What is Ryuk Ransomware? The Complete Breakdown

crowdstrike.com/blog/big-game-hunting-with-ryuk-another-lucrative-targeted-ransomware/

Alexander Hanel January 10, 2019

WIZARD SPIDER is a sophisticated eCrime group that has been operating the Ryuk ransomware since August 2018, targeting large organizations for a high-ransom return. This methodology, known as "big game hunting," signals a shift in operations for WIZARD SPIDER. This actor is a Russia-based criminal group known for the operation of the TrickBot banking malware that had focused primarily on wire fraud in the past.

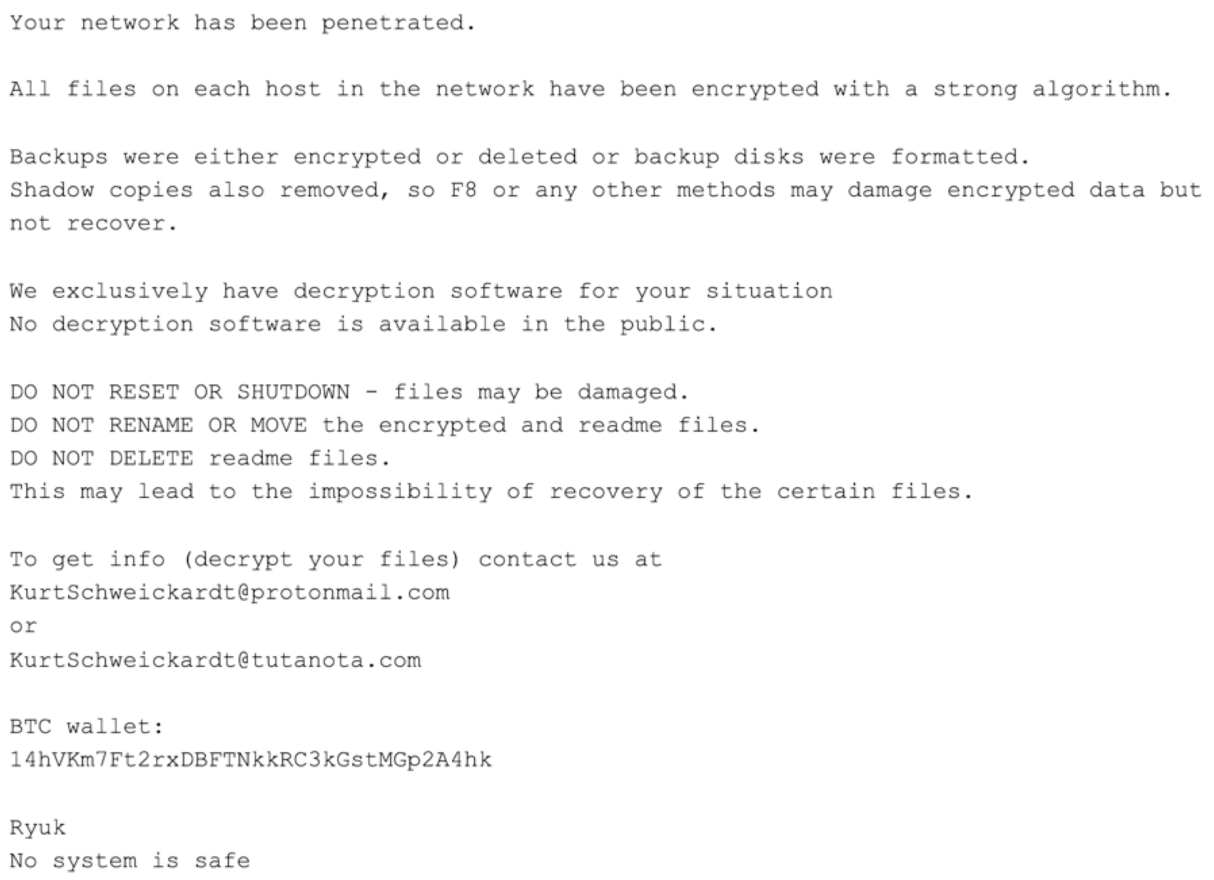
The actor name GRIM SPIDER was introduced into CrowdStrike's nomenclature in September 2018 for the group that operates the Ryuk ransomware as a distinct sub-group of the WIZARD SPIDER criminal enterprise. However, in June 2019, further evidence emerged that allowed CrowdStrike to assess with high confidence that Ryuk is in fact operated as part of the core WIZARD SPIDER actor group.

CrowdStrike Intelligence will now solely use the actor name WIZARD SPIDER in association with TrickBot and Ryuk. The GRIM SPIDER actor name has been deprecated.

Similar to *Samas* and *BitPaymer*, **Ryuk is specifically used to target enterprise environments.** Code comparison between versions of Ryuk and *Hermes* ransomware indicates that Ryuk was derived from the Hermes source code and has been under steady development since its release. Hermes is commodity ransomware that has been observed for sale on forums and used by multiple threat actors. However, Ryuk is only used by WIZARD SPIDER and, unlike Hermes, Ryuk has only been used to target enterprise environments. Since Ryuk's appearance in August, the threat actors operating it have **netted over 705.80 BTC across 52 transactions for a total current value of $3,701,893.98 USD.**

