

MAR-10271944-1.v1 – North Korean Trojan: HOTCROISSANT

 us-cert.gov/ncas/analysis-reports/ar20-045d

Notification

This report is provided "as is" for informational purposes only. The Department of Homeland Security (DHS) does not provide any warranties of any information contained herein. The DHS does not endorse any commercial product or service referenced in this bulletin or otherwise.

This document is marked TLP:WHITE--Disclosure is not limited. Sources may use TLP:WHITE when information carries minimal or no foreseeable accordance with applicable rules and procedures for public release. Subject to standard copyright rules, TLP:WHITE information may be distributed. For more information on the Traffic Light Protocol (TLP), see <http://www.us-cert.gov/tlp>.

Summary

Description

This Malware Analysis Report (MAR) is the result of analytic efforts between Department of Homeland Security (DHS), the Federal Bureau of Investigation (FBI), and the Department of Defense (DoD). Working with U.S. Government partners, DHS, FBI, and DoD identified Trojan malware variants used by the North Korean government. This malware variant has been identified as HOTCROISSANT. The U.S. Government refers to malicious cyber activity by the North Korean government as HIDDEN COBRA. For more information on HIDDEN COBRA activity, visit <https://www.us-cert.gov/hiddencobra>.

DHS, FBI, and DoD are distributing this MAR to enable network defense and reduce exposure to North Korean government malicious cyber activity.

This MAR includes malware descriptions related to HIDDEN COBRA, suggested response actions and recommended mitigation techniques. Use should flag activity associated with the malware and report the activity to the Cybersecurity and Infrastructure Security Agency (CISA) or the FBI (CyWatch), and give the activity the highest priority for enhanced mitigation.

This report looks at a full-featured beaconing implant. This sample performs a custom XOR network encoding and is capable of many features including system surveys, file upload/download, process and command execution, and performing screen captures.

For a downloadable copy of IOCs, see [MAR-10271944-1.v1.stix](#).

Submitted Files (1)

8ee7da59f68c691c9eca1ac70ff03155ed07808c7a66dee49886b51a59e00085 (svchost.exe)

IPs (1)

94.177.123.138

Findings

8ee7da59f68c691c9eca1ac70ff03155ed07808c7a66dee49886b51a59e00085

Tags

trojan

Details

Name	svchost.exe
Size	117760 bytes
Type	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	062e9cd9cdcab928fc6186c3921e945
SHA1	566347f8bf30f66aec670d660091fb6bb03a0650
SHA256	8ee7da59f68c691c9eca1ac70ff03155ed07808c7a66dee49886b51a59e00085
SHA512	e16fefb72fb466e31f982ea1d3f5e5754af289dfe7c8e7c2c6859b462b02e8715eaedf271985465931983fe0800f93e2943c715929f73136
ssdeep	3072:kRdIGZdOwoyeCJkLURXSOpw1yIR3vbRY7a:y3wMae2W9O+NR3DR0a
Entropy	6.282477

Antivirus

Ahnlab	Trojan/Win32.Agent
Avira	HEUR/AGEN.1039759
BitDefender	Gen:Variant.Jaiko.2546
Emsisoft	Gen:Variant.Jaiko.2546 (B)
Ikarus	Trojan.Win32.KillAV

YARA Rules

```
rule CryptographyFunction
{
  meta:
    author = "CISA trusted 3rd party"
    incident = "10271944.r1.v1"
    date = "2019-12-25"
    category = "Hidden_Cobra"
    family = "HOTCROISSANT"
  strings:
    $ALGO_crypto_1 = { 8A [1-5] 32 [1-4] 32 [1-4] 32 [1-4] 88 [1-5] 8A [1-4] 32 [1-4] 22 [1-4] 8B [1-5] 8D [3-7] 33 [1-4] 81 [3-7] C1 [1-5] C1 [
5] 33 [1-4] 22 [1-4] C1 [1-5] 33 [1-4] 32 [1-4] 8B [1-4] 83 [1-5] C1 [1-5] 33 [1-4] C1 [1-5] C1 }
  condition:
    uint16(0) == 0x5A4D and any of them
}
```

ssdeep Matches

No matches found.

