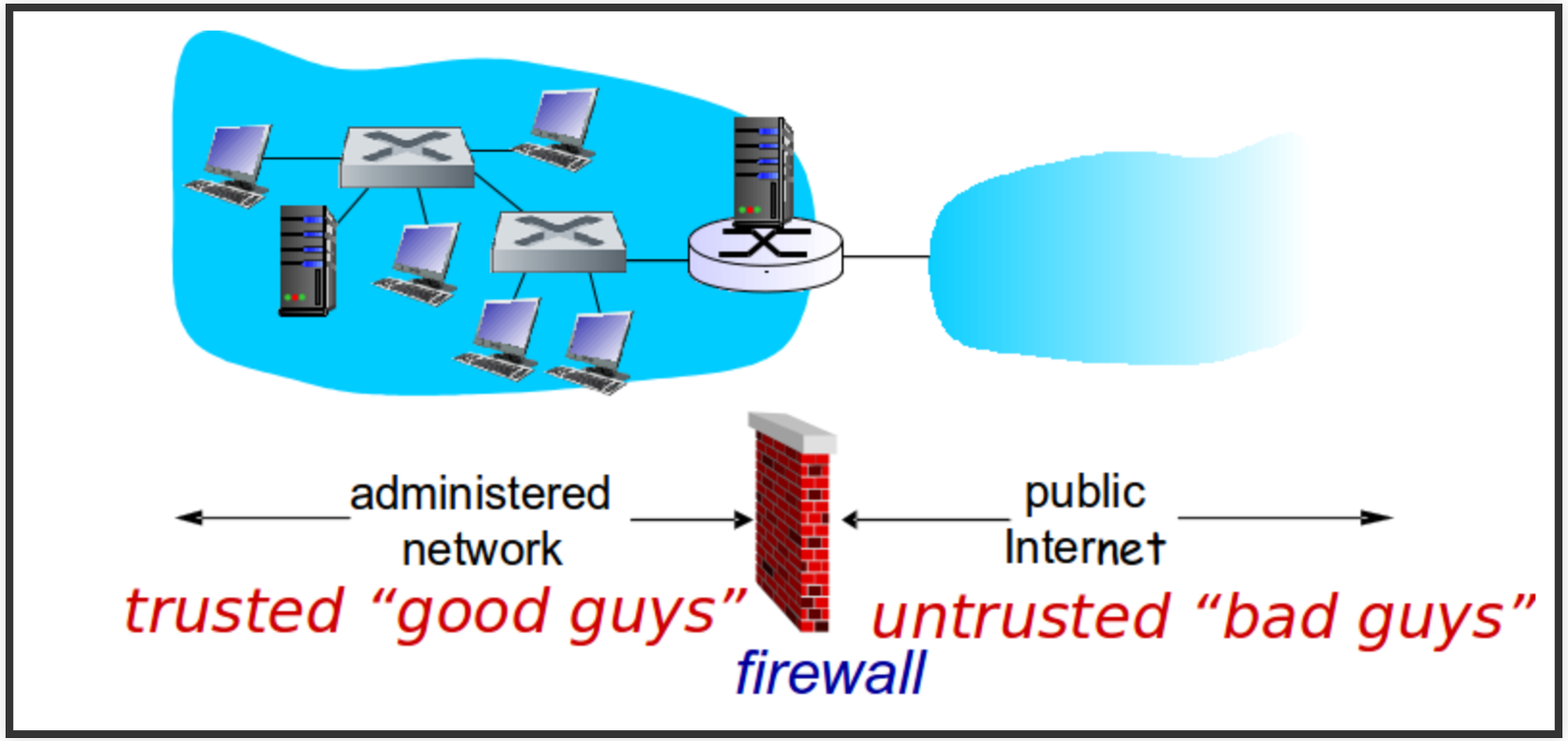
