Mason Competitive Cyber

Data Exfiltration: What's yours is mine

Upcoming Competitions

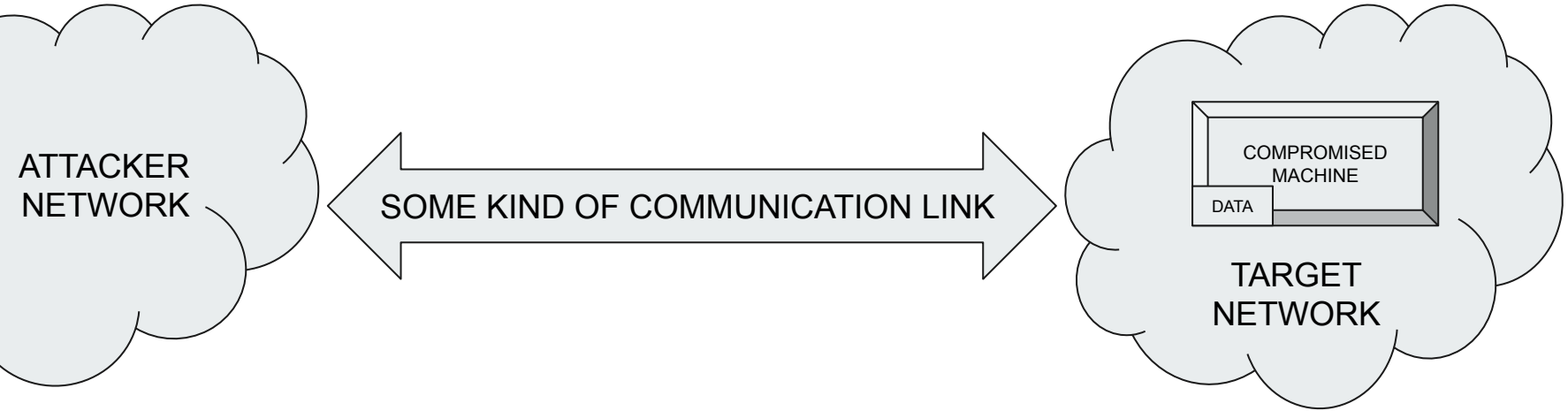
- National Cyber League Spring
  - Occurs throughout Spring semester
  - Registration happening soon! If you're interested, try the qualifier challenges
- CyberFusion State Cup (Team already full)
  - February 22-23
- VT Summit
  - March 28
- PatriotCTF 2020 :
  - April 11
  - Want to be a challenge writer? Talk to exec!
- UMDCTF
  - April 18

News

- Iowa caucuses: app whoopsie
- Citrix N-Day attack still affecting people because they’re not patching (CVE-2019-19781)
- Minebridge backdoor targeting US Finance Sector
  - VBA Stomping to bypass AV
  - Basically, a fancy macro attack
- Developing Story: Surprise JC Remodel

What is Exfiltration?

- Put simply:



Assumptions

- Attacker has sufficient control over the target machine to communicate on the network
- Attacker has some network path back to a Command and Control server
- Attacker has found some data that they want to get out of the network

What’s there to steal?

- Usernames/Passwords
- Industrial Secrets
- Cryptographic Material
- Credit Card Numbers
- CEO’s documents

The list goes on

How do we steal it?

- Insider threat: Just put it on a flash drive and walk out the door
- Remote threat (unencrypted/unobfuscated):
  - HTTP/S File Retrieval
  - FTP
  - Email
  - IM
- Remote threat (encrypted/obfuscated):
  - SSH/SFTP
  - Protocol tunneling
  - Steganography

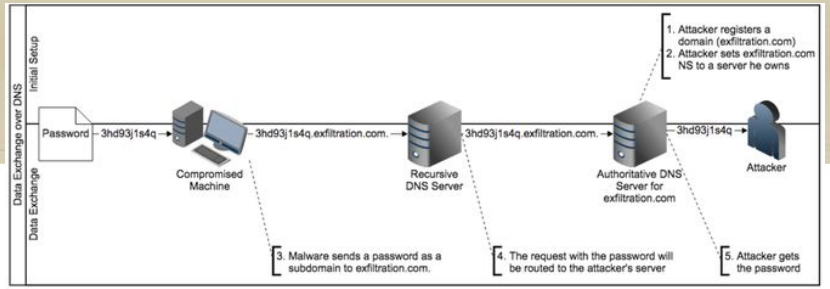
Of course, there's other techniques that are much more interesting and fun

