28 - 30 September, 2022 / Prague, Czech Republic

(ENCRYPTION) TIME FLIES WHEN YOU'RE HAVING FUN: THE CASE OF THE EXOTIC BLACKCAT RANSOMWARE

Aleksandar Milenkoski

Cybereason, Germany

milenkoski.aleksandar@gmail.com

ABSTRACT

Time is critical for ransomware operators – the faster they encrypt the victim's files, the less likely they are to be detected in the process. Encryption can be a time-consuming process, and ransomware developers know this. That is why they get creative when programming encryption routines – the goal is to minimize the time spent on encryption and maximize the amount of encrypted file content. In this way, the greatest possible irretrievable damage is done in the shortest possible time.

BlackCat is a new and very high-profile player in the current ransomware scene. The ALPHV threat group, which is behind the ransomware, provides the malware to affiliates in exchange for a share in the ransom payments.

The way BlackCat performs encryption is highly customizable and ALPHV uses this as an advertising tool to attract affiliates. BlackCat operators can choose between six encryption modes and two encryption algorithms. Ransomware operators can further configure each encryption mode with mode-specific settings. Each encryption mode and algorithm occupies a specific position on the trade-off scale between encryption speed and completeness.

We reverse-engineered the BlackCat ransomware to provide a first look into the inner workings of the encryption modes that BlackCat implements. Our analysis provides a unique insight into the design decisions that ransomware developers make to achieve an optimal balance between encryption speed and encryption completeness.

This work also tests the encryption modes and encryption algorithms that BlackCat implements. We conducted a series of experiments to measure in numbers the trade-off between encryption speed and completeness that the different modes achieve. We examine metrics such as encryption speed, time spent on encryption, and amount of file content encrypted. For example, we observed differences in the time spent encrypting a file in the order of minutes for different ransomware configuration points. Our measurements provide a hard look at the numbers – they show how much response time is available once a carefully configured BlackCat has started encrypting files.

INTRODUCTION

The BlackCat (or ALPHV) ransomware first came to prominence in late 2021 as the first ransomware written in the Rust programming language. The ALPHV threat group, which is behind the ransomware, follows the ransomware-as-a-service (RaaS) business model and provides the malware to affiliates in exchange for a share in the ransom payments.

Since the malware's appearance in the threat landscape, the BlackCat ransomware has made headlines after targeting major organizations and businesses at a global scale. For example, in January 2022, the BlackCat ransomware targeted a major oil supplier in Germany [1], forcing the energy giant *Shell* to reroute oil supplies [2]. In addition, the ALPHV threat group claimed responsibility for the ransomware attack on the aviation services company *Swissport International* in February 2022. The attack caused flight delays and service disruptions [3].

BlackCat operators use a double extortion tactic: the operators exfiltrate data from compromised systems before encrypting the data, and if the victim refuses to pay ransom for data decryption, the malicious actors threaten to leak the exfiltrated data online or sell the data for profit. To apply additional pressure on victims, the BlackCat threat group started in June 2022 to create websites that allow the customers and employees of their victims to check if their data has been stolen in an attack [4].

BlackCat is one of the most sophisticated pieces of ransomware in the current threat landscape. The ransomware is human-operated, feature-rich, and has a large configuration space that enables attackers to fine-tune the ransomware to their needs. For example, the BlackCat ransomware attack operators (i.e. ALPHV affiliates) can configure the ransomware to self-propagate, to terminate running virtual machines, to conduct network reconnaissance, and to encrypt files according to encryption exclusion and inclusion lists [5]. Since encryption can be a time-consuming process [6], the BlackCat operators can also configure the trade-off between time spent on encryption (encryption speed) and the amount of encrypted file content (encryption completeness). The ransomware operators can choose between different encryption modes and encryption algorithms. The operators can further fine-tune each encryption mode with mode-specific settings.

Previous research focuses on the techniques and tools that BlackCat operators use in attacks [7, 8] or on the execution behaviour of the ransomware and the command-line parameters that the ransomware supports [5, 9]. This work complements previous research on the BlackCat ransomware by focusing on the encryption-related configuration space of BlackCat. This work focuses on the encryption modes and algorithms that BlackCat supports and experimentally evaluates the trade-off between encryption speed and completeness that BlackCat exhibits for different configurations.

This work looks at the BlackCat developers as software designers and discusses the design decisions the developers have made about the malware's encryption-related configuration space. This perspective is relevant to defenders in order to better understand what advantages the encryption modes and algorithms that BlackCat supports bring to the ransomware operators in terms of achieving their malicious goals. In addition, this perspective equips defenders with detailed knowledge about the encryption activities that the BlackCat ransomware may conduct on compromised systems and provides opportunities to better detect and respond to the BlackCat threat.

FILE ENCRYPTION

The BlackCat ransomware enumerates the files and folders on the filesystem of a compromised system and encrypts file content using a symmetric encryption algorithm. BlackCat worker threads encrypt file content simultaneously using a symmetric encryption key that BlackCat generates for each file under encryption.

After encrypting a file, BlackCat appends to the file data that stores file-specific information about the encryption operation that the ransomware has conducted on the file. This work refers to this data as *file encryption data*. The information that the file encryption data stores includes the name of the symmetric encryption algorithm, the symmetric encryption key, and a keyword that indicates whether BlackCat has successfully finished encrypting the file. The BlackCat ransomware operators use the file encryption data to decrypt the file contents. BlackCat encrypts file encryption data using an RSA public key and then appends the encrypted data to the file.

BlackCat ransomware samples that run on compromised systems store sample-specific configuration data in the form of a string resource in JSON (JavaScript Object Notation) format. The configuration data stores multiple settings, such as:

- Settings that control what files the ransomware excludes from encryption, such as *exclude_directory_names*, *exclude_file_names* and *exclude_file_extensions.*
- Settings that control the content of the ransom note that BlackCat stores on the filesystem and places as a wallpaper (*note_full_text* and *note_short_text*).

