XIOBE

Incident Response Consulting

5

HACKB

Wireshark Workshop

$ whois

- Erik Vanderhasselt ()
- Xiobe does 3 things:
  - Risk Management
  - Incident Response (= risk mitigation strategy for some residual risks)
  - Social Engineering (my offensive side)

Wireshark

- Website: https://www.wireshark.org/
- Wiki: https://wiki.wireshark.org/
- Workshop time 1 hour, training 3 to 5 days
- Workshop means you work
- Traces during this workshop:
    - https://www.wireshark.org/assets/webinar/wireshark2webinar_traces.zip

Wireshark

Install wireshark/download if you didn’t while I give you this intro …

Every packet analysis starts with 4 basic questions:

- (Who?) What system is talking to what system?
- (How?) What are they using to communicate?
- (Direction?) Is it one way, both ways? Who initiated the connection?
- (When?) What was the start and end time of the communication? Is there periodicity?

Sometimes we will be able to look at the content, but there is no guarantee.

Wireshark - PCAP Naming

Not a convention but this is how Laura Chappell does it:

```
sec-<customer>-<caseID>-<classification>-<yyyy-mm-dd>-<clientside|serverside|tap>.pcapng
```

Wireshark - Exercise

```
sec-EHACKB-007-DoS-20171215-clientside.pcapng
```

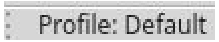
What would the name above mean?

Wireshark - Profiles

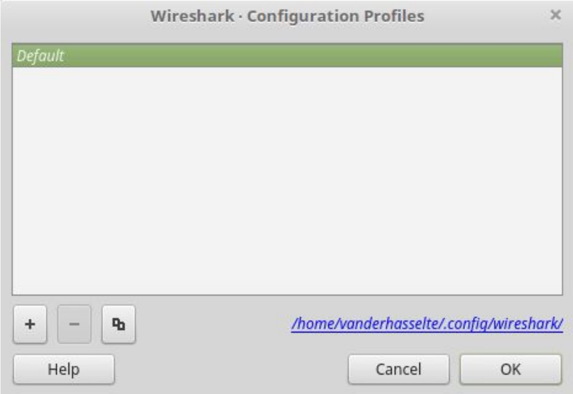
- Profiles contain preferences
- Preferences depend on person but also on the job you are doing.

Exercise:

- Create your own profiles:
  - Regular one for daily use
  - One for HTTP
  - Look at where the folders are on the file system



Profile: Default



Wireshark · Configuration Profiles

Default

/home/vanderhasselte/.config/wireshark/

Wireshark - Common Mistakes

- DNS already in DNS cache, thus you don’t see the DNS query
- DoS on Wireshark with big files
  - Split capture up with command line tools
  - Lower the number of columns on the screen
  - Turn off GEOIP
  - Lower the number of coloring rules

Wireshark - Client or Server

- TTL Value (255, 128, 64) can tell you if you are either client or server side.
- SYN with full TTL means it hasn’t been routed
- From SYN/ACK we can determine the hops

Exercise: Open http-winpcap.pcapng and determine if this is server or client side.

Wireshark - Capture File Properties

- Making sure you are investigating the right file
- Start and end times are part of the stats

Exercise: Open http-winpcap.pcapng and figure out start and end time of the capture.

Wireshark - Endpoints

- Making sure that you have the right endpoints in the file
- Endpoints are parts of the stats

Exercise: Open http-winpcap.pcapng, determine IP and MAC addresses

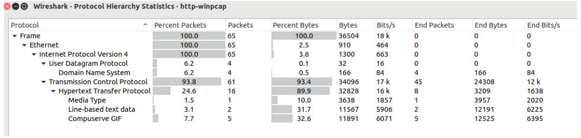
Wireshark - Resolved Addresses

- History of DNS resolution part of stats
- Only available if you ask the capturing software/device to do DNS resolution.

Exercise: Open http-winpcap.pcapng and determine names and IP addresses.

