Authorization

Shuai Wang

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

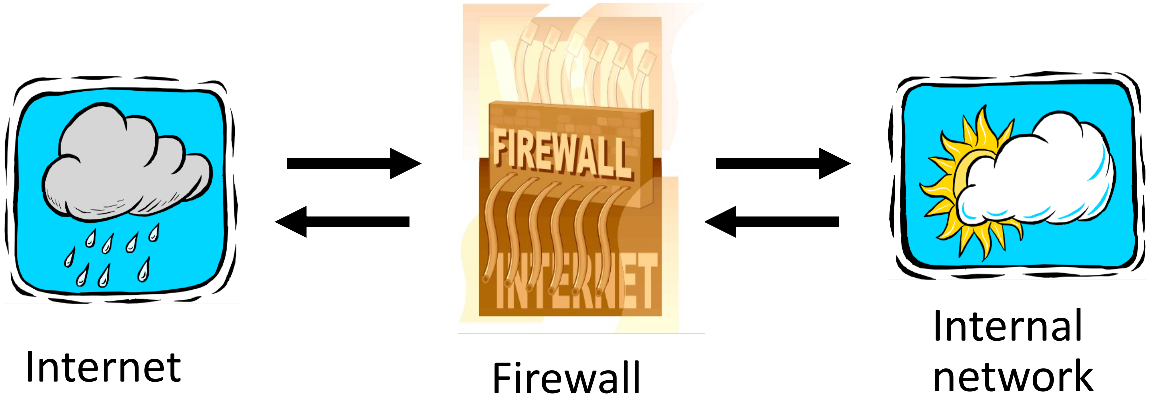
Some slides are from Mark Stamp.

Authentication vs Authorization

- Authentication — Are you who you say you are?
  - Restrictions on who (or what) can access system
- **Authorization** — Are you allowed to do that?
  - Restrictions on actions of authenticated users
- Authorization is a form of **access control**
- Two major views of authorization…
  - Firewall → networking (our topic today)
  - Access Control Lists (ACLs)/Capabilities (C-lists) → OS kernel

Firewalls

- All network flows were possible
  - Into or out of our network
  - To/from individual hosts and their processes



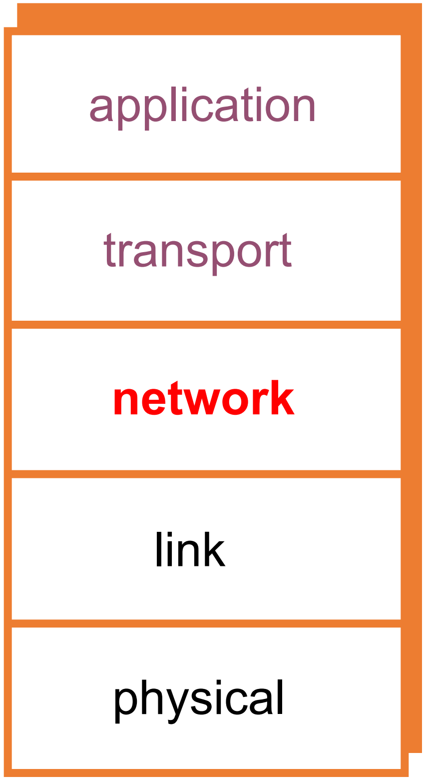
Firewall decides what to let in to internal network and/or what to let out → access control

Firewall

- No standard firewall terminology
- Types of firewalls
  - **Packet filter** — works at network layer
  - **Stateful packet filter** — transport layer
  - **Application proxy** — application layer
- Lots of other terms often used
  - E.g., “deep packet inspection”
  - Some marketing strategies…

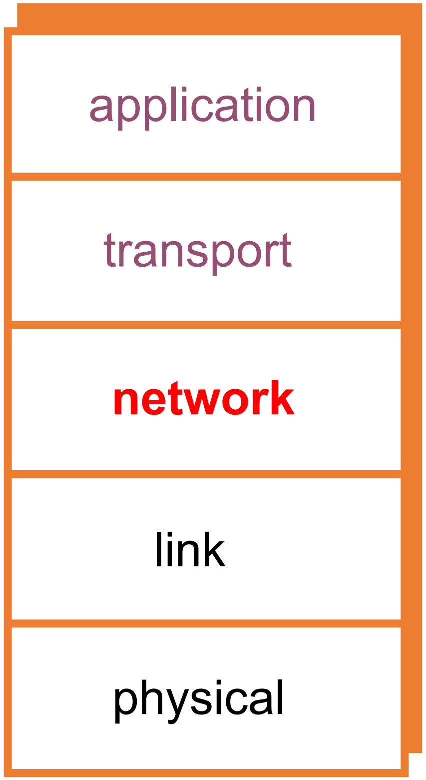
Packet Filter

- Operates at network layer
  - What do we have on network layer?
- Can filters based on…
  - Source IP address
  - Destination IP address
  - Source Port
  - Destination Port
- But also "cheat" to some extent:
  - Flag bits (SYN, ACK, etc.)