Figure 1 depicts a BlackCat ransomware sample configuration.


```
{
    "config_id": "",
    "public_key": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApw3tWdMaWJvNf2Mejy5H0Y6kuj+lstNpwFyismGDEYhWKPps9c68xl+84o6uLKfqPzNvLnSxlVa6DitcJGeKJEQkzN+C1e1KsfzM63jHybREB2hs+dHbqBq4dbamI
    QcTrrr4mKzuHJ7aok4mlpRx2Un1XOJaodoV7xOHO7ui5v6uK39MJ3rvitSEBvv5oI0WDlp3IFmtd6UM6r2nygY1ncAUuasalZgF1Vaz7VXOWyX2ReQHbYWWRCR1qyKMQcBtjT5POXx9B8ek1pnU4p65kGe9M79
    4Bhhh20GN24qYSa+zwXwstaNTO9luwd4xjjRQAVsDgjrjkzti27G11ICn6wIDAQAB",
    "extension": "7954i9r",
    "note_file_name": "RECOVER-${EXTENSION}-FILES.txt",
    "note_full_text": ">> Introduction\n\nImportant files on your system was ENCRYPTED and now they have have \"${EXTENSION}\" extension.\nIn order to recover your files you need to follow
    instructions below.\n\n>> Sensitive Data\n\nSensitive data on your system was DOWNLOADED and it will be PUBLISHED if you refuse to cooperate.\n\nData includes:\n- Employees personal
    [.....]
    STRONGLY ENCRYPTED, YOU CAN NOT DECRYPT IT WITHOUT CIPHER KEY.\n\n>> Recovery procedure\n\nFollow these simple steps to get in touch and recover your data:\n1) Download and install
    Tor Browser from: https: //torproject.org/\n2) Navigate to: http://sty5r4hhb5oihbq2mwevrofdiqbgesi66rvxr5sr573xgvtuvr4cs5yd.onion/?access-key=${ACCESS_KEY}",
    "note_short_text": "Important files on your system was ENCRYPTED.\nSensitive data on your system was DOWNLOADED.\nTo recover your files and prevent publishing of sensitive information
    follow instructions in \"${NOTE_FILE_NAME}\" file.",
    "default_file_mode": "Auto",
    "default_file_cipher": "Best",
    "credentials": [],
    "kill_services": [
        "mepocs",
        [...]
        "sql*"
    ],
    "kill_processes": [
        "encsvc",
        [...]
        "sql*"
    ],
    "exclude_directory_names": [
        "system volume information",
        [...]
        "windows.old"
    ],
    "exclude_file_names": [
        "desktop.ini",
        [...]
        "ntuser.dat.log"
    ],
    "exclude_file_extensions": [
        "themepack",
        "nls",
        [...]
        "msu"
    ],
    "exclude_file_path_wildcard": [],
    "enable_network_discovery": true,
    "enable_self_propagation": true,
    "enable_set_wallpaper": true,
    "enable_esxi_vm_kill": true,
    "enable_esxi_vm_snapshot_kill": true,
    "strict_include_paths": [],
    "esxi_vm_kill_exclude": []
}
```


Figure 1: A BlackCat ransomware sample configuration (trimmed for brevity).

Previous research covers the BlackCat ransomware settings in greater detail [5]. Three settings are relevant to the scope of this work:

- *default_file_cipher*: This setting specifies the symmetric encryption algorithm that BlackCat uses for encrypting file content. (The 'Symmetric encryption algorithms' section discusses the symmetric encryption algorithms that BlackCat supports in greater detail.)
- *default_file_mode*: This setting specifies the file encryption mode. The file encryption mode dictates what portions of a file under encryption the BlackCat ransomware does and does not encrypt. (The 'File encryption modes' section discusses the file encryption modes that BlackCat supports in greater detail.)
- *public_key*: This setting specifies a 256-byte-long RSA PKCS (Public-Key Cryptography Standards) #1 v1.5 public key in ASN.1 (Abstract Syntax Notation One) format and PEM (Privacy Enhanced Mail) encoding. BlackCat uses this key for encrypting the file encryption data. (The 'File encryption data' section discusses file encryption data in greater detail.)

Symmetric encryption algorithms

The BlackCat ransomware supports the individual AES (Advanced Encryption Standard) and ChaCha20 symmetric algorithms and the combinatory *Best* algorithm for encrypting file content. The ransomware operator can configure a symmetric algorithm by configuring the *default_file_cipher* setting.

When the ransomware operator configures *default_file_cipher* to *Best*, BlackCat automatically configures the fastest symmetric encryption algorithm as follows: BlackCat evaluates whether the processor on the platform where the ransomware runs supports the AES-NI (Advanced Encryption Standard New Instructions) instruction set. This instruction set improves the encryption speed and security of applications that use the AES algorithm for encryption. If AES-NI is available, BlackCat encrypts file content using AES to use the available hardware support and optimize encryption performance. If AES-NI is not available, BlackCat falls back to the ChaCha20 algorithm that is fully implemented in software.

On processors that support the *cpuid* instruction, BlackCat evaluates AES-NI support by executing the *cpuid* instruction and checking whether the 25th bit of the returned value (the *AESNI* bit) is set. Figure 2 depicts BlackCat evaluating AES-NI support (*is_aes_hw_support_available* in Figure 2) and falling back to ChaCha20 by setting a flag value to 1 (*chacha20_enabled* in Figure 2).

```
if ( is_aes_hw_support_available == -1 )
{
    _EAX = 1;
    __asm { cpuid }
    _EAX = 7;
    v66 = _ECX;
    __asm { cpuid }
    is_aes_hw_support_available = (v66 & 0x2000000) != 0;
}
if ( is_aes_hw_support_available != 1 )
    goto set_chacha;

[...]
set_chacha:
[...]
chacha20_enabled = 1;
[...]
```

Figure 2: BlackCat evaluates AES-NI support (IDA Pro pseudo-code).

File encryption modes

The BlackCat ransomware supports six file encryption modes according to which it encrypts file content: five individual modes and one combinatory mode. Table 1 provides an overview of these modes.

File encryption mode	Description
Full	BlackCat encrypts all file content.
HeadOnly [N]	BlackCat encrypts the first *N* bytes of the file under encryption, starting from the beginning of the file.
DotPattern [N,Y]	BlackCat encrypts every *N* bytes of the file under encryption, starting from the beginning of the file, with a step of *Y* bytes.
SmartPattern [N,P]	BlackCat encrypts the first *N* bytes of the file under encryption, starting from the beginning of the file. BlackCat divides the rest of the file into equal-sized blocks, such that each block is 10% of the rest of the file in size. BlackCat encrypts *P%* of the bytes of each block.
AdvancedSmartPattern [N,P,B]	BlackCat encrypts the first *N* bytes of the file under encryption, starting from the beginning of the file. BlackCat divides the rest of the file into *B* equal-sized blocks. BlackCat encrypts *P%* of the bytes of each block.
Auto	*Auto* is a combinatory file encryption mode. BlackCat encrypts the content of the file under encryption according to one of the file encryption modes *Full*, *DotPattern [N,Y]*, and *AdvancedSmartPattern [N,P,B]*. BlackCat selects and parametrizes a file encryption mode based on the filename extension and the size of the file.

Table 1: BlackCat file encryption modes.