Stealy Wheely Datamobile-y

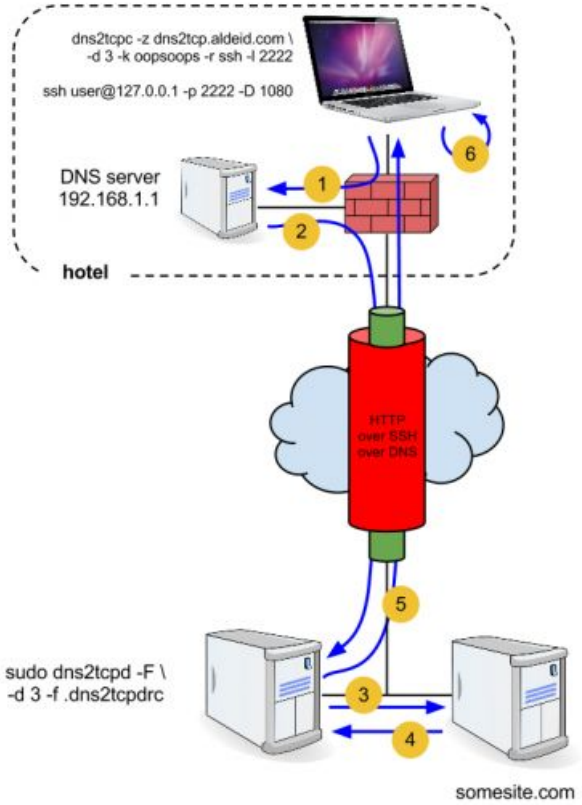
Other fun techniques:

- Public service exfiltration
  - Youtube comments, GitHub gists, PasteBin pastes, StackOverflow questions etc
- Layer 2 exfiltration
  - Cram stuff into wifi/ethernet headers
  - Obviously an attacker has to be in close proximity (same subnet for wired attacks, close enough to receive wireless data for wifi attacks)

DNS



- Put your data in as a subdomain (usually just called DNS exfiltration)
- Cram your whole connection over a DNS tunnel
  - https://www.aldeid.com/wiki/Dns2tcp
- Pros: DNS is rarely restricted through outbound firewalls
  - Easy to break things if you're restricting it
- Cons: Efficiency
  - Like breathing through a straw, but with the internet
  - DNS limits messages to 255 bytes with a fairly restrictive character set
  - But if you thought that was bad...

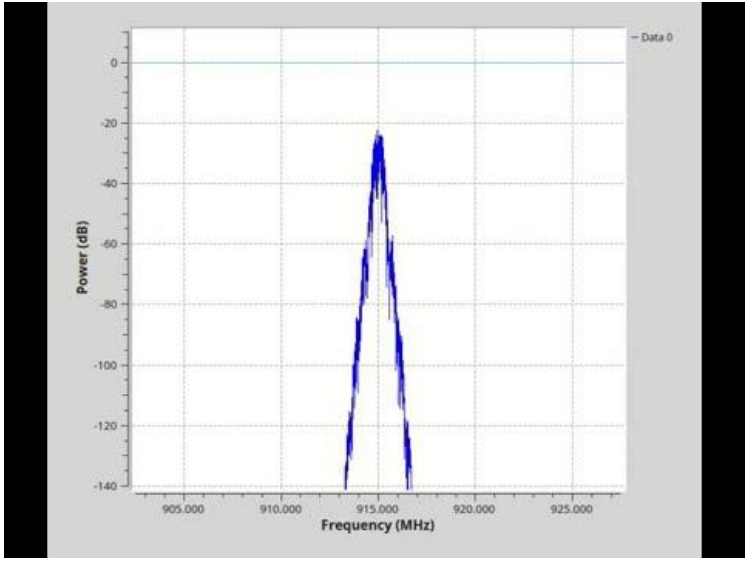


EXTREME protocol abuse

- Why just abuse one protocol? (DNS)
- Why not do a bunch?
- At the same time?
- And also make it so that even if engineers are looking for exfiltrated data, it’d be damn hard to find?

Example tool: Protocol Hopping

- Basis: Frequency hopping in RF
- Used in Military Radios (and WiFi [and fancy Civilian Radios])
- Highly resistant to jamming



Protocol Hopping

1. Choose a random UDP protocol from a list of available protocols
2. Embed data into a packet of that protocol
3. Send packet to c2
4. C2 has several different UDP servers running on it
5. Depending on which server the packet hits, we know which bytes are data bytes
6. Pull out the data and append it to the running total of received data
7. Repeat until out of data to exfiltrate
8. When no more packets are being received, that means that exfil is completed and we can read the received data on the c2 server
9. Optionally, sort packets and read to file

