System and Network Administration

Firewalls &
Intrusion Detection

Secure Systems

1. Security policy
   - **What needs to be protected**
   - **Kinds / level of protection**
   - **Responsibilities**
   - **Auditing policy**

2. Security environment
   - **Physical environment**
   - **Physical security**
   - **Hardware, operating system**
   - **firewalls, etc**

3. Security mechanisms
   - **cryptography**
   - **authentication**
   - **security protocols**

4. Monitoring and auditing procedures
   - **monitor access**
   - **audit trails**
   - **feedback on failures, security breaches**
   - **containment & recovery**

Controlling Access to Services

"Trust is a substitute for security"

"More Security = Less Convenience"

Trusted hosts – listed in `/etc/hosts.equiv`

`rsh, rlogin`	*remote shell/login for command execution (like telnet, but not interactive)*
`rcp, rdist, rsync`	*push files*

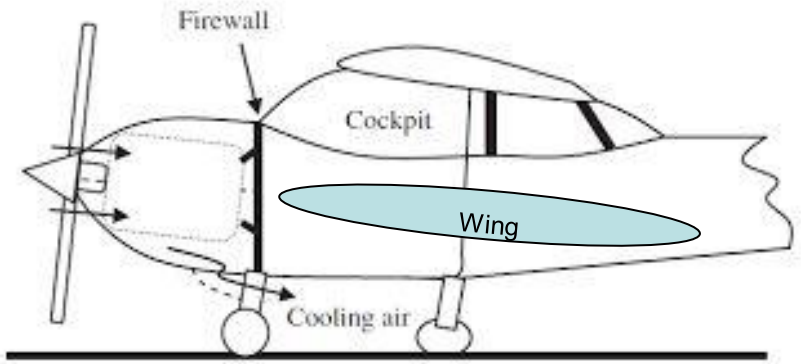
- Insecure, but useful for maintenance
- Must be carefully controlled

Security Policy

- Strict control over access to system configuration files
- Control privileges
  - Users should only have access to the services that they have been specifically authorised to use and any access by remote users or connections to remote computer systems must be authenticated.
- Encrypted communication channels
  - Close telnet and ftp ports – use ssh
- Monitor system use and performance
- Track reports of attacks and vulnerabilities
- Test and evaluate failure modes
  - Avoid single point of failure

Firewall

- In automotive engineering, the firewall separates the engine compartment from the passenger compartment (driver and passengers).
- The name originates from steam-powered vehicles, where the firewall separated the driver from the fire heating the boiler.

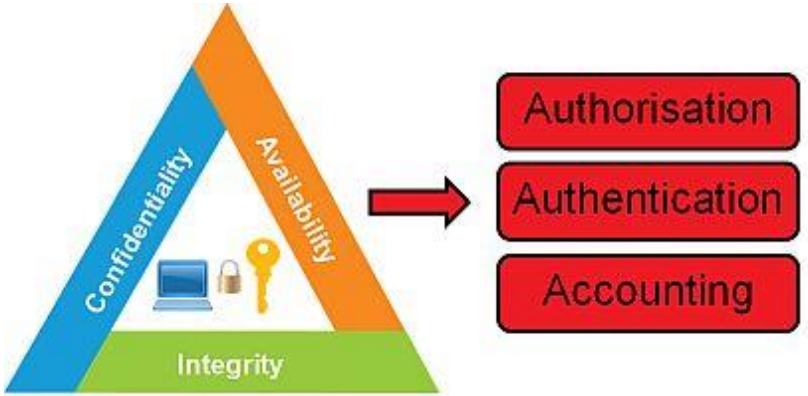


Firewall

- In computer networking, a firewall is hardware or software or combination of both designed to prevent unauthorized access to or from a private network
  - The firewall examines all messages entering or leaving the internal network and blocks those that do not meet specified security criteria
- It is placed at the junction point or gateway between the two networks, which is usually a private network and a public network such as the internet
- ***To be effective, the firewall must be the only interconnection device between the networks.***

AAA

- Firewalls provide a *single block point*, where security and auditing can be imposed.
- Firewalls provide an important *logging and auditing* function, providing summaries to the Administrator about what type/volume of traffic has been processed through it.
- This is an important benefit: Providing this block point can serve the same purpose on the network as an armed guard does for physical premises.




Q
&
A

Types of Firewalls

- **Network layer firewalls**
  - Network layer firewalls make their decisions based on the source address, destination address and ports in individual IP packets.
  - A *simple router* is the traditional network layer firewall: not able to make particularly complicated decisions about what a packet is actually talking to or where it actually came from.