Packet Filter

- Advantages?
  - Speed
- Disadvantages?
  - No concept of state
  - Cannot see TCP connections
  - Blind to application data



Packet Filter

- Configured via Access Control Lists (ACLs)
  - Note that this is a bit different from the ACLs is OS

Action	Source IP	Dest IP	Source Port	Dest Port	Protocol	Flag Bits
Allow	Inside	Outside	Any	80	HTTP	Any
Allow	Outside	Inside	80	> 1023	HTTP	Ack
Deny	All	All	All	All	All	All

- **Q**: Intention?
- **A**: Restrict traffic to Web browsing

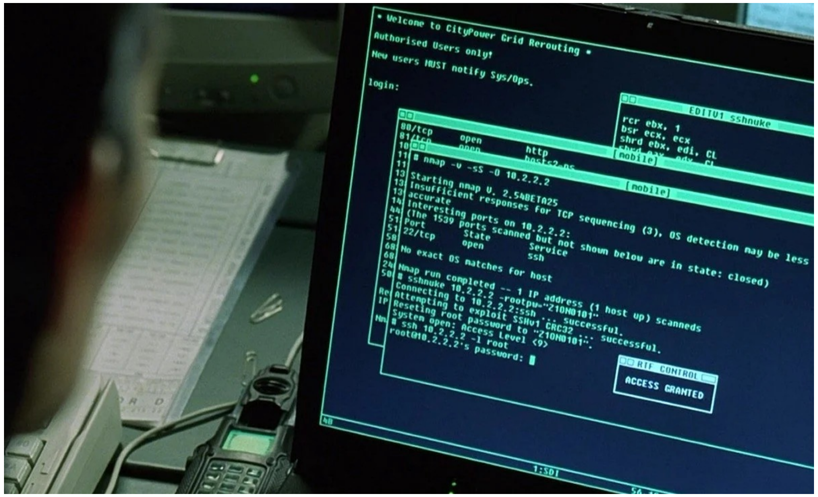
Vulnerable! But let's first introduce Port Scan Attacks…

Port Scanning

- Attacker scans for open ports thru firewall
  - Port scanning often *the prerequisite* in network attack
  - Knock on “doors” (ports) to see which are open
- Attackers wants to determine open ports
  - 65k TCP ports and 65k UPD ports
  - Well-known ports correspond to services
  - Open port is a doorway into machine

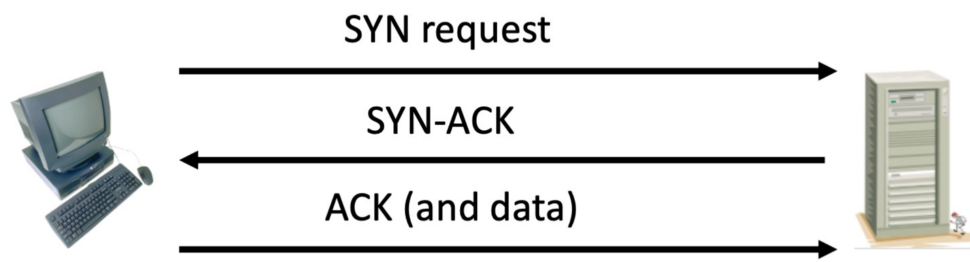
Nmap

- Nmap --- most popular port scan tool
  - Developed by Fydor
  - Free at www.insecure.org
  - Unix, Linux and Windows versions
  - Command line and GUI
  - Appeared in *The Matrix Reloaded*



TCP 3-Way Handshake

- Recall the 3-way handshake…



- **SYN** — synchronization requested
- **SYN-ACK** — acknowledge SYN request
- **ACK** — acknowledge SYN-ACK (send data)
- Then TCP “connection” established

TCP Connect Scan

- *"Polite scan"*
- Complete the TCP 3-way handshake
  - Nmap sends SYN, wait for SYN-ACK
  - If port is open, Nmap sends ACK, then FIN
  - If closed, no reply
- Pros?
  - Should not cause problem for target
- Cons?
  - attacker's IP address in logs, etc.