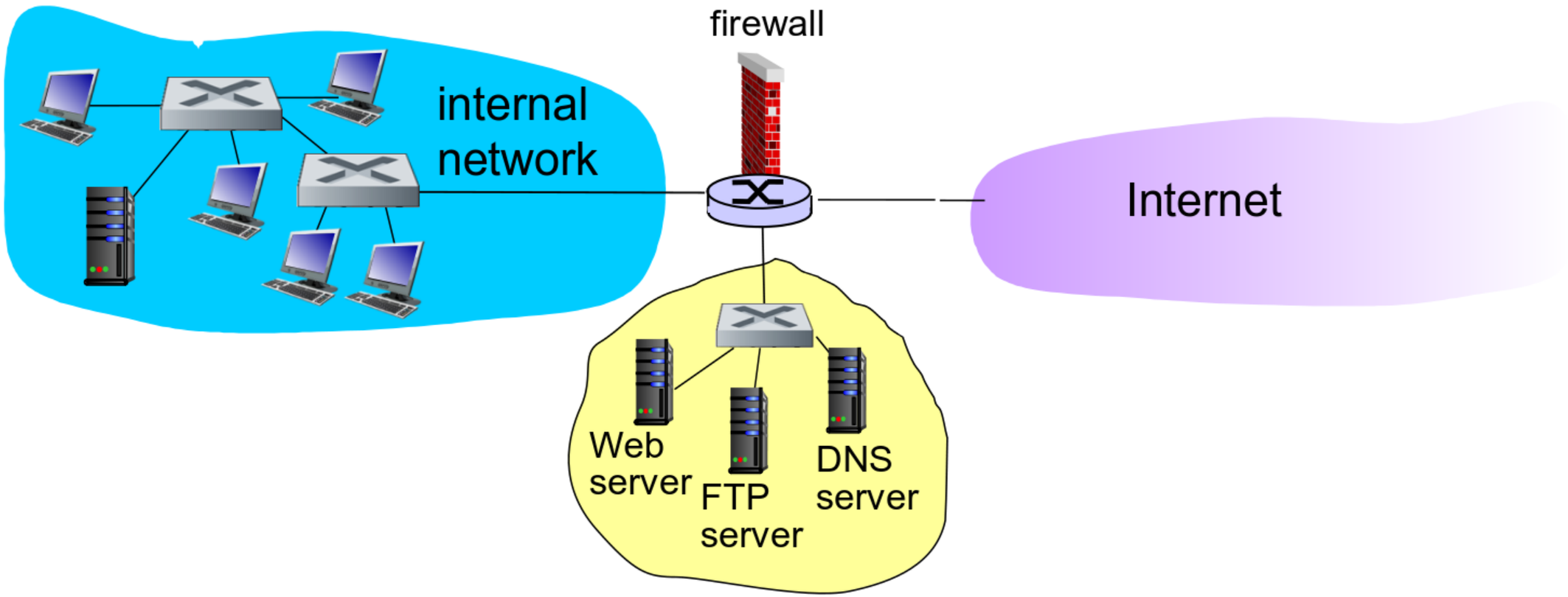
FIREWALLS

