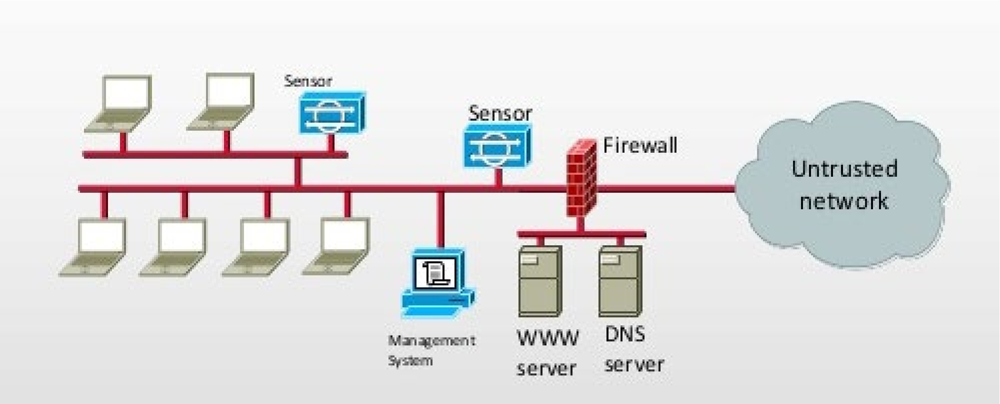
12.1 TYPES OF IDS

- Detecting Intrusions
- IDS Types
- IDS Components
- HIDS
- NIDS
- Indicators
- WIPS

TYPES OF INTRUSION DETECTION SYSTEMS

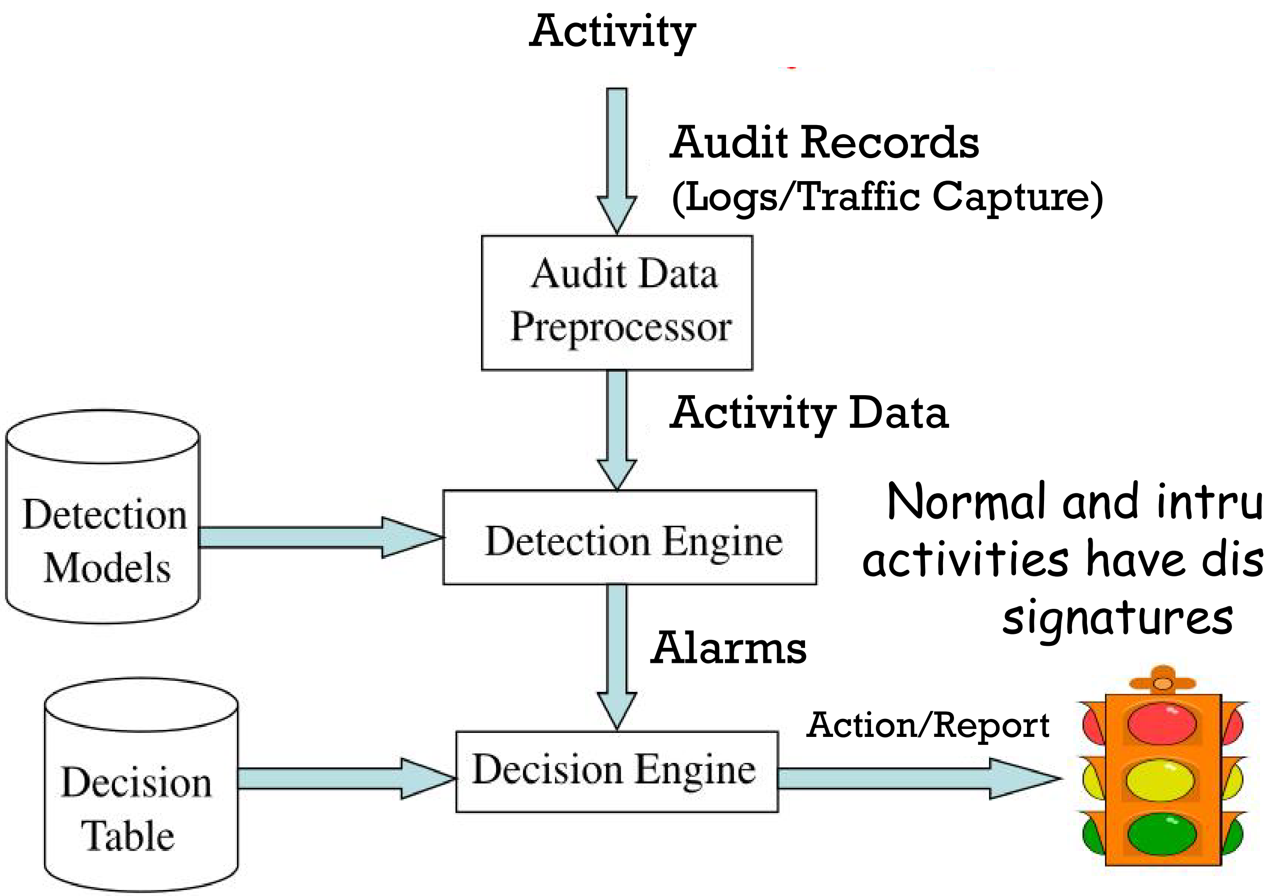
- Network-Based Intrusion Detection Systems
  - NIDS / NIPS
  - Black box on network in promiscuous mode
  - Detects malicious activity on the network
  - Does not detect anything going on in a host
- Host-Based Intrusion Detection Systems
  - HIDS / HIPS
  - Audits for events on a specific host
  - Requires overhead to monitor every system event
  - Only detects activity inside the host
  - Does not detect anything happening on the network



WAYS TO DETECT INTRUSIONS

- Signature-based
  - Can only detect known attacks for which a signature has previously been created
  - Must regularly download signatures from the vendor
  - Is at risk of false negatives
  - More commonly used by IDS
- Anomaly-based
  - Can identify unknown attacks
  - Must pre-create a baseline of "normal" network traffic
    - Capture network traffic for about two weeks
    - Analyze protocols and usage statistics to identify "normal"
  - Is at risk of false positives
  - More commonly used by IPS
- Protocol Anomaly Detection
  - Uses models to determine anomalies in how TCP/IP specifications are deployed

IDS COMPONENTS



Normal and intrusive activities have distinct signatures

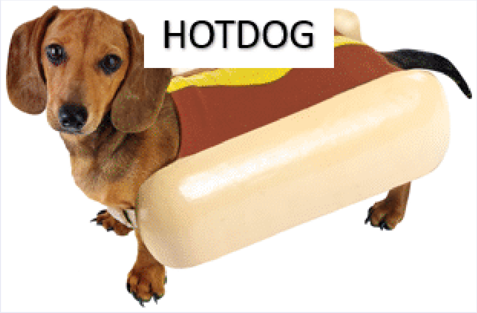
IDS RESULTS

- True Positive
  - There truly was a security incident
  - A real attack was detected
- True Negative
  - There truly was NOT any incident
  - Most desirable! Security controls are working!
- False Positive
  - False alarm
  - An incident was reported, but it didn’t actually happen
  - Too many false positives can become annoying
- False Negative
  - A security incident actually happened, but was not detected
  - IDS falsely reports that everything is ok
  - This is the most serious and dangerous of all!

IDS RESULTS EXAMPLE

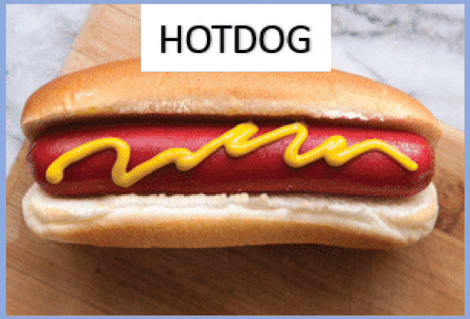
True negative

False positive



False negative

True positive

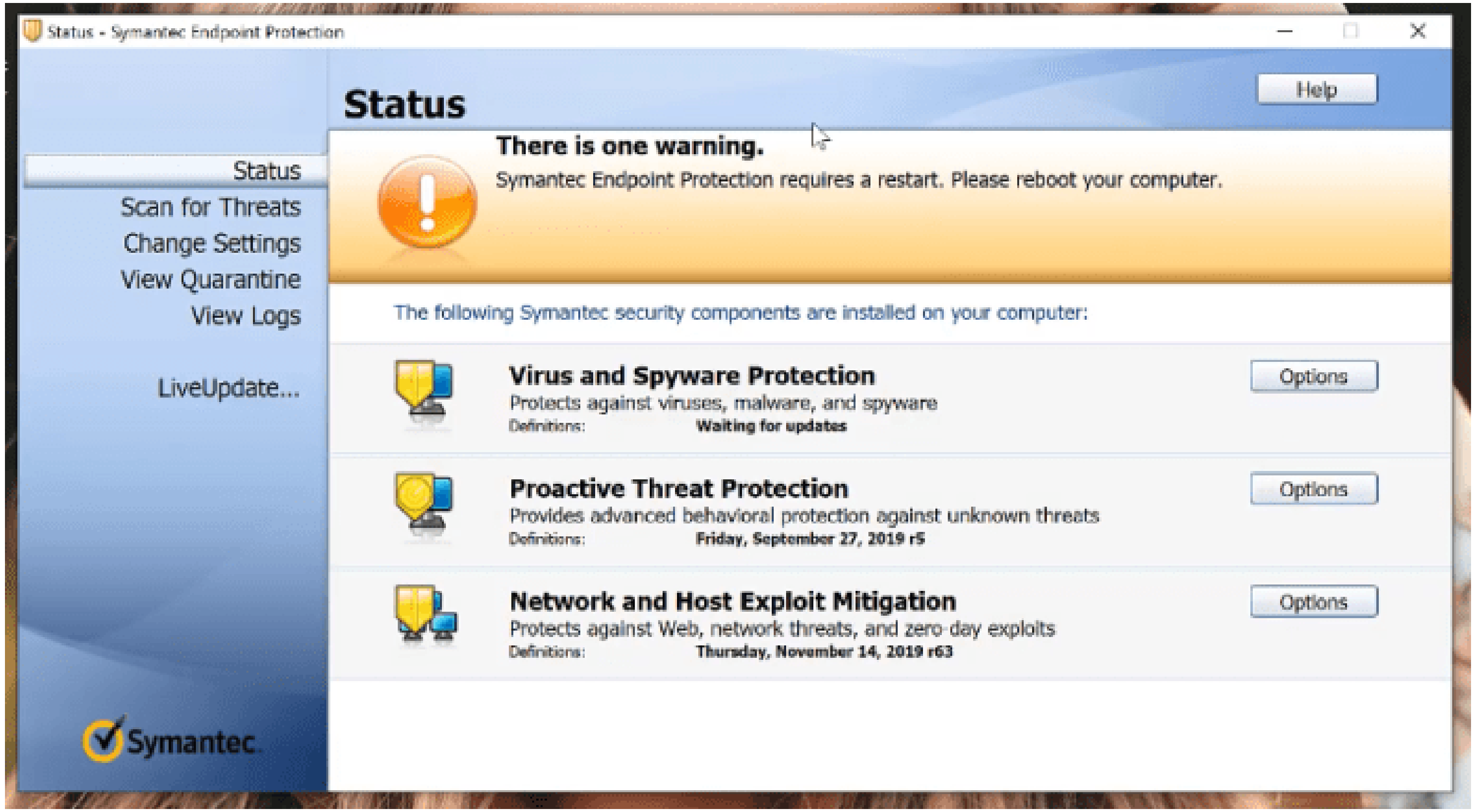


HOST-BASED IDS/IPS (HIDS/HIPS)

- Only activities inside the host are monitored:
  - File activity
  - Processes
  - Logons
  - Privileged actions
  - User account changes
  - Software installation/deletion
- Host-based IDS/IPS does not monitor network activity
  - Port and vulnerability scans, denial-of-service attacks against the host
- HIDS logs suspicious activities
- HIPS prevents suspicious activities

HIDS/HIPS PRODUCTS

- SolarWinds Log and Event Manager
- ManageEngine Log 360
- OSSEC
- Samhain
- Fail2Ban
- AIDE
- Sagan
- Security Onion
- Splunk
- Symantec Endpoint Protection



Status

There is one warning.

Symantec Endpoint Protection requires a restart. Please reboot your computer.

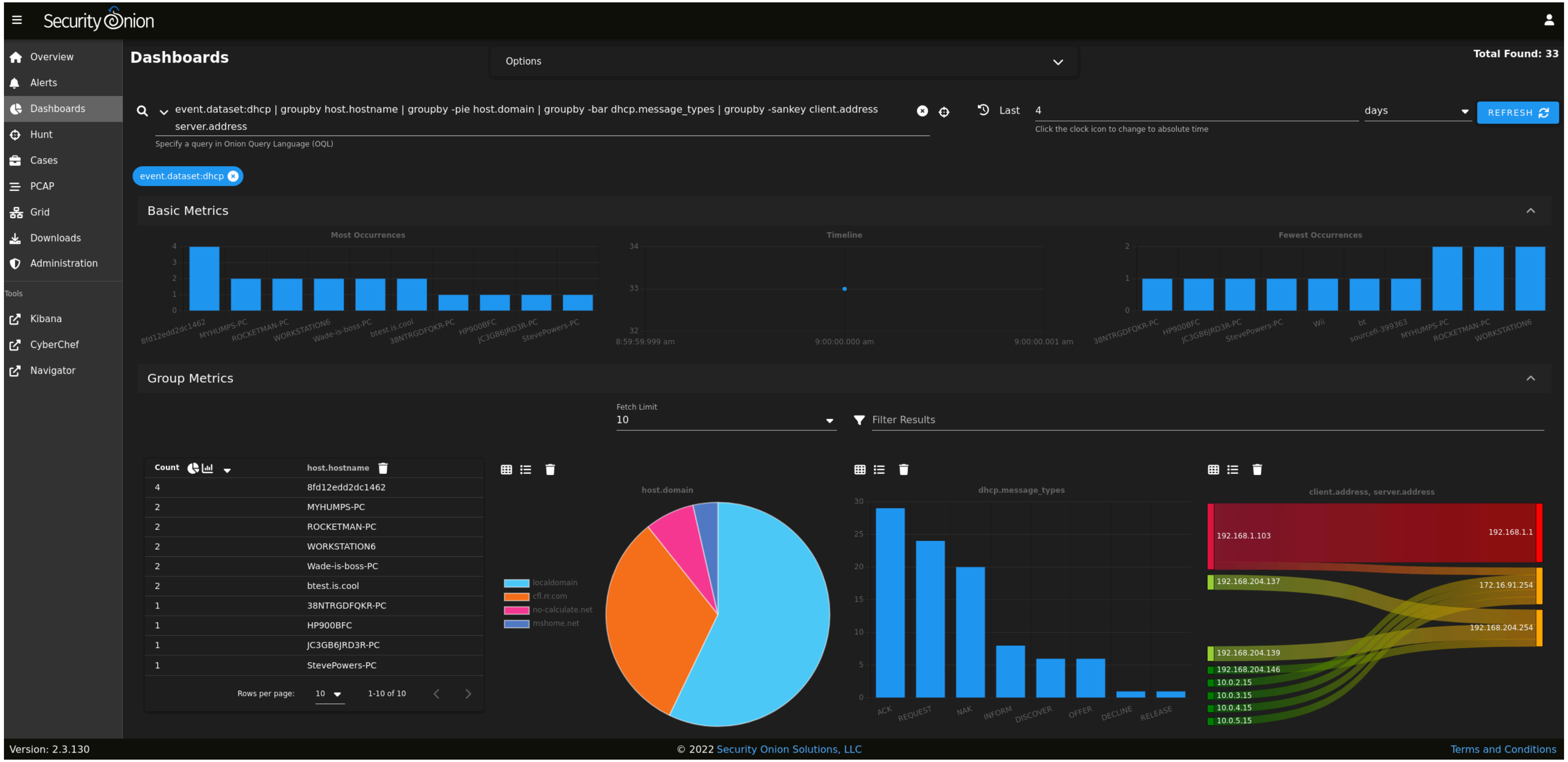
The following Symantec security components are installed on your computer:

Virus and Spyware Protection
Protects against viruses, malware, and spyware
Definitions: Waiting for updates

Proactive Threat Protection
Provides advanced behavioral protection against unknown threats
Definitions: Friday, September 27, 2019 r5

Network and Host Exploit Mitigation
Protects against Web, network threats, and zero-day exploits
Definitions: Thursday, November 14, 2019 r63

SECURITY ONION EXAMPLE



Dashboards

Total Found: 33

```
event.dataset:dhcp | groupby host.hostname | groupby -pie host.domain | groupby -bar dhcp.message_types | groupby -sankey client.address server.address
```

Specify a query in Onion Query Language (OQL)

Last 4 days

Click the clock icon to change to absolute time

event.dataset:dhcp

Fetch Limit 10

Count	host.hostname
4	8fd12edd2dc1462
2	MYHUMPS-PC
2	ROCKETMAN-PC
2	WORKSTATION6
2	Wade-is-boss-PC
2	btest.is.cool
1	38NTRGDFQKR-PC
1	HP900BFC
1	JC3GB6JRD3R-PC
1	StevePowers-PC

Rows per page: 10 1-10 of 10

Version: 2.3.130

© 2022 Security Onion Solutions, LLC

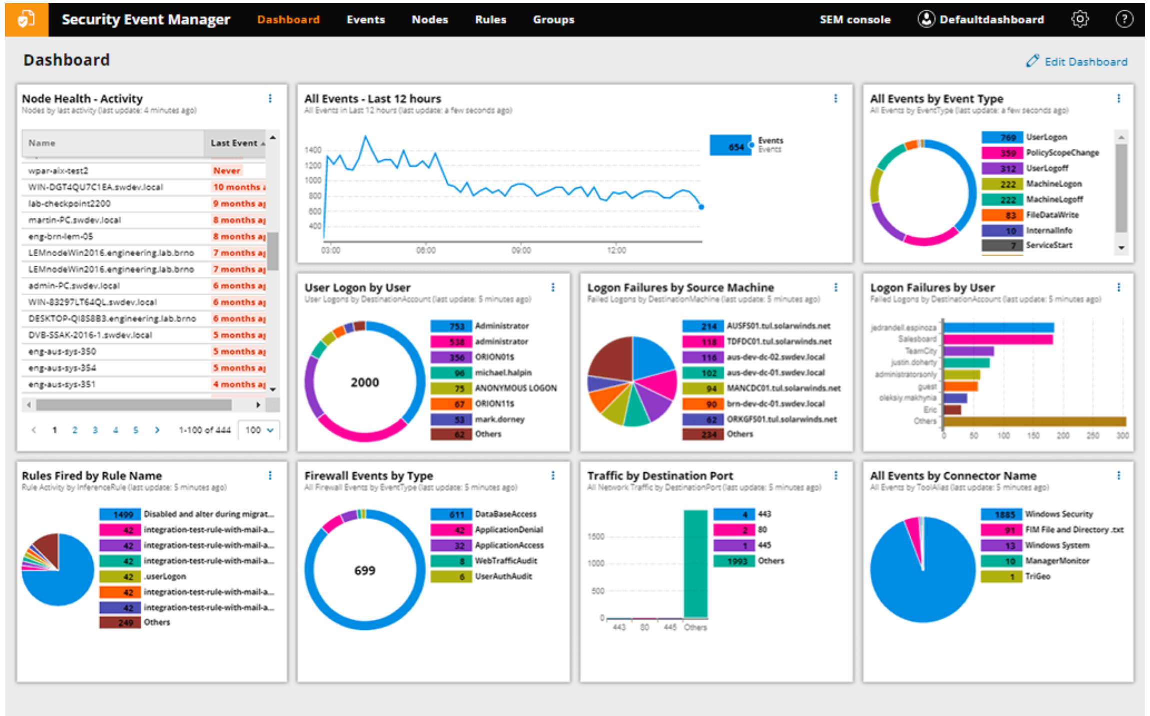
Terms and Conditions

NETWORK INTRUSION DETECTION (NIDS)

- A NIDS is a passive monitoring system
- Network traffic is examined as it passes by an IDS sensor
- The traffic is compared to a rule set
- If the traffic matches a rule it is logged
  - Optionally triggers an alert

NIDS EXAMPLE PRODUCTS

- SolarWinds
- Bro
- OSSEC
- Snort
- Suricata
- Security Onion
- Open WIPS-NG
- Sagan
- McAfee Network Security Platform
- Palo Alto Networks



