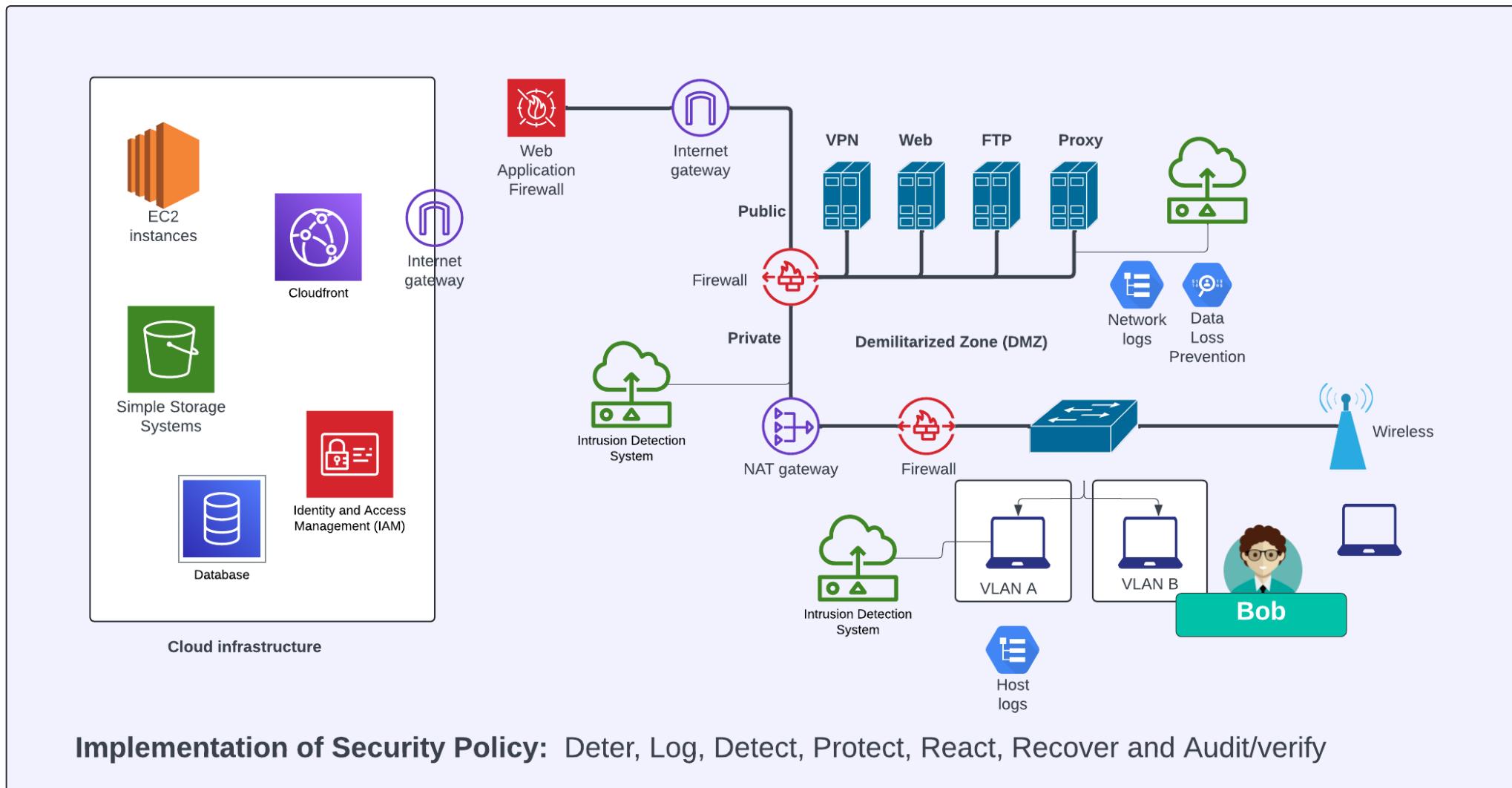




“From bits to information”

Intrusion
Detection
Systems

Information Security



Outline

- Attack Patterns.
- SNORT.
- IDS Types



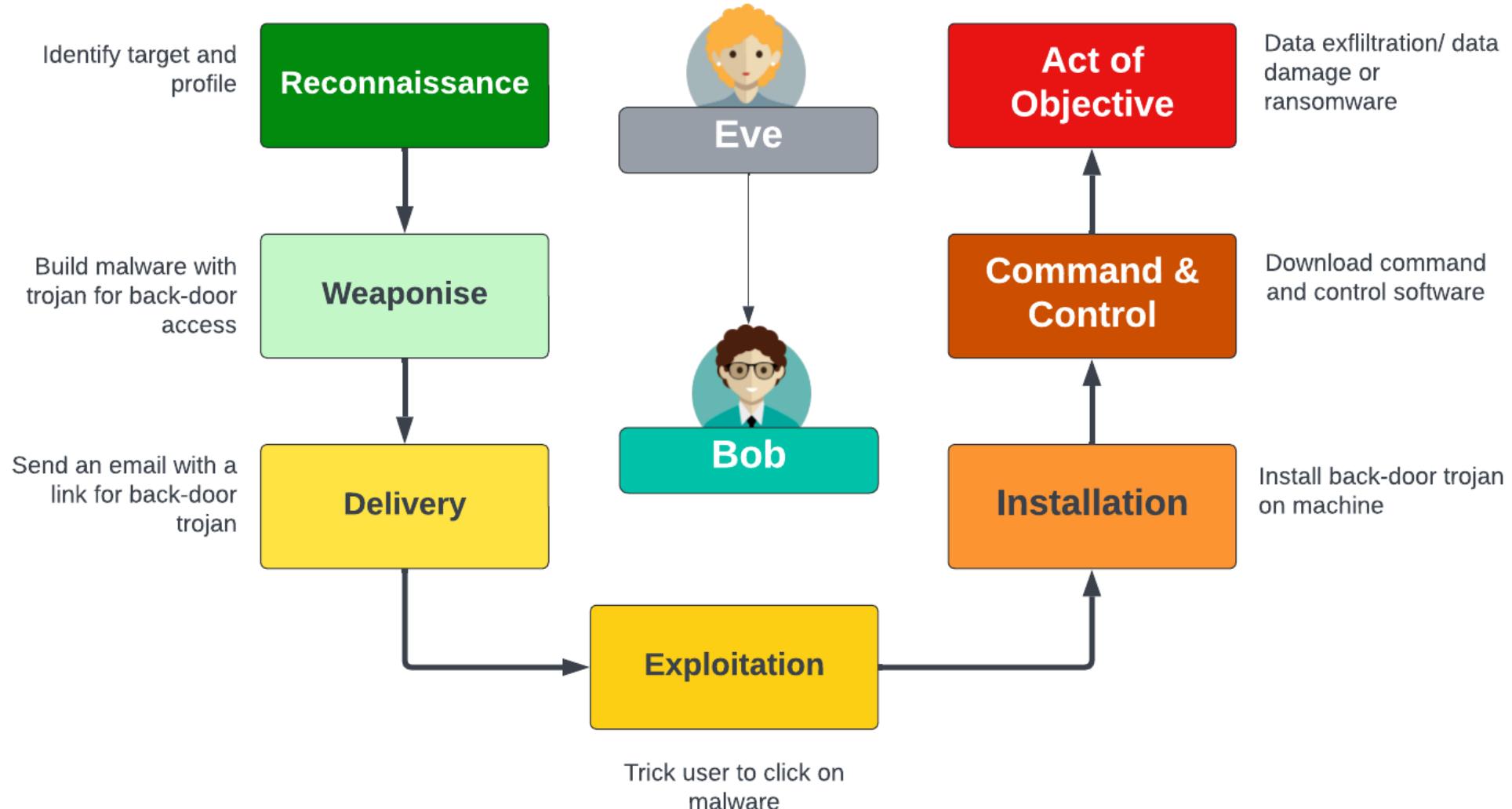
cyber
& data

cyber & data

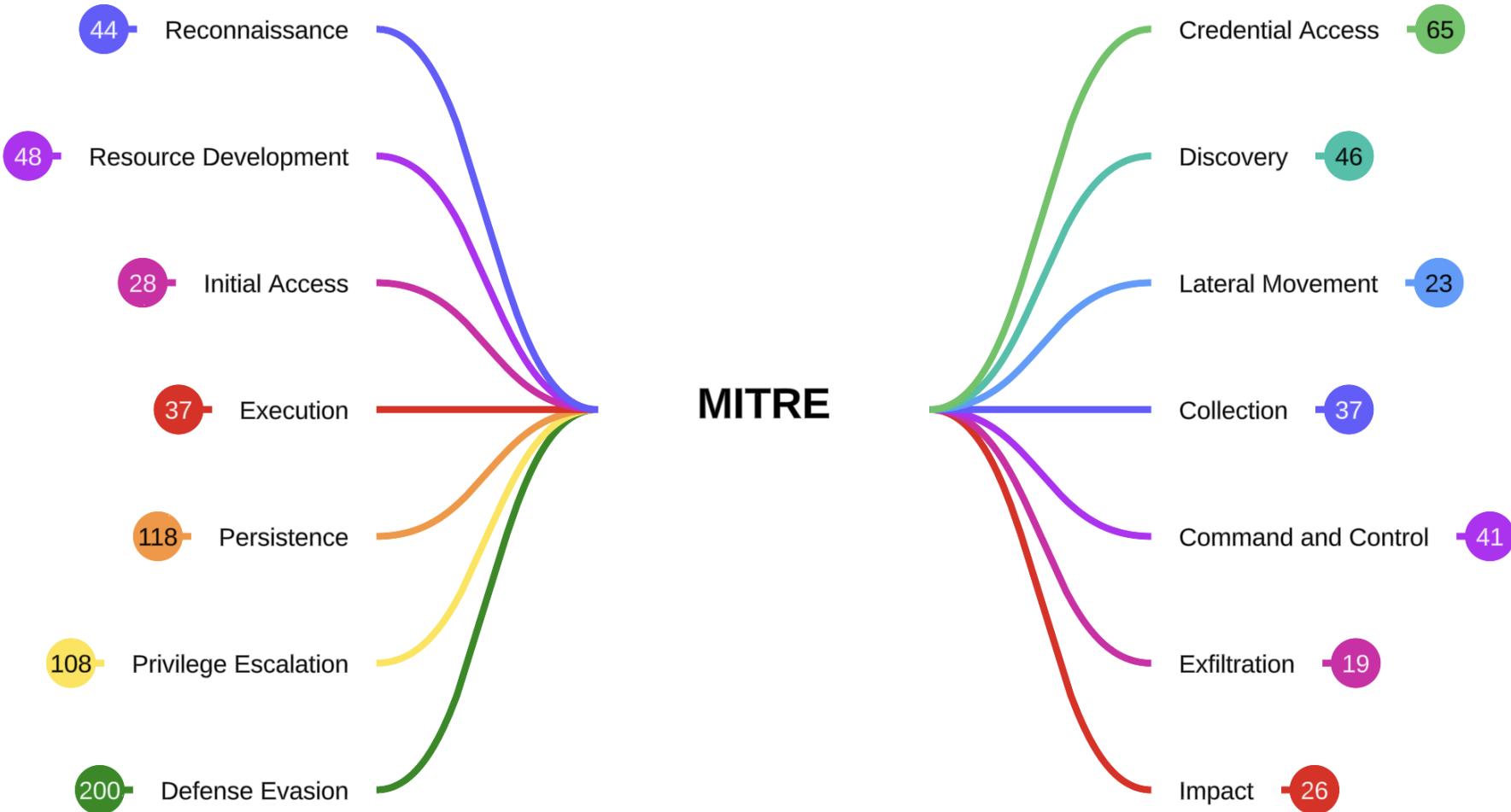
“From bits to information”