FIREWALLS

Firewall: isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others



FIREWALLS - DMZ



DMZ:
Demilitarized zone

FIREWALLS: WHY

Prevent denial of service attacks:

SYN flooding: attacker establishes many bogus TCP connections, no resources left for “real” connections

Prevent illegal modification/access of internal data

e.g., attacker replaces CIA’s homepage with something else

Allow only authorized access to inside network

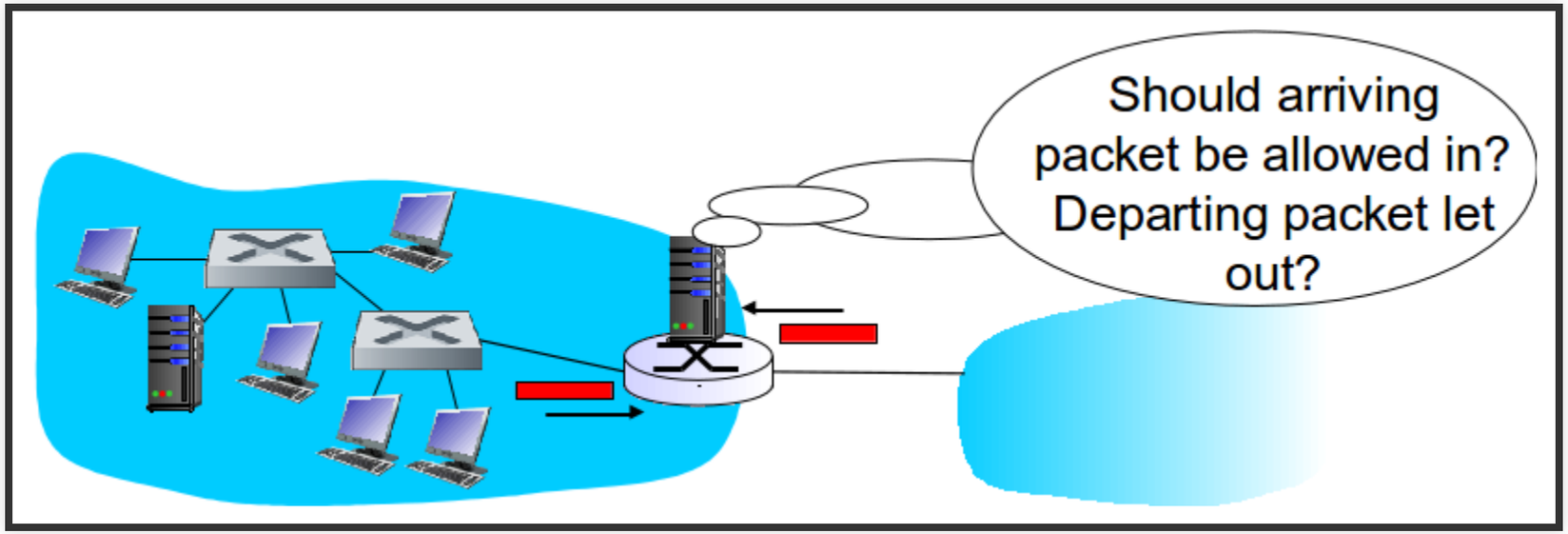
set of authenticated users/hosts

TYPES

Three types of firewalls:

1. Stateless packet filters
2. Stateful packet filters
3. Application gateways

STATELESS PACKET FILTERING



STATELESS PACKET FILTERING

- Internal network connected to Internet via **router firewall**
- Router filters packet-by-packet, decision to forward/drop packet based on:
  - Source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits

EXAMPLE 1

Block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23

Result: All incoming, outgoing UDP flows and telnet connections are blocked

EXAMPLE 2

Block inbound TCP segments with ACK=0.

Result: Prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

MORE EXAMPLES

Policy	Firewall Setting
No outside Web access.	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for institution's public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios from eating up the available bandwidth.	Drop all incoming UDP packets - except DNS and router broadcasts.

MORE EXAMPLES

Policy	Firewall Setting
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP packets going to a “broadcast” address (e.g. 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

ACCESS CONTROL LISTS

- **ACL:** Table of rules, applied top to bottom to incoming packets: (action, condition) pairs

ACCESS CONTROL LISTS (1)

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside 222.22/16	TCP	> 1023	80	any
allow	outside 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside 222.22/16	UDP	> 1023	80	-

ACCESS CONTROL LISTS (2)

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside 222.22/16	222.22/16	UDP	80	> 1023	-
deny	all	all	all	all	all	all

STATELESS PACKET FILTERING

Stateless packet filter: heavy handed tool

Admits packets that "make no sense," e.g., dest port = 80, ACK bit set, even though no TCP connection established:

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside 222.22/16	222.22/16	TCP	80	> 1023	ACK

Will repair this with stateful firewall.