Security Event Manager

Dashboard

Node Health - Activity

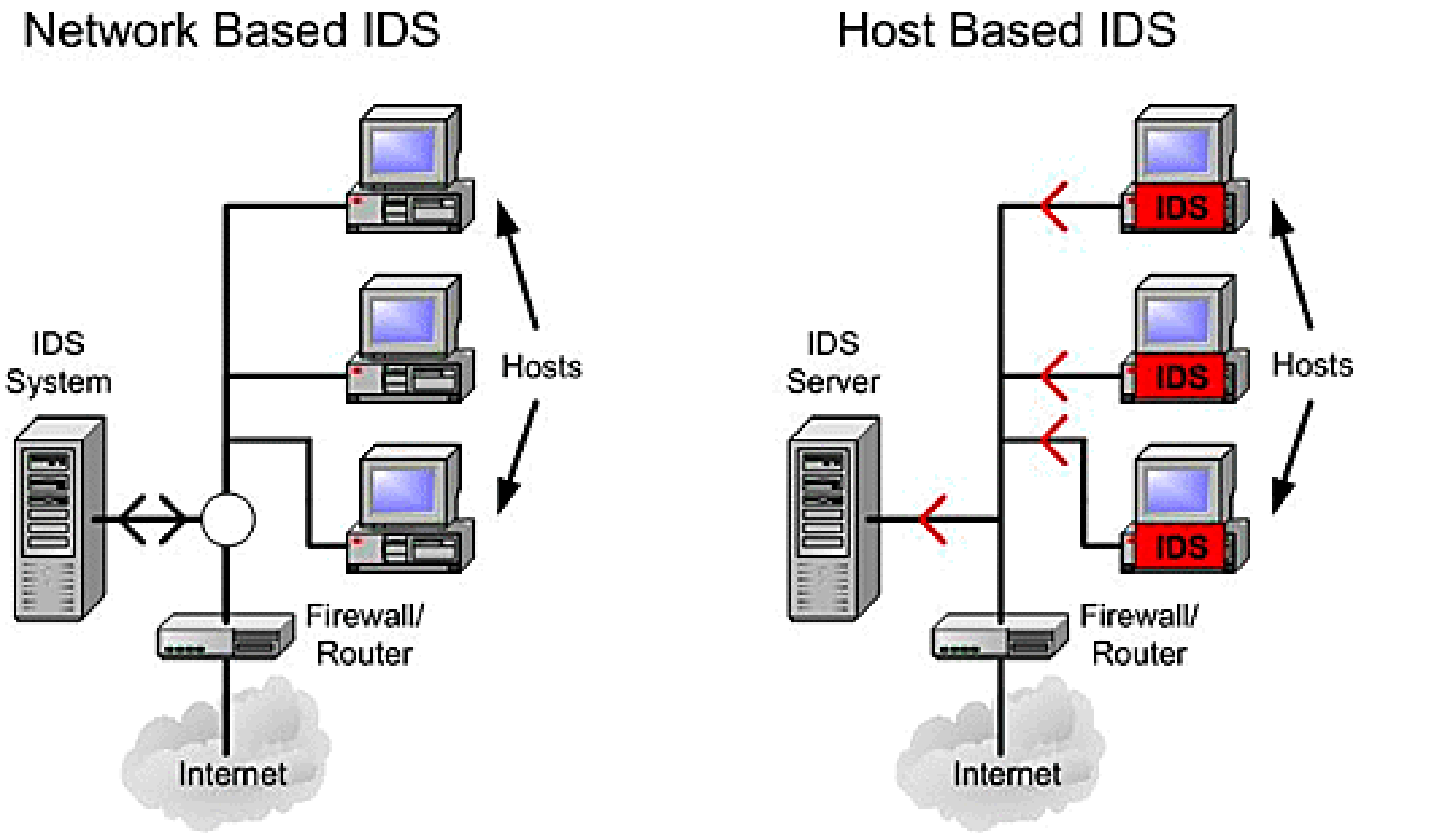
Nodes by last activity (last update: 4 minutes ago)

Name	Last Event
wpar-aix-test2	Never
WIN-DGT4QU7C1EA.swdev.local	10 months a
lab-checkpoint2200	9 months ag
martin-PC.swdev.local	8 months ag
eng-brn-lem-05	8 months ag
LEMnodeWin2016.engineering.lab.brno	7 months ag
LEMnodeWin2016.engineering.lab.brno	7 months ag
admin-PC.swdev.local	6 months ag
WIN-83297LT64QL.swdev.local	6 months ag
DESKTOP-QI8S8B3.engineering.lab.brno	6 months ag
DVB-SSAK-2016-1.swdev.local	5 months ag
eng-aus-sys-350	5 months ag
eng-aus-sys-354	5 months ag
eng-aus-sys-351	4 months ag

1-100 of 444

Dashboard
Events
Nodes
Rules
Groups
SEM console
Defaultdashboard
Edit Dashboard
1 2 3 4 5
100
All Events - Last 12 hours
All Events in Last 12 hours (last update: a few seconds ago)
1400
1200
1000
800
600
400
03:00
06:00
09:00
12:00
654 Events
Events
All Events by Event Type
All Events by EventType (last update: a few seconds ago)
769 UserLogon
359 PolicyScopeChange
312 UserLogoff
222 MachineLogon
222 MachineLogoff
83 FileDataWrite
10 InternalInfo
7 ServiceStart
User Logon by User
User Logons by DestinationAccount (last update: 5 minutes ago)
2000
753 Administrator
538 administrator
356 ORION01$
96 michael.halpin
75 ANONYMOUS LOGON
67 ORION11$
53 mark.dorney
62 Others
Logon Failures by Source Machine
Failed Logons by DestinationMachine (last update: 5 minutes ago)
214 AUSFS01.tul.solarwinds.net
118 TDFDC01.tul.solarwinds.net
116 aus-dev-dc-02.swdev.local
102 aus-dev-dc-01.swdev.local
94 MANCDC01.tul.solarwinds.net
90 brn-dev-dc-01.swdev.local
62 ORKGFS01.tul.solarwinds.net
234 Others
Logon Failures by User
Failed Logons by DestinationAccount (last update: 5 minutes ago)
jedrandell.espinoza
Salesboard
TeamCity
justin.doherty
administratorsonly
guest
oleksiy.makhynia
Eric
Others
0
50
100
150
200
250
300
Rules Fired by Rule Name
Rule Activity by InferenceRule (last update: 5 minutes ago)
1499 Disabled and alter during migrat...
42 integration-test-rule-with-mail-a...
42 integration-test-rule-with-mail-a...
42 integration-test-rule-with-mail-a...
42 .userLogon
42 integration-test-rule-with-mail-a...
42 integration-test-rule-with-mail-a...
249 Others
Firewall Events by Type
All Firewall Events by EventType (last update: 5 minutes ago)
699
611 DataBaseAccess
42 ApplicationDenial
32 ApplicationAccess
8 WebTrafficAudit
6 UserAuthAudit
Traffic by Destination Port
All Network Traffic by DestinationPort (last update: 5 minutes ago)
1500
1000
500
0
443
80
445
Others
4 443
2 80
1 445
1993 Others
All Events by Connector Name
All Events by ToolAlias (last update: 5 minutes ago)
1885 Windows Security
91 FIM File and Directory .txt
13 Windows System
10 ManagerMonitor
1 TriGeo

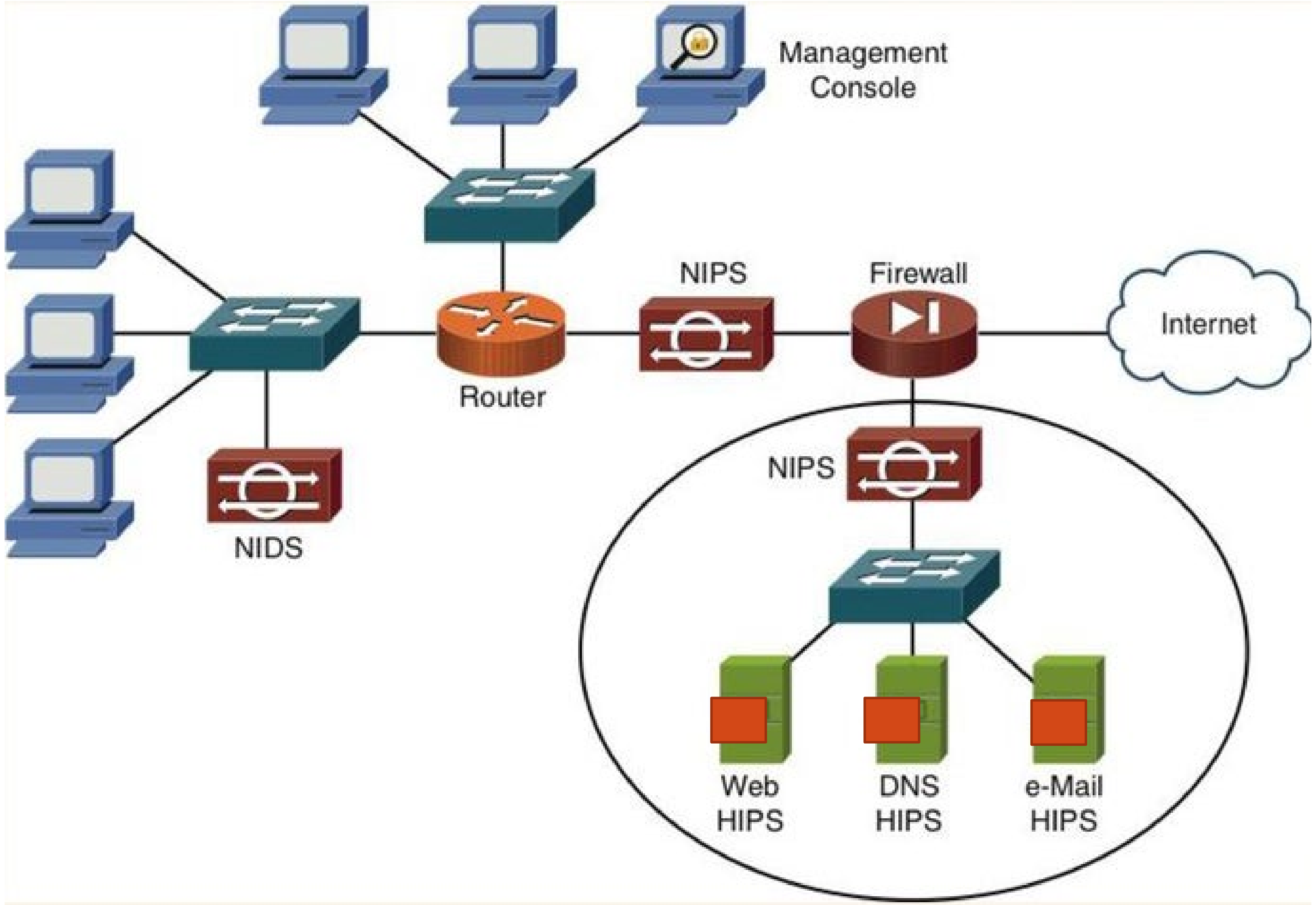
NIDS VS HIDS


Network Based IDS
IDS
System
Hosts
Firewall/
Router
Internet
Host Based IDS
IDS
Server
IDS
IDS
IDS
Hosts
Firewall/
Router
Internet

NETWORK INTRUSION PREVENTION (NIPS)

- A NIPS is an active monitoring and control system
- A network packet comes from Internet and passes through Firewall
- The packet passes through IDS and undergoes signature comparison
- If there is no match the packet is sent to the switch and into enterprise network
- If there is a match:
  - An alarm sent and logged
  - The packet is sent through anomaly detection and stateful protocol analysis
  - Connections from the source are cut
  - The packet is dropped

NIDS, NIPS, HIDS/HIPS PLACEMENT



INDICATORS OF NETWORK INTRUSIONS

- Ongoing probes of services on your network
- Unusual locations connecting to your network
- Ongoing remote login attempts
- Unauthorized data exfiltration
- Hosts with unexpected outbound connections
- Outbound connections to unusual destination ports

INDICATORS OF SYSTEM INTRUSIONS

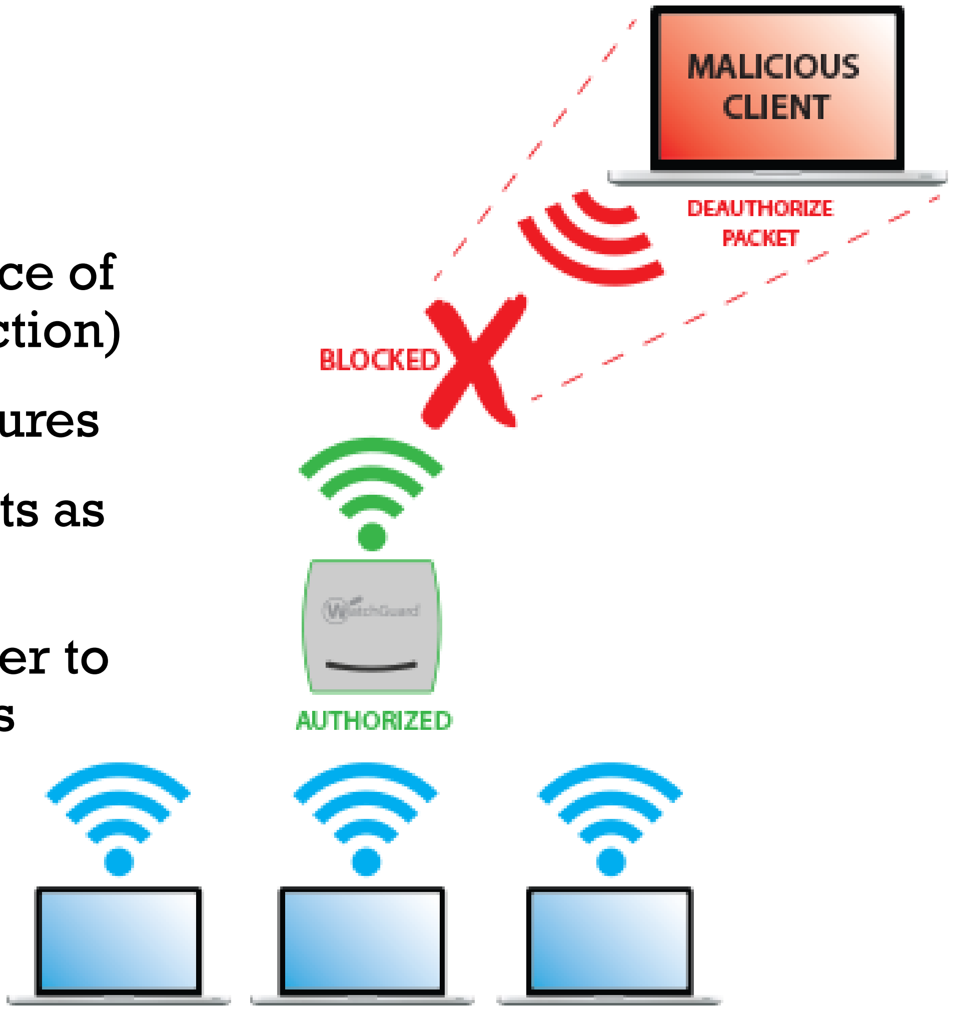
- New/unfamiliar files or programs detected
- Unfamiliar files names
- Files that are missing
- File permissions changed
- Files sizes changed unexpectedly
- Rogue files not on master list of signed files

INDICATORS OF SYSTEM INTRUSIONS (CONT'D)

- Incomplete/short logs
- Logs that are missing/have incorrect permissions
- Random data in log files that might cause DoS or a service crash
- Slow performance of the system
- Graphic displays/text messages that are unusual
- Alterations to system software/configuration files
- System crashes/reboots
- Processes that are unfamiliar

WI-FI IPS

- Wireless intrusion prevention system
- Monitors the radio spectrum for the presence of unauthorized access points (intrusion detection)
- Can automatically implement countermeasures
- The WIPS system uses wireless access points as sensors
- Management software is installed on a server to collect, analyze, and aggregate Wi-Fi events

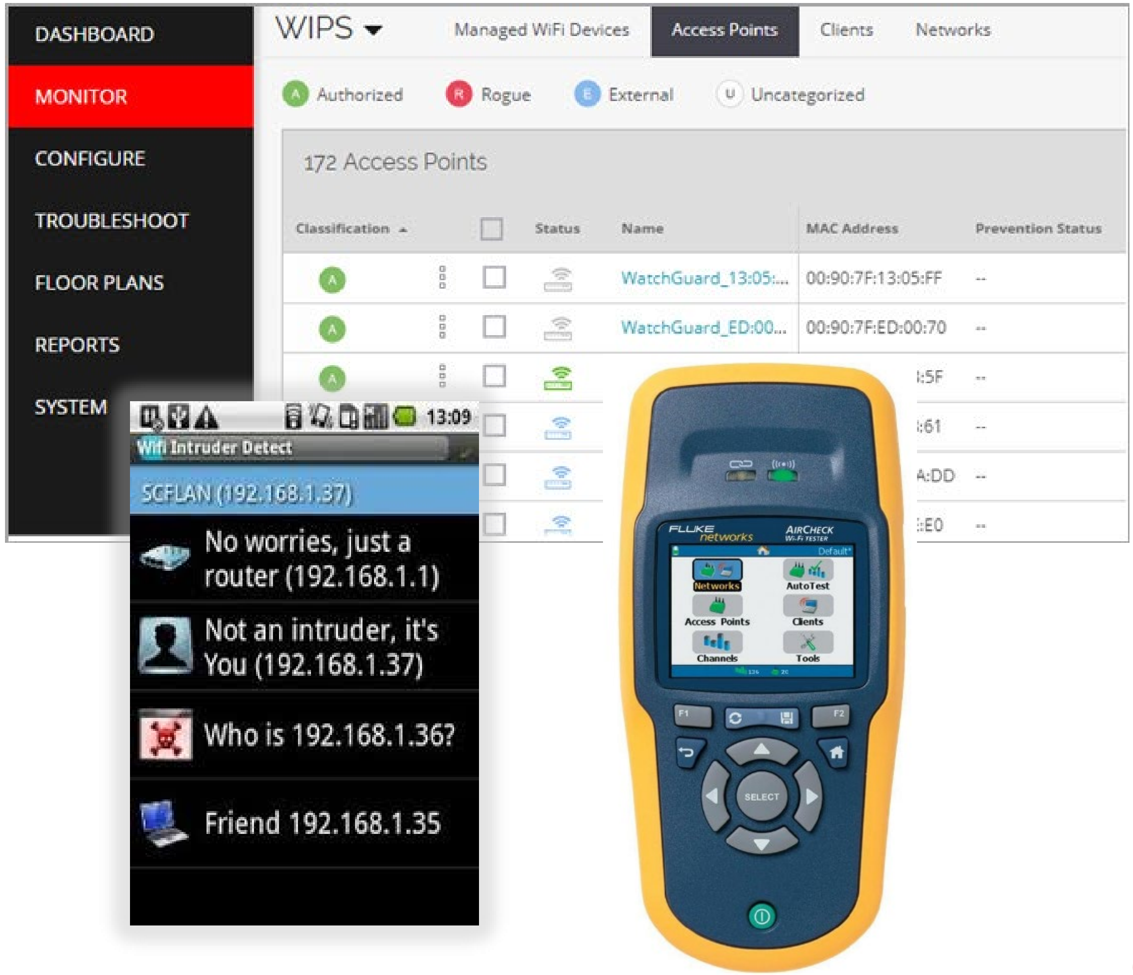


WIPS DEPLOYMENT MODELS

- The AP performs WIPS functions part of the time
    - Alternates between WIPS and its regular network connectivity functions
- The AP has dedicated WIPS functionality built into it
    - Performs network connectivity and WIPS functions at the same time
- The WIPS is deployed through dedicated sensors instead of the APs

WIPS EXAMPLE PRODUCTS

- Open WIPS-NG
- AirTight WIPS
- HP RFProtect
- Cisco Adaptive Wireless IPS
- Fluke Networks AirMagnet Enterprise
- HP Mobility Security IDS/IPS
- Zebra Technologies AirDefense
- WatchGuard
- WiFi Intruder Detector Pro
- WiFi Inspector



172 Access Points

Classification	□	Status	Name	MAC Address	Prevention Status
A	□		WatchGuard_13:05:...	00:90:7F:13:05:FF	--
A	□		WatchGuard_ED:00...	00:90:7F:ED:00:70	--
A	□			:5F	--
	□			:61	--
	□			A:DD	--
	□			:E0	--

13:09

Wifi Intruder Detect

SCFLAN (192.168.1.37)

No worries, just a router (192.168.1.1)

Not an intruder, it's You (192.168.1.37)

Who is 192.168.1.36?

