Ares - Technical Analysis | Zscaler Blog

By Brett Stone-Gross

Published: 2021-03-30 · Archived: 2026-04-02 11:04:00 UTC

Kronos is a banking trojan that first emerged in 2014 and marketed in underground forums as a crimeware kit to conduct credit card, identity theft, and wire fraud. In September 2018, a new Kronos variant named *Osiris* introduced several new features including TOR for command and control (C2) communications. The last update to Osiris appears to have been around mid-2019. In February 2021, Zscaler ThreatLabz identified a new Kronos variant that surfaced via spam campaigns to German speakers, which calls itself *Ares*. In Greek mythology, Ares is the son of Zeus and grandson of Kronos. Thus, the naming convention appears to refer to this new malware variant as the third generation of Kronos. Ares still appears to be in development alongside an information stealer that harvests credentials from various applications including VPN clients, web browsers, and the malware can exfiltrate arbitrary files and cryptocurrency wallets.

The threat actor behind this new variant continues to use both Osiris and Ares in parallel. In this blog post, we will examine these new malware developments and campaigns.

DarkCrypter

Recent samples of Osiris and Ares have been protected by a malware packer written in C++ that calls itself *DarkCrypter*. The packer contains the PDB path d:\scm\Italy\dopplegang\DarkCrypter\Bin\Clean.pdb. The code is not related to the commercial packer, DarkCrypter, that has been cracked and leaked online. Interestingly, the packer shares code with Kronos and Osiris including the string encryption algorithm. When the string table is decrypted, the first 41 entries are identical to older Kronos variants with eight new string additions (shown below) to detect sandbox environments:

```
atcuf32.dll
umengx86.dll
sandboxie.dll
libctc_sandbox.dll
atcuf64.dll
antimalware_provider32.dll
antimalware_provider64.dll
libctc_onexecute.dll
```

If the anti-analysis checks pass, the packer proceeds to the next step. There are at least two variants of the packer.

The first variant decrypts the next-stage payload using Blowfish. However, the decryption process uses a non-standard Blowfish key size. Typically, Blowfish key sizes are between 4 bytes and 56 bytes. However, the Blowfish decryption implementation in DarkCrypter supports a hardcoded key size that is 288 bytes (although only the first 72 bytes are effectively used). This may be designed to break cryptographic libraries that implement

Blowfish and follow the standard, where the maximum key size is limited to 56 bytes. The Blowfish key is located by computing a djb2 hash of each section name in the PE header. The code compares the resulting hash value with two hardcoded values that map to the section names .text (0xb80c0d8) and .sjdata (0xecae6faa).

The second variant of the DarkCrypter packer embeds the second-stage payload in a compressed format rather than an encrypted Blowfish format. The compression algorithm is identical to that found in Ares, and components related to Ares, including a packer that impersonates a bitmap image header.

Modified UPX Packer

The threat actor has also experimented with modifying UPX headers, which has well known section names. The modifications that have been made by the threat actor replace the UPX section names (UPX0, UPX1, ...) with standard section names like .text, .data, and .rdata. This breaks compatibility with the command-line UPX decompression tool, although the file can still be decompressed and executed. An example of the file header modifications are shown below in Figure 1 on the left, with the alterations highlighted in red.

