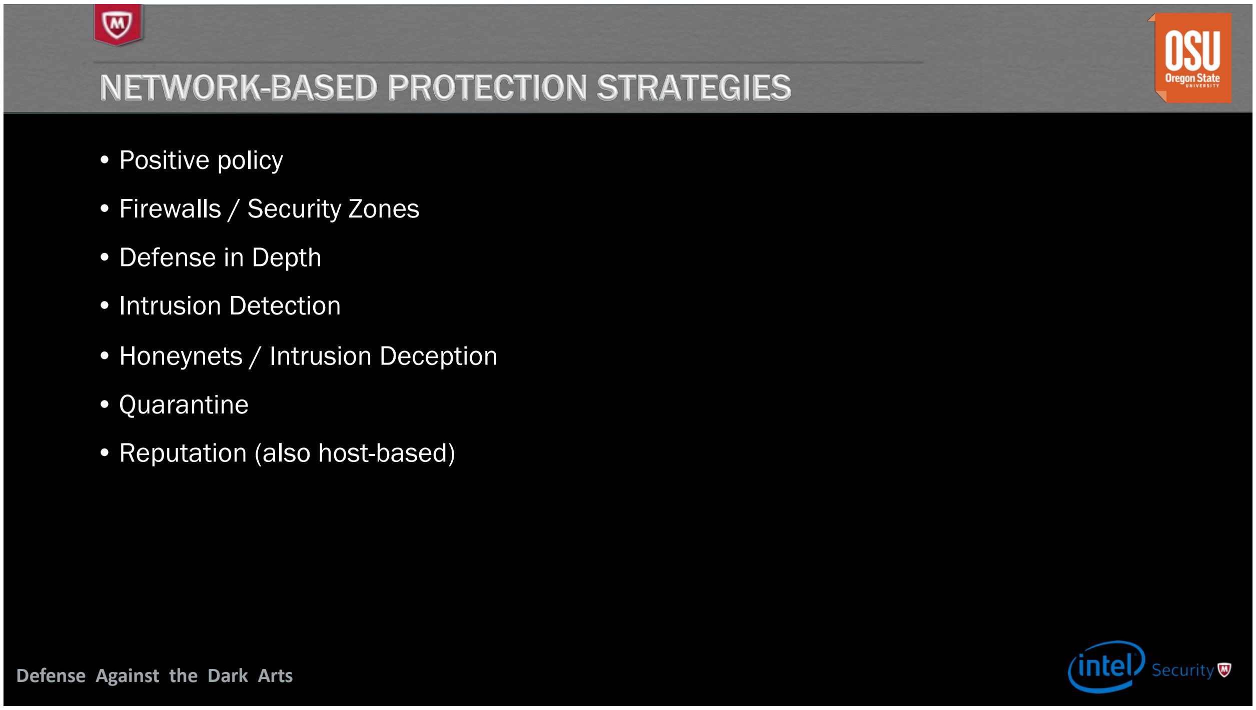
Johnny Zhong
CS373 – Defense Against The Dark Arts
Week 6 – Network Security Technologies

Network Security Purposes

- Prevention
- Critical data flowing in/out of network
- Prevention of DDOS
- Dealing with threats to network proper:
    - Turn off
    - Disable actions for

"Be liberal in what you accept and be conservative in what you send."