Friend 192.168.1.35

12.2 SNORT

- Snort Overview
- Running Snort
- Snort Rules

SNORT

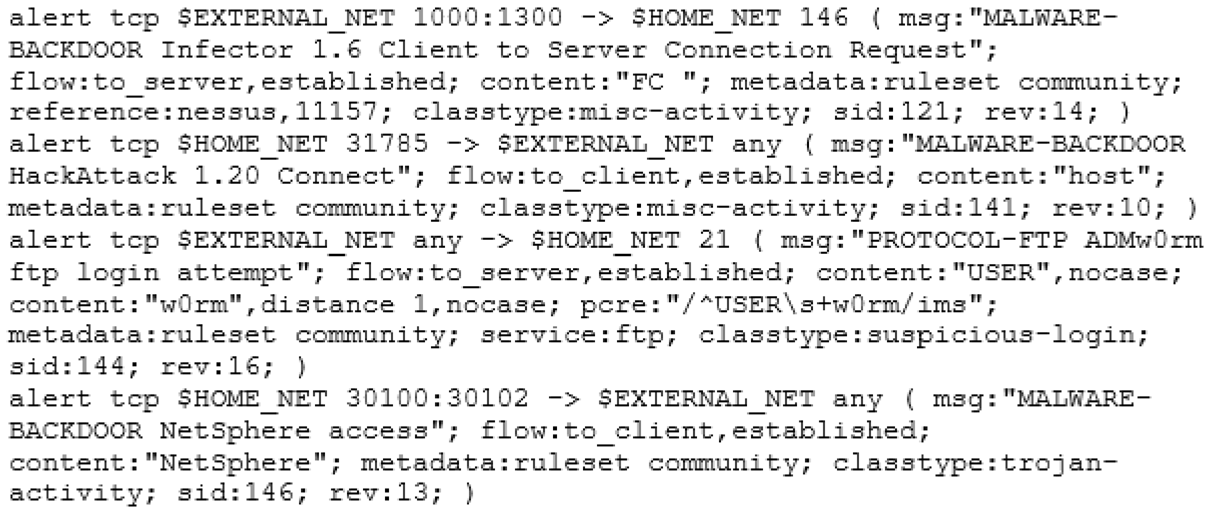
- Popular open source NIDS
- Runs in Linux or Windows
- You can create your own custom rules
- Will not block the connection or drop the packet
- Evaluates the entire packet against all alert rules
- Logs any matches it finds
- Allows packet to continue onward to its destination

SNORT MODES

- **Packet Sniffer**
  - SNORT's packet sniffer mode means the software will read IP packets then display them to the user on its console
- **Packet Logger**
  - In packet logger mode, SNORT will log all IP packets that visit the network
  - The network admin can then see who has visited their network and gain insight into the OS and protocols they were using
- **NIDS (Network Intrusion and Prevention Detection System)**
  - In NIDS mode, SNORT will only log packets that are considered malicious
  - It does this using the preset characteristics of malicious packets, which are defined in its rules
  - The action that SNORT takes is also defined in the rules the network admin sets out

SNORT CONFIGURATION FILE

- Snort.conf
- Located in:
  - /etc/snort on Linux
  - C:\snort\etc in Windows



```
alert tcp $EXTERNAL_NET 1000:1300 -> $HOME_NET 146 ( msg:"MALWARE-
BACKDOOR Infector 1.6 Client to Server Connection Request";
flow:to_server,established; content:"FC "; metadata:ruleset community;
reference:nessus,11157; classtype:misc-activity; sid:121; rev:14; )
alert tcp $HOME_NET 31785 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR
HackAttack 1.20 Connect"; flow:to_client,established; content:"host";
metadata:ruleset community; classtype:misc-activity; sid:141; rev:10; )
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 ( msg:"PROTOCOL-FTP ADMw0rm
ftp login attempt"; flow:to_server,established; content:"USER",nocase;
content:"w0rm",distance 1,nocase; pcre:"/^USER\s+w0rm/ims";
metadata:ruleset community; service:ftp; classtype:suspicious-login;
sid:144; rev:16; )
alert tcp $HOME_NET 30100:30102 -> $EXTERNAL_NET any ( msg:"MALWARE-
BACKDOOR NetSphere access"; flow:to_client,established;
content:"NetSphere"; metadata:ruleset community; classtype:trojan-
activity; sid:146; rev:13; )
```

SPECIFYING THE SNORT OPERATIONAL MODE

- Snort as Sniffer ---> snort -v
- Snort as Packet logger ---> snort -l
- Snort as NIDS ---> snort -A or snort -c <path_to_conf_file>

TESTING SNORT

- Test snort configuration and rules; check if there is any errors without starting up:

```
snort -i 4 -l c:\Snort\log -c c:\Snort\etc\snort.conf -T
```

- -i 4 ---> interface specifier, in case is interface 4
- -l ---> for logging
- -c ---> use Snort rules file specifying path
- -T ---> Only For testing, this prevent Snort from start up; Essentially to check if there is any error and if the rules are good.

STARTING SNORT

- This command starts a Snort NIDS, logging everything in ASCII:

```
snort -i 4 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii
```

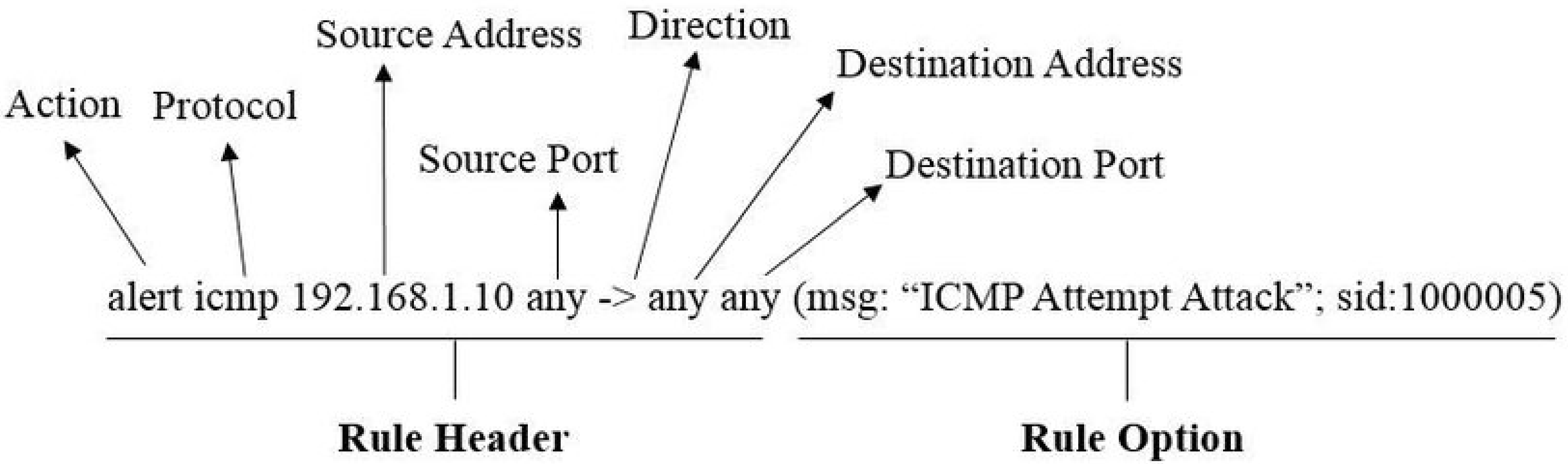
Flag	Information
-A	Set alert mode: fast, full, console, test or none
-b	Log packets in tcpdump format (much faster!)
-B <mask>	Obfuscate IP addresses in alerts and packet dumps using CIDR mask
-c <rules>	Use Rules file
-C	Print out payloads with character data only (no hex)
-l	Specifies the logging directory
-i <interface number>	Specifies which interface Snort should listen on
-K	Logging mode (pcap[default], ascii, none)
-?	Lists all switches and options and then exits

SNORT RULES

- Monitored protocols:
  - TCP
  - UDP
  - ICMP
- Rule Actions
  - Alert
  - Pass
  - Log

UNDERSTANDING SNORT RULES

- A security analyst should be able to read Snort IDS rules and pick out generic content such as:
  - The type of protocol covered by the signature
  - The port be analyzed
  - The direction of traffic flow



```
alert icmp 192.168.1.10 any -> any any (msg: “ICMP Attempt Attack”; sid:1000005)
```

BREAKING DOWN A SNORT RULE

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

Rule part	Information
alert icmp any any -> $HOME_NET any	**Rule Header** ⬇
alert	Rule action. Snort will generate an alert when the set condition is met.
any (1st)	Source IP. Snort will look at all sources
any (2nd)	Source port. Snort will look at all ports
->	Direction. From source to destination; *(source -> destination)*

BREAKING DOWN A SNORT RULE (CONT'D)

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

Rule part	Information
&HOME_NET	Destination IP. We are using the HOME_NET value from the snort.conf file which means a variable that defines the network or networks you are trying to protect.
any (3rd)	Destination port. Snort will look at all ports on the protected network

BREAKING DOWN A SNORT RULE (CONT'D)

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

Rule part	Information
msg:"ICMP test"	Snort will include this message with the alert
sid:1000001	Snort rule ID. All numbers below 1,000,000 are reserved If you create your own rule, assign it with any available number greater than 1,000,000

BREAKING DOWN A SNORT RULE (CONT'D)

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

Rule part	Information
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. Categories help with organizing rules

SNORT RULE EXAMPLES

TCP alert in a source IP address 192.168.x.x with any port; HOME_NET destination on port 21:

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

TCP alert in HOME_NET port 21 (FTP) as a source, to any destination IP address and port:

```
alert tcp $HOME_NET 21 -> any any (msg:"FTP failed login";
content:"Login or password incorrent"; sid:1000003; rev:1;)
```

This alerts about traffic that originated anywhere other than the internal network, going to the internal network port 31337:

```
alert tcp !HOME_NET any -> $HOME_NET 31337 (msg : "BACKDOOR
ATTEMPT-BackOrifice")
```

SNORT RULE EXAMPLE - SYN FLOOD

TCP Flag - NOT Ack

```
Detect TCP SYN FLOOD:

Alert tcp any any -> 192.168.10.5 443 (msg: “TCP SYN flood”; flags:!A;
flow: stateless; detection_filter: track by_dst, count 70, seconds 10;
sid:2000003;)
```

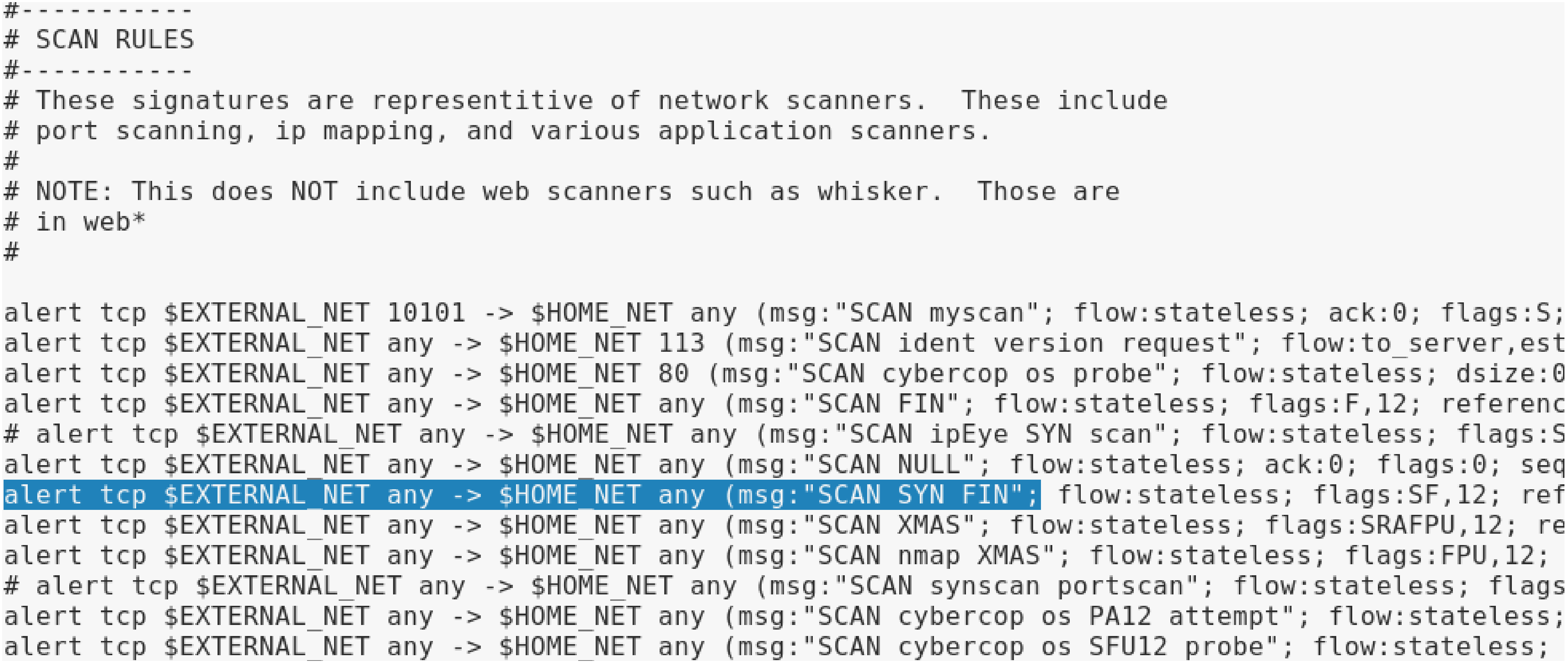
If we get more than 70 rule matches in a 10 second period, it’s a SYN flood!

Download Snort and Rules at: https://www.snort.org/downloads

SNORT RULE EXAMPLE - 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|\;|b3 c0 96 96 95 92 96|\;|f3|\;|24|i| 95 92|QO|8f f8|O|88 cf bc
c7 0f f7|2I|d0|w|c7 95 e4|O|d6 c7 17 f7 04 05 04 c3 f6 c6 86|D|fe c4
b1|1|ff 01 b0 c2 82 ff b5 dc b6 1b|O|95 e0 c7 17 cb|s|d0 b6|O|85 d8 c7
07|O|c0|T|c7 07 9a 9d 07 a4|fN|b2 e2|Dh|0c b1 b6 a8 a9 ab aa c4|]|e7
99 1d ac b0 b0 b4 fe eb eb|”; sid: 2000002; rev: 1;)
```

SNORT RULES TO IDENTIFY NETWORK SCANNERS



```
#-----------
# SCAN RULES
#-----------
# These signatures are representitive of network scanners.  These include
# port scanning, ip mapping, and various application scanners.
#
# NOTE: This does NOT include web scanners such as whisker.  Those are
# in web*
#

alert tcp $EXTERNAL_NET 10101 -> $HOME_NET any (msg:"SCAN myscan"; flow:stateless; ack:0; flags:S;
alert tcp $EXTERNAL_NET any -> $HOME_NET 113 (msg:"SCAN ident version request"; flow:to_server,est
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"SCAN cybercop os probe"; flow:stateless; dsize:0
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN FIN"; flow:stateless; flags:F,12; referenc
# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN ipEye SYN scan"; flow:stateless; flags:S
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN NULL"; flow:stateless; ack:0; flags:0; seq
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN"; flow:stateless; flags:SF,12; ref
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN XMAS"; flow:stateless; flags:SRAFPU,12; re
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap XMAS"; flow:stateless; flags:FPU,12;
# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN synscan portscan"; flow:stateless; flags
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN cybercop os PA12 attempt"; flow:stateless;
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN cybercop os SFU12 probe"; flow:stateless;
```

SNORT SCENARIO

```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any
msg: “BROWSER-IE Microsoft Internet Explorer
CacheSize exploit attempt”;
flow: to_client,established;
file_data;
     content:”recordset”; offset:14; depth:9;
     content:”.CacheSize”; distance:0; within:100;
     pcre:”/CacheSize\s*=\s*/”;
     byte_test:10,>,0x3ffffffe,0,relative,string;
max-detect-ips drop, service http;
reference:cve,2016-8077;
classtype: attempted-user;
sid:65535;rev:1;
```

What’s going on in this example?

SNORT SCENARIO