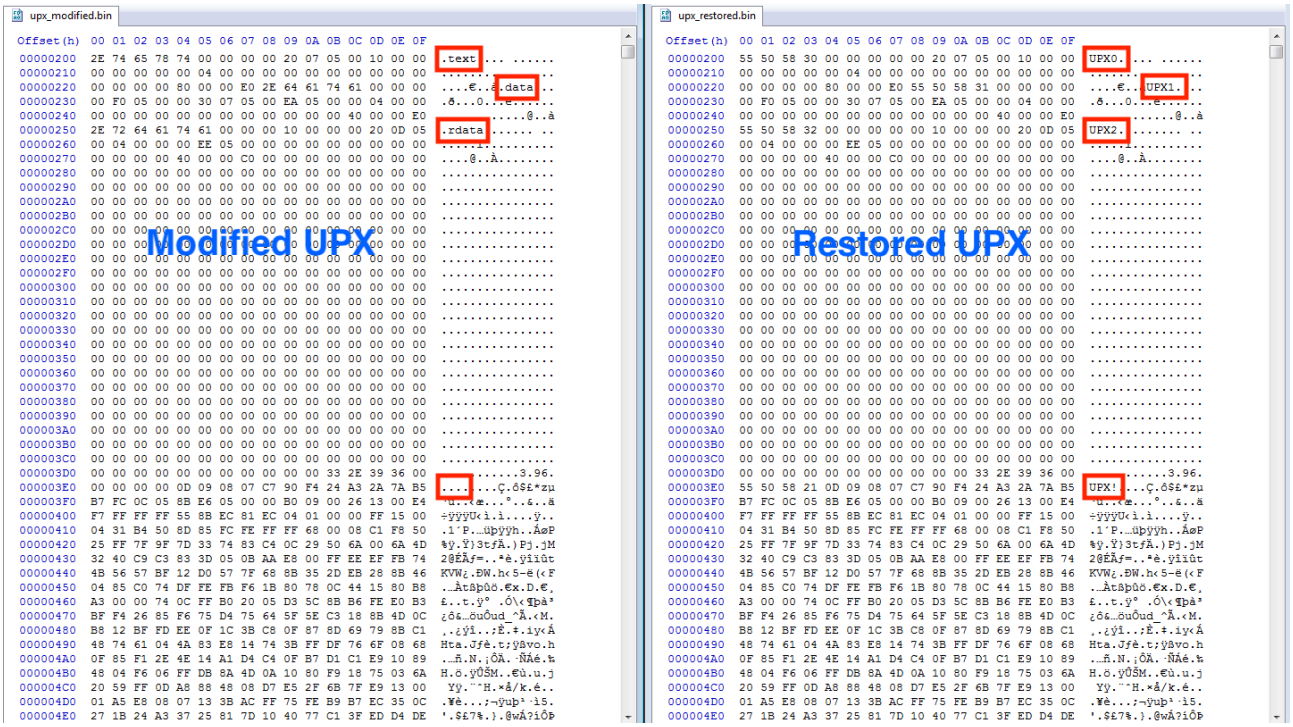

Figure 1. Modified and Restored UPX Headers

These changes can easily be restored to the original UPX section names as shown on the right in Figure 1. The UPX command-line utility can then be used to statically unpack this binary, producing the final executable payload.

BMPack

The threat actor has also been using another packer that Zscaler ThreatLabZ has dubbed BMPack. This packer has been utilized to pack both Osiris and Ares payloads. BMPack first decrypts embedded data using an XOR-based

algorithm, followed by RC4. After the decryption stage, the file appears to be a bitmap image as shown in Figure 2.

Figure 2. Fake Bitmap Image Used to Unpack Osiris and Ares Malware Payloads

However, a closer inspection reveals that the data is not actually a bitmap image, but has a specific sequence of data structures. By reverse engineering the packer, the format of the data structures can be determined, which consist of three DWORD values that represent the compressed size (red), uncompressed size (green), next offset (blue), followed by the compressed data (orange). An example of the first data structure is shown below in Figure 3.

```
00000000  42 4d b6 02 05 00 00 00  00 00 36 00 00 00 28 00  |BM........6...(.|
00000010  00 00 e0 01 00 00 1c ff  ff ff 01 00 18 00 00 00  |................|
00000020  00 00 80 02 05 00 00 00  00 00 00 00 00 00 00 00  |................|
00000030  00 00 00 00 00 00 0b 01  00 00 f0 01 00 00 86 07  |................|
00000040  00 00 9a 68 81 04 60 00  00 00 00 08 06 06 ff fe  |...h..`.........|
00000050  01 00 80 3b 70 1c 00 80  0a 1c 61 01 00 fe c0 c0  |...;p.....a.....|
00000060  c1 07 5d 0e 00 d0 4a d0  2c 04 ae 00 4c 9a 85 a0  |..]...J.,...L...|
```

Figure 3. Format of BMPack Data Structures

Each decompressed structure holds a different section of a PE file that is reconstructed and stitched together by a custom loader, and executed.