STATEFUL PACKET FILTERING

STATEFUL PACKET FILTERING

Track status of every TCP connection

- Track connection setup (SYN), teardown (FIN): determine whether incoming, outgoing packets "makes sense"
  - Requires extra table of active connections
- Timeout inactive connections at firewall: No longer admit packets

ACL

ACL augmented to indicate need to check connection state table before admitting packet

ACL (1)

action	source address	dest address	ptcl	source port	dest port	flag bit	check conxi
allow	222.22/16	outside 222.22/16	TCP	> 1023	80	any	
allow	outside 222.22/16	222.22/16	TCP	80	> 1023	ACK	**X**
allow	222.22/16	outside 222.22/16	UDP	> 1023	80	-	

ACL (2)

action	source address	dest address	ptcl	source port	dest port	flag bit	check conxio
allow	outside 222.22/16	222.22/16	UDP	80	> 1023	-	
deny	all	all	all	all	all	all	

CONNECTION TABLE

Src Address	Dest Address	Src port	Dest port
222.22.1.24	37.123.12.213	12699	80
222.22.4.54	102.32.42.121	37823	80

Also have TTL/expires stamp

APPLICATION GATEWAYS

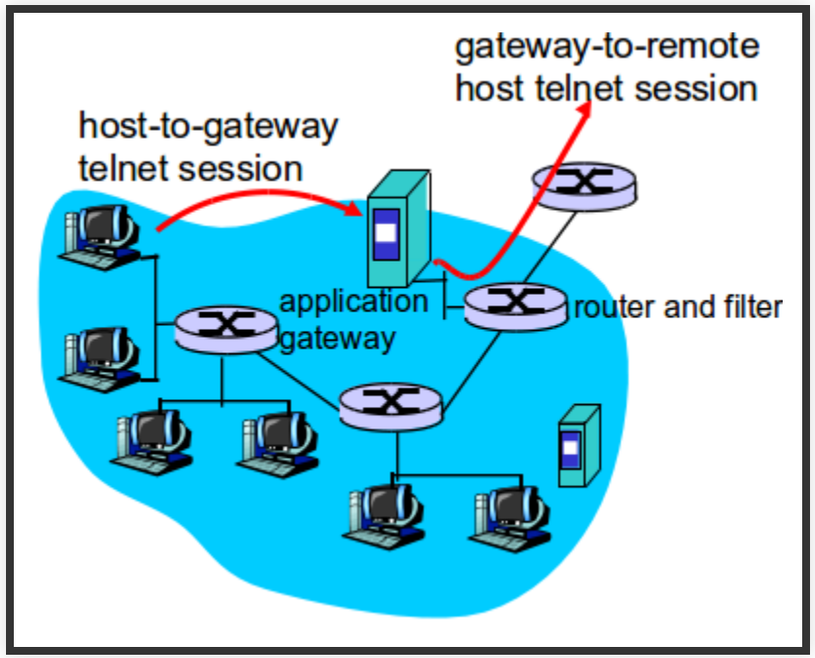
Filters packets on application data as well as on IP/TCP/UDP fields.

EXAMPLE: TELNET

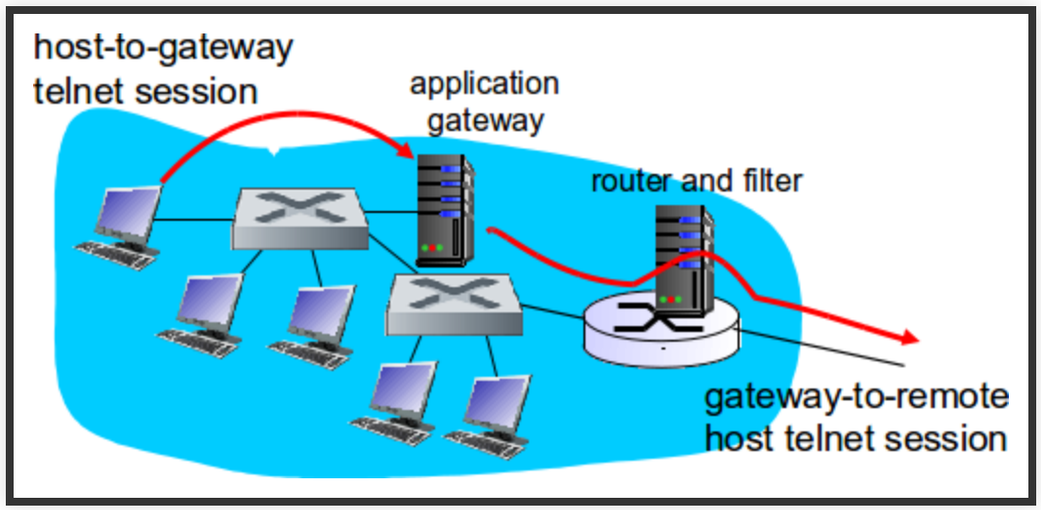
Allow selected internal users to telnet outside.

