MAR-10257062-1.v2 - North Korean Remote Access Tool: FASTCASH for Windows | CISA

Published: 2020-09-01 · Archived: 2026-04-06 00:42:37 UTC

Notification

Summary

Description

This Malware Analysis Report (MAR) is the result of analytic efforts between the 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 Remote Access Tool (RAT) malware variants used by the North Korean government. This malware variant has been identified as FASTCASH for Windows. 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.

FBI has high confidence that HIDDEN COBRA actors are using malware variants in conjunction with proxy servers to maintain a presence on victim networks and to further network exploitation. 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. Users or administrators should flag activity associated with the malware and report the activity to the Cybersecurity and Infrastructure Security Agency (CISA) or the FBI Cyber Watch (CyWatch), and give the activity the highest priority for enhanced mitigation.

This submission included two unique files. The first file is a malicious application, which can be utilized to inject a dynamic link library (DLL) into a remote Windows process. The second file is a malicious Windows DLL. The DLL contains two functions that can hook callbacks to the Windows application programming interfaces (APIs) "Send" and "Recv" within a targeted process. These hook functions are utilized to intercept traffic received by the target process. In received Financial Messages, the malicious functions will look for targeted Primary Account Numbers (PAN) to deliver a custom response. It appears the malware will target a system on a bank infrastructure, which is designed to process automated teller machine (ATM) transactions.

This updated report included an additional sample that is used by advanced persistent threat (APT) cyber actors in the targeting of banking payment systems. The sample is a man-in-the-middle bank transaction modification malware. Once the malware is injected into an executable, it takes control of the send and receive functions in order to identify, log, and modify ISO 8583 messages. ISO 8583 is an international standard for financial transaction card originated interchanged messaging. This functionality enables the actor to withdraw more money than is actually available. The malware specifically targets ISO 8583 Point of Sale (POS) system messages, ATM transaction requests, and ATM balance inquiries. The sample uses code from open source repositories on the Internet and modifies the parsing code to support Extended Binary Coded Decimal Interchange Code (EBCDIC) encoding. EBCDIC is a character encoding format like the more commonly ASCII.

For a downloadable copy of IOCs, see MAR-10257062-1.v2.stix.

Submitted Files (3)

129b8825eaf61dcc2321aad7b84632233fa4bbc7e24bdf123b507157353930f0 (switch.dll)

39cbad3b2aac6298537a85f0463453d54ab2660c913f4f35ba98fffeb0b15655 (switch.exe)

5cb7a352535b447609849e20aec18c84d8b58e377d9c6365eafb45cdb7ef949b (A2B1A45A242CEE03FAB0BEDB2E4605...)

Findings

129b8825eaf61dcc2321aad7b84632233fa4bbc7e24bdf123b507157353930f0

Tags

HIDDEN-COBRAtrojan

Details

Name	switch.dll
Size	118784 bytes
Type	PE32 executable (DLL) (console) Intel 80386, for MS Windows
MD5	c4141ee8e9594511f528862519480d36
SHA1	2b22d9c673d031dfd07986906184e1d31908cea1
SHA256	129b8825eaf61dcc2321aad7b84632233fa4bbc7e24bdf123b507157353930f0
SHA512	dfc1ad2cb2df2b79ac0f2254b605a2012b94529ac220350a4075e60b06717918175cff5c22e52765237b78ec4edffd6df20f333e28a405a4339a1
ssdeep	3072:lUGDXTpE8AKDKDOf+8ZagCfG4aAzFdIARrhxg6/ZpDA:+GDXTpFDKDMZagX4aAB2Cg6hpD
Entropy	6.454745

Antivirus

Antiy	Trojan/Win32.Tiggre
Avira	TR/Spy.Banker.pubvd
BitDefender	Trojan.GenericKD.32541173
ClamAV	Win.Trojan.Alreay-7189205-0
Comodo	Malware
ESET	a variant of Win32/NukeSped.GA trojan
Emsisoft	Trojan.GenericKD.32541173 (B)
Ikarus	Trojan.Spy.Banker
K7	Riskware (0040eff71)
Lavasoft	Trojan.GenericKD.32541173
McAfee	Trojan-Banking
NANOAV	Trojan.Win32.NukeSped.gexoae
Sophos	Troj/Banker-GYS
Symantec	Trojan Horse
TrendMicro	Backdoo.62DC2502
TrendMicro House Call	Backdoo.62DC2502
VirusBlokAda	BScope.TrojanBanker.Agent
Zillya!	Trojan.NukeSped.Win32.183

