aDvens

Security for the greater good

Monthly Cyber Threat Intelligence report

June 2024

2024-07-08 I **TLP:CLEAR** I CERT aDvens - CTI
Advens - 38 rue des Jeuneurs - 75002 Paris

Table of content

1. Executive summary

This month, aDvens' CERT presents three noteworthy vulnerabilities, in addition to those already published.

Through two articles, the CERT's analysts discuss:

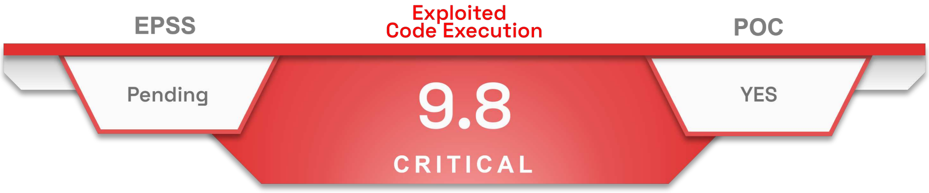
- GOMIR malware exploited by the North Korean Kimsuky group.
- A threat assessment just days before the opening of the 2024 Olympic Games.

2. Vulnerabilities

This month, aDvens' CERT focused on **three** vulnerabilities affecting technologies frequently used within companies.
They are presented in order of severity (proof of concept available, exploitation, etc.). The application of their patches or workarounds is strongly recommended.

2.1. CVE-2024-4577

On 6 June 2024, PHP issued several security bulletins concerning a critical vulnerability in PHP CGI and published appropriate patches. This vulnerability has existed since 2012, and is a bypass of CVE-2012-1823.



An error in PHP CGI when installed on a Windows server allows an unauthenticated attacker to execute arbitrary code on the system by sending specifically forged requests.

2.1.1. Vulnerability type

- **CWE-78**: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')

2.1.2. Risk

- Code execution

2.1.3. Severity (base score CVSS v3.1)

Attack vector	Network	Scope	Unchanged
Attack complexity	Low	Impact on confidentiality	High
Privileges Required	None	Impact on integrity	High
User Interaction	None	Impact on availability	High

2.1.4. Impacted products

- PHP versions 5.X to 8.3.X

2.1.5. Recommendations

Upgrade PHP to version 8.1.29, 8.2.20 or 8.3.8 or later.

Versions 5.x and 8.0.x are no longer supported. We recommend replacing vulnerable products with other versions that have been patched.

Further information is available in the PHP bulletins for the various versions of the product.

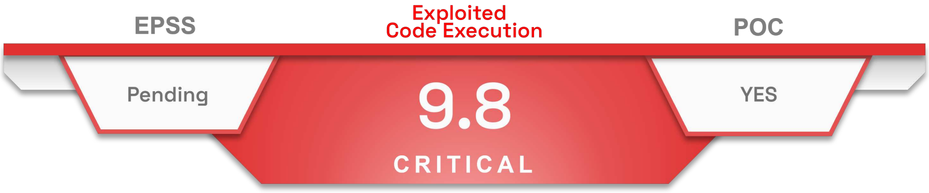
- Version 8.3.8
- Version 8.2.20
- Version 8.1.29

2.1.6. Proof of Concept

A proof of concept is available in open source.

2.2. CVE-2024-29973

On 4 June 2024, Zyxel published a security advisory to correct the CVE-2024-29973 vulnerability affecting several NAS servers.



EPSS

Pending

Exploited Code Execution

9.8

CRITICAL

POC

YES

A flaw in the "setCookie" parameter of Zyxel NAS326 and NAS542 allows an unauthenticated attacker to send specially crafted HTTP POST requests, with the aim of executing arbitrary code.

2.2.1. Vulnerability type

- CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')

2.2.2. Risk

- Code Execution

2.2.3. Severity (base score CVSS v3.1)

Attack vector	Network	Scope	Unchanged
Attack complexity	Low	Impact on confidentiality	High
Privileges Required	None	Impact on integrity	High
User Interaction	None	Impact on availability	High

2.2.4. Impacted products

- Zyxel NAS326 versions 5.21(AAZF.16)C0 and earlier
- Zyxel NAS542 versions 5.21(ABAG.13)C0 and earlier

2.2.5. Recommendations

- Update NAS326 to version 5.21(AAZF.17)C0 or later.
- Update NAS542 to version 5.21(ABAG.14)C0 or later.

Zyxel specifies that NAS326 and NAS542 are products for which support is no longer provided from 31 December 2023.

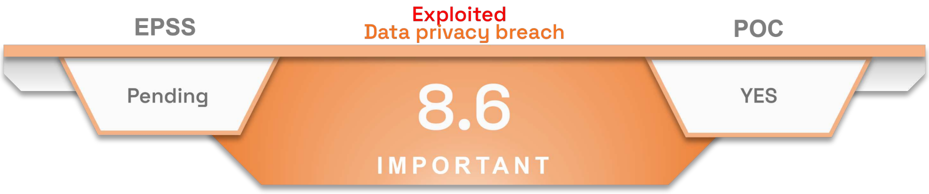
- Further information is available in their advisory.

2.2.6. Proof of Concept

A proof of concept is available in open source.

2.3. CVE-2024-28995

On 5 June 2024, SolarWinds published an alert concerning a critical "directory traversal" vulnerability affecting the SolarWinds Serv-U software.



This vulnerability in SolarWinds Serv-U allows an unauthenticated attacker to send specially crafted requests with the aim of breaching the confidentiality of data on the host machine.

2.3.1. Vulnerability type

- **CWE-22**: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

2.3.2. Risk

- Breach of data confidentiality

2.3.3. Severity (base score CVSS v3.1)

Attack vector	Network	Scope	Changed
Attack complexity	Low	Impact on confidentiality	High
Privileges Required	None	Impact on integrity	None
User Interaction	None	Impact on availability	None

2.3.4. Impacted products

- SolarWinds Serv-U version 15.4.2 HF 1 and earlier

2.3.5. Recommendations

- Update SolarWinds Serv-U to version 15.4.2 HF 2 or later.
- Further information is available at bulletin.

2.3.6. Proof of Concept

A proof of concept is available in open source.

