Threat Research

UNC2447 SOMBRAT and FIVEHANDS Ransomware: A Sophisticated Financial Threat

April 29, 2021 | by Tyler McLellan, Justin Moore, Raymond Leong

RANSOMWARE ZERO-DAY VULNERABILITY UNC

Mandiant has observed an aggressive financially motivated group, UNC2447, exploiting one SonicWall VPN zero-day vulnerability prior to a patch being available and deploying sophisticated malware previously reported by other vendors as SOMBRAT. Mandiant has linked the use of SOMBRAT to the deployment of ransomware, which has not been previously reported publicly.

UNC2447 monetizes intrusions by extorting their victims first with FIVEHANDS ransomware followed by aggressively applying pressure through threats of media attention and offering victim data for sale on hacker forums. UNC2447 has been observed targeting organizations in Europe and North America and has consistently displayed advanced capabilities to evade detection and minimize post-intrusion forensics.

Mandiant has observed evidence of UNC2447 affiliated actors previously using RAGNARLOCKER ransomware. Based on technical and temporal observations of HELLOKITTY and FIVEHANDS deployments, Mandiant suspects that HELLOKITTY may have been used by an overall affiliate program from May 2020 through December 2020, and FIVEHANDS since approximately January 2021.

Background

In November 2020, Mandiant created UNC2447, an uncategorized group observed using the novel WARPRISM PowerShell dropper to install BEACON at two Mandiant Managed Defense clients. Mandiant Managed Defence quicky neutralized these intrusions and did not observe attempts to deploy ransomware.

In January and February 2021, Mandiant Consulting observed a novel rewrite of DEATHRANSOM—dubbed FIVEHANDS—along with SOMBRAT at multiple victims that were extorted. During one of the ransomware intrusions, the same WARPRISM and BEACON samples previously clustered under UNC2447 were observed. Mandiant was able to forensically link the use of WARPRISM, BEACON, SOMBRAT and FIVEHANDS to the same actor.

Mandiant suspects that HELLOKITTY activity in late-2020 may be related to the overall affiliate program and that usage shifted to FIVEHANDS ransomware beginning in January 2021.

- In April 2021, Mandiant observed a private FIVEHANDS TOR chat using a HELLOKITTY favicon (Figure 1).