```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any
msg: “BROWSER-IE Microsoft Internet Explorer
CacheSize exploit attempt”;
flow: to_client,established;
file_data;
     content:”recordset”; offset:14; depth:9;
     content:”.CacheSize”; distance:0; within:100;
     pcre:”/CacheSize\s*=\s*/”;
     byte_test:10,>,0x3ffffffe,0,relative,string;
max-detect-ips drop, service http;
reference:cve,2016-8077;
classtype: attempted-user;
sid:65535;rev:1;
```

Alert only on TCP packets

Only analyze inbound traffic for established connections

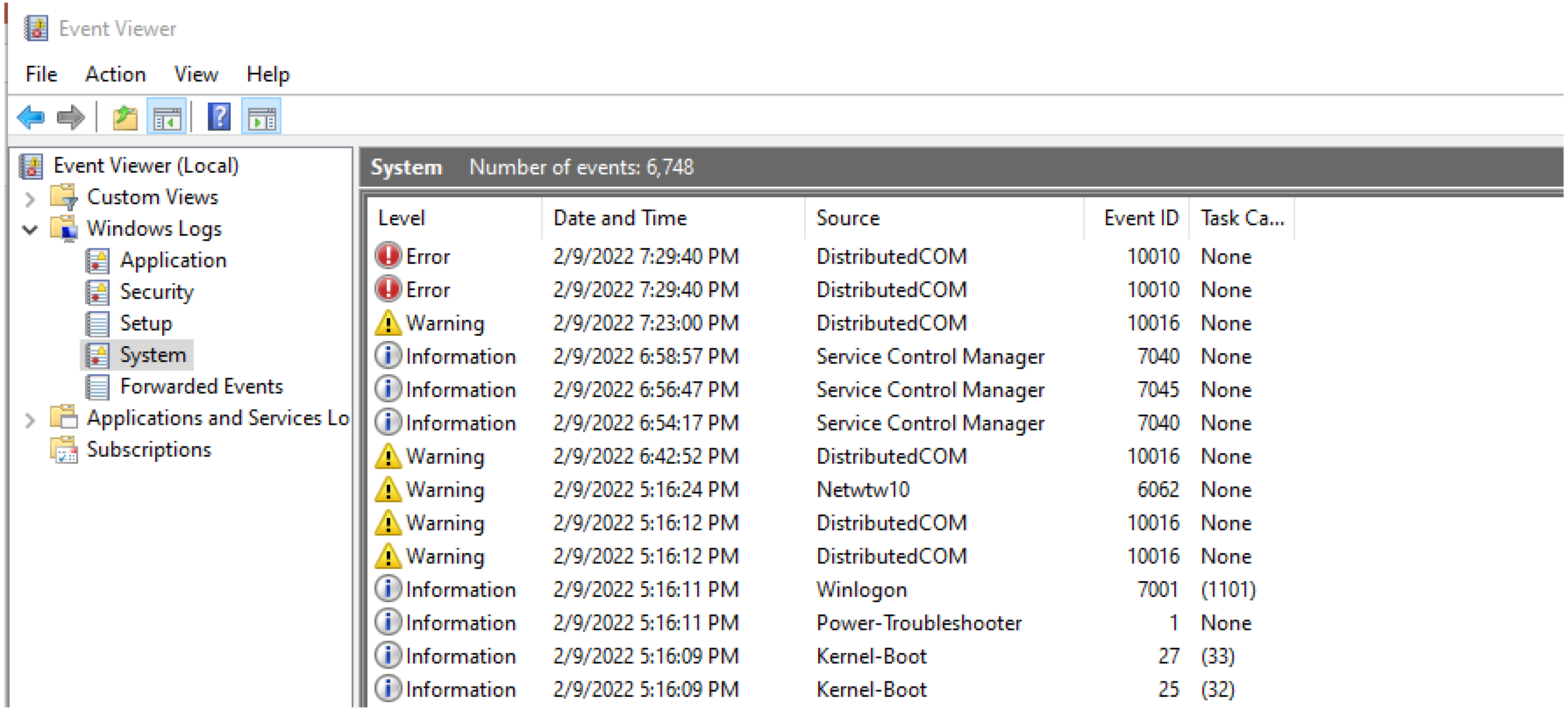
12.3 SYSTEM LOGS

- OS Logs
- Syslog

SYSTEM LOGS

- Nearly all devices have a system log
  - Windows Event Viewer
  - Linux has many logs located in /var/log
  - Most routers, switches, firewalls and other network devices have their own logs
- Log events can be sent to a central Syslog server
  - On Windows, install the Syslog server's client software
  - On Linux, edit /etc/syslog.conf to point to the Syslog server
- Logs can also be queried by a SIEM or IDS

SYSTEM LOG EXAMPLE



System Number of events: 6,748

Level	Date and Time	Source	Event ID	Task Ca...
Error	2/9/2022 7:29:40 PM	DistributedCOM	10010	None
Error	2/9/2022 7:29:40 PM	DistributedCOM	10010	None
Warning	2/9/2022 7:23:00 PM	DistributedCOM	10016	None
Information	2/9/2022 6:58:57 PM	Service Control Manager	7040	None
Information	2/9/2022 6:56:47 PM	Service Control Manager	7045	None
Information	2/9/2022 6:54:17 PM	Service Control Manager	7040	None
Warning	2/9/2022 6:42:52 PM	DistributedCOM	10016	None
Warning	2/9/2022 5:16:24 PM	Netwtw10	6062	None
Warning	2/9/2022 5:16:12 PM	DistributedCOM	10016	None
Warning	2/9/2022 5:16:12 PM	DistributedCOM	10016	None
Information	2/9/2022 5:16:11 PM	Winlogon	7001	(1101)
Information	2/9/2022 5:16:11 PM	Power-Troubleshooter	1	None
Information	2/9/2022 5:16:09 PM	Kernel-Boot	27	(33)
Information	2/9/2022 5:16:09 PM	Kernel-Boot	25	(32)

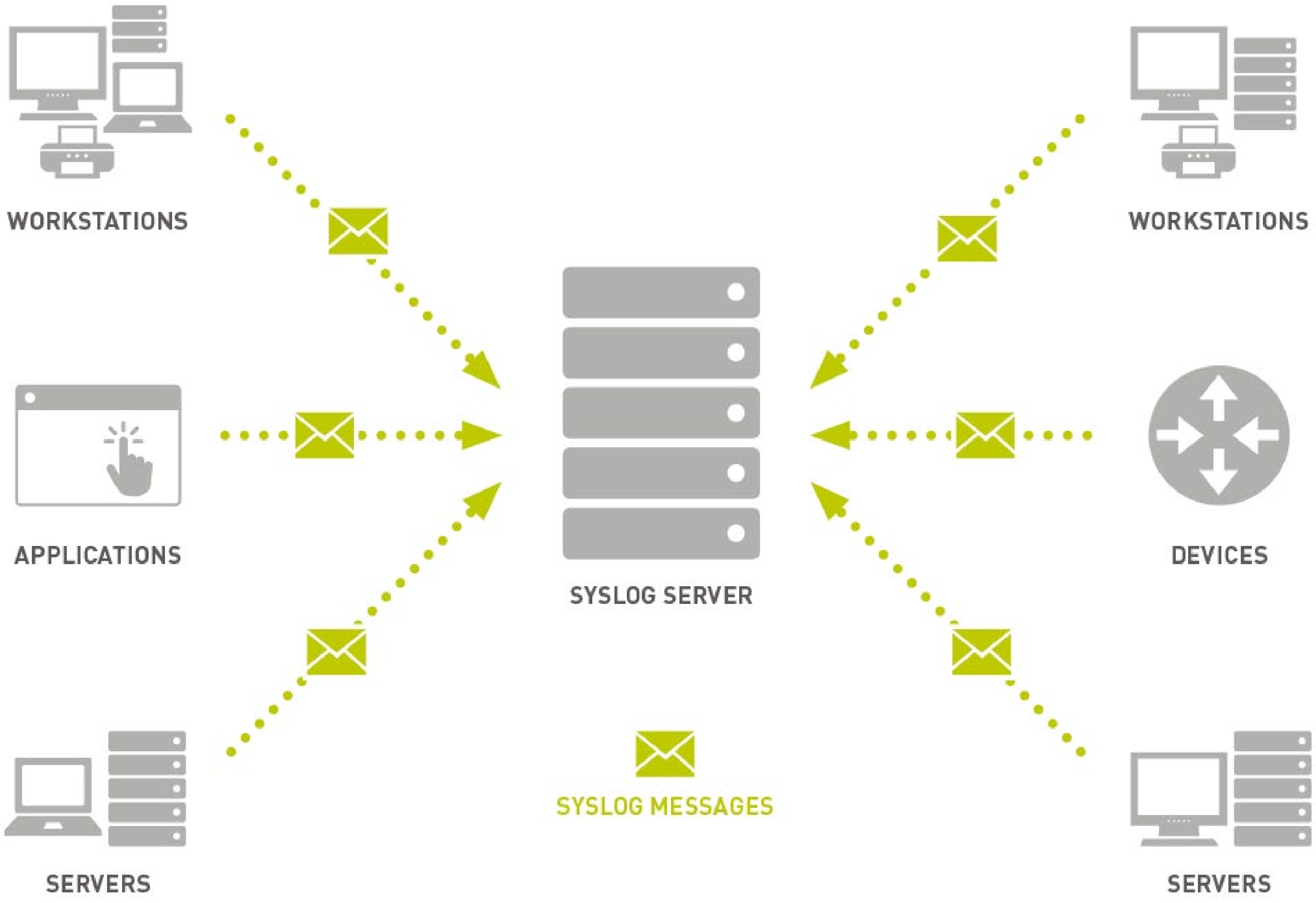
SYSTEM LOGGING CONSIDERATIONS

- System logs of hosts and network devices must be time-synchronized
- IDS and sysadmin must be able to cross-correlate events
- Logging can be resource intensive for the device
  - If you are about to enable logging / OS auditing on a system for the first time, consider the impact of enabling the audit feature on system performance

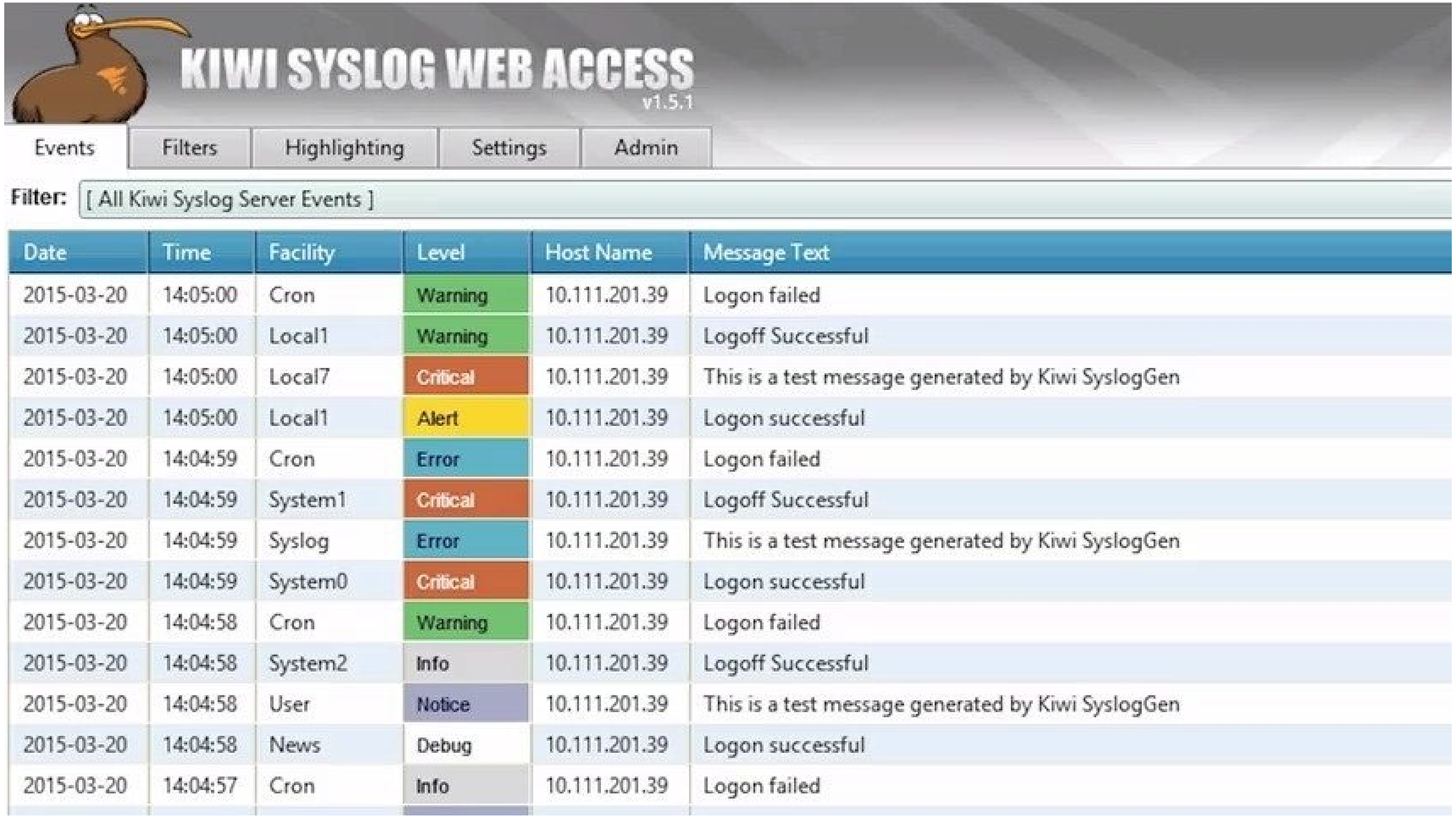
SYSLOG

- Widely used standard for capture log messages from a variety of devices
  - Events are sent by devices to a centralized syslog server
  - Even if the attacker manages to delete logs on the compromised system, the syslog server has a copy
- Separates logging roles into:
  - Software that generates messages
  - The system that stores the messages
  - The software that analyzes the messages and creates reports
- Messages include:
  - time stamps, event messages, severity, host IP addresses, diagnostics, and more
- Kiwi syslog server is a popular syslog product
- Syslog uses UDP 514

SYSLOG EXAMPLE



KIWI SYSLOG SERVER EXAMPLE



Filter: [All Kiwi Syslog Server Events]

Date	Time	Facility	Level	Host Name	Message Text
2015-03-20	14:05:00	Cron	Warning	10.111.201.39	Logon failed
2015-03-20	14:05:00	Local1	Warning	10.111.201.39	Logoff Successful
2015-03-20	14:05:00	Local7	Critical	10.111.201.39	This is a test message generated by Kiwi SyslogGen
2015-03-20	14:05:00	Local1	Alert	10.111.201.39	Logon successful
2015-03-20	14:04:59	Cron	Error	10.111.201.39	Logon failed
2015-03-20	14:04:59	System1	Critical	10.111.201.39	Logoff Successful
2015-03-20	14:04:59	Syslog	Error	10.111.201.39	This is a test message generated by Kiwi SyslogGen
2015-03-20	14:04:59	System0	Critical	10.111.201.39	Logon successful
2015-03-20	14:04:58	Cron	Warning	10.111.201.39	Logon failed
2015-03-20	14:04:58	System2	Info	10.111.201.39	Logoff Successful
2015-03-20	14:04:58	User	Notice	10.111.201.39	This is a test message generated by Kiwi SyslogGen
2015-03-20	14:04:58	News	Debug	10.111.201.39	Logon successful
2015-03-20	14:04:57	Cron	Info	10.111.201.39	Logon failed

12.4 IDS CONSIDERATIONS

- NIDS Considerations
- HIDS Considerations

NIDS CONSIDERATIONS

- System clock of all monitored and monitoring devices must be synchronized
- Network sensors (taps) must be strategically placed
  - Must have traffic pass their interface
  - Best way is to configure port spanning (mirroring) on a switch
  - The switch copies all traffic to/from a particular port (where the server is connected) to the mirrored port

NIPS CONSIDERATIONS

- Provides defense-in-depth protection in addition to a firewall
  - It is not typically used as a replacement
  - A NIPS cannot handle the same heavy workload that a firewall can handle
- A false positive by a NIPS is more damaging than one by a NIDS
  - Legitimate traffic will be denied
  - This may cause production problems
  - A NIPS usually has a smaller set of rules compared to a NIDS for this reason
    - Only the most trustworthy rules are used
- A NIPS is not a replacement for a NIDS
  - Many networks use both a NIDS and a NIPS
  - To assist a NIPS, you can turn on the built-in auditing feature in the operating system
  - Can slow system performance as well as take up a lot of disk space

HIDS/HIPS CONSIDERATIONS

- Once installed, nearly impossible to uninstall
  - The product replaces some OS components
- Can only detect activity happening within the OS
  - Cannot detect ping sweeps, port scans, and non-intrusive vulnerability scans
- Does not prevent intrusions or attacks
- Can be installed on network points such as routers or servers, but cannot monitor at the network level
- Does not filter incoming/outgoing traffic based on rules, the way a firewall does, or a bandwidth monitor does
- Is most effective as a solution if it can forward events from individual hosts to a centralized log server, or even a cloud-based SIEM

TUNING IDS/IPS SECURITY ALERTS

- Some IDS/IPS products allow you to tune them for greater accuracy
- When tuning security alerts, attempt to tune to reduce false positives and false negatives
- General tuning steps:
  1. Identify Potential Locations for Sensors
  2. Apply an Initial Configuration
  3. Monitor the Sensor While Tuning
  4. Analyze Alarms, Tune Out False Positives, and Implement Signature Tuning\
  5. Selectively Implement Response Actions:
     - IP logging, TCP resets, shunning (dynamically dropping/not allowing certain connections)
  6. Update sensors with new signatures

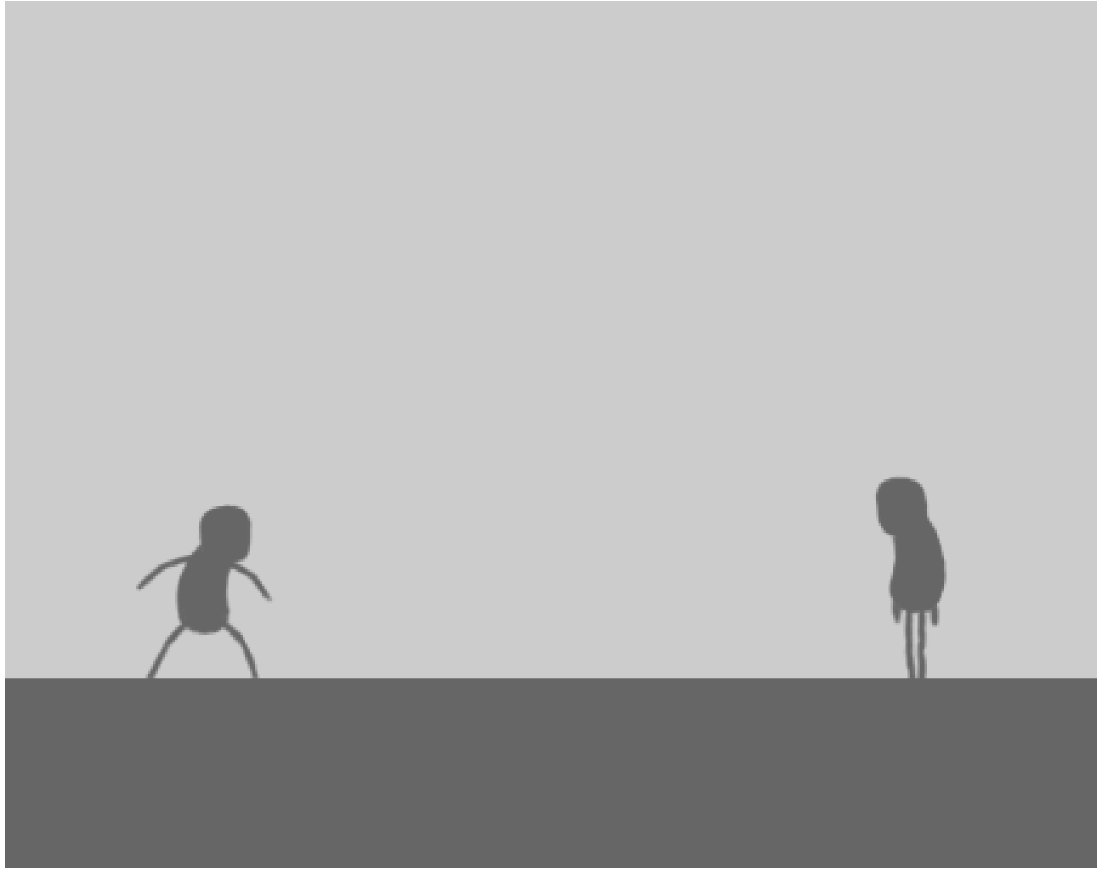
IDS SCENARIO

- A bank stores and processes sensitive privacy information related to home loans
- However, auditing has never been enabled on the system
- What is the first step that the bank should take before enabling the audit feature?
- **You must first determine the impact of enabling the audit feature**

12.5
IDS EVASION

- Evasion Techniques
- Evasion Tools

EVADING IDS AND FIREWALLS



There is no magic bullet for detecting and bypassing a firewalls or IDS system.
What it requires is skill and experience

GENERAL IDS EVASION TECHNIQUES

- Use a proxy/anonymizer to make the attack difficult to trace
- Spoofed source IP, source routing and source port manipulation
  - Make the packets seem to come from a trusted source
- Customize packets so they don't make any signatures
  - Append binary or text data
- IP fragmentation and session splicing
  - Send attack in small packets, making it difficult to determine overall attack signature
- Using character encoding in a URL to obfuscate a destination or intent
- Create confusion by flooding the network with decoys, DoS and false positives

GENERAL IDS EVASION TECHNIQUES (CONT'D)

- Encrypt incoming malicious traffic
  - An inside host will have to be able to decrypt
- Encrypt outgoing exfiltrated stolen data
  - Use a tool such as CryptCat to encrypt stolen data before you exfiltrate it out of the network
- Avoid scan types that an IDS will recognize
  - Stealth scans
  - Other scans with unusual TCP flag combinations
  - Scans that go too fast
  - Scanning hosts in sequential order

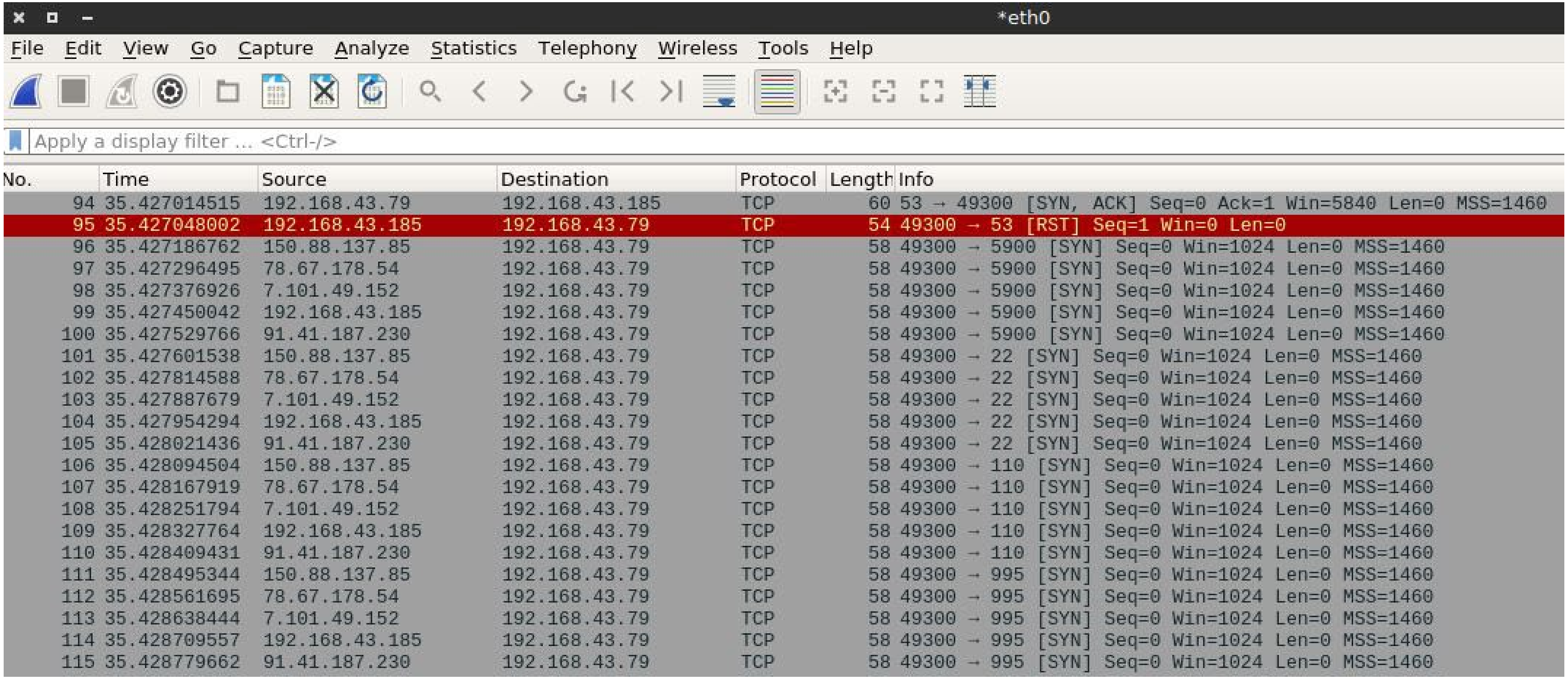
TIMING EVASION

- A very slow scan will just appear as random noise to the IDS
- It will fall below the threshold necessary to fire an alert
- Make sure addresses and ports are targeted in random order
- Scan using nmap’s -T 5 switch
- A SIEM is likely to detect a very slow scan whereas an IDS might not

IP ADDRESS DECOYS

- Generates "noise" you can hide in
- Multiple IP addresses appear to be scanning a target simultaneously
- This makes it very difficult for the IDS or sysadmin to determine who the real attacker is
- You can explicitly specify source addresses or allow the scanner to randomly generate addresses

DECOY EXAMPLE



```
No.   Time         Source          Destination     Protocol Length Info
   94 35.427014515 192.168.43.79   192.168.43.185  TCP      60 53 → 49300 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
   95 35.427048002 192.168.43.185  192.168.43.79   TCP      54 49300 → 53 [RST] Seq=1 Win=0 Len=0
   96 35.427186762 150.88.137.85   192.168.43.79   TCP      58 49300 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   97 35.427296495 78.67.178.54    192.168.43.79   TCP      58 49300 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   98 35.427376926 7.101.49.152    192.168.43.79   TCP      58 49300 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   99 35.427450042 192.168.43.185  192.168.43.79   TCP      58 49300 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  100 35.427529766 91.41.187.230   192.168.43.79   TCP      58 49300 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  101 35.427601538 150.88.137.85   192.168.43.79   TCP      58 49300 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  102 35.427814588 78.67.178.54    192.168.43.79   TCP      58 49300 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  103 35.427887679 7.101.49.152    192.168.43.79   TCP      58 49300 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  104 35.427954294 192.168.43.185  192.168.43.79   TCP      58 49300 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  105 35.428021436 91.41.187.230   192.168.43.79   TCP      58 49300 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  106 35.428094504 150.88.137.85   192.168.43.79   TCP      58 49300 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  107 35.428167919 78.67.178.54    192.168.43.79   TCP      58 49300 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  108 35.428251794 7.101.49.152    192.168.43.79   TCP      58 49300 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  109 35.428327764 192.168.43.185  192.168.43.79   TCP      58 49300 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  110 35.428409431 91.41.187.230   192.168.43.79   TCP      58 49300 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  111 35.428495344 150.88.137.85   192.168.43.79   TCP      58 49300 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  112 35.428561695 78.67.178.54    192.168.43.79   TCP      58 49300 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  113 35.428638444 7.101.49.152    192.168.43.79   TCP      58 49300 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  114 35.428709557 192.168.43.185  192.168.43.79   TCP      58 49300 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  115 35.428779662 91.41.187.230   192.168.43.79   TCP      58 49300 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
```

INSERTION AND OBFUSCATION

- **Insertion Attack**
  - Attacker forces the IDS to process invalid packets
- **Obfuscation**
  - Encoding the attack packets in such a way that the target is able to decode them, but the IDS cannot
    - Unicode - Use Unicode characters rather than ASCII so it doesn’t match any signature
    - Polymorphic code - Change the attack code so it doesn’t match any IDS signature
    - Encryption - Encrypt the attack code so it can’t be read
    - Path manipulation to cause signature mismatch

FALSE POSITIVES AND FRAGMENTS

- **False Positive Generation Events**
  - Craft malicious packets designed to set off alarms
  - Attempt to distract/overwhelm the IDS and admin
- **Overlapping Fragments**
  - Generate a bunch of tiny fragments overlapping TCP sequence numbers.
- **Fragmentation / Session Splicing**
  - The pre-created endpoint must reassemble the packets
  - Use can use Whisker to perform this attack

TCP FLAGS

- **Desynchronization**
  - Manipulate the TCP SYN flag
  - Fool IDS into not paying attention to the sequence numbers of the illegitimate attack traffic
  - Give the IDS a false set of sequences to follow
- **Invalid RST Packets**
  - Manipulate the RST flag to trick the IDS into ignoring the communication session with the target
- **Urgency Flag - URG**
  - Manipulate the URG flag to cause the target and IDS to have different sets of packets
  - The IDS will processes ALL packets irrespective of the URG flag
  - The target will only process URG traffic

PATTERN-MATCHING ATTACKS

- **Polymorphic Shellcode**
  - Blow up the pattern matching by constantly changing the shellcode
- **ASCII Shellcode**
  - Use ASCII characters to bypass pattern matching
- **Application-Level Attacks**
  - Taking advantage of the compression used to transfer large files and hide attacks in compressed data, as it cannot be examined by the IDS.

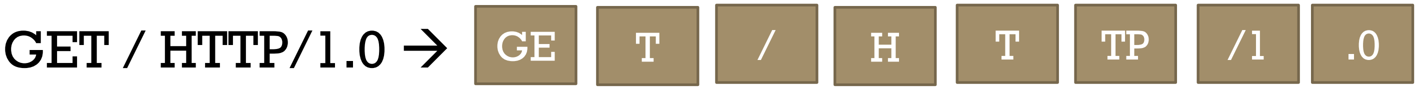
SESSION SPLICING

- Exploits IDSs that do not reconstruct sessions before performing pattern matching
- Fragments the attack across multiple packets
  - No single packet triggers an alert
  - IDS reassembly times out if fragments sit too long it its buffer
- Whisker is a popular tool for session splicing