YARA Rules

- rule CISA_10257062_01 : ATM_Malware
  {
    meta:
      Author = "CISA Code & Media Analysis"
      Incident = "10257062"
      Date = "2019-09-26"
      Last_Modified = "20200117_1732"
      Actor = "n/a"
      Category = "Financial"
      Family = "ATM_Malware"

```
        Description = "n/a"
        MD5_1 = "c4141ee8e9594511f528862519480d36"
        SHA256_1 = "129b8825eaf61dcc2321aad7b84632233fa4bbc7e24bdf123b507157353930f0"
    strings:
        $x3 = "RECV SOCK= 0x%p, BUF= 0x%p, LEN= 0x%08X, RET= %08X, IP= %s, Port= %d" fullword ascii
        $x4 = "init_hashmap succ" fullword ascii
        $x5 = "89*(w8y92r3y9*yI2H28Y9(*y3@*" fullword ascii
    condition:
        ($x3) and ($x4) and ($x5)
}
```

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2019-06-22 01:59:31-04:00
Import Hash	0ab159bd939411cb8df935bd9e7b5835

PE Sections

MD5	Name	Raw Size	Entropy
00f8301c11847b70346d6271098d8f1c	header	1024	2.296500
c3bee35076d728ce32b67f5bc66587f3	.text	84992	6.641787
6b094443cad879acc7285f991243ddb0	.rdata	17920	5.170073
11060bd3e49075b78be8670ff46d9a48	.data	7168	4.275765
3637e0cd32608b060e308fdd9742ea97	.reloc	7680	4.792696

Packers/Compilers/Cryptors

Microsoft Visual C++ DLL *sign by CodeRipper

Description

This file is a malicious Windows 32-bit DLL. Upon execution, it attempts to read the file "c:\\temp\info.dat". Analysis of this implant indicates the encrypted file "info.dat" will contain targeted PAN numbers, which are expected to be contained within transactions possibly originating from ATM systems. Analysis indicates the malware decrypts "info.dat" utilizing what appears to be the AES encryption algorithm. The key utilized for this decryption is displayed below:

--Begin Decryption Key--

89*(w8y92r3y9*yIy(8Y23RHWIEFH238

--End Decryption Key--

The decrypted contents of "info.dat" are then parsed. Sub-components of the file are then further decoded using a hard-coded rotating XOR cipher (Figure 1). The data used as the rotating XOR cipher key is displayed below:

--Begin Rotating XOR Cipher Key--

963007772C610EEEBA51099919C46D078FF46A7035A563E9A395649E3288DB0EA4B8DC791EE9D5E088D9D2972B4CB609BD7CB17E072DB8E

--End Rotating XOR Cipher Key--

This application will not run without the file "info.dat", which was not available at the time of analysis.

Upon execution, the malware creates the directory "C:\tmp_DMP". The malware will use this location as a working directory on the targeted system. The malware will store run time logs within this folder. When executed, the malware will create a log file with the following file name format "c:\\tmp_DMP\\TMPL_%d_%d.tmp" in this folder and stamps it with the data "HK-Start".

This binary contains two functions, which provides context to the malware's purpose and capability. Analysis indicates this DLL is injected into a targeted process. In order to capture and analyze incoming network traffic, the malware hooks the "Send" and "Recv" Windows API within a targeted process. One of these functions, located at offset "0x00004f60", appears

to search for incoming network traffic for "x200" Financial Request Messages, such as the type that may be generated from an ATM banking system. When the malware captures data it uses the "getpeername" API to get the IP address of the connected host. It then converts this IP address to integer value using the "ntohs API". If the integer value of the IP address matches either "16843029" or "33620245" the malware will search it for a "Financial Request Message" (Figure 6). If not, it will process the incoming data as normal, however it still attempts to log it to a file named "c:\\tmp_DMP\\TMPL_%d_%d.tmp" in the format RECV SOCK= 0x%p, BUF= 0x%p, LEN= 0x%08X, RET= %08X, IP= %s, Port=.

Upon receipt of one of these Financial Request Messages, this structure will create a log file that is named with the following format: "c:\\tmp_DMP\\TMPL_%d_%d.tmp". The format of the data logged in this log file will be as follows:

--Begin Logged Message Data--

Message(msg=%d, ct=%d, pc=%d, sd=%d, pan=%s, date=%s)

--End Logged Message Data--

Upon receipt of a Financial Request Message the malware will decode a portion of the data, which was AES decrypted from the file “info.dat” to see if portions of it match the incoming Financial Request Message (Figure 3). Although the file “info.dat” was not available for analysis, it appears the malware is ensuring the PAN numbers of the incoming message match one of the PAN numbers contained within “info.dat”.

