

01flip: Multi-Platform Ransomware Written in Rust

 unit42.paloaltonetworks.com/new-ransomware-01flip-written-in-rust

Hiroaki Hara

December 10, 2025

Executive Summary

In June 2025, we observed a new ransomware family named 01flip targeting a limited set of victims in the Asia-Pacific region. 01flip ransomware is fully written in the Rust programming language and supports multi-platform architectures by leveraging the cross-compilation feature of Rust.

These financially motivated attackers likely carried this out through manual means. We have confirmed an alleged data leak from an affected organization on a dark web forum shortly after the attack. We are currently [tracking this activity](#) as CL-CRI-1036, signifying a cluster of malicious activity that is likely related to cybercrime.

Our key findings are:

- Financially motivated attackers behind CL-CRI-1036 use 01flip ransomware, a newly observed ransomware family purely written in Rust
- This ransomware supports multi-platform architecture, including Windows and Linux
- A threat actor potentially associated with CL-CRI-1036 is offering data for sale on dark web forums (likely stolen with 01flip ransomware)

While the impact of CL-CRI-1036 is limited at this point, it's likely that this activity is related to alleged data leaks.

Palo Alto Networks customers are better protected from the threats described through the following products and services:

- [Advanced WildFire](#)
- [Cortex XDR](#) and [XSIAM](#)
- [Cortex Xpanse](#)

If you think you might have been compromised or have an urgent matter, contact the [Unit 42 Incident Response team](#).

Related Unit 42 Topics [Rust](#), [Sliver](#), [Ransomware](#), [Cybercrime](#)

Background

In early June 2025, Unit 42 researchers investigated a suspicious Windows executable. The executable caught our attention because it is a Rust-based binary that exhibited ransomware-like behavior in our sandbox.

[Our initial analysis](#) revealed that this is a new ransomware family fully written in Rust, called “01flip.” This name is based on the appended file extension (.01flip) and the email address (01Flip@proton[.]me) found in the ransom note.

After further investigation, we discovered a [Linux version of 01flip ransomware](#), which showed zero detection rate at least three months after the sample had initially been submitted to VirusTotal.

Campaign Overview

Victimology

At the time of writing, we observed a minimal set of victims. But victims of this ransomware include organizations responsible for critical infrastructure in Southeast Asia.

After further investigation, we found [a post on an online security forum](#) by an alleged victim of 01flip ransomware, claiming that attackers had compromised their Zimbra Server, which is an email solution for enterprises.

Due to the lack of information about victims, we assume that the use of 01flip ransomware is at a very early stage. However, there might be several victims in the Philippines and Taiwan, based on a post in a dark web forum that we believe comes from a threat actor linked to CL-CRI-1036.

Initial Access

Investigation revealed that attackers had been attempting exploits targeting older vulnerabilities such as [CVE-2019-11580](#) against a victim’s internet-facing applications since early April 2025. It is unclear which activities specifically succeeded in granting attackers access to this system. However, a month later, they successfully deployed a Linux version of [Sliver](#), a publicly available cross-platform adversary emulation framework written in Go.

Post-Exploitation

In late May 2025, the threat actor behind CL-CRI-1036 successfully performed lateral movement to another Linux machine by downloading another Sliver implant, which was a [TCP Pivot](#) profiled implant. A week later, we confirmed that attackers deployed multiple 01flip ransomware instances onto many devices within the network, including both Windows and Linux machines.

The exact methods the attackers used to deploy the ransomware after the initial compromise remain unclear. However, given the rapid distribution of the ransomware to multiple devices, it is highly likely that the attackers conducted the following activities, possibly through Sliver and its modules:

- Hands-on reconnaissance
- Credential dumping
- Lateral movement

Technical Analysis of 01flip Ransomware

Initial Analysis

As of late October, we had observed both Windows and Linux versions of 01flip ransomware. The 01flip ransomware sample is not packed or heavily obfuscated, unlike other malware used by cybercriminals. Therefore, we can easily identify that it is compiled from Rust source code, due to the file extension (.rs), as shown in Figure 1.

```
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/cipher-0.4.4/src/stream_core.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/cipher-0.4.4/src/stream_wrapper.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/rsa-0.9.8/src/algorithms/pad.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/rsa-0.9.8/src/algorithms/pkcs1v15.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/spin-0.9.8/src/once.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/tokio-1.44.2/src/fs/open_options.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/tokio-1.44.2/src/fs/rename.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/tokio-1.44.2/src/fs/mod.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/tokio-1.44.2/src/task/join_set.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/tokio-1.44.2/src/util/idle_notified_set.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/base64-0.22.1/src/engine/mod.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/cipher-0.4.4/src/stream.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/ctr-0.9.2/src/flavors/ctr64.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/tokio-1.44.2/src/io/util/write_all.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/generic-array-0.14.7/src/lib.rs
yfrxwtqnkbuvxoag/.cargo/registry/src/index.crates.io-6f17d22bba15001f/aes-0.8.4/src/soft/fixslice64.rs
```

Figure 1. Visible Rust-related strings in the 01flip ransomware sample.

Figure 2 shows the differences in results using [rustbininfo](#) between the Windows and Linux platform samples. The rustbininfo tool generates a list of dependencies (known as “crates” in Rust), which we used to compare the two samples.

Aside from architecture-specific libraries, we can see that the Rust version, commit hash and library versions mostly match. Therefore, most of the functionality is identical, but we also noticed a few differences that we will describe later.

```

1 TargetRustInfo{
2   rustc_version='1.81.0',
3   rustc_commit_hash='eeb90cda1969383f56a2637cbd3037bdf598841c',
4   dependencies=[
5     Crate(name='addr2line', version='0.22.0'),
6     Crate(name='aes', version='0.8.4'),
7     Crate(name='base64', version='0.22.1'),
8     Crate(name='base64ct', version='1.7.3'),
9     Crate(name='cipher', version='0.4.4'),
10    Crate(name='const-oid', version='0.9.6'),
11    Crate(name='ctr', version='0.9.2'),
12    Crate(name='der', version='0.7.10'),
13    Crate(name='generic-array', version='0.14.7'),
14    Crate(name='gimli', version='0.29.0'),
15    Crate(name='hashbrown', version='0.14.5'),
16    Crate(name='mio', version='1.0.3'),
17    Crate(name='num-bigint-dig', version='0.8.4'),
18    Crate(name='parking_lot', version='0.12.3'),
19    Crate(name='parking_lot_core', version='0.9.10'),
20    Crate(name='pem-rfc7468', version='0.7.0'),
21    Crate(name='rand', version='0.8.5'),
22    Crate(name='rand_chacha', version='0.3.1'),
23    Crate(name='rand_core', version='0.6.4'),
24    Crate(name='rsa', version='0.9.8'),
25    Crate(name='smallvec', version='1.15.0'),
26    Crate(name='spin', version='0.9.8'),
27    Crate(name='tokio', version='1.44.2'),
28    Crate(name='walkdir', version='2.5.0'),
29    Crate(name='winapi-util', version='0.1.9'),
30    Crate(name='zeroize', version='1.8.1')
31  ],
32  rust_dependencies_imphash='fbabb88cc49610a7accdd7178694d8cc6',
33  guessed_toolchain='Mingw-w64 (Could not find mingw/gcc version)'
34 }