NETWORK-BASED PROTECTION STRATEGIES

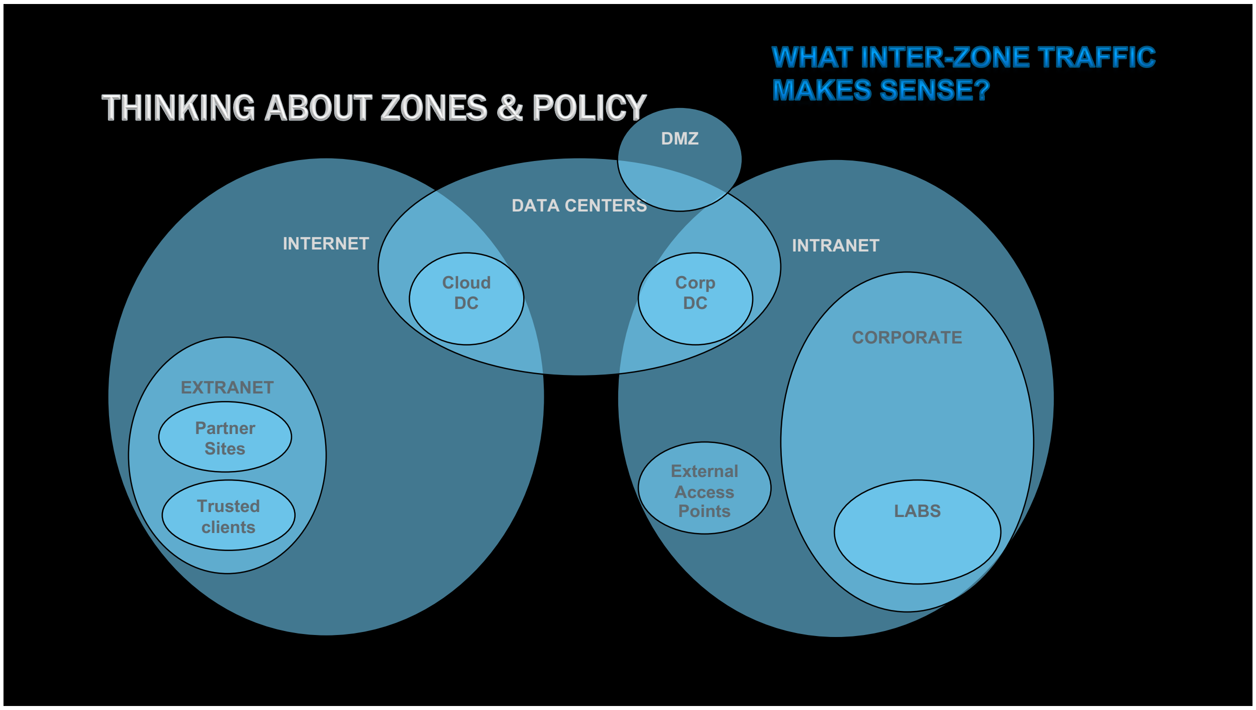
- Positive policy
- Firewalls / Security Zones
- Defense in Depth
- Intrusion Detection
- Honeynets / Intrusion Deception
- Quarantine
- Reputation (also host-based)

Defense Against the Dark Arts

Protection Strategies

- Positive Policy/whitelisting - Allow only what you're expecting to have happen
- Firewall
    - allows the defender to have the advantage
    - Attack surface – set of operations the network is willing to respond to
    - Many kinds of firewalls
        - Web gateway – acts as a proxy (any traffic bound for the outside world goes through this device), can be more selective than traditional firewalls
        - Email gateway – smtp – filters mail
- Defense in Depth – defense method that assumes that outer security measures will fail. – castle analogy

- Intrusion Detection/Protection System
    - Blacklist method as opposed to whitelist method
    - Fails with unknown attacks (zero day attack)
    - False positives
    - YARA is a method of detection for known threats
- Honeynet
    - Leaving a vulnerable system out to be attacked
    - Requires specific information for deployment



THINKING ABOUT ZONES & POLICY

WHAT INTER-ZONE TRAFFIC MAKES SENSE?

- Quarantine
    - Put the threat in a part of a network which does not allow the threat to access anything else
    - Allows us to analyze behavior
- Reputation
    - Security Certificates
    - Past behavior or aggregated score
    - Big data – a point where the amount of data changes the value of the data
- NextGen Firewall
    - allows machines to look into packets to determine the safety of the packet.
    - Allows for app identification

Threats

- MTM – Man in the Middle
    - Can be used for good and evil
    - Intercepts traffic on network
    - TCP Hijacking

        - We don’t protect against MITM
        - Rewrite packet, change checksum
        - Might be used for security reasons – removal of malware, change of outgoing file
    - Terminating Proxy – change connection
    - Remove meta characters from obfuscated URLs
    - SSL MITM – encrypting your data – firewalls
    - Detection of MITM – HMAC – large number of bits turn into a small number of bits.
        - Create HMAC off of the base data being sent
        - Attacker can double up the packet, causing issues
        - But can chain HMACs in order to prevent this from happening
    - N-squared problem solution – public key cryptography
        - Public key to lock
        - Private key to unlock
        - Many mechanisms to share data, through SSL
    - Dependency of each layer on each other introduces a vulnerability (hierarchy of trust)
    - Change to trusted certificates
    - Precise set of guarantees, not full guarantee