Types of Firewalls

- **Proxy (application layer) firewalls**
  - Traditionally, application layer firewalls are hosts that run *proxy servers*. They do not permit traffic directly between networks.
  - They can thoroughly examine traffic passing through them, and perform elaborate logging.
  - They can also be used for Network Address Translation (NAT, which uses port forwarding), to effectively mask the origin of an initiating connection.

Types of Firewalls

- **Layer 7 firewalls**
  - Also called
    - application layer gateways
    - deep packet inspection
    - next generation firewalls
  - These watch traffic at the packet level but examine the data within each one. They are able to decode that information and determine exactly what application is generating those data frames.
    - e.g., Microsoft SQL Server, Twitter, Facebook, etc.

Types of Firewalls

- **Host-based firewall**
  - These run on the local operating system.
    - It already knows what applications you're using.
  - Allows you to define rules that allow or disallow specific executables to run and communicate across the network.
    - Good example: Windows Firewall -> Advanced Security settings

Types of Firewalls

- **“Stateful” firewall**
  - A firewall that can keep track of TCP sessions that are currently active
    - More properly refers to filtering rules rather than the firewall itself
  - Only TCP maintains state (3-way handshake)
  - IP, UDP, ICMP are stateless (every packet is independent)


Q
&
A

Access Control Lists (ACLs)

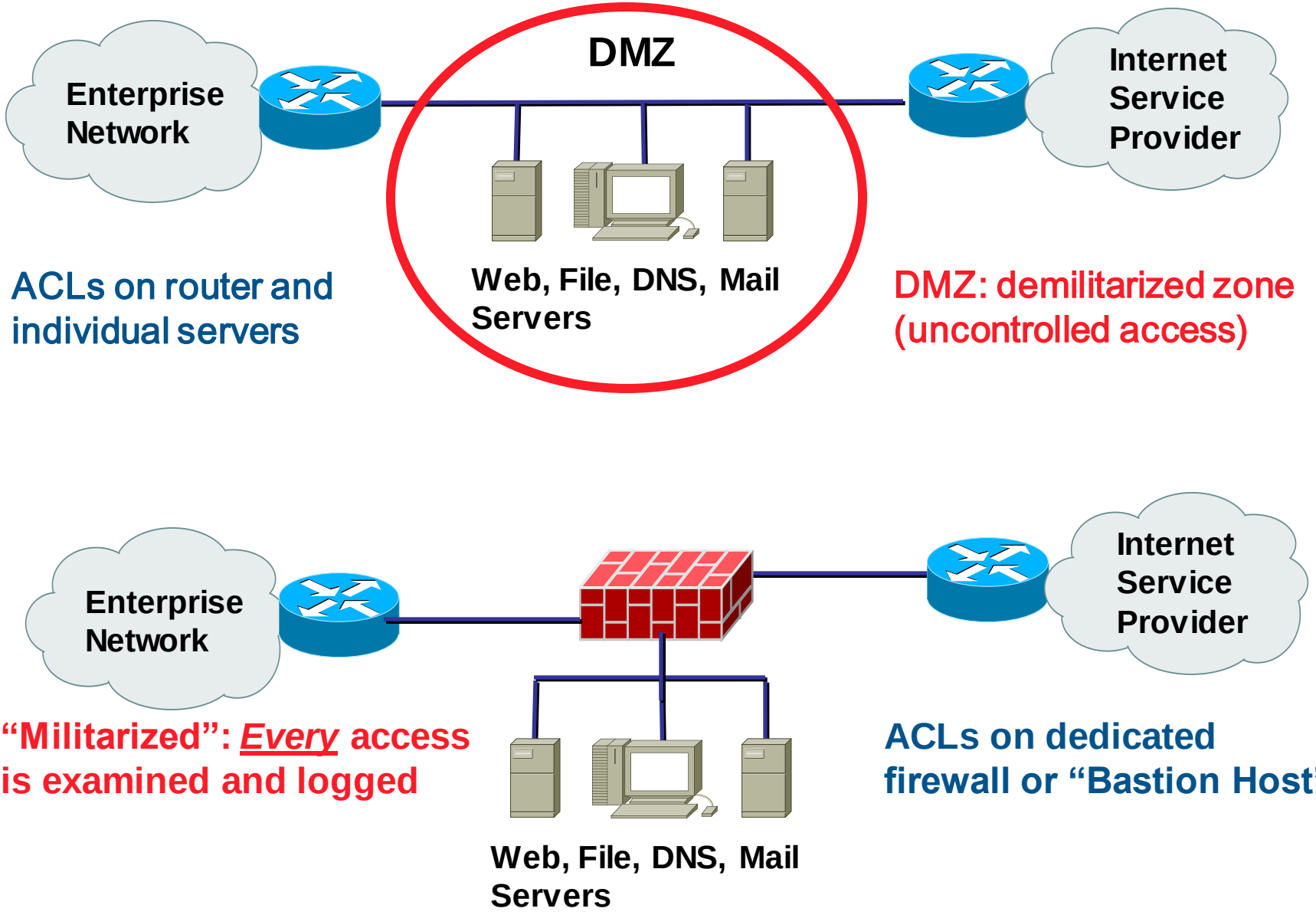
Access Control Lists (ACLs) can be set up on routers and servers to accept or deny packets from particular addresses or services.