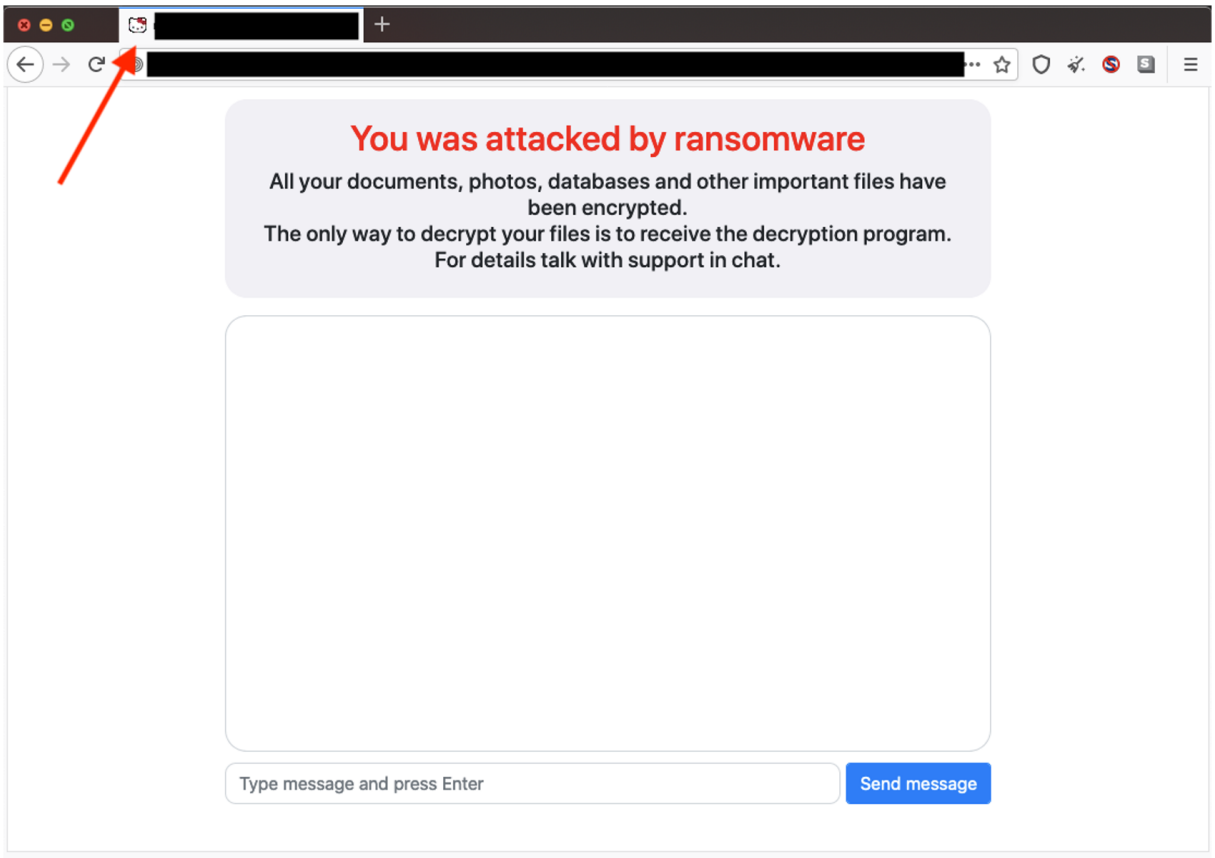

Figure 1: FIVEHANDS Hello Kitty icon

When affiliate-based ransomware is observed by Mandiant, uncategorized clusters are assigned based on the infrastructure used, and in the case of UNC2447 were based on the SOMBRAT and Cobalt Strike BEACON infrastructure used across 5 intrusions between November 2020 and February 2021. Generally, Mandiant uses caution even with novel malware such as SOMBRAT and WARPRISM and clusters each use rigorously according to all observed activity. For more information on uncategorized threats, refer to our post, "DebUNCing Attribution: How Mandiant Tracks Uncategorized Threat Actors."

SonicWall SMA 100 Series Appliance Vulnerability

CVE-2021-20016 is a critical SQL injection vulnerability that exploits unpatched SonicWall Secure Mobile Access SMA 100 series remote access products. A remote, unauthenticated attacker could submit a specially crafted query in order to exploit the vulnerability. Successful exploitation would grant an attacker the ability to access login credentials (username, password) as well as session information that could then be used to log into a vulnerable unpatched SMA 100 series appliance. This vulnerability only impacted the SMA 100 series and was patched by SonicWall in February 2021. For more information on this vulnerability, please refer to SonicWall PSIRT advisory SNWLID-2021-0001.

WARPRISM

WARPRISM is a PowerShell dropper that has been observed by Mandiant delivering SUNCRYPT, BEACON, and MIMIKATZ. WARPRISM is used to evade endpoint detection and will load its payload directly into memory. WARPRISM may be used by multiple groups.

FOXGRABBER

FOXGRABBER is a command line utility used to harvest FireFox credential files from remote systems. It contains the PDB path: C:\Users\kolobko\Source\Repos\grabff\obj\Debug\grabff.pdb. FOXGRABBER has also been observed in DARKSIDE ransomware intrusions.

BEACON Malleable Profiles

In the initial stages of an intrusion, UNC2447 uses the Cobalt Strike BEACON HTTPSSTAGER implant for persistence to communicate with command-and-control (C2) servers over HTTPS and has been observed using ‘chches_APT10’ and ‘Havex’ Malleable profiles.

UNC2447 Toolbox

During the recon and exfiltration stage of intrusions, UNC2447 has been observed using the following tools: ADFIND, BLOODHOUND, MIMIKATZ, PCHUNTER, RCLONE, ROUTERSCAN, S3BROWSER, ZAP and 7ZIP. UNC2447 may tamper with windows security settings, firewall rules, and antivirus protection.

SOMBRAT Overview

SOMBRAT was first reported by Blackberry Cylance in November 2020 as "The CostaRicto Campaign: Cyber-Espionage Outsourced" as a potential espionage-for-hire criminal group. Mandiant has now observed SOMBRAT alongside FIVEHANDS ransomware intrusions.
The SOMBRAT backdoor is packaged as a 64-bit Windows executable. It communicates with a configurable command and control (C2) server via multiple protocols, including DNS, TLS-encrypted TCP, and potentially WebSockets. Although the backdoor supports dozens of commands, most of them enable the operator to manipulate an encrypted storage file and reconfigure the implant. The backdoor's primary purpose is to download and execute plugins provided via the C2 server. In contrast to the SOMBRAT version published in November 2020, Mandiant observed additional obfuscation and armoring to evade detection, this SOMBRAT variant has been hardened to discourage analysis. Program metadata typically included by the compiler has been stripped and strings have been inlined and encoded via XOR-based routines.

The SOMBRAT Launcher

This SOMBRAT backdoor variant must be deployed alongside four additional resources that serve as launchers. They are typically installed to the hardcoded directory path `C:\ProgramData\Microsoft`.

- path: `C:\programdata\Microsoft\WwanSvc.bat` - launcher for `WwanSvc.txt`
- path: `C:\programdata\Microsoft\WwanSvc.txt` - decoder and launcher for `WwanSvc.c`
- path: `C:\programdata\Microsoft\WwanSvc.c` - decoder and launcher for `WwanSvc.b`
- path: `C:\programdata\Microsoft\WwanSvc.a` - XOR key
- path: `C:\programdata\Microsoft\WwanSvc.b` - encoded SOMBRAT backdoor
- path: `%TEMP%\<possibly unique random name>` - encrypted storage file
- path: `%TEMP%\<possibly unique random name _<integer>` - encrypted storage file
- path: `C:\ProgramData\<possibly unique random name ` - encrypted configuration file

Other variations of the filenames were observed such as ntuser and wapsvc.

