typical characteristics used for a firewall access policy include:

- IP Addresses and Protocols
- Applications
- User Identity
- Network Activity
- . ...

For more details, see Section 4 of NIST SP 800-41 (*Guidelines on Firewalls and Firewall policy*) available at `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-41r1.pdf` 3.9

1.3 Design Goals

Design Goals



Basic design goals and considerations:

- single choke point
  - This can allow for offering other services: IPSec, Auditing, NAT, etc.
- only authorized traffic should get through
- the firewall itself should be immune from attacks

3.10

2 Types

2.1 Packet Filtering

IP Addresses and Protocols

Firewalls can monitor and manage traffic based on different types of network information.

Attempt 1: **Packet filtering** - Apply a set of rules to each individual incoming/outgoing packet.

Example 1: Write a firewall access policy that allows any type of communication between a user's machine inside your network with an IP address 10.0.2.12 and the user's machine at home with an IP address 174.90.120.3.

3.11

Packet Filtering

Example 2: Now supposed that you only want to allow to 'only' allow SMTP communication between the machines in the previous example, i.e. 10.0.2.12 and 174.90.120.3. How would you achieve that? Note that SMTP runs on TCP and uses port 25 for the server and port $\geq$ 1024 for the client.

Important: Packet filtering implements set of rules based on Internet and Transport layers info. Typically, the first instance of the rules that match a packet is applied.

- What happens if there are other rules matching the same packet? - What if there are no rules about that packet in the list?

Two *default* policies:

- **Permissive**: Let every packet through. Will ensure accessibility, but $\cdots$
- **Restrictive**: say no to all. secure, but $\cdots$

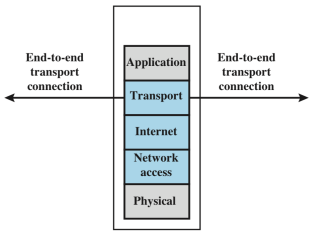
3.12

Packet Filtering

Typical information used in packet filtering (based on information on a single network packet)

- Source and destination IP addresses
- Source and destination transport-layer addresses
- IP protocol field
- Interface

Aside: *Packet Filtering Firewall* are considerd an OSI layer 3 solution, or a TCP/IP Internet layer solution, even though they may use some of the information for the transport layer.



(b) Packet filtering firewall

Packet Filtering

Rule	Direction	Src address	Dest addresss	Protocol	Dest port	Action
1	In	External	Internal	TCP	25	Permit
2	Out	Internal	External	TCP	> 1023	Permit
3	Out	Internal	External	TCP	25	Permit
4	In	External	Internal	TCP	> 1023	Permit
5	Either	Any	Any	Any	Any	Deny

Advantageous of packet filtering firewalls:

- *Simple* (Although they can be rather complex as the number of rules increases)
- *Transparent to users*
- *Speed*

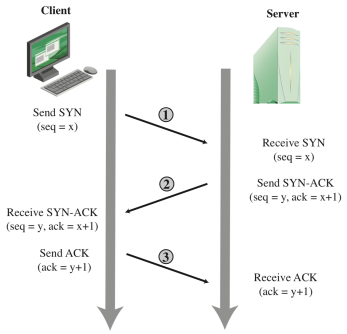
Packet Filtering

Disadvantageous of packet filtering firewalls:

- Can be complex to create and maintain the rule set ← improper configuration can have serious consequences
- Does not examine Application layer data ← similar to far-sighted vision that can see the rough shape, but cannot quite make the object!
- Does not support user authentication
- Provide for very limited logging
- Susceptible to certain TCP/IP protocol attacks

2.2 Stateful Inspection

TCP ACK Scan Attack

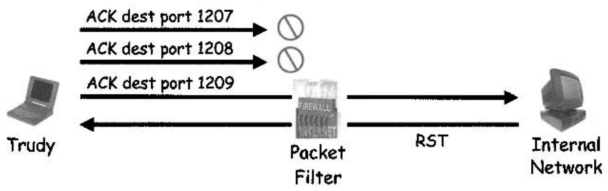


Attacker aim: Find an open port on a network protected by a packet filter

3.16

TCP ACK Scan Attack

1. Send a packet that has the ACK bit set, without the prior two steps of the TCP three-way handshake
2. Observe the response which will be
   - *no response/unreachable* $\rightarrow$ filtered
   - *RST* $\rightarrow$ not filtered, open port