Security policy: packet filter will either

1) **Deny specific types of packets and accept all else**
   - requires a thorough understanding of specific security threats
   - can be hard to implement
2) **Accept specific types of packets and deny all else**
   - does not have to predict future attacks
   - requires a good understanding of network requirements

ACL

- Firewall rule lists are usually examined from the top of the list, down through to the bottom.
- Firewall will look at the first rule, see if anything matches in that particular rule. If it doesn't, it goes to the next rule and see if that rule matches the traffic, and so on.
- On most firewalls, at the bottom of the list is an "implicit deny". This means that if the traffic coming through the firewall doesn't match any of the rules above, it is dropped by the firewall.


Enterprise Network
DMZ
Internet Service Provider
Web, File, DNS, Mail Servers
ACLs on router and individual servers
DMZ: demilitarized zone (uncontrolled access)
Enterprise Network
Internet Service Provider
"Militarized": Every access is examined and logged
ACLs on dedicated firewall or "Bastion Host"
Web, File, DNS, Mail Servers


Q
&
A

Packet Filtering: iptables

- Configure tables of packet-filter rules in Linux kernel
  - Each table has a number of *chains*
  - Each chain consists of a list of rules
  - Each rule specifies what to do with a matching packet

- The default table (*filter)* has 3 built-in chains:
  - INPUT controls traffic from outside the LAN
  - OUTPUT places restrictions on outbound connections
  - FORWARD allows control over routing between interfaces

iptables - rules

IPTables uses policies (-P) to create default rules

- The following rules will block all traffic on a network gateway,

```
iptables -P INPUT DENY
iptables -P OUTPUT REJECT
iptables -P FORWARD REJECT
```

  - DROP and DENY silently drop packets, REJECT returns a *connection refused* error to users

- Specific rules are appended (-A) at the end of an existing ruleset.

  -A is followed by the name of the chain for a rule. then Flags like

  - (p)rotocol
  - (i)nterface
  - (s)ource (d)estination (sport) (dport)

iptables – Rule examples

```
# refuse telnet connections
iptables -A INPUT -p tcp --sport telnet -j REJECT
iptables -A INPUT -p udp --sport telnet -j REJECT

# allow ssh connections
iptables -A INPUT -p tcp --sport 22 -j ACCEPT
iptables -A INPUT -p udp --sport 22 -j ACCEPT

# Allow new TCP connections from hosts in the
# 192.168.1.0/24 network to port 137 (stateful)
iptables -A INPUT -m state --state NEW -m tcp -p tcp
         -s 192.168.1.0/24 --dport 137 -j ACCEPT
```

Rules are activated at boot time
Rules are defined in `/etc/sysconfig/iptables`

Actual rules

Allow new TCP connections from hosts in the 192.168.1.0/24 network to port 137 (stateful)

```
iptables -A INPUT -m state --state NEW -m tcp -p tcp
-s 192.168.1.0/24 --dport 137 -j ACCEPT
```

Once an approved connection is up, go no further than this rule

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Actual rules

--state
A comma separated list of the connection states to match.
Possible states are:

NEW the packet has started a new connection, or otherwise associated with a connection which has not seen packets in both directions

ESTABLISHED the packet is associated with a connection which has seen packets in both directions

RELATED the packet is starting a new connection that is associated with an existing connection, such as an FTP data transfer or an ICMP error

NSF Shares: Protecting Portmap

- Also add iptables rules to the server restricting access to specific networks.
  - This example allows connections to the portmap service (listening on port 111) from the 192.168.0/24 network and from localhost - All other packets are dropped.

```
iptables -A INPUT -p tcp -s! 192.168.0.0/24 --dport 111 -j DROP
iptables -A INPUT -p udp -s! 192.168.0.0/24 --dport 111 -j DROP
iptables -A INPUT -p tcp -s  127.0.0.1 --dport 111 -j ACCEPT
iptables -A INPUT -p udp -s  127.0.0.1 --dport 111 -j ACCEPT
```

TCP Wrapper

- In general terms, a TCP wrapped service is one that has been compiled with the `libwrap` library
  - These typically include sshd, sendmail, xinetd, etc.
- When a connection attempt is made the service checks if connection is allowed based on rules in `/etc/hosts.allow` and `/etc/hosts.deny`
  - If no rules for the service are found in either file, or if neither file exists, access to the service is granted.
  - Changes to the `hosts.allow` and `hosts.deny` are dynamic: they take effect as soon as the file is saved.