PE Metadata

Compile Date 2019-07-25 11:38:54-04:00

Import Hash 9e7d183f56ad974fbd6c056d20051ef8

PE Sections

MD5	Name	Raw Size	Entropy
760c39c49aa3a2cb4ec9f6fd5d4524e6	header	1024	2.537779
8480a50e20d57bcb86fa649691ca9e0c	.text	80896	6.619532
36d3f909d39d54fd628e1d66d6acd26e	.rdata	18432	5.282847
a497350b0c256c943b59382e0a2e884a	.data	9216	2.905698
2d5b9737e8cd3def95c4fc6527741f91	.rsrc	1024	2.112640
9b5d24778302d0f050a93778c9cab3ef	.reloc	7168	4.675041

Packers/Compilers/Cryptors

Microsoft Visual C++ ?..?

Description

The sample performs dynamic DLL importing and API lookups using LoadLibrary and GetProcAddress on obfuscated strings in an attempt to hide network functions. However, only a small number of API calls are obfuscated this way, and their selection is not consistent through the sample.

The sample obfuscates strings used for API lookups as well as the strings used during the network handshake using a simple Byte xor with 0x0f.

The sample attempts to connect to a hardcoded C2 IP and then immediately sends it's Victim Info. It then listens for commands from the C2 and r Network communications are first zipped and then encoded with a custom xor algorithm. The session structure (Figure 1), packet format (Figure 2 (Figure 3), a Python 3 script to decrypt network traffic, and implant functionality (Figure 4) are given below.

--Begin Hardcoded IP and Port--

94.177.123.138:8088

--End Hardcoded IP and Port--

--Begin Python 3 Network Communication Decode Script--

```
def decode(data):
  dec = []
  key1 = 0x17
  key2 = 0x00b8d68b
  key3 = 0x02497029
  for i in range(len(data)):
    temp2 = key2
    temp3 = key3
    dec.append((data[i] ^ temp2 ^ temp3 ^ key1) & 0xff)
    key2 = key2 >> 8 | (((key2 * 8 ^ key2) & 0x7f8) << 0x14) & 0xffffffff)
    key1 = key1 & temp3 ^ (temp3 ^ key1) & temp2
```

```
key3 = key3 >> 8 | (((((((key3 * 2 ^ key3) << 4) & 0xfffffff) ^ key3) &
0xfffff80 ^ key3 << 7) & 0xfffffff) << 0x11) & 0xfffffff);
return bytes(dec)
```

--End Python 3 Network Communication Decode Script--
Screenshots

Implant	Direction	C2
Victim Info	>>>>>>>>	
	<<<<<<<<	Command
Command results	>>>>>>>>	

Figure 1 - Session Structure.

Data	Example
Username	user
Administrator	"Administrator" or "Not Administrator"
Victim IP	127.0.0.1
Windows Product Name	Windows 10 Enterprise /
Windows Product Name	Windows 10 Enterprise /
Processor Name	
Screen Resolution	1920 x 1080
Physical RAM	2047 MB

Figure 2 - Victim Information Structure.

Opcode	Operation	Arguments	Description
0x07c7, 0x07c8	KeepAlive		Prevents the connection from closing due to inactivity
0x07d2	ProcessList		Lists all running processes
0x07d3	ProcessKill	<process_name>	Kills process with specified name.
0x07d4	WindowsList		Lists all open windows.
0x07d7	WindowClose	<window_title>	Closes window with the specified title.
0x07da	DriveList		List used drive letters and disk space
0x07de	DirectoryList	<directory>	Returns a list of all files in the current or specified directory
0x07de	FileCopy	<old_filename> <new_filename>	Copies a specified file to a new location
0x07e0	FileDelete	<filename>	Deletes a specified file or directory
0x07e2	FileMove	<old_filename> <new_filename>	Moves a specified file to a new location
0x07e6	FileReadSend	<filename>	Sends a file from the victim machine to the C2
0x07e8	FileFind	<filename>! <search_directory>	Recursively searches given path for given filename
0x07ea	Execute	<filename>	Uses ShellExecuteA open to execute the file. opt2 determines if window is hidden or not
0x07ed, 0x07e4, 0x07e5	FileRecvWrite	<filename>	Victim machine receives a file from the C2. (0x07ed open file, 0x07e4 write chunk, 0x07e5 close file)
0x07ee	DirectoryGet	<directory>	Recursively downloads all files in the given directory
0x0802	ScreensCaptureStart		Starts capturing the victim's screen
0x0804	ScreensCaptureStop		Stops capturing the victim's screen
0x0820	ServiceList		Lists all services
0x0821	ServiceStart	<service_name>	Starts the specified service
0x0822	ServiceQuery	<service_name>	Queries the status of the specified service
0x0bc2	AppList		Enumerates registry key "SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths"
0x0fa1, 0x0fa2, 0x0fa3	ReverseShell	<command>	Runs the specified shell command. (0x0fa1 runs cmd.exe, 0x0fa2 accepts/runs commands, 0x0fa3 kills cmd.exe)