- Require all telnet users to telnet through gateway.
- For authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
- Router filter blocks all telnet connections not originating from gateway.

EXAMPLE: TELNET



EXAMPLE: TELNET



LIMITATIONS OF FIREWALLS, GATEWAYS

INTRUSION DETECTION SYSTEMS

WHY

For packet filtering:

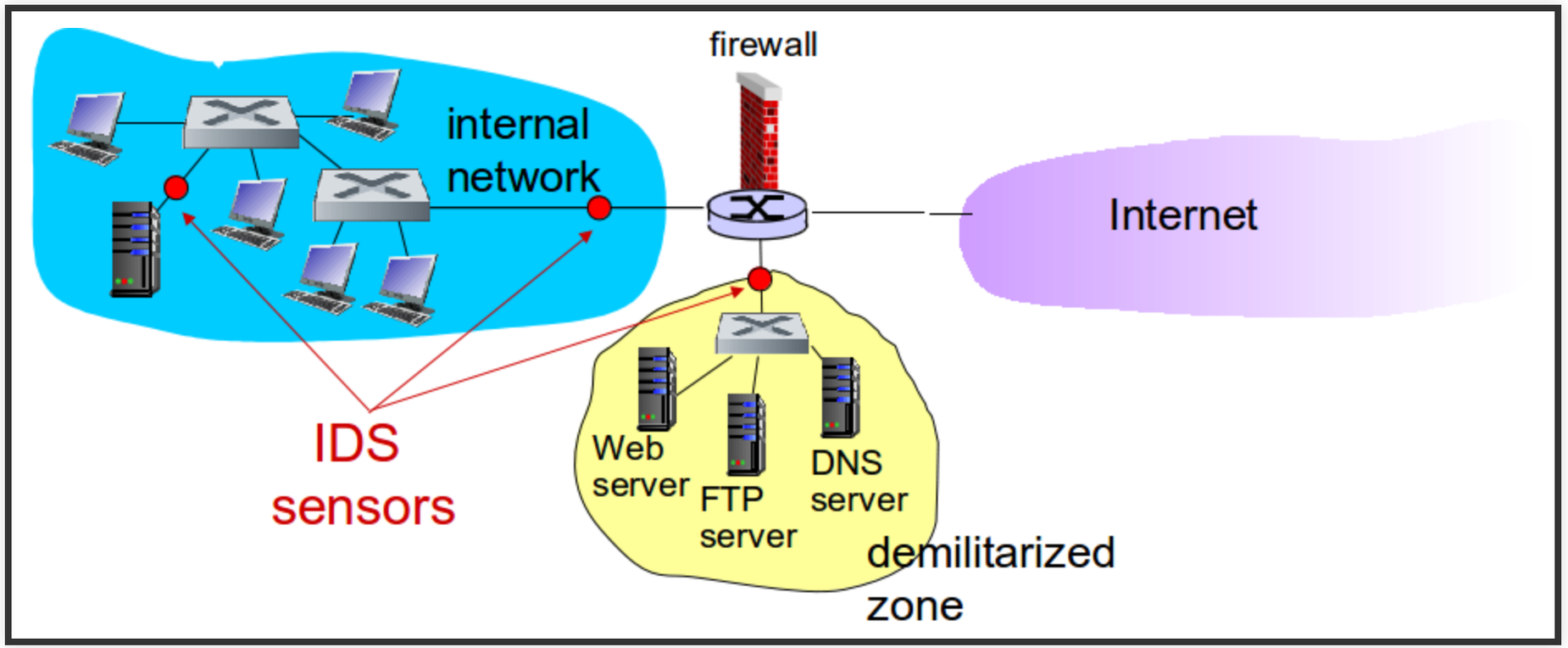
- Operates on TCP/IP headers only
- No correlation check among sessions

IDS: INTRUSION DETECTION SYSTEM

- **Deep packet inspection:** look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
- **Examine correlation** among multiple packets
  - Port scanning
  - Network mapping
  - DoS attack

INTRUSION DETECTION SYSTEMS

Multiple IDSs: different types of checking at different locations



INTRUSION PREVENTION SYSTEMS

- Intrusion detection systems typically raises an alarm by email/sms to the network admin
- An **Intrusion Prevention Systems** simply closes the connection in the firewall, if something suspicious is detected.