Wireshark - Protocols

- Tell you if you are looking at the right capture
- Stats tell you how many
- Shows issues in network easily



Wireshark · Protocol Hierarchy Statistics · http-winpcap

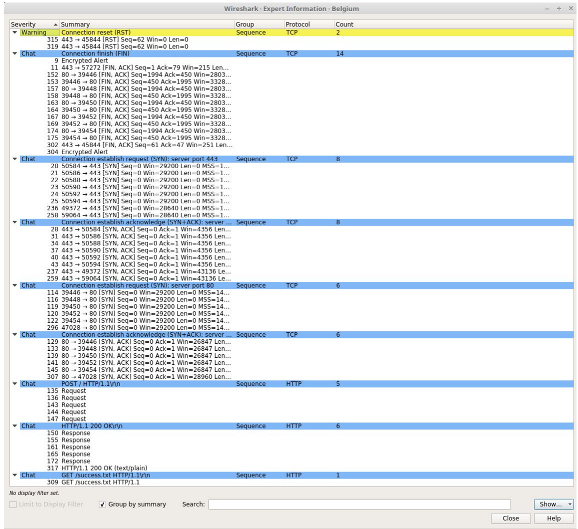
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
▾ Frame	100.0	65	100.0	36504	18 k	0	0	0
▾ Ethernet	100.0	65	2.5	910	464	0	0	0
▾ Internet Protocol Version 4	100.0	65	3.6	1300	663	0	0	0
▾ User Datagram Protocol	6.2	4	0.1	32	16	0	0	0
Domain Name System	6.2	4	0.5	166	84	4	166	84
▾ Transmission Control Protocol	93.8	61	93.4	34096	17 k	45	24308	12 k
▾ Hypertext Transfer Protocol	24.6	16	89.9	32828	16 k	8	3209	1638
Media Type	1.5	1	10.0	3638	1857	1	3957	2020
Line-based text data	3.1	2	31.7	11567	5906	2	12191	6225
Compuserve GIF	7.7	5	32.6	11891	6071	5	12525	6395

Exercise: Open sec-sickclient.pcapng and determine the protocols that were used.

Wireshark - Expert Info

- Yellow dot bottom left
- View per conversation

Exercise: Open http-download-good.pcapng and determine from the expert info what executable is being downloaded.



Wireshark · Expert Information · Belgium

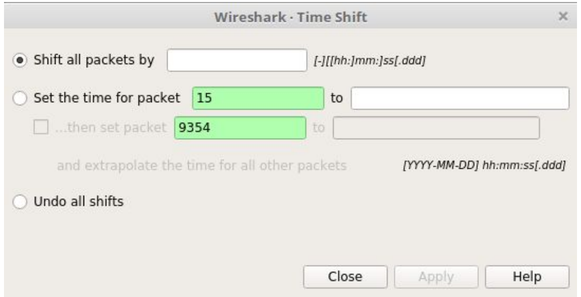
```
Severity        Summary                                              Group     Protocol  Count
▼ Warning       Connection reset (RST)                               Sequence  TCP       2
            315 443 → 45844 [RST] Seq=62 Win=0 Len=0
            319 443 → 45844 [RST] Seq=62 Win=0 Len=0
▼ Chat          Connection finish (FIN)                              Sequence  TCP       14
              9 Encrypted Alert
             11 443 → 57272 [FIN, ACK] Seq=1 Ack=79 Win=215 Len...
            152 80 → 39446 [FIN, ACK] Seq=1994 Ack=450 Win=2803...
            153 39446 → 80 [FIN, ACK] Seq=450 Ack=1995 Win=3328...
            157 80 → 39448 [FIN, ACK] Seq=1994 Ack=450 Win=2803...
            158 39448 → 80 [FIN, ACK] Seq=450 Ack=1995 Win=3328...
            163 80 → 39450 [FIN, ACK] Seq=1994 Ack=450 Win=2803...
            164 39450 → 80 [FIN, ACK] Seq=450 Ack=1995 Win=3328...
            167 80 → 39452 [FIN, ACK] Seq=1994 Ack=450 Win=2803...
            169 39452 → 80 [FIN, ACK] Seq=450 Ack=1995 Win=3328...
            174 80 → 39454 [FIN, ACK] Seq=1994 Ack=450 Win=2803...
            175 39454 → 80 [FIN, ACK] Seq=450 Ack=1995 Win=3328...
            302 443 → 45844 [FIN, ACK] Seq=61 Ack=47 Win=251 Len...
            304 Encrypted Alert
▼ Chat          Connection establish request (SYN): server port 443  Sequence  TCP       8
             20 50584 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1...
             21 50586 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1...
             22 50588 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1...
             23 50590 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1...
             24 50592 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1...
             25 50594 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1...
            236 49372 → 443 [SYN] Seq=0 Win=28640 Len=0 MSS=1...
            258 59064 → 443 [SYN] Seq=0 Win=28640 Len=0 MSS=1...
▼ Chat          Connection establish acknowledge (SYN+ACK): server ... Sequence TCP      8
             28 443 → 50584 [SYN, ACK] Seq=0 Ack=1 Win=4356 Len...
             31 443 → 50586 [SYN, ACK] Seq=0 Ack=1 Win=4356 Len...
             34 443 → 50588 [SYN, ACK] Seq=0 Ack=1 Win=4356 Len...
             37 443 → 50590 [SYN, ACK] Seq=0 Ack=1 Win=4356 Len...
             40 443 → 50592 [SYN, ACK] Seq=0 Ack=1 Win=4356 Len...
             43 443 → 50594 [SYN, ACK] Seq=0 Ack=1 Win=4356 Len...
            237 443 → 49372 [SYN, ACK] Seq=0 Ack=1 Win=43136 Le...
            259 443 → 59064 [SYN, ACK] Seq=0 Ack=1 Win=43136 Le...
▼ Chat          Connection establish request (SYN): server port 80   Sequence  TCP       6
            114 39446 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=14...
            116 39448 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=14...
            119 39450 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=14...
            120 39452 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=14...
            122 39454 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=14...
            296 47028 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=14...
▼ Chat          Connection establish acknowledge (SYN+ACK): server ... Sequence TCP      6
            129 80 → 39446 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len...
            133 80 → 39448 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len...
            139 80 → 39450 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len...
            141 80 → 39452 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len...
            145 80 → 39454 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len...
            307 80 → 47028 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len...
▼ Chat          POST / HTTP/1.1\r\n                                  Sequence  HTTP      5
            135 Request
            136 Request
            143 Request
            144 Request
            147 Request
▼ Chat          HTTP/1.1 200 OK\r\n                                  Sequence  HTTP      6
            150 Response
            155 Response
            161 Response
            165 Response
            172 Response
            317 HTTP/1.1 200 OK (text/plain)
▼ Chat          GET /success.txt HTTP/1.1\r\n                        Sequence  HTTP      1
            309 GET /success.txt HTTP/1.1
```