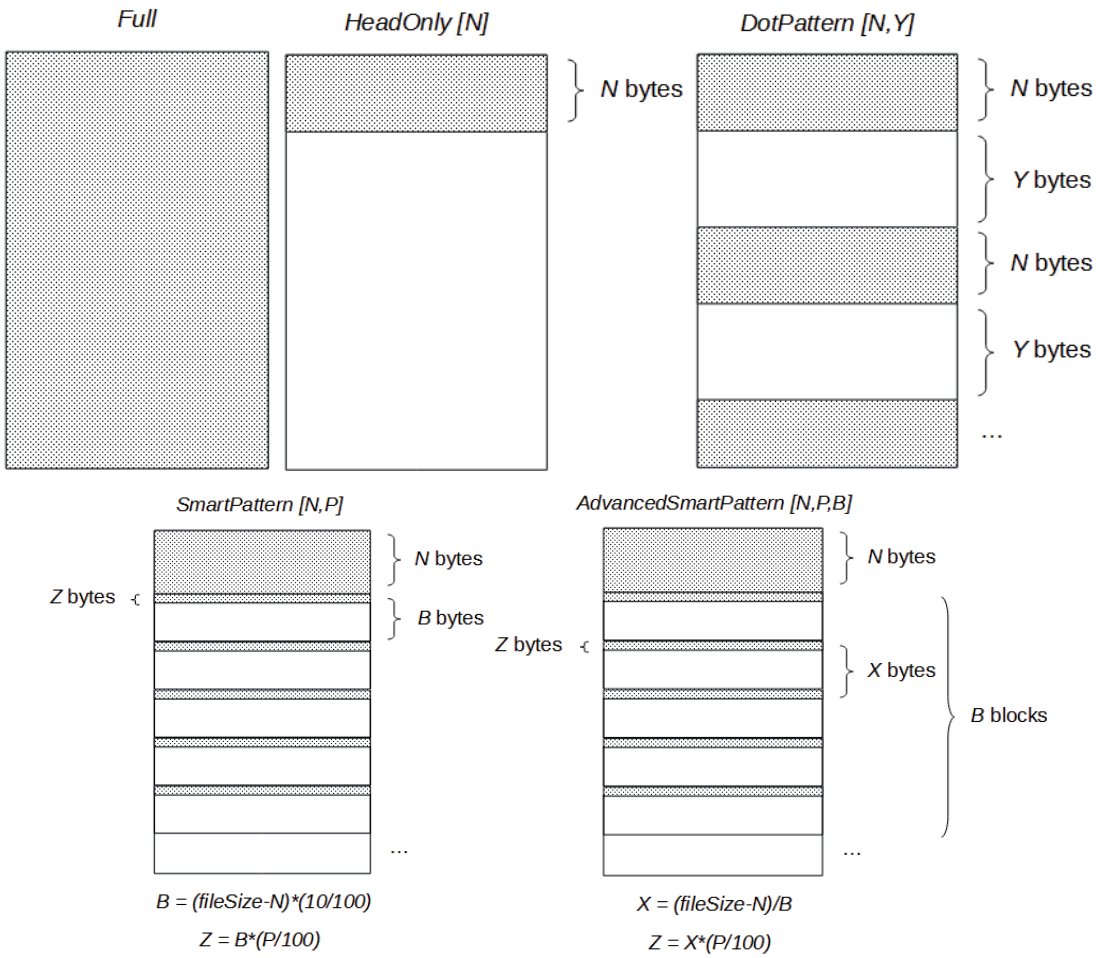

Figure 3: The individual file encryption modes of BlackCat (the dotted areas symbolize encrypted file content; fileSize is the size of the file under encryption, expressed in bytes).

The ransomware operator can enable the *Auto* file encryption mode by configuring the *default_file_mode* ransomware setting to *Auto*. When *Auto* is enabled, BlackCat first searches for the filename extension of the file under encryption in a list of filename extensions that the malware developers have hard coded in the ransomware executable (see Figure 4). If BlackCat finds the filename extension, the ransomware encrypts the content of the file according to the *Full* mode. This is to make sure that the victims cannot recover any data from the file, which could be the case if BlackCat only partially encrypts the file. The malware developers possibly consider the files of the formats with the extensions that Figure 4 depicts as critical to victims (e.g. *sql*, *pdf*) and/or some of these formats allow for the recovery of useful data from the file if the file is only partially encrypted (e.g. *txt, rtf*).

```
0:007> r $t1=eax; .for (r $t0=0; @$t0<0x10; r $t0=@$t0+1) { db poi(@$t1) Lpoi(@$t1+0x4); r $t1=@$t1+0x8 }
0061d869  73 71 6c                                        sql
0061d838  74 78 74                                        txt
0061d83b  64 6f 63                                        doc
0061d83e  72 74 66                                        rtf
0061d841  70 64 66                                        pdf
0061d844  78 6c 73                                        xls
0061d847  78 6c 73 78                                     xlsx
0061d84b  6a 70 67                                        jpg
0061d84e  6a 70 65 67                                     jpeg
0061d852  70 6e 67                                        png
0061d855  67 69 66                                        gif
0061d858  77 65 62 70                                     webp
0061d85c  74 69 66 66                                     tiff
0061d860  70 73 64                                        psd
0061d863  72 61 77                                        raw
0061d866  62 6d 70                                        bmp
```

Figure 4: Filename extensions in the BlackCat executable.

If BlackCat does not find the filename extension, it encrypts the content of the file according to one of the file encryption modes *Full*, *DotPattern [N,Y]*, and *AdvancedSmartPattern [N,P,B]*, depending on the size of the file. Figure 5 depicts pseudo-code that implements the logic of BlackCat for selecting and parametrizing a file encryption mode based on file size (see Table 1).

```
if ( fileSize <= 10 MB )
    Full

if ( fileSize > 10 MB and fileSize <= 100 MB )
    AdvancedSmartPattern[10485760, 30, 2]

if ( fileSize > 100 MB and fileSize <= 1 GB )
    AdvancedSmartPattern[25165824, 10, 5]

if ( fileSize > 1 GB and fileSize <= 10 GB )
    AdvancedSmartPattern[104857600, 5, 10]

if ( fileSize > 10 GB and fileSize <= 100 GB )
{
    step = fileSize/10 - 100 MB
    DotPattern[104857600, step]
}

if ( fileSize > 100 GB and fileSize <= 1 TB )
{
    step = fileSize/20 - 100 MB
    DotPattern[104857600, step]
}

if ( fileSize > 1 TB )
{
    step = fileSize/30 - 100 MB
    DotPattern[104857600, step]
}
```

Figure 5: The Auto file encryption mode (pseudo-code; fileSize is the size of the file under encryption, expressed in bytes).