Ares Malware

Ares is being actively developed and the malware author continues to create and test new plugins and web injects. In the most recent Ares samples, there is an embedded DLL module that is compressed within the binary. The module contains an export that is designed to establish persistence. The code first copies itself to the location %APPDATA%\Adobe\AdobeNotificationUpdates.exe. It then creates a scheduled task named *AdobeNotificationUpdates* that is designed to execute Ares every two hours (with an expiration date of 2050-05-02 12:05:00). Similar persistence code is also found in many DarkCrypter samples.

The Ares persistence module has the same compilation prefix as other modules in its PDB path D:\scm\Italy\ares\source_ob\Release\startup.pdb. Ares attempts to locate an export name with the hash value

F4S4G3S4U7C6P2P7, which maps to the string ?Startup@@YAHPA_W@Z. Once the address of this function is located, Ares executes the module.

Ares uses the same function hashing algorithm as Kronos, which consists of calculating a CRC64 hash, converting the digest to uppercase hexadecimal characters. The result is then mapped to an alphanumeric value as shown in the Python code below:

```
digest = hexdigest(crc64(function_name)).upper()
out = ""

for i in range(len(digest)):
  if i & 1 != 0:
    val = ord(digest[i]) % 9 + ord('0')
  else:
    val = ord(digest[i]) % 25 + ord('A')
  out += chr(val)
return out
```

Ares contains most of the same code as its predecessors: Kronos and Osiris. However, there are several notable differences between Osiris and Ares, especially with respect to the C2 communications. Most Ares samples currently do not communicate with C2 servers over TOR. It is not quite clear, why most Ares samples have the TOR component removed, but it may be to reduce the malware's file size and evade corporate firewalls that block TOR network traffic. However, without TOR, the C2 servers are more vulnerable to takedown attempts. Some Ares samples attempt to address this limitation by hardcoding a large number of C2 URLs in the binary. Zscaler ThreatLabz has observed one Ares sample with 101 hardcoded C2 URLs.

Ares has also slightly modified the bot ID generation code, replacing the string Kronos with the string Ares as shown in Figure 4.

Figure 4. Comparison Between Kronos and Ares Bot ID Generation

Ares uses the HTTP query string parameters shown in Table 1. The HTTP request that sends the report.zip file is unique to Ares and discussed in more detail below.

Query String	Description

a=0	Send log data
a=1	Download web injects
a=2	Send keylogger data
a=3	Send report.zip file created by Ares Stealer
a=4	Request new commands

Table 1. Ares Query String Parameters

Ares Commands

Ares supports many of the same commands as Kronos and Osiris. However, some of the commands have been modified and the malware uninstall command (0x1) was removed. There are four modified commands that are supported by Ares as shown below in Table 2.

Command Number	Description
0x3	Set registry value name MSE to 0
0x4	Set registry value name MSE to 1
0x6	Download, decompress, map Ares Stealer into memory, and execute
0xC	Download, decompress, map module into memory, and execute

Table 2. New Commands Introduced By Ares

The commands 0x3 and 0x4 attempt to set a registry value name *MSE* to zero and one, respectively, under the registry key HKEY_CURRENT_USER\Software\Microsoft\CurrentVersion. However, this registry key does not exist and both functions will fail. This is likely an oversight by the malware author who accidentally left out *Windows* in this registry path between *Microsoft* and *CurrentVersion*. The registry value is not referenced elsewhere in Ares, so it may hint at a future use.