Playing with ID fields

- A bunch of UDP protocols have these
- Between 2 and 4 bytes
- Not always clear where these values come from
  - In fact, in the case of DNS sometimes these are deliberately difficult to predict

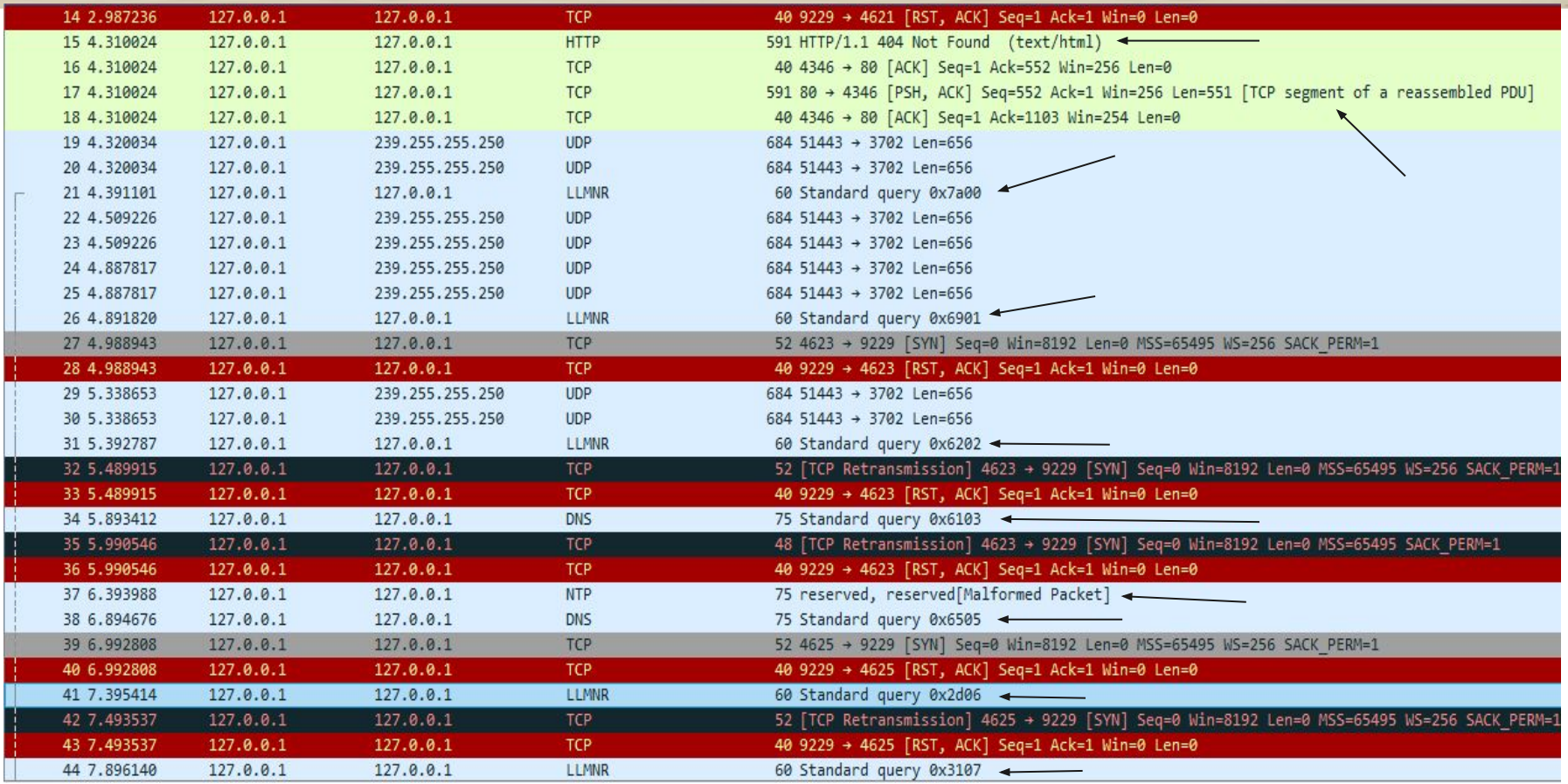
- Is anybody really going to miss that one field?

UDP Unreliability

D	C	X	X	DNS
D	C	X	X	LLMNR
D	D	D	C	NTP
D	D	D	C	BOOTP

- The data won't arrive *that* out of order
- They'll probably be within 256 places of where they're supposed to be
- Sort by queue number then by counter value

Result?



