cyber & data

"From bits to information"

Intrusion Detection Systems

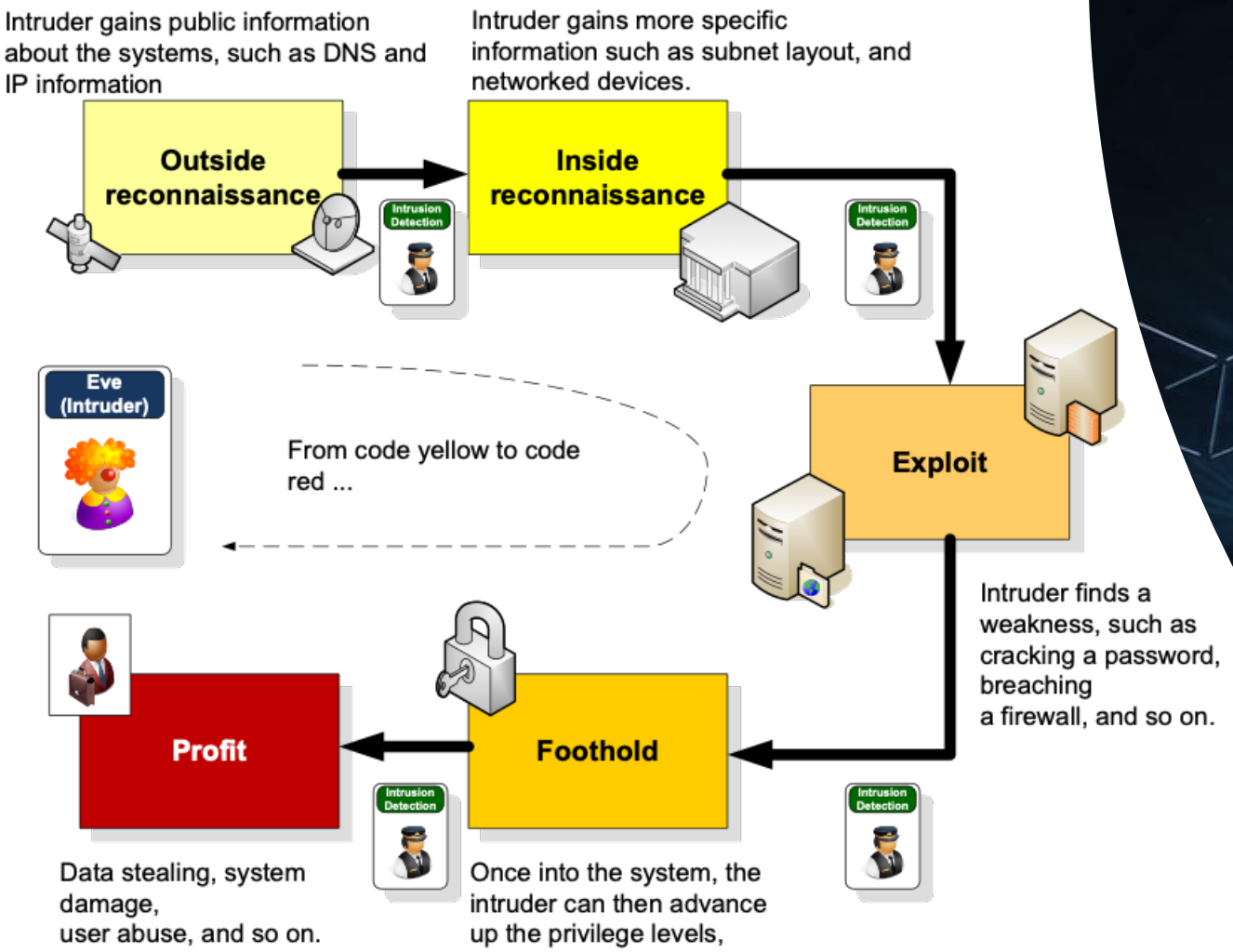
Outline

- Attack Patterns.
- SNORT.
- IDS Types

"From bits to information"

Attack Patterns

Intrusion Patterns



Kill Chain graphic: https://en.wikipedia.org/wiki/Kill_chain

&cyber data

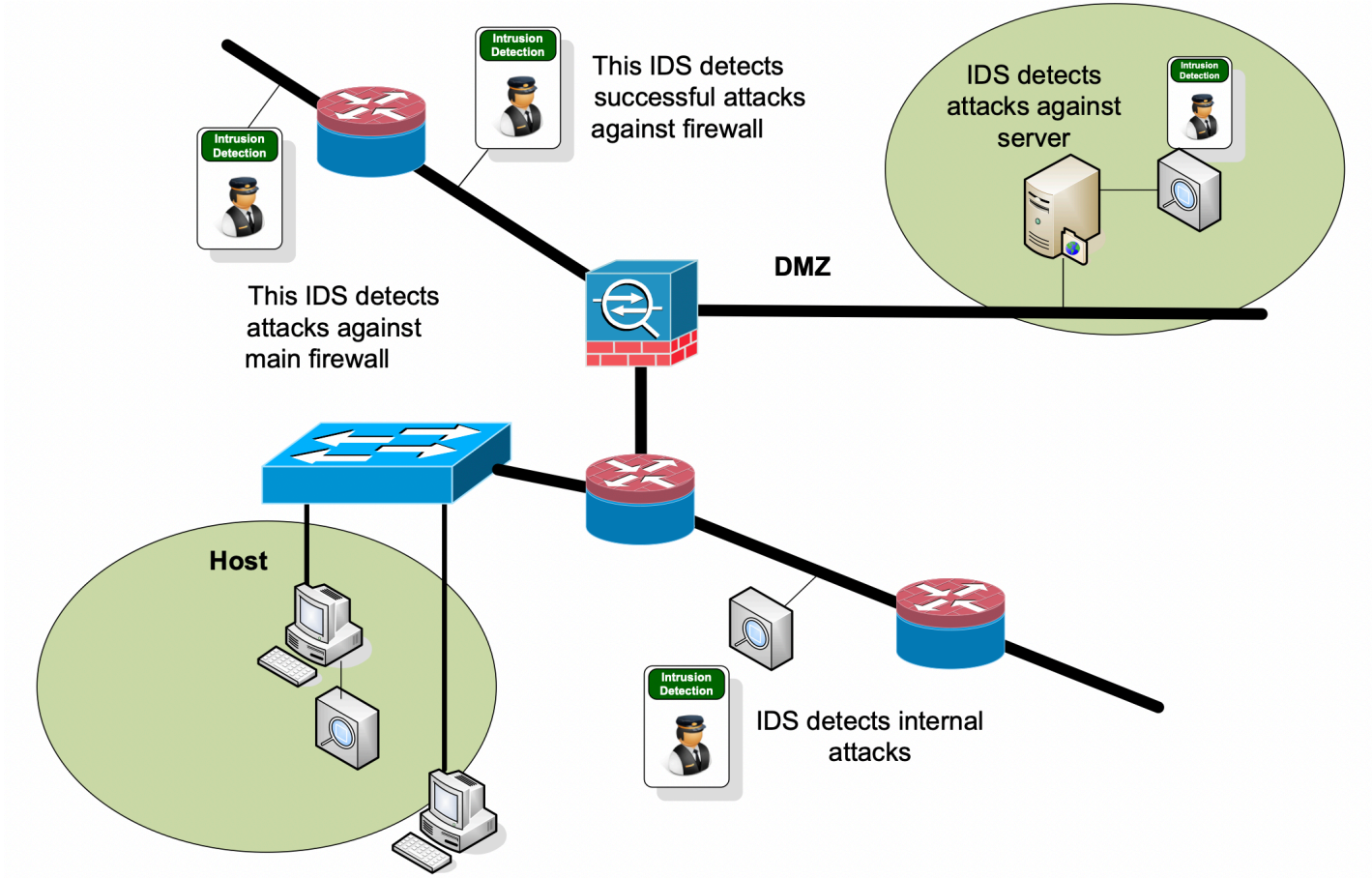
Intrusion Patterns

Phases of the Intrusion Kill Chain

Reconnaissance	Research, identification, and selection of targets
Weaponization	Pairing remote access malware with exploit into a deliverable payload (e.g. Adobe PDF and Microsoft Office files)
Delivery	Transmission of weapon to target (e.g. via email attachments, websites, or USB drives)
Exploitation	Once delivered, the weapon's code is triggered, exploiting vulnerable applications or systems
Installation	The weapon installs a backdoor on a target's system allowing persistent access
Command & Control	Outside server communicates with the weapons providing "hands on keyboard access" inside the target's network.
Actions on Objective	The attacker works to achieve the objective of the intrusion, which can include exfiltration or destruction of data, or intrusion of another target