1. Ryuk Ransom Notes

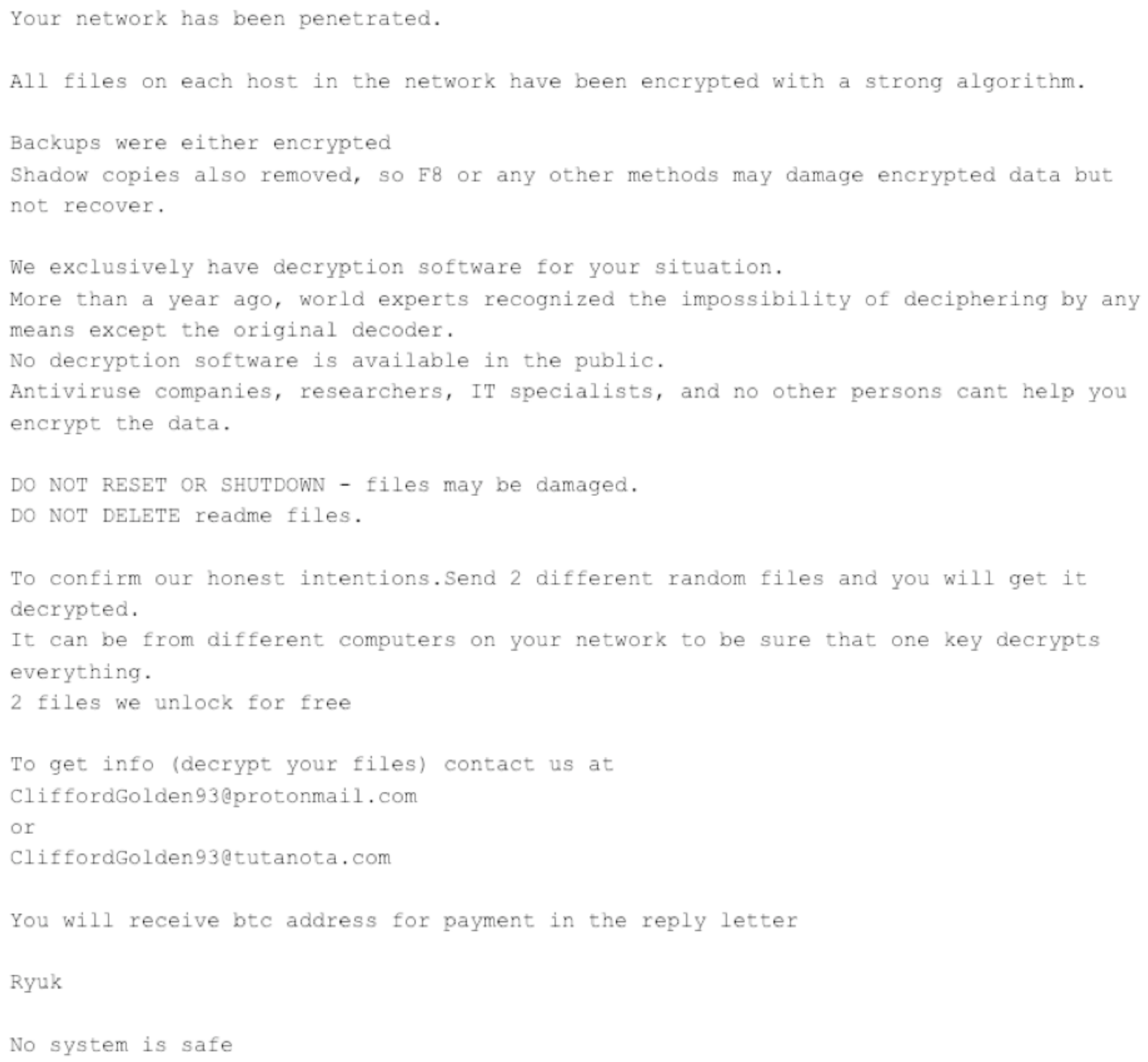
The Ryuk ransom note is written to a file named `RyukReadMe.txt`. A number of different ransom note templates have been observed. The body of the template is static with the exception of the email address and the Bitcoin (BTC) wallet address, which may change. The email addresses usually contain one address at protonmail.com and another address at tutanota.com. The email names typically are esoteric actors and directors, but *Instagram* models have also been observed. Interestingly, the ransom note in Figure 3 is remarkably similar to the BitPaymer ransom notes. As of this writing, it remains unclear if WIZARD SPIDER is copying the TTPs (tactics, techniques and procedures) and ransom notes of BitPaymer, or whether the groups may share information with each other.



```
Your network has been penetrated.

All files on each host in the network have been encrypted with a strong algorithm.

Backups were either encrypted or deleted or backup disks were formatted.
Shadow copies also removed, so F8 or any other methods may damage encrypted data but
not recover.

We exclusively have decryption software for your situation
No decryption software is available in the public.

DO NOT RESET OR SHUTDOWN - files may be damaged.
DO NOT RENAME OR MOVE the encrypted and readme files.
DO NOT DELETE readme files.
This may lead to the impossibility of recovery of the certain files.

To get info (decrypt your files) contact us at
KurtSchweickardt@protonmail.com
or
KurtSchweickardt@tutanota.com

BTC wallet:
14hVKm7Ft2rxDBFTNkkRC3kGstMGp2A4hk

Ryuk
No system is safe
```

Figure 3. Ryuk Ransom Note Bearing Strong Resemblance to BitPaymer

The ransom email used by Ryuk appears to be unique for each compiled executable. Our Using threat intelligence, our team has observed several different email addresses, but the same BTC addresses across multiple Ryuk executables. On Nov. 29, 2018, WIZARD SPIDER changed how they communicated with their victims. As seen in the previous ransom note version, WIZARD SPIDER included their BTC address and email addresses. However, recent variants of Ryuk no longer contain the BTC address — only the email addresses. The ransom note states that the victim will receive the BTC address as a reply from WIZARD SPIDER. The new ransom note can be seen below.



```
Your network has been penetrated.

All files on each host in the network have been encrypted with a strong algorithm.

Backups were either encrypted
Shadow copies also removed, so F8 or any other methods may damage encrypted data but
not recover.

We exclusively have decryption software for your situation.
More than a year ago, world experts recognized the impossibility of deciphering by any
means except the original decoder.
No decryption software is available in the public.
Antiviruse companies, researchers, IT specialists, and no other persons cant help you
encrypt the data.

DO NOT RESET OR SHUTDOWN - files may be damaged.
DO NOT DELETE readme files.

To confirm our honest intentions.Send 2 different random files and you will get it
decrypted.
It can be from different computers on your network to be sure that one key decrypts
everything.
2 files we unlock for free

To get info (decrypt your files) contact us at
CliffordGolden93@protonmail.com
or
CliffordGolden93@tutanota.com

You will receive btc address for payment in the reply letter

Ryuk

No system is safe
```

Figure 4. Ryuk Ransom With BTC Address Removed

Early Ryuk binaries with the removal of the BTC address contained a PDB path of `C:\Users\Admin\Documents\Visual Studio 2015\Projects\ConsoleApplication54new crypted try to clean\x64\Release\ConsoleApplication54.pdb` . This PDB path started appearing on Nov. 29, 2018. The removal of the BTC addresses occurred a day after the U.S. Department of Justice unsealed indictments for two individuals involved in facilitating cashouts from *Samas* Bitcoin addresses.