Static analysis indicates the malware utilizes an encrypted file named “blk.dat”. This file is expected to contain a denylist of ATM transactions, which will be denied by the hook function (Figure 2). This file was not available for analysis.

When the malware receives a request from an ATM, if it contains a PAN number configured in info.dat (Figure 3) and it is not on the denylist in “blk.dat”, the malware will craft a response and send it to the ATM system (Figure 4). It appears the response to the ATM will allow the transaction to proceed and potentially allow the hackers to illegally withdraw money. If the transaction is hijacked and approved, the malware records this success in the encrypted log file “suc.dat”.

If the transaction is rejected, because it is on the denylist in “blk.dat”, this error is logged to the file “err.dat”. If the transaction does not contain a configured PAN or a transaction on the denylist, the malware will pass it on as normal to the targeted application. When the malware receives an identified Financial Request Message, it will log it to a file with the name format "c:\\tmp_DMP\\TMPL_%d_%d.tmp". The message itself will be logged into this file with the format “Message(msg=%d, ct=%d, pc=%d, sd=%d, pan=%s, date=%s)”.

The actual response back to the ATM system will be logged into a file with the filename format "c:\\tmp_DMP\\TMPL_%d_%d.tmp". The format of the data written to this file will be send socket=0x%X, ret=%d, err=%d.

Analysis indicates the Send API is hooked with a function that uses the "getpeername" IP address of the connected host. The IP address of the host is converted using “ntohs” and if it matches one of the values “16843029” or “33620245” the sent traffic will be logged in a file named "c:\\tmp_DMP\\TMPL_%d_%d.tmp". The format of the sent data logged is SEND SOCK= 0x%p, BUF= 0x%p, LEN= 0x%08X, RET= %08X, IP= %s, Port= (Figure 7). Static analysis indicates successful hooks made to the “Send” and “Recv” APIs within the target process will be logged in a file named “c:\\tmp_DMP\\TMPL_%d_%d.tmp” with the format “g_hook_flag = %d”.

Screenshots

Figure 1 - Cipher used when decoding data in "info.dat".

Figure 2 - API "Recv" hook checking for incoming Financial Request Message for a targeted PAN.

Figure 3 - The malware searching for targeted PANs.

Figure 4 - Malware crafting and sending responses to the ATM.

Figure 5 - Hook function either searching network traffic for Financial Message or logging it and sending to the "RECV" API.

Figure 6 - "RECV" Hook API function checking if the connected host is one of the two IP addresses.

Figure 7 - Logging outbound traffic to the two specific IP addresses.

39cbad3b2aac6298537a85f0463453d54ab2660c913f4f35ba98fffeb0b15655

Tags

HIDDEN-COBRAtrojan

Details

Name	switch.exe
Size	67448 bytes
Type	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	89081f2e14e9266de8c042629b764926
SHA1	730c1b9e950932736fc4b02cbdb4e4e891485ac2
SHA256	39cbad3b2aac6298537a85f0463453d54ab2660c913f4f35ba98fffeb0b15655
SHA512	bbb5aa4d8e7a011daff71774ee9c74fa4d14627de1c25e0437c879bd1cd137223d5c2fb20fd101a511a95e59d91ea884b0947229ee67e40a4a21
ssdeep	768:aQ1PWoWzXyjJsTKJUniYs1pdLn4nDT622YuYDIhscWTJqLPNofEDy9nAXmIEHbKa:aQ5WDziX+nD0LWT6FYZDgs5ULPIJEYp
Entropy	6.396614

Antivirus

Ahnlab	HackTool/Win32.Injector
Antiy	Trojan[Banker]/Win32.Alreay
ClamAV	Win.Trojan.Alreay-7189192-0
Comodo	Malware
ESET	a variant of Generik.CWSORYC trojan
Emsisoft	Gen:Variant.Ursu.634943 (B)
Ikarus	Trojan.Inject
K7	Riskware (0040eff71)
McAfee	Trojan-Banking
Microsoft Security Essentials	Trojan:Win32/LazInjector.DD!MSR
NANOAV	Trojan.Win32.Alreay.geqrko
Sophos	Troj/Banker-GYS
Symantec	Trojan Horse
TrendMicro	TROJ_NO.4FADD924
TrendMicro House Call	TROJ_NO.4FADD924
VirusBlokAda	TrojanBanker.Alreay
Zillya!	Trojan.Alreay.Win32.96

YARA Rules

No matches found.