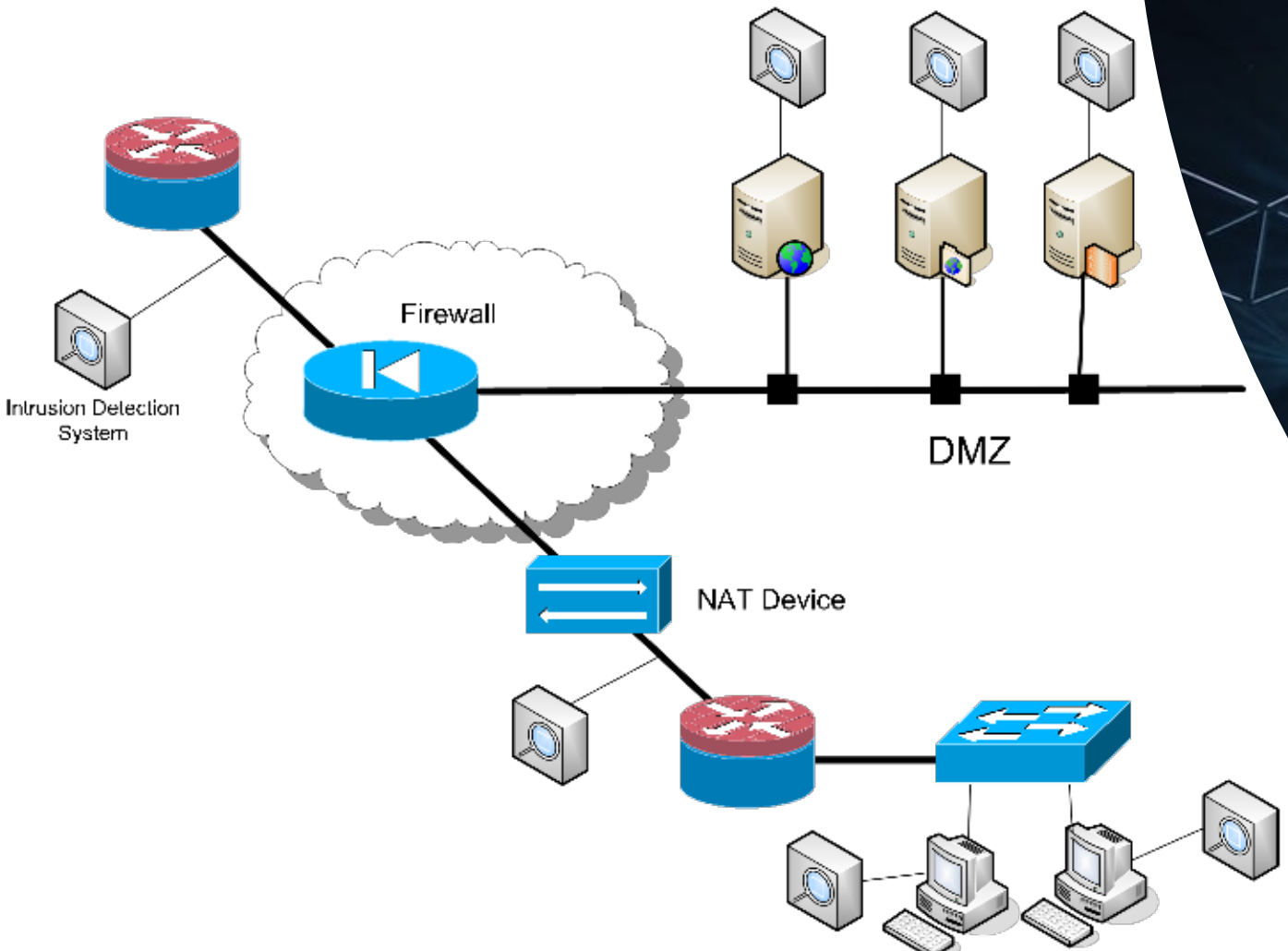
Kill Chain graphic: https://en.wikipedia.org/wiki/Kill_chain

Host-based or Network-based



&cyber data

IDS Placement

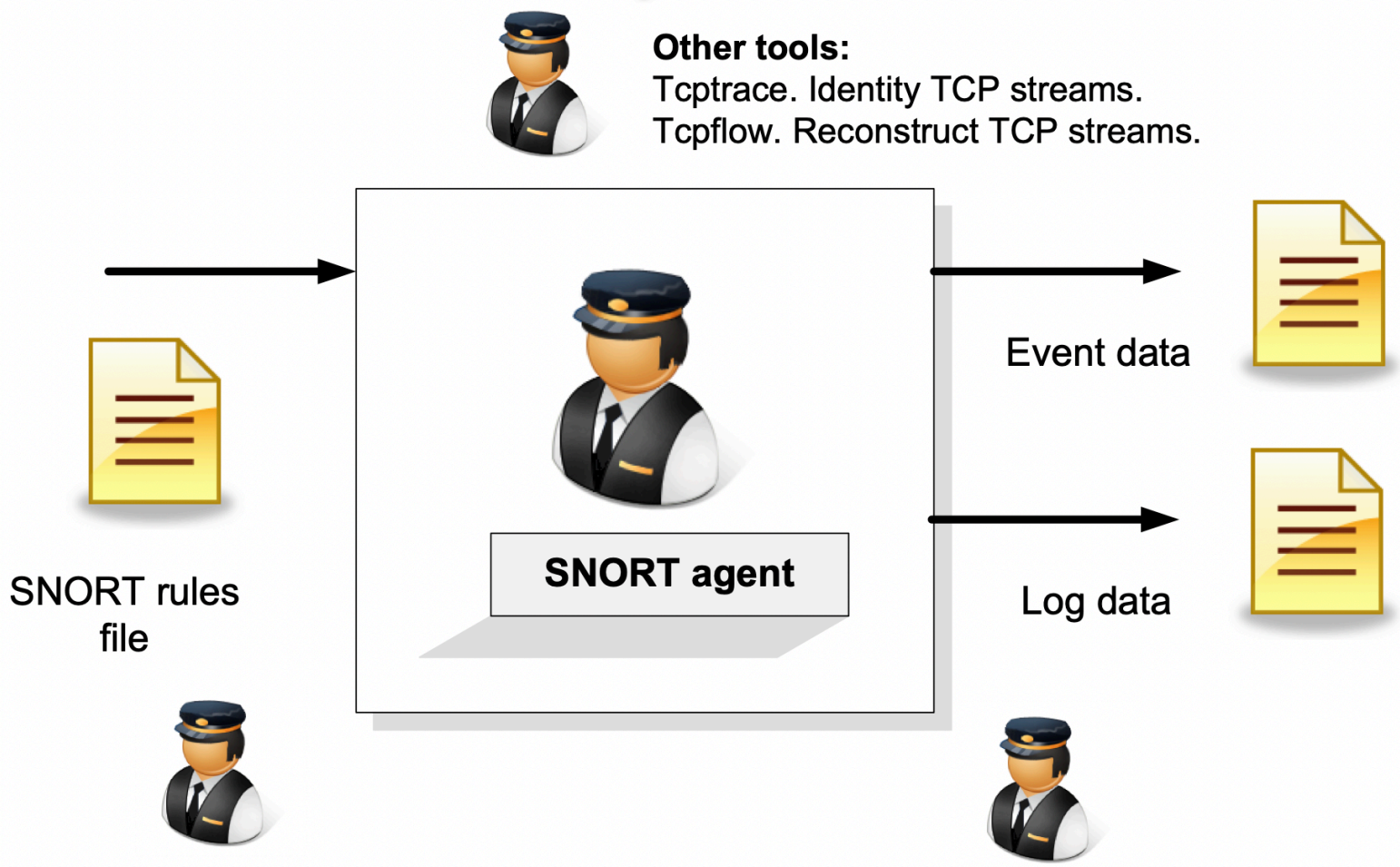


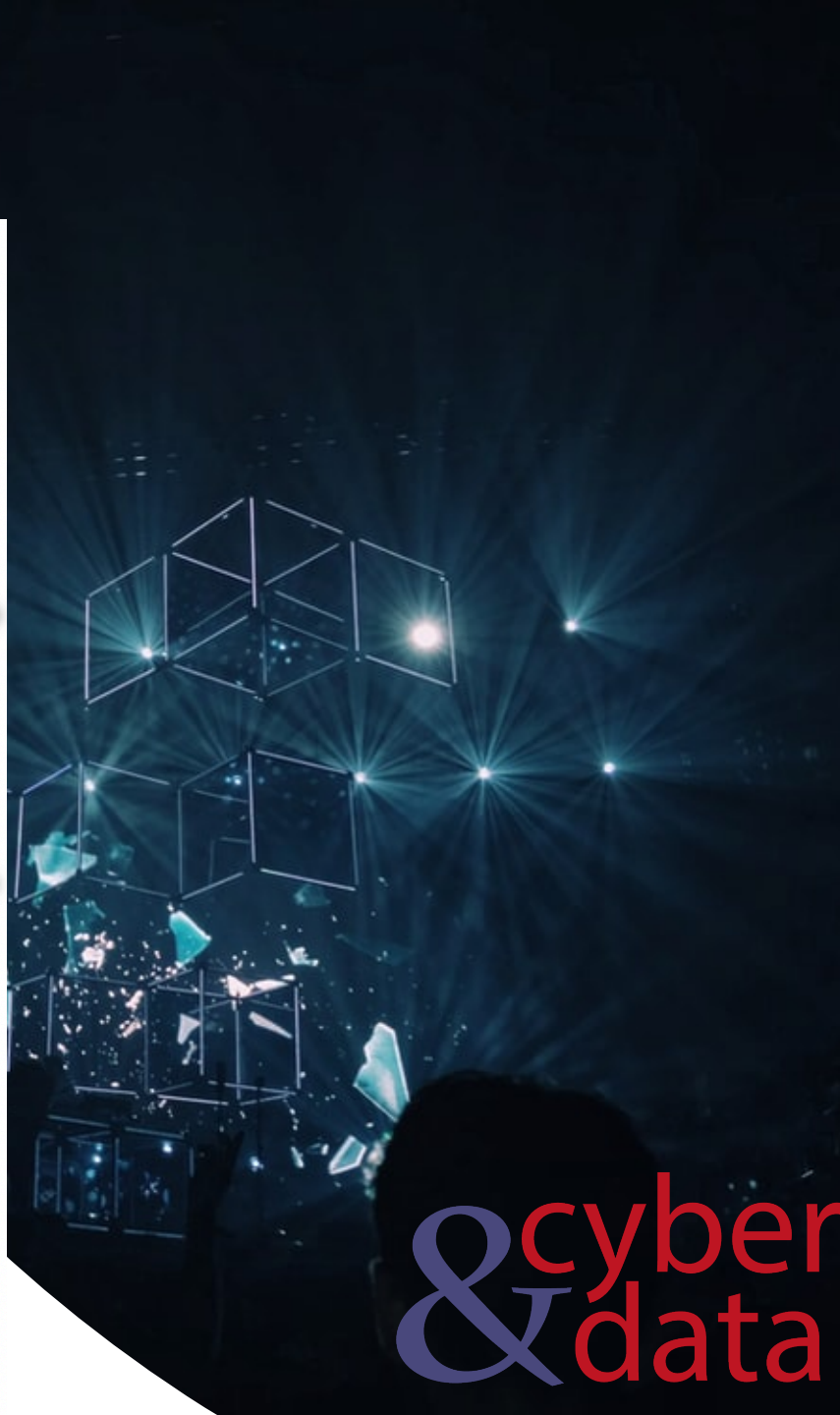
&cyber data

“From bits to information”

Snort

Snort


Other tools:
Tcptrace. Identity TCP streams.
Tcpflow. Reconstruct TCP streams.
Event data
SNORT agent
SNORT rules file
Log data
Signature detection.
Identify well-known patterns of attack.
Anomaly detection.
Statistical anomalies, such as user logins, changes to files, and so on.


