CS 489 / 698: Software and Systems Security

Module 7: Cloud Security

network and web security

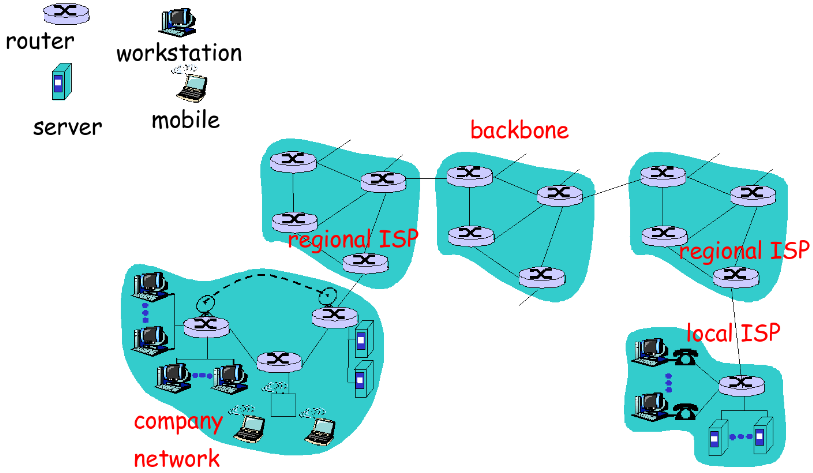
Meng Xu *(University of Waterloo)*

Spring 2023

Outline

1. Core concepts in networking
2. Extracting intelligence from networked systems
3. Denial-of-service attacks
4. Firewalls as security control
5. Intrusion detection system and honeypots

Architecture of the Internet



Slide adapted from “Computer Networking” by Kurose & Ross

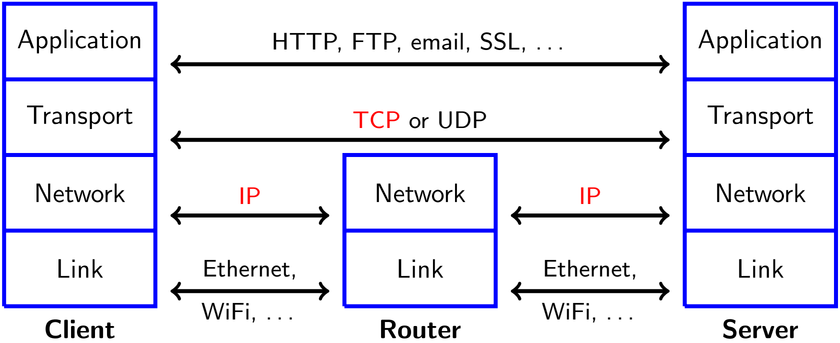
Characteristics of the Internet

- No single entity that controls the Internet
  - Traffic from a source to a destination likely flows through nodes controlled by different entities
  - Src/dst nodes cannot control through which nodes traffic flows
    - Worse, all traffic is split up into individual packets, and each packet could be routed along a different path

- Different types of nodes
  - Server, laptop, router, UNIX, Windows, . . .
- Different types of communication links
  - Wireless vs. wired

- TCP/IP suite of protocols
  - Packet format, routing of packets, dealing with packet loss,. . .

TCP/IP protocol suite



The protocol is designed assuming all nodes are faithful and honest.

IP packet format

```
    0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |  Ver. |  IHL  |Type Of Service|         Total Length          |
| I +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| P |        Identification         |Flags|    Fragment Offset      |
|   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| H | Time to Live  |   Protocol    |       Header Checksum         |
| E +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| A |                      Source IP Address                        |
| D +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| E |                   Destination IP Address                      |
| R +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |                 Optional Fields (variable)                    |
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |                                                               :
| P :                  Packet Payload (variable)                    :
|   :                                                               |
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TCP packet format