TCP Wrapper - Examples

```
ALL : .example.com EXCEPT cracker.example.com
```

- If this rule appears in /etc/hosts.allow, all example.com hosts are allowed to connect to all services except cracker.example.com

Only specifically allowed

- If this rule appears in /etc/hosts.deny, only cracker.example.com has access to all services, and other example.com hosts have access to none

Only specifically denied

So, what is the effect of this?

```
ALL EXCEPT telnetd : 192.168.0.
```

`hosts.deny` can simply specify

`ALL:ALL`

With all specific rules in `hosts.allow`

blocked unless explicitly allowed is also called **Implicit deny**

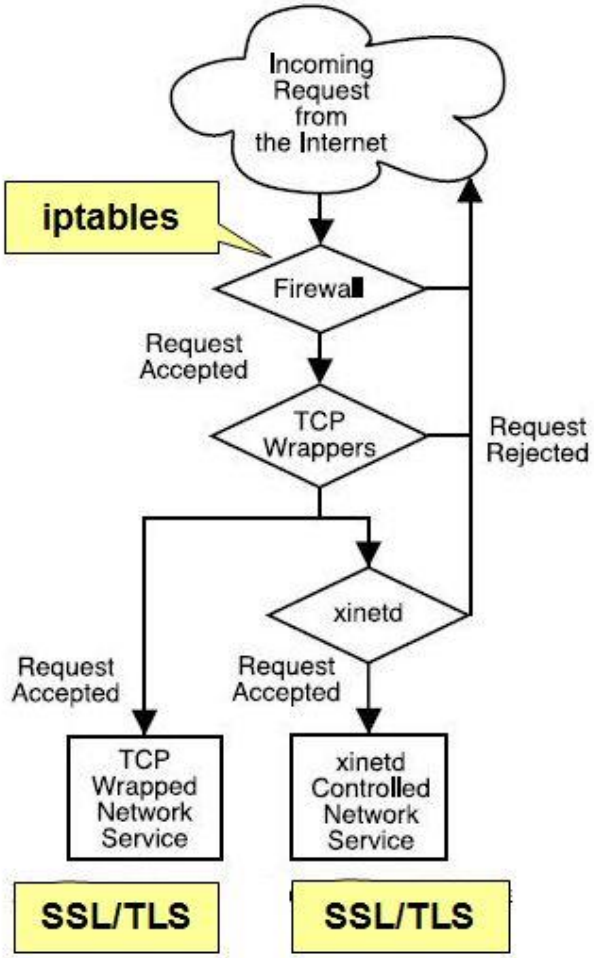
IPtables or TCPwrap ?

TCPwrap is simpler, rules take effect immediately, can use domain names

IPtables is faster, but rules are more difficult to write and require restart, limited to layers 3 and 4

No need for redundant rules
Lots of ways to divide things up

Need to use the logfiles over time to see traffic patterns and make informed decisions