```

```

1 TargetRustInfo{
2   rustc_version='1.81.0',
3   rustc_commit_hash='eeb90cda1969383f56a2637cbd3037bdf598841c',
4   dependencies=[
5     Crate(name='addr2line', version='0.22.0'),
6     Crate(name='aes', version='0.8.4'),
7     Crate(name='base64', version='0.22.1'),
8     Crate(name='base64ct', version='1.7.3'),
9     Crate(name='cipher', version='0.4.4'),
10    Crate(name='const-oid', version='0.9.6'),
11    Crate(name='ctr', version='0.9.2'),
12    Crate(name='der', version='0.7.10'),
13    Crate(name='generic-array', version='0.14.7'),
14    Crate(name='gimli', version='0.29.0'),
15    Crate(name='hashbrown', version='0.14.5'),
16+   Crate(name='miniz_oxide', version='0.7.4'),
17    Crate(name='mio', version='1.0.3'),
18    Crate(name='num-bigint-dig', version='0.8.4'),
19    Crate(name='parking_lot', version='0.12.3'),
20    Crate(name='parking_lot_core', version='0.9.10'),
21    Crate(name='pem-rfc7468', version='0.7.0'),
22    Crate(name='rand', version='0.8.5'),
23    Crate(name='rand_chacha', version='0.3.1'),
24    Crate(name='rand_core', version='0.6.4'),
25    Crate(name='rsa', version='0.9.8'),
26+   Crate(name='same-file', version='1.0.6'),
27+   Crate(name='signal-hook-registry', version='1.4.5'),
28    Crate(name='smallvec', version='1.15.0'),
29    Crate(name='spin', version='0.9.8'),
30    Crate(name='tokio', version='1.44.2'),
31    Crate(name='walkdir', version='2.5.0'),
32    Crate(name='zeroize', version='1.8.1')
33  ],
34  rust_dependencies_imphash='c28097471ca1e38bac03877f7e90852f',
35  guessed_toolchain='linux-musl'
36 }

```

Figure 2. The two 01flip ransomware samples used mostly the same crates (left: Windows, right: Linux).

The Rust compiler generally produces more complex assembly code than traditional C/C++ compilers. This added complexity can make reverse engineering Rust malware challenging for malware analysts. Despite this, the malware's capabilities are simple and straightforward.

Ransomware Functionality

The following are key functions of 01flip ransomware:

1. Enumerating all possible drives (e.g., from A: to Z:)
2. Creating ransom notes, RECOVER-YOUR-FILE.TXT, in all writable directories
3. Renaming files containing specific file extensions with the following naming convention:
<ORIGINAL_FILENAME>.<UNIQUE_ID>.<0 or 1>.01flip
4. Encrypting files using AES-128-CBC and RSA-2048
5. Deleting itself

Defense Evasion Techniques

01flip ransomware employs several evasion techniques. For example, both Linux and Windows versions are designed to use low-level APIs or system calls as much as possible, because such activity is less likely to stand out from normal operating system activity. Figure 3 shows an example of native APIs used in our sample of the Windows version of 01flip ransomware.

ntdll			
 00007FF626156768	NtReadFile		ntdll
 00007FF626156770	NtWriteFile		ntdll

Figure 3. Example of low-level native APIs used to read/write files in the Windows version.

Additionally, most of the user-defined strings in the ransomware code are encoded. These strings are decoded at runtime:

- The ransom note content
- The ransom note filename
- The extension list
- The RSA public key

The algorithm decodes each encoded string by performing a SUB operation every two bytes as follows.

```

1  from more_itertools import chunked
2
3  def decrypt_string(enc: bytes) -> str:
4
5  return "".join(chr(chunk[0] - chunk[1]) for chunk in chunked(enc, 2))