SOMBRAT Technical Details

The SOMBRAT backdoor is written in modern C++ and implemented as a collection of "plugins" that interoperate with one another. There are five plugins distributed with this variant: `core`, `network`, `storage`, `taskman`, and `debug` (the `config` plugin described by Blackberry is not present). The core plugins communicate with the C2 server via messages sent over a common networking layer; each plugin supports its own set of messages, and the backdoor protocol can be extended by dynamically loaded plugins.

The `core` plugin coordinates state tracking, such as network connectivity, and dynamic plugin loading and unloading. The `network` plugin configures the networking layer used to

communicate with the C2 server, for example enabling the operator to switch between DNS and TCP protocols. The `storage` plugin exposes logical operations, such as read and write, for an encrypted file used to store plugins, resources, and arbitrary data. The `taskman` plugin enables the operator to list and kill processes on the compromised system. Finally, the `debuglog` plugin supports a single command to records debug messages.

Given that the core plugins do not enable an operator directly execute arbitrary commands or reconfigure the system, the primary function of the SOMBRAT backdoor is to load plugins provided via the C2 server. These plugins may be shellcode or DLL modules to be dynamically loaded. The C2 server may instruct the backdoor to load the plugins directly or persist them into the encrypted storage file, where they may subsequently be reloaded, such as after upgrading the backdoor.

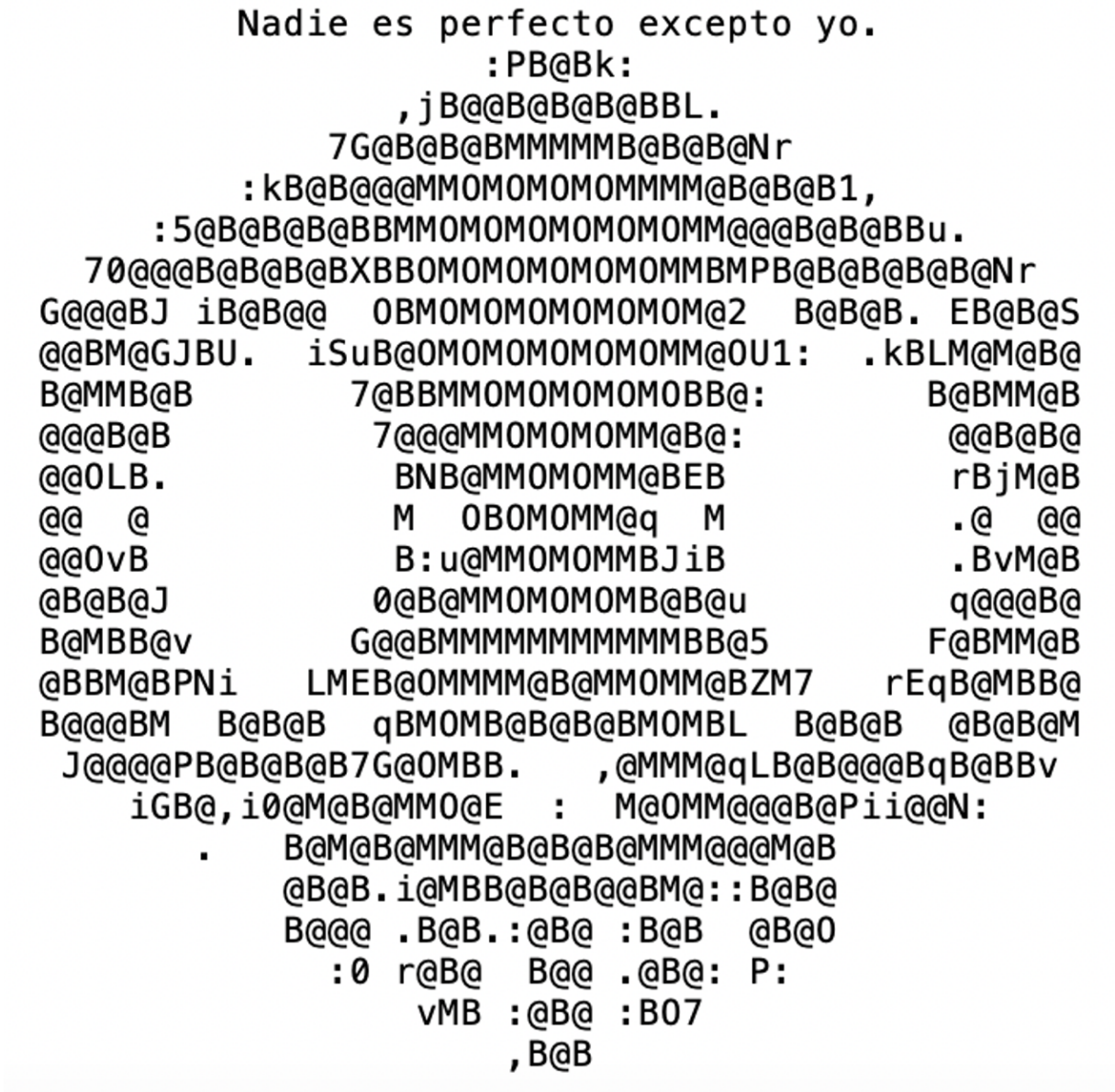

Figure 2: Malware author mark "No one is perfect except me."