Attack Patterns

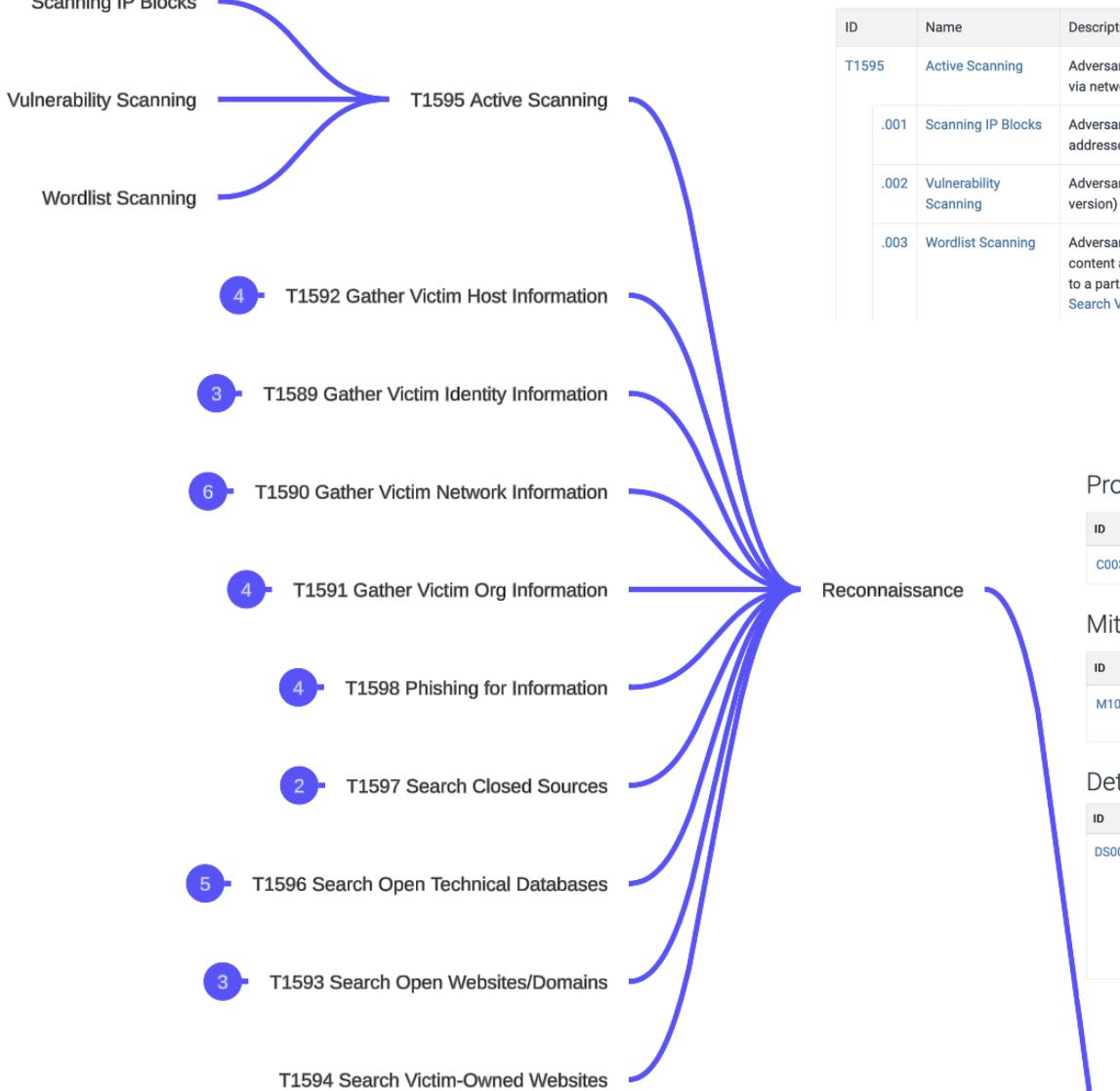
Kill Chain Model



MITRE ATT&CK (Enterprise)



MITRE ATT&CK (Enterprise)



Techniques

Techniques: 10		
ID	Name	Description
T1595	Active Scanning	Adversaries may execute active reconnaissance scans to gather information that can be used during targeting. Active scans are those where the adversary probes victim infrastructure via network traffic, as opposed to other forms of reconnaissance that do not involve direct interaction.
.001	Scanning IP Blocks	Adversaries may scan victim IP blocks to gather information that can be used during targeting. Public IP addresses may be allocated to organizations by block, or a range of sequential addresses.
.002	Vulnerability Scanning	Adversaries may scan victims for vulnerabilities that can be used during targeting. Vulnerability scans typically check if the configuration of a target host/application (ex: software and version) potentially aligns with the target of a specific exploit the adversary may seek to use.
.003	Wordlist Scanning	Adversaries may iteratively probe infrastructure using brute-forcing and crawling techniques. While this technique employs similar methods to Brute Force, its goal is the identification of content and infrastructure rather than the discovery of valid credentials. Wordlists used in these scans may contain generic, commonly used names and file extensions or terms specific to a particular software. Adversaries may also create custom, target-specific wordlists using data gathered from other Reconnaissance techniques (ex: Gather Victim Org Information, or Search Victim-Owned Websites).

Procedure Examples

ID	Name	Description
C0030	Triton Safety Instrumented System Attack	In the Triton Safety Instrumented System Attack, TEMP.Veles engaged in network reconnaissance against targets of interest. ^[3]

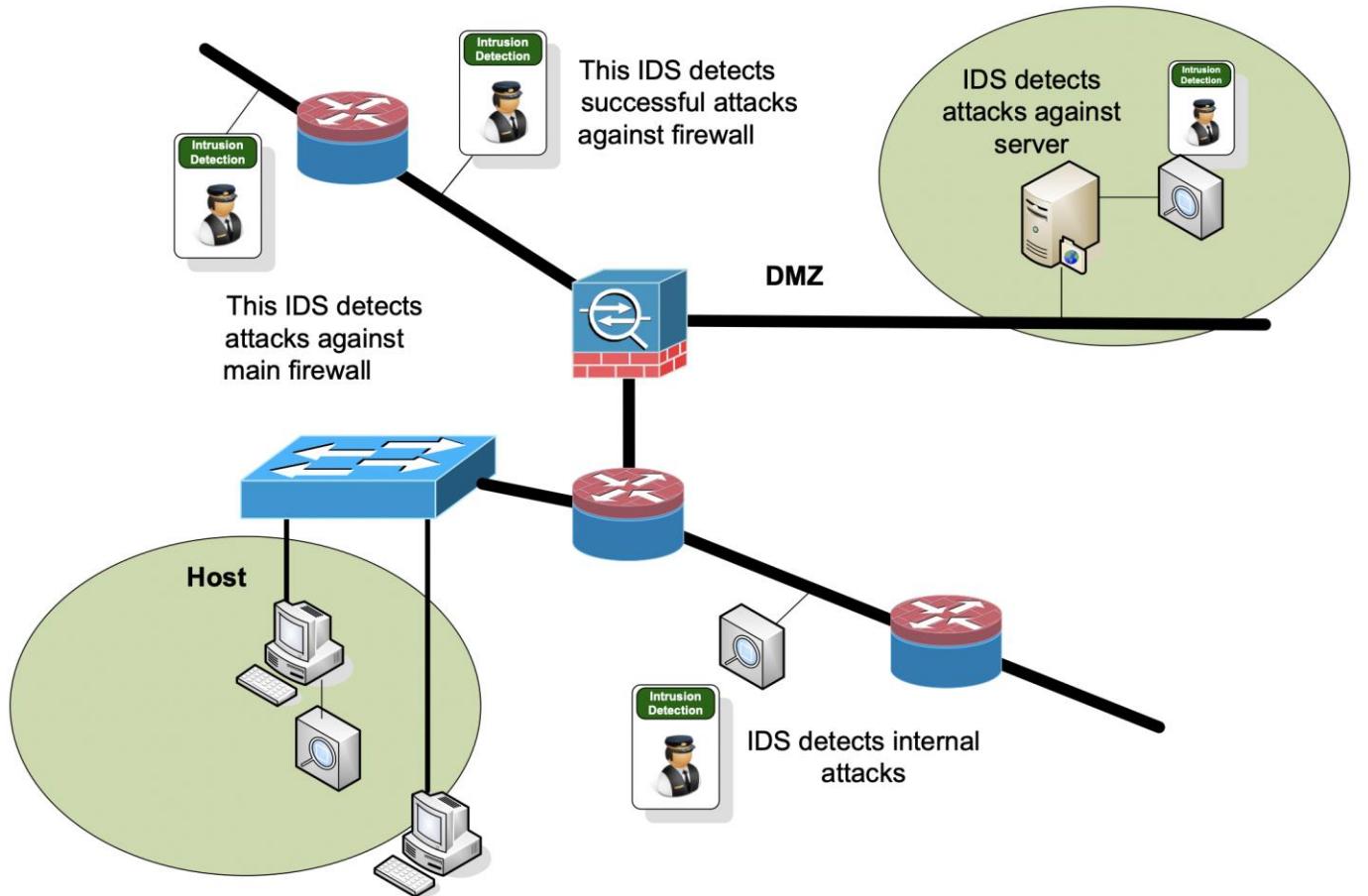
Mitigations

ID	Mitigation	Description
M1056	Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls. Efforts should focus on minimizing the amount and sensitivity of data available to external parties.

Detection

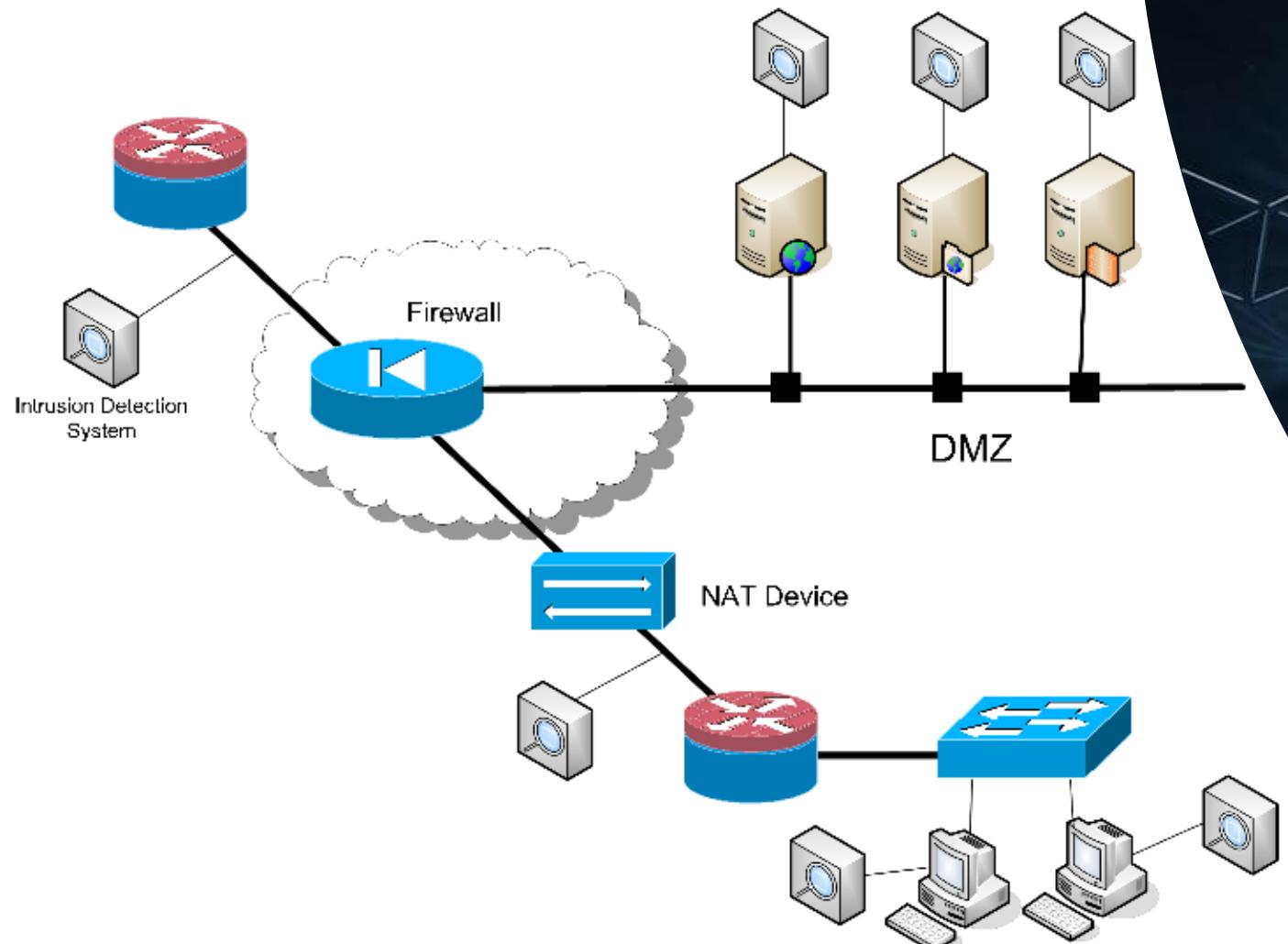
ID	Data Source	Data Component	Detects
DS0029	Network Traffic	Network Traffic Content	Monitor and analyze traffic patterns and packet inspection associated to protocol(s) that do not follow the expected protocol standards and traffic flows (e.g extraneous packets that do not belong to established flows, gratuitous or anomalous traffic patterns, anomalous syntax, or structure). Consider correlation with process monitoring and command line to detect anomalous processes execution and command line arguments associated to traffic patterns (e.g. monitor anomalies in use of files that do not normally initiate connections for respective protocol(s)).
		Network Traffic Flow	Monitor network data for uncommon data flows. Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious.