No display filter set.

☐ Limit to Display Filter ☑ Group by summary Search:

Wireshark - Time Shifting

- Right click on packet
- Gotcha : Only click apply once
- Useful when working with different timezones / winter and summer time.



Wireshark · Time Shift

● Shift all packets by [-][[hh:]mm:]ss[.ddd]

○ Set the time for packet 15 to

□ ...then set packet 9354 to

and extrapolate the time for all other packets [YYYY-MM-DD] hh:mm:ss[.ddd]

○ Undo all shifts

Exercise: Copy http-download-good.pcapng and open the copy in wireshark. Time shift the packets 7 hours in the past.

Wireshark - Marking Packets

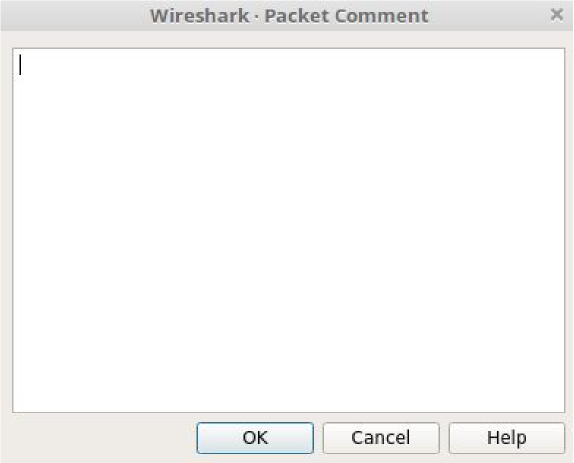
- Packets of interest
- Default color scheme is black, packets with errors are marked with black too.
- Color can be set in the preferences
- You can mark/unmark

Exercise: Open http-good-download.pcapng and mark the HTTP GET. Make marking color orange.

Wireshark - Commenting Packets

- Handy when doing analysis
- Right click on packet

Exercise: Comment on the previously marked packet. The comment is “The root of all evil”.



Wireshark - Protocol Help

- We don’t know all protocols
- We have sometimes uncertainty
- Wireshark wiki

Exercise: Open from http-good-download.pcapng the wiki page for HTTP.