One of the most significant modifications is the command 0x6 that downloads, decompresses, and maps a PE file into memory, and executes it. Command 0x6 specifically searches for an export name with the hash value C3E0Q6R7F1H2G5A4, which maps to the string CollectInfo. The code passes two string parameters to the CollectInfo export. The first string is a pattern provided by the C2 server and the second is hardcoded to the string %APPDATA%\Google\report.zip. Zscaler ThreatLabZ has observed this Ares command being used to download a file from the URL http://mydynamite.dynv6[.]net/panel/upload/stealer.dll. The first four bytes of the response are the uncompressed file size. The file is decompressed using the same compression algorithm as BMPack.

Ares has code artifacts from the development of command 0x6. Samples contain an unreferenced function that attempts to open a file located at d:\scm\Italy\ares\source_ob\Binaries\Release\KittyDll.dll.cmp. The file is decompressed and mapped into memory using the same process as command 0x6. After the file is mapped, the export CollectInfo is called with the parameters: %userprofile%Documents|*.txt|5 and NULL. The purpose of these fields will be described in the next section. Note that there is a missing backslash character between %userprofile% and Documents. This string serves as a directory path, and without the backslash the path is invalid.

Zscaler ThreatLabZ has also identified Ares samples that contain another unreferenced function that loads a VNC plugin by attempting to open a file located at d:\scm\Italy\ares\\source_ob\Binaries\Release\vnc.dll.cmp. Similar to the stealer plugin, the file is decompressed, mapped into memory, and the export MakeItStart is called. The MakeItStart export name is resolved similar to the other Ares functions using the same CRC64-based hash algorithm and comparing the result with F0U5R4R6Q8H1P3E5. Ares then will terminate the VNC plugin by mapping the export name MakeItStop using the same process and comparing the result with the hash value C6P3T6Q8H1P3E5A8.

The command 0xC is the most recent modification to Ares and only found in newer samples.

Ares Stealer

Ares Stealer is downloaded by Ares and invoked via the export name CollectInfo. The malware is written in C++ and uses the Boost and Curl libraries. Ares Stealer has compilation artifacts showing that the Boost library was compiled in the directory d:\scm\Italy\tools\boost_1_74_0\boost. This directory prefix is identical to the DarkCrypter's PDB path and the location where the Ares unreferenced test functions attempt to load plugins from. This artifact along with the shared compression code suggests that the malware author likely has developed DarkCrypter, BMPack, Ares, and Ares Stealer.

The Ares Stealer export CollectInfo takes two parameters: a pipe-delimited string and a filename string. The pipe-delimited string takes three arguments, which are used by the stealer's file grabber feature. The first parameter is the directory in which to start the file enumeration process, the second parameter is a search pattern, and the last parameter is the directory search depth. The filename string is used to store the results of the extraction, which are added to a zip file.

An example command string observed from an Ares C2 server is %userprofile%|pass*.txt|5. This command will search a victim's user profile directory up to five levels deep for text files that have the prefix pass.

Ares Stealer collects detailed system information and harvests credentials for numerous applications including FTP clients, VPN clients, web browsers, instant messengers, and email clients. It can also steal files, cryptocurrency wallets, cookies, and credit cards.

The stealer will attempt to extract information from the following applications:

FTP clients

- Filezilla

VPN clients

- NordVPN
- OpenVPN
- ProtonVPN

Web browsers

- Mozilla Firefox
- Google Chrome
- Microsoft Edge
- Microsoft Internet Explorer
- Chromium
- Cyberfox
- BlackHawk
- Comodo IceDragon
- CometBird
- SeaMonkey
- Pale Moon
- Waterfox
- Mail.ru Atom
- Chromodo
- Uran
- CocCoc
- Nichrome
- Sputnik
- K-Meleon
- Maxthon 3
- 360 Browser
- Amigo
- Comodo Dragon
- Orbitum
- QIP Surf
- Liebao
- Coowon