File encryption data

The BlackCat ransomware builds a string that stores file encryption data in JSON format. After encrypting a file, BlackCat first encrypts the data using the RSA public key that resides in the ransomware's configuration and then appends the encrypted data to the file. BlackCat places the encrypted file encryption data between two instances of a four-byte value (see Figure 7). This value is the first four bytes of the sum of an integer value that is hard coded in the ransomware (*value* and *value_1* in Figure 6) and the integer values of the last six bytes of the encoded RSA public key (*pubkey_bytes* in Figure 6).

```
value = 0x19473263;
[...]
pubkey_bytes = &pubkey_ptr[(unsigned int)pubkey_length & 0xFFFFFFF8];
[...]
  pubkey_end = &pubkey_1[(_DWORD)pubkey_length];
  do
  {
    pubkey_byte = *pubkey_bytes++;
    value += pubkey_byte;
  }
  while ( pubkey_bytes != pubkey_end );
[...]
value_1 = _byteswap_ulong(value);
if ( !WriteFile(..., &value_1, 4u, ...) )
  GetLastError();
[...]
```

Figure 6: BlackCat calculates and writes in a file a four-byte value that precedes and follows the file encryption data (IDA Pro pseudo-code).

```
98 55 D6 CB 5F 50 09 E5 75 D3 19 47 BC C3 76 AB D0 A0 75 C7 60 72 39 65 AE 46 DC B2
12 8F 49 C7 04 82 AB C4 A0 02 DA 71 D8 1D CC 71 3F 3B EA 32 10 56 E7 01 1F 20 CD 36
27 53 AE 78 7D 0C F3 15 E2 9D 8F 3B EE 6A 53 52 5C B5 27 3C 1D 54 54 C0 F5 37 FC 5E
4C 55 3B 45 57 FE 51 DA 15 5A A6 06 A3 E4 92 A7 01 F7 8E C0 D9 93 36 29 BC 68 7D 29
C9 43 89 A3 6D F3 42 A1 B8 A7 2D A3 17 5D D3 95 E3 67 2B 57 6A D2 2E C7 FE 2E 01 9C
03 B2 2A 08 1E F2 A7 16 E1 C7 C1 81 DE 4B FD 9A 09 75 9A 3F 86 A2 C8 A2 14 99 ED EF
5A 79 0C 95 20 F2 3C D2 00 85 2A 83 21 44 FD 61 FC C6 D2 DD A8 E7 4F 4B 26 06 B7 40
B8 FC FC C9 6B 46 2B CD E6 F3 A5 17 5B 1C 97 D2 2F 5B 0F 41 79 D7 34 58 6B A8 B3 69
67 E5 5D 17 F0 02 F1 3F 2C 4F F9 F3 B1 86 94 0C 35 D2 2B 0B 52 E9 2B 02 49 29 B2 0D
4F 44 1A 3C 6E D8 29 CA 47 BF FB E8 37 C5 48 ED 2D EE 00 00 01 00 19 47 BC C3
```

Figure 7: Encrypted file encryption data that BlackCat has appended to a file (the four-byte value that precedes and follows the file encryption data is placed in red boxes).

Some of the fields in the file encryption data are the following:

- *mode*: This field specifies the file encryption mode according to which the ransomware has encrypted file content and the configured mode parameters.
- *cipher* and *private_key:* These fields specify the symmetric encryption algorithm and key that the ransomware has used for encrypting file content. BlackCat stores each byte of the encryption key in decimal format in the file encryption data.
- *data_size*: This field specifies the size of the file (in bytes) that the ransomware has encrypted.
- *finished*: This field indicates whether BlackCat has successfully finished encrypting the file using a Boolean keyword (*true* or *false*).

Figures 8 and 9 depict file encryption data of two files. BlackCat has encrypted one file according to the *Full* (Figure 8) and the other according to the *AdvancedSmartPattern* encryption mode (Figure 9).

```
{
    "version": 0,
    "mode": "Full",
    "cipher": "Aes",
    "private_key":
    [204,235,42,138,225,160,89,245,
    41,85,184,250,236,197,132,165],
    "data_size": 86547,
    "chunk_size": 25362816,
    "finished": true
}
```

Figure 8: File encryption data: Full file encryption mode.

```
{
    "version": 0,
    "mode": {
        "AdvancedSmartPattern": [104857600,5,10]
    },
    "cipher": "Aes",
    "private_key": [
    174,53,251,52,212,130,168,184,
    73,35,249,254,49,79,180,220],
    "data_size": 7516192768,
    "chunk_size": 25362816,
    "finished": true
}
```

Figure 9: File encryption data: AdvancedSmartPattern file encryption mode.

EVALUATION STUDY

This section presents the results of experiments that evaluate the trade-off between encryption speed and completeness that the BlackCat ransomware exhibits. The focus of the experiments is on the combinatory *Best* file encryption algorithm and the *Auto* file encryption mode as showcases of the ability of the ransomware to self-configure. The discussions in this

section are based on an analysis of the BlackCat ransomware sample with the SHA-256 hash *f837f1cd60e9941aa60f7be50a8f2aaaac380f560db8ee001408f35c1b7a97cb*. Table 1 presents the metrics that this work examines.

We executed the BlackCat ransomware on an idle machine with 2.6 GHz *Intel Core i7* processor with six cores, 32 GB of DDR 4 physical memory, and 500 GB SSD. We replicated the experiments multiple times so that the results are statistically accurate and significant. For some experiments, we configured the BlackCat sample at runtime by modifying values of internal ransomware variables using a debugger. We measured wallclock processing time, data throughput, and unencrypted content by measuring, using a debugger, the time the ransomware spent executing and the amount of data the ransomware read from and wrote to a file, between two points in the ransomware's execution flow that mark the beginning and the end of the ransomware's file processing implementation.

Metric	**Description**
Data throughput (MB/sec.)	The average amount of data (in MB) that the ransomware processes (reads, encrypts and writes) per second over the execution time of the ransomware. An observer can interpret this metric in different ways. The metric expresses the data processing workload that the ransomware is subjected to. In addition, the metric expresses the intensity of file I/O (input/output) operations that the ransomware conducts, that is, the 'noisiness' of the ransomware at file I/O level. This is relevant to the detection of the ransomware's operation at file level. Finally, the metric expresses the speed of the ransomware's data processing activity.
Wallclock processing time (sec.)	The total wallclock time (in seconds) that the ransomware spends on processing a file. Processing a file includes reading, encrypting, and writing file content.
Unencrypted content (%)	The amount of data (file content) that the ransomware has not encrypted after it has finished processing a file according to the configured file encryption mode. The unencrypted amount of data is expressed as percentage of the file size.

Table 2: Metrics.

Figures 10 and 11 depict the data throughput and the wallclock processing time that BlackCat exhibited when the ransomware processed a single file of 50 MB, 500 MB, 5 GB, and 50 GB using the encryption algorithm AES (with AES-NI support) and ChaCha20. The file encryption mode was *Full*. This study measures the impact of BlackCat automatically switching to use the hardware-supported AES over the software-implemented ChaCha20 encryption algorithm on the ransomware's performance.