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2018-06-13 02:17:06-04:00
Import Hash	c9febdea3218b92a46f739082f26471e

PE Sections

MD5	Name	Raw Size	Entropy
cde81f1500263860f325ee8f80c483ce	header	1024	2.497464
a8c0a36524287fef367821e833a68350	.text	38912	6.518662

MD5	Name	Raw Size	Entropy
e1c66ff8e5f0e1909e2691360c974420	.rdata	10752	4.878020
22783e6c2539d6828f3d42b030ca08e9	.data	4096	2.117927
81195ca9b22c050f79e44175e9e7150e	.rsrc	512	5.105006
36571bcb45b1ae18dfcf7edc8c5c3d4a	.reloc	3584	4.791228

Packers/Compilers/Cryptors

Microsoft Visual C++ ?.?

Description

This file is a malicious 32-bit Windows executable. It is a command-line utility. Static analysis indicates its primary purpose is to allow a user to inject a DLL into a remote process.

5cb7a352535b447609849e20aec18c84d8b58e377d9c6365eafb45cdb7ef949b

Tags

HIDDEN-COBRAtrojan

Details

Name	A2B1A45A242CEE03FAB0BEDB2E460587
Size	130560 bytes
Type	PE32 executable (DLL) (console) Intel 80386, for MS Windows
MD5	a2b1a45a242cee03fab0bedb2e460587
SHA1	e9c9ef312370d995d303e8fc60de4e4765436f58
SHA256	5cb7a352535b447609849e20aec18c84d8b58e377d9c6365eafb45cdb7ef949b
SHA512	4ced785089832287d634c77c2b5fb16efb2147b75da9014320c98d1bc0933504bfba77273576c35b97548d25acb88a0f2944cbef6a78509f945a
ssdeep	3072:j5KO2SQhF+VJbGHMjjNNyCkeZjDYJklGCx:oO2SQT+nGHADyAZjJwC
Entropy	6.431962

Antivirus

VirusBlokAda	BScope.TrojanBanker.Agent

YARA Rules

```
rule CISA_3P_10257062 : HiddenCobra FASTCASH trojan
{
  meta:
    Author = "CISA Trusted Third Party"
    Incident = "10257062"
    Date = "2020-08-11"
    Actor = "Hidden Cobra"
    Category = "Trojan"
    Family = "FASTCASH"
    Description = "Detects HiddenCobra FASTCASH samples"
    MD5_1 = "a2b1a45a242cee03fab0bedb2e460587"
    SHA256_1 = "5cb7a352535b447609849e20aec18c84d8b58e377d9c6365eafb45cdb7ef949b"
  strings:
    $sn_config_key1 = "Slsklqc^mNgq`lyznqr[q^123"
    $sn_config_key2 = "zRuaDglxjec^tDttSlsklqc^m"
    $sn_logfile1 = "C:\\intel\\_DMP_V\\spvmdl.dat"
    $sn_logfile2 = "C:\\intel\\_DMP_V\\spvmlog_%X.dat"
    $sn_logfile3 = "C:\\intel\\_DMP_V\\TMPL_%X.dat"
```

```
        $sn_logfile4 = "C:\\intel\\mvblk.dat"
        $sn_logfile5 = "C:\\intel\\_DMP_V\\spvmsuc.dat"
    condition:
        all of ($sn*)
}
```

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2018-07-03 08:11:16-04:00
Import Hash	76e8a4f811b021cf503340a0077515cc

PE Sections

MD5	Name	Raw Size	Entropy
cbe7e7fdab96c22785fa8d7c03ca6b2b	header	1024	2.429436
03d36f4d9ae3e002027c981c399ab8c6	.text	89600	6.630313
d1f983704c508544b315d577fe3563e1	.rdata	23040	5.215776
a4b79dca294053725e2b2091453d9d85	.data	8192	4.358771
d762ef71411860ae50212e14c0a5ba72	.rsrc	512	5.115767
2e4eb6056385f6f721d970cafe65bebe	.reloc	8192	4.774185

Packers/Compilers/Cryptors

Microsoft Visual C++ DLL *sign by CodeRipper

Description

The file uses a configuration file, a deny-list, and a series of log files:

--Begin files--
C:\intel\myconf.ini: Configuration file that contains account numbers (encrypted) C:\intel\myblk.dat: Deny-listed account numbers (encrypted) C:\intel_DMP_V\spvmlog_<PID>.dat: Logs general messages and errors.
Entry Format: [<YYYY-MM-DD HH:MM:SS.sss>][PID:<PID>][TID:<TID>] <Message>"]
C:\intel_DMP_V\spvmdl.dat: Logs API hooking/unhooking success and failure.
Entry Format:
Hook Success Entry: 'Windows'
Hook Error Entry: 'Linux'
UnHook Success Entry: 'Acer'
UnHook Error Entry: 'Lenovo'
C:\intel_DMP_V\TMPL<PID>.dat: Logs Send/Receive Message metadata
Entry Format:
Recv Entry: 'recv - SOCK=<socket_id>, Addr=<IP>, Port=<Port>, pBuf=<data>, size=<datasize>' Send Entry: 'send - SOCK=<socket_id>, Addr=<IP>, Port=<Port>, size=<datasize>' C:\intel_DMP_V\TMPR<PID>.tmp: Logs Received Messages
C:\intel_DMP_V\TMPS<PID>.tmp: Logs Sent Messages
C:\intel_DMP_V\TMPHSMS<PID>.tmp: Logs LocalHost ARQC sent messages C:\intel_DMP_V\TMPHSMR<PID>.tmp: Logs LocalHost ARQC received messages
C:\intel_DMP_V\spvmscap.dat: Logs modified sent messages
C:\intel_DMP_V\spvmsuc.dat: Logs modified sent messages metadata (encrypted)
--End files--

Upon attaching to a process, the sample will decrypt the encrypted config from the configuration file and read it into memory. Next, it will hook the processes send and recv winAPIs. When the “send" function is called, it will check to see if the port is 7029, if so, it will log the data and metadata in the above log files, if not it will just pass through calling send as the program normally would. When the "receive" function is called, it will check to see if the port is 7029, if so, it will wait for packets received from port 7029 and parse the following ISO8583 fields out of the incoming datagram:

--Begin fields--
MESSAGE_TYPE_INDICATOR (MTI)
PRIMARY_ACCOUNT_NUMBER (PAN)
PROCESSING_CODE
RESERVED_NATIONAL_3
--End fields--

Next, it checks the loaded configuration for the PAN. If it exists, it will continue processing, otherwise it will pass. Then it will check the denylist file for the PAN. If denylist contains 'all' or the PAN, will set the RESPONSE_CODE to 51 (Insufficient funds) in the response message. It looks for the following message types:

--Begin message types--
POS system message
ATM transaction request
ATM balance inquiry
--End message types--

Next it, constructs what appears to be an Authorization Request Cryptogram (ARQC) message:

--Begin format--
Uses the PRIMARY_ACCOUNT_NUMBER and ICC_DATA
Contains the hardcoded string: "U8BFE0AE12F9000C1480B297BE43CAC97"
Sends to localhost on port 9990
Parses the response Authorization Response Cryptogram (ARPC) message
--End format--

Finally, it constructs and sends a ISO8583 response message.

When detaching from the process, the sample unhooks the “send” and “recv” WINAPI functions, returning them to their normal state. It will then overwrite the first 0x400 bytes of the in-memory DLL from the process, effectively cleaning up any trace of the sample.

The sample frequently uses code that is taken from GitHub with a few modifications in some cases. The sample uses code that is taken from github.com/petewarden/c_hashmap to load the configuration file into memory in a hashmap, API hooking using Microsoft’s Detour library at github.com/Microsoft/Detours and the ISO8583 parsing code is taken from github.com/sabit/Oscar-ISO8583 (slightly modified to facilitate parsing of IBM037 formatted data).

The encryption that is used for all log/config files is likely an AES variant with the following keys:

--Begin keys--
zRuaDglxjec^tDtt
Slsklqc^mNgq`lyz
--End keys--

Recommendations

CISA recommends that users and administrators consider using the following best practices to strengthen the security posture of their organization's systems. Any 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 required.
- 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 header).
- 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 800-83, **"Guide to Malware Incident Prevention & Handling for Desktops and Laptops".**

Contact Information

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 instances this 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 reverse engineering. 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 the CISA at 1-844-Say-CISA or CISA Central✉.

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-related scams. Reporting forms can be found on CISA's homepage at www.cisa.gov.

Revisions

August 26, 2020: Initial Version