3. Virology: analysis of a Gomir sample (APT Kimsuky)

3.1. A sophisticated backdoor

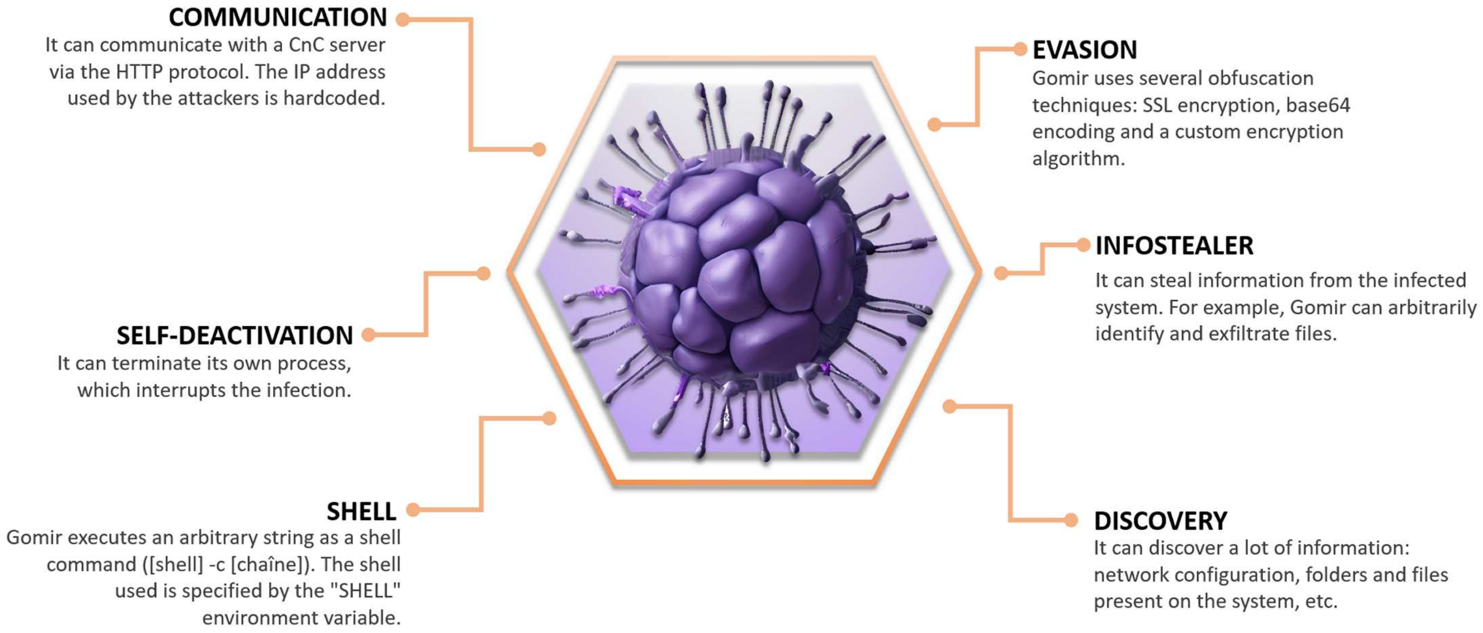
Discovered during the month of May 2024, Gomir is a backdoor used by APT Kimsuky (North Korea). Written in **GO** in **ELF 32** format, this backdoor is specially crafted for **Linux** operating systems.

Gomir was distributed during a cyber-espionage campaign targeting organisations located in South Korea.

Analysis of Gomir reveals a high level of sophistication and many similarities with Gobear (Windows), another backdoor known to belong to the arsenal of APT Kimsuky.

3.2. Features

Below are the main features of the Gomir malware.



COMMUNICATION
It can communicate with a CnC server via the HTTP protocol. The IP address used by the attackers is hardcoded.

SELF-DEACTIVATION
It can terminate its own process, which interrupts the infection.

SHELL
Gomir executes an arbitrary string as a shell command ([shell] -c [chaîne]). The shell used is specified by the "SHELL" environment variable.

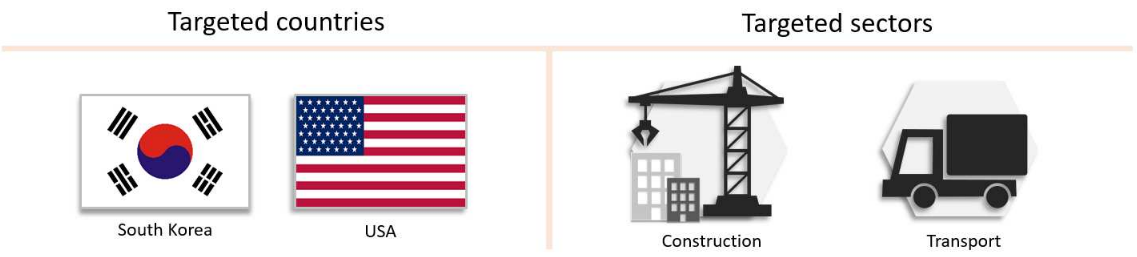
EVASION
Gomir uses several obfuscation techniques: SSL encryption, base64 encoding and a custom encryption algorithm.

INFOSTEALER
It can steal information from the infected system. For example, Gomir can arbitrarily identify and exfiltrate files.

DISCOVERY
It can discover a lot of information: network configuration, folders and files present on the system, etc.

Figure 1. Gomir features: a multifunction backdoor.