- Catalina Group Citrio
- Fenrir Sleipnir
- Elements
- Kometa
- Chedot
- CentBrowser
- 7 Star
- Iridium
- MapleStudio ChromePlus
- Torch
- Yandex Browser
- Epic Privacy Browser
- Opera
- Brave Browser
- Vivaldi
- Blisk

Cryptocurrency wallet applications

- Coinomi
- Guarda
- Atomic Wallet
- Electrum
- Ethereum
- Exodus
- Bytecoin
- Armory
- Zcash
- Bitcoin
- Litecoin

Instant messenger clients

- Pidgin

Email clients

- Outlook

Osiris

The Osiris version that has been used by this threat actor contains a number of new features since the original version that appeared in April 2018. These updates were introduced around mid-2019 and include the following changes:

- New beacon request format that includes information about the compromised system
- Zlib compression to reduce the size of requests and responses (including web injects)
- Ability to deploy TeamViewer on a compromised host
- Ability to steal a victim's Outlook contacts via Nirsoft's OutlookAddressBookView utility
- Send spam emails to a victim's contact list
- New remote access capabilities

The threat actor has an Osiris C2 server that is located at http://ylnfkeznzg7o4xjf[.]onion/kpanel/connect.php, which has been instructing infected systems to steal and exfiltrate web browser and email credentials. The web browser harvesting command downloads a sqlite3 library from http://qqkzfkax24p4elax[.]onion/kpanel/upload/sqlite3.dll, which is a dependency to extract Google Chrome passwords. A second module for harvesting Firefox credentials from a 64-bit system is downloaded from http://qqkzfkax24p4elax[.]onion/kpanel/upload/ffc64.exe.

The C2 is also serving a web inject configuration file, which targets clients at German financial institutions with the URL patterns shown below:

set_url https://*commerzbank.de* GPI

set_url https://*.de/*/entry* GPI

set_url https://*.de/banking-*/portal?* GPI

set_url https://*.de/banking-*/portal;* GPI

set_url https://*.de/portal/portal* GPI

set_url https://*.de/privatkunden/* GPI

set_url https://*.de*abmelden* GPI

set_url https://*.de/de/home* GPI

set_url https://*.de/en/home* GPI

set_url https://*.de/fi/home* GPI

set_url https://*banking.sparda.de* GPI

set_url https://*banking.sparda-* GPI

set_url https://*banking.sparda.de/wps/loggedout.jsp GPI

set_url https://*meine.deutsche-bank.de/trxm/db* GPI

set_url https://*banking.berliner-bank.de/trxm* GPI

set_url https://*meine.norisbank.de/trxm/noris* GPI

set_url https://*targobank.de* GPI

When a victim browses to a website that matches one of these patterns, JavaScript code will be injected from the threat actor's domain https://securebankingapp[.]com/.

The full list of web injects for this Osiris instance is shown here.

The threat actor has another active Osiris C2 server located at http://qqkzfkax24p4elax[.]onion/kpanel/connect.php. This C2 server is also serving commands to exfiltrate credentials, but the web inject configuration file is blank. However, the C2 server is also providing commands to extract a victim's email contact list using Nirsoft's OutlookAddressBookView, which is downloaded from the following locations:

http://qqkzfkax24p4elax[.]onion/kpanel/upload/oabv32.exe (32-bit)

http://qqkzfkax24p4elax[.]onion/kpanel/upload/oabv64.exe (64-bit)

Conclusion

Ares is a new fork of the Kronos banking trojan that appears to be in the early stages of development. The code contains several bugs and unreferenced code segments that are likely used for debugging purposes. The threat actor has invested significant resources in building DarkCrypter, BMPack, Ares, and Ares Stealer. Therefore, activity related to this threat is likely to increase as the malware continues to mature.