Ransom Payments

Based on observed transactions to known Ryuk BTC addresses, the ransom demand varies significantly. This suggests that WIZARD SPIDER (like INDRIK SPIDER with BitPaymer) calculates the ransom amount based on the size and value of the victim organization. To date, the lowest observed ransom was for 1.7 BTC and the highest was for 99 BTC. With 52 known transactions spread across 37 BTC addresses (as of this writing), WIZARD SPIDER has made 705.80 BTC, which has a current value of $3.7 million (USD). With the recent decline in BTC to USD value, it is likely GRIM SPIDER has netted more. The tables in the Appendix include a set of known Ryuk BTC addresses extracted from Ryuk binaries, which are believed to be only a subset of the Ryuk BTC addresses.

2. How Ryuk Ransomware is Distributed

CrowdStrike® has conducted multiple incident response (IR) engagements responding to Ryuk infections in which TrickBot was also identified on hosts in the victim environment. CrowdStrike Falcon Intelligence™® believes that the initial compromise is performed through TrickBot, which is typically distributed either via spam email or, through the use of the Emotet (developed and operated by MUMMY SPIDER) geo-based download function. Falcon Intelligence has been monitoring the geo-based download activity from Emotet and, during 2018, MUMMY SPIDER has been an avid supporter of WIZARD SPIDER, predominantly distributing TrickBot to Emotet victims in the U.K., the U.S., and Canada.

Some of TrickBot's modules (such as `pwgrab`) could aid in recovering the credentials needed to compromise environments — the SOCKS module in particular has been observed tunneling PowerShell Empire traffic to perform reconnaissance and lateral movement. Through CrowdStrike IR engagements, WIZARD SPIDER has been observed performing the following events on the victim's network, with the end goal of pushing out the Ryuk binary:

- An obfuscated PowerShell script is executed and connects to a remote IP address.
- A reverse shell is downloaded and executed on the compromised host.
- PowerShell anti-logging scripts are executed on the host.
- Reconnaissance of the network is conducted using standard Windows command line tools along with external uploaded tools.
- Lateral movement throughout the network is enabled using Remote Desktop Protocol (RDP).
- Service User Accounts are created.
- PowerShell Empire is downloaded and installed as a service.
- Lateral movement is continued until privileges are recovered to obtain access to a domain controller.
- PSEXEC is used to push out the Ryuk binary to individual hosts.
- Batch scripts are executed to terminate processes/services and remove backups, followed by the Ryuk binary.

3. From Hermes to Ryuk: Similarities & Differences

Hermes ransomware, the predecessor to Ryuk, was first distributed in February 2017. Only one month after its release, a decryptor was written for Hermes, followed by the release of version 2.0 in April 2017, which fixed vulnerabilities in its cryptographic implementation.

Since this release, the only way for a victim to recover files is with the private encryption key, which is obtained by paying the ransom. In late August 2017, Hermes version 2.1 was released.

Hermes was originally sold on forums for $300 USD. When purchased, the buyer received a build that supported two email addresses, a decryptor and a unique RSA key pair. If the purchaser desired more email addresses, they were required to purchase another build for an additional $50. The seller of Hermes ransomware appears to have stopped or limited advertising on forums in 2017.

Early versions of Hermes were reportedly installed via internet-accessible RDP servers protected by weak credentials. In October 2017, Hermes was deployed as a destructive distraction for a Society for Worldwide Interbank Financial Telecommunication (SWIFT) compromise at the Far Eastern International Bank (FEIB) in Taiwan. Hermes' role in the SWIFT attack is described in more detail in the *Attribution* section at the end of this blog. In March 2018, Hermes was observed targeting users in South Korea via the GreenFlash Sundown exploit kit.

In mid-August 2018, a modified version of Hermes, dubbed Ryuk, started appearing in a public malware repository. Ryuk was tailored to target enterprise environments and some of the modifications include removing anti-analysis checks. These checks include querying the Process Environment Block (PEB) to see if the field is `BeingDebugged`, or querying the PEB to see if the field `NtGlobalFlag` has been set; checking to see if the host is running VirtualBox by calling the instruction CPUID; and ensuring that the host language is not Russian, Ukrainian, or Belarusian. **From a process and file perspective, Hermes and Ryuk target files in a similar fashion.** The core differences are Ryuk's logic that handles file access, and the use of a second, embedded public RSA key.

The following are characteristics that have not changed:

- Encrypts files using RSA-2048 and AES-256
- Stores keys in the executable using the proprietary Microsoft SIMPLEBLOB format
- Encrypts mounted devices and remote hosts
- Uses a file marker of `HERMES` to mark or check if a file has been encrypted

Another notable difference between Hermes and Ryuk is how the encryption keys are created. Hermes starts the encryption initialization by first generating an RSA public and private key pair — referred to as a "victim key." An AES-256 key is generated and the victim's RSA private key is encrypted in AES-CBC mode. The attacker-controlled public RSA key is used to encrypt the AES key (previously used to encrypt the victim's RSA private key). Then, for each file encrypted, an AES key is generated, which is used to encrypt the file. Finally, the AES key for each file is encrypted with the victim's RSA public key, then stored at the end of the file.

Ryuk contains the same logic, but no longer generates the victim-specific RSA key pair. Instead, Ryuk has two public RSA keys embedded in the executable, and what was previously the victim's RSA private key is encrypted and embedded in the executable. Because Ryuk does not generate a victim-specific RSA key pair, all hosts can be decrypted with the same decryption key. This might appear to be a design flaw but is not, since Ryuk has a unique key for each executable.

If a single executable is used for a single victim environment, then there are no repercussions if the private keys are leaked because it will only decrypt the damage from a single Ryuk executable. Thus, it is highly likely that Ryuk pre-generates the RSA key pairs for each victim. This is arguably more secure, since the victim's system will never have access to the unencrypted RSA key pair parameters without paying the ransom. This approach is similar to INDRIK SPIDER's BitPaymer ransomware, which generates a victim-specific sample with a hard-coded public key.

4. Ryuk Functionality: A Technical Analysis

There are two types of Ryuk binaries: a dropper (which is not commonly observed) and the Ryuk executable payload. Recovery of Ryuk droppers are rare, due to the Ryuk executable payload deleting the dropper when executed. Upon execution, the dropper constructs an installation folder path. The folder path is created by calling `GetWindowsDirectoryW` and then inserting a null byte at the fourth character of the path. This is used to create a string that contains the drive letter path. If the host operating system is Windows XP or earlier, the string `Documents and Settings\Default User\` is appended to the drive letter path. If the host is Windows Vista or newer, the string `users\Public\` is appended to the drive letter path. For Windows XP, an example folder path would be `C:\Documents and Settings\Default User\`, and for Window Vista or higher, the path would be `C:\Users\Public.`

A random executable file name is then constructed. It is created by calling `_srand` with a seed value returned from calling `GetTickCount`, then `_rand` is continuously called until five alphabetic characters are concatenated together. The extension `.exe` is then appended. The dropper checks whether the host is 32-bit or 64-bit by calling `IsWow64Process` and writes one of two embedded payload executables corresponding to the host's architecture. The newly written executable is then run by calling `ShellExecuteW`. The Ryuk payload executable written by the dropper is the Ryuk component that contains the core logic for encrypting files on the host.