3.3. Victimology



3.4. Infectiology

3.4.1. Infection chain: summary

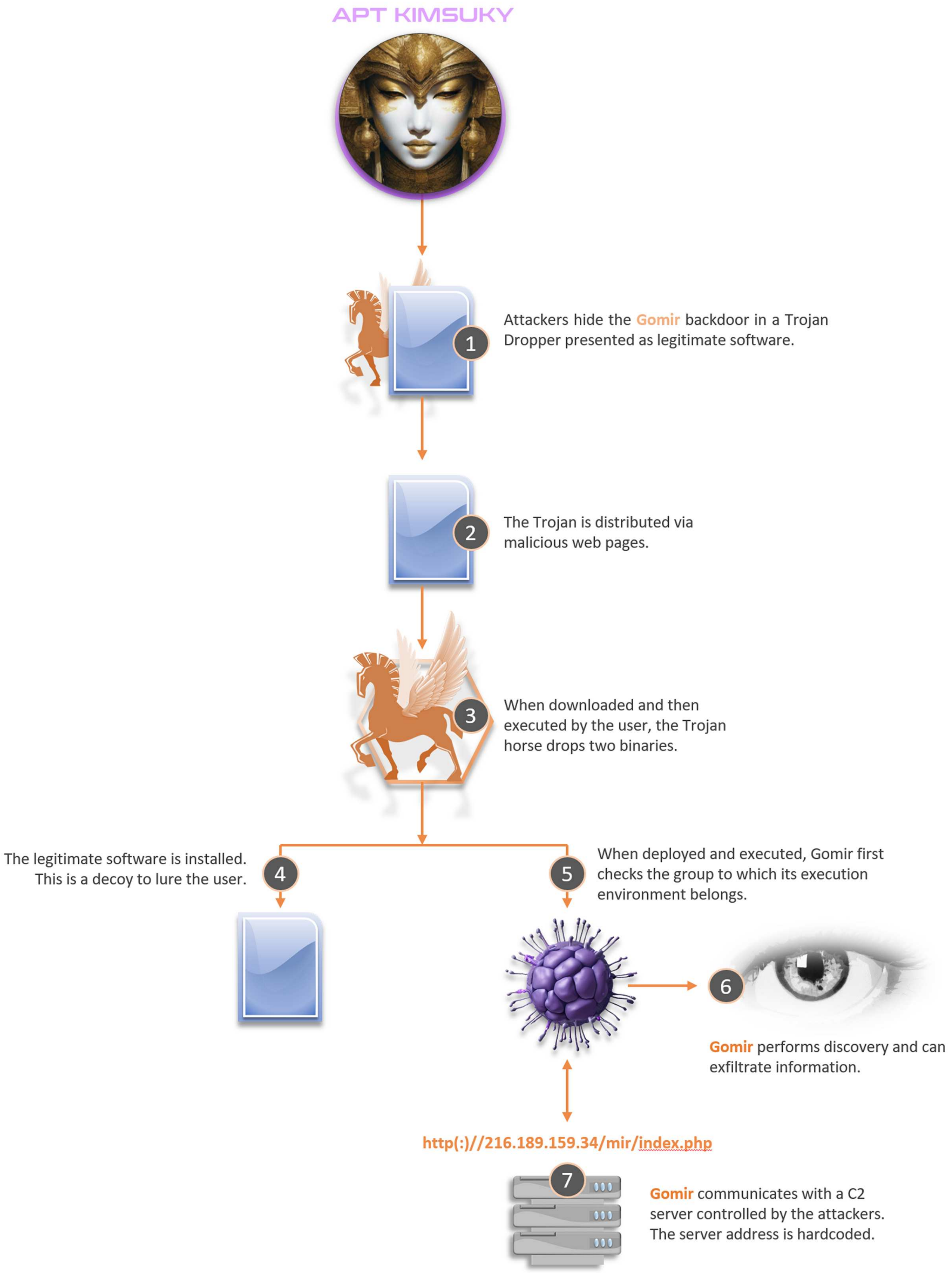

APT KIMSUKY
1
Attackers hide the Gomir backdoor in a Trojan Dropper presented as legitimate software.
2
The Trojan is distributed via malicious web pages.
3
When downloaded and then executed by the user, the Trojan horse drops two binaries.
4
The legitimate software is installed. This is a decoy to lure the user.
5
When deployed and executed, Gomir first checks the group to which its execution environment belongs.
6
Gomir performs discovery and can exfiltrate information.
http(:)//216.189.159.34/mir/index.php
7
Gomir communicates with a C2 server controlled by the attackers. The server address is hardcoded.

Figure 2. Infographic summary of the infection chain.

3.4.2. Infection chain: detailed analysis

Infection vector

The main infection vector used by attackers is the distribution of Trojans Dropper. They are crafted to appear as legitimate software and embed two binaries:

- The legitimate software: the software is deployed and then executed to lure the user. This is the "decoy".
- Gomir payload: a sophisticated backdoor that is installed discreetly.

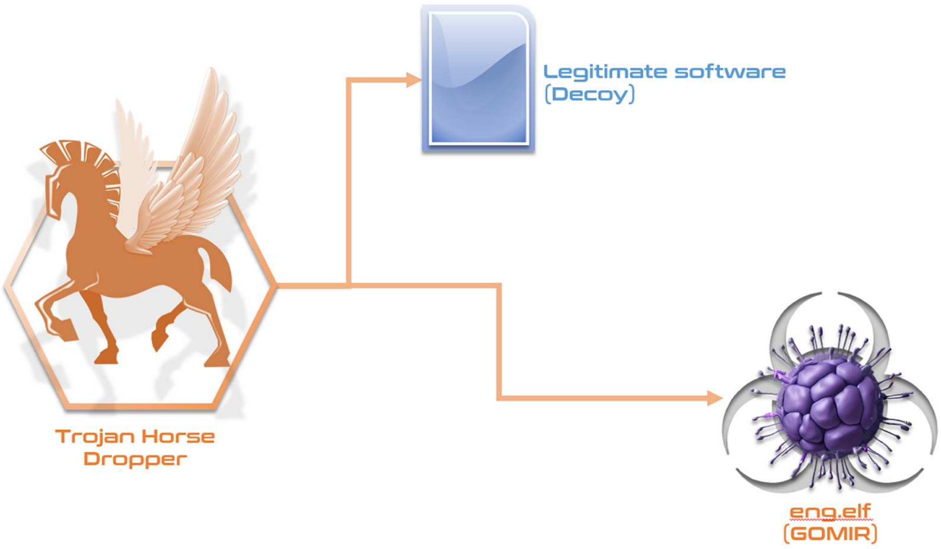


Figure 3. Main infection vector: Trojan horse dropper.

Since the beginning of 2024, various legitimate software programs have been maliciously exploited by APT Kimsuky to hide backdoors (Troll Stealer,Gobear # and [.orange]#Gomir).

- **TrustPKI** and **NX_PRNMAN** from the company *SGA Solutions*.
- **Wizvera VeraPort** from the company *Wizvera*.
- **Humetro** from the company *Humetro Busan Kr*.

Furthermore, the *Wizvera VeraPort* supply chain is known to have been the subject of cyberattacks by APT Lazarus (North Korea) in 2020.