14	2.987236	127.0.0.1	127.0.0.1	TCP	40 9229 → 4621 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15	4.310024	127.0.0.1	127.0.0.1	HTTP	591 HTTP/1.1 404 Not Found (text/html)
16	4.310024	127.0.0.1	127.0.0.1	TCP	40 4346 → 80 [ACK] Seq=1 Ack=552 Win=256 Len=0
17	4.310024	127.0.0.1	127.0.0.1	TCP	591 80 → 4346 [PSH, ACK] Seq=552 Ack=1 Win=256 Len=551 [TCP segment of a reassembled PDU]
18	4.310024	127.0.0.1	127.0.0.1	TCP	40 4346 → 80 [ACK] Seq=1 Ack=1103 Win=254 Len=0
19	4.320034	127.0.0.1	239.255.255.250	UDP	684 51443 → 3702 Len=656
20	4.320034	127.0.0.1	239.255.255.250	UDP	684 51443 → 3702 Len=656
21	4.391101	127.0.0.1	127.0.0.1	LLMNR	60 Standard query 0x7a00
22	4.509226	127.0.0.1	239.255.255.250	UDP	684 51443 → 3702 Len=656
23	4.509226	127.0.0.1	239.255.255.250	UDP	684 51443 → 3702 Len=656
24	4.887817	127.0.0.1	239.255.255.250	UDP	684 51443 → 3702 Len=656
25	4.887817	127.0.0.1	239.255.255.250	UDP	684 51443 → 3702 Len=656
26	4.891820	127.0.0.1	127.0.0.1	LLMNR	60 Standard query 0x6901
27	4.988943	127.0.0.1	127.0.0.1	TCP	52 4623 → 9229 [SYN] Seq=0 Win=8192 Len=0 MSS=65495 WS=256 SACK_PERM=1
28	4.988943	127.0.0.1	127.0.0.1	TCP	40 9229 → 4623 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
29	5.338653	127.0.0.1	239.255.255.250	UDP	684 51443 → 3702 Len=656
30	5.338653	127.0.0.1	239.255.255.250	UDP	684 51443 → 3702 Len=656
31	5.392787	127.0.0.1	127.0.0.1	LLMNR	60 Standard query 0x6202
32	5.489915	127.0.0.1	127.0.0.1	TCP	52 [TCP Retransmission] 4623 → 9229 [SYN] Seq=0 Win=8192 Len=0 MSS=65495 WS=256 SACK_PERM=1
33	5.489915	127.0.0.1	127.0.0.1	TCP	40 9229 → 4623 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
34	5.893412	127.0.0.1	127.0.0.1	DNS	75 Standard query 0x6103
35	5.990546	127.0.0.1	127.0.0.1	TCP	48 [TCP Retransmission] 4623 → 9229 [SYN] Seq=0 Win=8192 Len=0 MSS=65495 SACK_PERM=1
36	5.990546	127.0.0.1	127.0.0.1	TCP	40 9229 → 4623 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
37	6.393988	127.0.0.1	127.0.0.1	NTP	75 reserved, reserved[Malformed Packet]
38	6.894676	127.0.0.1	127.0.0.1	DNS	75 Standard query 0x6505
39	6.992808	127.0.0.1	127.0.0.1	TCP	52 4625 → 9229 [SYN] Seq=0 Win=8192 Len=0 MSS=65495 WS=256 SACK_PERM=1
40	6.992808	127.0.0.1	127.0.0.1	TCP	40 9229 → 4625 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
41	7.395414	127.0.0.1	127.0.0.1	LLMNR	60 Standard query 0x2d06
42	7.493537	127.0.0.1	127.0.0.1	TCP	52 [TCP Retransmission] 4625 → 9229 [SYN] Seq=0 Win=8192 Len=0 MSS=65495 WS=256 SACK_PERM=1
43	7.493537	127.0.0.1	127.0.0.1	TCP	40 9229 → 4625 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
44	7.896140	127.0.0.1	127.0.0.1	LLMNR	60 Standard query 0x3107