&cyber data

Example of Snort Rule

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

Source IP

Source port
(any,
or m:n for m to n)

Destination IP

Destination port
(any,
or m:n for m to n)

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

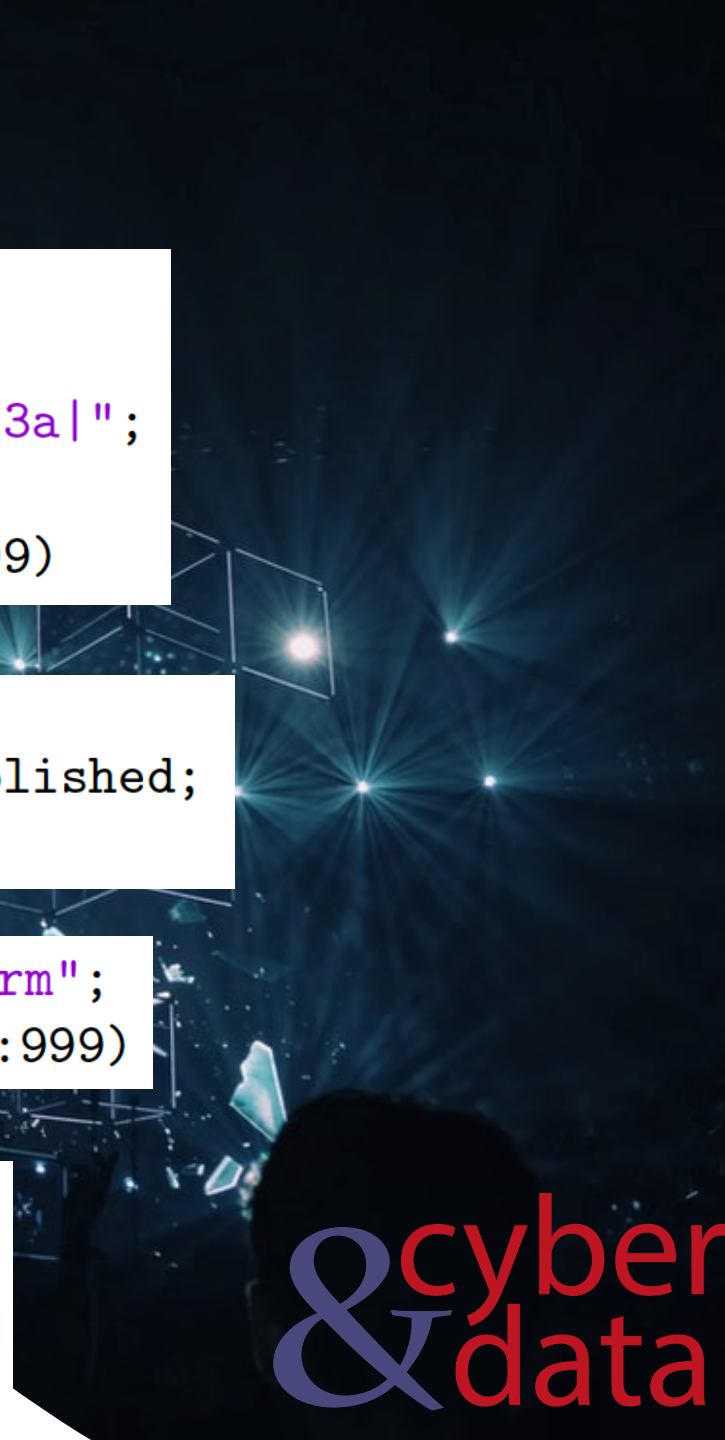
Example Snort Rule

```
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
```

"From bits to information"

A Simple Rule

Snort Rule

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

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

Alert.ids
(in \log)

16 January 10:27pm

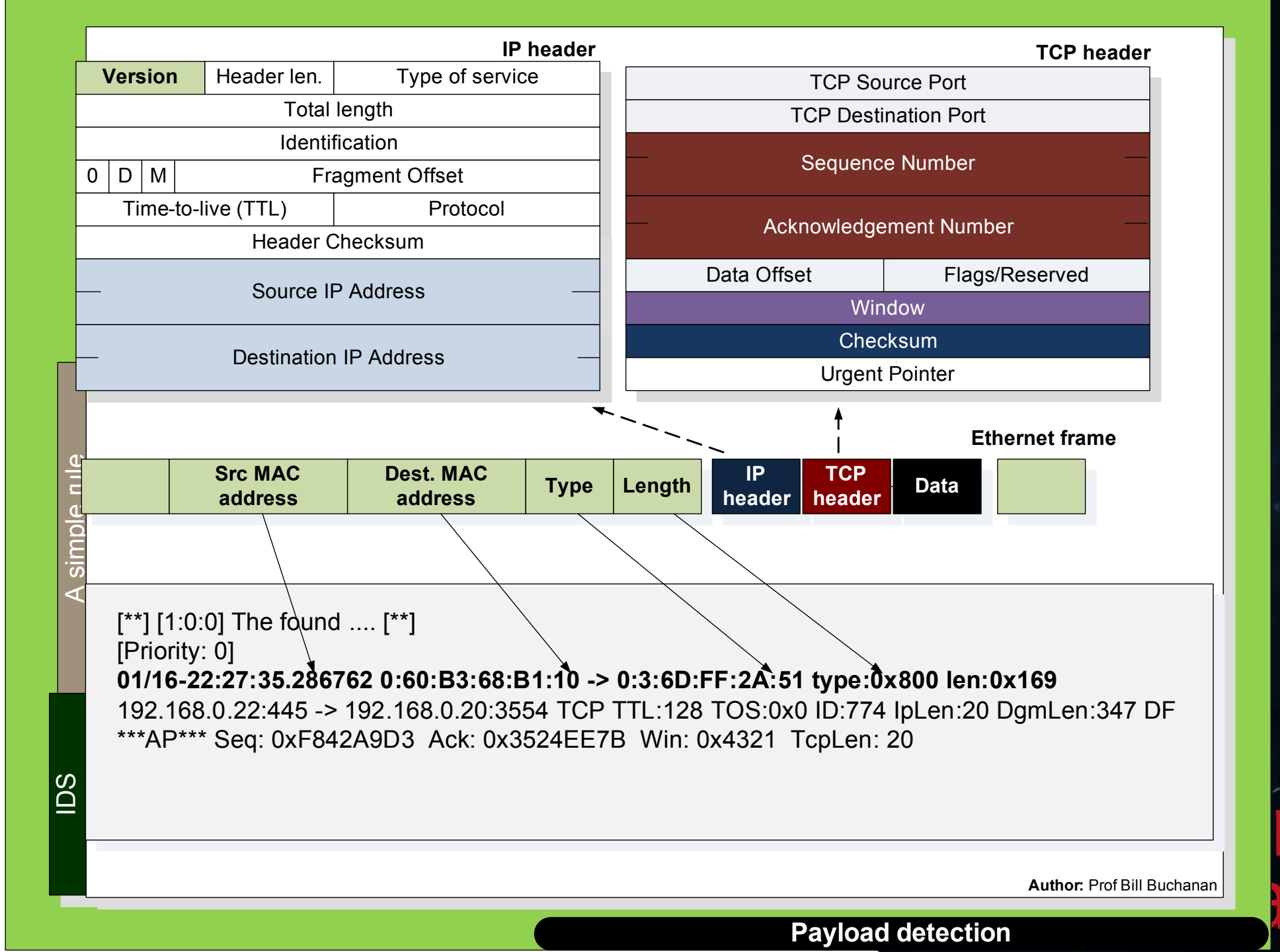
```
[**] [1:0:0] The found .... [**]
[Priority: 0]
01/16-22:27:35.286762 0:60:B3:68:B1:10 -> 0:3:6D:FF:2A:51 type:0x800 len:0x169
192.168.0.22:445 -> 192.168.0.20:3554 TCP TTL:128 TOS:0x0 ID:774 IpLen:20
DgmLen:347 DF
***AP*** Seq: 0xF842A9D3  Ack: 0x3524EE7B  Win: 0x4321  TcpLen: 20

[**] [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
```