Host-based or Network-based



cyber
& data

IDS Placement



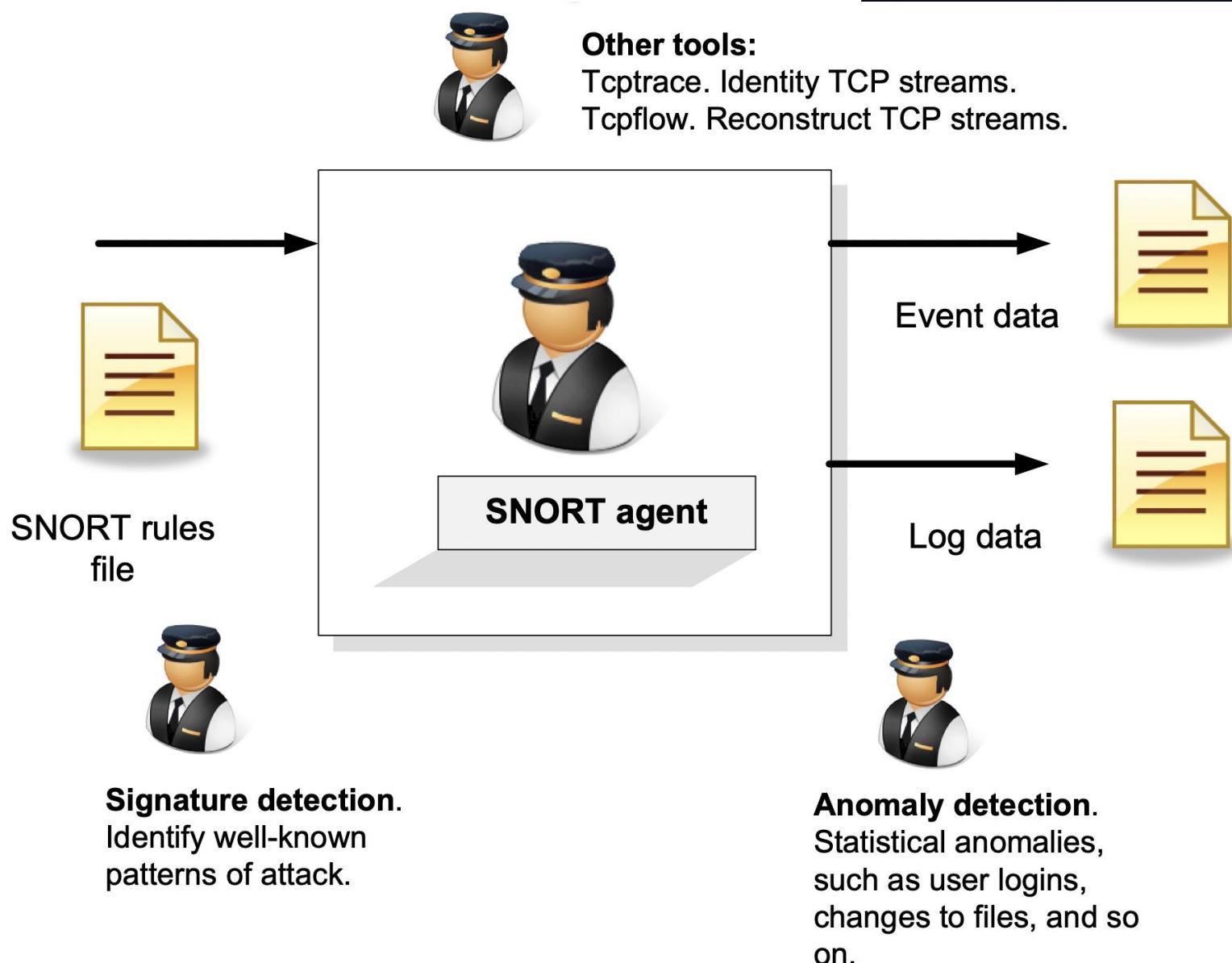
cyber
&
data

cyber & data

“From bits to information”

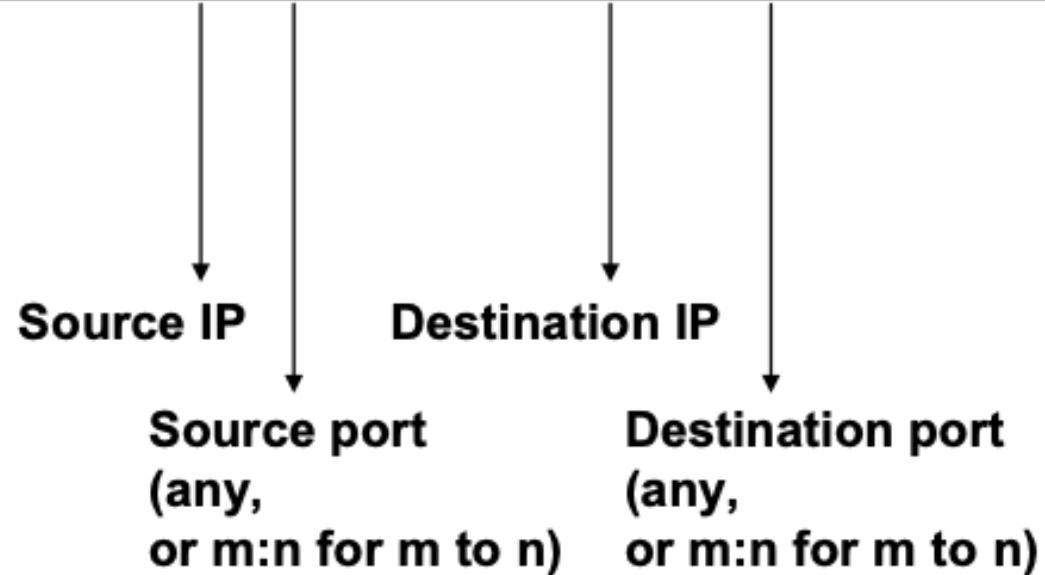
Snort

Snort



Example of Snort Rule

```
alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg:"mountd access";)
```



Example Snort Rule

```
alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg:"mountd access";)
```



Payload detection:

Hex sequence

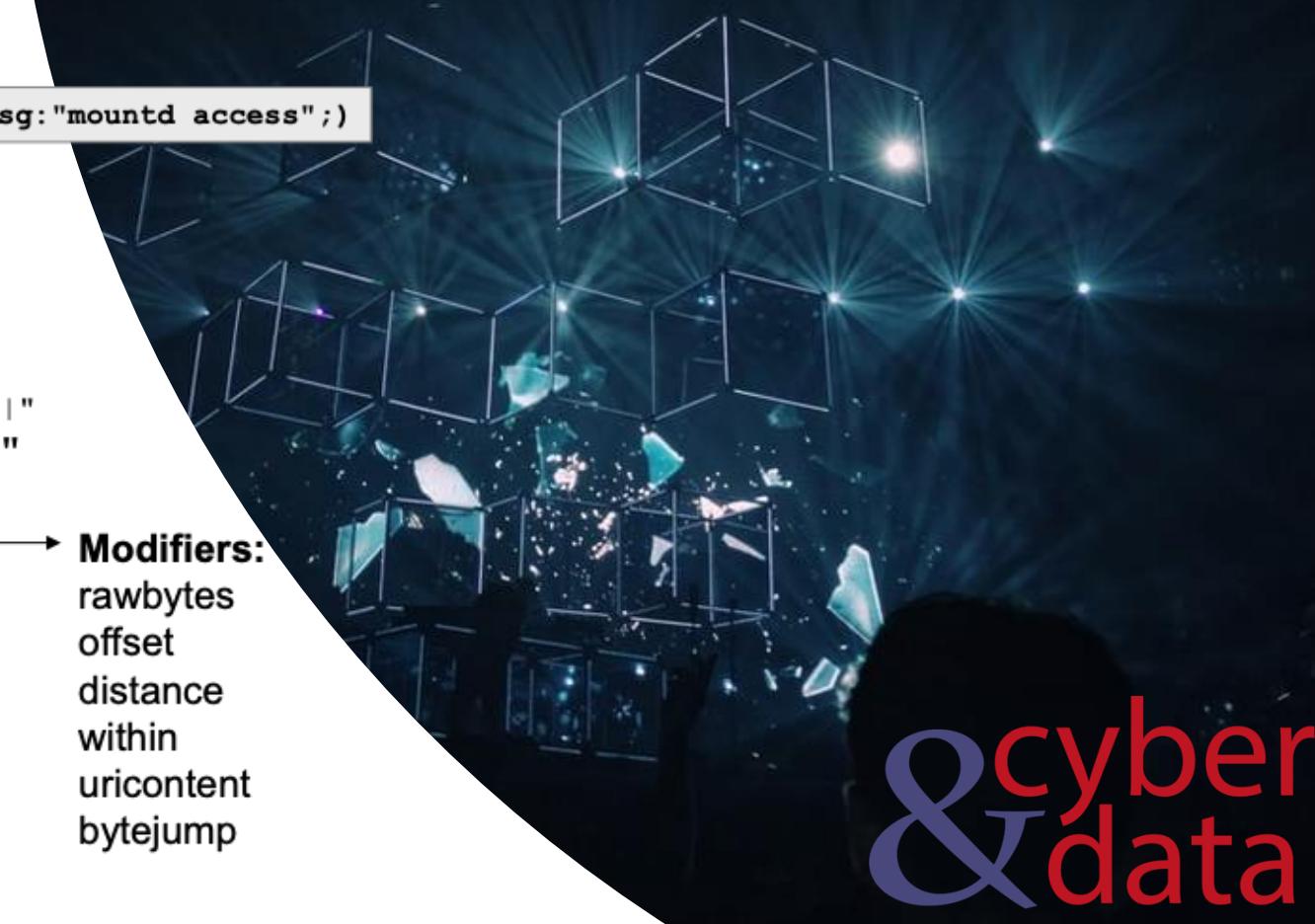
"|00 01 86 a5|"

Text sequence

"USER root"