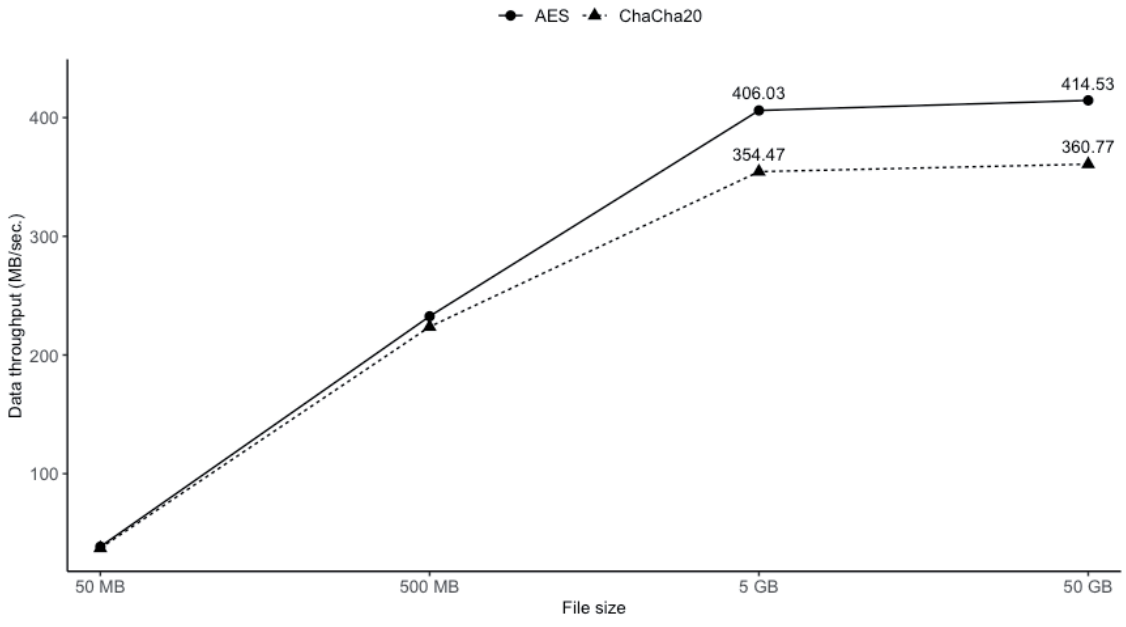

Figure 10: Data throughput [File encryption mode: Full; Encryption algorithms: AES, ChaCha20].

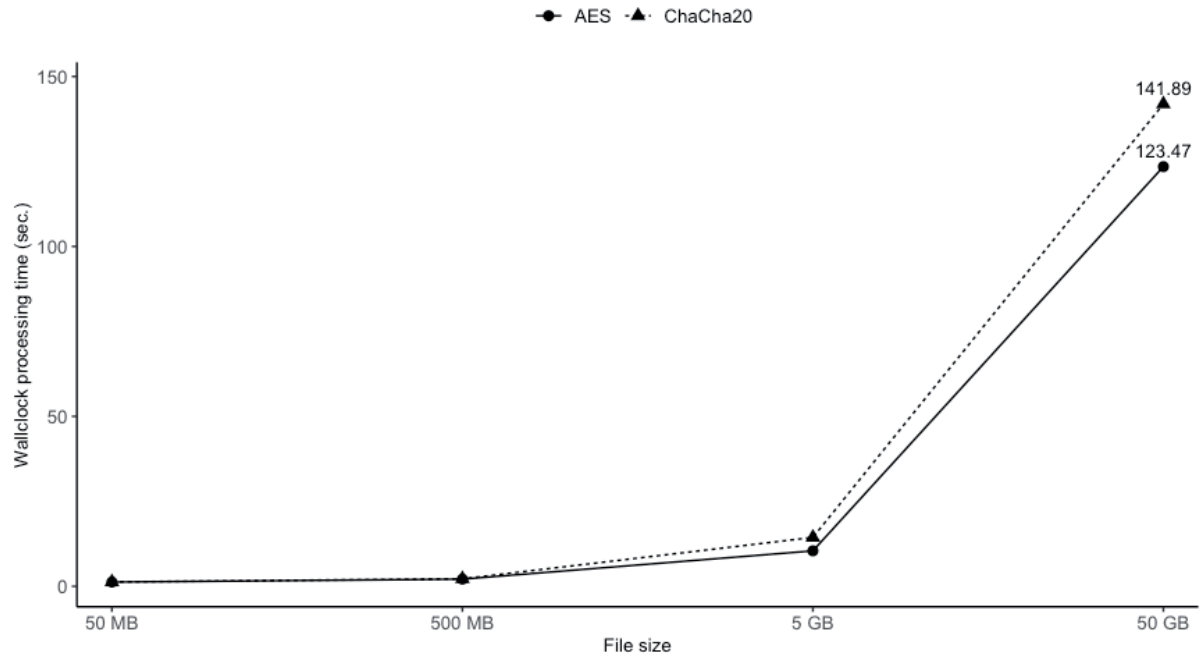

Figure 11: Wallclock processing time [File encryption mode: Full; Encryption algorithms: AES, ChaCha20].

BlackCat exhibited relative stagnation in the growth of data throughput starting at 5 GB file size. This file size was also the size at which BlackCat taking advantage of AES-NI began to be significantly noticeable, both in data throughput and wallclock processing time. The difference in data throughput was +51.56 MB/sec. and +53.76 MB/sec. when BlackCat processed a 5 GB and 50 GB big file using AES instead of ChaCha20. This difference marks a speed-up in the data processing activity of the ransomware due to faster data encryption. BlackCat exhibited the highest difference of -18.42 seconds in wallclock processing time when the ransomware processed a 50 GB file using AES instead of ChaCha20.

Figures 12, 13 and 14 depict the data throughput, the wallclock processing time, and the unencrypted content that BlackCat exhibited when the ransomware processed a single file of 50 MB, 500 MB, 5 GB, and 50 GB, and the file encryption mode was *Auto* and *Full*. The encryption algorithm was AES. This study measures the impact of BlackCat using the *Auto* over the *Full* file encryption mode on the ransomware's performance.