Ryuk is under constant development. In recent months, Ryuk binaries have continued to deviate further and further from the original Hermes source code, with the threat actors adding and removing functionality often. In November 2018, Falcon Intelligence identified new functionality added to Ryuk that included an anti-analysis infinite loop, a ping-like request to an IP address once the encryption process was completed, and the addition of an appended file extension for encrypted files. Of these three new features, only the file extension is still present in an executable compiled on Dec. 20, 2018.

File Encryption

Compared to other families of ransomware, Ryuk has very few safeguards to ensure stability of the host by not encrypting system files. For example, many ransomware families contain extensive lists of file extensions or folder names that should not be encrypted (whitelisted), but Ryuk only whitelists three extensions: It will not encrypt files with the extensions `exe, dll`, or `hrmlog`. The last extension appears to be a debug log filename created by the original Hermes developer. It should be noted that absent from this list is `sys` (system drivers), `ocx` (OLE control extension) and other executable file types. Encrypting these files could make the host unstable. Early versions of Ryuk included the whitelisting of `ini` and `lnk` files, but these have been removed in recent builds. The following folder names are also whitelisted and not encrypted.

- Chrome
- Mozilla
- Recycle.bin
- Windows
- Microsoft
- AhnLab

This is only a small subset of folder names that should be whitelisted in order to ensure stability on the host. While doing dynamic analysis, it was not uncommon to observe Ryuk attempting to encrypt files related to the Windows Bootloader (`C:\Boot`) or other critical files and folders. Due to the absence of proper whitelisting, an infected machine can become unstable over time and unbootable if restarted.

As mentioned in the *Hermes to Ryuk* section, Ryuk uses a combination of symmetric (AES) and asymmetric (RSA) encryption to encrypt files. Without the private key provided by WIZARD SPIDER, the files cannot be decrypted and are unrecoverable. A thread is created for the encryption of each file and each file is encrypted with its own AES key. After the file has been encrypted, a file extension of `.RYK` is appended to the file. All directories will have a ransom note of (`RyukReadMe.txt`) written to the directory.

Ryuk attempts to encrypt all mounted drives and hosts that have *Address Resolution Protocol* (ARP) entries (IP addresses) and it enumerates all mounted drives by calling `GetLogicalDrives`. For each mounted drive, Ryuk calls `GetDriveTypeW` to determine the drive's type. If the drive type is not a CD-ROM, files on the drive are encrypted. To retrieve IP addresses that have ARP entries, Ryuk calls `GetIpNetTable` . It iterates through all entries and then tries to enumerate files and folders on the remote host and encrypt the files.

Persistence

Current builds of Ryuk no longer contain persistence functionality. Previously, to remain persistent on the host, Ryuk created a registry entry under the Run key using Windows `cmd.exe` shell. The following command line was used to write to the Registry Run Key name `svchos` to `HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` with the value being the path to the Ryuk executable.

Process Injection

Ryuk does not encrypt files from within its own process memory space, but injects into a remote process. Before injecting into a remote process, Ryuk attempts to adjust its token privileges to have the `SeDebugPrivilege` . It takes no action if the adjustment of the token privileges fails. Before injecting into a remote process, Ryuk also calls `CreateToolhelp32Snapshot` to enumerate all running processes. If a process is found that is not named `csrss.exe, explorer.exe, lsaas.exe ,` or is running under `NT AUTHORITY` system account, Ryuk will inject itself into this single process. By ensuring that the process is not running under `NT AUTHORITY ,` the developers are assuming the process is not running under another account and therefore can be written to. Ryuk uses a combination of `VirtualAlloc, WriteProcessMemory` and `CreateRemoteThread` to inject itself into the remote process.

Process/Service Termination and Anti-Recovery Commands

Unlike other families of ransomware, Ryuk does not contain process/service termination and anti-recovery functionality embedded in the executable. In the past, Ryuk did contain these capabilities, but they have been removed and are contained within two batch files.

The batch file `kill.bat` contains commands for stopping services, disabling services and killing processes. The processes and services are stopped to ensure no open handles exist for files that will be encrypted. The following figure is a subset of each command.

```
 net stop avpsus /y
net stop McAfeeDLPAgentService /y
net stop mfewc /y
net stop BMR Boot Service /y
net stop NetBackup BMR MTFTP Service /y
…
sc config SQLTELEMETRY start= disabled
sc config SQLTELEMETRY$ECWDB2 start= disabled
sc config SQLWriter start= disabled
sc config SstpSvc start= disabled
…
taskkill /IM mspub.exe /F
taskkill /IM mydesktopqos.exe /F
taskkill /IM mydesktopservice.exe /F
```

Figure 1. Process/Services Termination `kill.bat` Commands

CrowdStrike has observed another batch file, named `windows.bat`, which makes file recovery more difficult on the victim’s machine. It should be noted that file names can be arbitrarily changed by the threat actors. The contents of the batch file are shown below in Figure 2.

```
 vssadmin Delete Shadows /all /quiet
vssadmin resize shadowstorage /for=c: /on=c: /maxsize=401MB
vssadmin resize shadowstorage /for=c: /on=c: /maxsize=unbounded
vssadmin resize shadowstorage /for=d: /on=d: /maxsize=401MB
vssadmin resize shadowstorage /for=d: /on=d: /maxsize=unbounded
vssadmin resize shadowstorage /for=e: /on=e: /maxsize=401MB
vssadmin resize shadowstorage /for=e: /on=e: /maxsize=unbounded
vssadmin resize shadowstorage /for=f: /on=f: /maxsize=401MB
vssadmin resize shadowstorage /for=f: /on=f: /maxsize=unbounded
vssadmin resize shadowstorage /for=g: /on=g: /maxsize=401MB
vssadmin resize shadowstorage /for=g: /on=g: /maxsize=unbounded
vssadmin resize shadowstorage /for=h: /on=h: /maxsize=401MB
vssadmin resize shadowstorage /for=h: /on=h: /maxsize=unbounded
vssadmin Delete Shadows /all /quiet
del /s /f /q c:\*.VHD c:\*.bac c:\*.bak c:\*.wbcat c:\*.bkf c:\Backup*.*
c:\backup*.* c:\*.set c:\*.win c:\*.dsk
del /s /f /q d:\*.VHD d:\*.bac d:\*.bak d:\*.wbcat d:\*.bkf d:\Backup*.*
d:\backup*.* d:\*.set d:\*.win d:\*.dsk
del /s /f /q e:\*.VHD e:\*.bac e:\*.bak e:\*.wbcat e:\*.bkf e:\Backup*.*
e:\backup*.* e:\*.set e:\*.win e:\*.dsk
del /s /f /q f:\*.VHD f:\*.bac f:\*.bak f:\*.wbcat f:\*.bkf f:\Backup*.*
f:\backup*.* f:\*.set f:\*.win f:\*.dsk
del /s /f /q g:\*.VHD g:\*.bac g:\*.bak g:\*.wbcat g:\*.bkf g:\Backup*.*
g:\backup*.* g:\*.set g:\*.win g:\*.dsk
del /s /f /q h:\*.VHD h:\*.bac h:\*.bak h:\*.wbcat h:\*.bkf h:\Backup*.*
h:\backup*.* h:\*.set h:\*.win h:\*.dsk
del %0
```