```
Whisker
      -I 1 IDS-evasive mode 1 (URL encoding)
      -I 2 IDS-evasive mode 2 (/./ directory insertion)
      -I 3 IDS-evasive mode 3 (premature URL ending)
      -I 4 IDS-evasive mode 4 (long URL)
      -I 5 IDS-evasive mode 5 (fake parameter)
      -I 6 IDS-evasive mode 6 (TAB separation)  (not NT/IIS)
      -I 7 IDS-evasive mode 7 (case sensitivity)
      -I 8 IDS-evasive mode 8 (Windows  delimiter)
      -I 9 IDS-evasive mode 9 (session splicing)  (slow)
      -I 0 IDS-evasive mode 0 (NULL method)
```

WHISKER

- Splits an HTTP request across multiple packets
  - Not true IP fragmentation
  - The receiving webserver does not have to reassemble IP fragments
- The target views the attack as a very slow incoming HTTP request
  - Will keep adding the incoming data to its buffer until a complete request has been made
- Example:



GET / HTTP/1.0 → GE | T | / | H | T | TP | /1 | .0

- Whisker will put 1-3 characters in each packet
  - Depending on system and network speed

https://packetstormsecurity.com/files/download/11002/whiskerids.html

https://dl.packetstormsecurity.net/papers/IDS/whiskerids.html

IDS EVASION TOOLS

- Stick
  - An “IDS stress tool”
  - Overwhelms a NIDS with so many alerts using valid signatures
  - The admin can no longer distinguish between false positives and legitimate alerts
  - Produces 250 alarms per second
  - Can cause some IDSes, including Snort, to turn themselves off
- Snot
  - Similar to Stick
  - Attempts to randomize the sequence of rules or alerts generated so that a "Snot generation" rule is not triggered by Snort
  - Example: snot -r snort.rules -s www.somerandomhost.org/24 -d somesnortuser.com -l 10
- Fragroute
  - Packet fragmenter
- Nessus and Nikto
  - Vulnerability scanners with evasion capabilities

IDS EVASION TOOLS (CONT'D)

- sslproxy, TOR
  - Use proxies with encrypted traffic to evade detection
- ADMmutate
  - Creates scripts not recognizable by signature files
- NIDSbench
  - Older tool for fragmenting bits
- Inundator
  - Flooding tool
- IDS-Evasion
  - Multiple bash, PowerShell, and Python scripts to evade Snort
  - https://github.com/ahm3dhany/IDS-Evasion

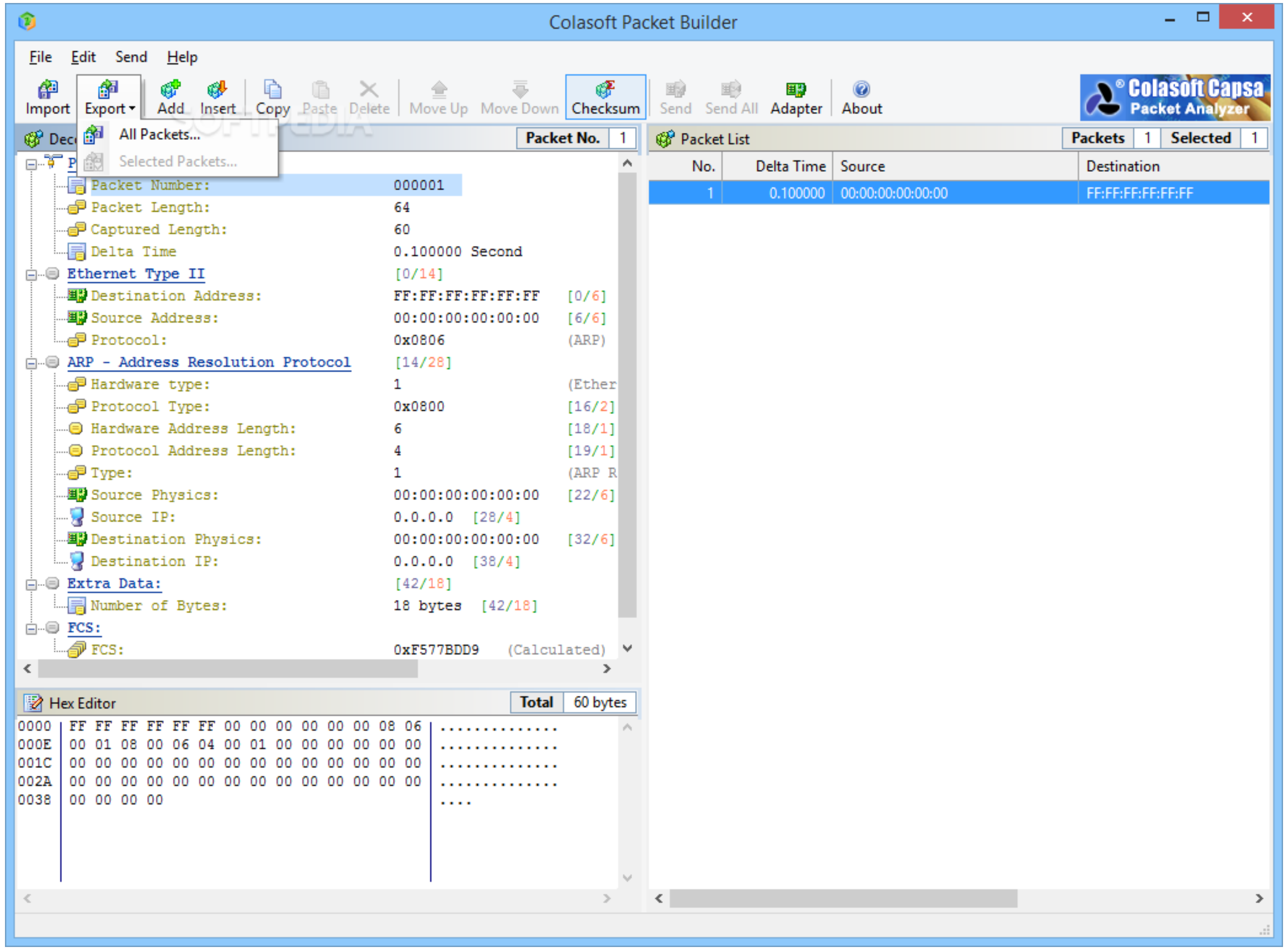
IDS/FIREWALL EVASION TOOLS

- Whisker
- Nmap
- Hping2, Hping3
- CryptCat
- Traffic IQ Professional
- tcp-over-dns
- Snare Agent for Windows
- AckCmd
- Your Freedom
- Tomahawk
- Atelier Web Firewall Tester
- Freenet
- Gtunnel
- Hotspot Shield
- Proifier
- VPN One Click

PACKET FRAGMENT GENERATORS

- Whisker
- Colasoft Packet Builder
- CommView
- hping3
- Multi-Generator (MGEN)
- Net-Inspect
- Ostinato
- fping 3
- NetScanTools Pro
- pktgen
- PACKETH
- Packet Generator

COLASOFT EXAMPLE



```
Packet Number:                 000001
Packet Length:                 64
Captured Length:               60
Delta Time                     0.100000 Second
Ethernet Type II               [0/14]
  Destination Address:         FF:FF:FF:FF:FF:FF   [0/6]
  Source Address:              00:00:00:00:00:00   [6/6]
  Protocol:                    0x0806              (ARP)
ARP - Address Resolution Protocol [14/28]
  Hardware type:               1                   (Ether
  Protocol Type:               0x0800              [16/2]
  Hardware Address Length:     6                   [18/1]
  Protocol Address Length:     4                   [19/1]
  Type:                        1                   (ARP R
  Source Physics:              00:00:00:00:00:00   [22/6]
  Source IP:                   0.0.0.0  [28/4]
  Destination Physics:         00:00:00:00:00:00   [32/6]
  Destination IP:              0.0.0.0  [38/4]
Extra Data:                    [42/18]
  Number of Bytes:             18 bytes  [42/18]
FCS:
  FCS:                         0xF577BDD9   (Calculated)
```

No.	Delta Time	Source	Destination
1	0.100000	00:00:00:00:00:00	FF:FF:FF:FF:FF:FF

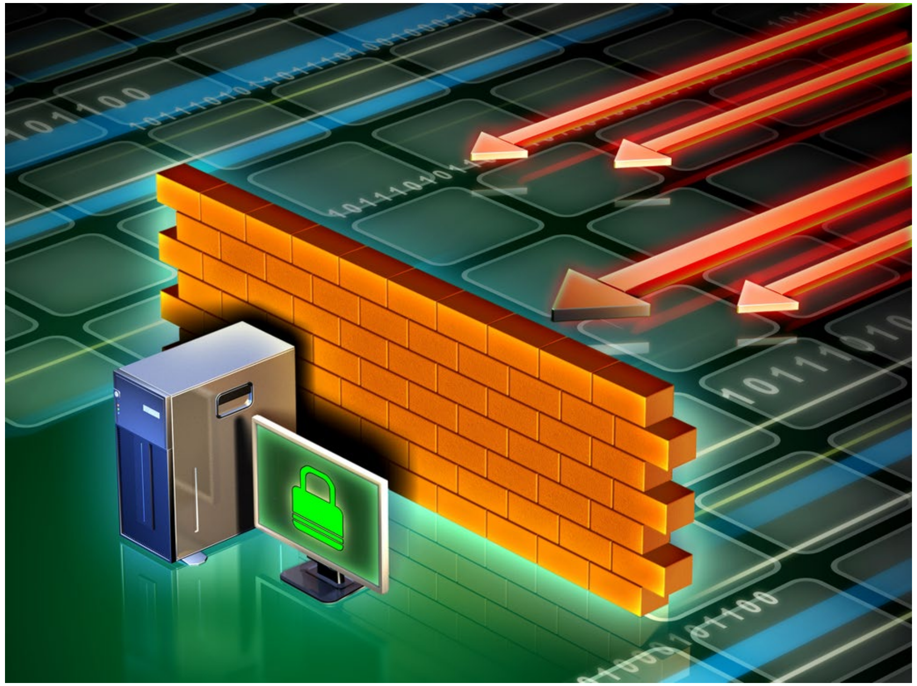
```
0000 | FF FF FF FF FF FF 00 00 00 00 00 00 08 06 | ..............
000E | 00 01 08 00 06 04 00 01 00 00 00 00 00 00 | ..............
001C | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..............
002A | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..............
0038 | 00 00 00 00                               | ....
```

12.6 FIREWALLS

- Overview
- Stateless
- Stateful
- Circuit-level
- Application
- UTM

FIREWALL

- Acts as a network choke point
  - Traffic must flow through it
  - Unauthorized traffic (in or out) is blocked
- Can detect:
  - Unauthorized protocols
  - Unauthorized source and destination IP addresses
  - Unauthorized source and destination ports
  - Unauthorized incoming connection attempts
  - Malicious site URLs
  - Malicious payloads



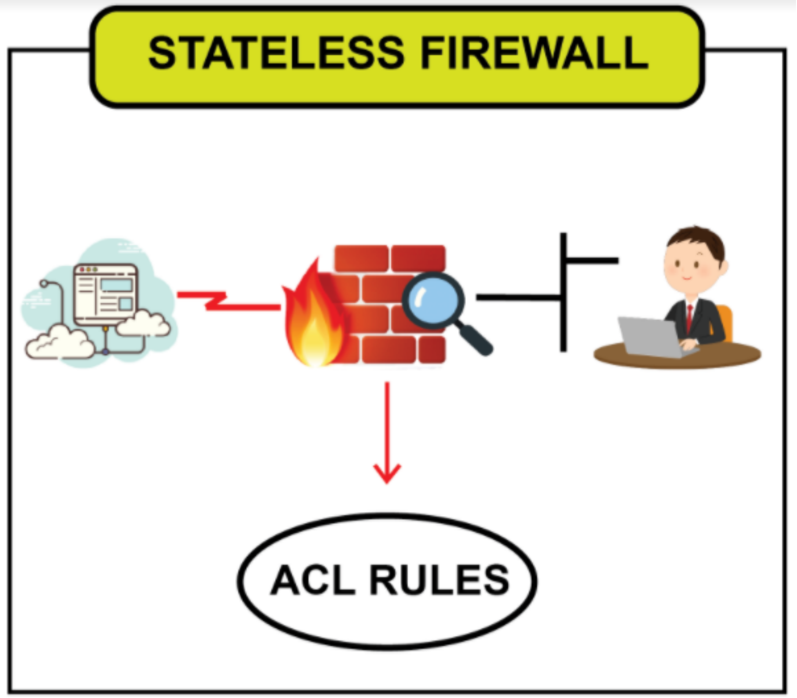
If you can reach a host using one port or protocol but not another, suspect that a firewall is blocking certain traffic types.

FIREWALL TYPES

- Hardware-based
  - AKA firewall appliance
  - Separate device
  - Placed at the network edge, between the “trusted” and “untrusted” networks
  - Blocks unauthorized traffic movement between the networks
- Software-based
  - Installed on a host
  - Prevents unauthorized traffic to/from the host itself

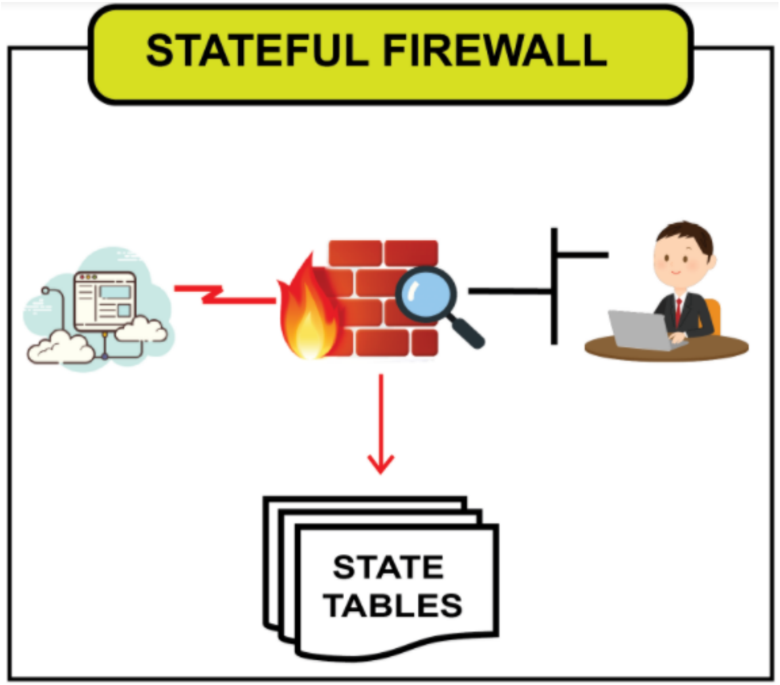
PACKET FILTERING/STATELESS FIREWALL

- Works at multiple OSI layers:
  - Layer 3 – IP addresses
  - Layer 4 – Protocol
  - Layer 5 - Ports
- Can be a stateless firewall or a packet filtering router
- Every packet is compared to a rule set
- Firewall can permit or deny the packet
- Rules may include:
  - IP address of source and/or destination
  - Port number of source and/or destination
  - Protocol (IP, ICMP, IGMP, TCP, UDP)
- There is no memory of the packet before
- You will have to configure rules for every contingency
- Best when high performance is critical



STATEFUL FIREWALL

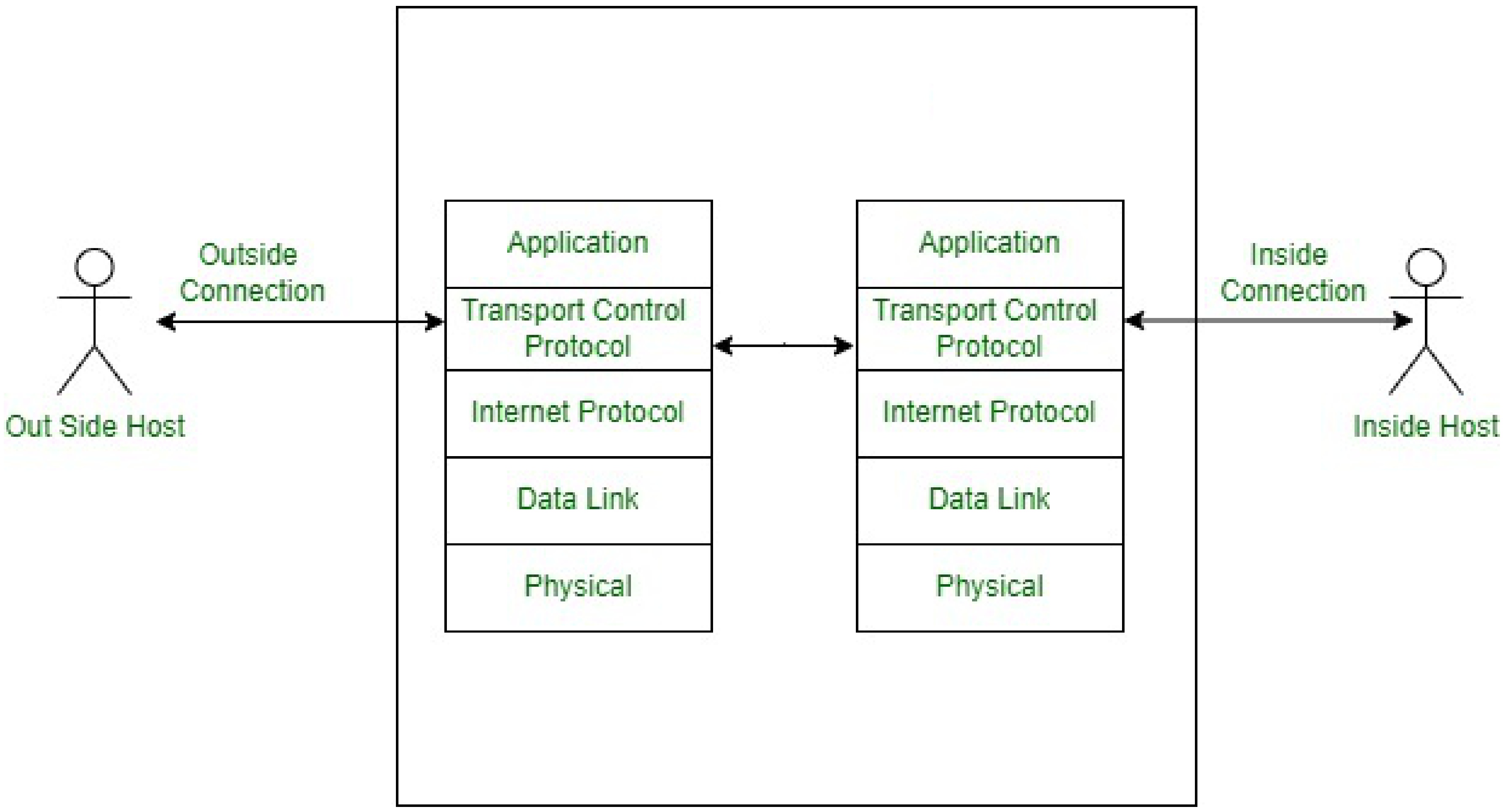
- Maintains a state table for every connection
- Disallows even outbound traffic if suspicious
- Tracks each connection
- Will notice if:
  - There is no proper TCP handshake to start the connection
  - Any port suddenly changes
  - There are any other anomalies in the conversation
- Filters packets at the network and transport layers
- Evaluates packet contents at the application layer
- Most modern firewalls are stateful



CIRCUIT LEVEL GATEWAY

- Works at the Session Layer (Layer 5)
- Allows/disallows entire circuits (connections), as opposed to individual packets
- Validates that TCP or UDP packets belong to an allowed connection
  - Examines TCP handshakes
  - Maintains a session state table
  - Makes IP spoofing more difficult
  - Compensates for UDP lack of source IP validation
- Typically host-based
  - Or a feature of a multi-layer firewall appliance

CIRCUIT LEVEL GATEWAY EXAMPLE



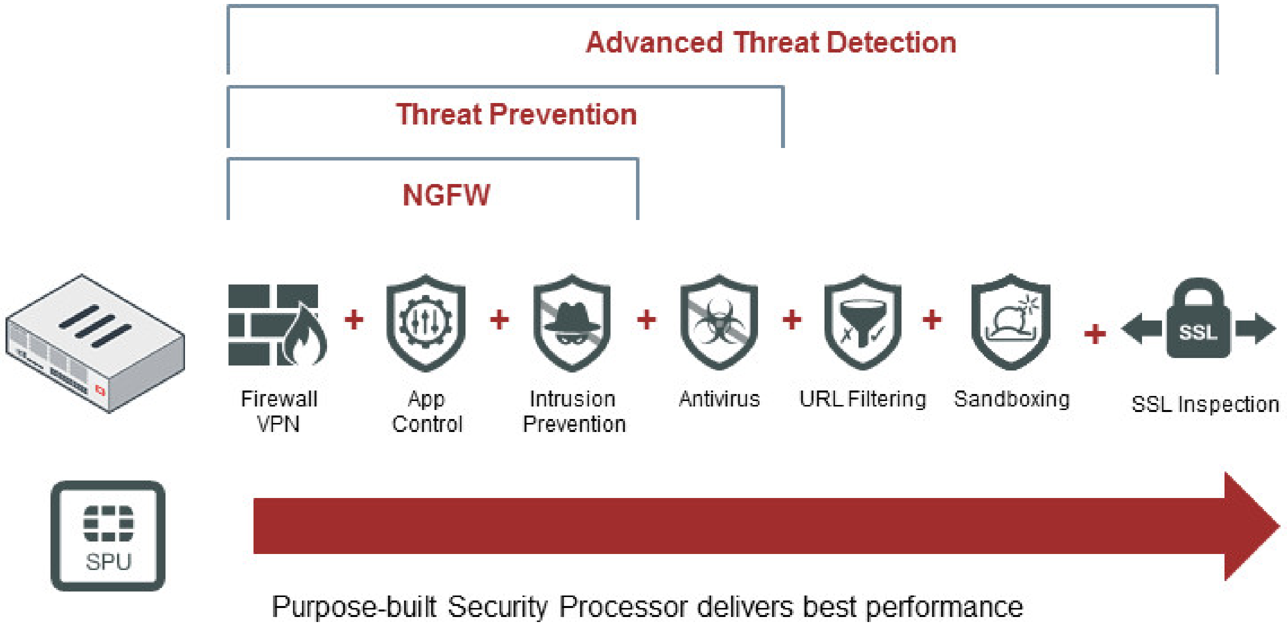
APPLICATION LEVEL GATEWAY

- Filters packets at the Application Layer (7) of OSI or Application layer of TCP/IP
- Examines payloads and Application Layer headers
  - Traffic is examined and filtered on application-specific commands
- If configured as a proxy:
  - Client session put on hold at the proxy
  - Proxy fetches approved content for the client
  - Proxy caches the content against future requests
  - Only protocols supported by the proxy are serviced
    - HTTP, HTTPS, SOCKS4, SOCKS5, and UDP
    - All other protocols are rejected
      - Or routed through packet filtering
- Slowest performance, deepest packet inspection

- SOCKS is a Layer 5 protocol
- Connects client to proxy
- Can forward TCP and UDP
- Optional authentication

UNIFIED THREAT MANAGEMENT (UTM)

- A device that combines multiple functions into a single piece of hardware including:
- Firewall
- Anti-malware
- URL filter
- Spam/phishing filter
- IDS/IPS
- VPN server
- Proxy
- Data Loss Prevention (DLP)



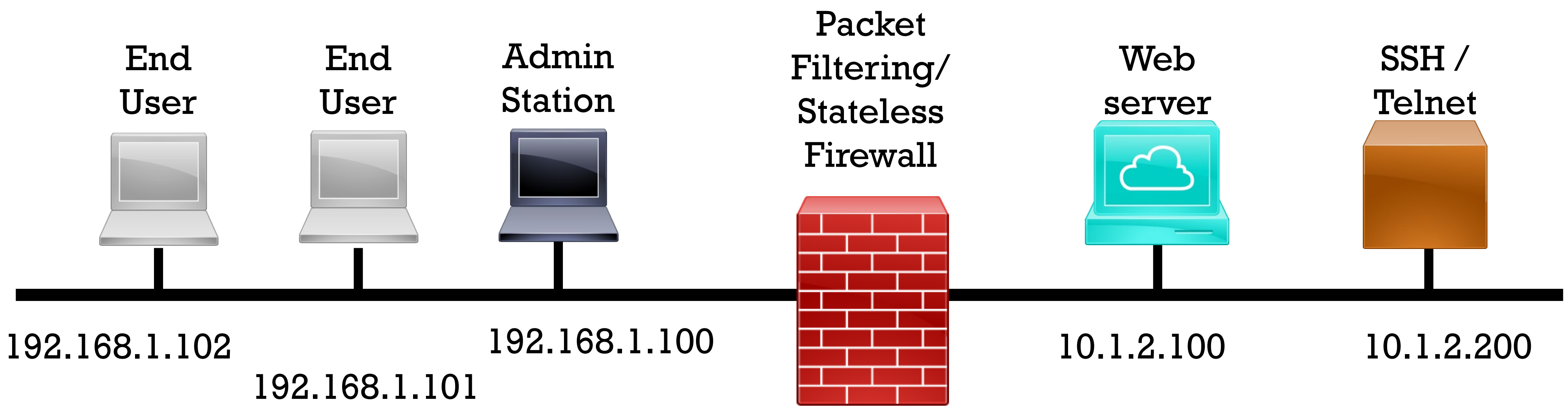
12.7 PACKET FILTERING RULES

- Rules Syntax
- Examples

PACKET FILTERING RULE SYNTAX

- Different products have different rules syntax
- Typical rules elements include:
  - Action
  - Protocol
  - Source IP
  - Source port
  - Destination IP
  - Destination port
  - Connection state
  - Interface
  - Traffic direction (in or out of an interface)

CISCO PACKET FILTERING EXAMPLE