```
    0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |          Source Port          |       Destination Port        |
| T +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| C |                        Sequence Number                        |
| P +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |                    Acknowledgment Number                      |
|   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| H |  Data |       |C|E|U|A|P|R|S|F|                               |
| E | Offset| Rsrvd |W|C|R|C|S|S|Y|I|            Window             |
| A |       |       |R|E|G|K|H|T|N|N|                               |
| D +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| E |           Checksum            |         Urgent Pointer        |
| R +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |                [Options] (variable in size)                   |
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |                                                               :
| P :                             Data                              :
|   :                                                               |
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TCP/IP combined packet format

```
    0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |  Ver. |  IHL  |Type Of Service|         Total Length          |
| I +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| P |        Identification         |Flags|    Fragment Offset      |
|   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| H | Time to Live  |   Protocol    |        Header Checksum        |
| E +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| A |                     Source IP Address                         |
| D +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| E |                  Destination IP Address                       |
| R +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |                Optional Fields (variable)                     |
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |          Source Port          |        Destination Port       |
| T +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| C |                       Sequence Number                         |
| P +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |                    Acknowledgment Number                      |
|   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| H |  Data |       |C|E|U|A|P|R|S|F|                               |
| E | Offset| Rsrvd |W|C|R|C|S|S|Y|I|            Window             |
| A |       |       |R|E|G|K|H|T|N|N|                               |
```

Threats in networks

- Information leak
- Attacks on confidentiality
- Impersonation and spoofing
- Attacks on integrity
- Protocols failures
- Website vulnerabilities
- Denial of service
- Botnets

Outline

1. Core concepts in networking
2. Extracting intelligence from networked systems
3. Denial-of-service attacks
4. Firewalls as security control
5. Intrusion detection system and honeypots

Port scan

- To distinguish between applications running on the same server, each application runs on a “port”
  - e.g., 22, 80, 443, 3306 (MySQL), 5432 (PostgreSQL)

- Attacker sends queries to ports on target machine and tries to identify whether and what kind of application is running on a port
- Identification based on loose-lipped applications or how exactly application implements a protocol

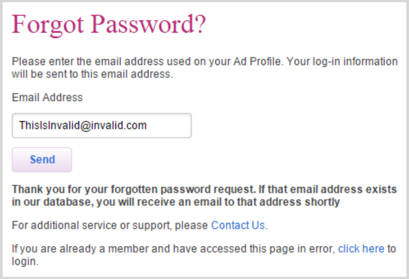
Port scan (cont.)

- Loose-lipped systems reveal (non-confidential) information that could facilitate an attack
  - Login application can reveal information about OS or whether a username is valid
  - Web servers typically return version information
- Nmap tool can identify many applications
  - Useful not only to attackers, but also to system administrators
- Goal of attacker is to find application with remotely exploitable flaw
  - e.g., Apache web server prior to version 1.3.26 is known to be vulnerable to a buffer overflow bug
  - Exploits for these flaws can be found on the Internet

An example of loose lips

Ashley Madison's Password Reset Form

Response for invalid email address



Forgot Password?

Please enter the email address used on your Ad Profile. Your log-in information will be sent to this email address.

Email Address

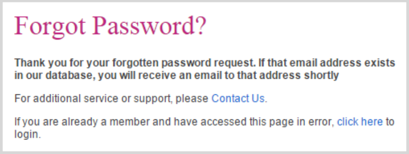
ThisIsInvalid@invalid.com

Thank you for your forgotten password request. If that email address exists in our database, you will receive an email to that address shortly

For additional service or support, please Contact Us.

If you are already a member and have accessed this page in error, click here to login.

Response for valid email address



Forgot Password?

Thank you for your forgotten password request. If that email address exists in our database, you will receive an email to that address shortly

For additional service or support, please Contact Us.

If you are already a member and have accessed this page in error, click here to login.

https://www.troyhunt.com/your-affairs-were-never-discrete-ashley/

Intelligence collection example

- Social Engineering
  - Attacker gathers sensitive information from person
  - Often, attacker pretends to be somebody within the person's organization who has a problem and exploits the person's willingness to help (or vice versa)
    - I forgot my password, I locked myself out, there's a problem with your Paypal account, . . .
- Dumpster diving
- Eavesdropping on oral communication
- Victim's Facebook profile
- Search engines
  - There's lots of information on the Internet that shouldn't be there
  - The right query will find it

Eavesdropping and wiretapping

- Owner of a node can always monitor communication flowing through the node
  - Eavesdropping or passive wiretapping
  - Active wiretapping involves modification or fabrication of communication
- Can also eavesdrop while communication is flowing across a link
  - Degree of vulnerability depends on type of communication medium
- Or when communication is accidentally sent to attacker’s node
- It is prudent to assume that your communication is wiretapped

The same applies when your code runs on a cloud environment!