Wireshark - MAC Addresses

- 6 octets AA:BB:CC:DD:EE:FF
- 3 octets AA:BB:CC are called OUI
- OUI assigned by IEEE to company
- ! Depending on where you capture it might not be the endpoint’s MAC

Exercise: Open http-good-download.pcapng and determine the brand of the NICs

Wireshark - Display Filters

- Same filter, different ways
  - Http (layer 7)
  - Tcp.port == 80 (layer 3)
- Green is valid, red is not
- Execute with little arrow on the right
- Allows CIDR notation, not 0-255 notation

Exercise: Make a filter for cookies, filter on live data and surf to your favorite website.

Wireshark - Display Filter Construction

- Expression builder
- Protocol.field
- Operators
  - >, >=
  - <, <=
  - == , !=
  - frame contains “<string>”
  - frame contains “<regular expression>”
  - And, &&
  - Or, ||
- Buttons

Wireshark - Display Filter Construction

1. Exercise: Open http-download-good.pcapng and create filter for host in the http-protocol. Run your filter and determine the host.
2. Exercise: Open http-download-good.pcapng and prepare a filter so that you obtain (tcp.seq == 2921) && (tcp.ack == 444) and thus not typing it directly into the display filter. What is special about this packet?

Wireshark - Default Columns

Default columns:

- No, the number of the packet
- Time, the time since the capture has been started
- Source, the IP of the source
- Destination, the IP of the destination
- Protocol, protocol of the packet
- Length, length of the packet
- Info, more information on the packet

Wireshark - Creating Columns

- Preferences
- Selection of field in packet

Exercise: Create a new column called timestamp that will show you the actual timestamp of each packet

Wireshark - Creating Columns

1. Exercise: Modify the default time column so that it shows the time since the previous packet expressed in milliseconds.
2. Exercise: Open http-download-good.pcapng and select a packet. Create from the fields the column source port and destination port. Put the source port column next to the source IP and the destination port next to the destination IP.
3. Exercise: Open http-download-good.pcap and create yourself a column from the field that holds the TTL.

Wireshark - Delta Time (Protocol Pref.)

- Delta Time same as TCP Delay
- Is a calculated value, need to configure your TCP protocol
  - Right click TCP packet
  - Protocol preferences for TCP
  - Calculate conversation timestamps
- TCP frame 2 new calculated fields
  - Time since first frame in this TCP stream
  - Time since previous frame in this TCP stream (= Delta Time)
- Useful for discussions when you are the cause of latency because you are doing something with your security tools.

Wireshark - Delta Time

- TCP delay > 1 second means issue (useful to see impact of DoS)
- Ignore:
    - HTTP GET requests (you are waiting on user)
    - FIN and RST flags (connection teardown, has no end user impact)

Wireshark - Delta Time

Exercise: Open http-download-bad.pcapng and create a column for the delta time

Wireshark - From baseline to bad

- There is no magic formula
- Capture the network on a regular basis and compare (diff)
- Select the traffic that is good, right click and choose follow stream
  - Select filter out this stream
- You can create buttons to remove the baseline traffic

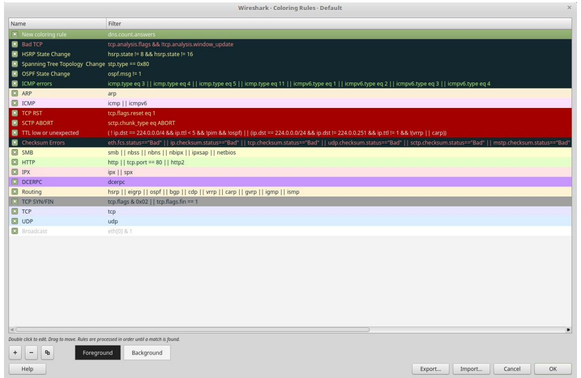
Wireshark - Finding Packets

- Search (Ctrl+F)
- UTF-8/ASCII or UTF-16
- Hex and regex also accepted

Exercise: Capture surf session to https://www.ehackb.be and search for Wireshark

Wireshark - Coloring Rules

- Handy to identify issues/bad stuff
- `View > Coloring Rules`
- Is a file (help > about > folders)
- `[Coloring Rule Name]`
- `[Coloring Rule String]`