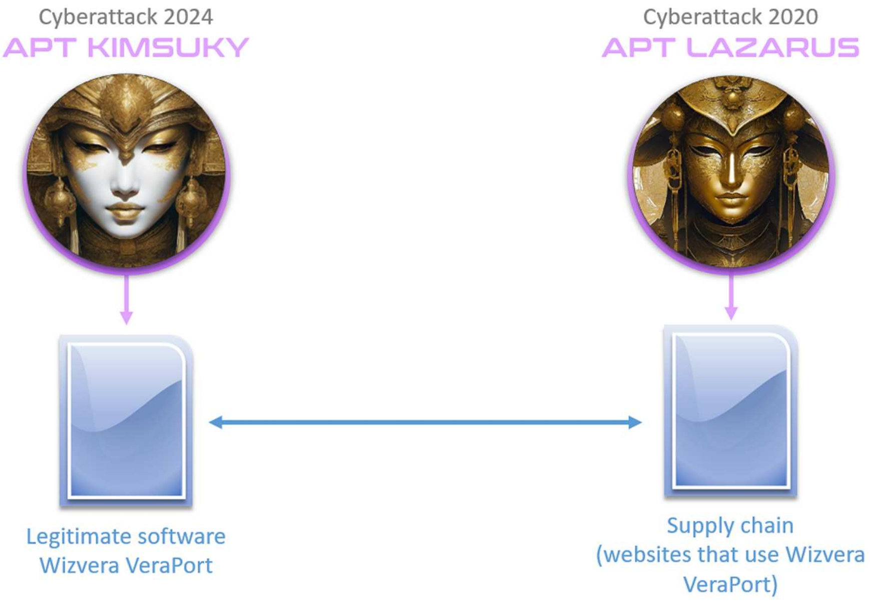


Figure 4. Wizvera VeraPort: a repeated target of the North Korean threat.

3.4.3. Analysis of the virus strain

Executing and verifying the group

When deployed and executed by the Trojan Dropper, Gomir first checks the group to which its execution process belongs. For this, the following function is used:

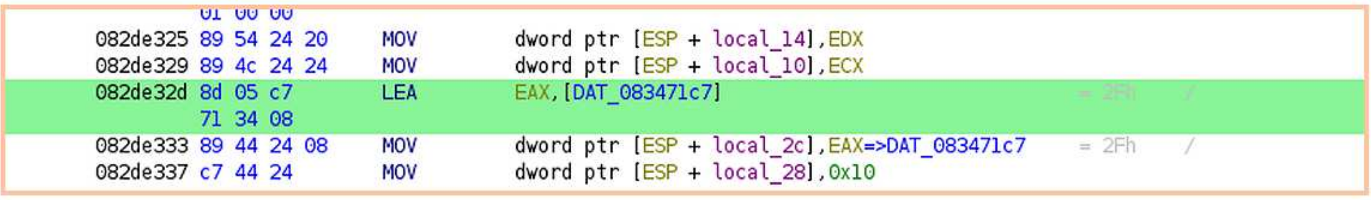
```
if ( syscall_rawSyscallNoError(202, 0, 0, 0) )
```

This resolves the **getegid32()** function. If its process belongs to group 0 (root privilege), then Gomir installs and establishes its persistence via **systemd**. Otherwise Gomir installs and establishes persistence via **crontab**.

Installing with persistence via Systemd

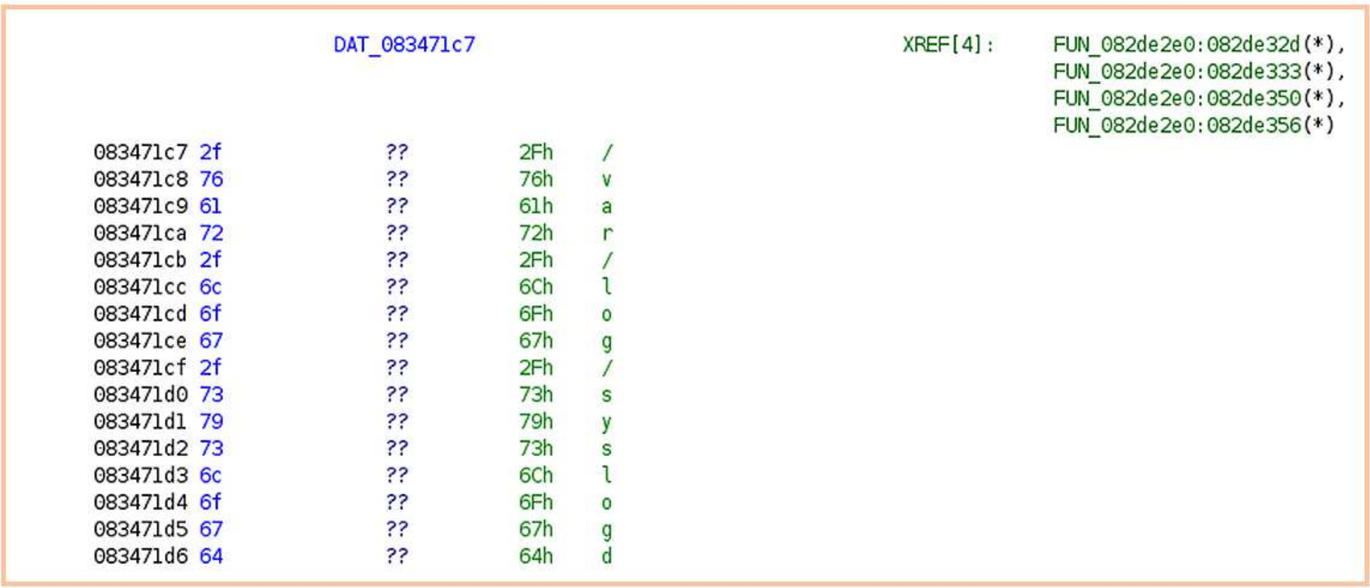
If its process belongs to group 0 (root privilege), then Gomir copies itself to the folder:

```
/var/log/syslogd
```



```
                  01 00 00
         082de325 89 54 24 20     MOV        dword ptr [ESP + local_14],EDX
         082de329 89 4c 24 24     MOV        dword ptr [ESP + local_10],ECX
         082de32d 8d 05 c7        LEA        EAX,[DAT_083471c7]                            = 2Fh     /
                  71 34 08
         082de333 89 44 24 08     MOV        dword ptr [ESP + local_2c],EAX=>DAT_083471c7  = 2Fh     /
         082de337 c7 44 24        MOV        dword ptr [ESP + local_28],0x10
```

Figure 5. GHIDRA - CodeBrowser: the LEA instruction of function 082de32d loads the EAX register with DAT_083471c7.



```
                    DAT_083471c7                          XREF[4]:     FUN_082de2e0:082de32d(*),
                                                                       FUN_082de2e0:082de333(*),
                                                                       FUN_082de2e0:082de350(*),
                                                                       FUN_082de2e0:082de356(*)
         083471c7 2f          ??         2Fh    /
         083471c8 76          ??         76h    v
         083471c9 61          ??         61h    a
         083471ca 72          ??         72h    r
         083471cb 2f          ??         2Fh    /
         083471cc 6c          ??         6Ch    l
         083471cd 6f          ??         6Fh    o
         083471ce 67          ??         67h    g
         083471cf 2f          ??         2Fh    /
         083471d0 73          ??         73h    s
         083471d1 79          ??         79h    y
         083471d2 73          ??         73h    s
         083471d3 6c          ??         6Ch    l
         083471d4 6f          ??         6Fh    o
         083471d5 67          ??         67h    g
         083471d6 64          ??         64h    d
```

Figure 6. GHIDRA - CodeBrowser: DAT_083471c7 corresponds to/var/log/syslogd.