Source: 'Information Security: Principles and Practice' by Mark Stamp, page 290

3.17

Firewalk Attack

A firewall is generally expected to hide the details and the topology of the protected network from the outside world. *Firewalk attack* is an active active reconnaissance (see `https://tools.kali.org/information-gathering/firewalk` to exploit a related vulnerability in packet filters (similar to previous attack, but different implementation)

Assumptions: Attackers knows

- the IP address of the firewall
- the IP address of one system on the inside network
- the IP address of one system on the inside network, say 'x'

3.18

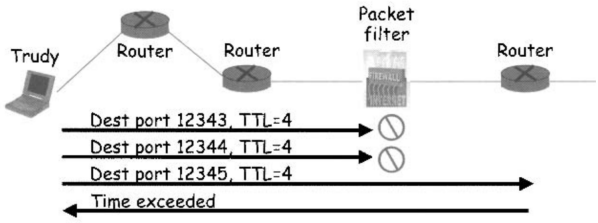
Firewalk Attack

How does it work?

- To check port p, attacker sends a packet to the IP address of the known host inside the firewall, with the TTL field set to $x + 1$
  - No response, If the firewall does not let data through on port p

- OW, 'a time exceeded error message'

- Repeat this process for different ports p to determine open ports



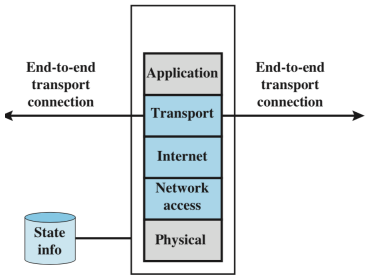
Source: 'Information Security: Principles and Practice' by Mark Stamp, page 292

See you also 'firewalk Usage Example' at `https://tools.kali.org/information-gathering/firewalk`

3.19

Part of limitation and weaknesses of packet filtering is due to

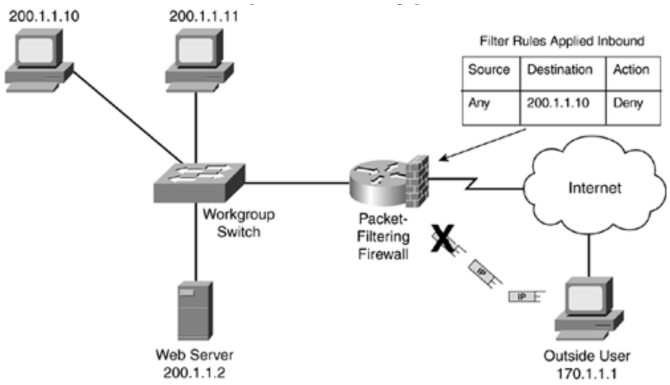
- lack of consideration for any higher-layer context
- rules applied to individual packets. ← it is *stateless*
- the filter has to allow for incoming traffic on high number ports, as illustrated in the next example.



(c) Stateful inspection firewall

3.20

Example



This example and the associated figures are taken from `http://etutorials.org/Networking/Router+firewall+security/Part+I+Security+Overview+and+Firewalls/Chapter+2.+Introduction+to+Firewalls/Firewall+Categories/`

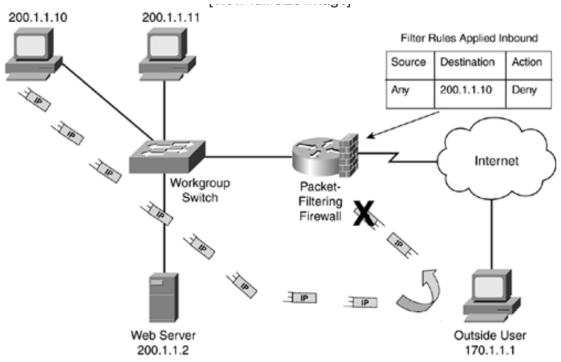
3.21

Example

• Suppose now that a web client on 200.1.1.10 send a HTTP request for web services to a server on 170.1.1.1.