A simple rule

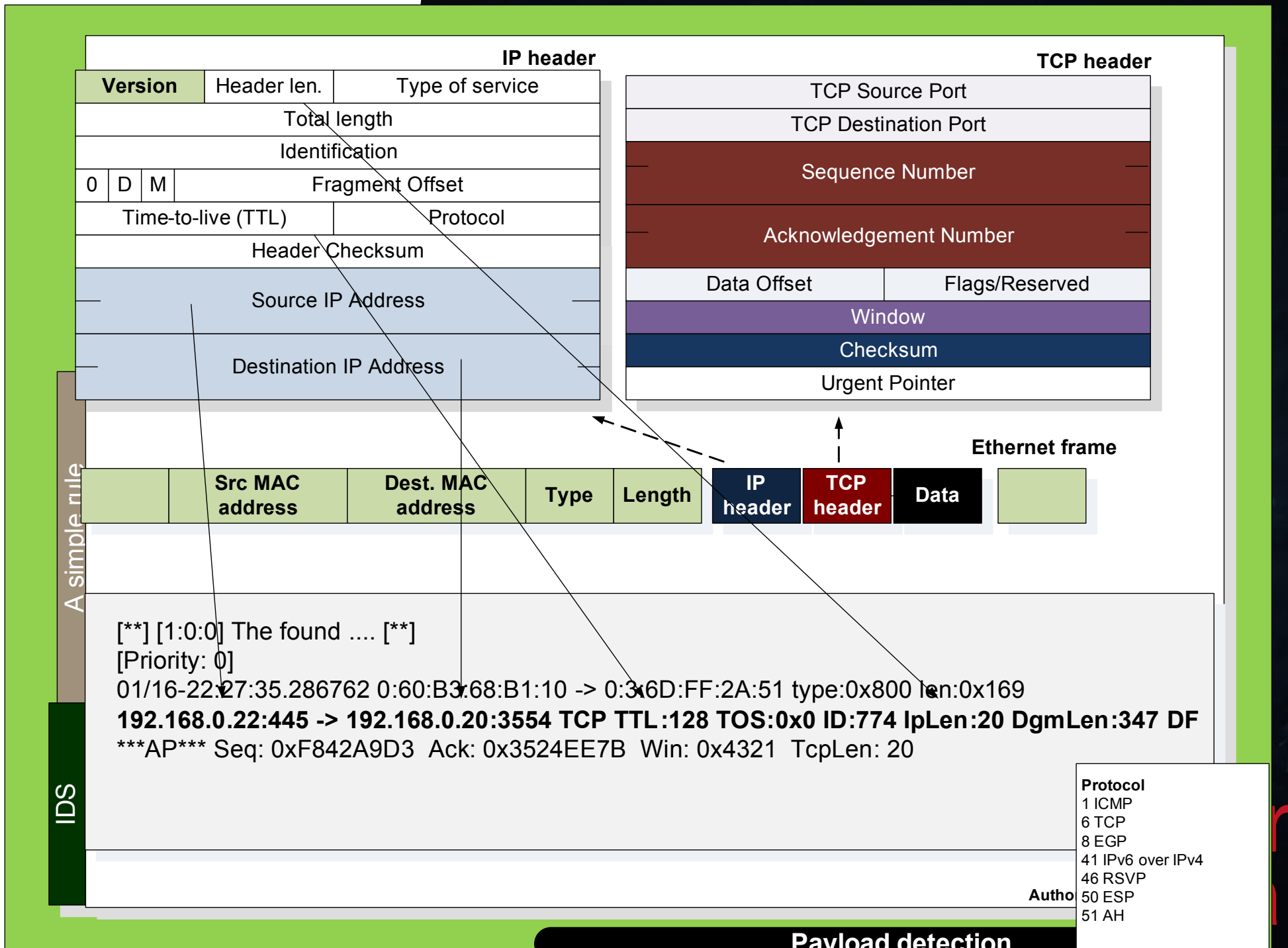
IDS

Author: Prof Bill Buchanan

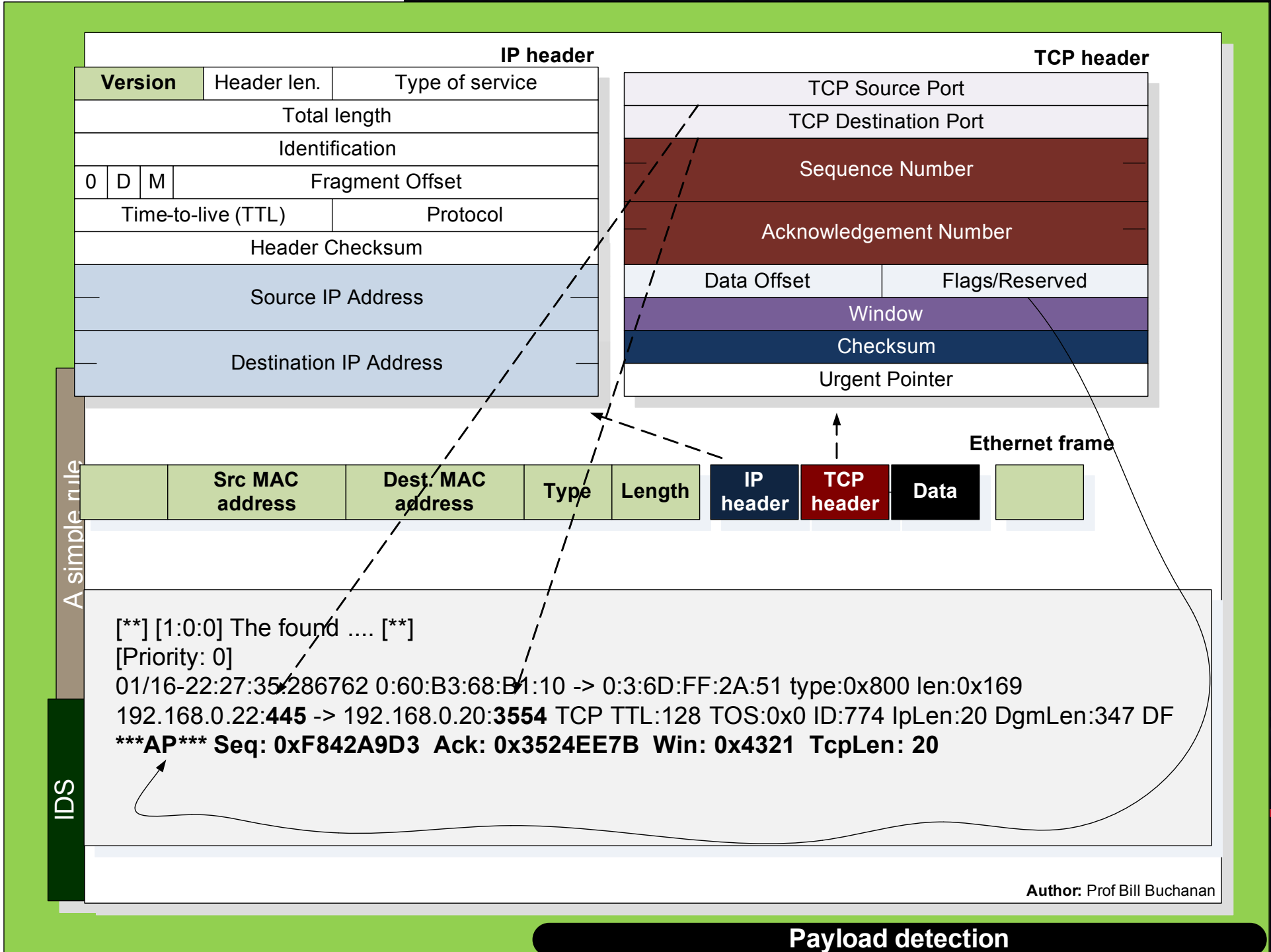
Snort Rule


IP header
Version
Header len.
Type of service
Total length
Identification
0
D
M
Fragment Offset
Time-to-live (TTL)
Protocol
Header Checksum
Source IP Address
Destination IP Address
TCP header
TCP Source Port
TCP Destination Port
Sequence Number
Acknowledgement Number
Data Offset
Flags/Reserved
Window
Checksum
Urgent Pointer
Ethernet frame
Src MAC address
Dest. MAC address
Type
Length
IP header
TCP header
Data
A simple rule
IDS
[**] [1:0:0] The found .... [**]
[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
Author: Prof Bill Buchanan
Payload detection

Snort Rule


IP header
Version
Header len.
Type of service
Total length
Identification
0
D
M
Fragment Offset
Time-to-live (TTL)
Protocol
Header Checksum
Source IP Address
Destination IP Address
TCP header
TCP Source Port
TCP Destination Port
Sequence Number
Acknowledgement Number
Data Offset
Flags/Reserved
Window
Checksum
Urgent Pointer
Ethernet frame
Src MAC address
Dest. MAC address
Type
Length
IP header
TCP header
Data
A simple rule
IDS
[**] [1:0:0] The found .... [**]
[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
Protocol
1 ICMP
6 TCP
8 EGP
41 IPv6 over IPv4
46 RSVP
50 ESP
51 AH

Snort Rule


IP header
Version
Header len.
Type of service
Total length
Identification
0
D
M
Fragment Offset
Time-to-live (TTL)
Protocol
Header Checksum
Source IP Address
Destination IP Address
TCP header
TCP Source Port
TCP Destination Port
Sequence Number
Acknowledgement Number
Data Offset
Flags/Reserved
Window
Checksum
Urgent Pointer
Ethernet frame
Src MAC address
Dest. MAC address
Type
Length
IP header
TCP header
Data
A simple rule
IDS
[**] [1:0:0] The found .... [**]
[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
Author: Prof Bill Buchanan
Payload detection

cyber & data

"From bits to information"

A Few Intrusions

An example

A few intrusions

IDS

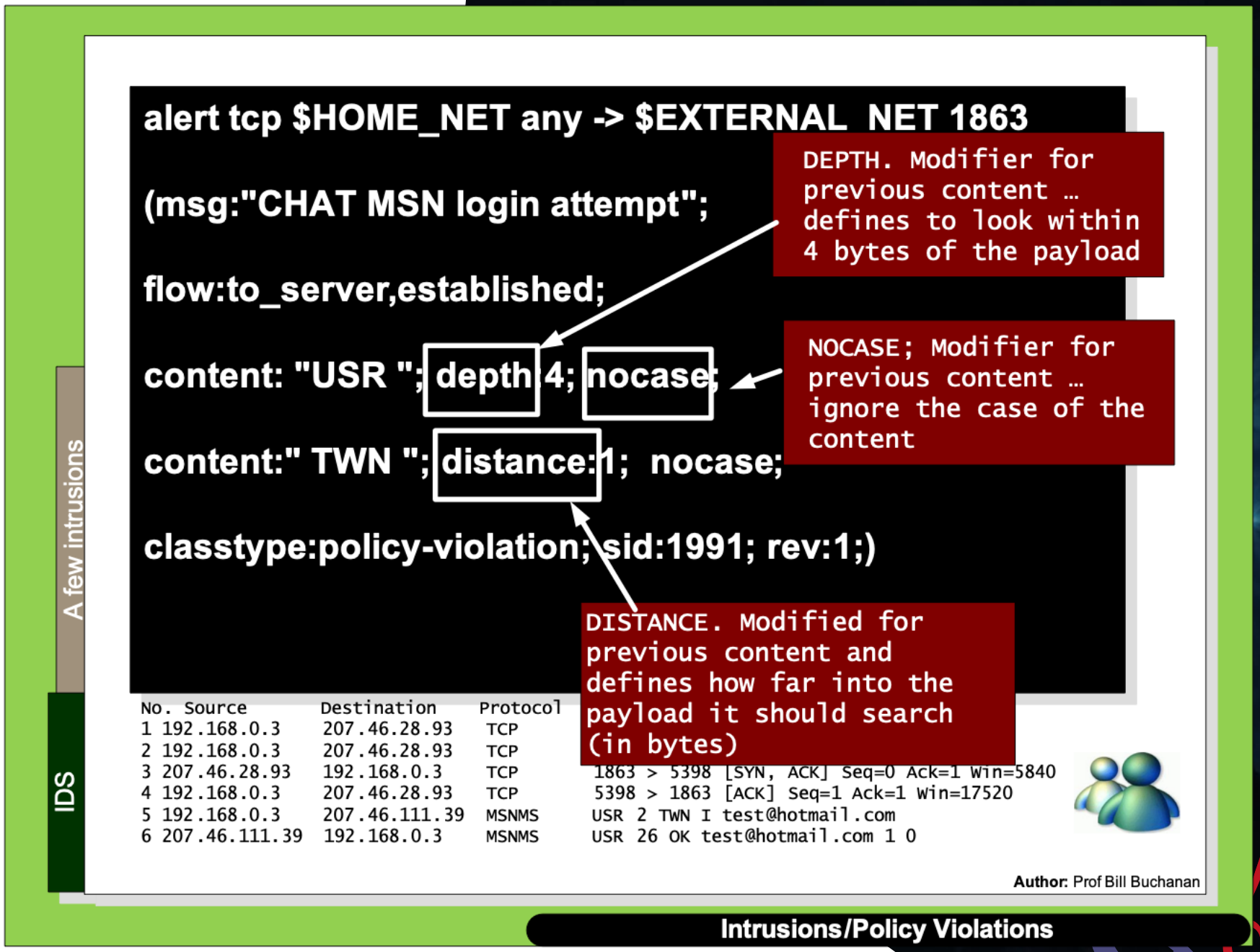
```
alert tcp $HOME_NET any -> $EXTERNAL_NET 1863
(msg:"CHAT MSN login attempt";  flow:to_server,established; content:"USR "; depth:4;
nocase; content:" TWN "; distance:1;  nocase;
classtype:policy-violation; sid:1991; rev:1;)
```