SOMBRAT evades forensic analysis by patching the process memory used to record command line arguments. It replaces the initial command line with the base filename of the program executable, removing any arguments. This means that investigators that inspect a process listing via memory forensics will see the innocuous-looking command line `powershell.exe` rather than references to the uncommon filename such as `WwanSvc.c`.

SOMBRAT Network Communications

The SOMBRAT backdoor can communicate with its C2 server using both DNS and a proxy-aware, TLS-encrypted stream protocol. By default, the backdoor uses the DNS protocol; however, this can be reconfigured by the C2 server. Mandiant observed the domains feticost[.]com and celomito[.]com used for DNS C2 communications.

When the backdoor communicates via its DNS protocol, it constructs and resolves FQDNs, interpreting the DNS results to extract C2 messages. The authoritative DNS server embeds data within the IP address field of DNS A record results and within the Name Administrator field of DNS TEXT record results. By making many requests to unique subdomains of the C2 domain, the backdoor can slowly transmit information a few bytes at a time.

Ransomware Similarities

Beginning in October 2020, Mandiant observed samples of a customized version of DEATHRANSOM. This newly modified version removed the language check feature (Figure 3 shows the language check of DEATHRANSOM).

```
32  LPDWORD lpcbData; // [esp+18h] [ebp-10108h]
33  int v30; // [esp+1Ch] [ebp-10104h]
34  BYTE Data; // [esp+20h] [ebp-10100h]
35  WCHAR Buffer; // [esp+120h] [ebp-10000h]
36
37  LangID = GetUserDefaultLangID();
38  lpcbData = (LPDWORD)0x419;
39  if ( LangID == 0x419 )                          // LANG_RUSSIAN
40    goto Exit_Process;
41  if ( LangID == 0x43F )                          // LANG_KAZAK
42    goto Exit_Process;
43  v30 = 0x423;
44  if ( LangID == 0x423 )                          // LANG_BELARUSIAN
45    goto Exit_Process;
46  lpType = (LPDWORD)0x422;
47  if ( LangID == 0x422 || LangID == 0x444 )       // LANG_UKRAINIAN or LANG_TATAR
48    goto Exit_Process;
```

Figure 3: Language check from Fortinet blog

- HELLOKITTY ransomware—used to target Polish video game developer CD Projekt Red—is reportedly built from DEATHRANSOM.
    - HELLOKITTY is named after a mutex named 'HELLOKITTYMutex,' used when the malware executable is launched (see Figure 4).