Detections

Zscaler's multilayered cloud security platform detects indicators at various levels, as shown below:

Win32.Banker.Kronos

Win32.Banker.Kronos.LZ

MITRE ATT&CK Table

Tactic	Technique
T0011	Command and Control
T1053	Scheduled Task/Job
T1078	Valid Accounts

T1087	Account Discovery
T1090	Proxy
T1185	Man in the Browser
T1219	Remote Access Software
T1497	Virtualization/Sandbox Evasion
T1552	Unsecured Credentials
T1573	Encrypted Channel
T1592	Gather Victim Host Information

Indicators of Compromise (IOCs)

The following IOCs can be used to detect Osiris and Ares infections.

Samples

SHA256 Hash	Module Name
da767e6faf97d73997f397eae71b372a549dd6331bf8ec0ebd398ef8cfe9a47e	Osiris sample
5e7642e945bd05ecea77921cb3464b6da8db59e5ff38240608e3cbb44b07fb1d	Osiris sample
7498e37c332d55c14247ae4b675e726336a8683900d8fd1da412905567d2de4a	Ares sample

e5d624b7060c0e885abe11a0973a43a355c9930fc6912ff5eac83d1a9eec9c29	Ares sample
035793d479c4229693fc6dcceaa639cd51ae89334b43e552b9c47a6dea68ce30	Ares sample with embedded Startup module
94b084ea925990742f4eaaada1eef9a42c13066bf4f4c7a3b12a1509e32ff9e6	Ares Stealer sample
09897c6ef88b9e9bc20917a2b47ec86ff2b727a2923678f5e2df6bb6437d3312	Ares VNC plugin
896cebf465257f60347e58ffd7ec61629cf530956ef9b00e94f8b40ef9b30581	DarkCrypter with second-stage BMPack and Osiris sample
956ae36f40d0d847daa00d7964906e7e9d1671d0f3f2e7d257d5a8d324388c31	DarkCrypter sample with encrypted Ares payload
6c5dac9043b2f112543f3eca6503d4bcc70d762b47d75dcb85f9767c603de56f	DarkCrypter sample with compressed Ares TOR payload
b3348405cd0fa66661b46bc6cbab97b55708be26a2ed7a745e1632b46d1b3f41	DarkCrypter sample with encrypted Ares payload
4044abad9a846e203f131c65b1f84bb2b79f94000d1d7be6c6d6a8e27ac76940	BMPack sample with Osiris payload

Network Indicators

Domain / IP Address	Description
http://ylnfkeznzg7o4xjf[.]onion/kpanel/connect.php	Osiris C2 URL
http://m3r7ifpzkdix4rf5[.]onion/kpanel/connect.php	Osiris C2 URL

http://qqkzfkax24p4elax[.]onion/kpanel/connect.php	Osiris C2 URL
https://securebankingapp[.]com	Osiris web inject domain
http://vbyrduc537l5po3w[.]onion/panel/connect.php	Ares C2 URL
http://wifoweijijfoiwjweoi[.]xyz/panel/connect.php	Ares C2 URL
http://ddkiiqefmiir[.]xyz/panel/connect.php	Ares C2 URL
http://ddkiilefmjim[.]xyz/panel/connect.php	Ares C2 URL
http://ddkiieeelkif[.]xyz/panel/connect.php	Ares C2 URL
http://ddkiiofelkkq[.]xyz/panel/connect.php	Ares C2 URL
http://ddkiihfelikh[.]xyz/panel/connect.php	Ares C2 URL
http://ddkiiffdkijh[.]xyz/panel/connect.php	Ares C2 URL
http://ddkiigedliji[.]xyz/panel/connect.php	Ares C2 URL
http://ddkiirfdmjks[.]xyz/panel/connect.php	Ares C2 URL
http://ddkiitefkkju[.]xyz/panel/connect.php	Ares C2 URL
http://mydynamite.dynv6[.]net/panel/connect.php	Ares C2 URL
http://cabletv[.]top/panel/connect.php	Ares C2 URL