whoami

46	2.065723	127.0.0.1	127.0.0.1	HTTP	488 GET / HTTP/1.1
126	7.164778	127.0.0.1	127.0.0.1	HTTP	488 GET / HTTP/1.1
132	8.524352	127.0.0.1	127.0.0.1	HTTP	488 GET / HTTP/1.1
179	12.557280	127.0.0.1	127.0.0.1	HTTP	488 GET / HTTP/1.1
42	2.062720	127.0.0.1	127.0.0.1	HTTP	591 HTTP/1.1 404 Not Found (text/html)
128	8.521351	127.0.0.1	127.0.0.1	HTTP	591 HTTP/1.1 404 Not Found (text/html)Continuation
70	3.660452	127.0.0.1	127.0.0.1	NTP	76 NTP Version 4, client
110	5.161997	127.0.0.1	127.0.0.1	NTP	76 NTP Version 4, client
134	8.552379	127.0.0.1	127.0.0.1	NTP	76 NTP Version 4, client
136	9.553555	127.0.0.1	127.0.0.1	NTP	76 NTP Version 4, client
176	11.055342	127.0.0.1	127.0.0.1	NTP	76 NTP Version 4, client
117	6.663983	127.0.0.1	127.0.0.1	DNS	76 Standard query 0x0a09 A watson.telemetry.microsoft.com
71	4.160990	127.0.0.1	127.0.0.1	LLMNR	61 Standard query 0x2d04 ANY DESKTOP-8TD8H4Q
174	10.054240	127.0.0.1	127.0.0.1	LLMNR	61 Standard query 0x3103 ANY DESKTOP-8TD8H4Q
72	4.661517	127.0.0.1	127.0.0.1	DNS	76 Standard query 0x3105 A watson.telemetry.microsoft.com
175	10.554867	127.0.0.1	127.0.0.1	LLMNR	61 Standard query 0x5c04 ANY DESKTOP-8TD8H4Q
49	2.659406	127.0.0.1	127.0.0.1	LLMNR	61 Standard query 0x6101 ANY DESKTOP-8TD8H4Q
111	5.662527	127.0.0.1	127.0.0.1	DNS	76 Standard query 0x6907 A watson.telemetry.microsoft.com
50	3.159946	127.0.0.1	127.0.0.1	DNS	76 Standard query 0x7702 A watson.telemetry.microsoft.com

Problems with this method

- Caleb the network admin is looking through his firewall logs and sees that LLMNR and DHCP are leaving through the gateway.
- This makes Caleb very sad.
- Caleb calls the security team and your red team engagement is now over.
- Also, the fact that you only get 1-3 bytes from a 100 byte packet sent

Stealthy vs Loud

- Stealthy is good, but really slow
- Sometimes get less than 1% useful data from each packet sent
- Stealth transfer can be unreliable for long messages
- Loud transfers are easier to implement and much faster