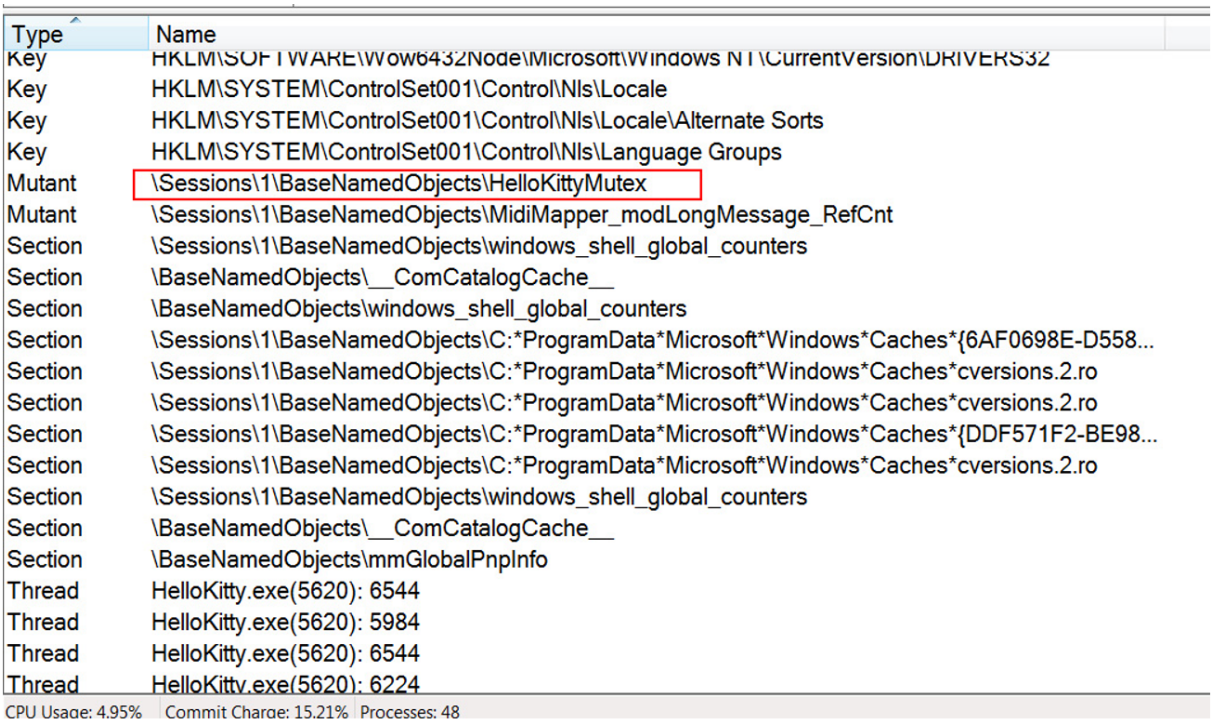

Figure 4: HELLOKITTY mutex shown in Process Explorer

- CEMIG (Companhia Energética de Minas Gerais), a Brazilian electric power company, revealed on Facebook in late December 2020 that it was a victim of HELLOKITTY cyber attack.

In January 2021, Mandiant observed a new ransomware deployed against a victim and assigned the name FIVEHANDS.

- Analysis of FIVEHANDS revealed high similarity to DEATHRANSOM, sharing several features, functions, and coding similarities. Absent in FIVEHANDS is a language check, similar to HELLOKITTY
- Both DEATHRANSOM and FIVEHANDS drops a ransom note in all non-excluded directories

Technical Comparison of FIVEHANDS, HELLOKITTY and DEATHRANSOM

DEATHRANSOM is written in C while the other two families are written in C++. DEATHRANSOM uses a distinct series of do/while loops to enumerate through network resources, logical drives, and directories. It also uses QueueUserWorkItem to implement thread pooling for its file encryption threads.

HELLOKITTY is written in C++, but reimplements a significant portion of DEATHRANSOM's functionality using similar loop operations and thread pooling via QueueUserWorkItem. The code structure to enumerate network resources, logical drives, and perform file encryption is very similar. Additionally, HELLOKITTY and DEATHRANSOM share very similar functions to check for the completion status of their encryption threads before exiting.

FIVEHANDS is written in C++ and although high level functionality is similar, the function calls and code structure to implement the majority of the functionality is written differently. Also, instead of executing threads using QueueUserWorkItem, FIVEHANDS uses IoCompletionPorts to more efficiently manage its encryption threads. FIVEHANDS also uses more functionality from the C++ standard template library (STL) than does HELLOKITTY.

Deletion of Volume Shadow Copies

DEATHRANSOM, HELLOKITTY, and FIVEHANDS use the same code to delete volume shadow copies via WMI by performing the query select * from Win32_ShadowCopy and then deleting each instance returned by its id.

Encryption Operations

Each of these three malware families utilizes a similar encryption scheme. An asymmetric public key is either hard-coded or generated. A unique symmetric key is generated for each encrypted file.

- After each file is encrypted, the asymmetric key will encrypt the symmetric key and append it to the encrypted file. Additionally, a unique four byte magic value is appended to the end of the encrypted file. The malware checks for these magic bytes to ensure it does not encrypt a previously encrypted file again.
- DEATHRANSOM and HELLOKITTY implement the file encryption operations using a very similar code structure and flow.
- FIVEHANDS implements its file encryption with a differing code structure and uses different embedded encryption libraries.
- In addition to the symmetric key, HELLOKITTY and FIVEHANDS also encrypts file metadata with the public key and appends this to the encrypted file.
- DEATHRANSOM generates an RSA key pair while HELLOKITTY and FIVEHANDS use an embedded RSA or NTRU public key.

DEATHRANSOM Encryption

- DEATHRANSOM creates an RSA-2048 public and private key pair. Using an Elliptic-curve Diffie–Hellman (ECDH) routine implemented with Curve25519, it computes a shared secret using two input values: 1) 32 random bytes from a RtlGenRandom call and 2) a hardcoded 32 byte value (attacker's public key). It also create a Curve25519 public key. The shared secret is SHA256 hashed and used as the key to Salsa20 encrypt the RSA public and private keys.
- The RSA public key is used to encrypt the individual symmetric keys that are used to encrypt each file. A Base64 encoded version of the encrypted RSA keys and the victim's Curve25519 public key is included in the ransom note, providing the threat actors the information needed to decrypt the victim's files.
- For the symmetric key, DEATHRANSOM calls RtlGenRandom to generate 32 random bytes. This is the 32 byte key used to AES encrypt each file. After the file is encrypted, the AES key is encrypted with the public RSA key and appended to the file.
- DEATHRANSOM lastly appends the four magic bytes of AB CD EF AB at the end of the encrypted file and uses this as a check to ensure that it does not encrypt an already encrypted file.
- The analyzed DEATHRANSOM sample used for comparison does not change the file extension.