SIGNATURE-BASED IDS

- Maintains an extensive database of attack signatures
  - A signature is a set of rules describing an intrusion activity
  - May simply be a list of characteristics of a single packet (src, dest, portnumbers)
  - Can be related to a series of packages
- Signatures normally made by skilled network security engineers
  - Local system administrators can customize and add own

SIGNATURE-BASED IDS

Operations of a signature based IDS

- Sniffs every packet passing by it
- Compares packet with each signature in database
  - If it matches → generate an alert

SIGNATURE-BASED IDS

Limitations

- Require previous knowledge of attack to generate signature
- Can generate false positives
- Large processing load, and may fail in detection of malicious packets

ANOMALY-BASED IDS

- Creates a profile of standard network traffic
  - As observed in normal operation
- Then looks for packet streams that are statistically different
  - Example: Exponention growth in portscans or ping sweeps

ANOMALY-BASED IDS

Positive

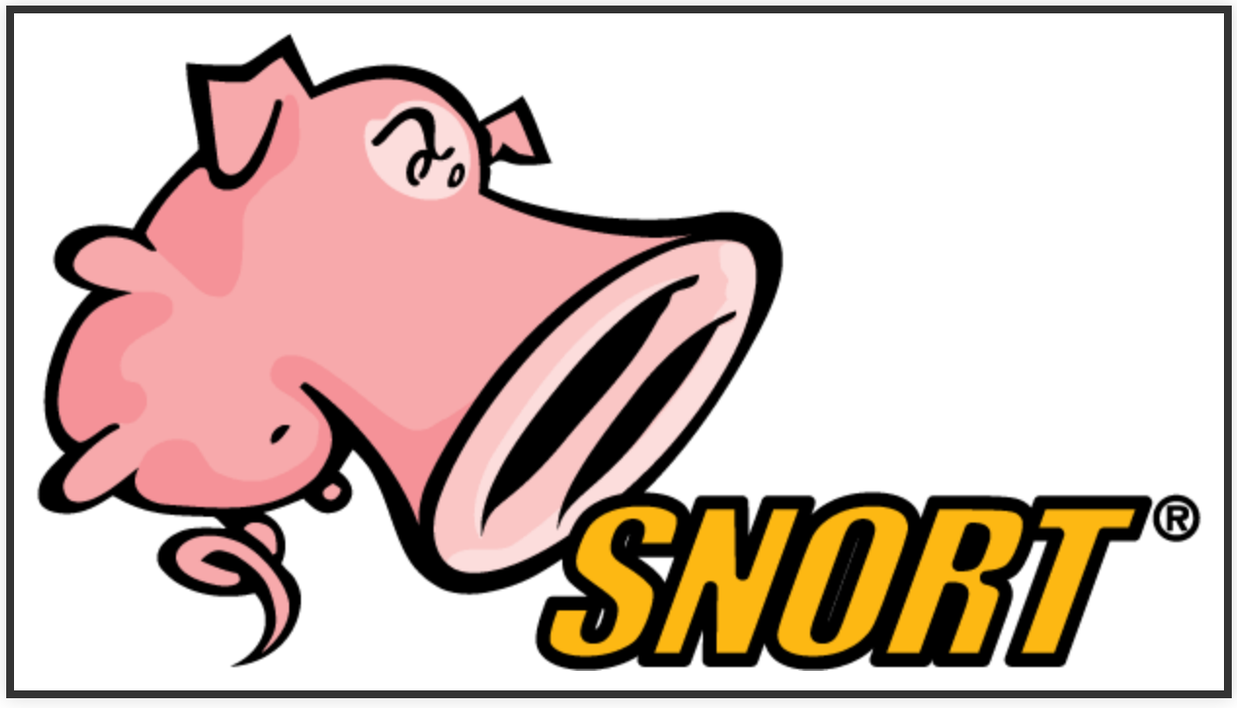
- Does not require prior knowledge to an attack