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

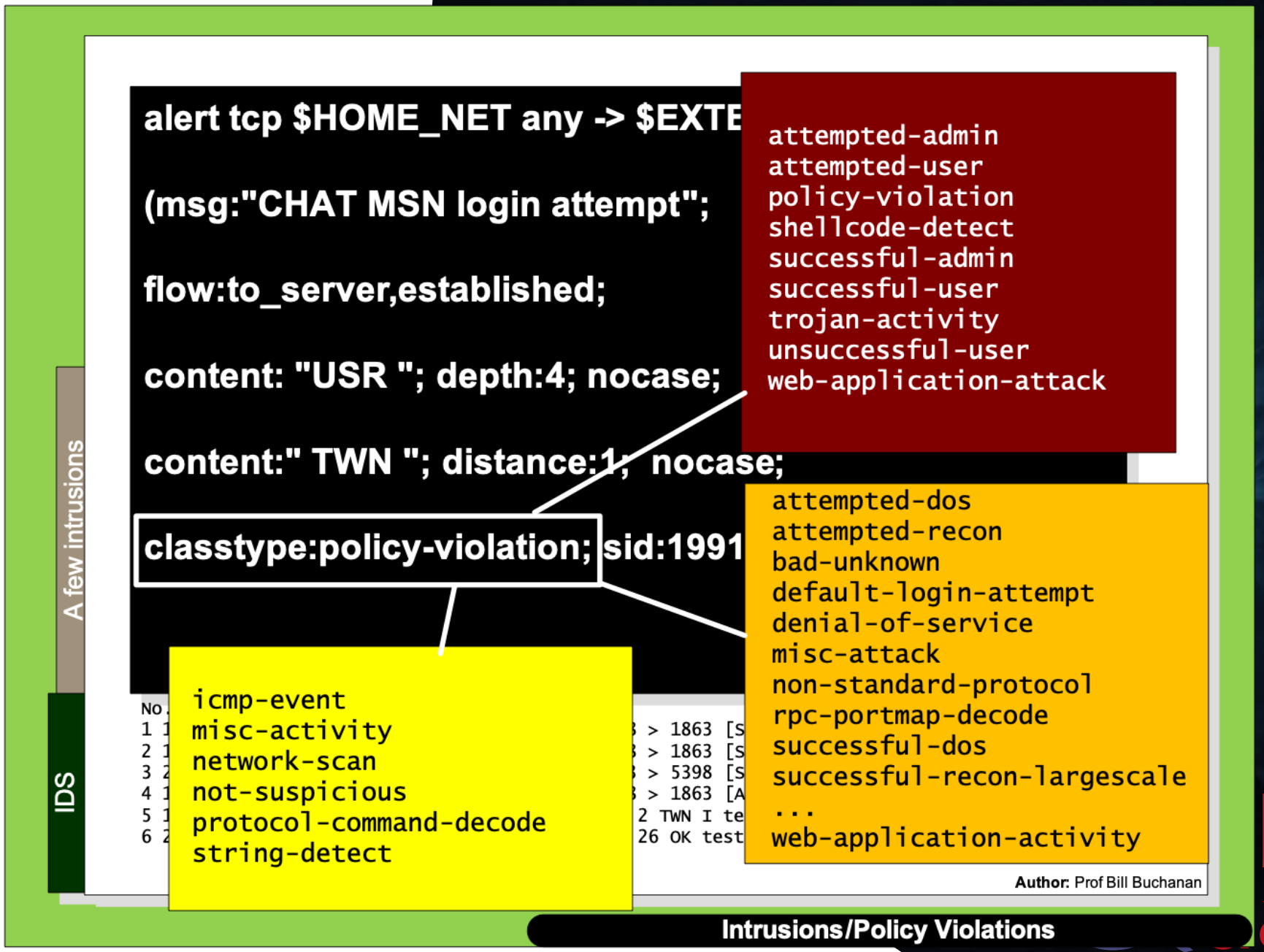
```
private static void device_PcapOnPacketArrival(…)
{
  if(packet is TCPPacket)
  {
   TCPPacket tcp = (TCPPacket)packet;
   int destPort = tcp.SourcePort; byte [] b = tcp.Data;
   ASCIIEncoding format = new ASCIIEncoding();
   string s = format.GetString(b); s=s.ToLower();
   if (destPort==1863 &&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

Author: Prof Bill Buchanan

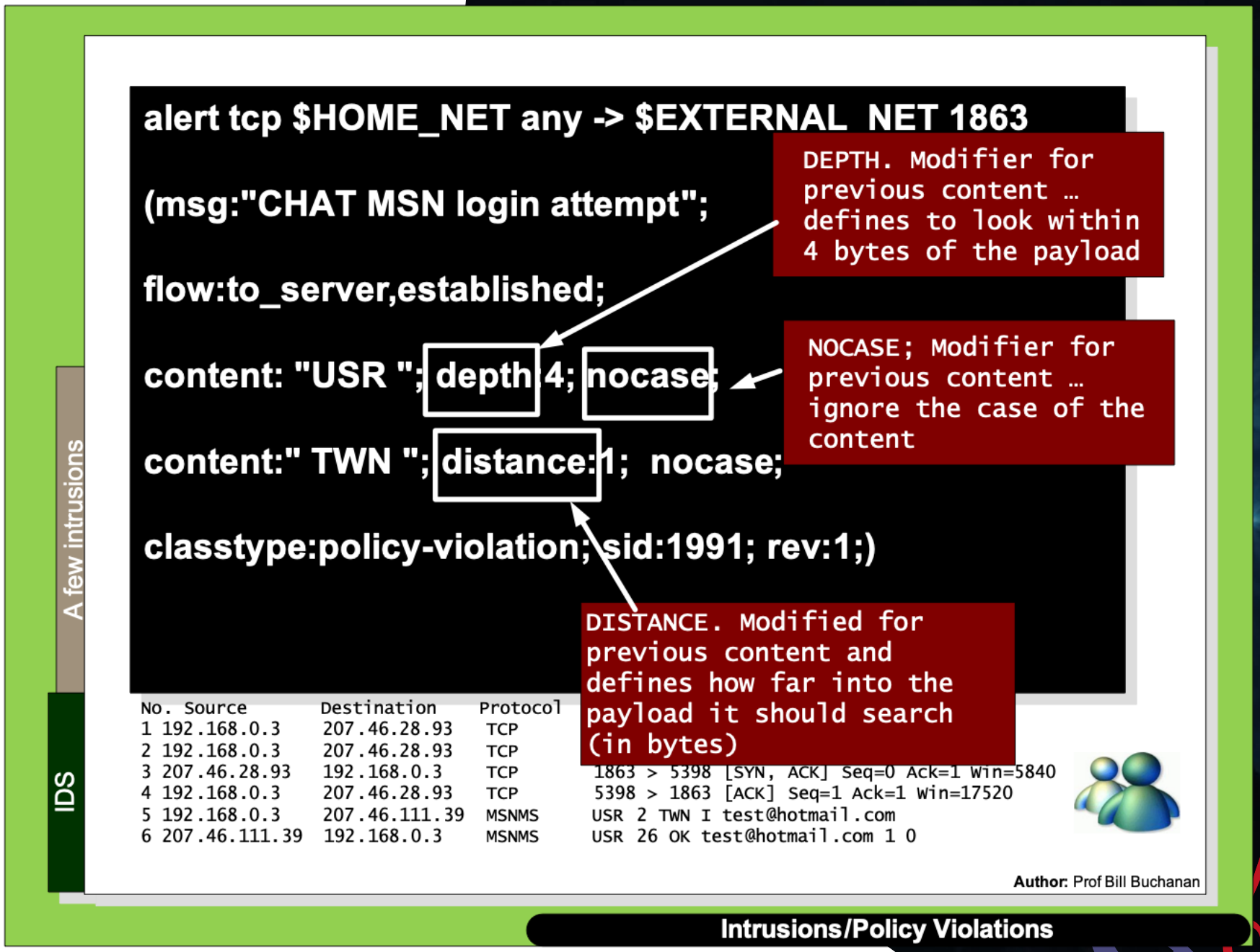
Rules


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;)
DEPTH. Modifier for previous content … defines to look within 4 bytes of the payload
NOCASE; Modifier for previous content … ignore the case of the content
DISTANCE. Modified for previous content and defines how far into the payload it should search (in bytes)
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
A few intrusions
IDS
Author: Prof Bill Buchanan
Intrusions/Policy Violations

Rules

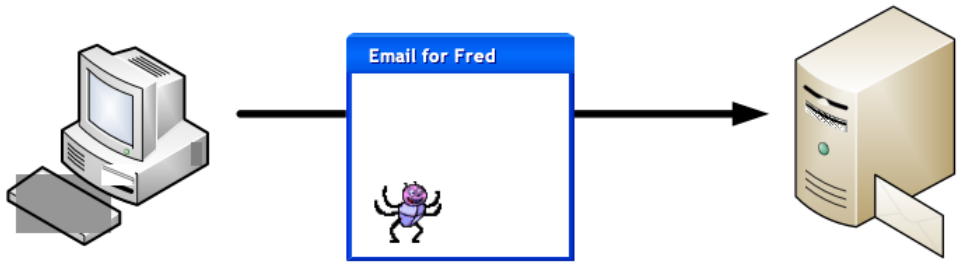

alert tcp $HOME_NET any -> $EXTE
(msg:"CHAT MSN login attempt";
flow:to_server,established;
content: "USR "; depth:4; nocase;
content:" TWN "; distance:1; nocase;
classtype:policy-violation; sid:1991
attempted-admin
attempted-user
policy-violation
shellcode-detect
successful-admin
successful-user
trojan-activity
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
icmp-event
misc-activity
network-scan
not-suspicious
protocol-command-decode
string-detect
A few intrusions
IDS
Author: Prof Bill Buchanan
Intrusions/Policy Violations

Rules


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;)
DEPTH. Modifier for previous content … defines to look within 4 bytes of the payload
NOCASE; Modifier for previous content … ignore the case of the content
DISTANCE. Modified for previous content and defines how far into the payload it should search (in bytes)
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
A few intrusions
IDS
Author: Prof Bill Buchanan
Intrusions/Policy Violations

Rules