HELLOKITTY Encryption

- HELLOKITTY contains an embedded RSA-2048 public key. This public key is SHA256 hashed and used as the victim ID within the ransom note. This RSA pubic key is also used to encrypt each file's symmetric key.
- For the symmetric key, HelloKitty generates a 32 byte seed value based on the CPU timestamp. A Salsa20 key is generated and encrypts a second 32 byte seed value. The encrypted result is XOR'd with the first seed, resulting in a 32 byte key used to AES encrypt each file.
- After each file is encrypted, the original file size, magic value of DE C0 AD BA, and AES key are encrypted with the public RSA key and appended to the file. HELLOKITTY and FIVEHANDS appends this additional metadata to the encrypted file, while DEATHRANSOM does not.
- Lastly it appends the four magic bytes DA DC CC AB to the end of the encrypted file.

- Depending on the version, HELLOKITTY may or may not change the file extension.
- Other samples of HELLOKITTY have used an embedded NTRU public key instead of RSA.

FIVEHANDS Encryption

- FIVEHANDS uses an embedded NTRU public key. This NTRU key is SHA512 hashed and the first 32 bytes are used as the victim ID within the ransom note. This NTRU pubic key is also used to encrypt each file's symmetric key.
- For the symmetric key, FIVEHANDS uses an embedded generation routine to produce 16 random bytes used for an AES key to encrypt each file.
- After each file is encrypted, the original file size, magic value of DE C0 AD BA, and AES key are encrypted with the public NTRU key and appended to the file.
- The four magic bytes DB DC CC AB are appended to the end of the encrypted file.
- FIVEHANDS includes additional code not found in DEATHRANSOM and HELLOKITTY to use the Windows Restart Manager to close a file currently in use so that it can be unlocked and successfully encrypted.
- The encrypted file extension is changed to .crypt extension
- FIVEHANDS's encryption flow and sequence is very different from the other two, partially because it incorporates asynchronous I/O requests and uses different embedded encryption libraries.

FIVEHANDS Encrypted Dropper

One significant change between DEATHRANSOM and FIVEHANDS is the use of a memory-only dropper, which upon execution, expects a command line switch of -key followed by the key value necessary to perform decryption of its payload. The payload is stored and encrypted with AES-128 using an IV of “85471kayecaxaubv”. The decrypted FIVEHANDS payload is immediately executed after decryption. To date, Mandiant has only observed encrypted droppers with a common imphash of 8517cf209c905e801241690648f36a97.

CLI arguments

FIVEHANDS can receive a CLI argument for a path, this limits the ransomware's file encryption activities to the specified directory. DEATHRANSOM and HELLOKITTY do not accept CLI arguments.

Locale and Mutex checks

DEATHRANSOM performs language ID and keyboard layout checks. If either of these match Russian, Kazakh, Belarusian, Ukrainian or Tatar it exits. Neither HELLOKITTY or FIVEHANDS perform language ID or keyboard checks.

HELLOKITTY performs a mutex check while the other two do not perform mutex checks.

File Exclusions

DEATHRANSOM and HELLOKITTY both exclude the same directories and files:

programdata, $recycle.bin, program files, windows, all users, appdata, read_me.txt, autoexec.bat, desktop.ini, autorun.inf, ntuser.dat, iconcache.db, bootsect.bak, boot.ini, ntuser.dat.log, or thumbs.db.

The exclusions for FIVEHANDS are more extensive and contain additional files and directories to ignore.

Additional Differences

- DEATHRANSOM makes an external HTTPS connection to download a file. Neither HELLOKITTY or FIVEHANDS initiate network connections.
- HELLOKITTY contains code to set the victims wallpaper to a ransom related image. The other samples do not have this functionality
- Different versions of DEATHRANSOM and HELLOKITTY are known to change the file extension
- Different versions of HELLOKITTY are known to check for specific processes to terminate.

Feature	FIVEHANDS	HELLOKITTY	DEATHRANSOM
Programming Language	C++	C++	C
Symmetric Encryption	AES 128	AES 256	AES 256
Asymmetric Encryption	Embedded NTRU Key	Embedded RSA or NTRU Key	Curve25519 ECDH and RSA key creation
Same directory and file name exclusions	No	Yes	Yes
Accepts CLI Arguments	Yes	No	No
Network Connections	No	No	Yes
Locale Check	No	No	Yes
Mutex Check	No	Yes	No
Bytes Appended to Encrypted Files	DB DC CC AB	DA DC CC AB	AB CD EF AB

Table 1: Ransomware feature comparison

Conclusion

Mandiant observed SOMBRAT and FIVEHANDS ransomware by the same group since January 2021. While similarities between HELLOKITTY and FIVEHANDS are notable, ransomware may be used by different groups through underground affiliate programs. Mandiant will assign an uncategorized cluster based on multiple factors including infrastructure used during intrusions and as such, not all SOMBRAT or FIVEHANDS ransomware intrusions may have been conducted by UNC2447. WARPRISM and FOXGRABBER have been used in SUNCRYPT and DARKSIDE ransomware demonstrating additional complexity and sharing between different ransomware affiliate programs.