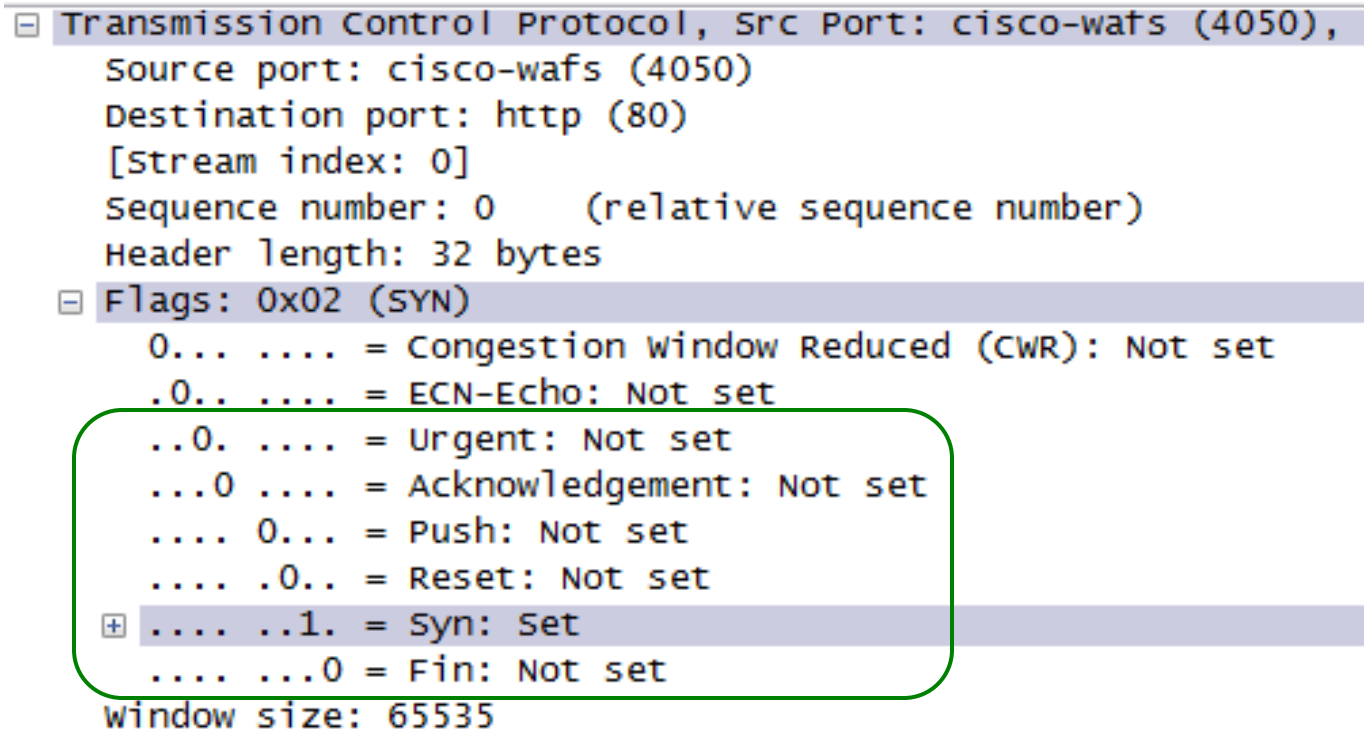

Q
&
A

hping

- hping is **a packet assembler/analyzer used for crafting TCP/IP packets.**
- It sends empty (0 data byte) packets by default, but has lots of options for tuning the payload
- You can experiment probing and analysing your firewalls and network by sending a variety of probe types with different flag settings.

Background: TCP

- The TCP segment (packet) header has six flag bits that can be set independently.
- A mnemonic to remember them in order is
  Unskilled **A**ttackers **P**ester **R**eal **S**ecurity **F**olks.



http://www.lovemytool.com/blog/2010/07/practical-tcp-series-tcp-flags-by-chris-greer.html

Background: TCP

Four flag bits can be set to describe the state of the connection:

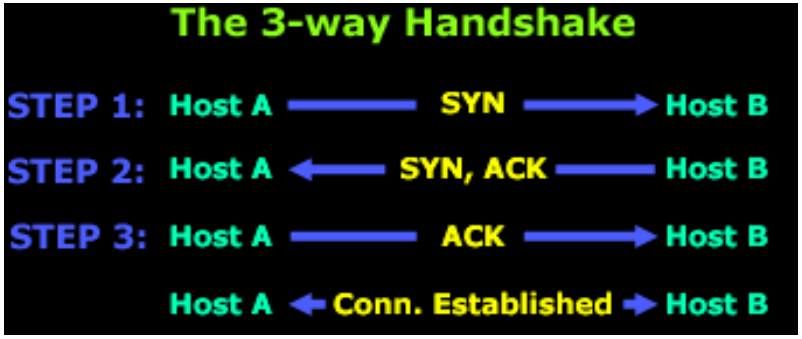
- SYN (SYNchronise) is used for starting a connection
- ACK (ACKnowledge) is used to confirm packets are received
- FIN (FINish) is used for ending a connection
- RST (ReSeT) is used to denote no service on the port (closed)
- PSH (PuSH) and URG (URGent) are used to ensure priority over other packets for processing when a packet is received.
- The CWR and ECN bits may be used together by a client and server that can use RFC 3168 Explicit Congestion Notification.