Modifiers:

rawbytes
offset
distance
within
uricontent
bytejump



cyber
&
data

Example Snort Rule

```
alert tcp $SMTP_SERVERS any -> $EXTERNAL_NET 25
(msg:"VIRUS OUTBOUND .vbs file attachment";
flow:to_server,established; content:"Content-Disposition|3a|";
content:"filename=|22|"; distance:0; within:30;
content:".vbs|22|"; distance:0; within:30; nocase; sid:999)
```



```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21
(msg:"FTP passwd retrieval attempt"; flow:to_server,established;
content:"RETR"; nocase; content:"passwd"; sid:999)
```



```
alert tcp any any -> any 139 (msg:"Virus - Possible QAZ Worm";
flags:A; content: "|71 61 7a 77 73 78 2e 68 73 71|"; sid:999)
```



```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21
(msg:"FTP CWD ~root attempt"; flow:to_server,established;
content:"CWD"; nocase; content:"~root"; nocase;
distance:1; sid:999)
```

Example Snort Rule

```
sfportscan: proto { all } memcap { 10000000 } sense_level { low }
```

```
preprocessor flow: stats_interval 0 hash 2
preprocessor sfportscan: proto { all } scan_type { all }
                     sense_level { low } logfile { portscan.log }
```

```
C:\> snort -c scan.rule -dev -i 3 -p -l c:\\bill -K ascii
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file scan.rule
,-----[Flow Config]-----
| Stats Interval: 0
| Hash Method: 2
| Memcap: 10485760
| Rows : 4096
| Overhead Bytes: 16388(%0.16)
'
Portscan Detection Config:
 Detect Protocols: TCP UDP ICMP IP
 Detect Scan Type: portscan portsweep decoy_portscan
                   distributed_portscan
 Sensitivity Level: Low
 Memcap (in bytes): 1048576
 Number of Nodes: 3869
 Logfile: c:\\bill/portscan.log

Tagged Packet Limit: 256
```

cyber & data

“From bits to information”

A Simple Rule

Snort Rule

IDS
A simple rule

```
alert tcp any any -> any any (content:"the"; msg:"The found ....";)
```

```
Snort -v -c bill.rules -l /log
```

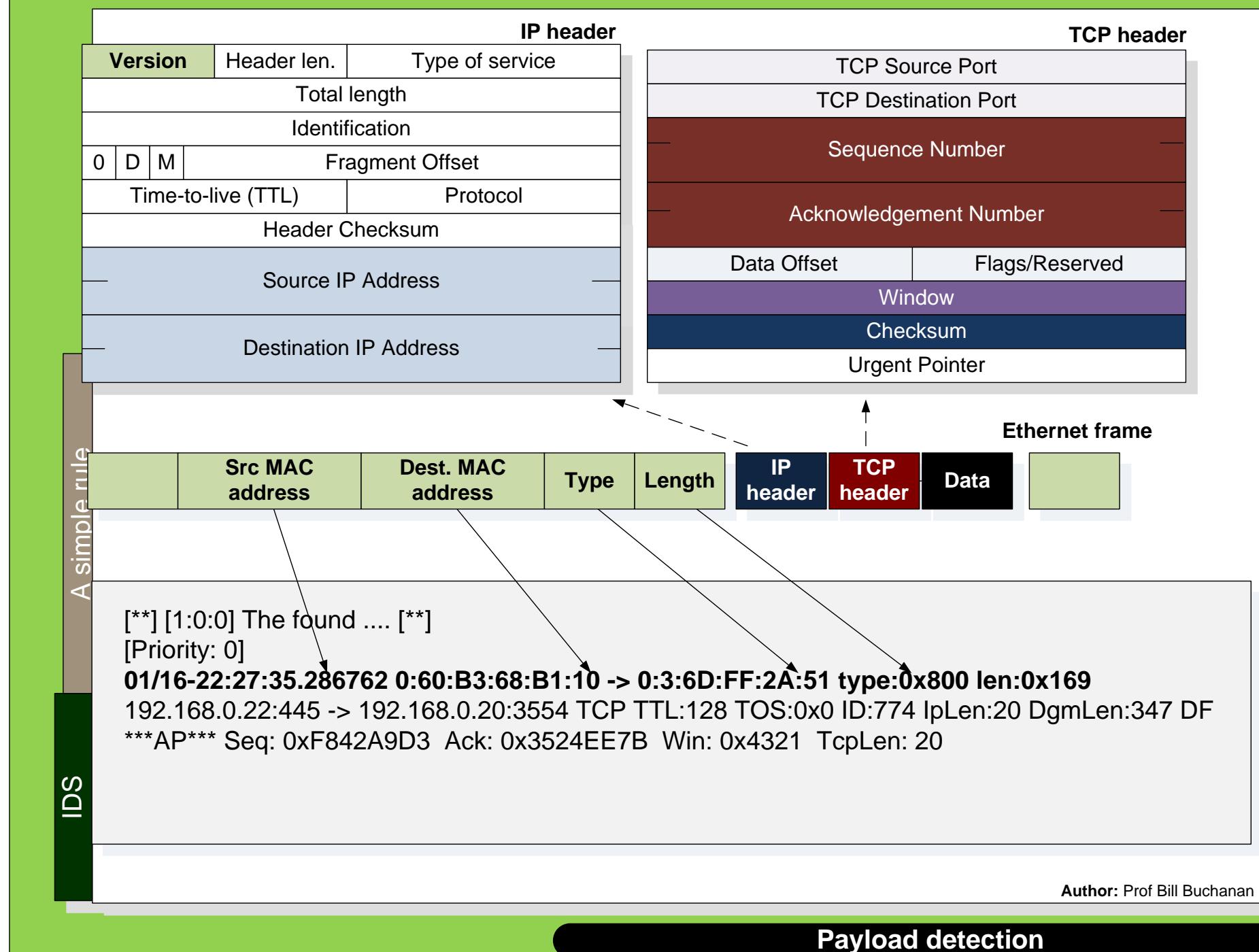
Alert.ids
(in \log)

```
[**] [1:0:0] The found .... [**] 16 January 10:27pm  
[Priority: 0]  
01/16-22:27:35.286762 0:60:B3:68:B1:10 -> 0:3:6D:FF:2A:51 type:0x800 len:0x169  
192.168.0.22:445 -> 192.168.0.20:3554 TCP TTL:128 TOS:0x0 ID:774 IpLen:20  
DgmLen:347 DF  
***AP*** Seq: 0xF842A9D3 Ack: 0x3524EE7B Win: 0x4321 TcpLen: 20  
  
[**] [1:0:0] The found .... [**]  
[Priority: 0]  
01/16-22:27:35.287084 0:3:6D:FF:2A:51 -> 0:60:B3:68:B1:10 type:0x800 len:0x198  
192.168.0.20:3554 -> 192.168.0.22:445 TCP TTL:128 TOS:0x0 ID:1086 IpLen:20  
DgmLen:394 DF  
***AP*** Seq: 0x3524EE7B Ack: 0xF842AB06 Win: 0x42E4 TcpLen: 20  
  
[**] [1:0:0] The found .... [**]  
[Priority: 0]  
01/16-22:27:35.290026 0:60:B3:68:B1:10 -> 0:3:6D:FF:2A:51 type:0x800 len:0x5D  
192.168.0.22:445 -> 192.168.0.20:3554 TCP TTL:128 TOS:0x0 ID:775 IpLen:20  
DgmLen:79 DF  
***AP*** Seq: 0xF842AB06 Ack: 0x3524EFDD Win: 0x41BF TcpLen: 20
```