```
access-list 101 deny   tcp any                          any              eq 23
access-list 101 permit tcp 192.168.1.0 0.0.0.255        host 10.1.2.100  eq http
access-list 101 permit tcp host 192.168.1.100           host 10.1.2.200  eq ssh
access-list 101 permit ip  any                          any
```

CISCO PACKET FILTERING RULE EXAMPLES

- Disallow any source from pinging any destination
  - `Deny ICMP any any`
- Disallow any source from 192.168.1.0/24 from querying any DNS server
  - `Deny UDP 192.168.1.0/24 any eq 53`
- Only permit host 10.1.2.3 to use SSL/TLS connect to webserver 172.16.5.4
  - `Permit TCP host 10.1.2.3 172.16.5.4 eq 443`

CISCO PACKET FILTERING RULE EXAMPLES (CONT'D)

- Only permit the admin station 192.168.1.100 to SSH to a Linux server 10.5.5.6
  - `Permit TCP host 192.168.1.100 10.5.5.6 eq 22`
- Only permit hosts from subnet 10.0.0.0/24 to use the client TCP source port 5555 to connect to a gaming server 1.1.1.1 that listens on port 7777
  - `Permit TCP 10.0.0.0 0.0.0.255 eq 5555 host 1.1.1.1 eq 7777`
- Disallow any host sending SNMP packets to 192.168.20.100
  - `Deny UDP any host 192.168.20.100 eq 161`

LINUX IP TABLES RULES EXAMPLE

- **Block a specific IP address**
- `BLOCK_THIS_IP=“192.168.1.2"`
- `iptables -A INPUT -s "$BLOCK_THIS_IP" -j DROP`

- **Allow Incoming SSH only from a Specific Network**
- `iptables -A INPUT -i eth0 -p tcp -s 192.168.100.0/24 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT`
- `iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT`

FIREWALL RULES SCENARIO

- You’ve been asked to review the firewall configuration to ensure that workstations in network 10.10.10.0/24 can only reach the bank web site 10.20.20.1 using https.
- Which rule satisfies the requirement?

- `if (source matches 10.10.10.0/24 and destination matches 10.20.20.1 and port matches 80 or 443) then permit`

- `if (source matches 10.20.20.1 and destination matches 10.10.10.0/24 and port matches 443) then permit`

- `if (source matches 10.10.10.0 and destination matches 10.20.20.1 and port matches 443) then permit`

- `if (source matches 10.10.10.0/24 and destination matches 10.20.20.1 and port matches 443) then permit`

FIREWALL RULES SCENARIO

- You’ve been asked to review the firewall configuration to ensure that workstations in network 10.10.10.0/24 can only reach the bank web site 10.20.20.1 using https.
- Which rule satisfies the requirement?

- `if (source matches 10.10.10.0/24 and destination matches 10.20.20.1 and port matches 80 or 443) then permit`

- `if (source matches 10.20.20.1 and destination matches 10.10.10.0/24 and port matches 443) then permit`

- `if (source matches 10.10.10.0 and destination matches 10.20.20.1 and port matches 443) then permit`

- `if (source matches 10.10.10.0/24 and destination matches 10.20.20.1 and port matches 443) then permit`

12.8 FIREWALL DEPLOYMENTS

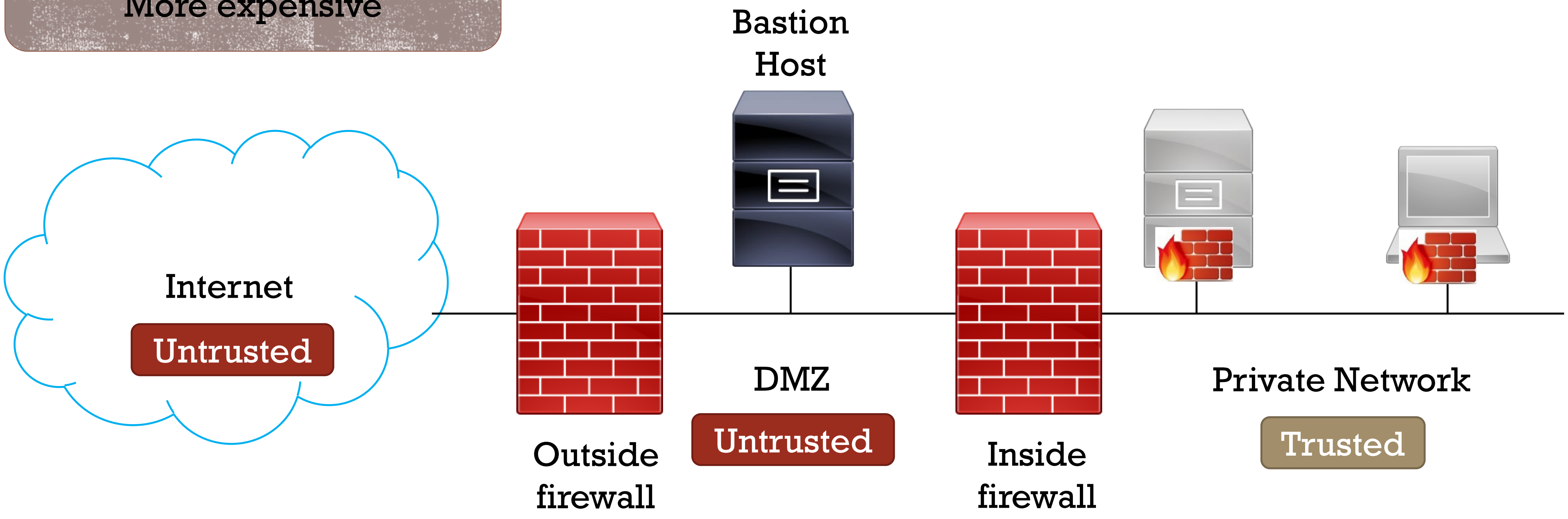
- Architecture
- DMZ Types
- Traffic Flow

FIREWALL ARCHITECTURE

- Bastion Host
  - A public-facing host
  - System that protects network resources from attack
  - Two interfaces: public and private
- Screened Subnet (DMZ)
  - External and Internal firewalls, back-to-back
  - Does not allow access to private zone
- Multi-homed Firewall