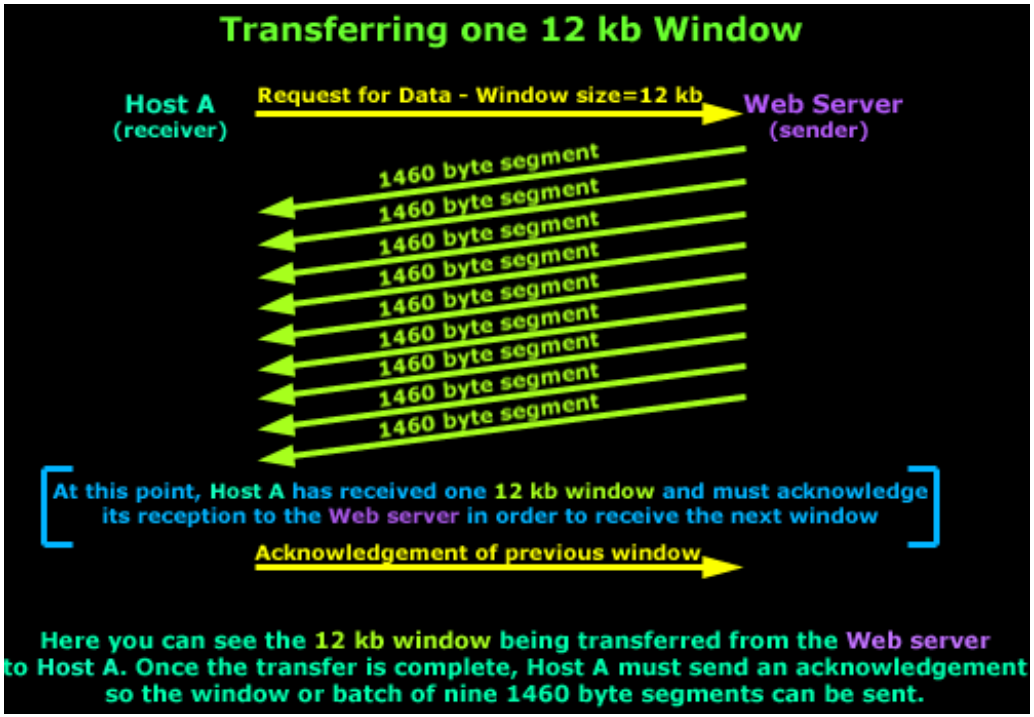
Background: TCP

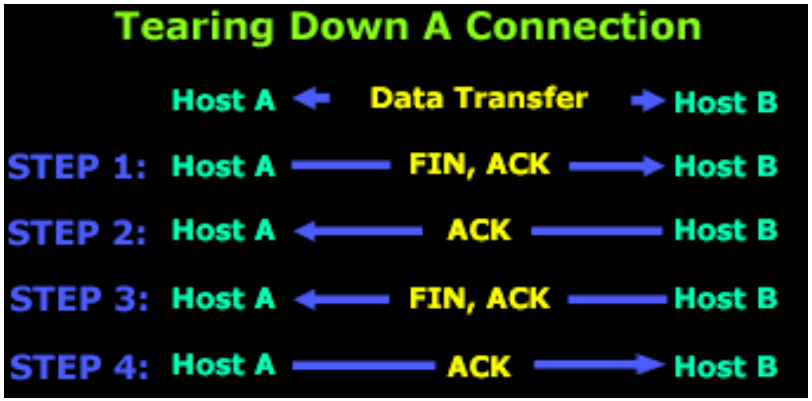
The initial connection with a TCP service is established with the "TCP 3 way handshake".

- First a SYN is sent to a port that has a service bound to it (an open port). Typical examples are HTTP (port 80), SMTP (port 25), or SSH (port 22).
- The server side will see the SYN and respond with SYN + ACK, with the client answering with an ACK. This completes the set up and the data of the service protocol can now be communicated.
- After this, an ACK is sent after receiving each "window size" group of packets (details are beyond our scope, we have nice graphic).
- When a host sends a FIN to close a connection, it may continue to receive data until the remote host has also closed the connection, although this only occurs under certain circumstances.

Background: TCP







http://www.firewall.cx/networking-topics/protocols/tcp/136-tcp-flag-options.html
http://www.firewall.cx/networking-topics/protocols/tcp/137-tcp-window-size-checksum.html

Testing Rules

Verifying that your firewall rules are performing as intended should be a regular procedure.

Once you have identified any unfiltered ports or other issues, you should review your firewall rules to ensure that access to all services is controlled, and that filters and rules are performing as intended.

After changes are implemented, run the audit scan again to ensure that your changes achieve the desired effect.

See this article about checking the status of your firewall rules
https://www.cyberciti.biz/tips/linux-iptables-examples.html

Note that there are a few tips that are RHEL / Fedora / CentOS Linux specific, but all of the IPTables rules themselves are valid for any distro.

References

- Red Hat Linux Security Guide
  - Chapter 2. Attackers and Vulnerabilities
  - Appendix A. Common Exploits and Attacks
  - 5.1. Securing Services With TCP Wrappers and xinetd
  - 5.2. Securing Portmap
  - 5.4. Securing NFS
  - 5.5. Securing Apache HTTP Server
  - 7.1.1. IPTables Overview
- Red Hat Linux Reference Guide
  - Chapter 15. TCP Wrappers and xinetd
  - Chapter 16. iptables


Q
&
A

IDS: Intrusion Detection System

- Gathers and analyzes information from within a computer or a network to identify the possible violations of security policy, including unauthorized access and misuse
  - The ultimate aim is to catch perpetrators before they do real damage to your resources
  - Based on your security policy and administrator intuition and experience
- Uses:
  - Rogue system detection.
  - Reconnaissance identification.
  - Attack pattern identification.