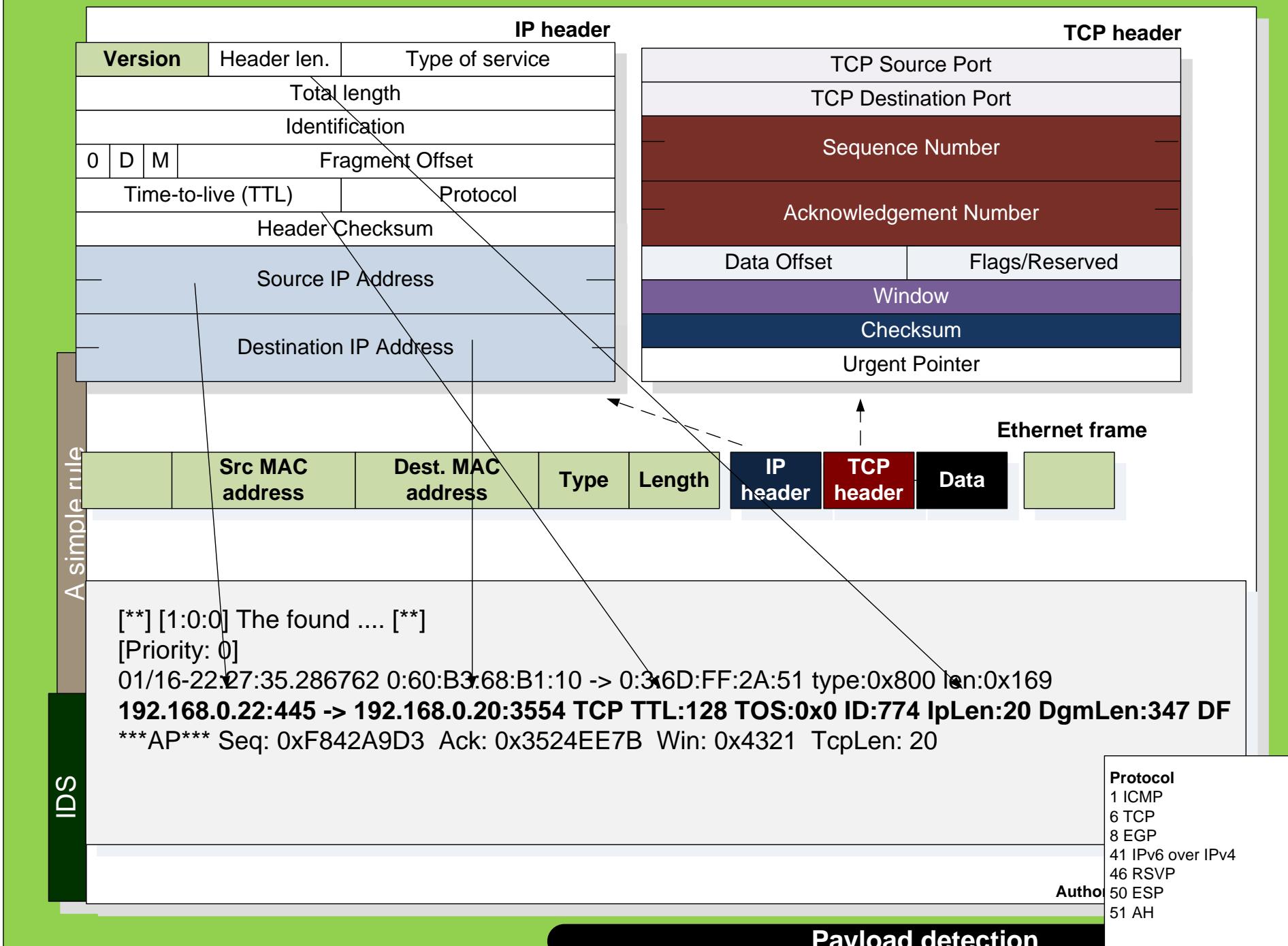
Author: Prof Bill Buchanan

Running Snort

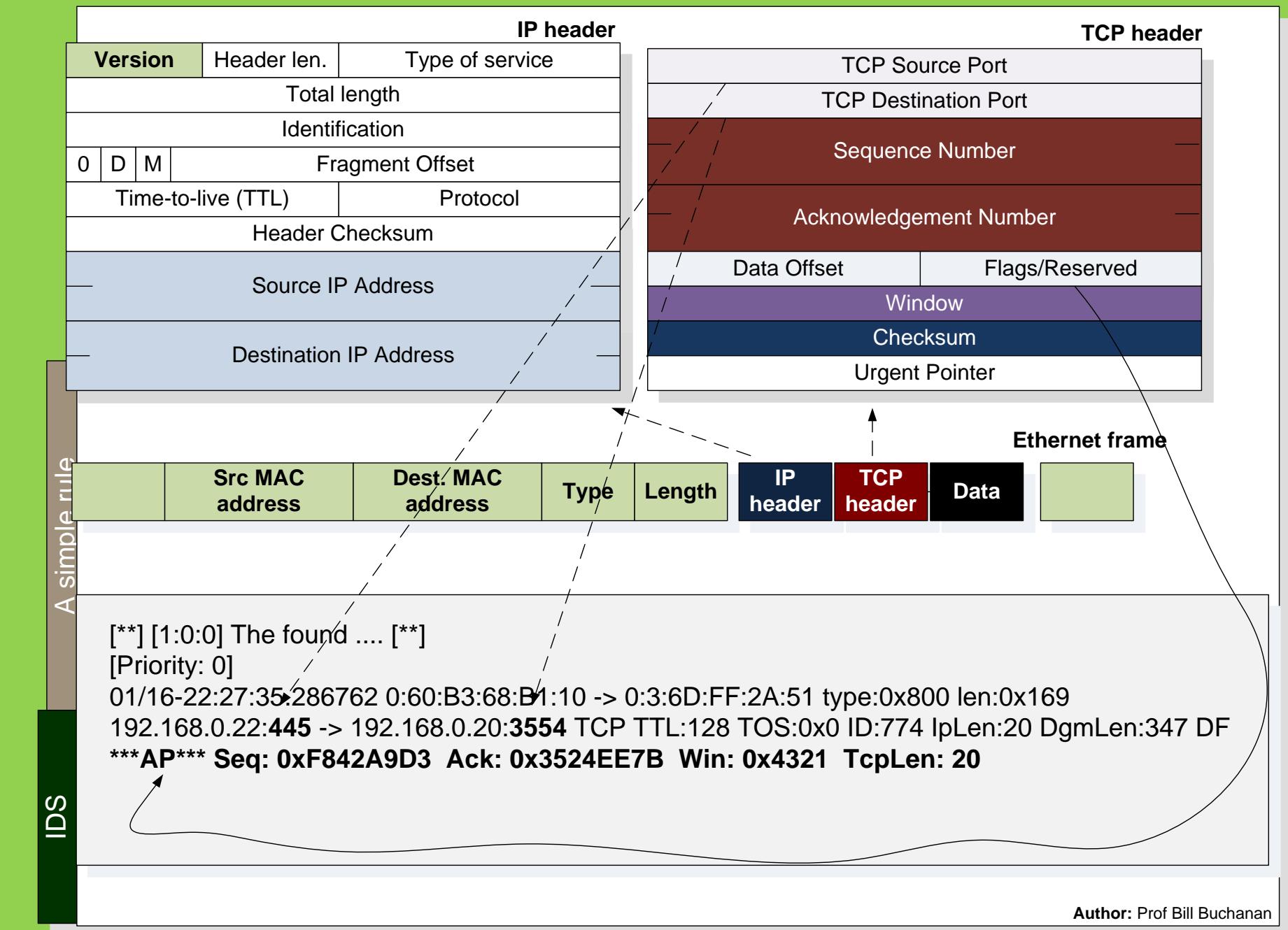
Snort Rule



Snort Rule



Snort Rule



cyber & data

"From bits to information"

A Few Intrusions

An example

IDS
A few intrusions

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 1863
(msg:"CHAT MSN login attempt"; flow:to_server,established; content:"USR "; depth:4;
nocase; content:" TWN "; distance:1; nocase;
classtype:policy-violation; sid:1991; rev:1;)
```



No.	Source	Destination	Protocol	Info
1	192.168.0.3	207.46.28.93	TCP	5398 > 1863 [SYN] Seq=0 Len=0 MSS=1460
2	207.46.28.93	192.168.0.3	TCP	1863 > 5398 [SYN, ACK] Seq=0 Ack=1 Win=5840
3	192.168.0.3	207.46.28.93	TCP	5398 > 1863 [ACK] Seq=1 Ack=1 Win=17520
4	192.168.0.3	207.46.111.39	MSNMS	USR 2 TWN I test@hotmail.com
5	207.46.111.39	192.168.0.3	MSNMS	USR 26 OK test@hotmail.com 1 0



```
private static void device_PcapOnPacketArrival(...)
{
    if(packet is TCPPacket)
    {
        TCPPacket tcp = (TCPPacket)packet;
        int destPort = tcp.SourcePort; byte [] b = tcp.Data;
        ASCIIEncoding format = new ASCIIEncoding();
        string s = format.GetString(b); s=s.ToLower();
        if (destPort==1863 && (s.StartsWith("usr ")) && s.IndexOf(" twn ")>0 )
            Console.WriteLine("MSN Messenger Login");
    }
}
```



Author: Prof Bill Buchanan

Rules

IDS
A few intrusions

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 1863
```

```
(msg:"CHAT MSN login attempt";
```

DEPTH. Modifier for previous content ... defines to look within 4 bytes of the payload

```
flow:to_server,established;
```

```
content: "USR "; depth:4; nocase;
```

NOCASE; Modifier for previous content ... ignore the case of the content

```
content:" TWN "; distance:1; nocase;
```

DISTANCE. Modified for previous content and defines how far into the payload it should search (in bytes)

```
classtype:policy-violation; sid:1991; rev:1;)
```

No.	Source	Destination	Protocol
1	192.168.0.3	207.46.28.93	TCP
2	192.168.0.3	207.46.28.93	TCP
3	207.46.28.93	192.168.0.3	TCP
4	192.168.0.3	207.46.28.93	TCP
5	192.168.0.3	207.46.111.39	MSNMS
6	207.46.111.39	192.168.0.3	MSNMS

1863 > 5398 [SYN, ACK] Seq=0 Ack=1 Win=5840
5398 > 1863 [ACK] Seq=1 Ack=1 Win=17520
USR 2 TWN I test@hotmail.com
USR 26 OK test@hotmail.com 1 0



Author: Prof Bill Buchanan

Intrusions/Policy Violations

cyber
data

Rules

A few intrusions

```

alert tcp $HOME_NET any -> $EXTE
  (msg:"CHAT MSN login attempt";
   flow:to_server,established;
   content: "USR "; depth:4; nocase;
   content:" TWN "; distance:1; nocase;
   classtype:policy-violation; sid:1991
  )
  
```

IDS

No.	1	2	3	4	5	6
	icmp-event	misc-activity				
		network-scan				
			not-suspicious			
			protocol-command-decode			
			string-detect			

IPS

- attempted-admin
- attempted-user
- policy-violation
- shellcode-detect
- successful-admin
- successful-user
- trojan-activity
- unsuccessful-user
- web-application-attack

- attempted-dos
- attempted-recon
- bad-unknown
- default-login-attempt
- denial-of-service
- misc-attack
- non-standard-protocol
- rpc-portmap-decode
- successful-dos
- successful-recon-largescale
- ...
- web-application-activity

Author: Prof Bill Buchanan

Rules

IDS
A few intrusions

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 1863
```

```
(msg:"CHAT MSN login attempt";
```

DEPTH. Modifier for previous content ... defines to look within 4 bytes of the payload

```
flow:to_server,established;
```

```
content: "USR "; depth:4; nocase;
```

NOCASE; Modifier for previous content ... ignore the case of the content

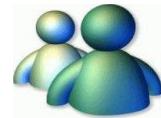
```
content:" TWN "; distance:1; nocase;
```

DISTANCE. Modified for previous content and defines how far into the payload it should search (in bytes)

```
classtype:policy-violation; sid:1991; rev:1;)
```

No.	Source	Destination	Protocol
1	192.168.0.3	207.46.28.93	TCP
2	192.168.0.3	207.46.28.93	TCP
3	207.46.28.93	192.168.0.3	TCP
4	192.168.0.3	207.46.28.93	TCP
5	192.168.0.3	207.46.111.39	MSNMS
6	207.46.111.39	192.168.0.3	MSNMS

1863 > 5398 [SYN, ACK] Seq=0 Ack=1 Win=5840
5398 > 1863 [ACK] Seq=1 Ack=1 Win=17520
USR 2 TWN I test@hotmail.com
USR 26 OK test@hotmail.com 1 0



Author: Prof Bill Buchanan

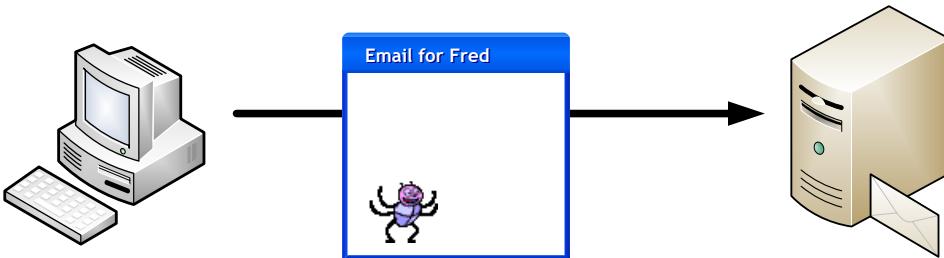
Intrusions/Policy Violations

cyber
data

Rules

IDS
A few intrusions

```
alert tcp $SMTP_SERVERS any -> $EXTERNAL_NET 25
  (msg:"VIRUS OUTBOUND .exe file attachment";
  flow:to_server,established; content:"Content-Disposition|3a|";
  content:"filename=|22|"; distance:0; within:30;
  content:".exe|22|"; distance:0; within:30; nocase;
  classtype:suspicious-filename-detect; sid:2160; rev:1;)
```



Email server
(SMTP –
listening on port
25)

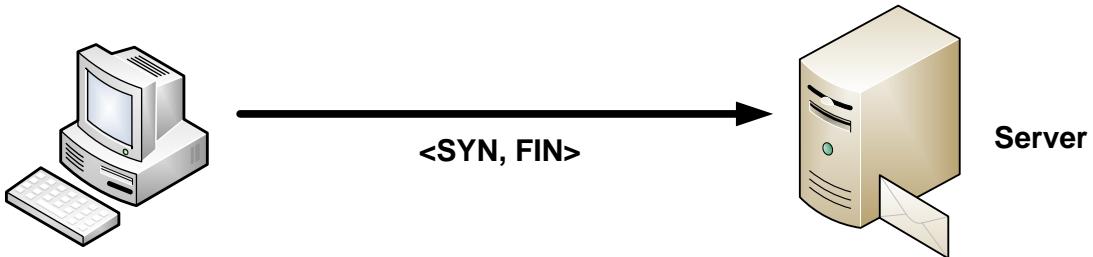


```
private static void device_PcapOnPacketArrival(..)
{
    TCPPacket tcp = (TCPPacket)packet;
    if(packet is TCPPacket)
    {
        int destPort = tcp.SourcePort;    byte [] b = tcp.Data;
        ASCIIEncoding format = new ASCIIEncoding();
        string s = format.GetString(b); s=s.ToLower();
        if (destPort==25 && s.IndexOf("Content-Disposition;")>0
            && s.IndexOf("filename=\"")>0 && s.IndexOf(".exe\"")>0 )
            Console.WriteLine("VIRUS OUTBOUND .exe file attachment");
    }
}
```

Rules

IDS A few intrusions

```
alert tcp any any -> any any (msg:"SYN FIN Scan"; flags: SF;)
alert tcp any any -> any any (msg:"FIN Scan"; flags: F;)
alert tcp any any -> any any (msg:"NULL Scan"; flags: 0;)
alert tcp any any -> any any (msg:"XMAS Scan";flags: FPU;)
alert tcp any any -> any any (msg:"FULL XMAS Scan";flags:SRAFPU;)
```



Server