```

Encoded strings are embedded in the .text or .data section of the Windows variant based on their length. Figure 4 demonstrates how 01flip ransomware decodes the ransom note template.

```
In [28]: enc = idaapi.get_bytes(0x7FF6261336B0, 992) # address of encoded string
enc[:0x100]

Out[28]: 00000000: 55 18 5E 21 2E 0E 74 2B 94 47 60 10 9C 4D 92 40 |U ^!. t+.G` .M.@|
00000010: 7C 28 48 07 5B 0D 76 22 5F 3F 8C 4F 55 18 1A 10 |(H [ v"? .OU |
00000020: 63 0A 96 27 B0 3B 77 05 27 07 99 33 B7 4E A6 3A |c .';w ' .3.N.:|
00000030: A4 3F AE 3B 20 00 7F 17 8F 2E B0 3A 70 0B 24 04 |.?.; ...:p $ |
00000040: 99 37 9B 36 82 1D BA 4C 6A 4A 71 0C B9 4B 8C 29 |.7.6. .LjJq .K.)|
00000050: A3 31 79 00 96 26 B7 43 6B 06 8D 29 43 15 55 35 |.ly .&.Ck .)C U5|
00000060: 84 40 8F 20 2E 0E 93 25 B1 42 93 1F 5D 3D 7F 1E |.@. . %.B. ]= |
00000070: AF 3B AD 39 8B 26 80 13 81 11 7B 07 36 16 86 12 |.;.9.&. . { 6 . |
00000080: 8D 1E 44 24 9F 2D 81 1C 9C 39 AD 3E 9B 25 67 02 |. D$.-. .9.>.%g |
00000090: C0 4E 20 00 7F 0B A8 40 AD 48 72 05 60 40 9B 22 |.N .@.Hr `@."|
000000A0: 73 04 86 11 9F 2D 7F 0C 9E 39 79 0D AC 46 62 36 |s . .- .9y .Fb6|
000000B0: 24 04 8C 2B 9A 27 68 48 B1 3D 6A 02 A0 37 8C 19 |$.+.'hH.=j .7. |
000000C0: 30 10 8F 22 87 26 A2 29 63 43 68 05 82 21 B8 43 |0 .".&.)cCh .!.C|
000000D0: AE 3B 9A 35 44 24 A6 3D A8 36 C0 4E AC 47 B0 3A |.;.5D$.=.6.N.G.:|
000000E0: A2 3D A4 32 77 04 AE 45 AD 4B AB 3F 78 13 40 20 |.-.2w .E.K.?x @ |
000000F0: 9F 3B 9A 39 AD 40 8E 2D 8F 28 98 33 77 49 2D 23 |.;.9.@.-.(.3wI-#|
```

```
In [29]: dec = decrypt_string(enc)
print(dec)

== IMPORTANT ==
Your files have been encrypted. Do not attempt to recover them yourself, as this may cause irreversible damage.
Once we receive payment, we will immediately provide the decryption key to restore your data.
Every hour of delay increases the risk of permanent data loss.

Email:
>> 01Flip@proton.me
Session (https://getsession.org, copy and paste to send a friend request):
>> [REDACTED]
Cipher:
>> ${cipher_data}
```

Figure 4. How 01flip ransomware decodes the ransom note stored with its binary.

Techniques like invoking system calls and encoded strings are not an effective evasion strategy on their own. Because the 01flip ransomware sample we analyzed works as designed in a sandbox environment, it is still relatively easy to detect.

Most ransomware is relatively straightforward and noisy. However, some of the 01flip ransomware samples implement a simple anti-sandbox technique by checking whether the filename contains the string 01flip. If the sample's filename contains the string 01flip, the ransomware proceeds to [indicator removal](#) without performing file encryption.

Data Encryption

01flip ransomware drops ransom notes in all writable directories before encrypting files. These ransom notes contain contact information and cipher data that is required to obtain a key to decrypt a victim's files, as shown in Figure 5.



Figure 5. Example of a 01flip ransom note.

01flip ransomware excludes files with specific extensions from encryption. This encryption exclusion extension list can be found in the . Finally, 01flip encrypts files using the AES encryption algorithm. The session key used for the ransomware's file-encrypting activity is itself encrypted using an embedded RSA public key. Figure 6 shows an example of this RSA key.

```
aBeginPublicKey db '-----BEGIN PUBLIC KEY-----',0Ah
db 'MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvIuA1GeAWa2G1PjljXc2',0Ah
db 'uCbvg8TjmbJx16Q+sURo0ZdBYL3roS65rLDW7kEpa4L4mtkAHkQmsbX/CzLdBfxJ',0Ah
db 'AVnbc7kPVU3L73U2cA+UWgy0TLX2Lu/4FbRtDM/W00BJtYzAcSDMLz6tB7SqGbA8',0Ah
db 'piXR369DY4LpvNY4oT8n6kpQJ33y3wTE5EC9m0Ij4PVac4lrOM64P1JaRqes5DWI',0Ah
db 'YNcq3fICEElVbL1sYSpmJJ9+gN5kHKbmTUyyb0P8TDkLmS1wdu7My1v2kM/LEPY2',0Ah
db 'q9rFs0kB8MdiZjyKl2+wgIHsEVe6+X2XqlSt97hZQrMFxtSN8YYwf2IEv/KjU3GJ',0Ah
db 'oQIDAQAB',0Ah
db '-----END PUBLIC KEY-----'
```

Figure 6. RSA public key from the 01flip ransomware sample.