Figure 3 - Implant Functionality. The following commands from the table above appear to be broken: ProcessKill - Programmer coding error that r violation. It attempts to decode an obfuscated string (Kernel32.dll) in-place instead of doing a string copy first like they do everywhere else. Wind handle used to loop through all windows is never initialized.

<p>Packet Format: <4 Bytes Compressed size> <4 Bytes Decompressed size> <Compressed/Encoded data></p> <p>Compressed/Encoded Data Format: <4 Byte Opcode> <4 Bytes opt1> <4 Bytes opt2> <4 Bytes opt3> <4 Bytes command size> <command/arguments></p>
--

Figure 4 - Packet Structure.

94.177.123.138

Tags

command-and-control

Ports

Description

8EE7DA59F68C691C9ECA1AC70FF03155ED07808C7A66DEE49886B51A59E00085 connects to this C2 IP address.

Recommendations

CISA recommends that users and administrators consider using the following best practices to strengthen the security posture of their organization. Configuration changes should be reviewed by system owners and administrators prior to implementation to avoid unwanted impacts.

- Maintain up-to-date antivirus signatures and engines.
- Keep operating system patches up-to-date.
- Disable File and Printer sharing services. If these services are required, use strong passwords or Active Directory authentication.
- Restrict users' ability (permissions) to install and run unwanted software applications. Do not add users to the local administrators group unless necessary.
- Enforce a strong password policy and implement regular password changes.
- Exercise caution when opening e-mail attachments even if the attachment is expected and the sender appears to be known.
- Enable a personal firewall on agency workstations, configured to deny unsolicited connection requests.
- Disable unnecessary services on agency workstations and servers.
- Scan for and remove suspicious e-mail attachments; ensure the scanned attachment is its "true file type" (i.e., the extension matches the file name).
- Monitor users' web browsing habits; restrict access to sites with unfavorable content.
- Exercise caution when using removable media (e.g., USB thumb drives, external drives, CDs, etc.).
- Scan all software downloaded from the Internet prior to executing.
- Maintain situational awareness of the latest threats and implement appropriate Access Control Lists (ACLs).

Additional information on malware incident prevention and handling can be found in National Institute of Standards and Technology (NIST) Special Publication 800-151, **"Guide to Malware Incident Prevention & Handling for Desktops and Laptops"**.

Contact Information

CISA continuously strives to improve its products and services. You can help by answering a very short series of questions about this product at <https://us-cert.gov/forms/feedback/>

Document FAQ

What is a MIFR? A Malware Initial Findings Report (MIFR) is intended to provide organizations with malware analysis in a timely manner. In most cases, the report will provide initial indicators for computer and network defense. To request additional analysis, please contact CISA and provide information regarding the level of desired analysis.

What is a MAR? A Malware Analysis Report (MAR) is intended to provide organizations with more detailed malware analysis acquired via manual analysis. To request additional analysis, please contact CISA and provide information regarding the level of desired analysis.

Can I edit this document? This document is not to be edited in any way by recipients. All comments or questions related to this document should be directed to CISA at 1-888-282-0870 or soc@us-cert.gov.

Can I submit malware to CISA? Malware samples can be submitted via three methods:

- Web: <https://malware.us-cert.gov>
- E-Mail: submit@malware.us-cert.gov
- FTP: <ftp://malware.us-cert.gov> (anonymous)

CISA encourages you to report any suspicious activity, including cybersecurity incidents, possible malicious code, software vulnerabilities, and phishing scams. Reporting forms can be found on CISA's homepage at www.us-cert.gov.

Revisions

February 14, 2020: Initial Version

This product is provided subject to this [Notification](#) and this [Privacy & Use](#) policy.

Please share your thoughts.

We recently updated our anonymous [product survey](#); we'd welcome your feedback.