• HTTP uses TCP, and TCP goes through its three-way handshake - SYN, SYN/ACK, and ACK - to establish a session before data transfer ← initially, it will send a SYN with destination port 80, and a source port > 1023

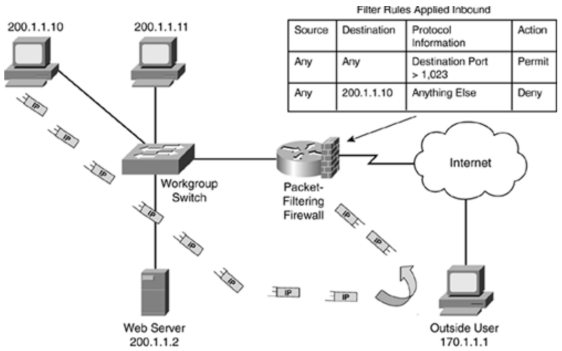
• If the packet filtering is set to allow all outgoing traffic, the SYN packet will go through, but not the reply from the server!



Example

Q. How to handle this?

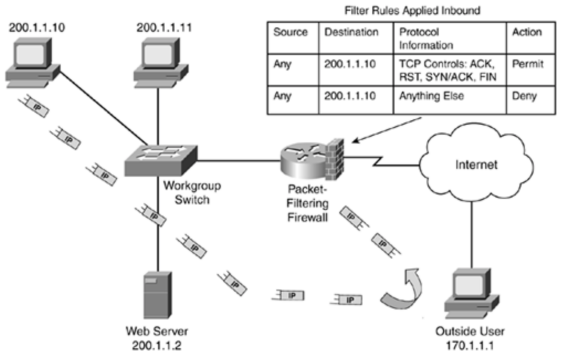
Possible solution 1: Open destination ports greater than 1023 to incoming traffic



Huge security hole in the firewall ← not recommended!

Example

Possible solution 2: Use the TCP control bits



- Not all transport layer protocols support control codes
- Attackers can manipulate the TPC control bits

3.24

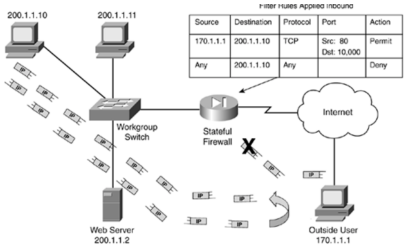
Example

Possible solution 3: Use State Table (*Stateful inspection firewall*): use a mechanism to keep track of the state of a connection

- How? When the stateful firewall receive the SYN request from 200.1.1.10, checks to see if it allows. If so, adds a filtering rule to its configuration by added a new rule to the top of the existing filtering rule set or is placed into a state table.

- The state information is only kept while session is alive.

Sr Add	Sr Port	Dest Add	Dest Port	Conn State
200.1.1.10	10,000	170.1.1.1	80	Established
200.1.1.10	10,100	170.1.1.1	25	Initiated



3.25

Stateful Inspection Firewalls

Q. How about non-stateful protocols, such as UDP, DNS, ICMP, $\cdots$?
The FW can use a timer, but this can potentially be costly.

Advantageous of Stateful Inspection Firewalls

- Stateful firewalls are aware of the state of a connection
- Stateful firewalls do not have to open up a large range of ports to allow inbound communication
- Stateful firewalls can prevent more attacks, such as certain types of DoS attacks, than their packet filtering counterparts.

3.26

Stateful Inspection Firewalls

Disadvantageous of Stateful Inspection Firewalls (they still share some of packet filtering firewalls):

- Can be complex to maintain the rule set
- Does not examine Application layer data
- Does not support user authentication
- Not all protocols have stateful information
- They have (slightly?) more overhead in maintaining a state table
- They provide more stringent controls over security than packet filtering

3.27

2.3 Application Proxy

Application-Level Gateway

Application-Level Gateway, or Application Proxy

‘ `Catch me if you can!` ’ or ‘threats are coming along for the ride!’