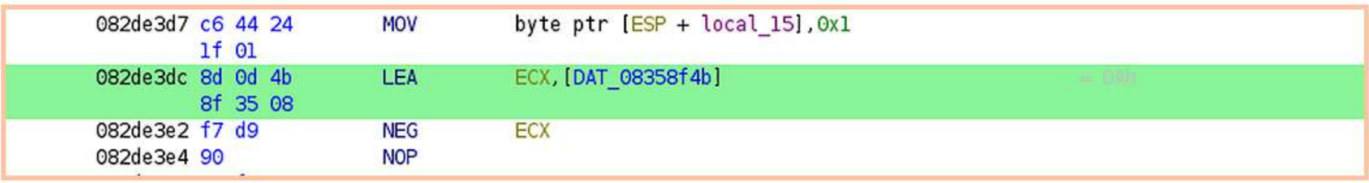
Gomir writes a file to the following folder:

```
/etc/systemd/system/syslogd.service
```

It contains the information below:

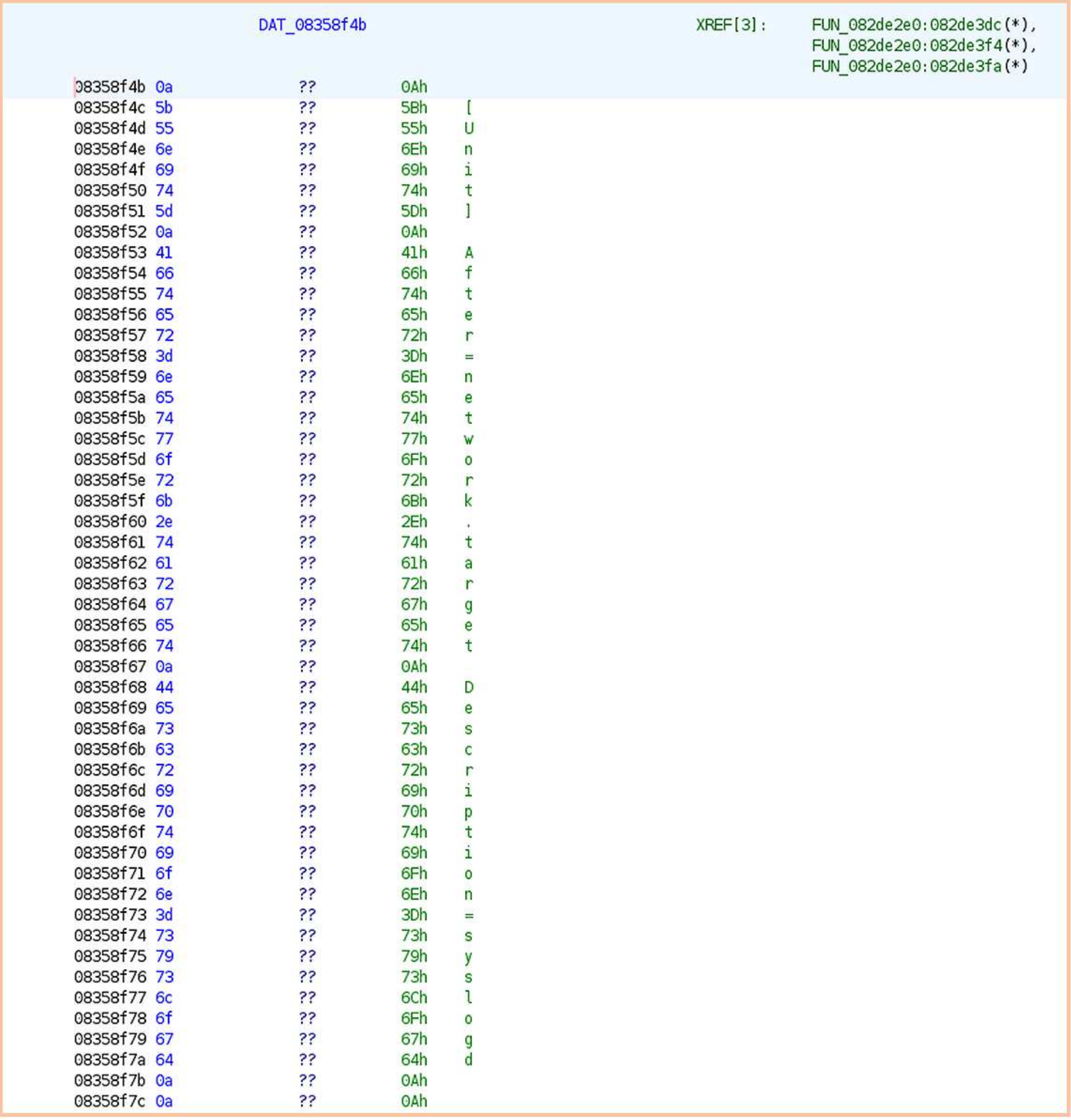
```
[Unit]
After=network.target
Description=syslogd
[Service]
ExecStart=/bin/sh -c "/var/log/syslogd"
Restart=always
[Install]
```

```
WantedBy=multi-user.target
```



```
        082de3d7 c6 44 24        MOV        byte ptr [ESP + local_15],0x1
                 1f 01
        082de3dc 8d 0d 4b        LEA        ECX,[DAT_08358f4b]                        = 0Ah
                 8f 35 08
        082de3e2 f7 d9           NEG        ECX
        082de3e4 90              NOP
```