  - Firewall with two or more interfaces to further subdivide the network based on security goals
  - Often has a third interface that connects to a DMZ
    - Sometimes called a perimeter network

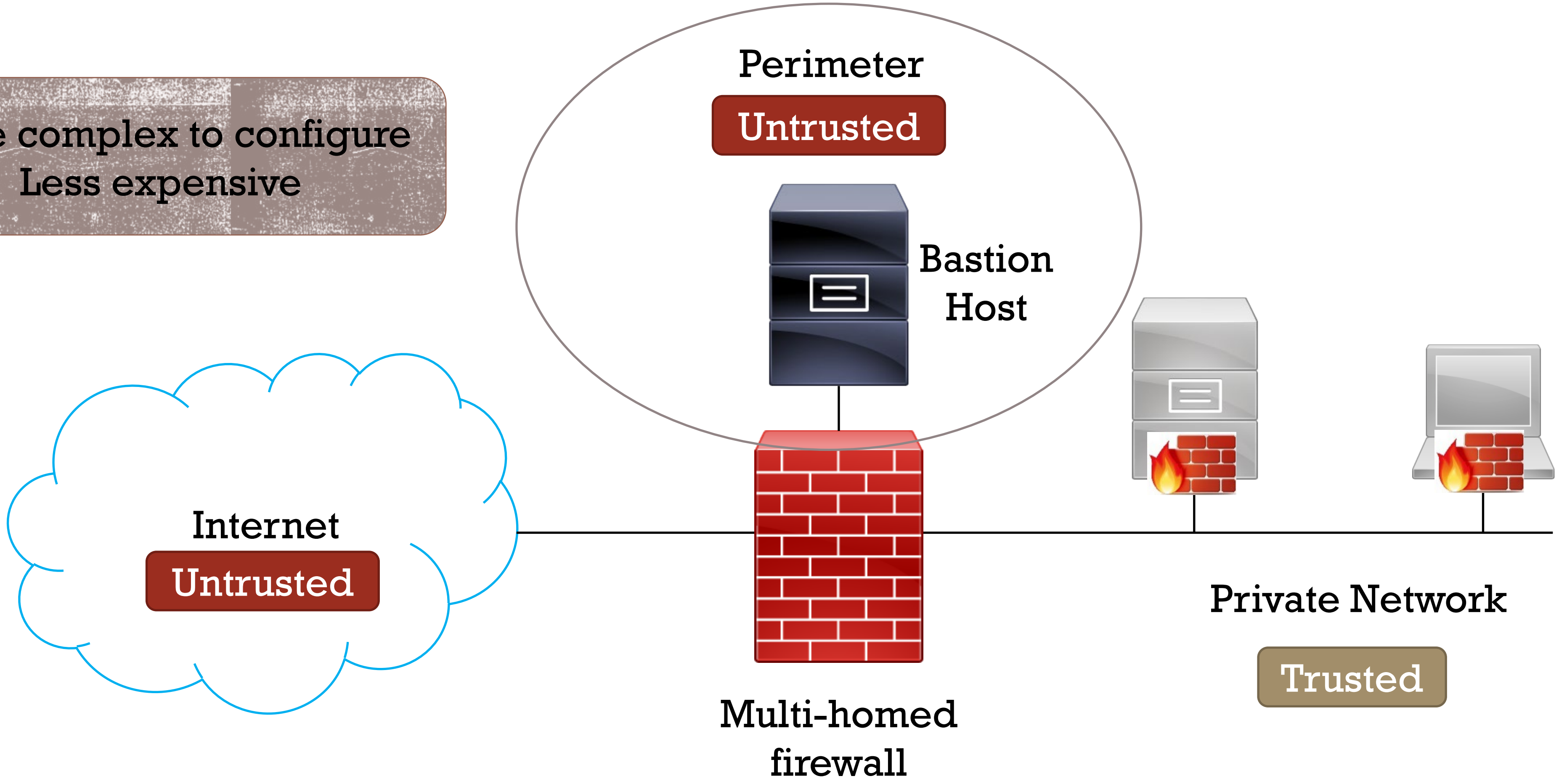
SCREENED SUBNET / DMZ EXAMPLE

Simpler to configure
More expensive



PERIMETER NETWORK / DMZ EXAMPLE

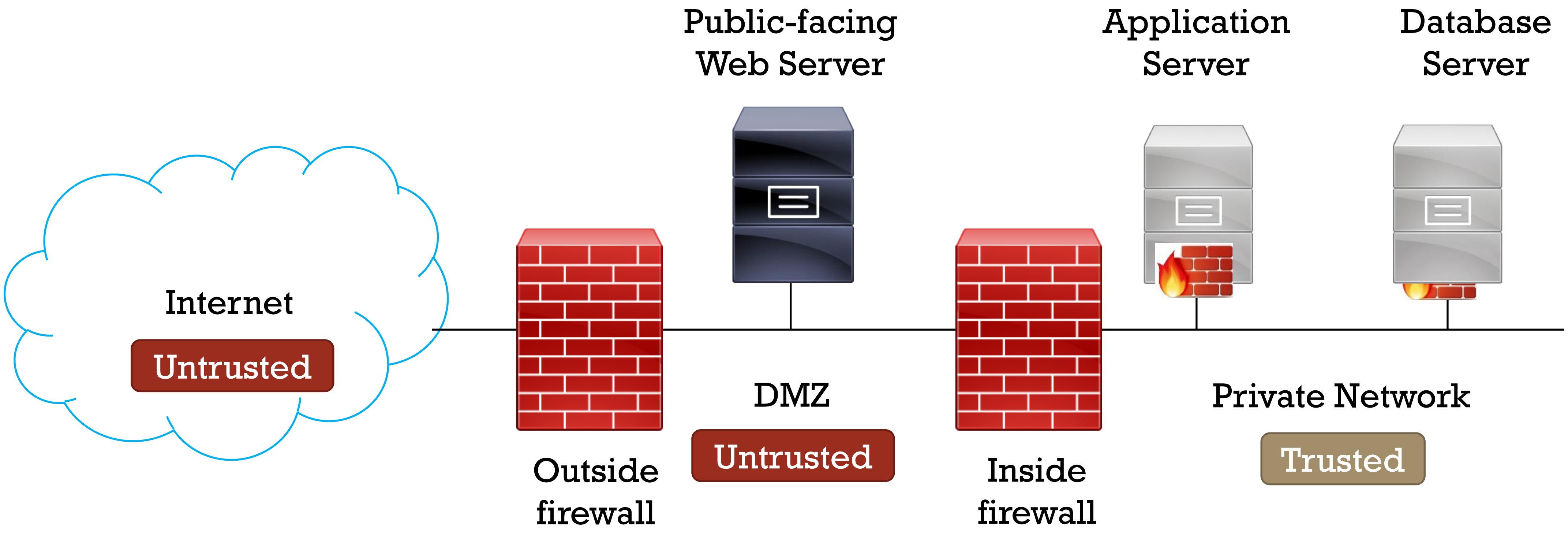
More complex to configure
Less expensive



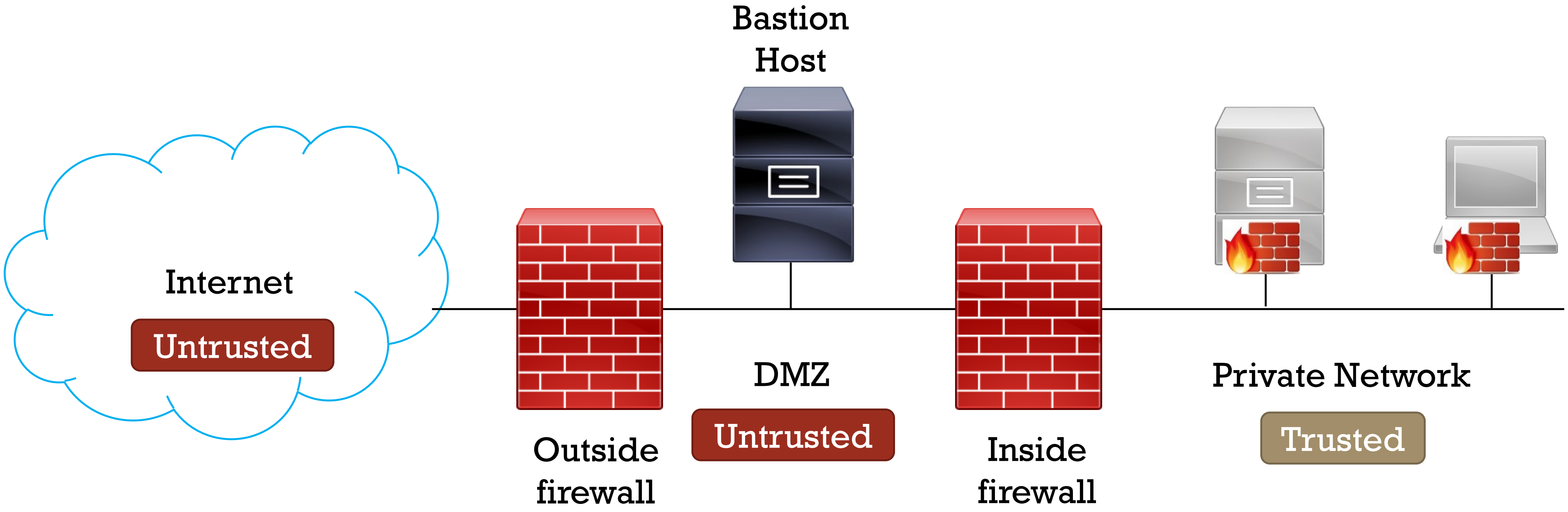
TYPICAL USE OF A DMZ

Let the public-facing bastion host (typically a web server) "take one for the team"

Keep the application and database servers in the private network to protect them

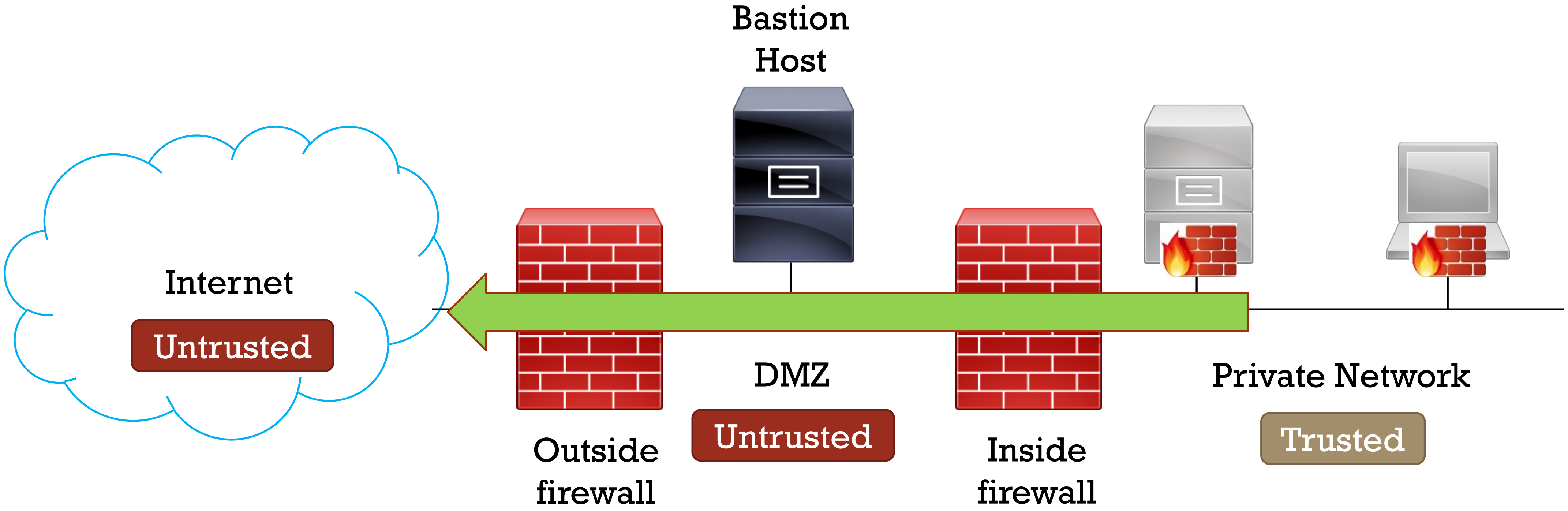


TRAFFIC FLOW THROUGH FIREWALLS



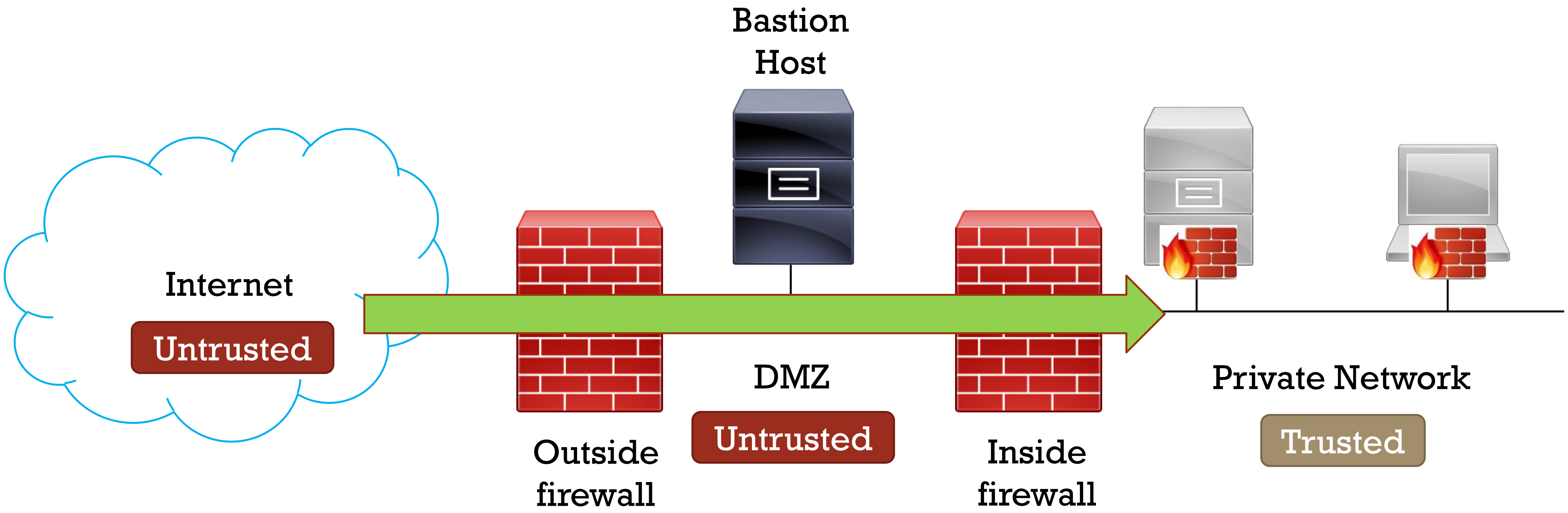
Traffic flow follows the same principles regardless of your firewall configuration

OUTBOUND TRAFFIC FLOW



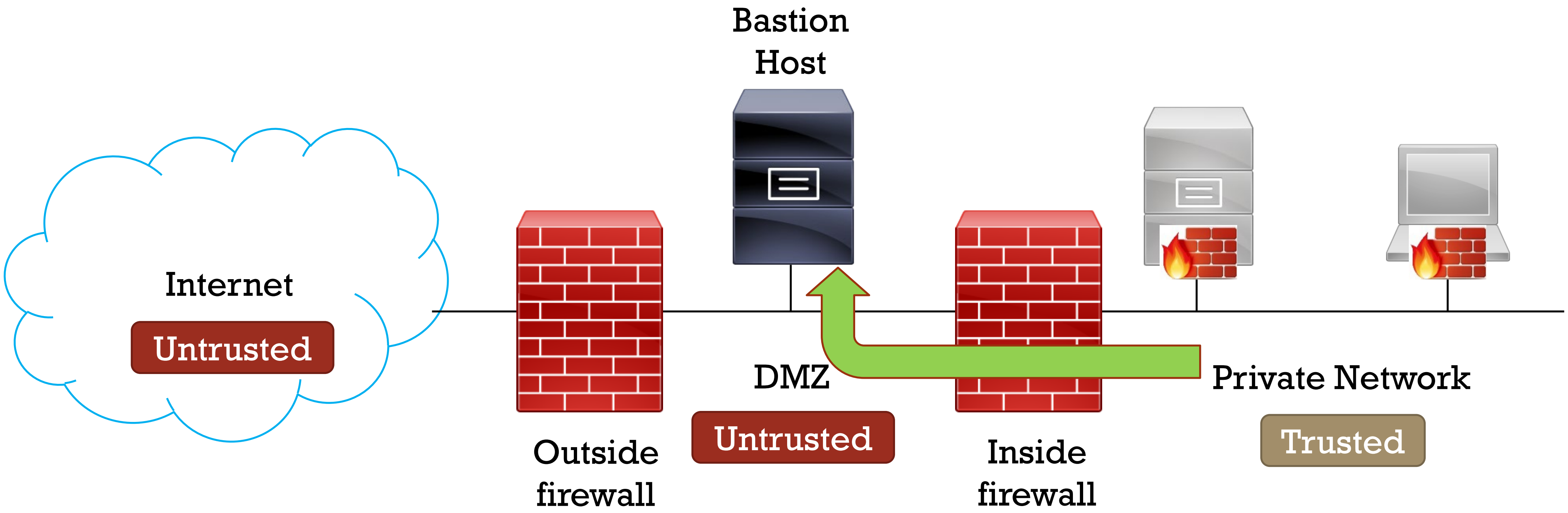
Outbound connection = connection started from the private network

INBOUND REPLIES TRAFFIC FLOW



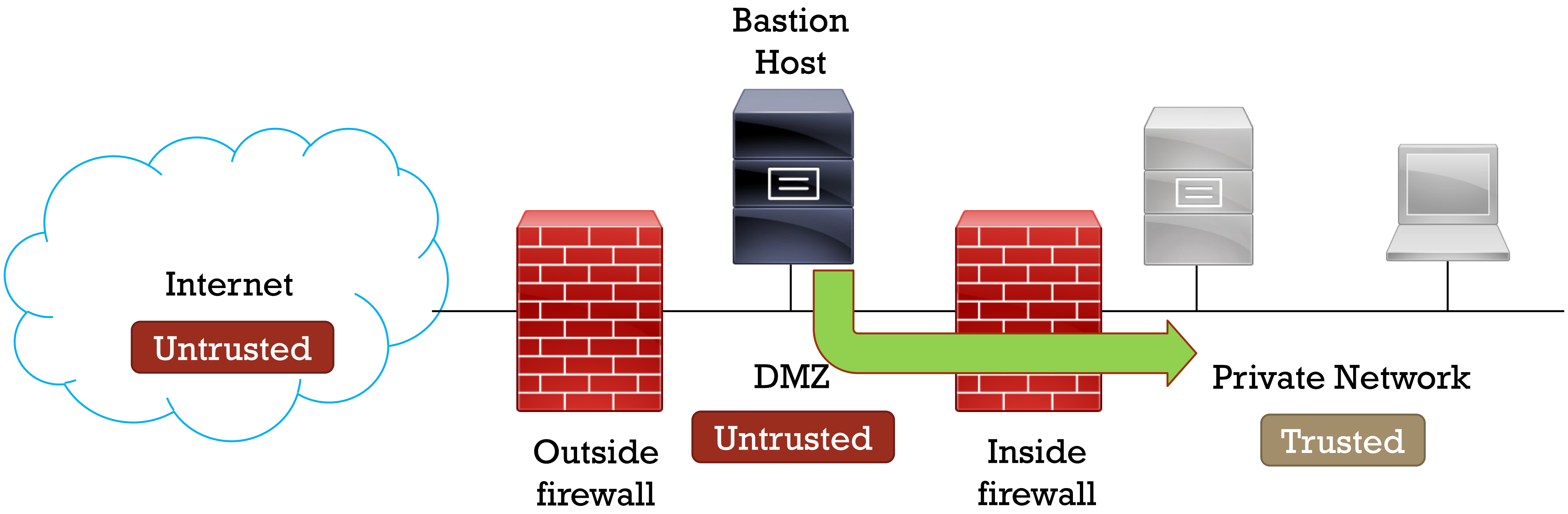
Replies from the untrusted network are permitted
If both firewalls are stateful, they will remember that an internal host started the session

PRIVATE-DMZ TRAFFIC FLOW



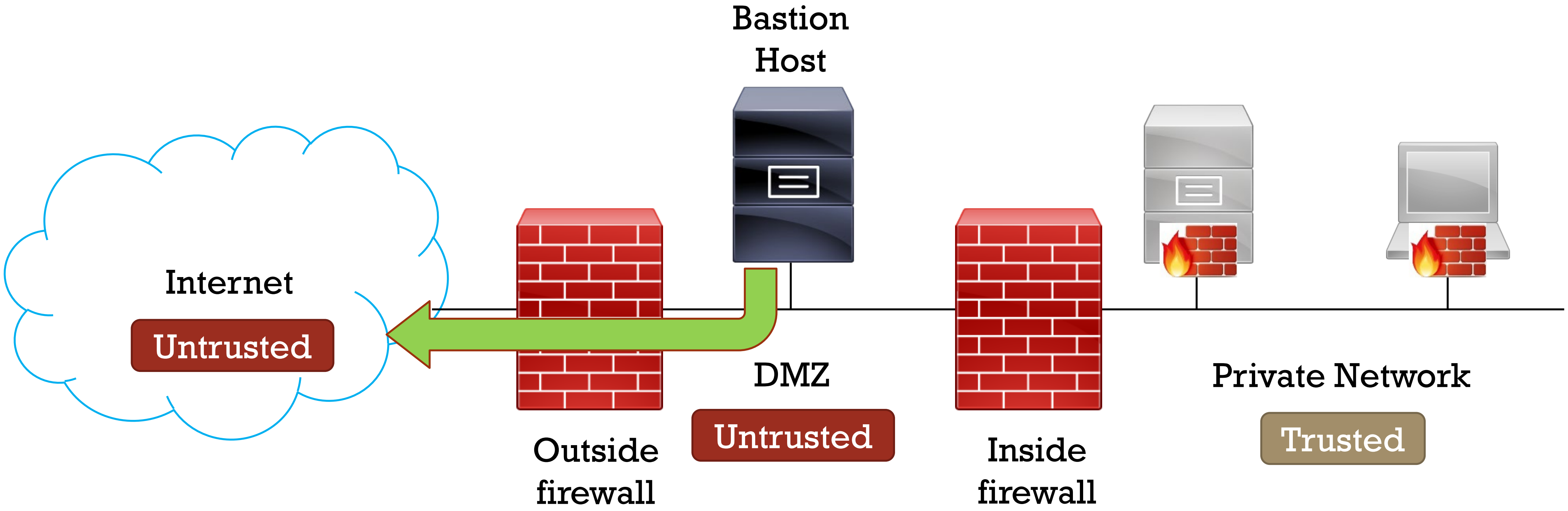
An outbound connection can be from the trusted network to any untrusted network (Internet or DMZ)

PRIVATE-DMZ REPLIES TRAFFIC FLOW



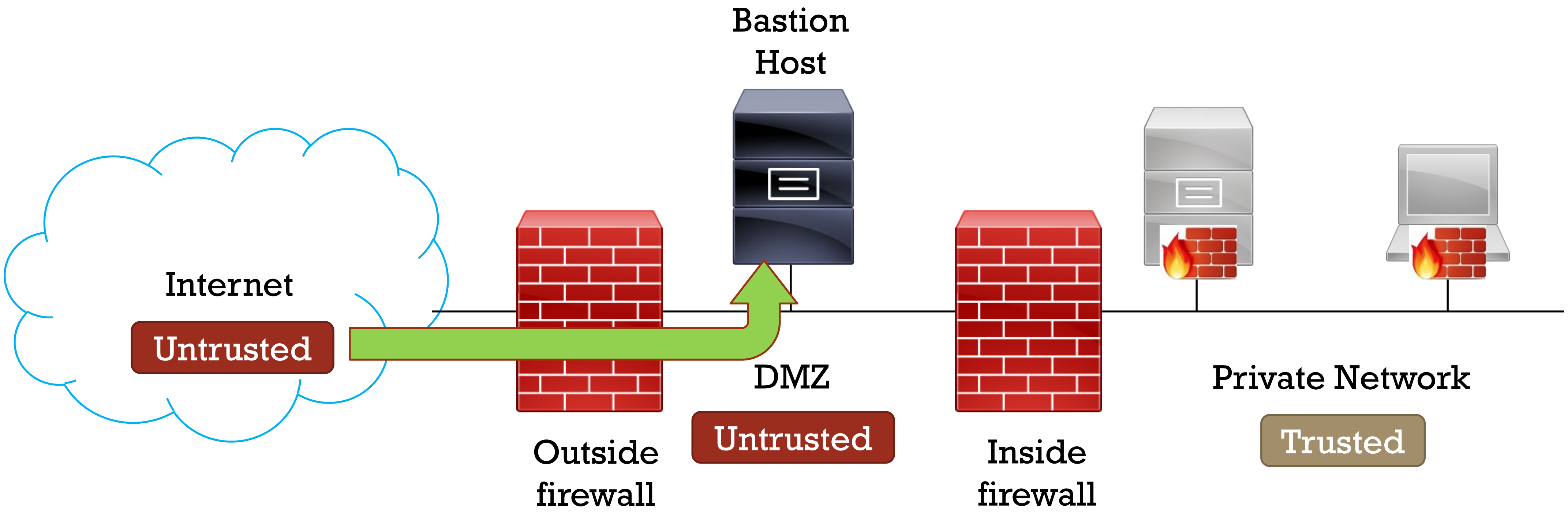
If the trusted network initiates the connection, a response from the untrusted network is permitted back into the trusted network

DMZ-INTERNET TRAFFIC FLOW



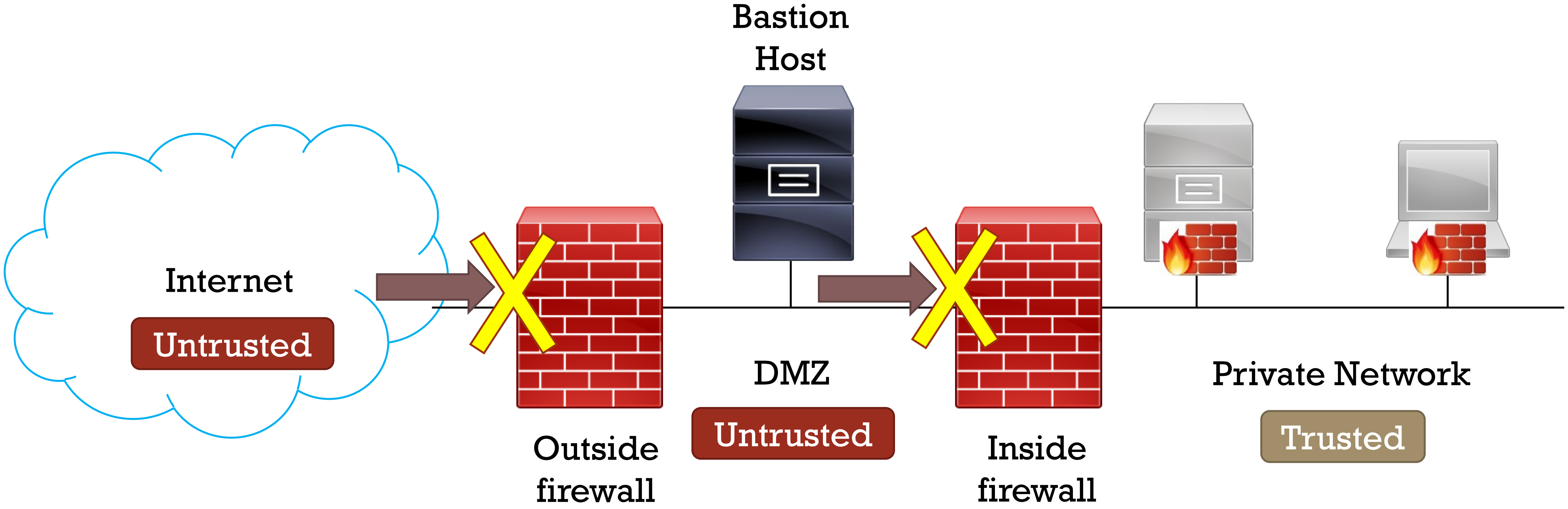
An outbound connection can also be from the DMZ to the Internet
Both networks are considered "untrusted"
The DMZ will have some protections for its bastion hosts

TRAFFIC FLOW



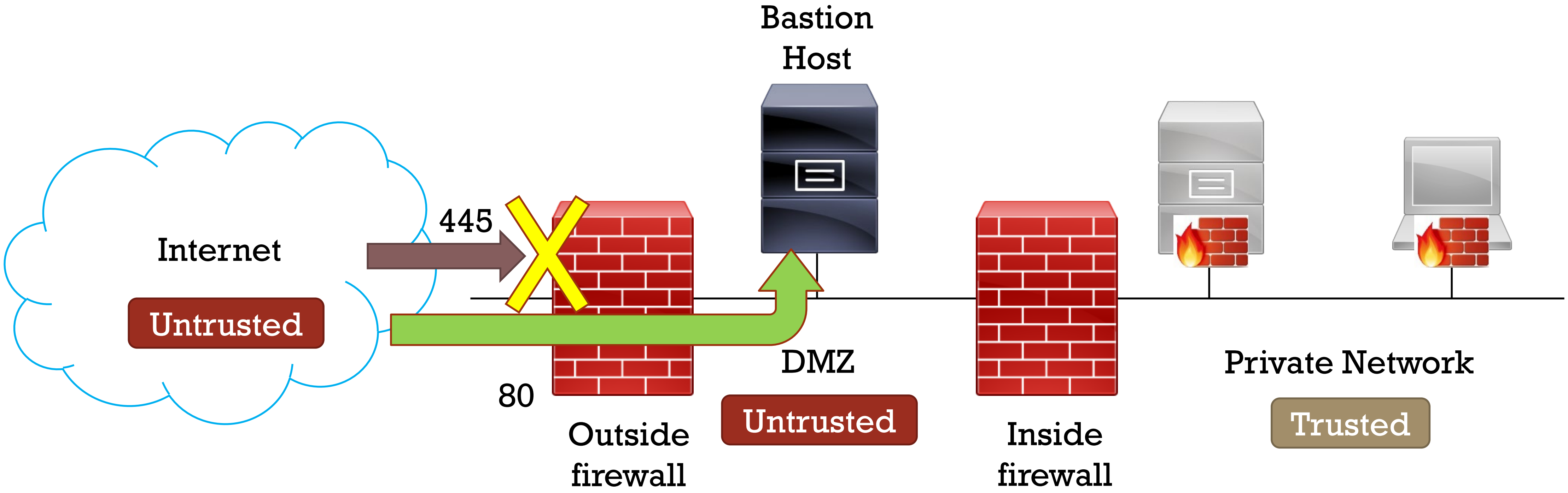
Both a stateful and stateless firewall should be configured to permit responses

INBOUND CONNECTION ATTEMPT



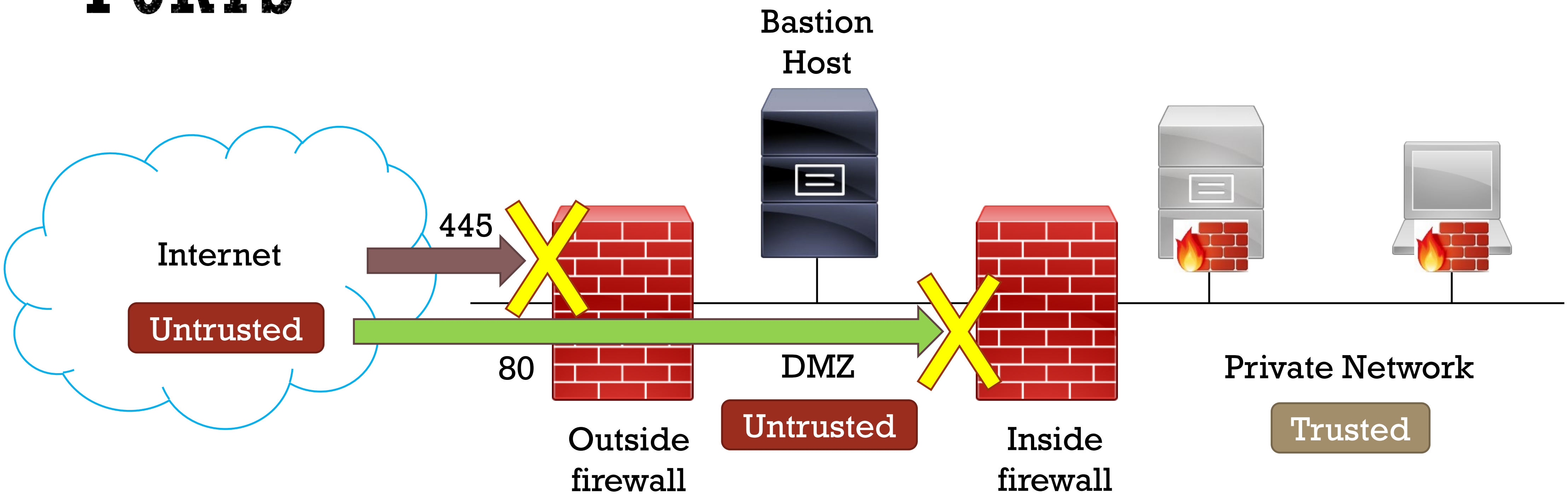
An Inbound connection is one initiated from a (less) trusted network to a (more) trusted network
A stateful firewall will know from its state table that the connection was not started from the inside
A stateless firewall should be configured to not accept any packet with just the TCP SYN flag raised and the ACK flag set to 0

INBOUND CONNECTION TO DMZ



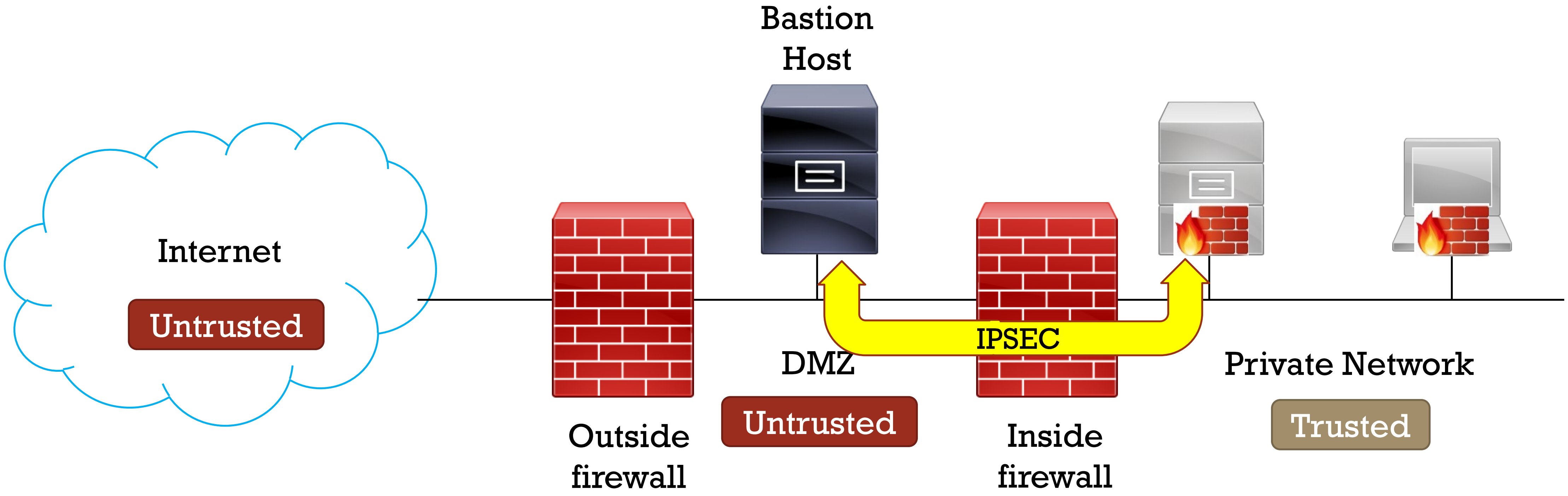
If there is a bastion host offering a service to the Internet, the outside firewall should be configured to permit incoming connections on that port

INBOUND CONNECTIONS TO DIFFERENT PORTS



While the Outside firewall will allow inbound connections to the DMZ
The Inside firewall is typically configured to allow NO inbound connections

TRAFFIC FLOW



The exception is that you might have a host in the DMZ that needs to communicate with a host in the private network. The safest way to allow this is to have an IPSEC VPN between the two. The inside firewall should have very strict rules that only allow the VPN, and only to a specific internal host

NOTE: IPSEC works at Layer 3 and 4. It does not care what the Layer 2 protocol is. Nor does it care what the payload is.

HOSTS THAT MIGHT NEED TO COMMUNICATE BETWEEN DMZ AND PRIVATE NETWORK

- Web server front end - Database server back end