Lecture 2

Lab Notes:
Class A network – 8 bit
Class B network – 16 bit
Class C network – 24 bit subnet mask

Threat Recon

Why? – want to know where you want to deliver a payload

Active and Passive Recon

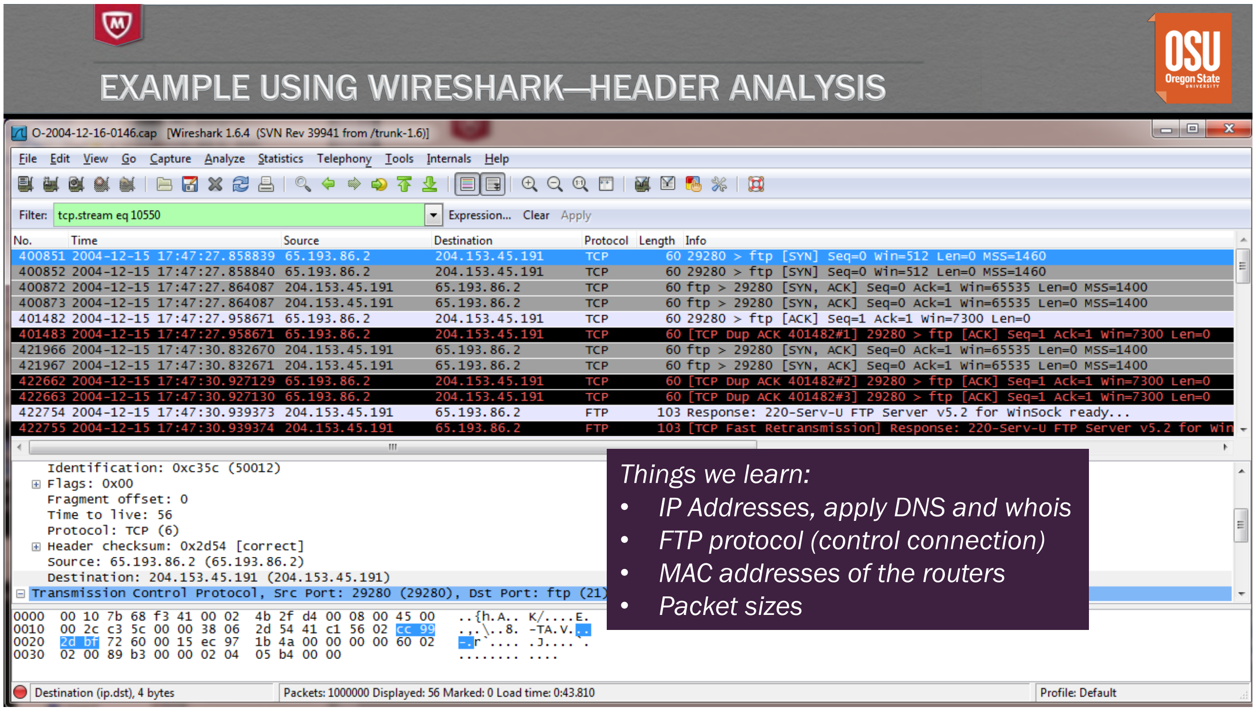
Passive Recon

- listening in to what’s going on
- via switches, routers

Active Recon

- nmap
- ping

Wireshark Demo


OSU
Oregon State University
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply

EXAMPLE USING WIRESHARK—HEADER ANALYSIS

O-2004-12-16-0146.cap [Wireshark 1.6.4 (SVN Rev 39941 from /trunk-1.6)]