Figure 2. Anti-Recovery `window.bat` Commands

These anti-forensic recovery commands are quite interesting and appear to make use of an undocumented feature of the `vssadmin` resize command. While the first command in Figure 2 above, `vssadmin` `Delete Shadows /all /quiet`, is commonly used by ransomware, the command option `vssadmin` `resize shadowstorage` is rarely used. In situations where shadow copies were not created by `vssadmin`, but by third-party applications (such as backup software), `vssadmin` can display an error and not delete the backups. One such error states: “Snapshots were found, but they were outside of your allowed context. Try removing them with the backup application which created them.” The `vssadmin resize shadowstorage` command is a “hack” that relies on `vssadmin` to delete storage when the shadow copies are resized. It forces the shadow copies to be deleted regardless of their context. The command works by resizing the default shadow volume size from 10 percent to 401 MB (the minimum size is 300 MB). Then the shadow storage is set to unbounded, which allows it to use all available disk space. The shadow copies are then deleted by calling the command `vssadmin Delete Shadows /all /quiet` a second time.

The final set of commands deletes files based on their extension or folder locations. The command arguments are for `del` delete files in all sub-directories (`/s`) in quiet mode (`/q)` without asking the user for confirmation and to force (/f) the deletion of a file. The file extensions are for Virtual Hard Disk (`.VHD`), Avantrix Backup Plus files (`.bac`), backup copy (`.bak`), Windows Backup Catalog File (`.wbcat`), Windows Backup Utility File (`.bfk`), setting files (`.set`), Windows Backup File (`.win`), Disk Images (`.dsk`) and all folders that start with Backup. Note that since the del command does not securely delete a file (i.e., overwrite a file before deletion), some level of file recovery may be possible using forensic tools. The last command of `del %0` deletes the executing `.bat` file.

The anti-recovery commands used by Ryuk are more extensive than most ransomware families. These commands have not been observed being used by other ransomware families yet. This indicates that the threat actors have a thorough understanding of enterprise backup software.

5. Attribution

North Korea

Open-source reporting has claimed that the Hermes ransomware was developed by the North Korean group STARDUST CHOLLIMA (activities of which have been public reported as part of the “Lazarus Group”), because Hermes was executed on a host during the SWIFT compromise of FEIB in October 2017. Table 1 contains samples that are possibly attributed to the compromise. The two executables related to Hermes are `bitsran.exe` and `RSW7B37.tmp`.

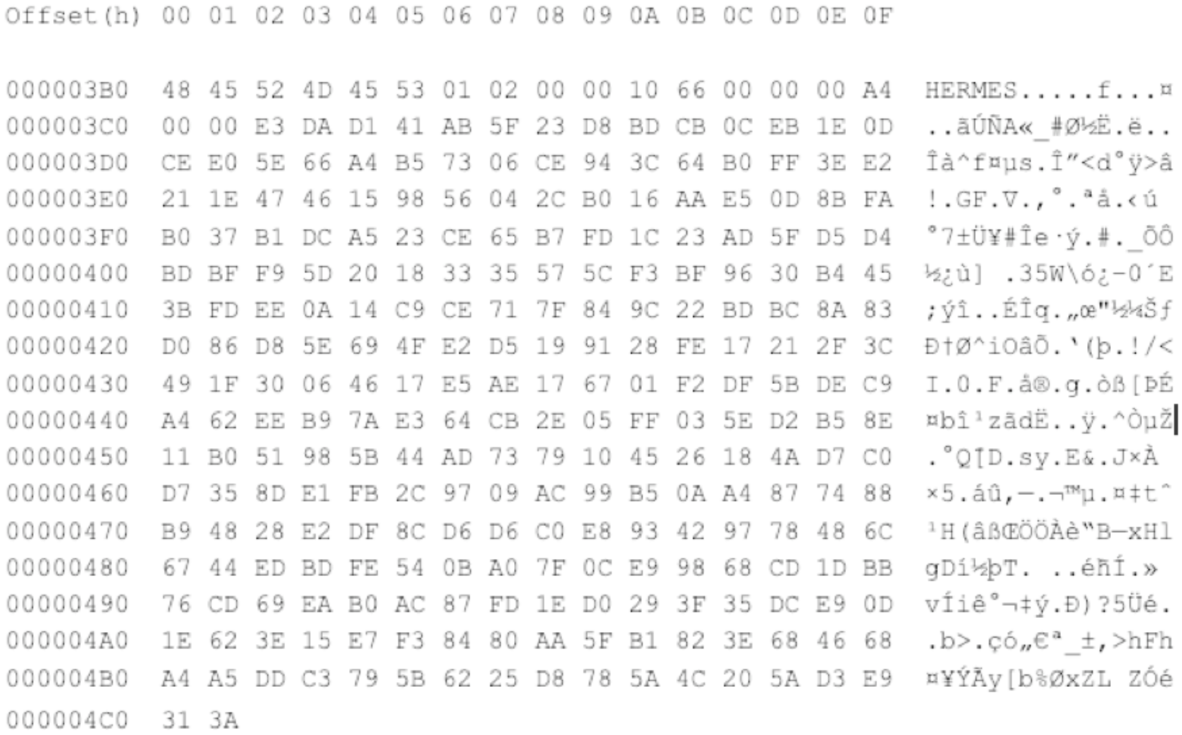
Name	Functionality	Compile Time	Compiler Version	Linker Version