Yara rules

These rules are valid on unpacked Kronos, Osiris, and Ares binaries.

```
rule kronos_string_decryption
{
  strings:
    $ = {6a 1e 5f f7 f7 8b 45 08 8d 3c 1e 8a 04 38 8a ?? ?? ?? ?? ?? 32 c2}
    $ = {55 8b ec 51 8b 4d 08 c1 e1 04 8b ?? ?? ?? ?? ?? 8a}
  condition:
    all of them
}

rule kronos_api_strings
{
  strings:
    $ = "D7T1H5F0F5A4C6S3"
    $ = "H2G3F4F0F5A4D5E6"
    $ = "X1U5U8H8F5A4C8C5"
    $ = "E3D7R6B3R4H5F3R7"
    $ = "X8D3U3P7S6Q3S5R1"
    $ = "X8D3T6Q6U3S3A6R1"
    $ = "R6G2D2R3A5E3C4U5"
    $ = "H7Y6G2R3A5F4D3S8"
    $ = "P7Y3Q5P0Y8C2Y6F6"
    $ = "R6Y7B3C6E7E6T7U7"
    $ = "G2F3G6A6R3F1P6G2"
    $ = "S3H8T8Y5F5B5B0X0"
    $ = "C8G2T3U3B1H3T5B5"
    $ = "C4R7A2P4X3B1H5A4"
    $ = "R3Q7T7Q2R6S1Y3R5"
    $ = "E3C3A2Y3C4U6S5F5"
    $ = "F3P7Y6P3U3E2U5F3"
    $ = "E5X0A4Q4F0Y0D6E2"
    $ = "X2R0A4Q4F0Y0D6F3"
    $ = "H1G7R4Y7D1E6R5F8"
    $ = "G3C3R4H7R5T8E5R8"
    $ = "F6H5P7T4F6D6Y6D4"
    $ = "E3C7U2Y3C3R6R5D5"
    $ = "F5E8X5G3Q6T7E6T3"
    $ = "E1U3D5F7R2Y5S0H4"
    $ = "H3Y5C8Y2D4U8Y4S3"
    $ = "U0U6H1T2F6S1P2Y5"
    $ = "D5R3T8D5D3H0B4E2"
    $ = "D5B6G6R4A6H1P7A3"
    $ = "F1Q3D0H4H3T6U1X5"
    $ = "A4T6P1G7D6G0F3S5"
```

```
    $ = "C7G5T6P7U5B1H0F5"
    $ = "X2C7E3U6F3A7Y1D5"
    $ = "P4Y7T7R7R8X3E3A3"
    $ = "C5Y7R2R2H1R7A1B2"
    $ = "S4A3E3S3S4T1T3D1"
    $ = "B4Y2H7F8A2T3G4H3"
    $ = "B5D6X4H5G6S3R2B5"
    $ = "B6F6X4A8R5D3A7C6"
    $ = "C6P7E6P7A1R5Q4R7"
    $ = "R8S7D7S8H6Y4T6B7"
    $ = "U0S3T3D3U5F5B4E8"
    $ = "F6C3U4P4X3B1H3T5"
    $ = "T2F2T3U2H5B1C1A7"
    $ = "T0E0H4U0X3A3D4D8"
    $ = "C5R4X4H7R5T7A5R6"
    $ = "D3S0A7R4F6C8F2R5"
    $ = "Y1C1B6A7H3C0E7E7"
    $ = "H2E7A5B8Q6G3S7Y3"
    $ = "D3Q5F2F3R5Y5Y8S2"
    $ = "Y2C3G8R5R3A5F5B4"
    $ = "F1D2B6A5T3X2C8R1"
    $ = "G5D3P2G0F6G2H8E6"
    $ = "Y6Q6P2G0E5E6G2H8"
    $ = "Y7D3F3S7X2S4F2X3"
    $ = "X7D0E3R2R4Q0E4D3"
  condition:
    25 of them
}
```

Snort rules

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Zscaler TROJAN Ares Command Beacon"; flow:establi
```

Explore more Zscaler blogs

Source: https://www.zscaler.com/blogs/security-research/ares-malware-grandson-kronos-banking-trojan