Filter: tcp.stream eq 10550

```
No.    Time                       Source          Destination     Protocol Length Info
400851 2004-12-15 17:47:27.858839 65.193.86.2     204.153.45.191  TCP       60 29280 > ftp [SYN] Seq=0 Win=512 Len=0 MSS=1460
400852 2004-12-15 17:47:27.858840 65.193.86.2     204.153.45.191  TCP       60 29280 > ftp [SYN] Seq=0 Win=512 Len=0 MSS=1460
400872 2004-12-15 17:47:27.864087 204.153.45.191  65.193.86.2     TCP       60 ftp > 29280 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400
400873 2004-12-15 17:47:27.864087 204.153.45.191  65.193.86.2     TCP       60 ftp > 29280 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400
401482 2004-12-15 17:47:27.958671 65.193.86.2     204.153.45.191  TCP       60 29280 > ftp [ACK] Seq=1 Ack=1 Win=7300 Len=0
401483 2004-12-15 17:47:27.958671 65.193.86.2     204.153.45.191  TCP       60 [TCP Dup ACK 401482#1] 29280 > ftp [ACK] Seq=1 Ack=1 Win=7300 Len=0
421966 2004-12-15 17:47:30.832670 204.153.45.191  65.193.86.2     TCP       60 ftp > 29280 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400
421967 2004-12-15 17:47:30.832671 204.153.45.191  65.193.86.2     TCP       60 ftp > 29280 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400
422662 2004-12-15 17:47:30.927129 65.193.86.2     204.153.45.191  TCP       60 [TCP Dup ACK 401482#2] 29280 > ftp [ACK] Seq=1 Ack=1 Win=7300 Len=0
422663 2004-12-15 17:47:30.927130 65.193.86.2     204.153.45.191  TCP       60 [TCP Dup ACK 401482#3] 29280 > ftp [ACK] Seq=1 Ack=1 Win=7300 Len=0
422754 2004-12-15 17:47:30.939373 204.153.45.191  65.193.86.2     FTP      103 Response: 220-Serv-U FTP Server v5.2 for WinSock ready...
422755 2004-12-15 17:47:30.939374 204.153.45.191  65.193.86.2     FTP      103 [TCP Fast Retransmission] Response: 220-Serv-U FTP Server v5.2 for Win
```

```
    Identification: 0xc35c (50012)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 56
    Protocol: TCP (6)
  ⊞ Header checksum: 0x2d54 [correct]
    Source: 65.193.86.2 (65.193.86.2)
    Destination: 204.153.45.191 (204.153.45.191)
⊟ Transmission Control Protocol, Src Port: 29280 (29280), Dst Port: ftp (21)
```

```
0000  00 10 7b 68 f3 41 00 02  4b 2f d4 00 08 00 45 00   ..{h.A.. K/....E.
0010  00 2c c3 5c 00 00 38 06  2d 54 41 c1 56 02 cc 99   .,.\..8. -TA.V...
0020  2d bf 72 60 00 15 ec 97  1b 4a 00 00 00 00 60 02   -.r`.... .J....`.
0030  02 00 89 b3 00 00 02 04  05 b4 00 00               ........ ....
```

Destination (ip.dst), 4 bytes | Packets: 1000000 Displayed: 56 Marked: 0 Load time: 0:43.810 | Profile: Default

Things we learn:

- *IP Addresses, apply DNS and whois*
- *FTP protocol (control connection)*
- *MAC addresses of the routers*
- *Packet sizes*