Limitation

- Extremely challenging to distinguis between normal an unusual traffic

> Most systems today are signature based

EXAMPLE IDS: SNORT



EXAMPLE IDS: SNORT

- https://www.snort.org/
- Multi platform
- Open source

EXAMPLE IDS: SNORT (1)

```
alert icmp any any -> $HOME_NET any (msg: "ICMP Test";
    sid: 1000001; rev:1;classtype:icmp-event;)
```

Anatomy: `<Rule header> (<Rule Options>)

RULE HEADER

```
alert icmp any any -> $HOME_NET any ()
```

- **alert**: Rule action. Snort will alert when the set condition is met.
- **icmp** :Protocol
- **any**: Source IP. Snort will look at all sources.
- **any** Source port. Snort will look at all ports.
- → Direction. From source to destination.
- **$HOME_NET**: Destination IP. Here HOME_NET value from the snort.conf file.
- **any** Destination port. Will look at all ports on the protected network.

RULE OPTIONS

```
(msg: "ICMP Test"; sid: 1000001; rev:1;classtype:icmp-event;)
```

- **msg:”ICMP test”**: Snort will include this message with the alert.
- **sid:1000001**: Snort rule ID. All numbers < 1,000,000 are reserved, this is why we are starting with 1000001 (you may use any number, as long as it’s greater than 1,000,000).
- **rev:1**: Revision number. This option allows for easier rule maintenance.
- **classtype:icmp-event** Categorizes the rule as an “icmp-event”, one of the predefined Snort categories. This option helps with rule organization.

SNORT DEMO

```
docker run -it --rm --net=host --cap-add=NET_ADMIN linton/docker-snort /bin/bash
cat /etc/snort/rules/local.rules
```

Check which interface you will monitor trafic using `ifconfig`

```
snort -i wlp2s0 -c /etc/snort/etc/snort.conf -A console
```

Run `ping 8.8.8.8` in terminal (new terminal, outside docker)

SNORT DEMO (2)

```
vim  /etc/snort/rules/local.rules
# Add line below
alert icmp any any -> $HOME_NET any (msg: "ICMP Test"; sid: 1000001; rev:1;classty
snort -i wlp2s0 -c /etc/snort/etc/snort.conf -A console
```

Outside, try ping, traceroute etc.

EXAMPLE IDS: SNORT (2)

```
alert tcp $EXTERNAL_NET any -> $HOME_NET
 3306 (msg:"SERVER-MYSQL MySQL COM_TABLE_DUMP Function Stack Overflow attempt";
 sid:11619; gid:3; rev:6; classtype:attempted-admin; reference:cve,2006-1518;
 reference:bugtraq,17780; reference:url,www.wisec.it/vulns.php?page=8;
 reference:cve,2006-1516; reference:cve,2006-1517; metadata: engine shared,
 soid 3|11619, service mysql;)
```

- https://nvd.nist.gov/vuln/detail/CVE-2006-1516
- Allows remote attackers to read portions of memory via a username without a trailing null byte, which causes a buffer over-read.

DDOS MITIGATION

DDoS mitigation refers to the process of successfully protecting a targeted server or network from a distributed denial-of-service (DDoS) attack.

By utilizing specially designed network equipment or a cloud-based protection service, a targeted victim is able to mitigate the incoming threat.