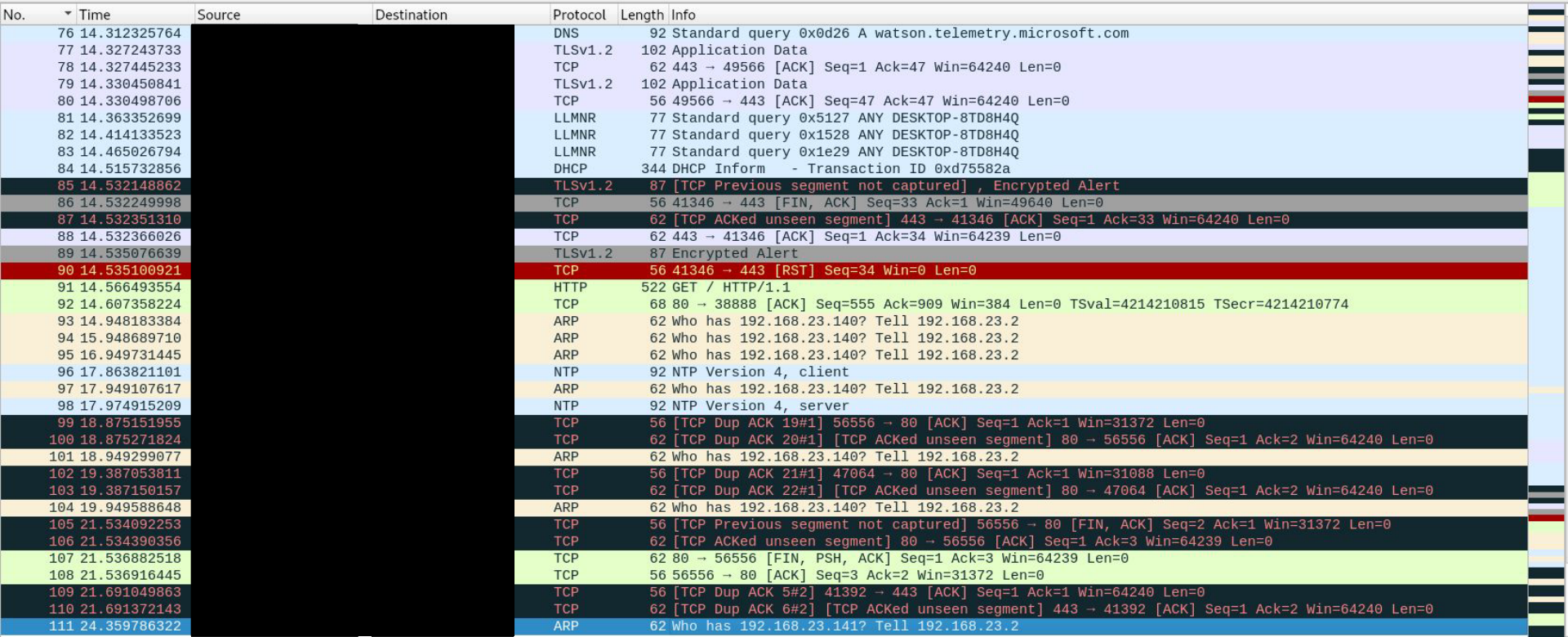
OCSP hollowing

Disadvantages of OCSP

Let’s play

SPOT
THAT
C2 DATA

Full Stealth, UDP Protocol hopping



No.	Time	Source	Destination	Protocol	Length	Info
76	14.312325764	[redacted]		DNS	92	Standard query 0x0d26 A watson.telemetry.microsoft.com
77	14.327243733			TLSv1.2	102	Application Data
78	14.327445233			TCP	62	443 → 49566 [ACK] Seq=1 Ack=47 Win=64240 Len=0
79	14.330450841			TLSv1.2	102	Application Data
80	14.330498706			TCP	56	49566 → 443 [ACK] Seq=47 Ack=47 Win=64240 Len=0
81	14.363352699			LLMNR	77	Standard query 0x5127 ANY DESKTOP-8TD8H4Q
82	14.414133523			LLMNR	77	Standard query 0x1528 ANY DESKTOP-8TD8H4Q
83	14.465026794			LLMNR	77	Standard query 0x1e29 ANY DESKTOP-8TD8H4Q
84	14.515732856			DHCP	344	DHCP Inform - Transaction ID 0xd75582a
85	14.532148862			TLSv1.2	87	[TCP Previous segment not captured] , Encrypted Alert
86	14.532249998			TCP	56	41346 → 443 [FIN, ACK] Seq=33 Ack=1 Win=49640 Len=0
87	14.532351310			TCP	62	[TCP ACKed unseen segment] 443 → 41346 [ACK] Seq=1 Ack=33 Win=64240 Len=0
88	14.532366026			TCP	62	443 → 41346 [ACK] Seq=1 Ack=34 Win=64239 Len=0
89	14.535076639			TLSv1.2	87	Encrypted Alert
90	14.535100921			TCP	56	41346 → 443 [RST] Seq=34 Win=0 Len=0
91	14.566493554			HTTP	522	GET / HTTP/1.1
92	14.607358224			TCP	68	80 → 38888 [ACK] Seq=555 Ack=909 Win=384 Len=0 TSval=4214210815 TSecr=4214210774
93	14.948183384			ARP	62	Who has 192.168.23.140? Tell 192.168.23.2
94	15.948689710			ARP	62	Who has 192.168.23.140? Tell 192.168.23.2
95	16.949731445			ARP	62	Who has 192.168.23.140? Tell 192.168.23.2
96	17.863821101			NTP	92	NTP Version 4, client
97	17.949107617			ARP	62	Who has 192.168.23.140? Tell 192.168.23.2
98	17.974915209			NTP	92	NTP Version 4, server
99	18.875151955			TCP	56	[TCP Dup ACK 19#1] 56556 → 80 [ACK] Seq=1 Ack=1 Win=31372 Len=0
100	18.875271824			TCP	62	[TCP Dup ACK 20#1] [TCP ACKed unseen segment] 80 → 56556 [ACK] Seq=1 Ack=2 Win=64240 Len=0
101	18.949299077			ARP	62	Who has 192.168.23.140? Tell 192.168.23.2
102	19.387053811			TCP	56	[TCP Dup ACK 21#1] 47064 → 80 [ACK] Seq=1 Ack=1 Win=31088 Len=0
103	19.387150157			TCP	62	[TCP Dup ACK 22#1] [TCP ACKed unseen segment] 80 → 47064 [ACK] Seq=1 Ack=2 Win=64240 Len=0
104	19.949588648			ARP	62	Who has 192.168.23.140? Tell 192.168.23.2
105	21.534092253			TCP	56	[TCP Previous segment not captured] 56556 → 80 [FIN, ACK] Seq=2 Ack=1 Win=31372 Len=0
106	21.534390356			TCP	62	[TCP ACKed unseen segment] 80 → 56556 [ACK] Seq=1 Ack=3 Win=64239 Len=0
107	21.536882518			TCP	62	80 → 56556 [FIN, PSH, ACK] Seq=1 Ack=3 Win=64239 Len=0
108	21.536916445			TCP	56	56556 → 80 [ACK] Seq=3 Ack=2 Win=31372 Len=0
109	21.691049863			TCP	56	[TCP Dup ACK 5#2] 41392 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
110	21.691372143			TCP	62	[TCP Dup ACK 6#2] [TCP ACKed unseen segment] 443 → 41392 [ACK] Seq=1 Ack=2 Win=64240 Len=0
111	24.359786322			ARP	62	Who has 192.168.23.141? Tell 192.168.23.2