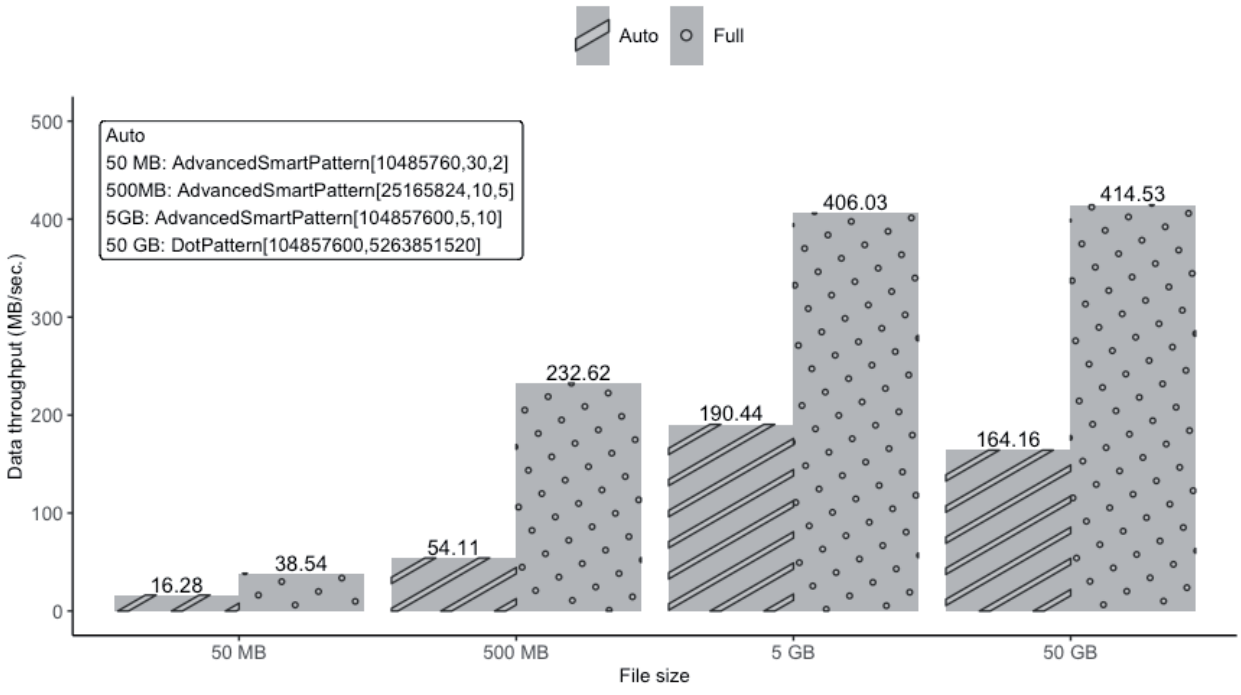

Figure 12: Data throughput [Encryption algorithm: AES; File encryption modes: Auto, Full].

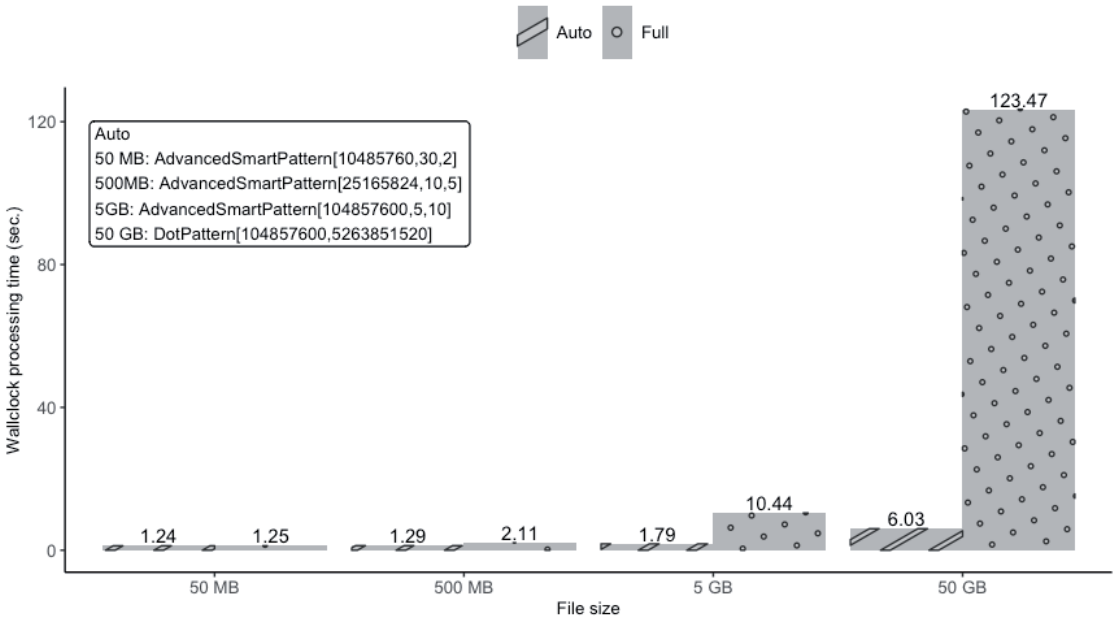

Figure 13: Wallclock processing time [Encryption algorithm: AES; File encryption modes: Auto, Full].

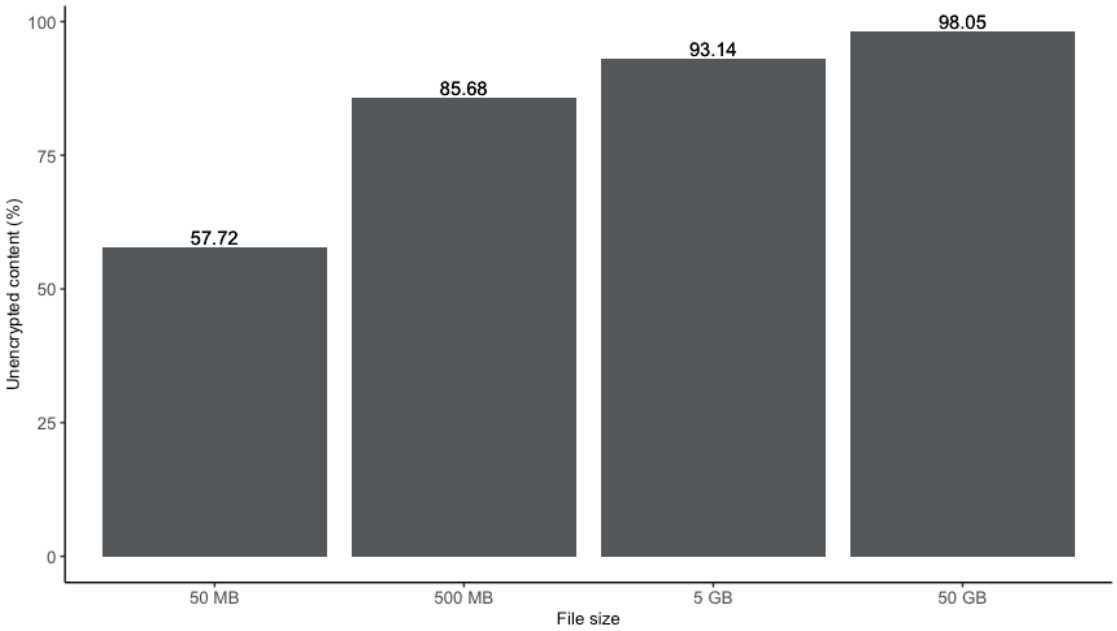

Figure 14: Unencrypted content [Encryption algorithm: AES; File encryption mode: Auto].