msmpeng.exe	TwoPence implant	Mon Feb 20 11:09:30 2017	Visual C++ 10.0 2010 SP1 (build 40219) & Visual C++ 9.0 2008 SP1 (build 30729)	Visual C++ 10.0 2010 SP1 (build 40219)
splwow32.exe	TwoPence implant	Mon Feb 20 11:09:30 2017	Visual C++ 10.0 2010 SP1 (build 40219) & Visual C++ 9.0 2008 SP1 (build 30729)	Visual C++ 10.0 2010 SP1 (build 40219)
FileTokenBroker.dll	Loader	Thu Jan 05 01:11:33 2017	Visual C++ 10.0 2010 SP1 (build 40219) & Visual C++ 9.0 2008 SP1 (build 30729)	Visual C++ 10.0 2010 SP1 (build 40219)
bitsran.exe	Dropper/spreader	Sun Oct 1 09:37:31 2017	Visual C++ 10.0 2010 SP1 (build 40219) & Visual C++ 9.0 2008 SP1 (build 30729)	Visual C++ 10.0 2010 (build 30319)
RSW7B37.tmp	Hermes ransomware	Sun Oct 1 05:34:07 2017	Visual C++ 9.0 2008 SP1 (build 30729)	*Unknown*

Table 1. File Information for Binaries Used in the FEIB SWIFT Compromise

The first executable, `bitsran.exe`, is a dropper, and `RSW7B37.tmp` is the Hermes ransomware executable. The dropper's goal is to propagate the Hermes executable within a network by creating scheduled tasks over SMB sessions using hard-coded credentials. The Hermes executable then encrypts files on the host. It is interesting to note that the compiler and linker for Hermes is different from the other executables. All of the executables except for Hermes were compiled with Visual Studio 10, with a linker of Visual Studio 10. Hermes, in contrast, was compiled with Visual Studio 9, with an unknown linker.

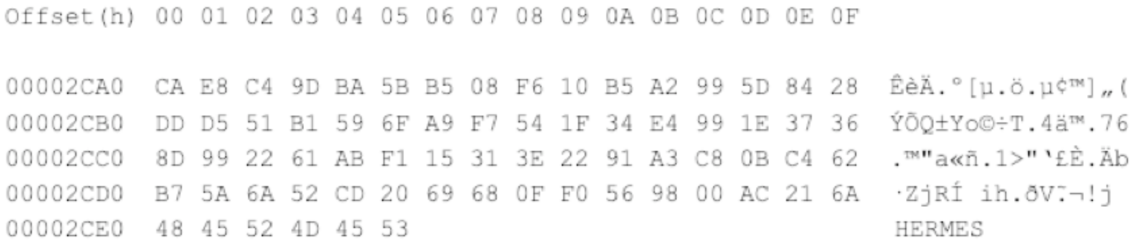
If the time stamps are correct, the two executables (`bitsran.exe` and `RSW7B37.tmp`) were compiled within four hours and three minutes of each other. Due to the short time frame of Hermes being bundled within an executable that was hard-coded with credentials of the FEIB network, Falcon Intelligence assesses that STARDUST CHOLLIMA likely had access to the Hermes source code, or a third party compiled and built a new version for them. Unlike other variants of Hermes, `RSW7B37.tmp` does not append the exported and encrypted AES key to the end of the file. Figure 5 is a file encrypted by Hermes with the exported AES key appended to the end of the file as a footer.



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

000003B0  48 45 52 4D 45 53 01 02 00 00 10 66 00 00 00 A4  HERMES.....f...¤
000003C0  00 00 E3 DA D1 41 AB 5F 23 D8 BD CB 0C EB 1E 0D  ..ãÚÑA«_#Ø½Ë.ë..
000003D0  CE E0 5E 66 A4 B5 73 06 CE 94 3C 64 B0 FF 3E E2  Îà^f¤µs.Î”<d°ÿ>â
000003E0  21 1E 47 46 15 98 56 04 2C B0 16 AA E5 0D 8B FA  !.GF.˜V.,°.ªå.‹ú
000003F0  B0 37 B1 DC A5 23 CE 65 B7 FD 1C 23 AD 5F D5 D4  °7±Ü¥#Îe·ý.#_ÕÔ
00000400  BD BF F9 5D 20 18 33 35 57 5C F3 BF 96 30 B4 45  ½¿ù] .35W\ó¿–0´E
00000410  3B FD EE 0A 14 C9 CE 71 7F 84 9C 22 BD BC 8A 83  ;ýî..ÉÎq.„œ"½¼Šƒ
00000420  D0 86 D8 5E 69 4F E2 D5 19 91 28 FE 17 21 2F 3C  Ð†Ø^iOâÕ.‘(þ.!/<
00000430  49 1F 30 06 46 17 E5 AE 17 67 01 F2 DF 5B DE C9  I.0.F.å®.g.òß[ÞÉ
00000440  A4 62 EE B9 7A E3 64 CB 2E 05 FF 03 5E D2 B5 8E  ¤bî¹zãdË..ÿ.^ÒµŽ
00000450  11 B0 51 98 5B 44 AD 73 79 10 45 26 18 4A D7 C0  .°Q˜[Dsy.E&.J×À
00000460  D7 35 8D E1 FB 2C 97 09 AC 99 B5 0A A4 87 74 88  ×5.áû,—.¬™µ.¤‡tˆ
00000470  B9 48 28 E2 DF 8C D6 D6 C0 E8 93 42 97 78 48 6C  ¹H(âßŒÖÖÀè“B—xHl
00000480  67 44 ED BD FE 54 0B A0 7F 0C E9 98 68 CD 1D BB  gDí½þT. ..é˜hÍ.»
00000490  76 CD 69 EA B0 AC 87 FD 1E D0 29 3F 35 DC E9 0D  vÍiê°¬‡ý.Ð)?5Üé.
000004A0  1E 62 3E 15 E7 F3 84 80 AA 5F B1 82 3E 68 46 68  .b>.çó„€ª_±‚>hFh
000004B0  A4 A5 DD C3 79 5B 62 25 D8 78 5A 4C 20 5A D3 E9  ¤¥ÝÃy[b%ØxZL ZÓé
000004C0  31 3A
```

Figure 5. Example Hermes Footer with Encrypted AES Key

Figure 6 is the end of a file encrypted by the Hermes variant `RSW7B37.tmp` used in the SWIFT attack. The footer only contains the marker `HERMES` but not the exported AES key.



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00002CA0  CA E8 C4 9D BA 5B B5 08 F6 10 B5 A2 99 5D 84 28  ÊèÄ.º[µ.ö.µ¢™]„(
00002CB0  DD D5 51 B1 59 6F A9 F7 54 1F 34 E4 99 1E 37 36  ÝÕQ±Yo©÷T.4ä™.76
00002CC0  8D 99 22 61 AB F1 15 31 3E 22 91 A3 C8 0B C4 62  .™"a«ñ.1>"‘£È.Äb
00002CD0  B7 5A 6A 52 CD 20 69 68 0F F0 56 98 00 AC 21 6A  ·ZjRÍ ih.ðV˜.¬!j
00002CE0  48 45 52 4D 45 53                                HERMES
```