```
private static void device_PcapOnPacketArrival(...)
{
    if(packet is TCPPacket)
    {
        TCPPacket tcp = (TCPPacket)packet;
        if (tcp.Syn==true && tcp.Fin==true)
            Console.WriteLine("SYN FIN Scan");
    }
}
```

Author: Prof Bill Buchanan

Rules

```
alert tcp $TELNET_SERVERS 23 -> $EXTERNAL_NET  
any (msg:"TELNET root login";  
flow:from_server,established;  
content:"login|3A| root";  
classtype:suspicious-login; sid:719; rev:7;)
```

IDS
A few intrusions

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: (ip.addr eq 192.168.0.4 and ip.addr eq 146.176) Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
2069	77.29781	pc165229.napier.ac.uk	192.168.0.4	TCP	telnet > 2403 [ACK] Seq=41 ACK=41 Win=5840 Len=0
2070	77.298170	pc165229.napier.ac.uk	192.168.0.4	TELNET	Telnet Data ...
2071	77.298329	192.168.0.4	pc165229.napier.ac.uk	TELNET	Telnet Data ...
2073	77.386208	pc165229.napier.ac.uk	192.168.0.4	TCP	telnet > 2403 [ACK] Seq=53 Ack=44 Win=5840 Len=0
2074	77.386276	192.168.0.4	pc165229.napier.ac.uk	TELNET	Telnet Data ...
2077	77.438728	pc165229.napier.ac.uk	192.168.0.4	TCP	telnet > 2403 [ACK] Seq=53 Ack=53 Win=5840 Len=0
2078	77.439196	pc165229.napier.ac.uk	192.168.0.4	TELNET	Telnet Data ...
2079	77.439345	192.168.0.4	pc165229.napier.ac.uk	TELNET	Telnet Data ...
2084	77.491337	pc165229.napier.ac.uk	192.168.0.4	TELNET	Telnet Data ...
2085	77.491427	192.168.0.4	pc165229.napier.ac.uk	TELNET	Telnet Data ...

Frame 2084 (142 bytes on wire, 142 bytes captured)
Ethernet II, Src: Netgear_b0:d6:8c (00:18:4d:b0:d6:8c), Dst: IntelCor_34:02:f0 (00:15:00:34:02:f0)
Internet Protocol, Src: pc165229.napier.ac.uk (146.176.165.229), Dst: 192.168.0.4 (192.168.0.4)
Transmission Control Protocol, Src Port: telnet (23), Dst Port: 2403 (2403), Seq: 59, Ack: 56, Len: 88
Telnet
Data: \r\n
Data: Please login to NETLAB device.\r\n
Data: Unauthorized access is prohibited.\r\n
Data: \r\n
Data: NETLAB user ID:

0000 00 15 00 34 02 f0 00 18 4d b0 d6 8c 08 00 45 00 M E.
0010 00 80 b1 c2 40 00 34 06 9b 73 92 b0 a5 e5 c0 a8 . . . @.4. . s
0020 00 04 00 17 09 63 5a 60 4f bc 63 b2 ec 4f 50 18 C' O.C. OP.
0030 16 d0 aa 74 00 00 0d 0a 50 6c 65 61 73 65 20 6c Please l
0040 6f 67 69 6e 20 74 6f 20 4e 45 54 4c 41 42 20 64 o g i n t o N E T L A B d
0050 65 76 69 63 65 2e 0d 0a 55 6e 61 75 74 68 6f 72 e v i c e U n a u t h o r i z e d
0060 69 7a 65 64 20 61 63 63 65 73 73 20 69 73 20 70 e s s i s p
0070 72 6f 68 69 62 69 74 65 64 2e 0d 0a 0d 0a 4e 45 r o h i b i t e d N E
0080 54 4c 41 42 20 75 73 65 72 20 49 44 3a 20 T L A B u s e r I D :

File: "C:\DOCUMENTS\WILLIA\LOCALS\Temp\etherXXXXa02060" 3237 KB 00:01:57 P: 4729 D: 64 M: 0 Drops: 0

PCAP

Intrusions/Policy Violations

Port sweep

Open port 10?
Open port 11?
..
Open port 8888?

A particular threat is the TCP/UDP port scanner, which scans for open ports on a host.

If an intruder finds one, it may try and connect to it.

Typical scans:
Ping sweeps.
TCP scans.
UDP scans.
OS identification scans.
Account scans.

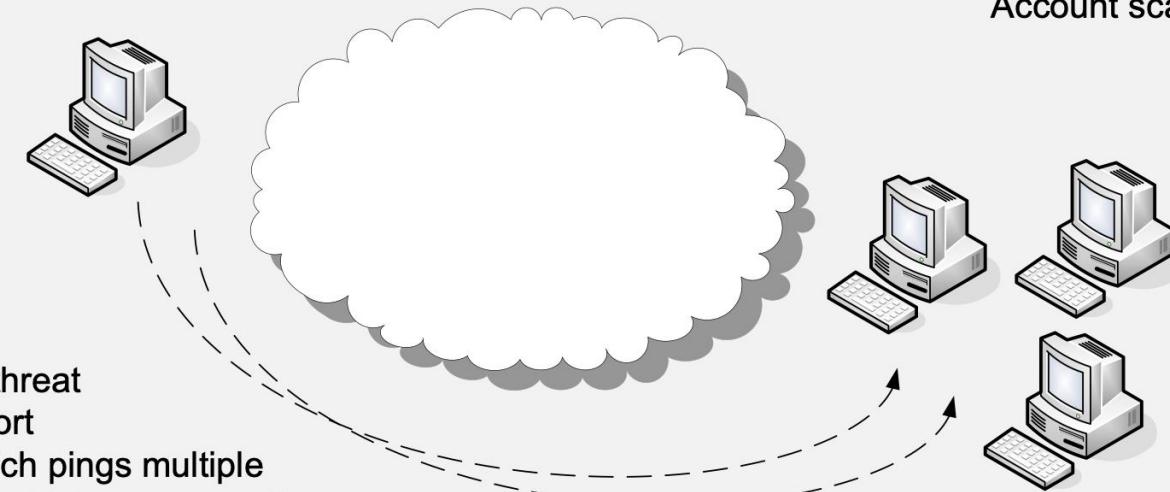
An open port is in the LISTEN state.

```
c:\log>netstat -a
Active Connections
Proto  Local Address          Foreign Address        State
TCP    bills:epmap             bills:0              LISTENING
TCP    bills:microsoft-ds      bills:0              LISTENING
TCP    bills:1035               bills:0              LISTENING
TCP    bills:3389               bills:0              LISTENING
```

ber
ata

Ping sweep

Ping 192.168.0.1?
Ping 192.168.0.1?
..
Ping 192.168.0.253?
Ping 192.168.0.254?



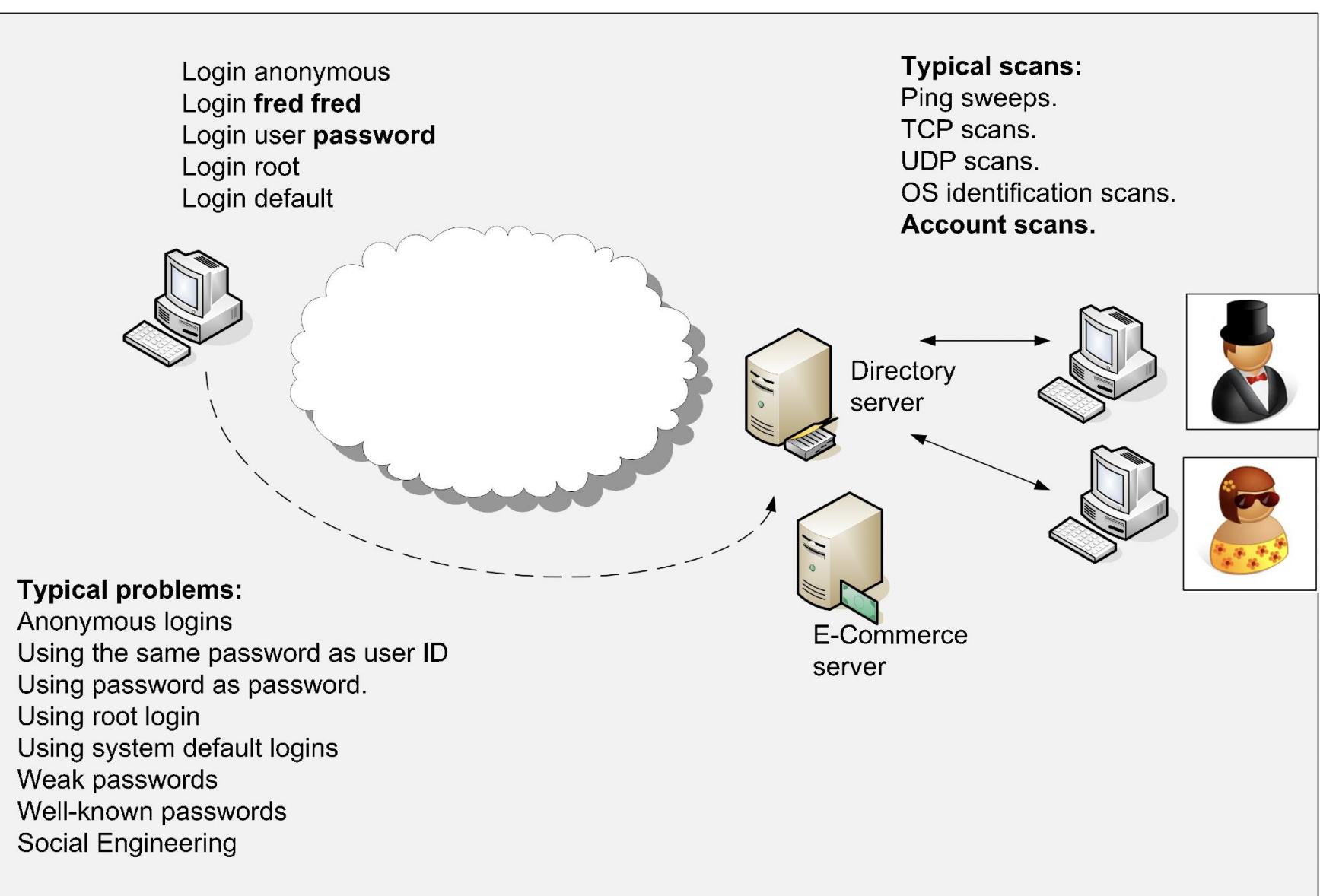
A particular threat
is the ping port
scanner, which pings multiple
hosts to see which ones are alive

If an intruder finds one, they may
try and connect to it.

Often ping (ICMP) is blocked on
the gateway of the network.

Typical scans:
Ping sweeps.
TCP scans.
UDP scans.
OS identification scans.
Account scans.

Login sweep



ber
data

Login sweep

seg8 - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: `tcp.port==21 && ip.addr==10.0.1.16`

No. Time Source Destination Protocol Info

652	6.879856	128.129.31.13	10.0.1.16	FTP	Request: USER admin
657	6.883046	10.0.1.16	128.129.31.13	FTP	Response: 331 Password re
658	6.883822	128.129.31.13	10.0.1.16	FTP	Request: PASS marietta
669	6.970991	10.0.1.16	128.129.31.13	TCP	ftp > 4197 [ACK] Seq=193
670	6.980873	10.0.1.16	128.129.31.13	TCP	ftp > 4198 [ACK] Seq=193
689	7.095867	128.129.31.13	10.0.1.16	FTP	Request: USER admin
690	7.100531	10.0.1.16	128.129.31.13	FTP	Response: 331 Password re
694	7.101144	128.129.31.13	10.0.1.16	TCP	4200 > ftp [ACK] Seq=13 A
695	7.101865	128.129.31.13	10.0.1.16	FTP	Request: PASS lorin
698	7.103811	128.129.31.13	10.0.1.16	FTP	Request: USER admin
699	7.108498	10.0.1.16	128.129.31.13	FTP	Response: 331 Password re
700	7.108912	128.129.31.13	10.0.1.16	TCP	4201 > ftp [ACK] Seq=13 A
701	7.109803	128.129.31.13	10.0.1.16	FTP	Request: PASS lorraine
702	7.110779	128.129.31.13	10.0.1.16	FTP	Request: USER admin
703	7.115535	10.0.1.16	128.129.31.13	FTP	Response: 331 Password re
704	7.116144	128.129.31.13	10.0.1.16	TCP	4202 > ftp [ACK] Seq=13 A
705	7.116858	128.129.31.13	10.0.1.16	FTP	Request: PASS louis
707	7.121790	128.129.31.13	10.0.1.16	FTP	Request: USER admin
708	7.126628	10.0.1.16	128.129.31.13	FTP	Response: 331 Password re
709	7.127067	128.129.31.13	10.0.1.16	TCP	4203 > ftp [ACK] Seq=13 A
710	7.127783	128.129.31.13	10.0.1.16	FTP	Request: PASS love
711	7.129812	128.129.31.13	10.0.1.16	FTP	Request: USER admin

IDS

FTP Username/
Password scan

Author: Prof Bill Buchanan

User account scans

Login sweep

seg8 - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: `tcp.port==21 && ip.addr==10.0.1.16`

No.	Time	Source	Destination	Protocol	Info
652	6.879856	128.129.31.13	10.0.1.16	FTP	Request: USER admin
657	6.883046	10.0.1.16	128.129.31.13	FTP	Response: 331 Password re
658	6.883822	128.129.31.13	10.0.1.16	FTP	Request: PASS marietta
659	6.883901	10.0.1.16	128.129.31.13	FTP	Response: 230 User admin

seg8 - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: `tcp.port==21 && ip.addr==10.0.1.16`

No.	Time	Source	Destination	Protocol	Info
94650	1149.666599	128.129.31.13	10.0.1.16	TCP	owserver > ftp [ACK] Seq:
94768	1151.196578	128.129.31.13	10.0.1.16	FTP	Request: USER admin
94770	1151.201368	10.0.1.16	128.129.31.13	FTP	Response: 331 Password re
94773	1151.205116	128.129.31.13	10.0.1.16	TCP	owserver > ftp [ACK] Seq:
95017	1151.205116	10.0.1.16	128.129.31.13	FTP	Request: PASS password
95019	1151.205116	128.129.31.13	10.0.1.16	FTP	Response: 230 User admin
95025	1151.205116	10.0.1.16	128.129.31.13	TCP	owserver > ftp [ACK] Seq:
95026	1151.205116	10.0.1.16	128.129.31.13	FTP	Request: SYST
95027	1153.453845	10.0.1.16	128.129.31.13	FTP	Response: 215 UNIX Type:
95030	1153.503236	128.129.31.13	10.0.1.16	TCP	owserver > ftp [ACK] Seq:
95200	1155.190379	128.129.31.13	10.0.1.16	FTP	Request: PWD
95201	1155.190903	10.0.1.16	128.129.31.13	FTP	Response: 257 "/usr/home,
95202	1155.191353	128.129.31.13	10.0.1.16	TCP	owserver > ftp [ACK] Seq:
95410	1157.539997	128.129.31.13	10.0.1.16	FTP	Request: CWD ..
95411	1157.540894	10.0.1.16	128.129.31.13	FTP	Response: 250 CWD command
95412	1157.541410	128.129.31.13	10.0.1.16	TCP	owserver > ftp [ACK] Seq:
95633	1159.491580	128.129.31.13	10.0.1.16	FTP	Request: PWD
95634	1159.492028	10.0.1.16	128.129.31.13	FTP	Response: 257 "/usr/home'
95635	1159.492642	128.129.31.13	10.0.1.16	TCP	owserver > ftp [ACK] Seq:
95780	1161.278380	128.129.31.13	10.0.1.16	FTP	Request: CWD ..
95781	1161.278939	10.0.1.16	128.129.31.13	FTP	Response: 250 CWD command
95782	1161.279441	128.129.31.13	10.0.1.16	TCP	owserver > ftp [ACK] Seq:

Successful login to
FTP server



berata

PCAP

cyber & data

“From bits to information”

Examples

FTP Detection

FTP

Add your rules here:

```
# Signature Detection  
alert tcp any any -> any 21 ( msg:"FTP";sid:10000)
```

Determine

Trace name: /log/ftp2.zip

Snort Output

Click [here](#) for the Pcap file. The Snort output is:

```
alert.ids:  
[**] [1:10000:0] FTP [**]  
[Priority: 0]  
08/31-20:24:40.417691 192.168.47.1:49430 -> 192.168.47.134:21  
TCP TTL:128 TOS:0x0 ID:16588 IpLen:20 DgmLen:52 DF  
*****S* Seq: 0x4372316F Ack: 0x0 Win: 0x2000 TcpLen: 32  
TCP Options (6) => MSS: 1460 NOP WS: 2 NOP NOP SackOK
```

FTP Detection

FTP

Add your rules here:

```
alert tcp any any -> any 21 (flags:S;msg:"FTP Connection";sid:9000005;rev:1;)
alert tcp any 21 -> any any (msg:"FTP Bad login"; content:"530 User "; nocase; flow:from_server,established; sid:491; rev:5;)
```

Determine

Trace name: /log/ftp2.zip

Snort Output

Click [here](#) for the Pcap file. The Snort output is:

```
alert.ids:
[**] [1:9000005:1] FTP Connection [**]
[Priority: 0]
08/31-20:24:40.417691 192.168.47.1:49430 -> 192.168.47.134:21
TCP TTL:128 TOS:0x0 ID:16588 IpLen:20 DgmLen:52 DF
*****S* Seq: 0x4372316F Ack: 0x0 Win: 0x2000 TcpLen: 32
TCP Options (6) => MSS: 1460 NOP WS: 2 NOP NOP SackOK
```

Telnet Detection

Hydra Telnet

Add your rules here:

```
alert tcp any any <> any 23 (flags:S; msg:"Telnet Login";sid:9000005;rev:1;)
```

Determine

Trace name: /log/hydra_telnet.zip

Snort Output

Click [here](#) for the Pcap file. The Snort output is:

```
alert.ids:  
[**] [1:9000005:1] Telnet Login [**]  
[Priority: 0]  
01/12-11:48:04.333781 192.168.47.171:7104 -> 192.168.47.200:23  
TCP TTL:128 TOS:0x0 ID:31573 IpLen:20 DgmLen:48 DF  
*****S* Seq: 0xB3747913 Ack: 0x0 Win: 0xFFFF TcpLen: 28  
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

File type Detection

Email with GIF

Add your rules here:

```
alert tcp any any -> any any (content:"GIF89a"; msg:"GIF";sid:10000)
alert tcp any any -> any any (content:"%PDF"; msg:"PDF";sid:10001)
alert tcp any any -> any any (content:"|89 50 4E 47|"; msg:"PNG";sid:10002)
alert tcp any any -> any any (content:"|50 4B 03 04|"; msg:"ZIP";sid:10003)
alert tcp any any -> any any (content:"|FF D8|"; msg:"JPEG";sid:10004)
alert tcp any any -> any any (content:"|49 44 33|"; msg:"MP3";sid:10005)
alert tcp any any -> any any (content:"|52 49 46 46|"; msg:"AVI";sid:10006)
alert tcp any any -> any any (content:"|46 57 53|"; msg:"Flash SWF";sid:10007)
alert tcp any any -> any any (content:"|46 4C 56|"; msg:"Flash Video";sid:10008)
alert tcp any any -> any any (content:"|1F 8B 08|"; msg:"GZip";sid:10009)
alert tcp any any -> any any (content:"|52 61 72 21 1A 07 00|"; msg:"RAR";sid:10010)
alert tcp any any -> any any (content:"|D0 CF 11 E0 A1 B1 1A E1|"; msg:"Office 2010";sid:10011)
```

Determine

Trace name: /log/with_gif.zip

Snort Output

Click [here](#) for the Pcap file. The Snort output is:

```
alert.ids:
[**] [1:10000:0] GIF [**]
[Priority: 0]
01/05-19:38:04.190265 77.72.118.168:80 -> 192.168.47.171:2641
TCP TTL:128 TOS:0x0 ID:61162 IpLen:20 DgmLen:83
```

Credit Card Detection

Email with credit card details

Add your rules here:

```
# Detecting credit card details
alert tcp any any <> any any (pcre:"/5\d{3}(\s|-)?\d{4}(\s|-)?\d{4}(\s|-)?\d{4}/"; \
    msg:"MasterCard number detected in clear text";content:"number";nocase;sid:9000003;rev:1;)

alert tcp any any <> any any (pcre:"/3\d{3}(\s|-)?\d{6}(\s|-)?\d{5}/"; \
    msg:"American Express number detected in clear text";content:"number";nocase;sid:9000004;rev:1;)

alert tcp any any <> any any (pcre:"/4\d{3}(\s|-)?\d{4}(\s|-)?\d{4}(\s|-)?\d{4}/"; \
    msg:"Visa number detected in clear text";content:"number";nocase;sid:9000005;rev:1;)
```

Determine

Trace name: /log/email_cc2.zip

Snort Output

Click [here](#) for the Pcap file. The Snort output is:

```
alert.ids:
[**] [1:9000005:1] Visa number detected in clear text [**]
[Priority: 0]
01/06-21:20:26.755456 192.168.47.171:1061 -> 192.168.47.134:25
TCP TTL:128 TOS:0x0 ID:628 Iplen:20 DgmLen:1500 DF
***A*** Seq: 0xCA178C7B Ack: 0x91870925 Win: 0xEF9 TcpLen: 20
```

ICMP Detection

Ping sweep

Add your rules here:

```
alert icmp any any -> any any (msg:"ICMP Packet found";sid:9000000;)
alert icmp any any -> any any (ctype: 0; msg: "ICMP Echo Reply";sid:9000001;)
alert icmp any any -> any any (ctype: 3; msg: "ICMP Destination Unreachable";sid:9000002;)
alert icmp any any -> any any (ctype: 4; msg: "ICMP Source Quench Message received";sid:9000003;)
alert icmp any any -> any any (ctype: 5; msg: "ICMP Redirect message";sid:9000004;)
alert icmp any any -> any any (ctype: 8; msg: "ICMP Echo Request";sid:9000005;)
alert icmp any any -> any any (ctype: 11; msg: "ICMP Time Exceeded";sid:9000006;)
```

Determine

Trace name: /log/ping_sweep.zip

Snort Output

Click [here](#) for the Pcap file. The Snort output is:

```
alert.ids:
[**] [1:9000005:0] ICMP Echo Request [**]
[Priority: 0]
08/25-15:48:52.876833 192.168.47.1 -> 192.168.47.2
ICMP TTL:59 TOS:0x0 ID:57989 IpLen:20 DgmLen:28
Type:8 Code:0 ID:12928 Seq:0 ECHO
```



“From bits to information”

Intrusion
Detection
Systems

Port scanning

```
SCAN.RULE
preprocessor flow: stats_interval 0 hash 2
preprocessor sfportscan: proto { all } scan_type { all }
sense_level { low } logfile { portscan.log }
```

```
C:\> snort -c scan.rule -dev -i 3 -p -l c:\\bill -K ascii
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file scan.rule
-----[Flow Config]-----
| Stats Interval: 0
| Hash Method: 2
| Memcap: 10485760
| Rows : 4096
| Overhead Bytes: 16388(%0.16)

Portscan Detection Config:
 Detect Protocols: TCP UDP ICMP IP
 Detect Scan Type: portscan portsweep decoy_portscan distributed_portscan
 Sensitivity Level: Low
 Memcap (in bytes): 1048576
 Number of Nodes: 3869
 Logfile: c:\\bill/portscan.log

Tagged Packet Limit: 256
...
C:\>nmap -o -A 192.168.0.1
Starting Nmap 4.20 ( http://insecure.org ) at 2007-01-09 21:58 GMT Standard Time
Interesting ports on 192.168.0.1:
Not shown: 1695 closed ports
PORT      STATE SERVICE
80/tcp    open  http
8888/tcp  open  sun-answerbook
MAC Address: 00:0B:44:F5:33:D5 (The Linksys Group)
Nmap finished: 1 IP address (1 host up) scanned in 1.500 seconds
```