Applications (i. e. threats) can be evasive:

- port hopping
- use of non- standard ports
- tunnelling within commonly used application
- hiding within SSL encryption*
- ...

One possible solution: *block applications!*

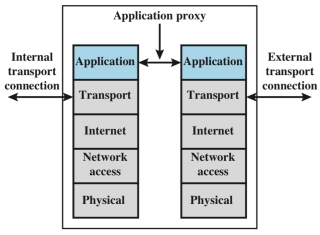
---

* check out an interesting article by John Pirc (NSS Labs) at `http://bemonitor.com.mx/docs/competitive/SSL%20Performance%20Problems.pdf`

3.28

Application-Level Gateway

Alternative solutions: Setting up a proxy that intercept traffic to/from an application (server), and inspect it first!



(d) Application proxy firewall

3.29

Application-Level Gateway

- All the network layer information (including Application) layer can be used → *Deep Pack Inspection* vs *Shallow Packet Inspection*
- It can be used to *monitor* and *filter* the data, as well as blocking it.
- User authentication can be enforced for support applications
- It can provided detailed logs.

3.30

Application-Level Gateway

However,

- May need to have a separate proxy for each application to be protected
- Not all applications used may be supported
- Process traffic in software
- Can result in a large performance overheads
- May require specialized client sofware
- It can have issues with encrypted traffic

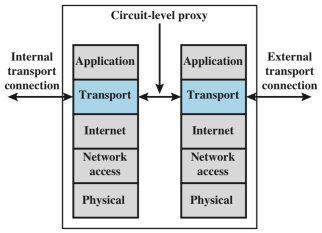
3.31

2.4 Circuit-Level Proxy

Circuit-Level Proxy

Circuit-Level Gateway, or Circuit-Level Proxy

- a stand-alone system, or as part of specialized function performed by an application-level gateway
- as a gateway sets up two Transport-layer connections, one to each sides of communication
- typically relays Transport-layer segments (once established) from one end to another without examining the contents
- typical use could be fore outbound traffic



(e) Circuit-level proxy firewall

3.32

Circuit-Level Proxy

A commonly used example of a circuit-level proxy: *Socket Secure (SOCKS)* - see the latest RFC draft at https://tools.ietf.org/id/draft-olteanu-intarea-socks-6-05.html

- performs at OSI Layer 5 - the *session layer*
- SOCKS server runs at TCP port 1080
- the protocol manages both TCP and UDP
- Since version 5, it also offers user authentication

SOCKS is used by Tor. Onion routing protocol.

3.33

Bastion Host

Q. Where/how to implement, and in particular in case proxy-type firewalls? SW/HW? → *bastion host*

- Make sure the host platform runs a secure version of OS. Generally, allow for read-only access to the configuration files.
- Only install essential services, including proxy applictions
- may require additional user authentication to access the proxy services
- Configure to only support a subset of the standard application's command set, when possible.
- Only allow access to specific host systems
- If more than one proxy, run them independent of each other. Run each proxy in a private, secure directory, as a non-privileged user mode

3.34

2.5 Host-based/Personal

Host-based Firewalls

- A software module used to secure an individual host → can be tailored to the host
- It is independent of network topology ← both internal- , external-based attacks have to go through it
- It is typically used with stand-alone firewalls, as additional layer of defence

Note: Network devices such as routers and switches can be made to provide FW functionalities, such as packet filtering and stateful inspection. Virtual firewalls have also become a common alternative.