Figure 6. Example Hermes Footer in FEIB SWIFT Attack with Encrypted AES Key Missing

Without the encrypted AES key appended to the encrypted files, even if the private key used for encryption was recovered, the files could not be decrypted. Therefore, the Hermes executable used in the FEIB SWIFT attack appears never to have been used to ransom the machine, but rather to destroy the victim’s data.

Criminal Actors Operating from Russia

Falcon Intelligence has medium-high confidence that the WIZARD SPIDER threat actors are operating out of Russia. Hermes was originally advertised on `exploit[.]in`. This Russian-speaking forum is a well-known marketplace for selling malware and related services to criminal threat actors. If Hermes was indeed related to STARDUST CHOLLIMA, it would imply that nation-state threat actors are selling their services on Russian-speaking forums, which is unlikely.

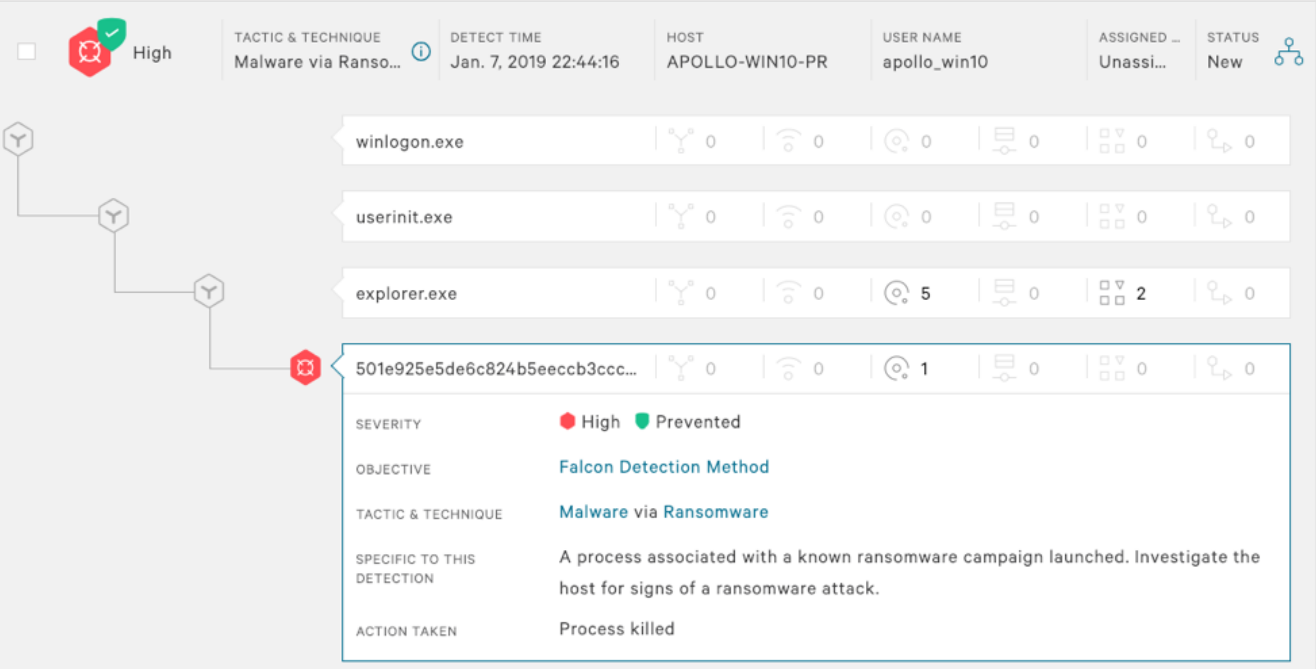
The Russian threat actor attribution theory is also supported by an early advertisement for Hermes, which stated that their “software did not work and will on work on RU, UA, BY” [sic]. This refers to functionality implemented in Hermes to check the host to ensure that it is not running on a Russian, Ukrainian, or Belarusian system. To check the host language, it queries the registry key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\Language\` and the value `InstallLanguage`. If the machine has the value 0419 (Russian), 0422 (Ukrainian) or 0423 (Belarusian), it call `ExitProcess` to stop executing. This functionality is commonly included by malware developers and sellers who are operating in Russia to reduce their risk of attracting local law enforcement’s attention and criminal prosecution.

While supporting an incident response investigation involving Ryuk, Falcon Intelligence noticed files related to the investigation being uploaded to a file-scanning website from an IP address in Moscow, Russia. The file in question was a variant of `kill.bat` that contained commands previously only observed executed by Ryuk calling `ShellExecute`. The files could have been uploaded by a victim in Russia, but the time frame between the functionality being removed from Ryuk binaries and included in `kill.bat` was very short. The most likely scenario is that threat actors were testing whether `kill.bat` would be detected by antivirus engines.

Also, during forensic investigation of a network compromised by WIZARD SPIDER, CrowdStrike Services recovered artifacts with filenames in Russian. One file was named `!!! files dlya raboty !!!.rar`, which translates to “files for work.” Based on these factors, there is considerably more evidence supporting the hypothesis that the WIZARD SPIDER threat actors are Russian speakers and not North Korean.

How CrowdStrike Can Prevent Ryuk

The Falcon platform has the ability to detect and prevent Ryuk by taking advantage of the behavioral patterns indicated by the ransomware. By turning on suspicious process blocking, Falcon ensures that Ryuk is killed in the very early stages of execution. In addition, CrowdStrike’s machine learning (ML) algorithm provides additional coverage against this malware family, as illustrated below.



High

TACTIC & TECHNIQUE: Malware via Ranso...

DETECT TIME: Jan. 7, 2019 22:44:16

HOST: APOLLO-WIN10-PR

USER NAME: apollo_win10

ASSIGNED ...: Unassi...