```
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;)
```



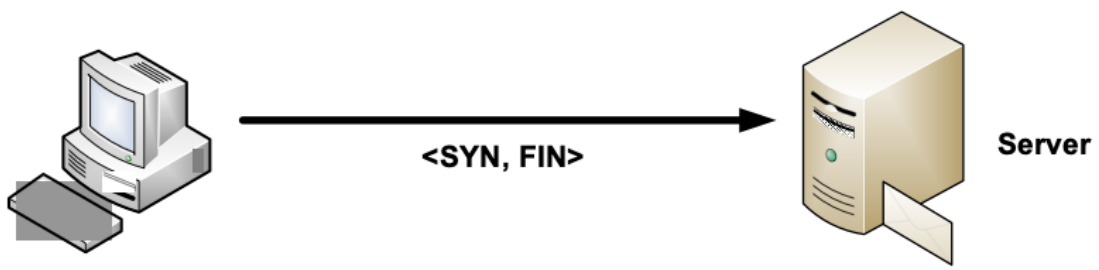
Email server (SMTP – listening on port 25)

```
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");
 }
}
```

Author: Prof Bill Buchanan

Rules

```
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;)
```




```
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");
  }
}
```

A few intrusions

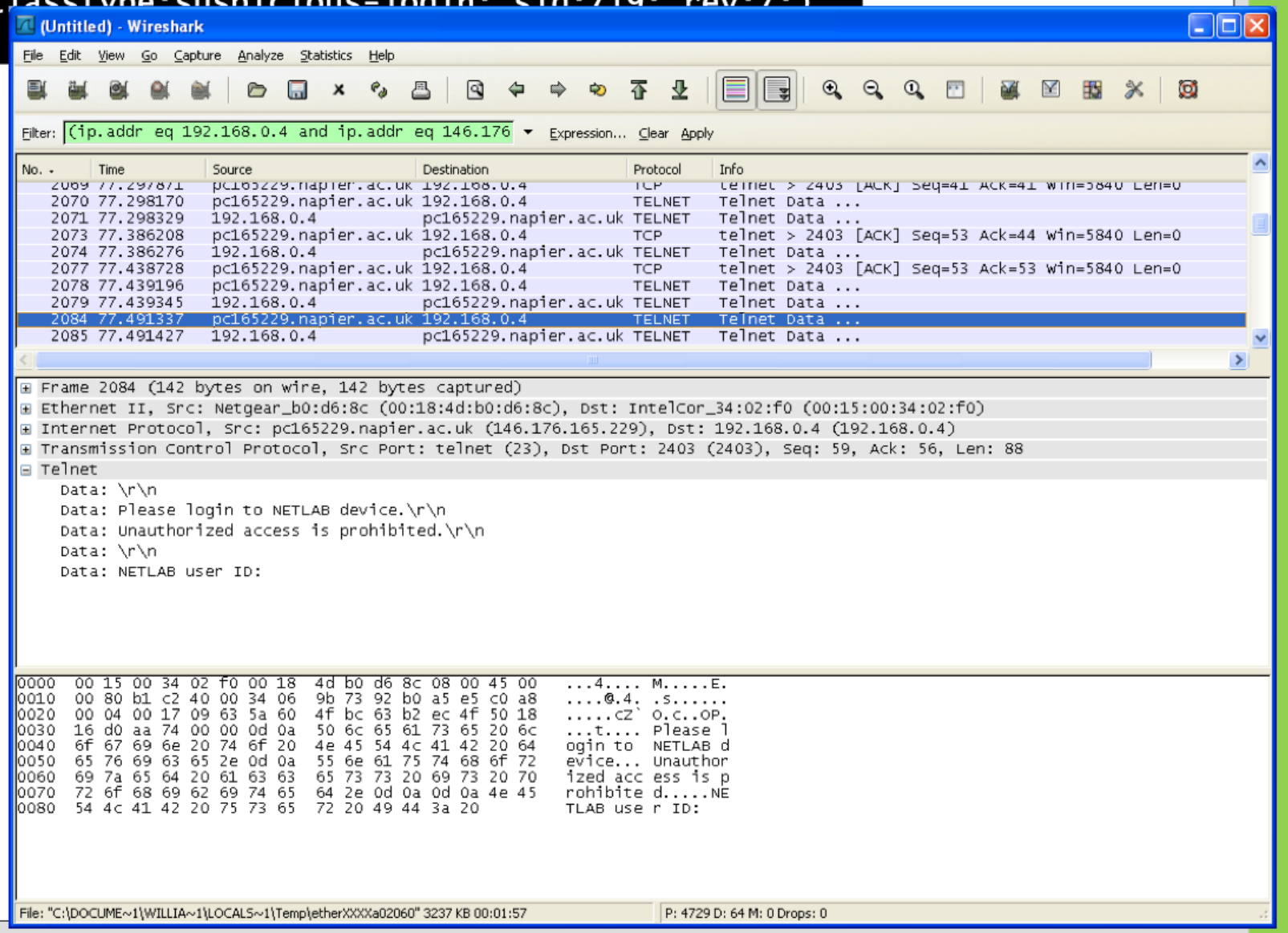
IDS

Author: Prof Bill Buchanan

cyber data

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;)
```



A few intrusions

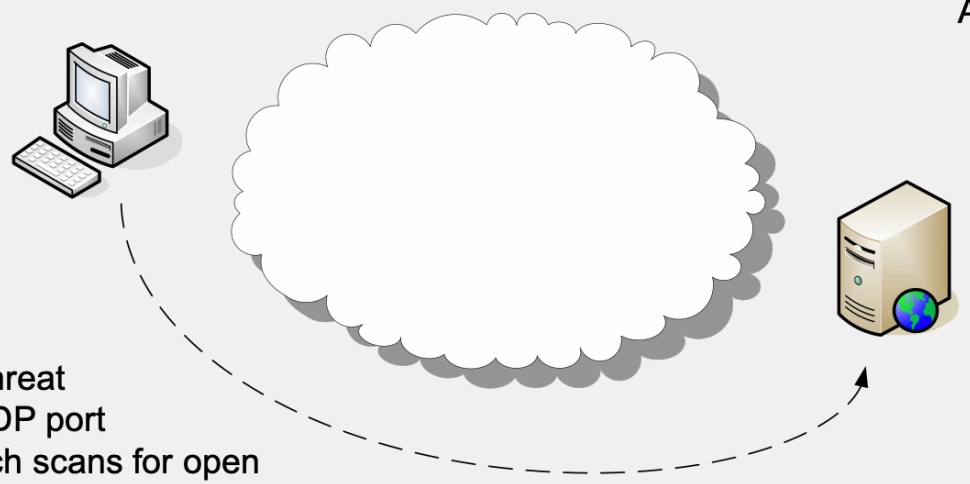
IDS

PCAP

Intrusions/Policy Violations

Port sweep

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



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

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.

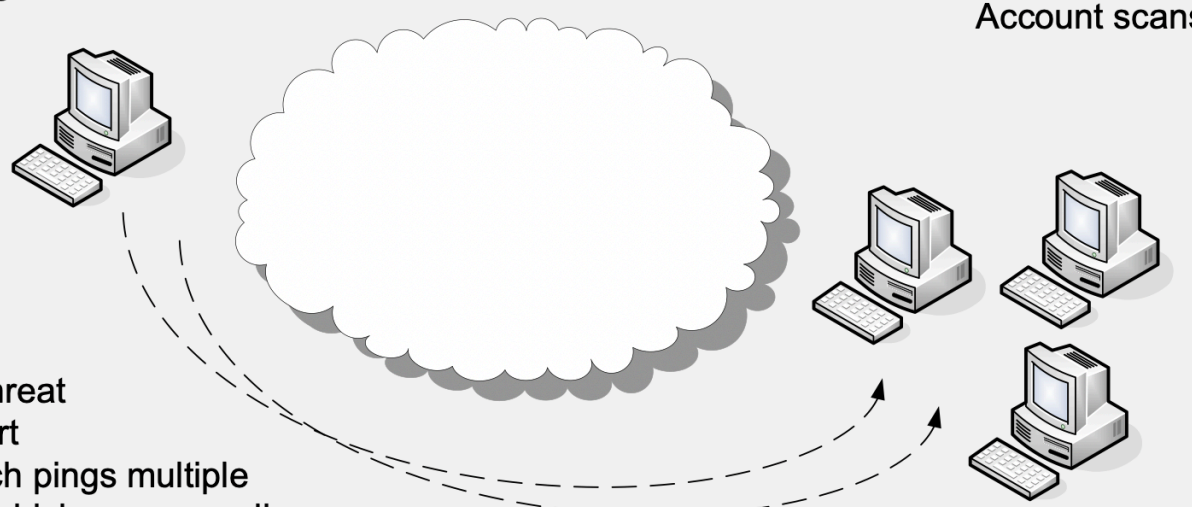
An open port is in the LISTEN state.

```
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
```