3.35

3 Locations and Configurations

3.1 DMZ

DMZ

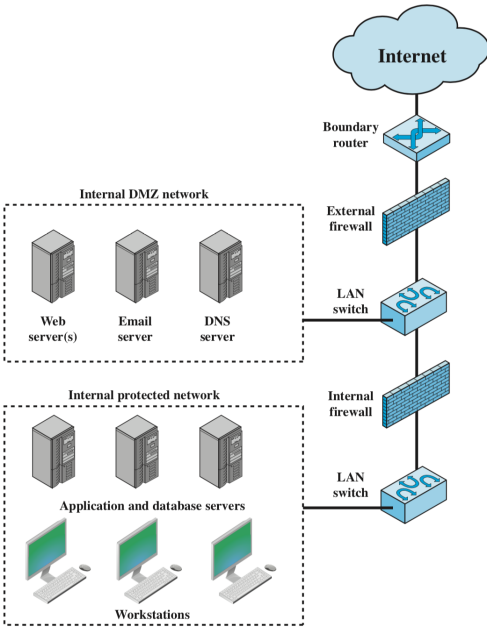
```
'Not all assets are equal value, or need the same level of
                        access!'
```

(DeMilitarized Zone (DMZ) Network

- Create a segment of networks for (public) services such as Mail, Web, and DNS
- Deploy multiple firewall with different stringent filtering capabilities to protect DMZ and the rest of the network
- Watch out for attacks from (own) DMZ

3.36

DMZ

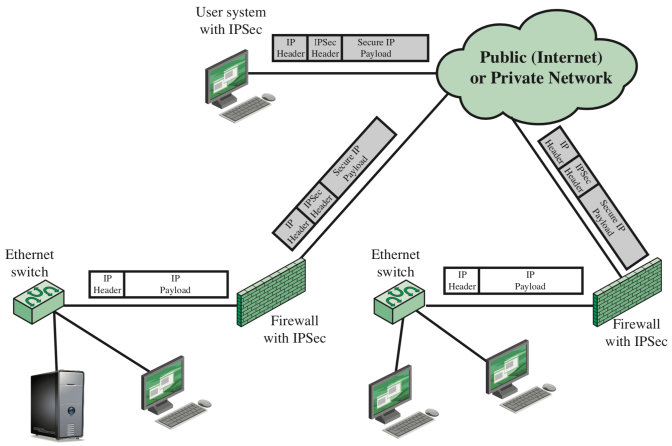


3.37

3.2 VPN

VPN

Virtual Private Networks (VPNs): providing authorized outside access over public/insecure channels → security procedures and tunnelling protocols at the IP level: IPSec



3.38

3.3 Distributed FWs

Distributed FWs

'Different tools for different jobs, and strength in depth!'

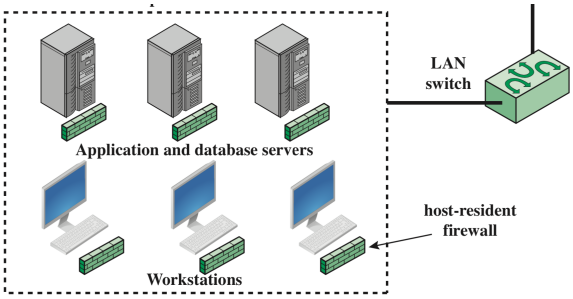
- Different assets have different values, and different visibility → use multiple, distributed FWs

- But which ones, and where?
- Make sure that there is a coordinated monitoring and analysis, and that all nodes in the system are covered.