TCP SYN Scans

- Nmap sends SYN
  - Gets SYN-ACK
  - In any case, Nmap sends RESET
  - I.e., only 2/3rds of 3-way handshake completed
- Pros?
  - may not be logged by host
  - Faster, fewer packets

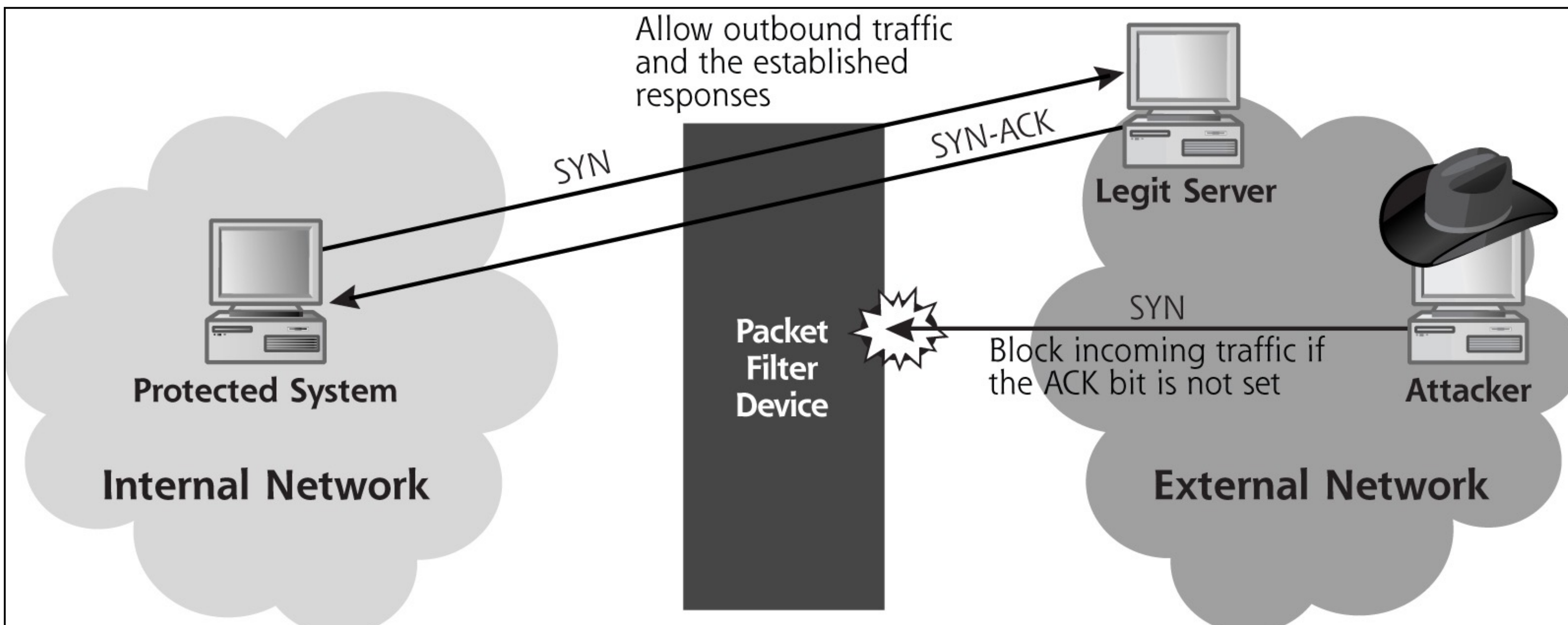
But we have firewall…

- Simpleminded packet filter might…
  - Allow outbound, established connections
  - Block incoming if ACK bit not set

Action	Source IP	Dest IP	Source Port	Dest Port	Protocol	Flag Bits
Allow	Inside	Outside	Any	80	HTTP	Any
Allow	Outside	Inside	80	> 1023	HTTP	ACK
Deny	All	All	All	All	All	All