Figure 7. GHIDRA - CodeBrowser: the LEA instruction of function 082de3dc loads the ECX register with DAT_08358f4b.



```
                          DAT_08358f4b                          XREF[3]:     FUN_082de2e0:082de3dc(*),
                                                                             FUN_082de2e0:082de3f4(*),
                                                                             FUN_082de2e0:082de3fa(*)
        08358f4b 0a              ??         0Ah
        08358f4c 5b              ??         5Bh    [
        08358f4d 55              ??         55h    U
        08358f4e 6e              ??         6Eh    n
        08358f4f 69              ??         69h    i
        08358f50 74              ??         74h    t
        08358f51 5d              ??         5Dh    ]
        08358f52 0a              ??         0Ah
        08358f53 41              ??         41h    A
        08358f54 66              ??         66h    f
        08358f55 74              ??         74h    t
        08358f56 65              ??         65h    e
        08358f57 72              ??         72h    r
        08358f58 3d              ??         3Dh    =
        08358f59 6e              ??         6Eh    n
        08358f5a 65              ??         65h    e
        08358f5b 74              ??         74h    t
        08358f5c 77              ??         77h    w
        08358f5d 6f              ??         6Fh    o
        08358f5e 72              ??         72h    r
        08358f5f 6b              ??         6Bh    k
        08358f60 2e              ??         2Eh    .
        08358f61 74              ??         74h    t
        08358f62 61              ??         61h    a
        08358f63 72              ??         72h    r
        08358f64 67              ??         67h    g
        08358f65 65              ??         65h    e
        08358f66 74              ??         74h    t
        08358f67 0a              ??         0Ah
        08358f68 44              ??         44h    D
        08358f69 65              ??         65h    e
        08358f6a 73              ??         73h    s
        08358f6b 63              ??         63h    c
        08358f6c 72              ??         72h    r
        08358f6d 69              ??         69h    i
        08358f6e 70              ??         70h    p
        08358f6f 74              ??         74h    t
        08358f70 69              ??         69h    i
        08358f71 6f              ??         6Fh    o
        08358f72 6e              ??         6Eh    n
        08358f73 3d              ??         3Dh    =
        08358f74 73              ??         73h    s
        08358f75 79              ??         79h    y
        08358f76 73              ??         73h    s
        08358f77 6c              ??         6Ch    l
        08358f78 6f              ??         6Fh    o
        08358f79 67              ??         67h    g
        08358f7a 64              ??         64h    d
        08358f7b 0a              ??         0Ah
        08358f7c 0a              ??         0Ah
```

Figure 8. GHIDRA - CodeBrowser: DAT_08358f4b corresponds to the content of the syslogd.service artifact.

The **syslogd** file allows you to configure event logging on Linux environments. In order for the changes to be taken into account, Gomir executes the following commands:

```
${SHELL} -c systemctl daemon-reload
${SHELL} -c systemctl reenable syslogd
${SHELL} -c systemctl start syslogd
```

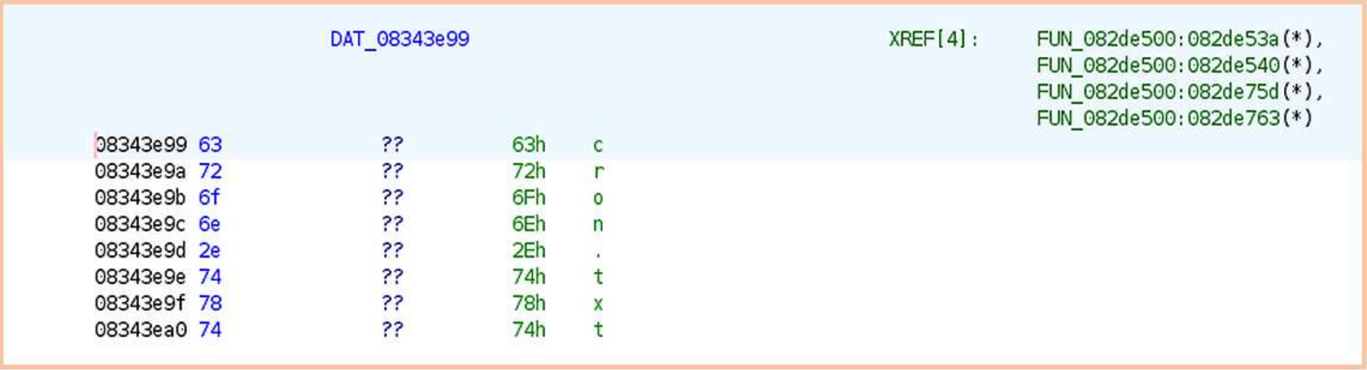
When this service is executed, Gomir interrupts its process and deletes itself.

Installation with persistence via crontab

If its process does not belong to group 0 (root privilege), then Gomir uses crontab to establish its persistence. To do this, the **cron.txt** file is created in the folder where Gomir is present.

The **cron.txt** file contains the following code:

```
@reboot [PROCESS_PATHNAME]
```

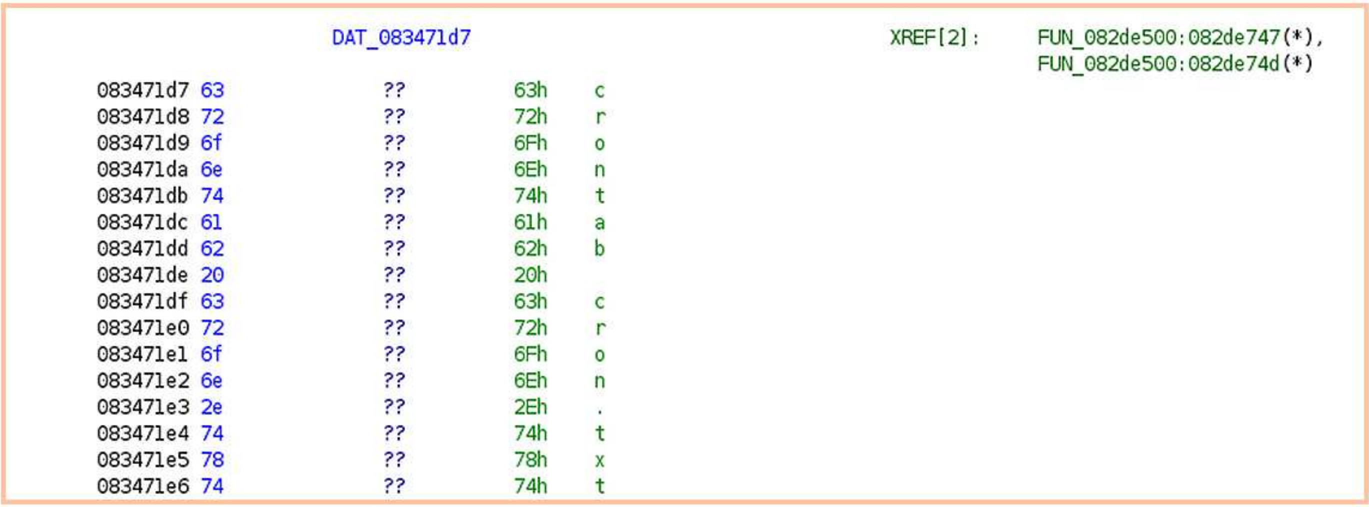


```
                    DAT_08343e99                            XREF[4]:     FUN_082de500:082de53a(*),
                                                                         FUN_082de500:082de540(*),
                                                                         FUN_082de500:082de75d(*),
                                                                         FUN_082de500:082de763(*)
08343e99 63          ??         63h    c
08343e9a 72          ??         72h    r
08343e9b 6f          ??         6Fh    o
08343e9c 6e          ??         6Eh    n
08343e9d 2e          ??         2Eh    .
08343e9e 74          ??         74h    t
08343e9f 78          ??         78h    x
08343ea0 74          ??         74h    t
```