The use of the *Auto* file encryption mode began to result in noticeably reduced wallclock processing time at 5 GB file size (8.65 seconds), achieving the maximum reduction in wallclock processing time of 1.95 minutes at 50 GB file size. The amount of unencrypted content was relatively high at all file sizes, such that 57.72% and 85.68% unencrypted content translated to only 0.01 seconds and 0.82 seconds reduction in wallclock processing time at 50 MB and 500 MB file size, respectively. The use of the *Auto* file encryption mode significantly reduced the workload that BlackCat was subjected to. This resulted in a reduced intensity of file I/O operations, making the ransomware less 'noisy' at file I/O level.

The significant reduction in wallclock processing time and the high unencrypted content that BlackCat exhibited when it used the *Auto* file encryption mode was present when BlackCat processed multiple files as well. Figure 15 contrasts the total wallclock processing time and unencrypted content that BlackCat exhibited when the ransomware processed a set of 51 files and the file encryption mode was *Auto* and *Full*. The files had random file extensions (i.e. none of the file extensions depicted in Figure 4) and were of diverse sizes. This resulted in the *Auto* combinatory mode configuring the distribution of individual file encryption modes that Figure 15 depicts. The median file size was 298.24 MB and the total file size was 33.5 GB. The encryption algorithm was AES. BlackCat exhibited a reduction in wallclock processing time of 70.99 seconds and 88.499% unencrypted content when the ransomware used the *Auto* mode.

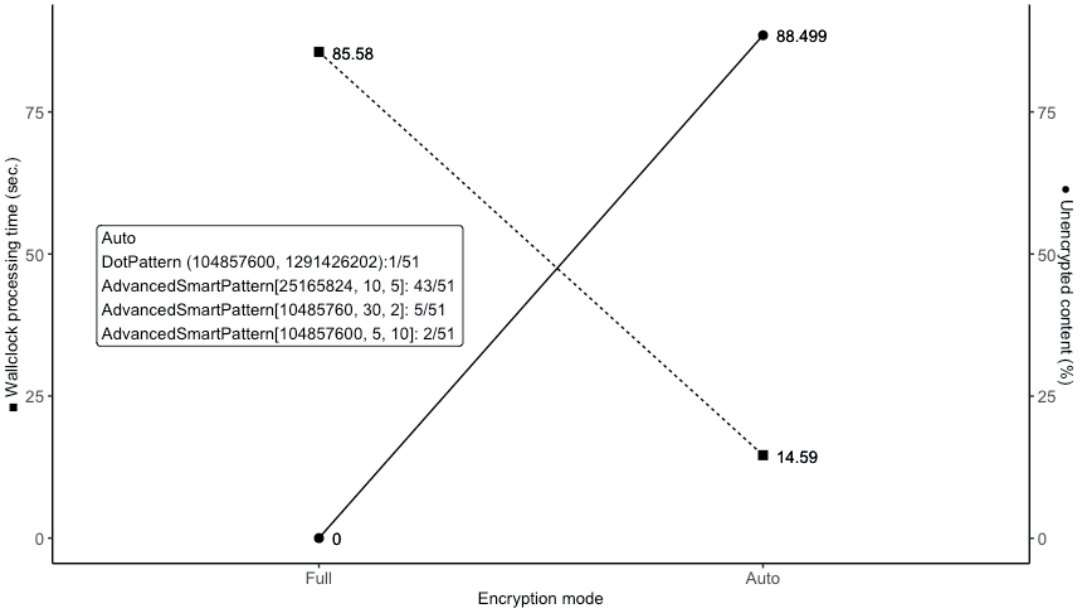

Figure 15: Wallclock processing time and unencrypted content [Encryption algorithm: AES; File encryption modes: Auto, Full].

Figure 16 contrasts the wallclock processing time and unencrypted content that BlackCat exhibited when the ransomware processed a single file of 5 GB file size and the file encryption mode was *Auto*, *AdvancedSmartPattern [104857600, 50, 10]*, and *AdvancedSmartPattern [104857600, 50, 200]*. The encryption algorithm was AES. This study measures the impact of BlackCat using the configuration of *AdvancedSmartPattern* when *Auto* is enabled over other *AdvancedSmartPattern* configurations on the ransomware's performance. The study also examines the parameter sensitivity of *AdvancedSmartPattern*.

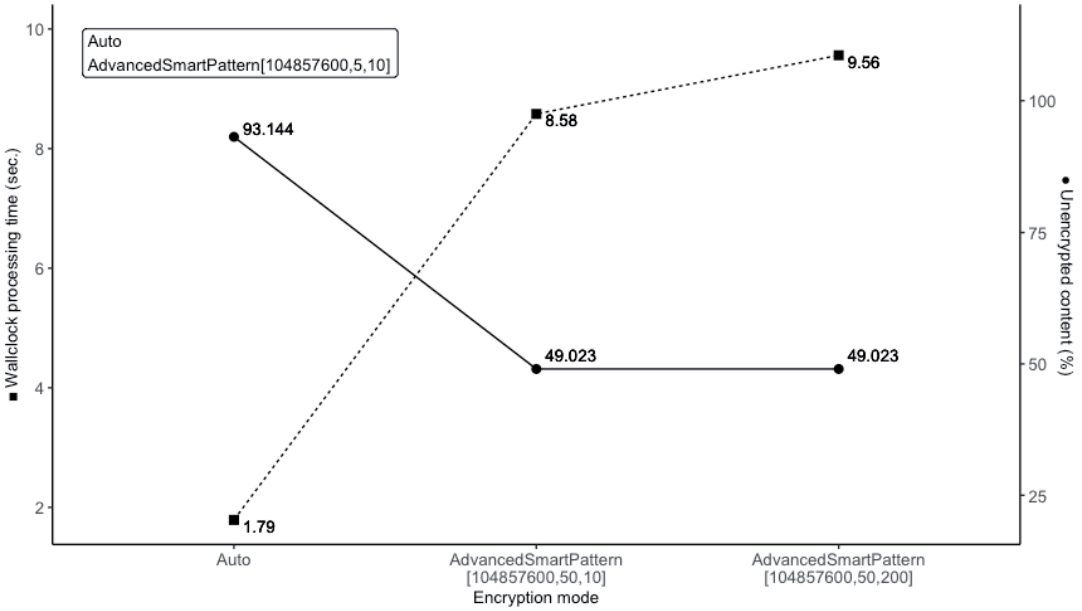

Figure 16: Wallclock processing time and unencrypted content [Encryption algorithm: AES; File encryption modes: Auto, AdvancedSmartPattern].