But we have firewall…

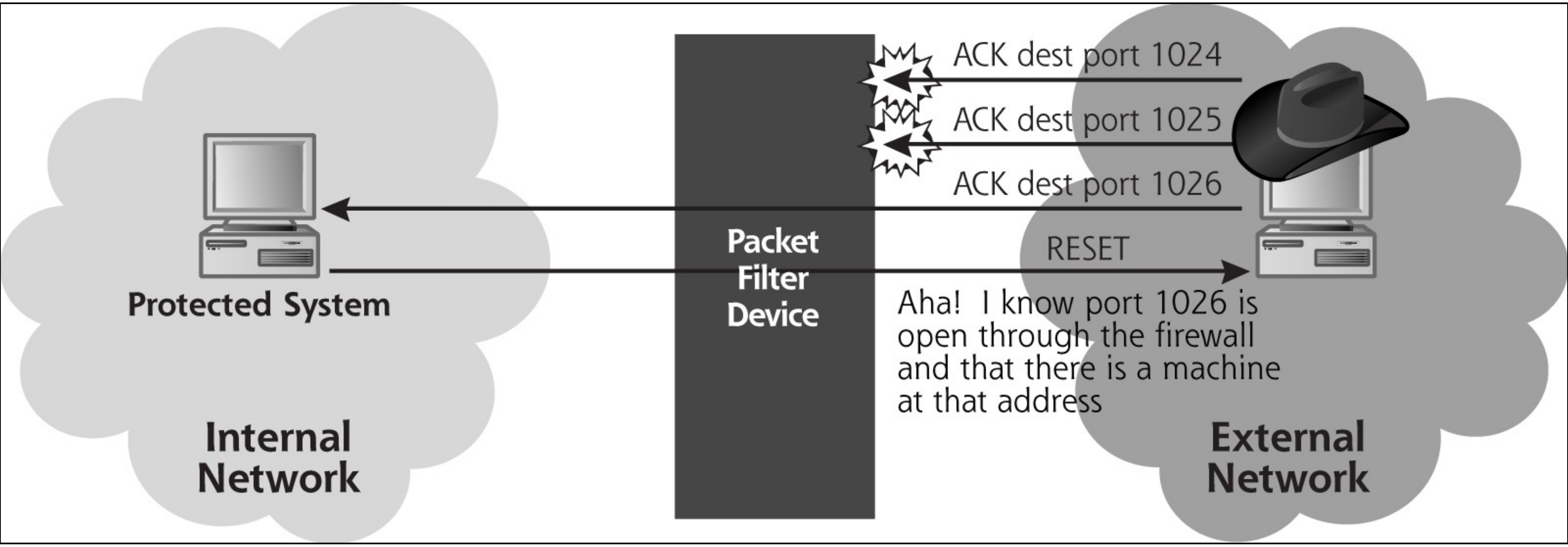
- Simpleminded packet filter might…
  - Allow outbound, established connections
  - Block incoming if ACK bit not set



But we can do TCP ACK Scan…

- Packet filter assumes
  - ACK bit set $\Rightarrow$ established connection
- How can the Attacker take advantage of this?
- Send packets with ACK bit set!
  - These pass thru open ports
  - Allows for simple port scan of firewall

But we can do TCP ACK Scan…



- No response/unreachable: filtered
- RESET if port is not filtered

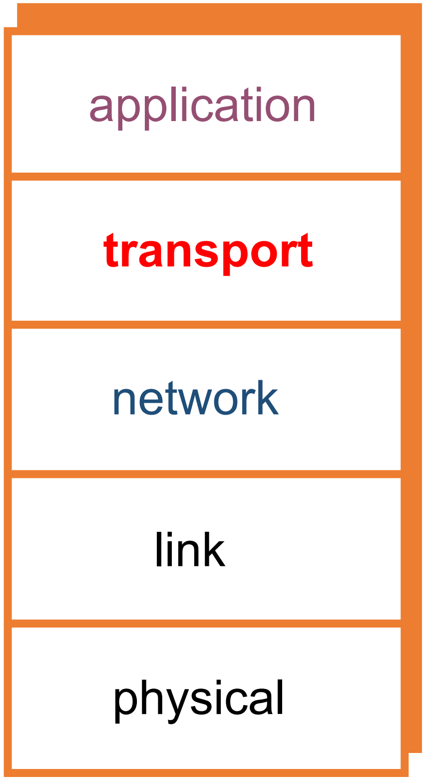
So how to prevent this? Our firewall does not have the "state" of TCP connection in mind.

A **stateful packet filter** can prevent this

Since scans not part of established connections

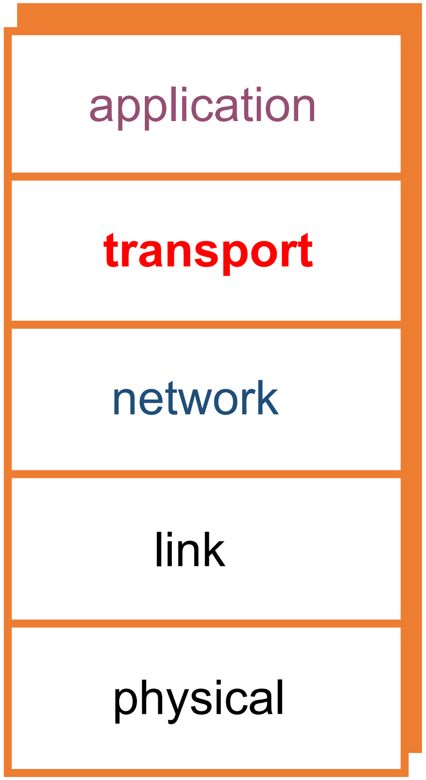
Stateful Packet Filter

- Adds **state** to packet filter
- Operates at transport layer
- ***Remembers*** TCP connections, flag bits, etc.