Figure 9. GHIDRA - CodeBrowser: the filename (cron.txt) is hardcoded in DAT_08343e99.

Gomir attempts to list all existing crontab entries, concatenates them into the **cron.txt** file. After loading the new crontab configuration (command below), the file is deleted.

```
/bin/sh -c crontab -l
${SHELL} -c crontab cron.txt
```



```
                    DAT_083471d7                            XREF[2]:     FUN_082de500:082de747(*),
                                                                         FUN_082de500:082de74d(*)
083471d7 63          ??         63h    c
083471d8 72          ??         72h    r
083471d9 6f          ??         6Fh    o
083471da 6e          ??         6Eh    n
083471db 74          ??         74h    t
083471dc 61          ??         61h    a
083471dd 62          ??         62h    b
083471de 20          ??         20h
083471df 63          ??         63h    c
083471e0 72          ??         72h    r
083471e1 6f          ??         6Fh    o
083471e2 6e          ??         6Eh    n
083471e3 2e          ??         2Eh    .
083471e4 74          ??         74h    t
083471e5 78          ??         78h    x
083471e6 74          ??         74h    t
```

Figure 10. GHIDRA - CodeBrowser: the command string (crontab cron.txt) is hardcoded in the data DAT_083471d7.

Generating the Victim ID

When infecting the system, Gomir generates a victim ID via **generate_infection_id**:

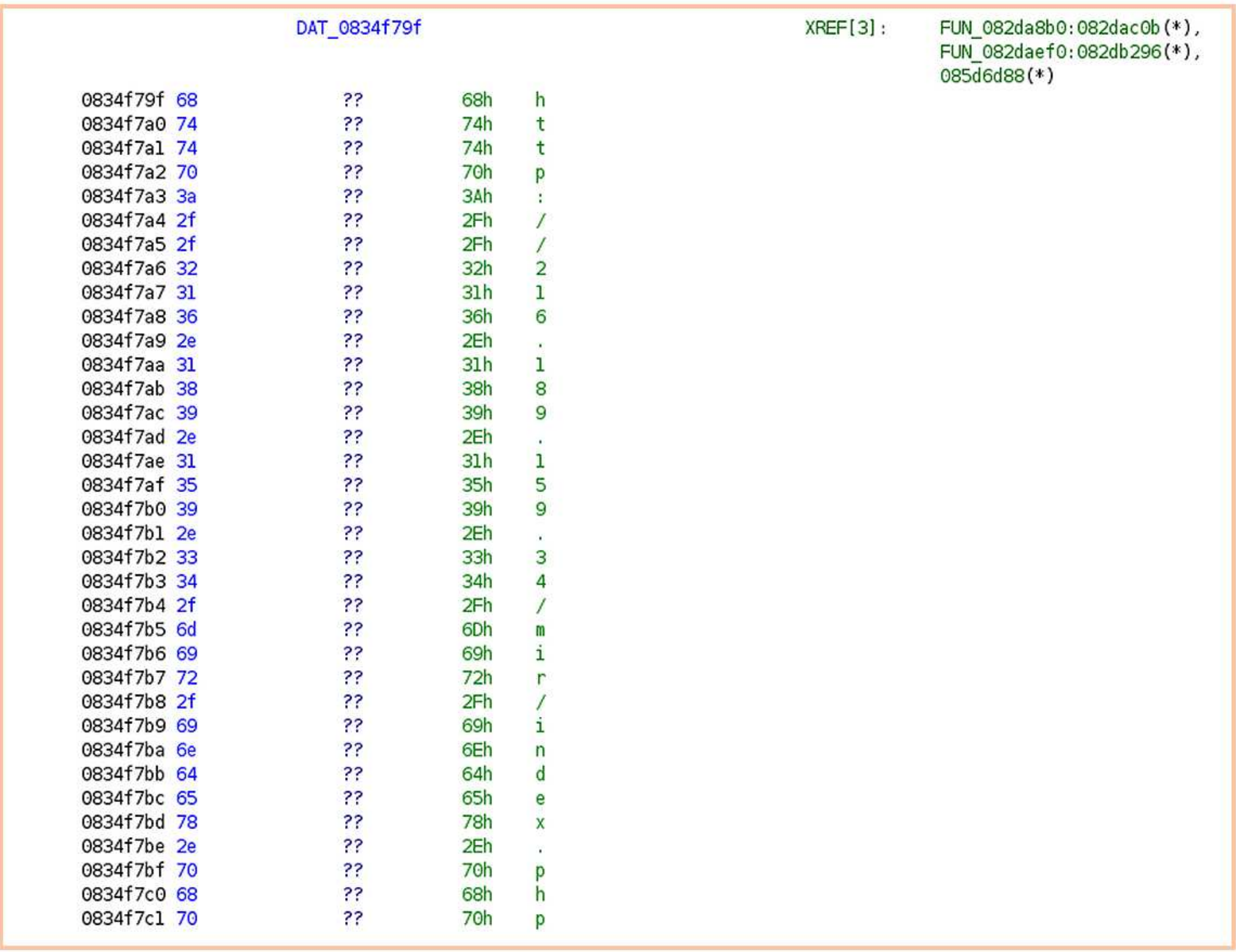
```
def generate_infection_id(hostname, username): hexdigest = hashlib.md5(hostname + username).hexdigest()
return "g-" + hexdigest[:10]
```

This identifier is used when communicating with the CnC server.