Defense

Honeynet – used to slow down attackers

- act like a desirable target
- slow down responses

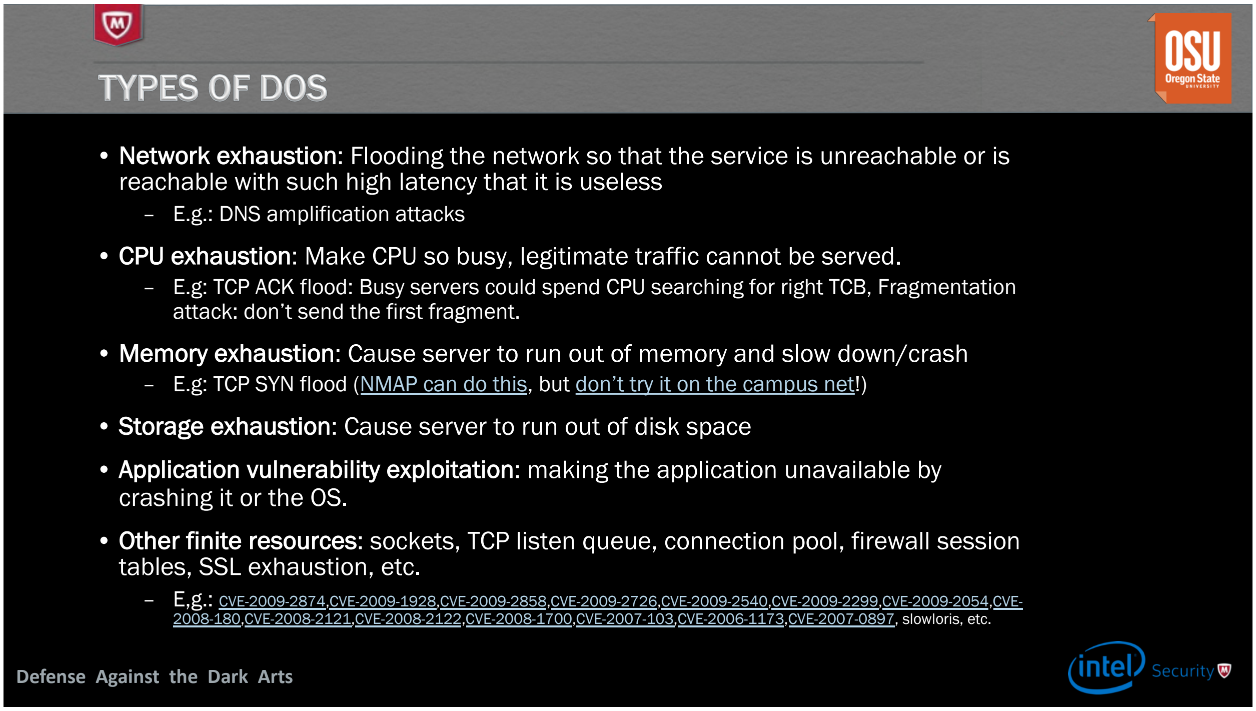
Spoofing

- get data that was intended for a different recipient
- DoS attacking
- LAND – LAN DoS
- Used in the industry - Can be used to mimic many hosts for testing
- Protection against spoofing
  - Egress filtering, ingress filtering
  - Ensure that the packet that we're receiving would normally come in through the port that it's actually coming in from.
  - What would the host do if this packet was to be sent from the host?
  - Aka reverse packet filtering
  - Ensuring we're getting the data through the correct interface

DOS Attacks:

- Why? – bring down network, hide real intent
- How? –
  - Send many requests to flood the target with requests.
  - Spoofing –
  - Ex: Slowloris.- tries to keep a connection open for a long time

- Unintentional



TYPES OF DOS

- **Network exhaustion:** Flooding the network so that the service is unreachable or is reachable with such high latency that it is useless
  - E.g.: DNS amplification attacks
- **CPU exhaustion**: Make CPU so busy, legitimate traffic cannot be served.
  - E.g: TCP ACK flood: Busy servers could spend CPU searching for right TCB, Fragmentation attack: don't send the first fragment.
- **Memory exhaustion:** Cause server to run out of memory and slow down/crash
  - E.g: TCP SYN flood (NMAP can do this, but don't try it on the campus net!)
- **Storage exhaustion**: Cause server to run out of disk space
- **Application vulnerability exploitation:** making the application unavailable by crashing it or the OS.
- **Other finite resources:** sockets, TCP listen queue, connection pool, firewall session tables, SSL exhaustion, etc.
  - E,g.: CVE-2009-2874,CVE-2009-1928,CVE-2009-2858,CVE-2009-2726,CVE-2009-2540,CVE-2009-2299,CVE-2009-2054,CVE-2008-180,CVE-2008-2121,CVE-2008-2122,CVE-2008-1700,CVE-2007-103,CVE-2006-1173,CVE-2007-0897, slowloris, etc.