STATUS: New

winlogon.exe 0 0 0 0 0 0

userinit.exe 0 0 0 0 0 0

explorer.exe 0 0 5 0 2 0

501e925e5de6c824b5eeccb3ccc... 0 0 1 0 0 0

SEVERITY: High Prevented

OBJECTIVE: Falcon Detection Method

TACTIC & TECHNIQUE: Malware via Ransomware

SPECIFIC TO THIS DETECTION: A process associated with a known ransomware campaign launched. Investigate the host for signs of a ransomware attack.

ACTION TAKEN: Process killed

Appendix

Known Ryuk BTC Wallet Addresses and Payments

BTC Address	Total Received	No Received	Total Value (USD)
12vsQry1XrPjPCaH8gWzDJeYT7dhTmpcjL	55.00	3	$221,685.46
1Kx9TT76PHwk8sw7Ur6PsMWyEtaogX7wWY	182.99	10	$734,601.91
1FtQnqvjxEK5GJD9PthHM4MtdmkAeTeoRt	48.250	4	$188,974.93
14aJo5L9PTZhv8XX6qRPncbTXecb8Qohqb	25.00	2	$113,342.70
1E4fQqzCvS8wgqy5T7n1DW8JMNMaUbeFAS	0.001	1	$6.47
1GXgngwDMSJZ1Vahmf6iexKVePPXsxGS6H	30.00	3	$132,654.91
1Cyh35KqhhDewmXy63yp9ZMqBnAWe4oJRr	0.00	0	$0.00
15LsUgfnuGc1PsHJPcfLQJEnHm2FnGAgYC	0.00	0	$0.00
1CbP3cgi1Bcjuz6g2Fwvk4tVhqohqAVpDQ	13.00	2	$82,917.49
1Jq3WwsaPA7LXwRNYsfySsd8aojdmkFnW	35.00	1	$221,979.83
129L4gRSYgVJTRCgbPDtvYPabnk2QnY9sq	0.00	0	$0.00
1ET85GTps8eFbgF1MvVhFVZQeNp2a6LeGw	3.325	1	$12,661.74

1FRNVupsCyTjUvF36GxHZrvLaPtY6hgkTm	38.99	3	$246,893.95
1CW4kTqeoedinSmZiPYH7kvn4qP3mDJQVa	24.077	2	$152,727.13
13rTF3AYsf8xEdafUMT5W1E5Ab2aqPhkPi	0.00	0	$0.00
17zTcgKhF8XkWvkD4Y1N8634Qw37KwYkZT	0.00	0	$0.00
14dpmsn9rmdcS4dKD4GeqY2dYY6pwu4nVV	0.00	0	$0.00
17v2cu8RDXhAxufQ1YKiauBq6GGAZzfnFw	0.00	0	$0.00
1KUbXkjDZL6HC3Er34HwJiQUAE9H81Wcsr	10.00	1	$63,358.27
12UbZzhJrdDvdyv9NdCox1Zj1FAQ5onwx3	0.00	0	$0.00
1NMgARKzfaDExDSEsNijeT3QWbvTF7FXxS	0.00	0	$0.00
19AE1YN6Jo8ognKdJQ3xeQQL1mSZyX16op	25.00	1	$164,774.21
1L9fYHJJxeLMD2yyhh1cMFU2EWF5ihgAmJ	40.035	4	$259,478.16
18eu6KrFgzv8yTMVvKJkRM3YBAyHLonk5G	30.00	1	$198,651.35
1C8n86EEttnDjNKM9Tjm7QNVgwGBncQhDs	30.0082	2	$194,113.76
12N7W9ycLhuck9Q2wT8E6BaN6XzZ4DMLau	0.00	0	$0.00
162DVnddxsbXeVgdCy66RxEPADPETBGVBR	0.00	0	$0.00
1ChnbV4Rt7nsb5acw5YfYyvBFDj1RXcVQu	28.00	2	$175,177.98
1K6MBjz79QqfLBN7XBnwxCJb8DYUmmDWAt	1.7	2	$12,455.95
1EoyVz2tbGXWL1sLZuCnSX72eR7Ju6qohH	0.00	0	$0.00
1NQ42zc51stA4WAVkUK8uqFAjo1DbWv4Kz	0.00	0	$0.00
15FC73BdkpDMUWmxo7e7gtLRtM8gQgXyb4	0.00	0	$0.00
14hVKm7Ft2rxDBFTNkkRC3kGstMGp2A4hk	10.00	2	$64,990.62
1CN2iQbBikFK9jM34Nb3WLx5DCenQLnbXp	15.00	1	$92,934.80
1LKULheYnNtJXgQNWMo24MeLrBBCouECH7	0.00	0	$0.00
15RLWdVnY5n1n7mTvU1zjg67wt86dhYqNj	50.41	3	$326,477.83
1KURvApbe1yC7qYxkkkvtdZ7hrNjdp18sQ	0.00	0	$0.00
1NuMXQMUxCngJ7MNQ276KdaXQgGjpjFPhK	10	1	$41,034.54

Indicators

The following table contains the hashes of recently compiled Ryuk payloads:

SHA256	Build Time (UTC)
795db7bdad1befdd3ad942be79715f6b0c5083d859901b81657b590c9628790f	2018-12-27 01:10:12
501e925e5de6c824b5eeccb3ccc5111cf6e312258c0877634935df06b9d0f8b9	2018-12-21 02:33:34
fe909d18cf0fde089594689f9a69fbc6d57b69291a09f3b9df1e9b1fb724222b	2018-12-21 00:15:31

The following table contains hashes of Hermes executables that were previously analyzed:

SHA256	Build Time (UTC)
ac648d11f695cf98993fa519803fa26cd43ec32a7a8713bfa34eb618659aff77	2018-07-20 13:35:25
5e2c9ec5a108af92f177cabe23451d20e592ae54bb84265d1f972fcbd4f6a409	2018-07-23 03:47:58
78c6042067216a5d47f4a338dd951848b122bbcbcd3e61290b2f709543448d90	2018-07-1522:37:30

Additional Resources

- *For more information on how to incorporate intelligence on dangerous threat actors into your security strategy, please visit the Falcon Intelligence product page.*
- *Read Stories from the front lines of incident response and get insights that can help inform your security strategy for 2019 in the CrowdStrike Services Cyber Intrusion Casebook 2018.*
- *Test Falcon Prevent™ next-gen antivirus for yourself with a free 15-day trial today.*