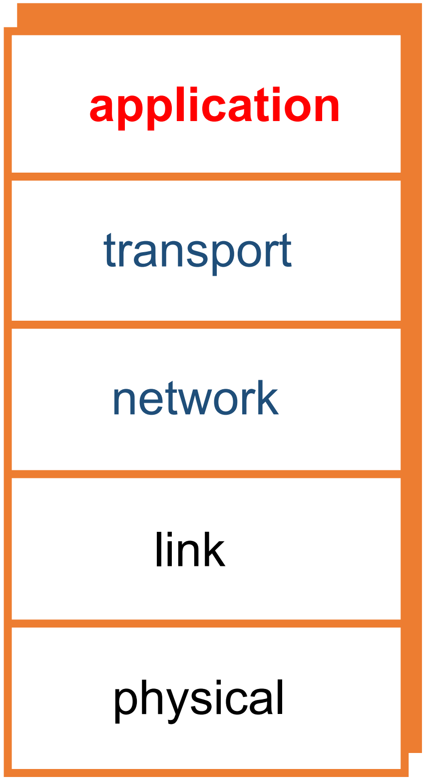
Stateful Packet Filter

- Advantages?
  - Can do everything a packet filter can do plus...
  - Keep track of ongoing connections (e.g., prevents TCP ACK scan)
- Disadvantages?
  - Cannot see application data
  - Slower than packet filtering



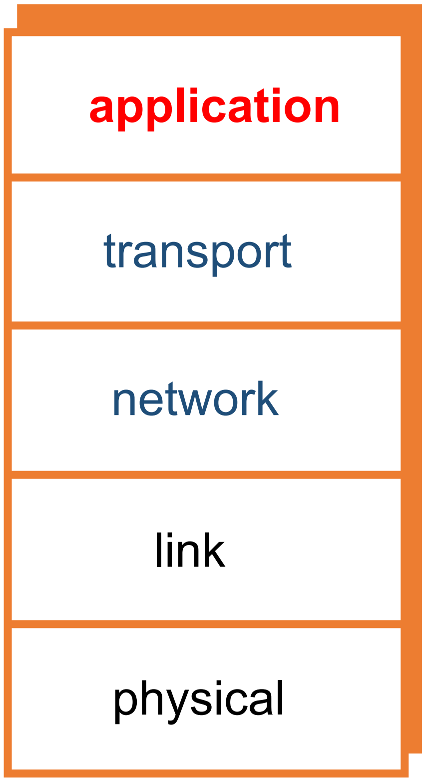
Application Proxy

- A **proxy** is something that acts on your behalf
- Application proxy looks at incoming application data
- Verifies that data is safe before letting it in
- Proxy firewall; application firewall; gateway firewall.



Application Proxy

- Advantages?
  - Complete view of connections and applications data
  - Filter bad data at application layer (viruses, Word macros)
- Disadvantages?
  - Speed



Application Proxy

- Creates a ***new packet*** before sending it thru to internal network
- Proxy has complete view of connection
- A “by-product”: can prevent some scans stateful packet filter cannot → Firewalk style scanning