Wireshark · Coloring Rules · Default

Name	Filter
☒ New coloring rule	dns.count.answers
☒ Bad TCP	tcp.analysis.flags && !tcp.analysis.window_update
☒ HSRP State Change	hsrp.state != 8 && hsrp.state != 16
☒ Spanning Tree Topology Change	stp.type == 0x80
☒ OSPF State Change	ospf.msg != 1
☒ ICMP errors	icmp.type eq 3 || icmp.type eq 4 || icmp.type eq 5 || icmp.type eq 11 || icmpv6.type eq 1 || icmpv6.type eq 2 || icmpv6.type eq 3 || icmpv6.type eq 4
☒ ARP	arp
☒ ICMP	icmp || icmpv6
☒ TCP RST	tcp.flags.reset eq 1
☒ SCTP ABORT	sctp.chunk_type eq ABORT
☒ TTL low or unexpected	(! ip.dst == 224.0.0.0/4 && ip.ttl < 5 && !pim && !ospf) || (ip.dst == 224.0.0.0/24 && ip.dst != 224.0.0.251 && ip.ttl != 1 && !(vrrp || carp))
☒ Checksum Errors	eth.fcs.status=="Bad" || ip.checksum.status=="Bad" || tcp.checksum.status=="Bad" || udp.checksum.status=="Bad" || sctp.checksum.status=="Bad" || mstp.checksum.status=="Bad"
☒ SMB	smb || nbss || nbns || nbipx || ipxsap || netbios
☒ HTTP	http || tcp.port == 80 || http2
☒ IPX	ipx || spx
☒ DCERPC	dcerpc
☒ Routing	hsrp || eigrp || ospf || bgp || cdp || vrrp || carp || gvrp || igmp || ismp
☒ TCP SYN/FIN	tcp.flags & 0x02 || tcp.flags.fin == 1
☒ TCP	tcp
☒ UDP	udp
☒ Broadcast	eth[0] & 1

Double click to edit. Drag to move. Rules are processed in order until a match is found.

Exercise: Open sec-sickclient.pcapng go to packet 10.
DNS contains different IP addresses from C2 servers. Apply a coloring rule where DNS contains more than 5 IP addresses. Set the background color to yellow and font to black.

Wireshark - Filter Buttons

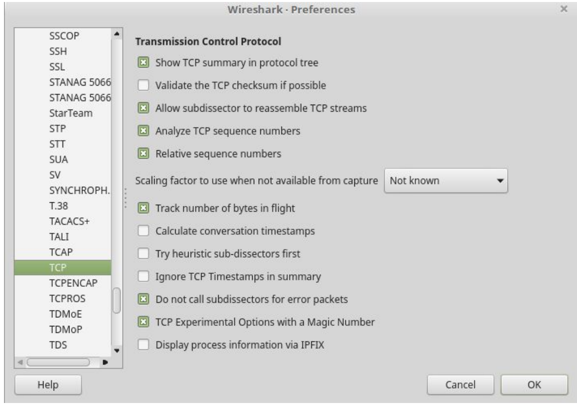
- Repetitive display filters
- Removal via `preferences > filter expressions`

Exercise: Create a button to show you HTTP and HTTPS traffic

Wireshark - Exporting Artifacts

- Protocols:
  - DICOM (Medicine)
  - HTTP
  - SMB
  - TFT
- Allow subdissector to reassemble streams

  `Edit > Preferences > Protocols > TCP`



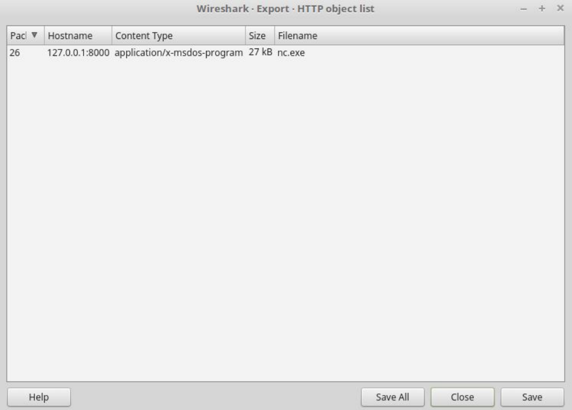
Transmission Control Protocol

☒ Show TCP summary in protocol tree
☐ Validate the TCP checksum if possible
☒ Allow subdissector to reassemble TCP streams
☒ Analyze TCP sequence numbers
☒ Relative sequence numbers
Scaling factor to use when not available from capture: Not known
☒ Track number of bytes in flight
☐ Calculate conversation timestamps
☐ Try heuristic sub-dissectors first
☐ Ignore TCP Timestamps in summary
☒ Do not call subdissectors for error packets
☒ TCP Experimental Options with a Magic Number
☐ Display process information via IPFIX