Communication media (link layer)

- Copper cable
  - Inductance allows a physically close attacker to eavesdrop without making physical contact
  - Cutting cable and splicing in secondary cable is another option
- Optical fiber
  - No inductance, and signal loss by splicing is likely detectable
- Microwave/satellite communication
  - Signal path at receiver tends to be wide, so attacker close to receiver can eavesdrop

All these attacks are feasible in practice, but require physical expenses/effort.

Communication media (link layer, Wi-Fi)

- Can be easily intercepted by anyone with a WiFi-capable device
  - Don't need additional hardware, which would cause suspicion
- Maybe from kilometers away using a directed antenna
- WiFi also raises other security problems
  - Physical barriers (walls) help against random devices being connected to a wired network, but are (nearly) useless in case of wireless network
  - Need authentication mechanism to defend against free riders

Misdelivered information

Local Area Network (LAN)

- Connects all computers within a company or a university
- Technical reasons might cause a packet to be sent to multiple nodes, not only to the intended receiver

- By default, a network card ignores wrongly delivered packets
- An attacker can change this and use a packet sniffer to capture these packets

Impersonation

- Impersonate a person by stealing their password
  - Guessing attack
  - Exploit default passwords that have not been changed
  - Sniff password (or information about it) while it is being transmitted between two nodes
  - Social engineering

- Exploit trust relationships between machines/accounts
  - `rhosts`/`rlogin` allows user A on machine X to specify that user B on machine Y can act as A on X without having to enter password
    - `ssh` has a similar mechanism
    - Attacker breaking into machine Y can exploit this
    - Or attacker might be able to masquerade as machine Y
  - a.k.a., confused deputy

Spoofing

Definition: Object (e.g., node, person, URL, web page, email, WiFi access point, . . .) masquerades as another one.

Common examples include:

- URL spoofing
  - Exploit typos: `www.uwaterlo.ca`
  - Exploit ambiguities: `www.foobar.com` or `www.foo-bar.com`?
  - Exploit similarities: `www.paypa1.com`
- Web page spoofing and URL spoofing are used in phishing attacks
- “Evil Twin” attack for WiFi access points

Spoofing is also used as a building block in session hijacking and man-in-the-middle attacks.

Session hijacking

- TCP protocol sets up state at sender and receiver end nodes and uses this state while exchanging packets
  - e.g., sequence numbers for detecting lost packets
  - Attacker can hijack such a session and masquerade as one of the endpoints
- Web servers sometimes have client keep a little piece of data (“cookie”) to re-identify client for future visits
  - Attacker can sniff or steal cookie and masquerade as client
- Man-in-the-middle attacks are similar; attacker becomes stealth intermediate node, not end node

Traffic analysis

- Sometimes, the mere existence of communication between two parties is sensitive and should be hidden
  - Whistle blower
  - Military environments
  - Two CEOs

- TCP/IP has each packet include unique addresses for the packet's sender and receiver end nodes, which makes traffic analysis easy
- Attacker can learn these addresses by sniffing packets

Integrity attacks

- Attacker can modify packets while they are being transmitted
  - Change payload of packet
  - Change address of sender or receiver end node
  - Replay previously seen packets
  - Delete or create packets

- Line noise, network congestion, or software errors could also cause these problems (i.e., alteration of packets)
  - TCP/IP will likely detect environmental problems, but fail in the presence of an active attacker

Q: Why?