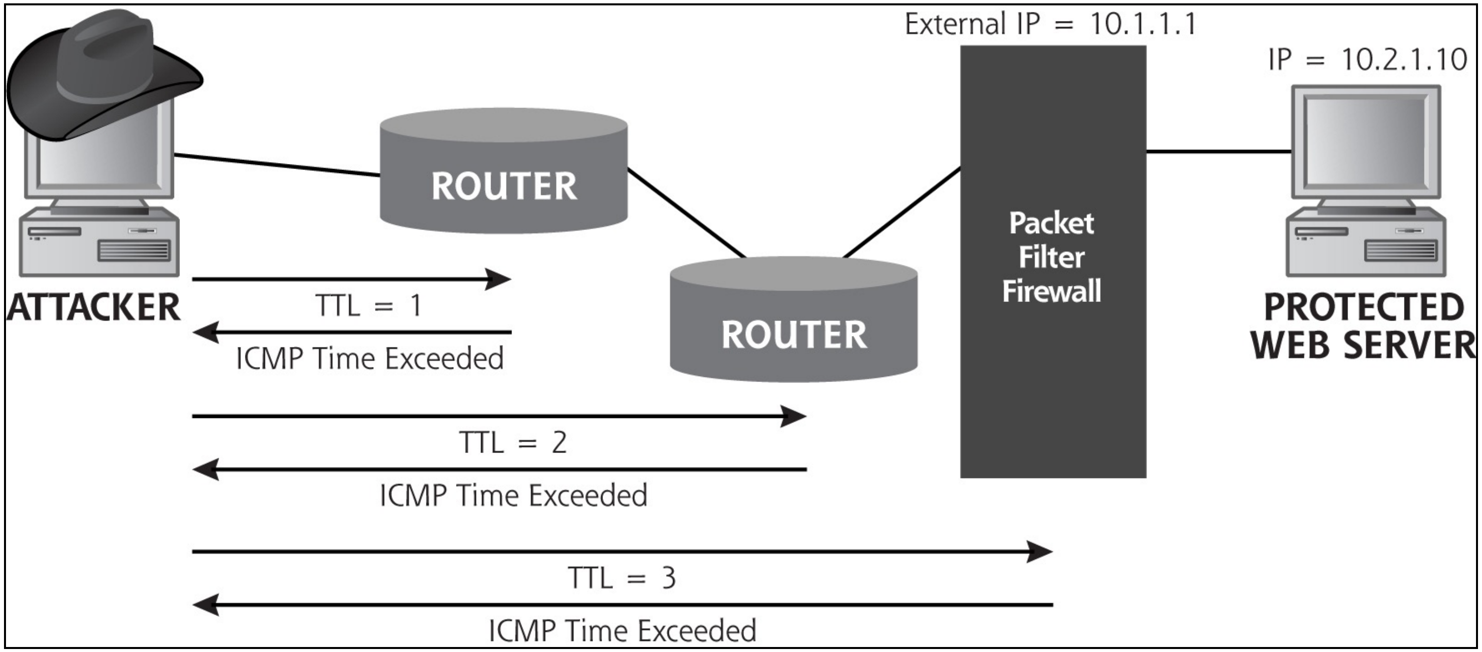
But let’s first introduce Firewalk…

Firewalk

- Determines what gets thru firewall
  - Assuming a packet filter firewall
- Firewalk has 2 phases
  - Network discovery
  - Actual scanning

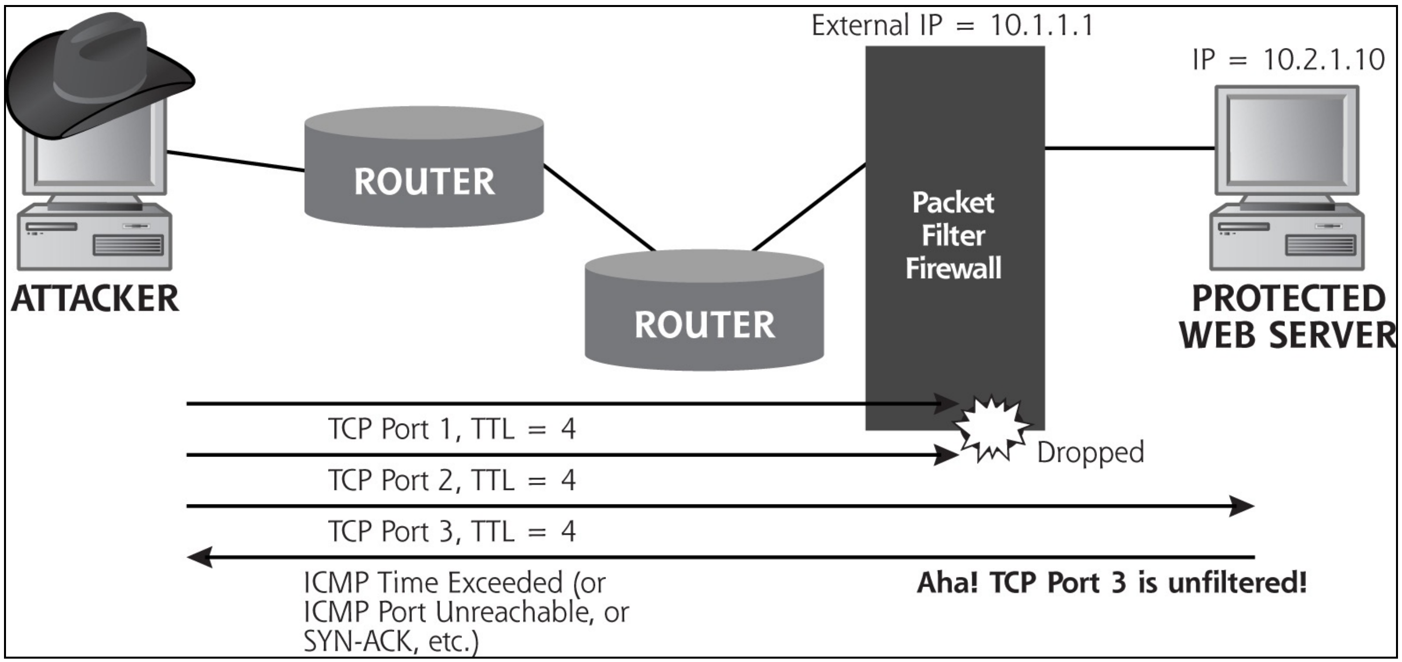
Firewalk

- Network discovery phase
  - Use TTL to find hops to firewall
    - TTL: Time to Live