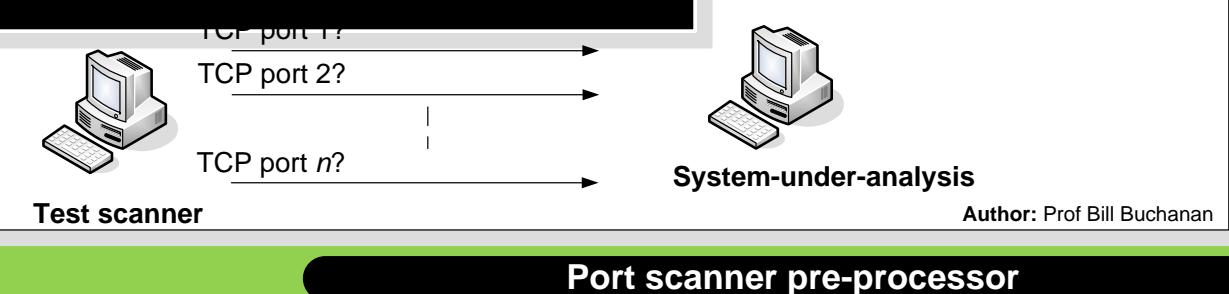
NOTE:

The NMAP program should only be used on machines which you are under control of, and in a local, and isolated environment. It should only be used to determine possible weaknesses and vulnerabilities.

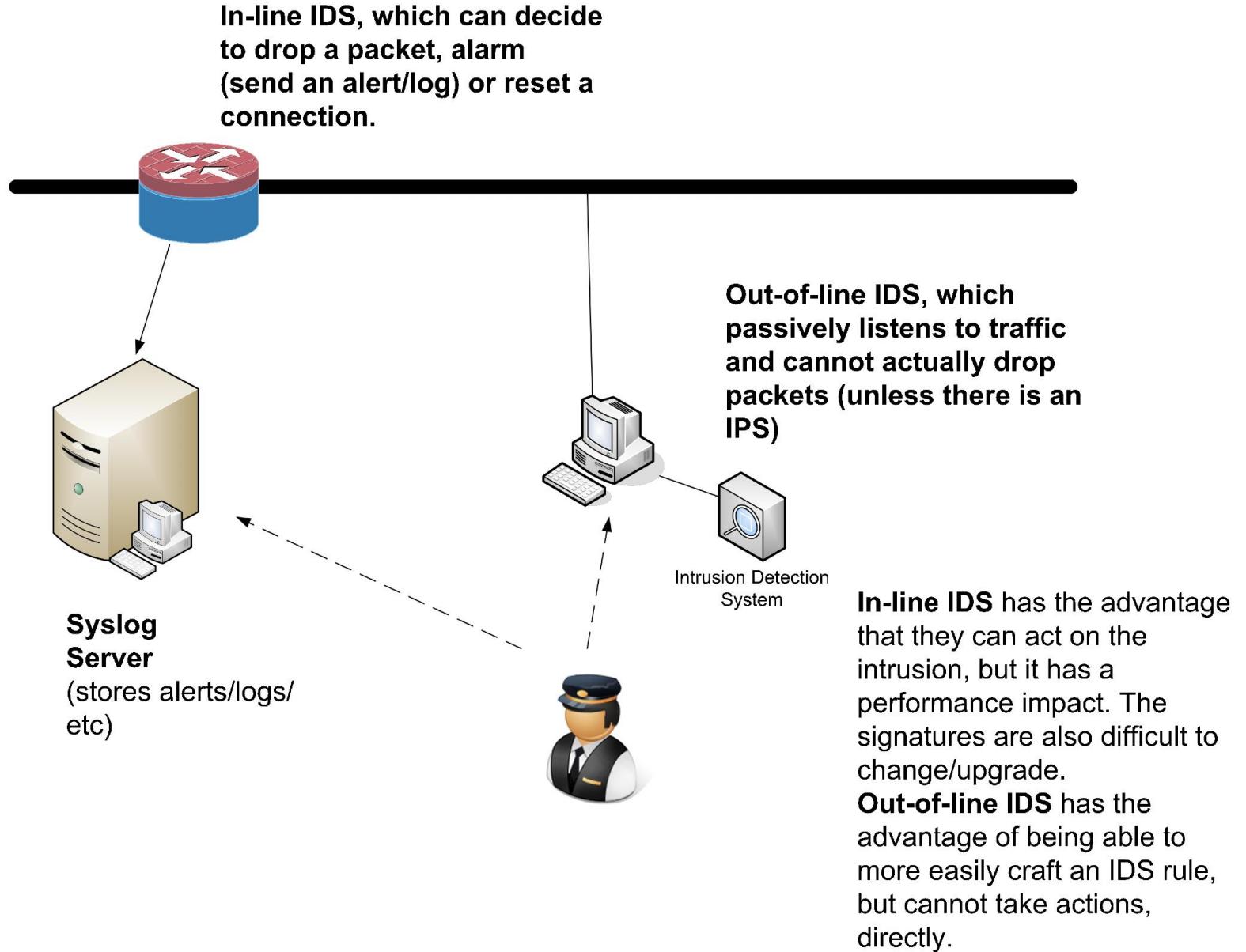
```
PORSCAN.LOG
Time: 08/17-14:41:54.495296
event_ref: 0
192.168.0.3 -> 64.13.134.49 (portscan) TCP Portsweep
Priority Count: 5
Connection Count: 135
IP Count: 43
Scanned IP Range: 64.13.134.49:216.239.59.99
Port/Proto Count: 1
Port/Proto Range: 80:80

Time: 08/17-14:42:52.431092
event_ref: 0
192.168.0.3 -> 192.168.0.1 (portscan) TCP Portsweep
Priority Count: 5
Connection Count: 10
IP Count: 5
Scanned IP Range: 66.249.93.165:192.168.0.7
Port/Proto Count: 3
Port/Proto Range: 80:2869

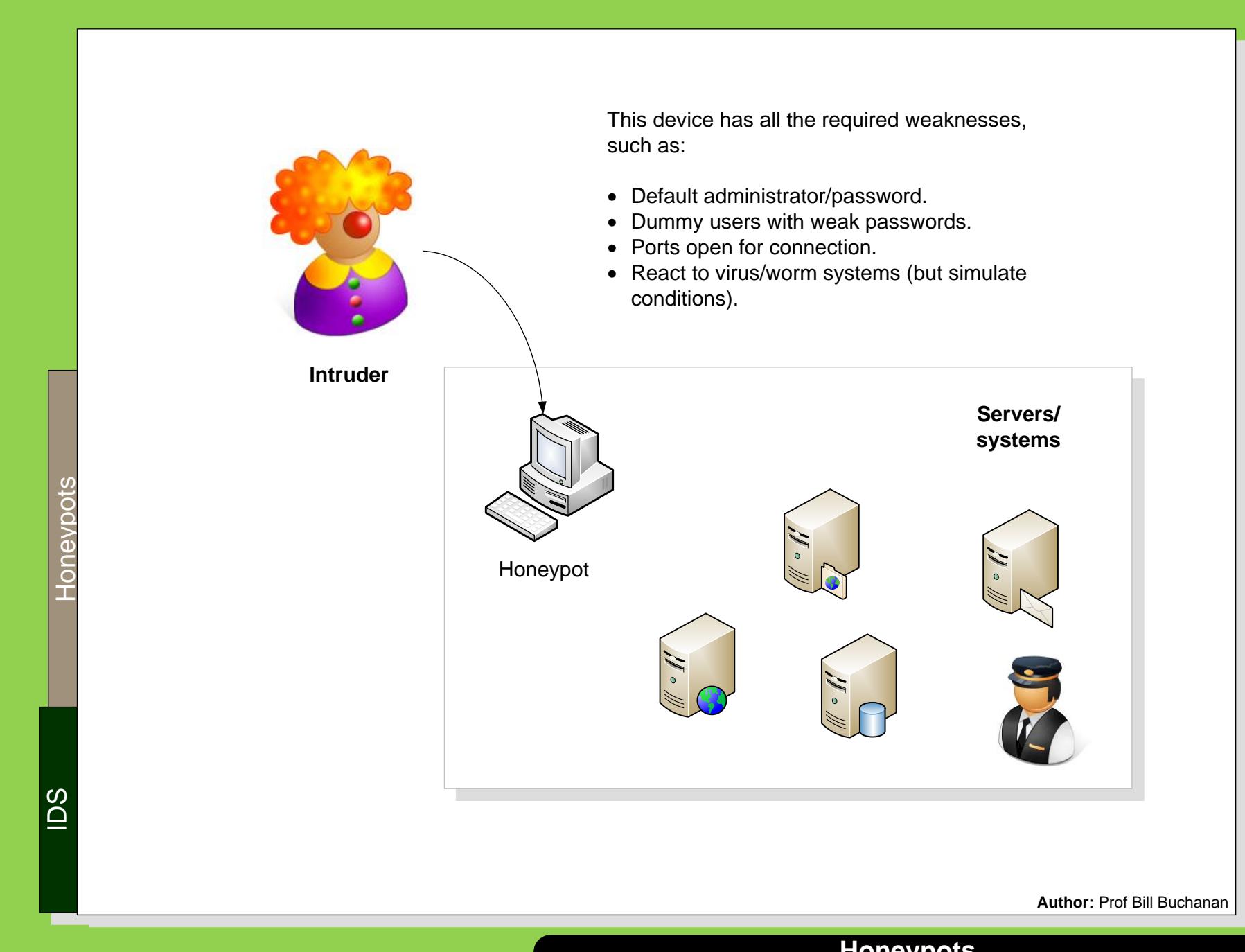
Time: 08/17-14:42:52.434852
event_ref: 0
192.168.0.3 -> 192.168.0.1 (portscan) TCP Portscan
Priority Count: 5
Connection Count: 9
IP Count: 1
Scanner IP Range: 192.168.0.3:192.168.0.3
Port/Proto Count: 10
Port/Proto Range: 21:636
```



In-line/out-of-line



Honeypot



Honeypot

IDS

Honeypots



High-interaction honeypot. This simulates all the aspects of the operating system

Open ports: 110 (POP-3), 80 (HTTP), 21 (FTP), 22 (SSH)



Low-interaction honeypot. This simulates only part of the network stack (such as for **Honeyd**)

- can be virtual (from a virtual machine) or simulated by another machine.



Honeyd.conf

```
create default
set default personality "windows XP"
set default default tcp action reset
add default tcp port 110 "sh scripts/pop.sh"
add default tcp port 80 "perl scripts/iis-0.95/main.pl"
add default tcp port 25 block
add default tcp port 21 "sh scripts/ftp.sh"
add default tcp port 22 proxy $ipsrc:22
add default udp port 139 drop
set default uptime 3284460

### Cisco router
create router
set router personality "Cisco PIX Firewall (PixOS 5.2 - 6.1)"
add router tcp port 23 "/usr/bin/perl scripts/router-telnet.pl"
set router default tcp action reset
set router uid 32767 gid 32767
set router uptime 1327650
# Bind specific templates to specific IP address
# If not bound, default to windows template
bind 192.168.1.150 router
```

Author: Prof Bill Buchanan

Honeypot types



“From bits to information”

Intrusion
Detection
Systems