A: Checksum (in both IP and TCP packets)

Other TCP/IP protocol failure

The main issue with TCP/IP is that the protocol assumes that all nodes implement protocols faithfully, which is not always the case:

- TCP includes a mechanism that asks a sender node to slow down if the network is congested
- Some implementations do not check whether a packet is well formatted
  - e.g., the value in the packet's length field could be smaller than the packet's actual length, making buffer overflow possible
- Protocols can be very complex, behavior in rare cases might not be (uniquely) defined

Outline

1. Core concepts in networking
2. Extracting intelligence from networked systems
3. Denial-of-service attacks
4. Firewalls as security control
5. Intrusion detection system and honeypots

Typical ways of DoS attacks

- Isolate a node from the network
  - e.g., cutting wire or jamming wireless signal
- Crash a node
  - e.g., "ping of death", sending malformed ping packets to crash victim's network stack
- Overloading the processing capacity of a node
  - more on this later
- Exploit knowledge of implementation details about a node to make it perform poorly
  - e.g., exploiting algorithmetic complexity by crafting packets such that they are all hashed into the same bucket in a hash table
- Actively collect and drop the packets
  - e.g., black hole attack (AKA packet drop attack)
    - Routing of packets in the Internet is based on a distributed protocol
    - Each router informs other routers of its cost to reach a set of destinations
    - Malicious router announces low cost for victim destination and discards any traffic destined for victim
    - Might also happen due to router misconfiguration

More on flooding

- Ping flood
  - Node receiving a ping packet is expected to generate a reply
- Smurf attack
  - Spoof (source) address of sender end node in ping packet by setting it to victim's address
  - Broadcast ping packet to all nodes in a LAN
- SYN flood
  - TCP initializes state by having the two end nodes exchange three packets (SYN, SYN-ACK, ACK)
  - Server queues SYN from client and removes it when corresponding ACK is received
  - Attacker sends many SYNs, but no ACKs

Reflection & amplification in DoS

An attack style where the victim is flooded with legitimate-looking traffic that originates from unsuspecting network nodes.

- Amplification: A vulnerable network node (e.g., a home Wi-Fi router) runs a service (e.g., SNMP) that responds to queries with much more data than the query itself
- Reflection: The attacker spoofs the source address of the queries to that of the victim so that the vulnerable network nodes send (reflect) responses to the victim

This style of DoS is hard to combat:

- The response traffic is coming from innocent nodes
- It is hard to identify the real source (perhaps bots) of the queries due to spoofing

Distributed denial-of-service (DDoS)

- If there is only a single machine to launch attack,
  - the machine can generate and send limited number of attacking packets within a fixed amount of duration
  - it might be possible to identify the machine and to have routers discard its traffic

- More powerful if there are lots of attacking machines
  - Attacker breaks into machines using Trojan, buffer overflow, . . . and installs malicious software
  - A compromised machine becomes a zombie/bot and waits for attack commands from the attacker
  - A network of bots is called a botnet

Botnets' infrastructure

Distributed and dynamic control infrastructure with redundancy

- P2P system for distributing updates
- "Fast Flux"
  - A single hostname can potentially map to hundreds of addresses of infected machines
  - Machines proxy to malicious websites or to "mothership"
  - Machines are constantly swapped in/out of DNS to make tracking difficult
- Domain Generation Algorithm
  - Infected machine generates a large set (50,000 in the case of Conficker) of domain names that changes every day
  - It contacts a random subset of these names for updates
  - To control the botnet, authorities would have to take control of 50,000 different domain names each day

Sample botnet: Storm

- In September 2007, Storm Worm botnet included hundreds of thousands or even millions of machines
- Bots were used to send out junk emails advertising web links that when clicked attempted to download and install worm, or to host these websites
- Botnet was also rented out for pharmacy and investment spam
- As a self-defense mechanism, it ran DDoS attacks against Internet addresses that scanned for it
- **Detection**: implementation of its P2P protocol created >10 times normal traffic ($\Longrightarrow$ detectable)