Firewalk

- Scanning phase
  - Packet sent to host behind firewall



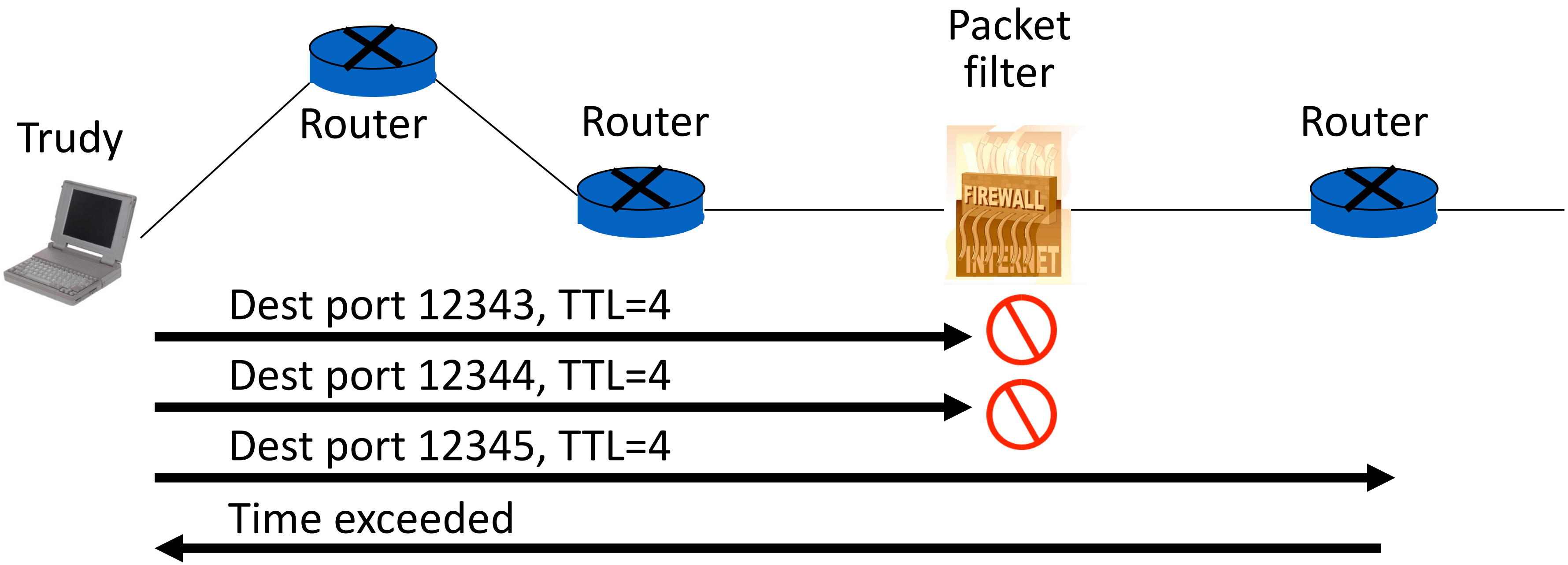
Firewalk

- Nmap vs Firewalk
  - Nmap does port scan of hosts
  - What happens if you Nmap a firewall?
    - Tells you ports firewall is **listening** on
  - What happens if you Firewalk a firewall?
    - Tells you the internal network topological structure

Firewalk

- TTL field crucial to Firewalk
- Packet filter and stateful packet filters both decrement TTL field
  - So Firewalk can work against these
- Application proxy firewall?
  - Proxy does not forward packet
  - Instead, creates a new packet… so what?

Firewalk and Proxy Firewall



- This will **not** work thru an application proxy (why?)
- The proxy creates a new packet, destroys old TTL
  - Usually the TTL is set [32, 64], and max value is 256

Intrusion Detection Systems

Conceptually similar to what we have discussed on “malware detection”

Intrusion Detection

- In spite of intrusion prevention, bad guys will sometime get in
- Intrusion detection systems (**IDS**)
  - Detect attacks in progress (or soon after)
  - Look for unusual or suspicious activity
- IDS evolved from log file analysis
- How to respond when intrusion detected?
  - We don't deal with this topic here…

Intrusion Detection Systems

- Who is likely intruder?
  - May be outsider who got thru firewall
  - May be evil insider
- What do intruders do?
  - Launch well-known attacks
  - Launch variations on well-known attacks
  - Launch new/little-known attacks
  - “Borrow” system resources
  - Use compromised system to attack others. etc.

IDS

- Intrusion detection **approaches**
  - Signature-based IDS (templates **pre-defined**!)
  - Anomaly-based IDS (metrics **dynamically** adjusted)
- Intrusion detection **architectures**
  - Host-based IDS
  - Network-based IDS

Host-Based IDS

- Monitor activities on hosts for
  - Known attacks
  - Suspicious behavior
- Designed to detect attacks such as
  - Buffer overflow
  - Escalation of privilege, …
- Little or no view of network activities

Network-Based IDS

- Monitor activity on the network for…
  - Known attacks
  - Suspicious network activity
- Designed to detect attacks such as
  - Denial of service
  - Network probes
  - Malformed packets, etc.
- Some overlap with firewall (often shipped together)
- Little or no view of host-base attacks
- Can have both host and network IDS

