

# MAR-10288834-3.v1 – North Korean Trojan: PEBBLEDASH | CISA

Published: 2020-05-12 · Archived: 2026-04-05 13:48:02 UTC

## Notification

This report is provided "as is" for informational purposes only. The Department of Homeland Security (DHS) does not provide any warranties of any kind regarding 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 risk of misuse, in accordance with applicable rules and procedures for public release. Subject to standard copyright rules, TLP:WHITE information may be distributed without restriction. 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 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 Trojan malware variants used by the North Korean government. This malware variant has been identified as PEBBLEDASH. 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 report looks at a full-featured beaconing implant. This sample uses FakeTLS for session authentication and for network encoding utilizing RC4. It has the capability to download, upload, delete, and execute files; enable Windows CLI access; create and terminate processes; and perform target system enumeration.

For a downloadable copy of IOCs, see [MAR-10288834-3.v1.stix](#).

### Submitted Files (1)

aab2868a6ebc6bdee5bd12104191db9fc1950b30bcf96eab99801624651e77b6 (D2DE01858417FA3B580B3A95857847...)

### IPs (1)

112.217.108.138

## Findings

**aab2868a6ebc6bdee5bd12104191db9fc1950b30bcf96eab99801624651e77b6**

### Tags

rootkittrojan

### Details

<b>Name</b>	D2DE01858417FA3B580B3A95857847D5
<b>Size</b>	167937 bytes
<b>Type</b>	PE32 executable (GUI) Intel 80386, for MS Windows
<b>MD5</b>	d2de01858417fa3b580b3a95857847d5

<b>SHA1</b>	2c879a1d4b6334c59ac5f11c2038d273d334befe
<b>SHA256</b>	aab2868a6ebc6bdee5bd12104191db9fc1950b30bcf96eab99801624651e77b6
<b>SHA512</b>	220c74af533f4565c4d6f0b4a4ac37c4c6e6238eba22d976a8c28889381a7d920e29077287144ec71f60e5a0b3f3780b6c688e34b8b63092
<b>ssdeep</b>	3072:LH+Sv//jDG2TJVw2URyELc1VVA9Rznhy7i+2JYI3mX2nvwjbtDKQ:qSn/jDGtUEWgE792nmX2Eb3
<b>Entropy</b>	6.131834

**Antivirus**

<b>Ahnlab</b>	Trojan/Win32.Akdoor
<b>Avira</b>	TR/Fuery.eipis
<b>BitDefender</b>	Trojan.GenericKD.5147779
<b>ESET</b>	a variant of Win32/NukeSped.G trojan
<b>Emsisoft</b>	Trojan.GenericKD.5147779 (B)
<b>Filseclab</b>	Rootkit.Agent.eki.zwum.mg
<b>Ikarus</b>	Trojan.Win32.NukeSped
<b>NANOAV</b>	Trojan.Win32.Fuery.ephjck
<b>Symantec</b>	Trojan Horse
<b>VirusBlokAda</b>	BScope.Trojan.Dynamer
<b>Zillya!</b>	Trojan.NukeSped.Win32.4

**YARA Rules**

- rule CISA\_3P\_10135536\_02 : rc4\_key\_2
  - {
  - meta:
    - Author = "CISA Trusted Third Party"
    - Incident = "10135536"
    - Date = "2018-04-19"
    - Actor = "Hidden Cobra"
    - Category = "n/a"
    - Family = "n/a"
    - Description = "n/a"
  - strings:
    - \$s1 = { c6 ?? ?? 79 c6 ?? ?? e1 c6 ?? ?? 0a c6 ?? ?? 5d c6 ?? ?? 87 c6 ?? ?? 7d c6 ?? ?? 9f c6 ?? ?? f7 c6 ?? ?? 5d c6 ?? ?? 12 c6 ?? ?? 2e c6 ?? ?? 11 c6 ?? ?? 65 c6 ?? ?? ac c6 ?? ?? e3 c6 ?? ?? 25 }
    - \$s2 = { c7 ?? ?? 79 e1 0a 5d c7 ?? ?? 87 7d 9f f7 c7 ?? ?? 5d 12 2e 11 c7 ?? ?? 65 ac e3 25 }
  - condition:
    - (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550) and any of them
  - }

**ssdeep Matches**

<b>100</b>	d620d88dfe1dbc0b407d0c3010ff18963e8bb1534f32998322f5a16746a1d0a6
------------	--

**PE Metadata**

<b>Compile Date</b>	2017-05-10 08:32:48-04:00
<b>Import Hash</b>	244a466b5f07e9bef21f34a777ede2