“Partial Stealth” OCSP packets

1000	13.946751	127.0.0.1	127.0.0.1	TCP	40 35782 → 80 [ACK] Seq=39610 Ack=31383 Win=253 Len=0
1001	13.966771	127.0.0.1	127.0.0.1	HTTP	245 GET /UQyNesimaNaCqmO8tDgbsi1gLwuJ/AmaRxtjPKd8DcXBe0yfMqd+XeklLNiHxBO3vIvvJ2FfGTjhCpGe+gbVr%3D HTTP/1.1
1002	13.966771	127.0.0.1	127.0.0.1	TCP	40 80 → 35782 [ACK] Seq=31383 Ack=39815 Win=254 Len=0
1003	13.968272	127.0.0.1	127.0.0.1	TCP	161 80 → 35782 [PSH, ACK] Seq=31383 Ack=39815 Win=254 Len=121 [TCP segment of a reassembled PDU]
1004	13.968272	127.0.0.1	127.0.0.1	TCP	40 35782 → 80 [ACK] Seq=39815 Ack=31504 Win=253 Len=0
1005	13.988792	127.0.0.1	127.0.0.1	HTTP	245 GET /Ffe89rGyxvedcwwVQyEYUITPoVhVD0YjiAdUv95TYT6MbVZJUqaxPPnTY/J4W/RDA6+0ygNlBUeAhg62Dfqgl%3D HTTP/1.1
1006	13.988792	127.0.0.1	127.0.0.1	TCP	40 80 → 35782 [ACK] Seq=31504 Ack=40020 Win=253 Len=0
1007	13.990293	127.0.0.1	127.0.0.1	TCP	161 80 → 35782 [PSH, ACK] Seq=31504 Ack=40020 Win=253 Len=121 [TCP segment of a reassembled PDU]
1008	13.990293	127.0.0.1	127.0.0.1	TCP	40 35782 → 80 [ACK] Seq=40020 Ack=31625 Win=252 Len=0
1009	14.010812	127.0.0.1	127.0.0.1	HTTP	245 GET /+ePE12QaGO/QuPpqU0/veo/Y6qdzHI9fN4WI9tDKH3hL1g/zvcY2TsExXZzmbCvY5FhzvJsh6p0BXkW/tlxjy%3D HTTP/1.1
1010	14.010812	127.0.0.1	127.0.0.1	TCP	40 80 → 35782 [ACK] Seq=31625 Ack=40225 Win=252 Len=0
1011	14.012314	127.0.0.1	127.0.0.1	TCP	161 80 → 35782 [PSH, ACK] Seq=31625 Ack=40225 Win=252 Len=121 [TCP segment of a reassembled PDU]
1012	14.012314	127.0.0.1	127.0.0.1	TCP	40 35782 → 80 [ACK] Seq=40225 Ack=31746 Win=252 Len=0
1013	14.032333	127.0.0.1	127.0.0.1	TCP	52 45332 → 9229 [SYN] Seq=0 Win=8192 Len=0 MSS=65495 WS=256 SACK_PERM=1
1014	14.032333	127.0.0.1	127.0.0.1	TCP	40 9229 → 45332 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1015	14.032833	127.0.0.1	127.0.0.1	HTTP	245 GET /wtDifUEyONV1i8aK/FHrAz3I4RHmtNtE+q293IWFhHd/WfHehqDyxm1ZZy5b0h9CcfziXEPrG29NH1K3VGNtd%3D HTTP/1.1
1016	14.032833	127.0.0.1	127.0.0.1	TCP	40 80 → 35782 [ACK] Seq=31746 Ack=40430 Win=251 Len=0
1017	14.034335	127.0.0.1	127.0.0.1	TCP	161 80 → 35782 [PSH, ACK] Seq=31746 Ack=40430 Win=251 Len=121 [TCP segment of a reassembled PDU]
1018	14.034335	127.0.0.1	127.0.0.1	TCP	40 35782 → 80 [ACK] Seq=40430 Ack=31867 Win=251 Len=0
1019	14.054854	127.0.0.1	127.0.0.1	HTTP	245 GET /fiimnJfS3V+9MipGHGvm9Ni5TNJnO3yxYFEQwMV1ZFLO8ZbT8Or5VCMiGpcqT1OSt8YTRCydLxDcGrYuQHs/N%3D HTTP/1.1
1020	14.054854	127.0.0.1	127.0.0.1	TCP	40 80 → 35782 [ACK] Seq=31867 Ack=40635 Win=251 Len=0
1021	14.056355	127.0.0.1	127.0.0.1	TCP	161 80 → 35782 [PSH, ACK] Seq=31867 Ack=40635 Win=251 Len=121 [TCP segment of a reassembled PDU]