Encrypted files are renamed with the specific naming convention <ORIGINAL_FILENAME>.<UNIQUE_ID>.<0 or 1>.01flip, as shown in Figure 7.

Name	Date
 image.jpg.9132999c14f34f8b0219a1368ef36add.0.01flip	6/22/2025 4:27 PM
 RECOVER-YOUR-FILE.TXT	6/24/2025 3:19 PM
 Screenshot.png.e86692ed45a0fe4006d74fdc16d5aa51.0.01flip	6/26/2023 3:13 PM

Figure 7. Example of encrypted filenames in a Windows environment infected with 01flip ransomware.

Indicator Removal

After completing encryption, the 01flip ransomware attempts to remove any trace of itself, to prevent it from being recovered from an infected host. The Windows and Linux variants invoke the following commands, respectively, after replacing `${self_name}` with its current filename.

```
1 # Windows
```

```
2
```

```
3 ping 127.0.0.7 -n 5 > Nul & fsutil file setZeroData offset=0 length=4194303 ${self_name} >
  Nul & Del /f /q ${self_name}
```

```
1 # Linux
```

```
2
```

```
3 sleep 5 && dd if=/dev/urandom of=${self_name} bs=1M count=4 > /dev/ 2>&1 && rm
  ${self_name} > /dev/ 2>&1
```

Possible Attribution

Attacker Activity

Attackers behind this campaign have so far demanded one bitcoin (BTC) for decrypting files. Communication with the attackers takes place in a secure email or in a private messaging channel. Figure 8 shows an example of the demand through messaging channel text.

As of late October, the attackers behind CL-CRI-1036 don't appear to operate a double extortion site like those commonly seen in recent ransomware-as-a-service (RaaS) groups.

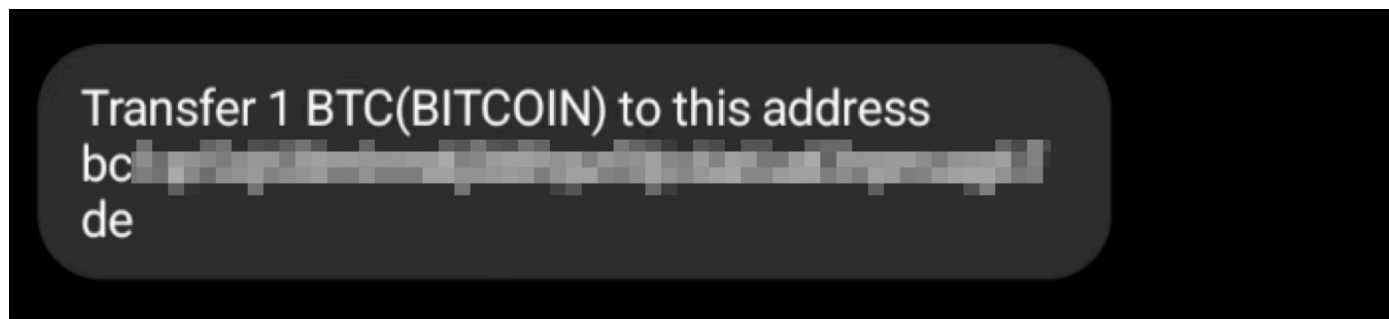


Figure 8. Message from the attackers in a private messaging channel.

Further investigation revealed an alleged data leak of the affected organization on a dark web forum posted on the day after the ransomware deployment. Figure 9 shows this post. While we were unable to verify the legitimacy of the post, the data appears to be credible based on the

positive reactions of other users in the forum.