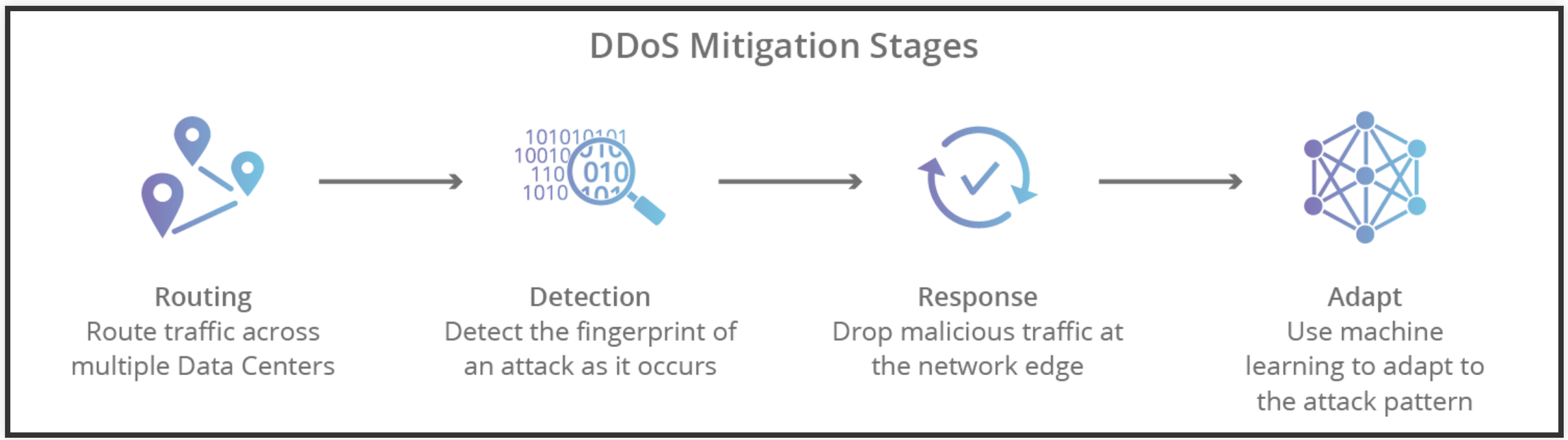
SPECIALLY DESIGNED NETWORK EQUIPMENT

Traditional DDoS mitigation solutions involved purchasing equipment that would live on site and filter incoming traffic. This approach involves purchasing and maintaining expensive equipment, and also relied on having a network capable of absorbing an attack.

Simple: Drop all traffic from foreign country IP’s in the companys firewall. This could work if you are a website in danish. Problem: Oversees citicents also excluded.

Configure firewall to drop traffic from all bots, based on list of analysed infected servers (paid service to get).

CLOUD-BASED PROTECTION SERVICE



STAGES OF MITIGATION

1. **Detection -** in order to stop a distributed attack, a website needs to be able to distinguish an attack from a high volume of normal traffic.
    - Product release or other announcement has a website swamped with legitimate new visitors, don-t prevent traffic.
2. **Response -** in this step, the DDoS protection network responds to an incoming identified threat by intelligently dropping malicious bot traffic, and absorbing the rest of the traffic. Using WAF page rules for application layer (L7) attacks, or another filtration process to handle lower level (L3/L4).

STAGES OF MITIGATION

3. **Routing** - By intelligently routing traffic, an effective DDoS mitigation solution will break the remaining traffic into manageable chunks preventing denial-of-service.
4. **Adaptation** - A good network analyzes traffic for patterns such as repeating offending IP blocks, particular attacks coming from certain countries, or particular protocols being used improperly. By adapting to attack patterns, a protection service can harden itself against future attacks.

EVALUATE CLOUD-BASED PROTECTION SERVICE

1. **Scalability** - an effective solution needs to be able to adapt to the needs of a growing business as well as respond to the growing size of DDoS attacks.
    - Attacks larger than 1 TB per second (TBPS) have occurred.
2. **Flexibility** - being able to create ad hoc policies and patterns allows a web property to adapt to incoming threats in real time.

EVALUATE CLOUD-BASED PROTECTION SERVICE

3. **Reliability** - DDoS protection is something you only need when you need it, but when that time comes it better be functional. Service should have high uptime rates and site reliability engineers working 24 hours a day to keep the network online and identify new threats.
4. **Network size** - DDoS attacks have patterns that occur across the Internet as particular protocols and attack vectors change over time. Having a large network with extensive data transfer allows a DDoS mitigation provider to analyze and respond to attacks quickly and efficiently, often stopping them before they ever occur.

QUESTIONS?