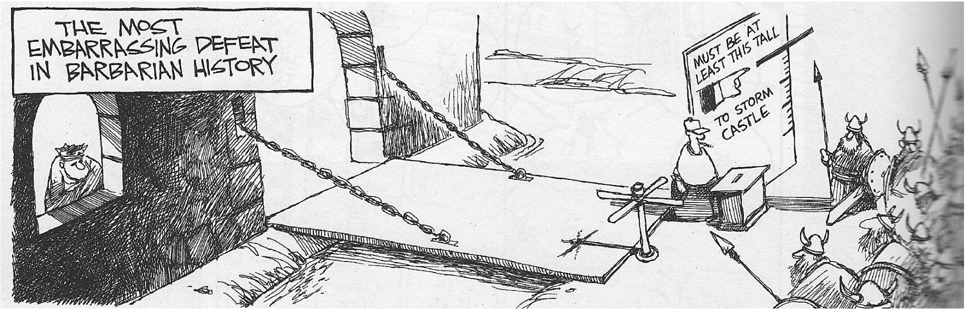
Sample botnet: Mirai

- In fall 2016, Mirai botnet attacked several high-profile targets, including a popular security blog and a large DNS provider
- Attack traffic of so far unseen 1 Tbps or more
- Botnet consisted of 600,000 IoT devices (routers, cameras) infected due to unchanged default passwords
- Distribution based on self-propagating worm
- Each bot flooded targets with UDP, TCP, and HTTP traffic, no amplification or reflection
- Botnet is now believed to be part of a rivalry between Minecraft server operators

Outline

1. Core concepts in networking
2. Extracting intelligence from networked systems
3. Denial-of-service attacks
4. Firewalls as security control
5. Intrusion detection system and honeypots

Firewalls


THE MOST EMBARRASSING DEFEAT IN BARBARIAN HISTORY
MUST BE AT LEAST THIS TALL TO STORM CASTLE

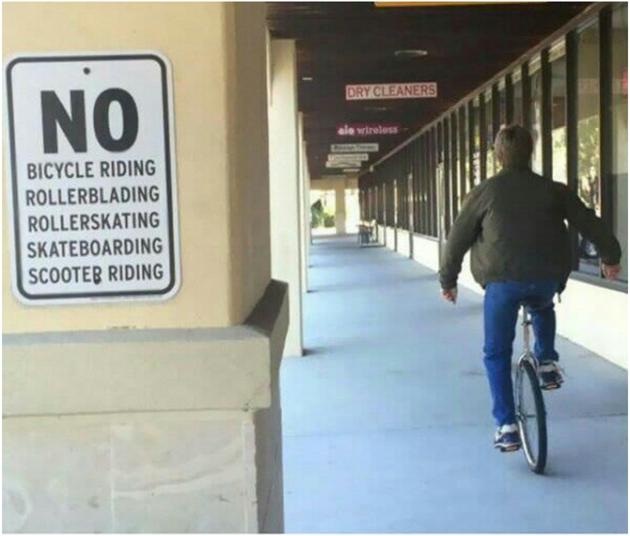
Downloaded from here

Firewalls are the castles of the Internet age

Firewalls

- All traffic into/out of a network (e.g., company network) has to go through a small number of gates (choke points)
- Choke points carefully examine traffic, especially incoming, and might refuse the access
  - Two common strategies
    - “permit everything unless explicitly forbidden”
    - “forbid everything unless explicitly allowed”
- Firewalls typically do not check traffic flowing inside the network, hence, they do not protect attacks originated from nodes within the same network
  - Need multiple layers of defense / defense in depth

Strategy: denylist vs allowlist



Retrieved from Reddit

Different types of firewalls

- Packet filtering gateways / screening routers
- Stateful inspection firewalls
- Application proxies
- Personal firewalls

Packet filtering gateways

- Simplest type of firewalls
- Make decision based on header of a packet
  - Header contains source and destination addresses and port numbers, port numbers can be used to infer type of packet
    - 80 → Web, 22 → SSH
    - e.g., allow Web, but not SSH
- Ignore payload of packet
- Can drop spoofed traffic
  - UW firewall could drop all packets originating from UW whose source address is not of the form `129.97.x.y`
  - And traffic originating from outside of UW whose source address is of the form `129.97.x.y`

Q: Does this eliminate spoofed traffic completely?

A: No. It doesn't prevent spoofed external traffic, or spoofed internal traffic that doesn't cross the firewall.