Defense Against the Dark Arts

Bugs and Backdoors

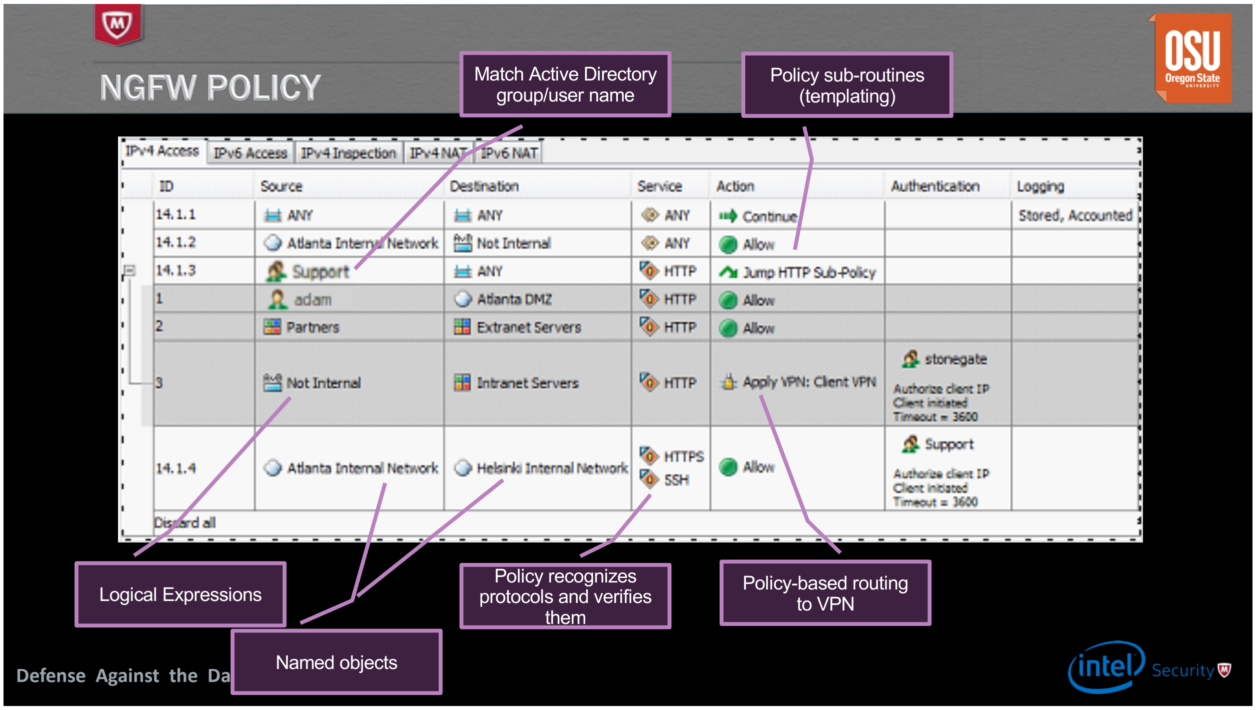
Shodan - Large collection of vulnerable routers
Not all users follow RFCs to the letter.

Defense

Stateful vs Stateless
Stateful – inspect packets to ensure they correspond to the existing connection
Stateless – apply policy rules and accept packets without checking

Fragmentation Attack: send highly fragmented packet so that the firewall has to reassemble the packet, taking up a large amount of buffer. Or send last packet first.

Deep Inspection: add inspection methods to packets



NGFW POLICY

IPv4 Access | IPv6 Access | IPv4 Inspection | IPv4 NAT | IPv6 NAT

ID	Source	Destination	Service	Action	Authentication	Logging
14.1.1	ANY	ANY	ANY	Continue		Stored, Accounted
14.1.2	Atlanta Internal Network	Not Internal	ANY	Allow		
14.1.3	Support	ANY	HTTP	Jump HTTP Sub-Policy		
1	adam	Atlanta DMZ	HTTP	Allow		
2	Partners	Extranet Servers	HTTP	Allow		
3	Not Internal	Intranet Servers	HTTP	Apply VPN: Client VPN	stonegate Authorize client IP Client initiated Timeout = 3600	
14.1.4	Atlanta Internal Network	Helsinki Internal Network	HTTPS SSH	Allow	Support Authorize client IP Client initiated Timeout = 3600	
Discard all						

Match Active Directory group/user name

Policy sub-routines (templating)

Logical Expressions

Named objects

Policy recognizes protocols and verifies them

Policy-based routing to VPN

IKE: Internet Key Exchange (UDP)
Works like a public key exchange.
Establish an IPv4 tunnel.

Endpoint Context: understanding the nature of the traffic can help with understanding if an attack is happening.

Dynamic Analysis: A device that executes malware to determine the nature of the malware. Allows a user to view the activity in the box.

Reputation: Prevalence (how often) and Age (how old)

Wireshark Notes:
Follow TCP Stream – provides data on the TCP stream
Stats and Conversation – provides more data on conversations
Export Objects – allows user to get files and data on files that are sent through HTTP
Find Packet – search

OSU – DADA

Network Security

Homework

Color the following paragraph green for parts you think are still valid, and red for parts you think are no longer valid from our perspective, 35 years later.   Add bullet points to justify your opinion.

Robustness Principle: 1980-1989 from RFC-1122 Jonathan Postel, 1989

Once there was a great man, named Postel. See RFC 2468.