Ping sweep

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

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

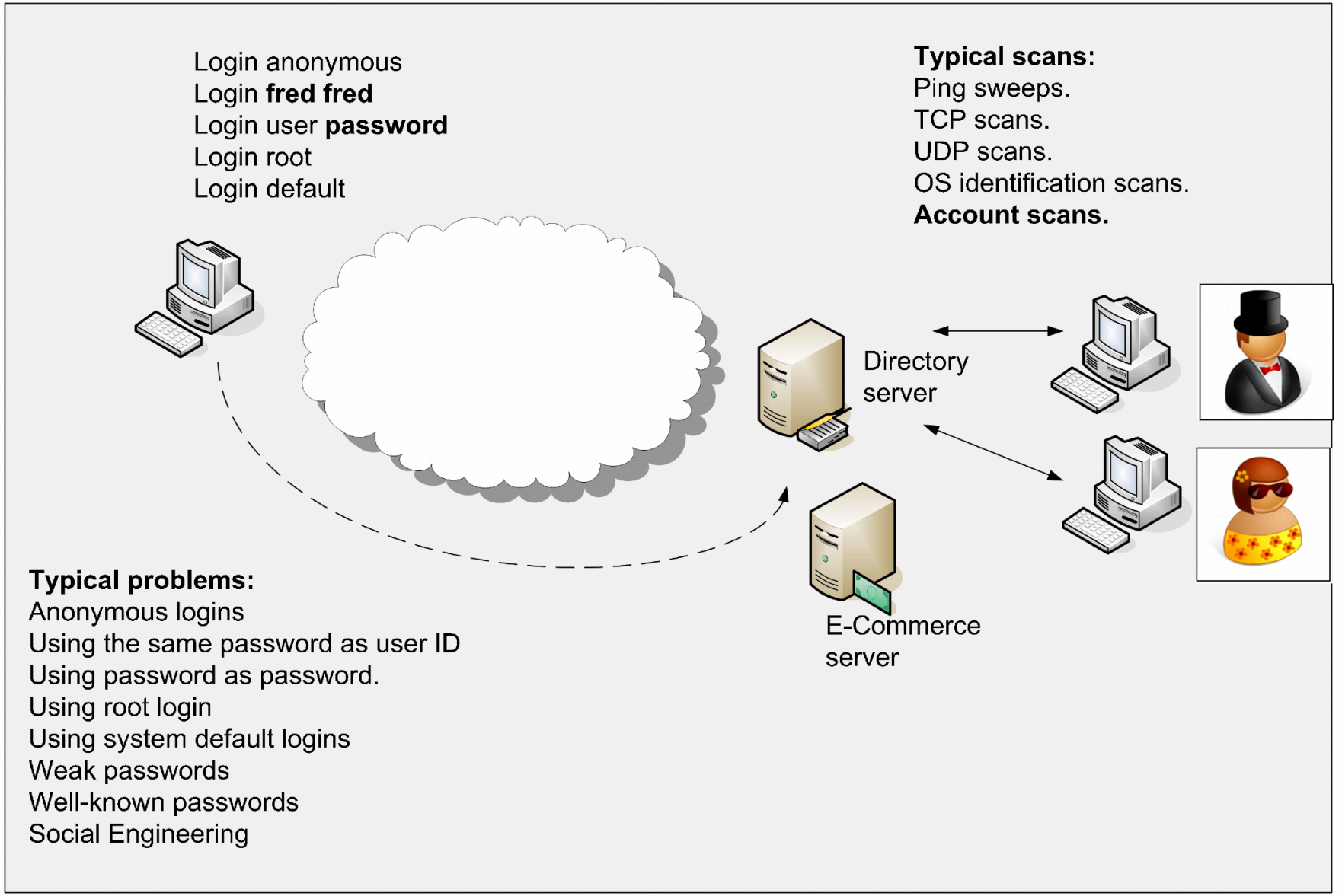


A particular threat is the ping port scanner, which pings multiple hosts to see which ones are alive

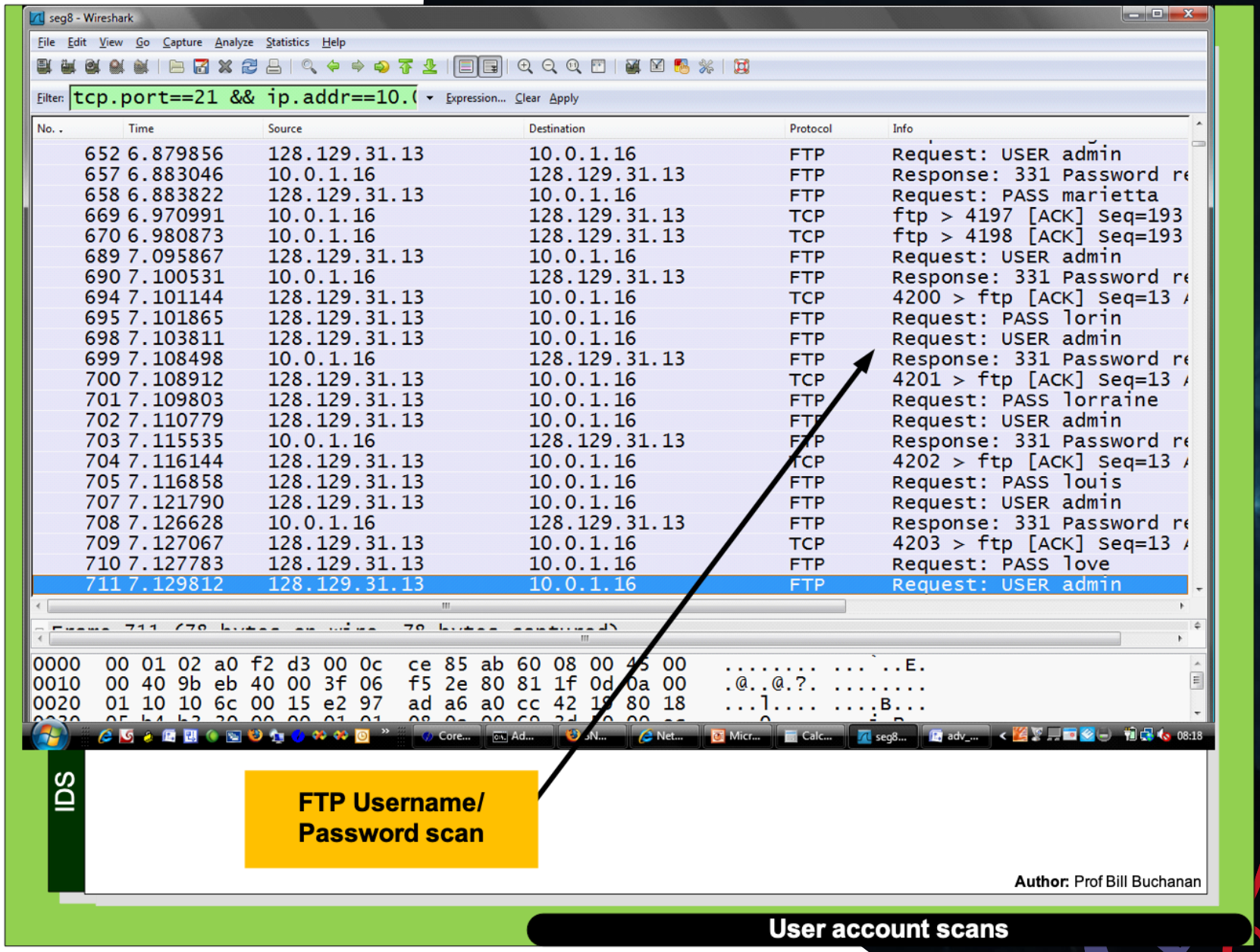
If an intruder finds one, they may try and connect to it.

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

Login sweep

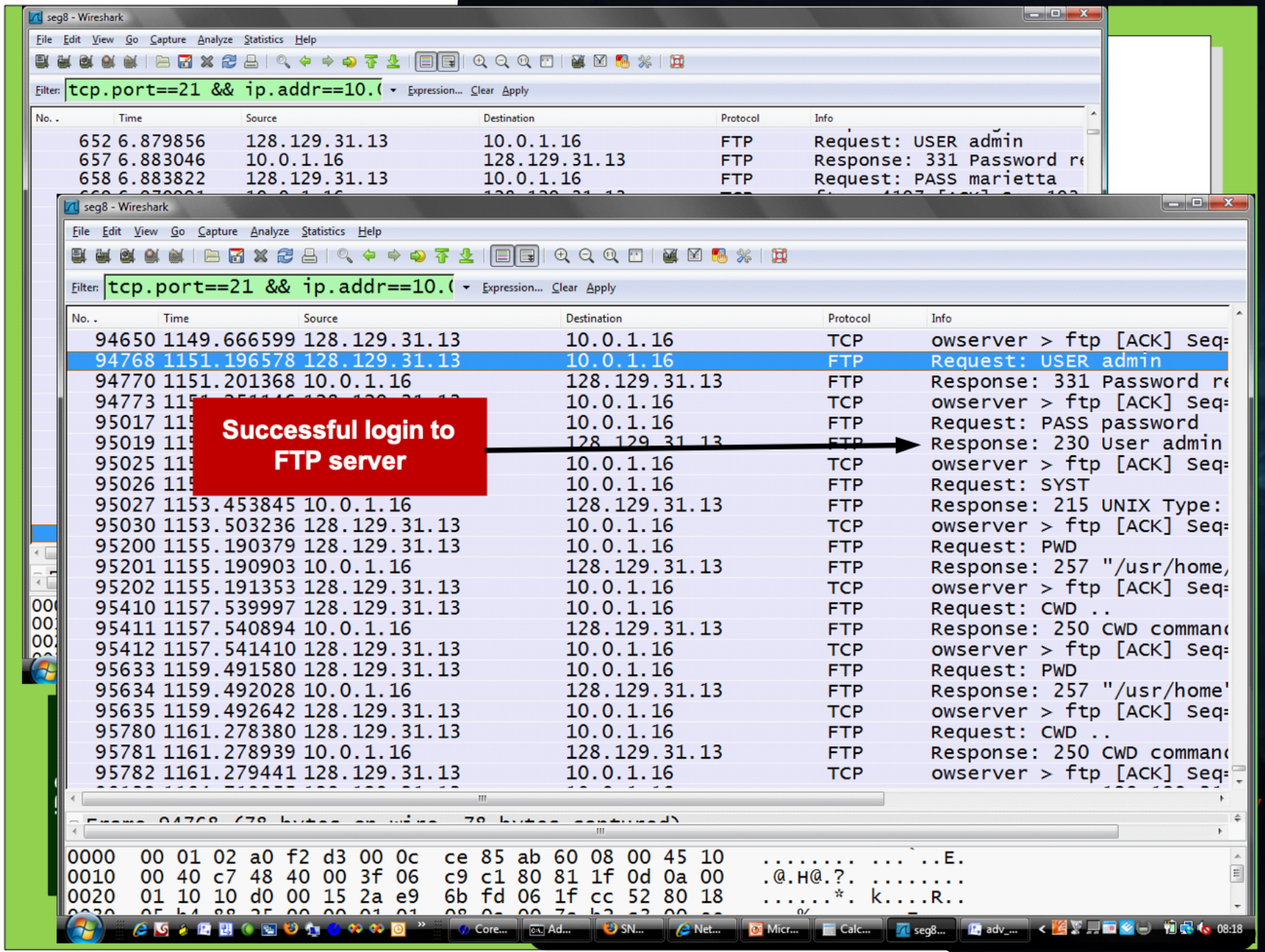

Login anonymous
Login **fred fred**
Login user **password**
Login root
Login default
Typical scans:
Ping sweeps.
TCP scans.
UDP scans.
OS identification scans.
Account scans.
Directory server
E-Commerce server
Typical problems:
Anonymous logins
Using the same password as user ID
Using password as password.
Using root login
Using system default logins
Weak passwords
Well-known passwords
Social Engineering

Login sweep



PCAP

Login sweep



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
```

Web

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
```

Web

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
```

Web

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
```

Web

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: 0xFEF9  TcpLen: 20
```

Web

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 (itype: 0; msg: "ICMP Echo Reply";sid:9000001;)
alert icmp any any -> any any (itype: 3; msg: "ICMP Destination Unreachable";sid:9000002;)
alert icmp any any -> any any (itype: 4; msg: "ICMP Source Quench Message received";sid:9000003;)
alert icmp any any -> any any (itype: 5; msg: "ICMP Redirect message";sid:9000004;)
alert icmp any any -> any any (itype: 8; msg: "ICMP Echo Request";sid:9000005;)
alert icmp any any -> any any (itype: 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
```