Defining Firewall Rules

ALLOW/DENY	SrcIP	DstIP	SrcPort	DstPort	TCP Flags

ALLOW/DENY SrcIP DstIP SrcPort DstPort TCP Flags

Stateful inspection firewalls

- More expensive than packet filtering
- Keep state to identify packets that belong together
  - When a client within the company opens a TCP connection to a server outside the company, firewall must recognize response packets from server and let (only) them through
  - Some application-layer protocols (e.g., FTP) require additional (expensive) inspection of packet content to figure out what kind of traffic should be let through
- IP layer can fragment packets, so firewall might have to re-assemble packets for stateful inspection

Application proxy

- Client talks to proxy, proxy talks to server
  - Specific for an application (email, Web, . . .)
  - Not as transparent as packet filtering or stateful inspection
  - Intercepting proxy requires no explicit configuration by client (or knowledge of this filtering by client)
  - All other traffic is blocked
- For users within the company wanting to access a server outside the company (forward proxy) and vice versa (reverse proxy)
- Proxy has full knowledge about communication and can do sophisticated processing
  - Limit types of allowed database queries, filter URLs, log all emails, scan for viruses
- Can also do strong user authentication

Personal firewalls

- Firewall that runs on a (home) user's computer
  - Especially important for computers that are always online
- Typically "forbid everything unless explicitly allowed"
  - Definitely for communication originating from other computers
  - Maybe also for communication originating on the user's computer
    - Why? What's the problem here?

Outline

1. Core concepts in networking
2. Extracting intelligence from networked systems
3. Denial-of-service attacks
4. Firewalls as security control
5. Intrusion detection system and honeypots

Intrusion detection systems (IDSs)

Q: Why do we need intrusion detection when we have firewalls?

A: Firewalls do not protect against inside attackers or insiders making mistakes and can be subverted.

An IDS monitors activities to actively identify malicious or suspicious events

- Receive events from sensors
- Store and analyze them
- Take action if necessary

Broadly, an IDS can be classified as:

- Host-based and network-based IDSs
- Signature-based and heuristic/anomaly-based IDSs

Host-based and network-based IDSs

- Host-based IDSs
  - Run on a host to protect this host
  - Can exploit lots of information (packets, disk, memory, . . .)
  - Miss out on information available to other (attacked) hosts
  - If host gets subverted, IDS likely gets subverted, too
- Network-based IDSs
  - Run on dedicated node to protect all hosts attached to a network
  - Have to rely on information available in monitored packets
  - Typically more difficult to subvert
- Distributed IDSs combine the two of them

Signature-based IDSs

- Each (known) attack has its signature
  - e.g., many SYNs to ports that are not open could be part of a port scan
- Signature-based IDSs try to detect attack signatures
- Fail for new attacks or if attacker manages to modify attack such that its signature changes
  - Polymorphic worms
- Might exploit statistical analysis

Heuristic/anomaly-based IDSs

- Look for behavior that is out of the ordinary
- By modelling good behavior and raising alert when system activity no longer resembles this model
- Or by modelling bad behavior and raising alert when system activity resembles this model
- All activity is classified as good/benign, suspicious, or unknown
- Over time, IDS learns to classify unknown events as good or suspicious
  - Maybe with machine learning

Deploying Honeypots / honeynets

- Set up an (intentionally unprotected) computer or an entire network as a trap for an attacker
- System has no production value, so any activity is suspicious
  - e.g., any received email is considered spam

- Observe attacker to learn about new attacks, to identify and stop attacker, or to divert attacker from attacking real system
- Obviously, attacker should not be able to learn that the attacked system is a honeypot/-net
  - Cat-and-mouse game

Types of honeypots/-nets

- Low interaction
  - Daemon that emulates one or multiple hosts, running different services
  - Easy to install and maintain
  - Limited amount of information gathering possible
  - Easier for the attacker to detect than high interaction honeynets
- High interaction
  - Deploy real hardware and software, use stealth network switches or keyloggers for logging data
  - More complex to deploy
  - Can capture lots of information
  - Can capture unexpected behaviour by attacker

$\langle$ **End** $\rangle$