Types of IDS

- Network-Based Intrusion Detection
  - Listening for patterns indicative of an intrusion
- Log File Monitoring
  - Programs that parse log files after an event has already occurred, eg. Failed log in attempts
- File Integrity Checking
  - Check files been modified indicating an intruder has already been there

Types of IDS

- Host-based: Logfile Tracker filters a combination of log files rather than network traffic.
  - This goes beyond the syslog ability to separate single events by their facility and severity.
  - Can be fast, but not real-time

- Host-based: File Tracker creates and stores a message digest for sensitive files. It periodically recalculates each message digest and compares it to the original, and will alert the administrator if they do not match.

Network-based IDS

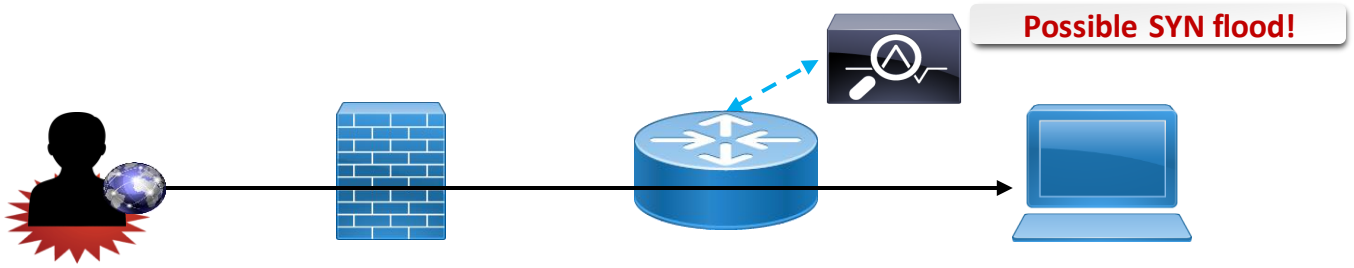
- The design philosophy of a **network-based IDS** is very much like an anti-virus system.
- It scans traffic and uses a database of known network attack signatures (traffic patterns) to assign a severity level to groups of suspicious packets and records these in a log file with extended information.
- If severity levels are high enough, a warning email or pager call is placed to security team members so they can investigate further.

Network IDS Techniques

Monitoring System	Description
Signature-based	• Uses predefined set of rules to identify unacceptable events. • Events have specific and known characteristics.
Anomaly-based	• Uses a definition of expected patterns to events. • Identifies events that don't follow these patterns. • Requires a preconfigured baseline of acceptable events.
Behavior-based	• Identifies how an entity acts and reviews future behavior against this. • Detects if future behavior deviates from the norm. • Records patterns in reaction to entity being monitored.
Heuristic	• Identifies how an entity acts in a specific environment. • May conclude that an entity is a threat to the environment.

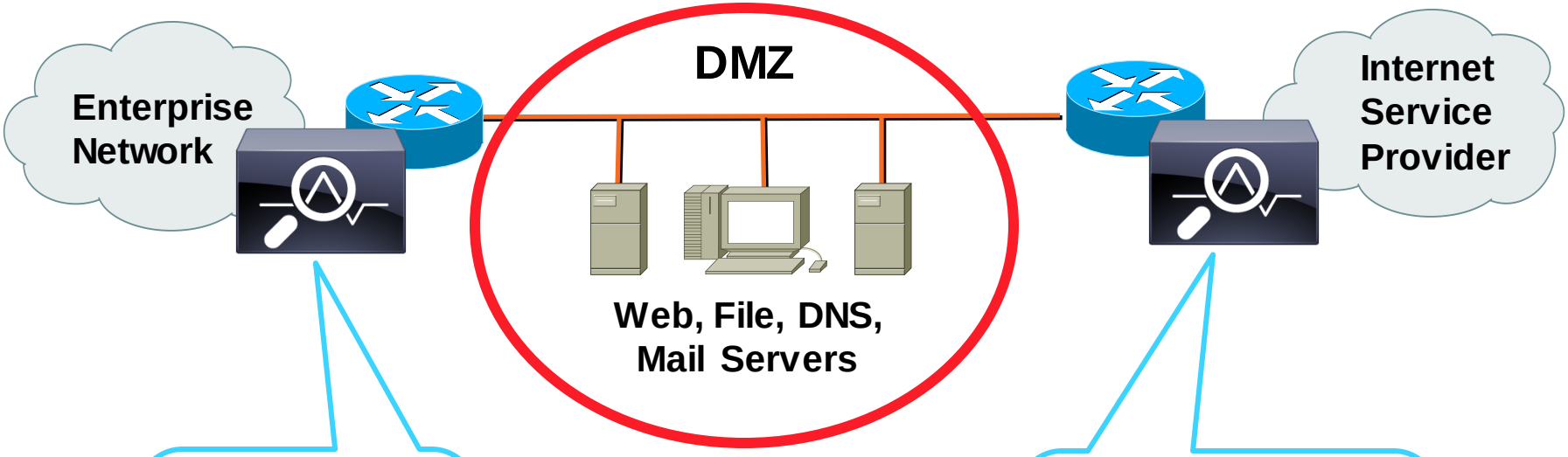
IDS vs. Firewall

- Firewall directly allows or denies access to services based on network and transport layer header information in **individual packets** such as address, service, and direction of the packet.
- IDS analyses **sets of packets (**traffic patterns) to determine misuse (malicious or abusive activity inside the network) or intrusion (from the outside).
- When something suspicious has been identified it signals an alarm