Indicators

SOMBRAT UNC2447

- 87c78d62fd35bb25e34abb8f4caace4a
- 6382d48fae675084d30ccb69b4664cbb (31dcd09eb9fa2050aadc0e6ca05957bf unxored)

SOMBRAT Launcher

- cf1b9284d239928cce1839ea8919a7af (wwansvc.a XOR key)
- 4aa3eab3f657498f52757dc46b8d1f11 (wwansvc.c)
- 1f6495ea7606a15daa79be93070159a8 (wwansvc.bat)
- 31dcd09eb9fa2050aadc0e6ca05957bf (wwansvc.b)
- edf567bd19d09b0bab4a8d068af15572 (wwansvc.b)
- a5b26931a1519e9ceda04b4c997bb01f (wwansvc.txt)
- f0751bef4804fadfe2b993bf25791c49 (4aa3eab3f657498f52757dc46b8d1f11 unxored)
- 87c78d62fd35bb25e34abb8f4caace4a (edf567bd19d09b0bab4a8d068af15572 unxored)

SOMBRAT domains

- Celomito[.]com (unc2447)
- Feticost[.]com (unc2447)
- Cosarm[.]com
- Portalcos[.]com

FIVEHANDS

- 39ea2394a6e6c39c5d7722dc996daf05
- f568229e696c0e82abb35ec73d162d5e

FIVEHANDS Encrypted Dropper

- 6c849920155f48d4b4aafce0fc49eb5b
- 22d35005e926fe29379cb07b810a6075
- 57824214710bc0cdb22463571a72afd0
- 87c0b190e3b4ab9214e10a2d1c182153
- 1b0b9e4cddcbcb02affe9c8124855e58
- 46ecc24ef6d20f3eaf71ff37610d57d1
- 1a79b6d169aac719c9323bc3ee4a8361
- a64d79eba40229ae9aaebbd73938b985

HELLOKITTY

- 136bd70f7aa98f52861879d7dca03cf2
- 06ce6cd8bde756265f95fcf4eecadbe9
- af568e8a6060812f040f0cb0fd6f5a7b
- d96adf82f061b1a6c80699364a1e3208

DEATHRANSOM

- c50ab1df254c185506ab892dc5c8e24b

WARPRISM

- c925822c6d5175c30ba96388b07e9e16 (unc2447)
- c171bcd34151cbcd48edbce13796e0ed
- d87fcd8d2bf450b0056a151e9a116f72
- f739977004981fbe4a54bc68be18ea79
- e18b27f75c95b4d50bfcbcd00a5bd6c5
- df6e6b3e53cc713276a03cce8361ae0f
- 1cd03c0d00f7bfa7ca73f7d73677d8f8
- 8071f66d64395911a7aa0d2057b9b00d

- c12a96e9c50db5f8b0b3b5f9f3f134f0
- e39184eacba2b05aaa529547abf41d2b
- 09a05a2212bd2c0fe0e2881401fbff17
- 8226d7615532f32eca8c04ac0d41a9fd
- a01a2ba3ae9f50a5aa8a5e3492891082
- 29e53b32d5b4aae6d9a3b3c81648653c
- a809068b052bc209d0ab13f6c5c8b4e7

BEACON UNC2447

- 64.227.24[.]12 Havex Profile January 2021
- 157.230.184[.]142 chches_ APT10 Profile November 2020-January 2021
- 74c688a22822b2ab8f18eafad2271cac
- 7d6e57cbc112ebd3d3c95d3c73451a38

FOXGRABBER

- 4d3d3919dda002511e03310c49b7b47f

FireEye Detections

FireEye Network Security FireEye Email Security FireEye Detection On Demand FireEye Malware Analysis FireEye Malware File Protect	FIVEHANDS • FE_Loader_Win32_Generic_162 • FE_Ransomware_Win32_FIVEHANDS_1 • Malware.Binary.exe • Ransomware.Win.Generic.MVX SOMBRAT • FE_Backdoor_Win64_SOMBRAT_1 • Backdoor.Win.SOMBRAT • Malware.Binary.exe • Backdoor.Win.SOMBRAT.MVX • FEC_Trojan_PS1_Generic_7 • FEC_Trojan_PS1_Generic_8 • FEC_Trojan_BAT_Generic_5 HELLOKITTY • Ransomware.Win.Generic.MVX • Malware.Binary.exe • Ransomware.Win.HELLOKITTY.MVX • FE_Ransomware_Win_HELLOKITTY_1 • FE_Ransomware_Win32_HELLOKITTY_1 DEATHRANSOM • FE_Loader_Win32_Generic_92 • Ransomware.Win.Generic.MVX • Malware.Binary.exe BEACON • FE_Loader_Win32_BLUESPINE_1 • Backdoor.BEACON • Malware.Binary.exe WARPRISM • FE_Loader_PS1_WARPRISM_1 • FEC_Loader_PS1_WARPRISM_1 • Backdoor.BEACON • Trojan.Generic