  - You could protect the internal financial database with a web server front end in the DMZ
- Email spam filter - Email server
- Webmail front end - Mailbox server back end

12.9 SPLIT DNS

- Public DNS
- Private DNS
- Split DNS Example

SPLIT DNS

- You manage two separate DNS servers:
  - External (public) DNS
  - Internal (private) DNS
- They should be SEPARATELY managed with NO communication between the servers
  - You will need to separately configure records for both
  - It is ok for both to have the same domain name
  - Internal hosts should be configured to ONLY use the internal DNS server

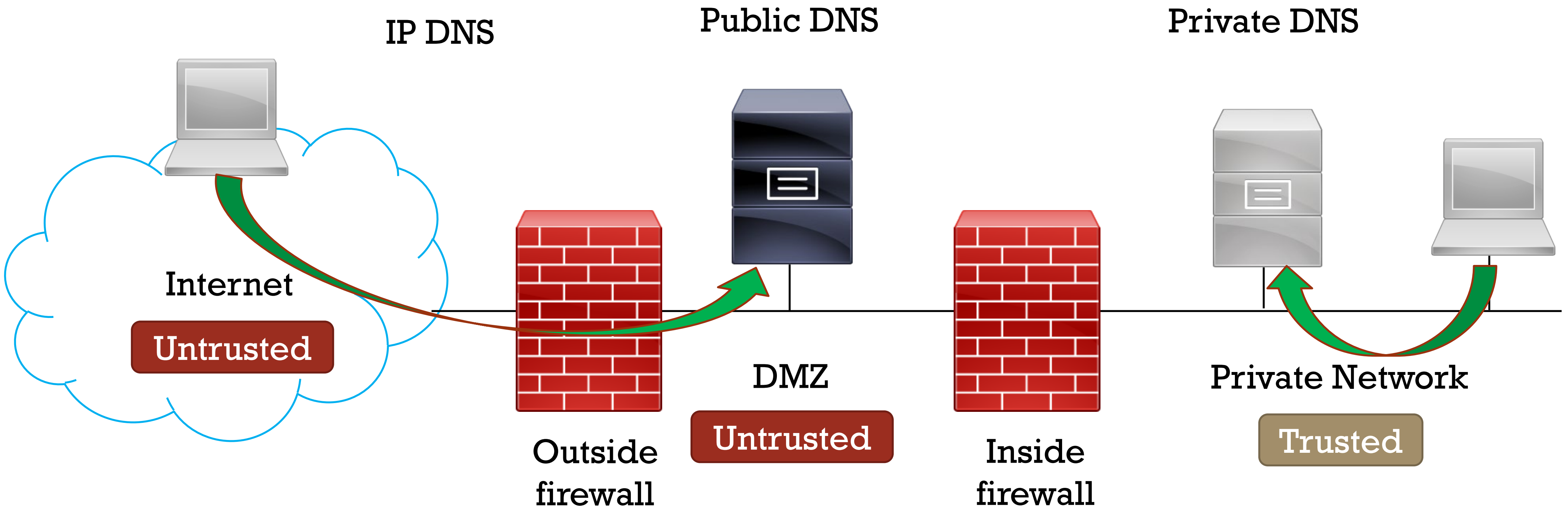
SPLIT DNS - PUBLIC DNS SERVER

- Should be in the DMZ or hosted by a provider
- Public-facing services such as public website, spam filter/email relay, VPN server
- Only has records the general public will need access to
- TLD and parent DNS zones should delegate (point) down to your DNS on the Internet
- Should not have to perform any non-authoritative lookups
- Should not have to query other DNS servers for anything

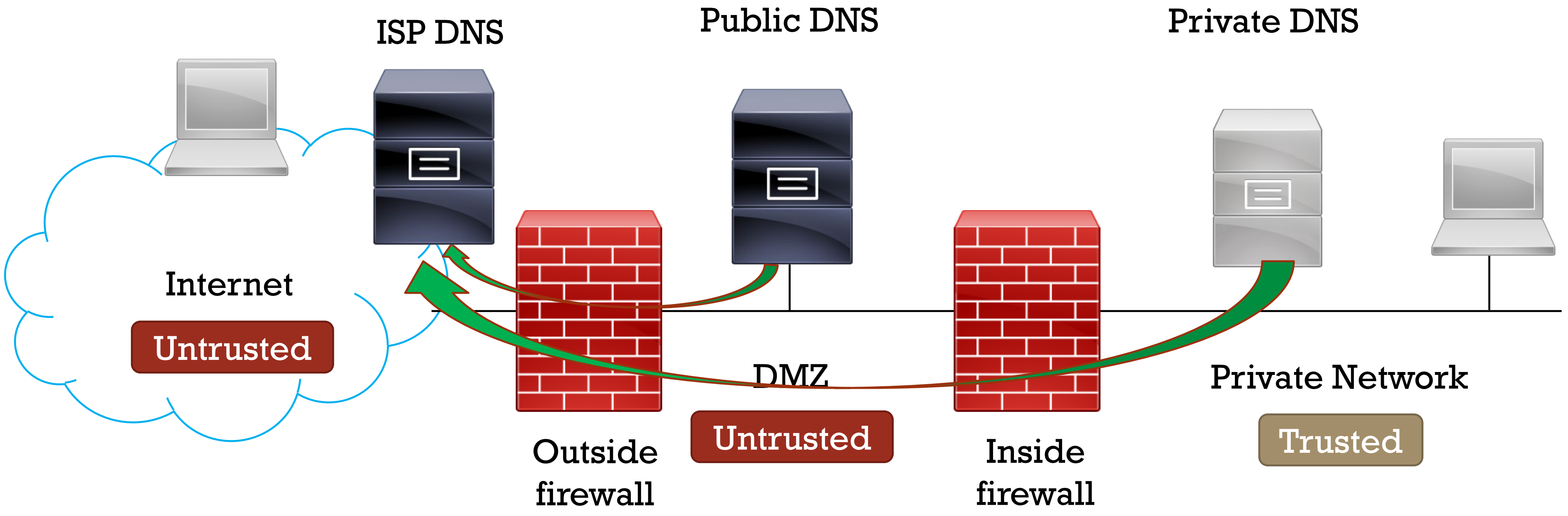
SPLIT DNS - PRIVATE DNS SERVER

- Should be in the private network
- Has records that internal clients will need access to:
  - Active Directory
  - Internal resources
- Should include manual entries for public services in the DMZ
- Should be able to perform recursive queries or search the Internet DNS tree for clients needing public records
- Configure all internal clients to use the private DNS only
- Have the private DNS go directly to an ISP DNS to do Internet name searches

SPLIT DNS EXAMPLE



SPLIT DNS EXAMPLE (CONT'D)



12.10 FIREWALL PRODUCT TYPES

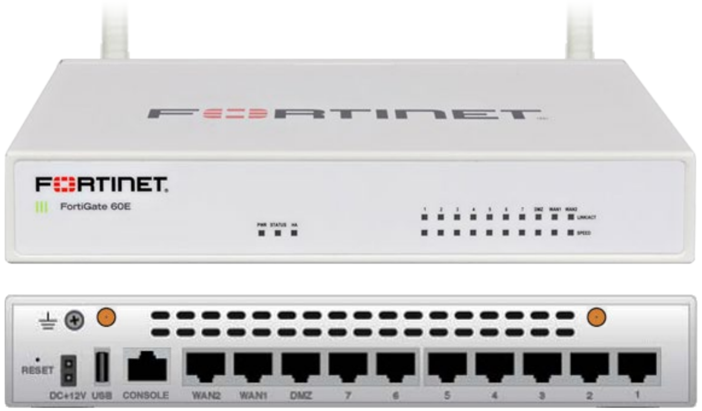
- Host-based
- Appliances
- FWaaS

FIREWALL EXAMPLES

- Comodo
- Cisco ASA
- Check Point
- Untangle NG Firewall
- Sonicwall
- Online Armor
- FortiGate
- ManageEngine
- Perimeter 81
- Total AV

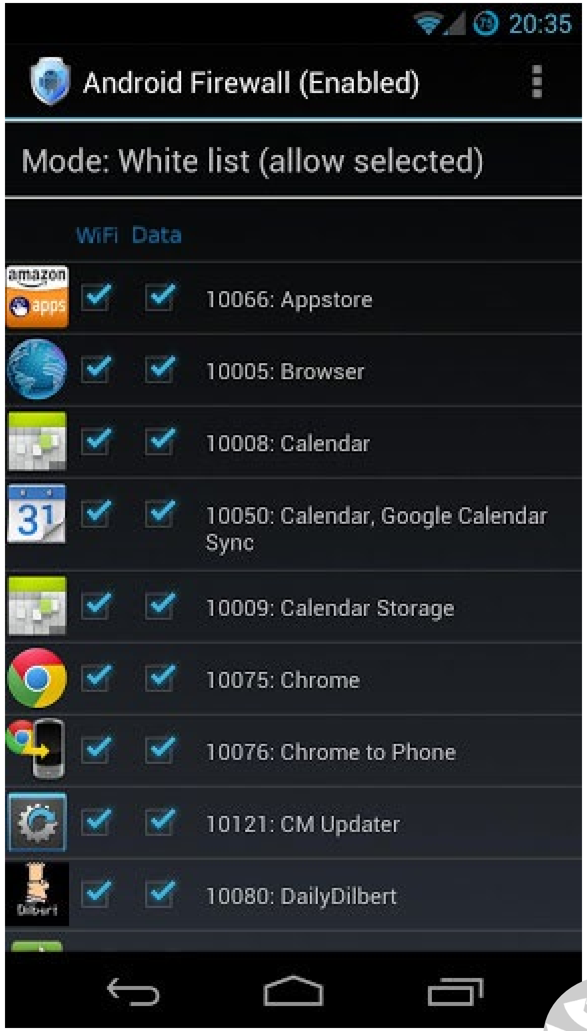
- VaultCore
- PC Protect
- Bitdefender
- McAfee
- ZoneAlarm PRO
- Windows Defender
- Linux iptables
- Linux UFW
- Cisco packet filtering router
  - Outside firewall to create a "Dirty" DMZ

HARDWARE FIREWALL EXAMPLES



FIREWALLS FOR MOBILE

- Android Firewall
- Firewall iP
- Mobiwol: NoRoot Firewall
- DroidWall
- AFWall+
- Firewall Plus
- Root Firewall
- Android Firewall Gold
- Droid Firewall
- Privacy Shield
- aFirewall
- NoRoot Firewall



20:35

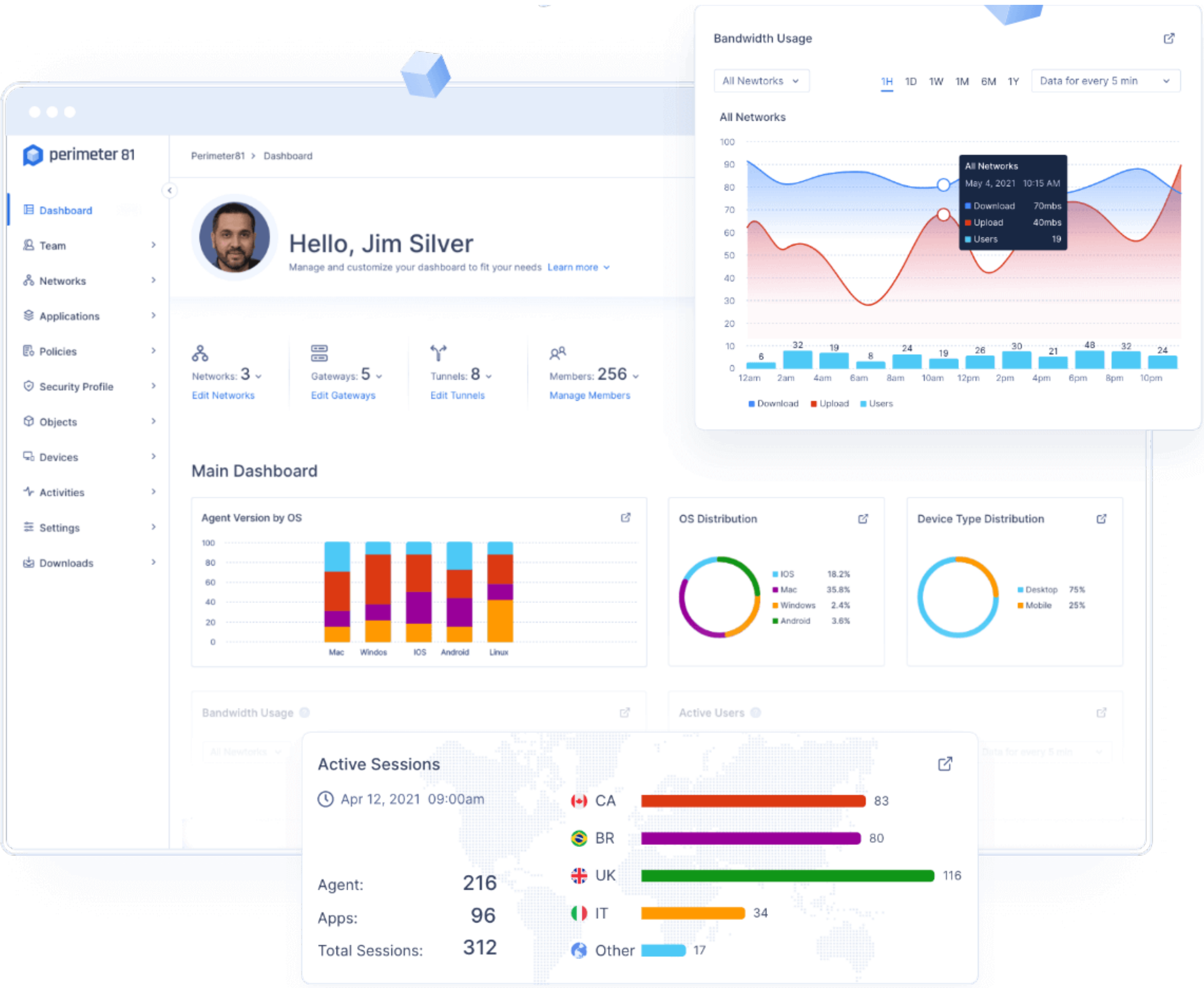
Android Firewall (Enabled)

Mode: White list (allow selected)

WiFi	Data	
☑	☑	10066: Appstore
☑	☑	10005: Browser
☑	☑	10008: Calendar
☑	☑	10050: Calendar, Google Calendar Sync
☑	☑	10009: Calendar Storage
☑	☑	10075: Chrome
☑	☑	10076: Chrome to Phone
☑	☑	10121: CM Updater
☑	☑	10080: DailyDilbert

CLOUD-BASED IDS AND FIREWALL SERVICES

- IDSaaS
  - Google Cloud IDS
  - AlienVault
  - Checkpoint
- FWaaS
  - Perimeter 81
  - Fortinet
  - Zscaler



Hello, Jim Silver

Manage and customize your dashboard to fit your needs Learn more

Networks: 3 · Gateways: 5 · Tunnels: 8 · Members: 256

All Networks — May 4, 2021 10:15 AM — Download 70mbs — Upload 40mbs — Users 19

IOS 18.2% · Mac 35.8% · Windows 2.4% · Android 3.6%

Desktop 75% · Mobile 25%

Apr 12, 2021 09:00am

Agent: 216

Apps: 96

Total Sessions: 312

CA 83 · BR 80 · UK 116 · IT 34 · Other 17