1.2.2 Robustness Principle

At every layer of the protocols, there is a general rule whose application can lead to enormous benefits in robustness and interoperability [ref to rfc760, 1980]:

> "Be liberal in what you accept, and conservative in what you send"

Software should be written to deal with every conceivable error, no matter how unlikely; sooner or later a packet will come in with that particular combination of errors and attributes, and unless the software is prepared, chaos can ensue. In general, it is best to assume that the network is filled with malevolent entities that will send in packets designed to have the worst possible effect.

- We should be cautious in what we accept, as unexpected outputs should result in a contained failure of the system/connection.
- In a long enough timescale, a host will most likely encounter inputs it has not accounted for.
- We can write code such that there is a failure followed by cleanup of resources or use general metrics (for example, connections with absurdly long TTL, we could take mean TTL +/- 3 stdev)
- With the rise of cybercrime, it seems appropriate to assume the worst.

This assumption will lead to suitable protective design, although the most serious problems in the Internet have been caused by unenvisaged mechanisms triggered by low-probability events; mere human malice would never have taken so devious a course!

- With the advent of machine learning and methods of generation of combinations of error, it is likely that low probability events could be of intentional design.

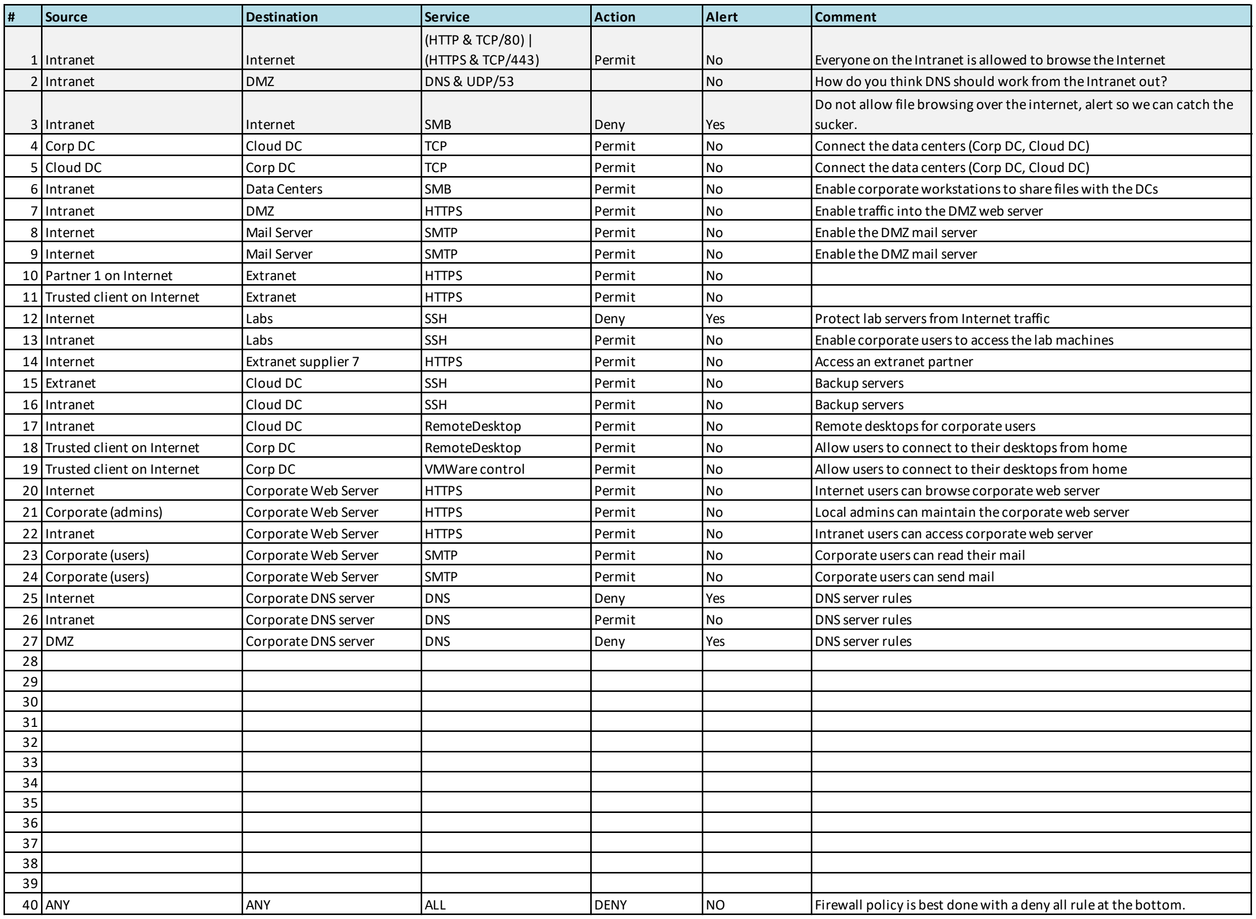
Adaptability to change must be designed into all levels of Internet host software. As a simple example, consider a protocol specification that contains an enumeration of values for a particular header field—e.g., a type field, a port number, or an error code; this enumeration must be assumed to be incomplete. Thus, if a protocol specification defines four possible error codes, the software must not break when a fifth code shows up.
An undefined code might be logged (see below), but it must not cause a failure.