3.39

Locations and Topologies

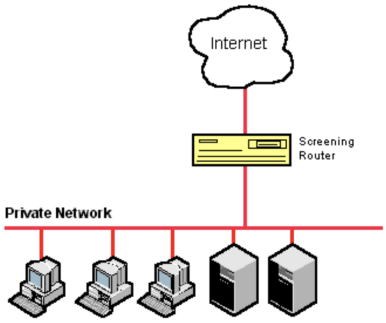
Host-based, or Host-resident FWs



3.40

Locations and Topologies

Screening Routers: with stateless or full packet filtering

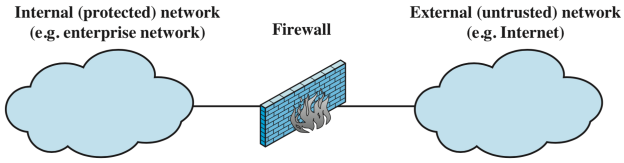


source: https://www.pcmag.com/encyclopedia/term/50923/screening-router

3.41

Locations and Topologies

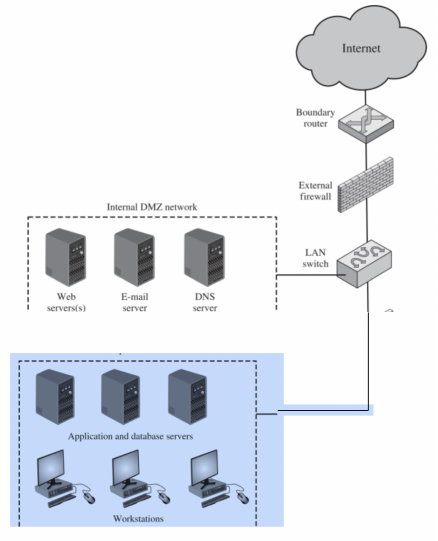
Single Bastion Inline: with stateful filters and/or application proxies



3.42

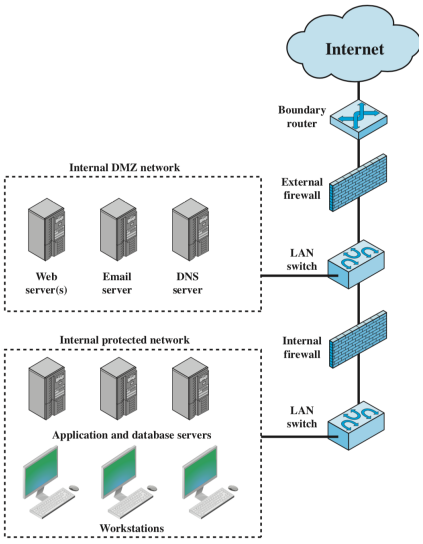
Locations and Topologies

Single bastion T: same as Single Bastion Inline, but has a third (DMZ) network interface



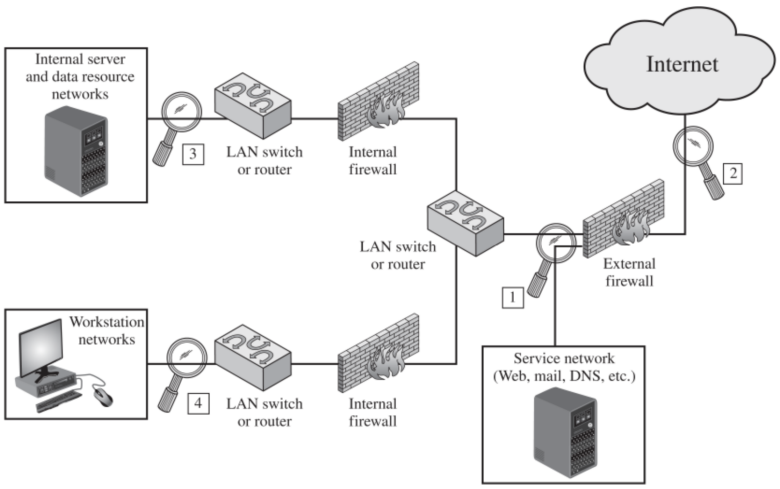
Locations and Topologies

Double bastion inline: DMZ is sandwiched between two bastion firewalls



Locations and Topologies

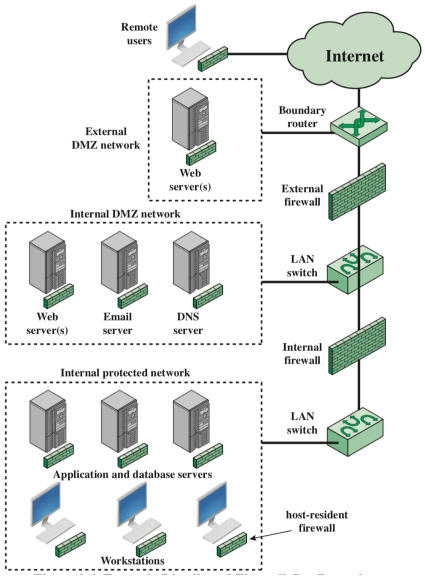
Double bastion T



3.45

Locations and Topologies

Distributed



3.46

4 IPS

Intrusion Prevention Systems (IPs)