**PE Sections**

MD5	Name	Raw Size	Entropy
735665170a22a6b60e78ba64be8f525a	header	4096	0.685116
03861d6eb2f7ce7eb5a2c20dae40d62b	.text	135168	6.307038
bfcf9ded9905d8f7d6afdcf03737a029	.rdata	12288	5.094334
16cb2fb46f6bf6aaae5d9daf38d0f5d4	.data	12288	5.001095
14f705208660fe080429a2fc23a6c181	.rsrc	4096	0.405655

#### Packers/Compilers/Cryptors

Microsoft Visual C++ v6.0

#### Relationships

aab2868a6e...	Connected_To	112.217.108.138
---------------	--------------	-----------------

#### Description

The sample performs dynamic dynamic link library (DLL) importing and application programming interface (API) lookups using LoadLibrary and GetProcAddress on obfuscated strings in an attempt to hide its usage of network functions. The sample obfuscates strings used for API lookups using a custom XOR algorithm. A Python3 script to decrypt the obfuscated strings is given below.

```
--Begin Python3 script--
# key = 69 A7 DD 86 0A 67 78 77 A6 78 9A DA 78 68 A7 78
def decode_string(enc, key):
    dec = b''
    for i in range(len(enc)):
        # rotate key:
        # [0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f] -> [x,0,1,2,3,4,5,6,7,8,9,a,b,c,d,e]
        # where x=(key[0]^key[2])^(key[6]&key[f])
        for j in range(15, 0, -1):
            key[j] = key[j-1]
        key[0] = (key[0] ^ key[2]) ^ (key[6] + key[15])

    dec += bytes([enc[i] ^ key[15]])
    return dec
--End Python3 script--
```

The sample obfuscates its callback descriptors (IP address and ports) using a different custom XOR algorithm. A Python3 script to decrypt the obfuscated data is given below.

```
--Begin Python3 script--
# key = 5E 85 41 FD 0C 37 57 71 D5 51 5D E3 B5 55 62 20
# C1 30 96 D3 77 4C 23 13 84 8B 63 5C 48 32 2C 5B
# 94 8F 3A 26 79 E2 6B 94 45 D1 6F 51 24 8F 86 72
# C8 D3 8D C1 C0 D3 88 56 84 B3 91 E2 B2 24 64 24
def decode_callback_descriptors(enc, key):
    dec = b''
    for i in range(len(enc)):
        dec += bytes([enc[i] ^ key[(i + 0x1378 + len(enc)) % 0x40] ^ 0x59])
    return dec
--End Python3 script--
```

The sample utilizes a "FakeTLS" scheme in an attempt to obfuscate its network communications. It picks a random Uniform Resource Locator (URL) from a list (Figure 1) to use in the TLS certificate. The sample and the command and control (C2) externally appear to perform a standard TLS authentication, however, most of the fields used are filled with random data from rand().

```
--Begin C2--
112.217.108.138:443
```

--End C2--

Once the FakeTLS handshake is complete, all further packets use a FakeTLS header, followed by RC4 encrypted data.

--Begin packet structure--

17 03 01 <2 Byte data length> <RC4 encrypted data>

RC4 Key: 79 E1 0A 5D 87 7D 9F F7 5D 12 2E 11 65 AC E3 25

--End packet structure--

The sample then waits for commands from the C2.

**Screenshots**

**Figure 1** - List of certificate URLs used in the TLS certificate.

**Figure 2** - The implant contains the commands displayed in the table.

**112.217.108.138**

**Tags**

command-and-control

**Relationships**

112.217.108.138	Connected_From	aab2868a6ebc6bdee5bd12104191db9fc1950b30bcf96eab99801624651e77b6
-----------------	----------------	--

**Description**

The malware attempts to connect to the IP address.

**Relationship Summary**

aab2868a6e...	Connected_To	112.217.108.138
112.217.108.138	Connected_From	aab2868a6ebc6bdee5bd12104191db9fc1950b30bcf96eab99801624651e77b6

**Mitigation**

The following Snort rule can be used to detect the FakeTLS RC4 encrypted command packets:

//Detects the FakeTLS RC4 encrypted command packets

// that use no arguments (i.e. nextlen = 0)

alert tcp any any -> any any (msg:"Malware Detected"; pcre:" ^\x17\x03\x01\x00\x08.\x20\x59\x2c"; rev:1; sid:99999999;)

**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 [contact@mail.cisa.dhs.gov](mailto:contact@mail.cisa.dhs.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](mailto: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.us-cert.gov](http://www.us-cert.gov).

---

Source: <https://www.us-cert.gov/ncas/analysis-reports/ar20-133c>