	• Trojan.Win.SYSTEMBC • Backdoor.Meterpreter • Loader.PS1.WARPRISM.MVX • Malware.Binary.exe • Malware.Binary.ps1 FOXGRABBER • FE_Tool_MSIL_FOXGRABBER_1 • FE_Trojan_MSIL_Generic_109

FireEye EndPoint Security	Real-Time (IOC)
	
	• SOMBRAT (BACKDOOR)
	• SUSPICIOUS POWERSHELL READ BASE64 DATA (METHODOLOGY)
	• FIVEHANDS RANSOMWARE (FAMILY)
	• DEATHRANSOM RANSOMWARE (FAMILY)
	• HELLOKITTY RANSOMWARE (FAMILY)
	• BEACON (FAMILY)
	Malware Protection (AV/MG)
	
	• SOMBRAT
	  • Generic.mg. 87c78d62fd35bb25
	  • Generic.mg.6382d48fae675084
	  • Trojan.GenericKD.45750384
	  • Trojan.GenericKD.36367848
	  • Generic.PwShell.RefA.CB5E962A
	• FIVEHANDS
	  • Generic.mg.39ea2394a6e6c39c
	  • Generic.mg.f568229e696c0e82
	  • Generic.mg.6c849920155f48d4
	  • Generic.mg.22d35005e926fe29
	  • Generic.mg.57824214710bc0cd
	  • Generic.mg.87c0b190e3b4ab92
	  • Generic.mg.1b0b9e4cddcbcb02
	  • Generic.mg.46ecc24ef6d20f3e
	  • Generic.mg.1a79b6d169aac719
	  • Generic.mg.a64d79eba40229ae
	  • Gen:Variant.Zusy.375932
	  • Gen:Variant.Zusy.366866
	  • Trojan.GenericKD.46059492
	  • Trojan.GenericKD.46059131
	  • Trojan.GenericKD.45996121
	  • Trojan.GenericKD.45702783
	• WARPRISM
	  • Generic.mg.a01a2ba3ae9f50a5
	  • Trojan.PowerShell.Agent.IJ
	  • Trojan.Agent.EXDR
	  • Trojan.PowerShell.Ransom.E
	  • Trojan.Agent.EUKPTrojan.GenericKD.45856129
	  • Heur.BZC.PZQ.Boxter.829.B5AEB7A6
	  • Heur.BZC.PZQ.Boxter.829.B84D01A7
	  • Heur.BZC.PZQ.Boxter.829.AE76D25C
	  • Trojan.PowerShell.Ransom.F
	  • Dropped:Heur.BZC.MNT.Boxter.826.0A2B3A87

	• Heur.BZC.PZQ.Boxter.829.A15701BD
	• DEATHRANSOM
	  • Generic.mg.c50ab1df254c1855
	  • Trojan.Ransomware.GenericKD.35760206
	• HELLOKITTY
	  • Generic.mg.136bd70f7aa98f52
	  • Generic.mg.06ce6cd8bde75626
	  • Generic.mg.af568e8a6060812f
	  • Generic.mg.d96adf82f061b1a6
	  • Generic.Malware.PfVPk!12.299C21F3
	  • Gen:Variant.Ransom.HelloKitty.1
	  • Generic.Malware.PfVPk!12.606CCA24
	  • Generic.Malware.PfVPk!12.1454636C
	• BEACON
	  • Generic.mg.74c688a22822b2ab
	  • Generic.mg.7d6e57cbc112ebd3
	  • Trojan.Agent.DDSN

MITRE ATT&CK

Tactic	Description
Initial Access	• T1078 Valid Accounts
Execution	• T1047 Windows Management Instrumentation • T1053.005 Scheduled Task / Job: Scheduled Task • T1059.001 Command and Scripting Interpreter: PowerShell • T1106 Execution through API
Defense Evasion	• T1045 Software Packing • T1055 Process Injection • T1140 Deobfuscate / Decode Files or Information
Discovery	• T1012 Query Registry • T1046 Network Service Scanning • T1057 Process Discovery • T1082 System Information Discovery • T1124 System Time Discovery • T1135 Network Share Discovery
Collection	• T1560.003 Archive Collected Data: Archive via Custom Method
Impact	• T1485 Data Destruction • T1486 Data Encrypted for Impact • T1490 Inhibit System Recovery
Command and Control	• T1071.001 Application Layer Protocol: Web Protocols • T1090.002 Proxy: External Proxy • T1572 Protocol Tunneling • T1573.002 Encrypted Channel: Asymmetric Cryptography
Exfiltration	• T1041 Exfiltration over C2 Channel

Acknowledgements

Thanks to Nick Richard for technical review, Genevieve Stark and Kimberly Goody for analytical contributions, and Jon Erickson, Jonathan Lepore, and Stephen Eckels for analysis incorporated into this blog post.