- Getting software to communicate properly should be the responsibility of the client, so long as it is following the API of the host (and the API is properly up to date and has few bugs)

The second part of the principle is almost as important: software on other hosts may contain deficiencies that make it unwise to exploit legal but obscure protocol features. It is unwise to stray far from the obvious and simple, lest untoward effects result elsewhere. A corollary of this is "watch out for misbehaving hosts"; host software should be prepared, not just to survive other misbehaving hosts, but also to cooperate to limit the amount of disruption such hosts can cause to the shared communication facility.

- To ensure sufficiently reliable communication, we should be hardening hosts. Reliable communication is paramount.

Firewall Worksheet



#	Source	Destination	Service	Action	Alert	Comment
1	Intranet	Internet	(HTTP & TCP/80) | (HTTPS & TCP/443)	Permit	No	Everyone on the Intranet is allowed to browse the Internet
2	Intranet	DMZ	DNS & UDP/53		No	How do you think DNS should work from the Intranet out?
3	Intranet	Internet	SMB	Deny	Yes	Do not allow file browsing over the internet, alert so we can catch the sucker.
4	Corp DC	Cloud DC	TCP	Permit	No	Connect the data centers (Corp DC, Cloud DC)
5	Cloud DC	Corp DC	TCP	Permit	No	Connect the data centers (Corp DC, Cloud DC)
6	Intranet	Data Centers	SMB	Permit	No	Enable corporate workstations to share files with the DCs
7	Intranet	DMZ	HTTPS	Permit	No	Enable traffic into the DMZ web server
8	Internet	Mail Server	SMTP	Permit	No	Enable the DMZ mail server
9	Internet	Mail Server	SMTP	Permit	No	Enable the DMZ mail server
10	Partner 1 on Internet	Extranet	HTTPS	Permit	No	
11	Trusted client on Internet	Extranet	HTTPS	Permit	No	
12	Internet	Labs	SSH	Deny	Yes	Protect lab servers from Internet traffic
13	Intranet	Labs	SSH	Permit	No	Enable corporate users to access the lab machines
14	Internet	Extranet supplier 7	HTTPS	Permit	No	Access an extranet partner
15	Extranet	Cloud DC	SSH	Permit	No	Backup servers
16	Intranet	Cloud DC	SSH	Permit	No	Backup servers
17	Intranet	Cloud DC	RemoteDesktop	Permit	No	Remote desktops for corporate users
18	Trusted client on Internet	Corp DC	RemoteDesktop	Permit	No	Allow users to connect to their desktops from home
19	Trusted client on Internet	Corp DC	VMWare control	Permit	No	Allow users to connect to their desktops from home
20	Internet	Corporate Web Server	HTTPS	Permit	No	Internet users can browse corporate web server
21	Corporate (admins)	Corporate Web Server	HTTPS	Permit	No	Local admins can maintain the corporate web server
22	Intranet	Corporate Web Server	HTTPS	Permit	No	Intranet users can access corporate web server
23	Corporate (users)	Corporate Web Server	SMTP	Permit	No	Corporate users can read their mail
24	Corporate (users)	Corporate Web Server	SMTP	Permit	No	Corporate users can send mail
25	Internet	Corporate DNS server	DNS	Deny	Yes	DNS server rules
26	Intranet	Corporate DNS server	DNS	Permit	No	DNS server rules
27	DMZ	Corporate DNS server	DNS	Deny	Yes	DNS server rules
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40	ANY	ANY	ALL	DENY	NO	Firewall policy is best done with a deny all rule at the bottom.