1022	14.056355	127.0.0.1	127.0.0.1	TCP	40 35782 → 80 [ACK] Seq=40635 Ack=31988 Win=251 Len=0
1023	14.076875	127.0.0.1	127.0.0.1	HTTP	245 GET /Qi/M97ScuRqA1rje/KBygT+Eli5uwTUnvN9w7Vp5YEmuyLfJ1dpTB728JHPgoBRbd2atu95a/FKY4yWpbuEK7%3D HTTP/1.1
1024	14.076875	127.0.0.1	127.0.0.1	TCP	40 80 → 35782 [ACK] Seq=31988 Ack=40840 Win=256 Len=0
1025	14.078377	127.0.0.1	127.0.0.1	TCP	161 80 → 35782 [PSH, ACK] Seq=31988 Ack=40840 Win=256 Len=121 [TCP segment of a reassembled PDU]
1026	14.078377	127.0.0.1	127.0.0.1	TCP	40 35782 → 80 [ACK] Seq=40840 Ack=32109 Win=250 Len=0
1027	14.098896	127.0.0.1	127.0.0.1	HTTP	245 GET //Th1i8NSFlSYsCTsfbkj0jzV8vHgCsf2Pyr2WTJGGHB/SZl8T5hbLVTJMb5Myvs4XgDSaNqbfar1T5dxAf3Za%3D HTTP/1.1
1028	14.098896	127.0.0.1	127.0.0.1	TCP	40 80 → 35782 [ACK] Seq=32109 Ack=41045 Win=255 Len=0
1029	14.100398	127.0.0.1	127.0.0.1	TCP	161 80 → 35782 [PSH, ACK] Seq=32109 Ack=41045 Win=255 Len=121 [TCP segment of a reassembled PDU]
1030	14.100398	127.0.0.1	127.0.0.1	TCP	40 35782 → 80 [ACK] Seq=41045 Ack=32230 Win=256 Len=0
1031	14.120497	127.0.0.1	127.0.0.1	HTTP	245 GET /oWm6vP8gdZM6uYXY3XW0qf13dHfR7dCuoksn0rOsd0UOFgiynIluITVlIYfno+0YZe1ou0pJSue5/ZRre7EWw%3D HTTP/1.1
1032	14.120497	127.0.0.1	127.0.0.1	TCP	40 80 → 35782 [ACK] Seq=32230 Ack=41250 Win=255 Len=0
1033	14.121998	127.0.0.1	127.0.0.1	TCP	161 80 → 35782 [PSH, ACK] Seq=32230 Ack=41250 Win=255 Len=121 [TCP segment of a reassembled PDU]
1034	14.121998	127.0.0.1	127.0.0.1	TCP	40 35782 → 80 [ACK] Seq=41250 Ack=32351 Win=256 Len=0
1035	14.142018	127.0.0.1	127.0.0.1	HTTP	245 GET //2GS6EE/uYYDaR1kWq6+u+cP+FyYJLncsYxI1jiCZTumF6rkBIkIojOU+AyWCCLMc6tNVe1GzQuFmuV9X6i0g%3D HTTP/1.1
1036	14.142018	127.0.0.1	127.0.0.1	TCP	40 80 → 35782 [ACK] Seq=32351 Ack=41455 Win=254 Len=0

And yet

- Did the glaring flaws in these exfiltration methods stop me from actually writing tools to do both of them?
- Of course it didn’t.
- In fact, if you want to try and recover some exfiltrated data, check out the Zaine’s Forensics category on TCTF

- Github link to tools: [redacted]

TCTF Challenge hints

- Youtube Procrastinator:
  - Flag is in that PCAP twice
- Localtoast/localtoastier are both protocol hopping stealth
- Whisper, Concert and Deafening all use OCSP cloaking
- Probably should automate Concert, Deafening, and Localtoastier

Upcoming Talks

- The full attack kill chain
- Lockpicking 101
- Hack The Box talks
- ...More!

Proud Sponsors

CACI

EVER VIGILANT

REDLattice

VOLGENAU SCHOOL

MASON

ENGINEERING

BATTELLE

It can be done™

CRYPSIS™