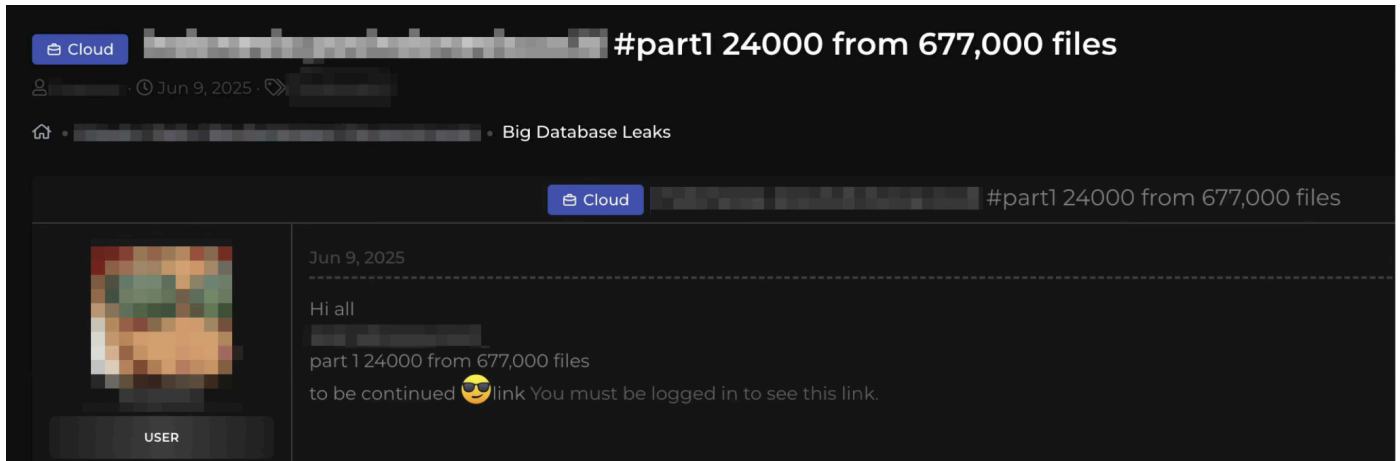


Figure 9. Post of the alleged data leak on a dark web forum.

While this user who made this post has been registered in this forum since April 2023, we were able to confirm only three posts since June 2025. The post indicates that victims in Taiwan and the Philippines were also targeted. The username itself is not particularly unique, but we have confirmed that a Russian-speaking user with the same username has been selling data and network access on the well-known dark web forum XSS since 2020.

Of note, 01flip ransomware cannot exfiltrate data, so the only connection between this poster and 01flip is the victim.

Possible Overlap With LockBit?

During our analysis of 01flip ransomware, we found that there is one particularly interesting extension in the list of file extensions excluded from encryption, which is lockbit as shown in Figure 10.

```

dq offset aIcs                ; "ics"
dq 3                          ; "Idf"
dq 3                          ; "idx"
dq offset aIdx                ; "ini"
dq 3                          ; "Ink"
dq offset aInk                ; "ipa"
dq 3                          ; "iso"
dq offset aIso                ; "key"
dq 3                          ; "lnk"
dq 3                          ; "lock"
dq 4                          ; "lockbit"
dq offset aLockbit            ; "mod"
dq 7                          ; "mp3"
dq offset aMod                ; "mp4"
dq 3                          ; "mpa"
dq offset aMp3                ; "mpa"
dq 3                          ; "mp4"
dq offset aMp4                ; "mpa"
dq 3                          ; "mpa"
dq offset aMpa                ; "mpa"
dq 3                          ; "mpa"
dq offset aMsc                ; "mpa"
dq 3                          ; "mpa"
dq offset aMsi                ; "mpa"
dq 3                          ; "mpa"
dq offset aMsp                ; "mpa"
dq 3                          ; "mpa"

```

Figure 10. Lockbit extension in the list of file extensions excluded from encryption by the 01flip ransomware sample.

Avoiding encrypting files with a lockbit file extension implies a possible overlap of the threat actor behind CL-CRI-1036 and the group behind LockBit ransomware, which we track as Flighty Scorpius. However, other than this odd bit of code, we can find no other connection between these two ransomware families.