Idea: make system protection stronger by combining it with Intrusion Detection Systems (IDSs) ← sometime is referred to as *Intrusion Detection and Prevention Systems (IDPSs)*

IPS categories:

- Host-Based IPS (HIPS)
- Network-Based IPS (NIPS)
- Hybrid IPS

3.47

4.1 HIPS

Host-based IPS (HIPS)

HIPS can provide end-point protection using:

- Signature-based detection techniques
- Anomaly-based detection techniques
- Sandbox approach

3.48

Host-based IPS (HIPS)

Malicious behaviours/attacks focused on may include:

- Modification of system resources
- Privilege-escalation exploits
- Buffer-overflow exploits
- Directory traversal
- Access to e-mail contact list, or similar type lists

Protection offered by IPS may include;

- System calls
- File system access
- System registry settings
- Host input/output

3.49

4.2 NIPS

Network-based IPS (NIPS)

NIPS used its detection capabilities to modify or discard packets or tear down connections

Two main categories of NIPS are:

- signature/heuristic detection-based
- anomaly detection-based

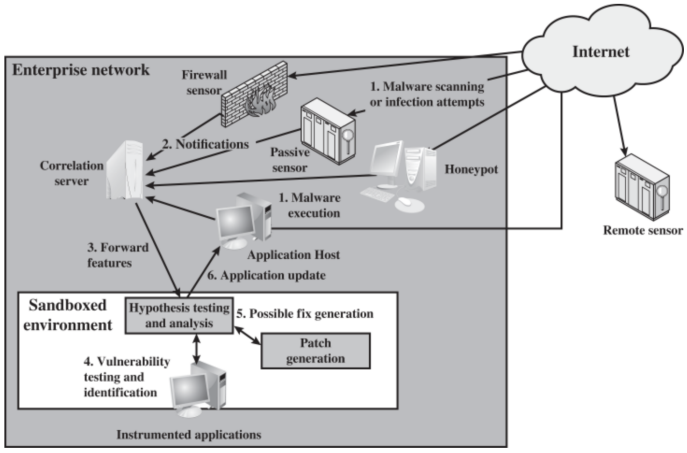
Typical techniques to detect identify malicious packets may include;

- Pattern matching
- Stateful matching
- Protocol anomaly
- Traffic anomaly
- Statistical anomaly

3.50

4.3 Hybrid

Hybrid IPS



IBM Digital Immune System

3.51

4.4 Suricata

Suricata

Hybrid IPS

Suricata: An “a free and open source, mature, fast and robust network threat detection engine" - available at `https://suricata-ids.org`

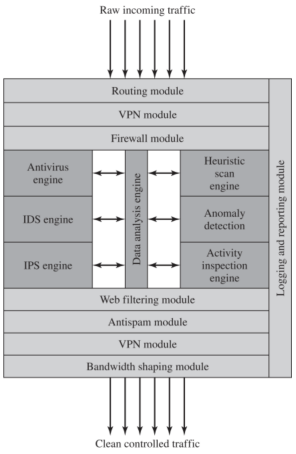
- Real time Intrusion Detection (IDS)
- Inline Intrusion Prevention (IPS)
- Network Security Monitoring (NSM)
- Offline pcap processing

3.52

4.5 UTM

UTMs

Unified Threat Management (UTM) systems: “Products that include multiple security features integrated into one box” (page 311)


Raw incoming traffic
Routing module
VPN module
Firewall module
Antivirus engine
IDS engine
IPS engine
Data analysis engine
Heuristic scan engine
Anomaly detection
Activity inspection engine
Logging and reporting module
Web filtering module
Antispam module
VPN module
Bandwidth shaping module
Clean controlled traffic

3.53