Wireshark - Exporting Artifacts

- To export `File > Export Objects`
- Calculate hash
- Sign hash with your GPG key

Exercise: open download_netcat.pcapng and export the nc.exe file



Pac	Hostname	Content Type	Size	Filename
26	127.0.0.1:8000	application/x-msdos-program	27 kB	nc.exe

Wireshark - Find Executables

Exercise: Search download_netcat.pcapng for the PE DOS header “MZ” and “This program cannot be run in DOS mode”.

Wireshark - Exporting Packets

- You only want to keep the interesting packets (marked packets)
- `File > Export Specified Packets`

Exercise: Open download_netcat.pcapng, mark the HTTP packets. Export the packets. Open that the export pcap file and try to export nc.exe.

Wireshark - Exporting Dissections

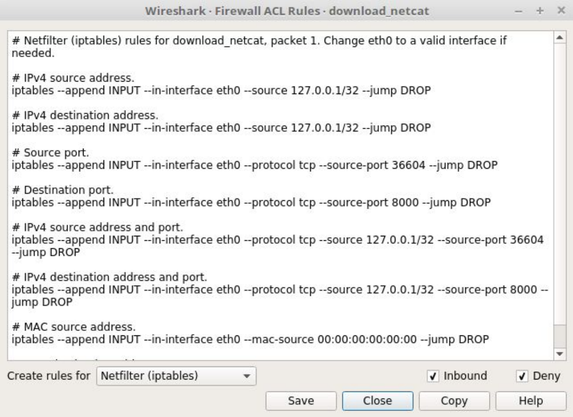
- Export PCAP to another format
  - Plaintext
  - CSV
  - JSON
  - …

Exercise: Open download_netcat.pcapng and export it to JSON.

Wireshark - Firewall Rules

- You can generate firewall rules from wireshark in the following formats
  - CISCO IOS (std and extended)
  - IPfilter
  - IPFirewall
  - Netfilter (iptables)
  - Packet Filter (pf)
  - Windows Firewall (netsh)

Exercise: Generate from sec-sickclient for Windows. Implement them on your windows VM.



```
# Netfilter (iptables) rules for download_netcat, packet 1. Change eth0 to a valid interface if
needed.

# IPv4 source address.
iptables --append INPUT --in-interface eth0 --source 127.0.0.1/32 --jump DROP

# IPv4 destination address.
iptables --append INPUT --in-interface eth0 --source 127.0.0.1/32 --jump DROP

# Source port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source-port 36604 --jump DROP

# Destination port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source-port 8000 --jump DROP

# IPv4 source address and port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source 127.0.0.1/32 --source-port 36604
--jump DROP

# IPv4 destination address and port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source 127.0.0.1/32 --source-port 8000 --
jump DROP

# MAC source address.
iptables --append INPUT --in-interface eth0 --mac-source 00:00:00:00:00:00 --jump DROP
```

Create rules for: Netfilter (iptables)

☑ Inbound ☑ Deny

Wireshark - Anomalies

- Window size 0
- SYN packet with no segment size
- TTL that is systematically the same
- Packet payload like 'AAAAAAAA....'
- Well known destination ports as source ports (<1024)
- Large packets for that protocol (DNS, ICMP, ...)
- Bot communication handshakes
- Packet length that is systematically the same
- DNS Query for ANY (DNS reflection DoS)
- Unusual TCP flag combinations like SYN and FIN, SYN and RST, ...
- ...

Wireshark - Anomalies

- `Tcp.analysis.flags` can help
  - TCP retransmissions
  - TCP out-of-order
  - Duplicate ACK
  - Zero Windows
  - ...

Exercise: run tcp.analysis.flags against sec-sickclient.pcapng by first creating a button.

Contact

online : erik[.]vanderhasselt[@]xiobe.net
here: sharing beers and ideas will get you closer to your answer
conferences: come and say hello