BlackCat exhibited the highest unencrypted content (a difference of +44.121%) and the lowest wallclock processing time (differences of -6.79 seconds and -7.77 seconds) when the ransomware used the *Auto* (i.e. *AdvancedSmartPattern [104857600, 5, 10]*, see Figure 5) file encryption mode. This indicates that the ransomware developers have designed *Auto* to minimize the wallclock processing time at the expense of high unencrypted content – for a file size of 5 GB, the *AdvancedSmartPattern* mode exhibited much higher sensitivity to the second parameter *P* than to the third parameter *B,* with a difference of 0% in unencrypted content and only +0.98 seconds in wallclock processing time when *AdvancedSmartPattern [104857600, 50, 200]* is compared to *AdvancedSmartPattern [104857600, 50, 10]*.

CONCLUSIONS AND FUTURE WORK

The experiments that we conducted allow for making the following conclusions.

For a single file, the ability of BlackCat to self-configure – to automatically switch to use the hardware-supported AES over ChaCha20 and to use the *Auto* file encryption mode over *Full* – is beginning to result in significant reductions in wallclock processing time at file sizes larger than 500 MB. Reductions in wallclock processing time are of advantage to ransomware actors who aim to cause irretrievable damage to data in the shortest possible time.

The BlackCat ransomware developers have designed the *Auto* file encryption mode to minimize the wallclock processing time at the expense of high unencrypted content. The ransomware developers may have made this decision since BlackCat, with *Auto* enabled, will fully encrypt files of the formats that the developers consider critical to victims and/or allow for recovery of useful data if the files are only partially encrypted (see Figure 4). Victims would be able to recover useful content from any file of a format that is not covered by the list of filename extensions that Figure 4 depicts, if the format allows for such a recovery.

Preventing BlackCat from executing is the best strategy for protecting against the ransomware. Once a malicious actor executes BlackCat, the ransomware encrypts a multitude of files in the order of minutes, even when the ransomware uses the time-consuming file encryption mode *Full* (see Figure 15). The current global median dwell time for ransomware-related intrusions (i.e. time-to-ransom) is five days [10]. Defenders have a significantly longer time window to react to an attack that involves BlackCat before ransomware execution than after. However, there are opportunities for further research at executable-level to reinforce the existing defensive barriers against an executing ransomware.

This work is an initial milestone of a broader research agenda of the author that looks at malware developers as software designers and focuses on the design decisions that the developers make about the malware configuration spaces. Some of the domains to which this perspective is relevant ,and in which the author bases his future work, are the following:

- Detection: researching malware configuration spaces can drive the development of new detection methods and guide detection development efforts. In the context of the BlackCat ransomware, novel detections could be based on profiles of file I/O operations that map to the different individual file encryption modes (*Full*, *HeadOnly*, *DotPattern*, *SmartPattern* and *AdvancedSmartPattern*). Given the different encryption algorithms that BlackCat supports, detection engineers could also develop profiles of AES-NI or ChaCha20 operations that are indicative of BlackCat. In addition, detection engineers could examine the feasibility of creating configuration-specific performance signatures based on, for example, CPU utilization, to detect BlackCat and the malware's active configuration.
- Response: detections that identify malware configurations can enable the automated estimation of the criticality of the malware's operation. For example, in the context of BlackCat, the different file encryption modes exhibit different levels of destructiveness to data. In addition, detections that identify malware configurations can enable the automated estimation of the available response time once the malware has executed. The response time is especially critical if the malware's operation is destructive to data, such as in the case of ransomware. For example, in the context of BlackCat, the different file encryption modes exhibit different wallclock processing times, which express upper response time limits.
- The automated estimation of criticality and response time can drive the design and decision-making processes of response mechanisms. For example, when suspecting a particular ransomware configuration, a response mechanism may prioritize responding and stop the suspected ransomware process within a time limit based on the estimated response time in favour of gathering more ransomware behaviour evidence to minimize the chance of a false positive.
- Attribution and threat intelligence: Detections that identify malware configurations that threat analysts have mapped to a threat actor or affiliates may complement other indicators of compromise in attribution efforts and increase attribution confidence. In addition, researching malware configuration spaces is useful to better understand malware development and usage trends. Analysing malware configuration spaces also helps to better understand the dynamics of the malware market in terms of what features malware developers offer to buyers to address their demands.

REFERENCES

[1] Scholz, C.; Verfürden, M. "Black Cat"-Erpressersoftware: Staatsanwaltschaft ermittelt nach Angriff auf Tankstellen-Zulieferer. Handelsblatt. 02 February 2022. https://www.handelsblatt.com/unternehmen/energie/benzinversorgung-black-cat-erpressersoftware-staatsanwaltschaft-ermittelt-nach-angriff-auf-tankstellen-zulieferer/28029264.html?nlayer=Themen_11804704.

[2] Pearson, J. UPDATE 4-Shell re-routes oil supplies after cyberattack on German firm. Reuters. 01 February 2022. https://www.reuters.com/article/germany-cyber-shell-idCNL1N2UC0QD.

[3] Sharma, A. BlackCat (ALPHV) claims Swissport ransomware attack, leaks data. BleepingComputer. 15 February 2022. https://www.bleepingcomputer.com/news/security/blackcat-alphv-claims-swissport-ransomware-attack-leaks-data/.

[4] Abrams, L. Ransomware gang creates site for employees to search for their stolen data. BleepingComputer. 14 June 2022. https://www.bleepingcomputer.com/news/security/ransomware-gang-creates-site-for-employees-to-search-for-their-stolen-data/.

[5] Hill, J. BlackCat Ransomware (ALPHV). Varonis. 26 January 2022. https://www.varonis.com/blog/blackcat-ransomware.

[6] Kovar, R.; S. Davis, S. Gone in 52 Seconds…and 42 Minutes: A Comparative Analysis of Ransomware Encryption Speed. Splunk. 23 March 2022. https://www.splunk.com/en_us/blog/security/gone-in-52-seconds-and-42-minutes-a-comparative-analysis-of-ransomware-encryption-speed.html.

[7] Pereira, T.; Huey, C. From BlackMatter to BlackCat: Analyzing two attacks from one affiliate. Talos Intelligence. 17 March 2022. https://blog.talosintelligence.com/2022/03/from-blackmatter-to-blackcat-analyzing.html.

[8] Microsoft 365 Defender Threat Intelligence Team. The many lives of BlackCat ransomware. 13 June 2022. https://www.microsoft.com/security/blog/2022/06/13/the-many-lives-of-blackcat-ransomware/.

[9] Walter, J. BlackCat Ransomware | Highly-Configurable, Rust-Driven RaaS On The Prowl For Victims. SentinelOne. 18 January 2022. https://www.sentinelone.com/labs/blackcat-ransomware-highly-configurable-rust-driven-raas-on-the-prowl-for-victims/.

[10] Mandiant. M-Trends 2022: Special Report. 2022.