Signature Detection Example

- Failed login attempts may indicate password cracking attack
- IDS could use the rule “N failed login attempts in M seconds” as **signature**
- If N or more failed login attempts in M seconds, IDS warns of attack

Signature Detection

- Suppose IDS warns whenever N or more failed logins in M seconds
  - Set N and M so false alarms not common (how?)
  - Can do this based on "normal" behavior
- But, if an attacker knows the signature, she can try $N - 1$ logins every M seconds…
- Then signature detection slows down the attacker, but might not stop her

Signature Detection

- Many techniques used to make signature detection more robust
- Goal is to detect "almost" signatures
- For example, if "about" N login attempts in "about" M seconds
  - Warn of possible password cracking attempt
  - What are reasonable values for "about"?
  - Can use statistical analysis, heuristics, etc.
  - Must not increase false alarm rate too much

Signature Detection

- Advantages of signature detection
  - Simple
  - Detect known attacks
  - Know which attack at time of detection
  - Efficient (if reasonable number of signatures)
- Disadvantages of signature detection
  - Signature files must be kept up to date
  - Number of signatures may become large
  - Can only detect known attacks
  - Variation on known attack may not be detected

Anomaly Detection

- Anomaly detection systems look for unusual or abnormal behavior
- There are (at least) two challenges
  - What is normal for this system?
  - How “far” from normal is abnormal?
- Some statistics…
  - **mean** defines normal
  - **variance** gives distance from normal to abnormal

How to Measure Abnormal?

- Abnormal is relative to some “normal”
  - Abnormal indicates possible attack
- Statistical discrimination techniques include
  - Bayesian statistics
  - Linear discriminant analysis (LDA)
  - Quadratic discriminant analysis (QDA)
  - hidden Markov models (HMMs), etc.
- Fancy modeling techniques also used
  - Artificial intelligence
  - Artificial immune system principles
  - Many, many, many others

Anomaly Detection (1)

- Over time, Alice has accessed file F_n at rate H_n

H_0	H_1	H_2	H_3
.10	.40	.40	.10

- Recently, "Alice" has accessed F_n at rate A_n

A_0	A_1	A_2	A_3
.10	.40	.30	.20

- Is this normal use for Alice?
- We compute $S = (H_0-A_0)^2+(H_1-A_1)^2+\ldots+(H_3-A_3)^2 = .02$
  - We consider $S < 0.1$ to be normal, so this is normal
- How to account for use that varies over time?

Anomaly Detection (1)

- To allow "normal" to adapt to new use, we update averages: $H_n = 0.2A_n + 0.8H_n$
- In this example, H_n are updated... $H_2 = .2*.3+.8*.4 = .38$ and $H_3 = .2*.2+.8*.1 = .12$
- And we now have

H_0	H_1	H_2	H_3
.10	.40	.38	.12

It's frequently referred as "Moving Target Defense (MTD)", a fancy term.

Anomaly Detection (1)

- The updated long term average is

H_0	H_1	H_2	H_3
.10	.40	.38	.12

- Suppose new observed rates…

A_0	A_1	A_2	A_3
.10	.30	.30	.30

- Is this normal use?
- Compute $S = (H_0 - A_0)^2 + \ldots + (H_3 - A_3)^2 = .0488$
  - Since $S = .0488 < 0.1$ we consider this normal
- And we again update the long term averages:

$$H_n = 0.2A_n + 0.8H_n$$

Anomaly Detection (2)

- The starting averages were:

H_0	H_1	H_2	H_3
.10	.40	.40	.10

- After 2 iterations, averages are:

H_0	H_1	H_2	H_3
.10	.38	.364	.156

- Statistics slowly evolve to match behavior
- This reduces false alarms
- But attackers can, well, move slow and gradually convince the IDS
  - At the end of the day, IDS (MTD) is not silver bullet.

Anomaly Detection (2)

- To make this approach more robust, must incorporate the variance
- Can also combine N stats S_i as, say,

  $$T = (S_{N-k} + \ldots + S_{N-1} + S_N) / N$$

  to obtain a more complete view of "normal"

Anomaly Detection Issues

- Systems constantly evolve and so must IDS
  - But evolving IDS makes it possible for attacker to (slowly) convince IDS that an attack is normal
  - Attacker may win simply by “going slow”
- What does “abnormal” really mean?
  - Indicates there may be an attack
  - Might not be any specific info about “attack”
  - How to respond to such vague information?
  - In contrast, signature (template) detection is very specific

Anomaly Detection

- Advantages?
  - Chance of detecting unknown attacks
- Disadvantages?
  - Reliability is unclear → arguable
  - May be subject to attack
  - Anomaly detection indicates "something unusual", but lacks specific info on possible attack
- Real-world approach combines anomaly & signature IDS