Conclusion

We have described emerging activity, which we currently track as CL-CRI-1036, where financially motivated attackers used a new Rust-based ransomware named 01flip. This activity highlights the challenges faced by defenders from attackers using modern programming languages in malware development. Based on our analysis, this campaign seems to be in its early stages and may be related to an alleged data leak on a dark web forum.

Palo Alto Networks Protection and Mitigation

Palo Alto Networks customers are better protected from the threats discussed above through the following products:

- The [Advanced WildFire](#) machine-learning models and analysis techniques have been reviewed and updated in light of the indicators shared in this research.
- [Cortex XDR](#) and [XSIAM](#) help prevent the threats outlined in this blog by employing the [Malware Prevention Engine](#). This approach combines several layers of protection, including [WildFire](#), Behavioral Threat Protection and the Local Analysis module, to prevent both known and unknown malware from causing harm to endpoints.

In addition, the [Anti-Ransomware module](#) enables Cortex XDR to protect against encryption-based activity associated with ransomware, to help analyze and halt ransomware before any data loss occurs.

[Cortex Xpanse](#) can help detect internet-facing instances of insecure software, such as Atlassian Crowd Server, which was noted as being exploited by CVE-2019-11580 and leading to initial access.

Cortex Xpanse has an [Attack Surface Rule](#) for “Insecure Atlassian Crowd Server” meant to identify this example, as well as many other detections out of the box for applications with RCE vulnerabilities which are attractive for Ransomware operators.

All detections in Cortex Xpanse are also available in Cortex XSIAM as part of the Attack Surface Management (ASM) add-on.

If you think you may have been compromised or have an urgent matter, get in touch with the [Unit 42 Incident Response team](#) or call:

- North America: Toll Free: +1 (866) 486-4842 (866.4.UNIT42)
- UK: +44.20.3743.3660
- Europe and Middle East: +31.20.299.3130
- Asia: +65.6983.8730
- Japan: +81.50.1790.0200
- Australia: +61.2.4062.7950

- India: 000 800 050 45107
- South Korea: +82.080.467.8774

Palo Alto Networks has shared these findings with our fellow Cyber Threat Alliance (CTA) members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. Learn more about the [Cyber Threat Alliance](#).

Indicators of Compromise

Malicious Samples

Windows Version of 01flip Ransomware

- SHA-256 hash:
6aad1c36ab9c7c44350ebe3a17178b4fd93c2aa296e2af212ab28d711c0889a3
- File size: 741,888 bytes
- File type: PE32+ executable (GUI) x86-64 (stripped to external PDB), for MS Windows

Linux Version of 01flip Ransomware

- SHA-256 hash:
e5834b7bdd70ec904470d541713e38fe933e96a4e49f80dbfb25148d9674f957
- File size: 948,640 bytes
- File type: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), static-pie linked, stripped

Linux Sliver Beacon, TCP Pivot as C2

- SHA-256 hash:
ba41f0c7ea36cefe7bc9827b3cf27308362a4d07a8c97109704df5d209bce191
- File size: 13,414,400 bytes
- File type: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, stripped

Appendix: Encryption Exclusion Extension List

A-L

- 01flip
- 386
- ldf
- lnk
- a
- adv
- ani
- apk

- app
- bat
- bin
- cab
- cmd
- com
- cpl
- cur
- deb
- deskthemepack
- diagcab
- diagcfg
- diagpkg
- dll
- dmg
- dmp
- drv
- encrypt
- exe
- fnt
- fon
- gadget
- hlp
- hta
- icl
- icns
- ico
- ics
- idx
- ini
- ipa
- iso
- key
- la
- lnk
- lock
- lockbit

M-Z

- mod
- mp3

- mp4
- mpa
- msc
- msi
- msp
- msstyles
- msu
- nls
- nomedia
- o
- ocx
- otf
- part
- pdb
- pif
- prf
- ps1
- reg
- rom
- rpd
- rtp
- scr
- sfcache
- shs
- so
- spl
- sys
- theme
- themepack
- tmp
- ttf
- wad
- wav
- winmd
- wma
- woff
- wpx
- xex