Web

cyber & data

"From bits to information"

Intrusion Detection Systems

Port scanning

SCAN.RULE

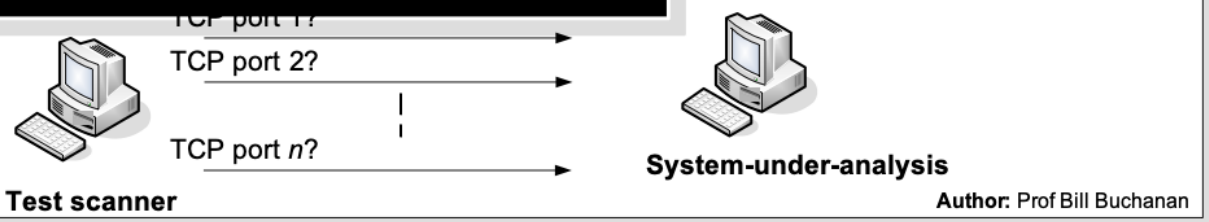
```
preprocessor flow: stats_interval 0 hash 2
preprocessor sfportscan: proto { all } scan_type { all }
                sense_level { low } logfile { portscan.log }
```

```
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 decoyportscan distributedportscan
    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 Ti
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
```

PORTSCAN.LOG

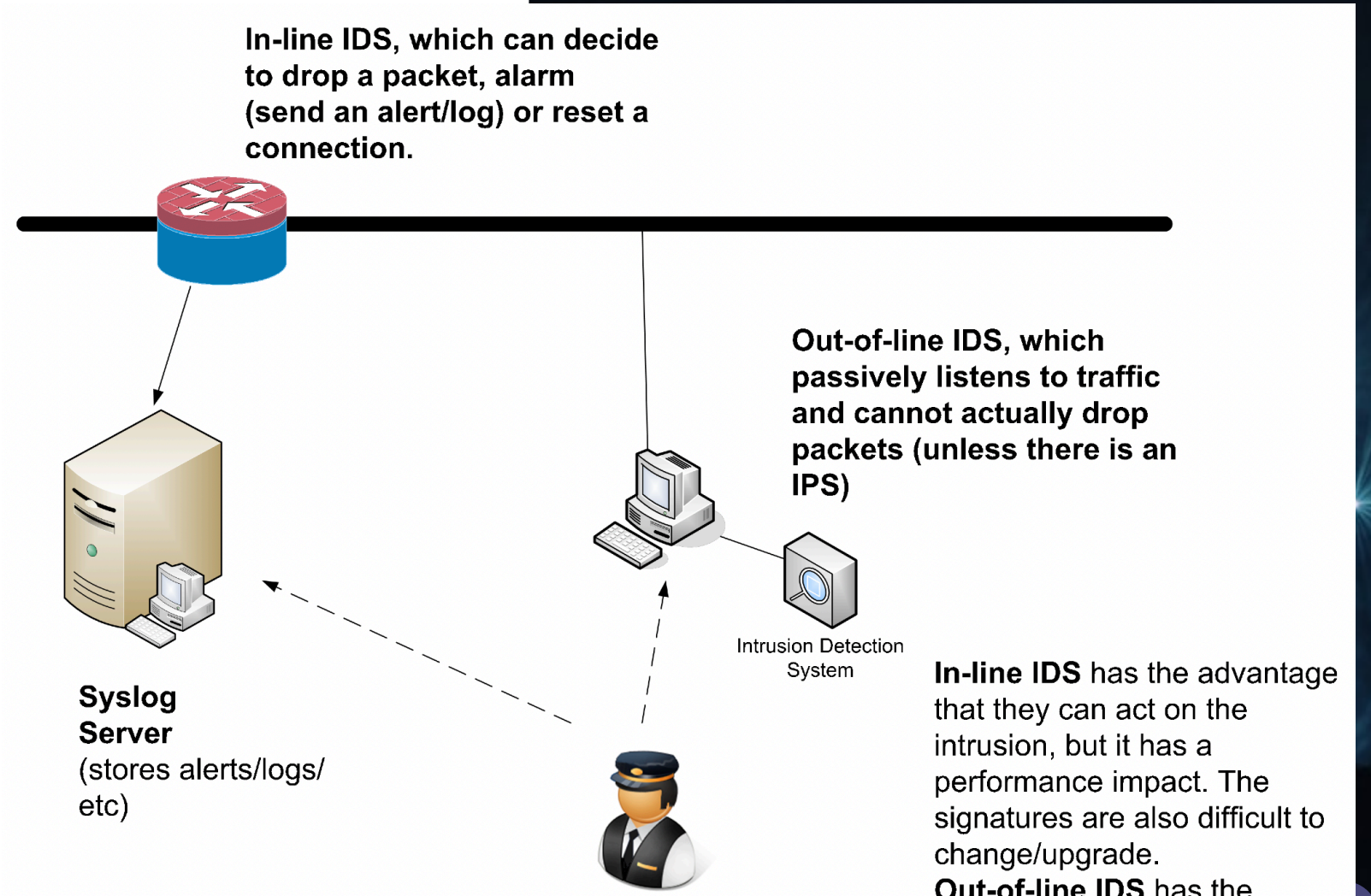
```
Time: 08/17-14:41:54.495296
event_ref: 0
192.168.0.3 -> 64.13.134.49 (portscan) TCP Portsweep
Priority Count: 5
Connection Count: 135
IP Count: 43
Scanned IP Range: 64.13.134.49:216.239.59.99
Port/Proto Count: 1
Port/Proto Range: 80:80

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
```

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.



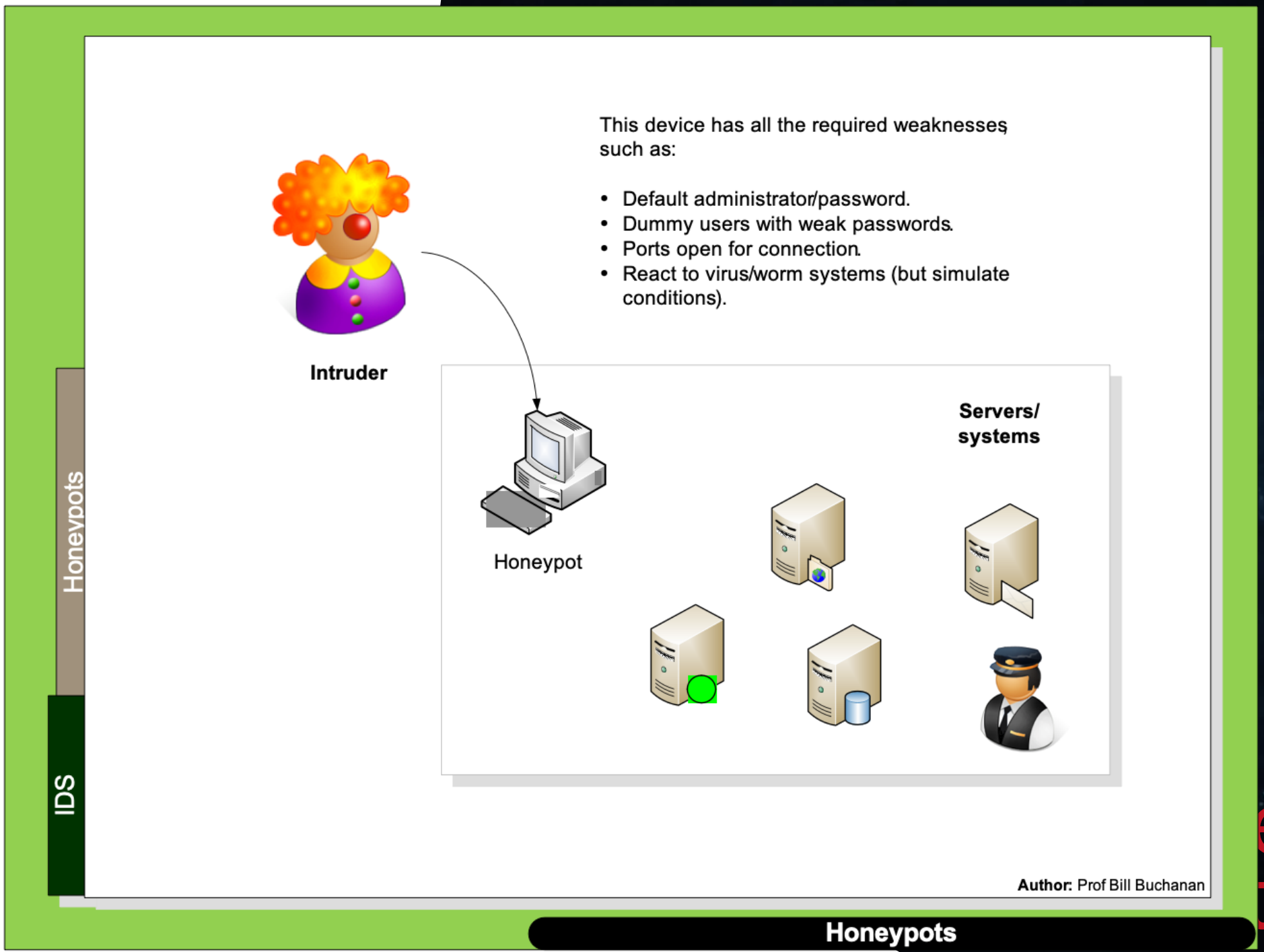
Port scanner pre-processor

In-line/out-of-line



In-line IDS has the advantage that they can act on the intrusion, but it has a performance impact. The signatures are also difficult to change/upgrade.
Out-of-line IDS has the advantage of being able to more easily craft an IDS rule, but cannot take actions, directly.

Honeypot

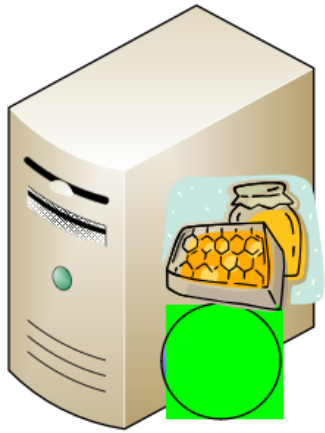

This device has all the required weaknesses, such as:
Default administrator/password.
Dummy users with weak passwords.
Ports open for connection.
React to virus/worm systems (but simulate conditions).
Intruder
Honeypot
Servers/ systems
Honeypots
IDS
Author: Prof Bill Buchanan
Honeypots

Honeypot

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

Honeypots

IDS



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

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

cyber & data

"From bits to information"

Intrusion Detection Systems