ACLs on router and individual servers

DMZ: demilitarized zone (uncontrolled access)

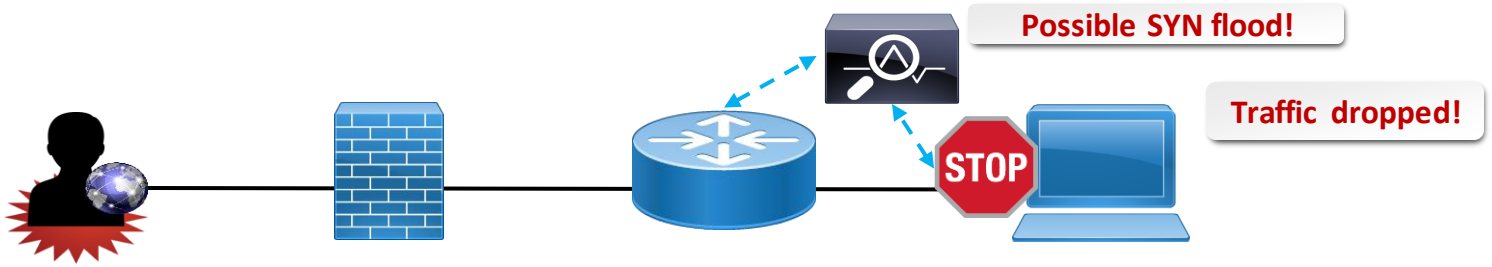


IDS Inside: Catches what was missed by the Firewall

IDS Outside: Catches everything the world throws at you – Good for research

Intrusion Prevention System (IPS)

- A system that performs the functions of an IDS but that can also take action (like a firewall) to block threats.
- Configure the threats that should be handled automatically.
  - Passive response still implemented for other incidents.
- Useful, but with some pitfalls:
  - False positives may lead to the blocking of legitimate behavior.
  - False negatives may lull you into a false sense of security.
- A well-managed and finely tuned IPS is a powerful defense option.



Unified Threat Management

A system that centralizes various security techniques into a single appliance.

- Usually include a single console from which to administrate defenses.
- Created in response to cost and complexity issues of discrete systems.
- Can streamline security process and make management of defenses easier.
- Downsides:
  - Creates a single point of failure in the network.
  - Can struggle with network latency issues.


Q
&
A

snort

Snort http://www.snort.org/ has a constantly updated database of attacks; users can create signatures based on new network attacks and submit them to http://www.snort.org/community so that all Snort users can benefit.

This community ethic of sharing has developed Snort into one of the most up-to-date and robust network-based IDSes available.

Note:

If you want to experiment with Snort, use the TinyNet version rather than the latest website version, because the attack database has been cut back to be more understandable. *It is easy to get lost in the thousands of attack signatures in a current Snort database*

IDS Types: Logfile Tracker

- Simple Log Watcher (SWATCH)
  http://sourceforge.net/projects/swatch/
- Simple Event Correlator
  http://simple-evcorr.sourceforge.net

- Both written in Perl
- SWATCH was designed to log any event that the user wants to add into the configuration file, and it has been adopted widely as a host-based IDS.
- Simple Event Correlator has more advanced features for message aggregation and multiline matching

IDS Types: File Tracker

- Host-based IDS that verifies the data integrity of important files and executables.
  - It checks a database of sensitive files, creates and stores a **message digest** for each one
  - The IDS periodically recalculates each message digest and compares it to the original
  - If the newly calculated message digest and the original stored one do not match the IDS will alert the administrator by email or cellular pager.

(this process should look familiar)

Tripwire http://www.tripwire.org/

IDS Types: all-in-one

OSSEC http://www.ossec.net/

- performs log analysis, file integrity checking, policy monitoring, real-time alerting and active response.
- has a log analysis engine that is able to correlate and analyze logs from multiple devices and formats

Distributed design:

- a central manager stores the file integrity checking databases, the logs, event rules, system auditing rules and major configuration options. The manager receives information from syslog, databases and from agents.
- an agent is a small program installed on the systems you monitor. It collects information on real time and forwards to the manager for analysis and correlation.