Communication with the CnC server

Gomir communicates with a CnC server whose address is hard-coded. Communication is carried out via HTTP requests.

```
http(:)//216(.)189.159.34/mir/index.php
```



```
                        DAT_0834f79f                              XREF[3]:     FUN_082da8b0:082dac0b(*),
                                                                               FUN_082daef0:082db296(*),
                                                                               085d6d88(*)
      0834f79f 68              ??         68h    h
      0834f7a0 74              ??         74h    t
      0834f7a1 74              ??         74h    t
      0834f7a2 70              ??         70h    p
      0834f7a3 3a              ??         3Ah    :
      0834f7a4 2f              ??         2Fh    /
      0834f7a5 2f              ??         2Fh    /
      0834f7a6 32              ??         32h    2
      0834f7a7 31              ??         31h    1
      0834f7a8 36              ??         36h    6
      0834f7a9 2e              ??         2Eh    .
      0834f7aa 31              ??         31h    1
      0834f7ab 38              ??         38h    8
      0834f7ac 39              ??         39h    9
      0834f7ad 2e              ??         2Eh    .
      0834f7ae 31              ??         31h    1
      0834f7af 35              ??         35h    5
      0834f7b0 39              ??         39h    9
      0834f7b1 2e              ??         2Eh    .
      0834f7b2 33              ??         33h    3
      0834f7b3 34              ??         34h    4
      0834f7b4 2f              ??         2Fh    /
      0834f7b5 6d              ??         6Dh    m
      0834f7b6 69              ??         69h    i
      0834f7b7 72              ??         72h    r
      0834f7b8 2f              ??         2Fh    /
      0834f7b9 69              ??         69h    i
      0834f7ba 6e              ??         6Eh    n
      0834f7bb 64              ??         64h    d
      0834f7bc 65              ??         65h    e
      0834f7bd 78              ??         78h    x
      0834f7be 2e              ??         2Eh    .
      0834f7bf 70              ??         70h    p
      0834f7c0 68              ??         68h    h
      0834f7c1 70              ??         70h    p
```

Figure 11. GHIDRA - CodeBrowser: the CnC server address is hardcoded into the virus strain. Location: DAT_0834f79f.

In order to receive new instructions, Gomir sends an HTTP POST request to the CnC server. The request is structured as follows:

```
a\w{9}=2&b\w{9}=[Victim ID]1&c\w{9}=
```

CnC instructions

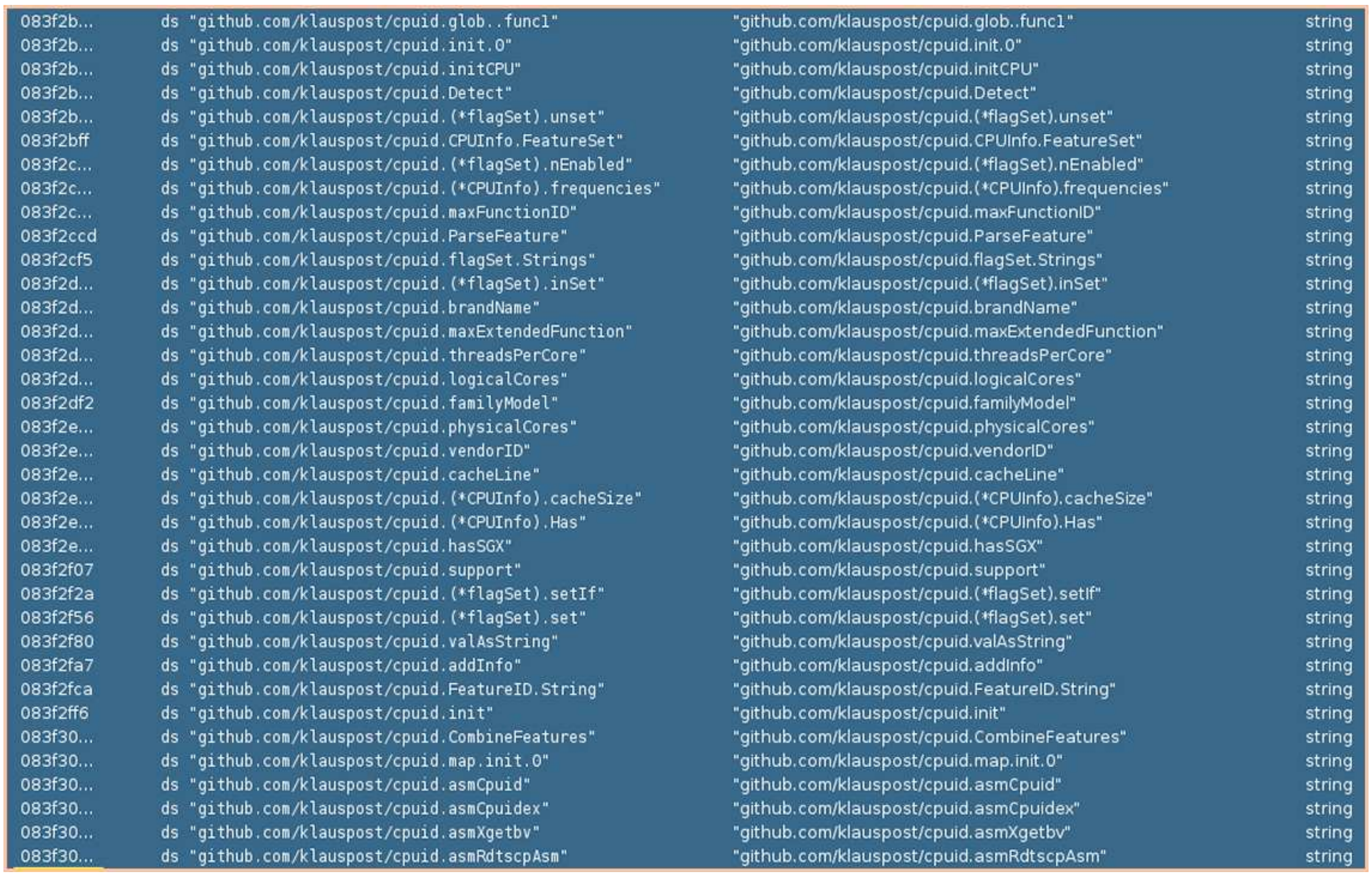
Gomir can receive 17 instructions from the CnC server:

OEPARATION	INSTRUCTIONS
01	Pauses communication with the C&C server for an arbitrary time duration.
02	Executes an arbitrary string as a shell command ("[shell]" "-c" "[arbitrary_string]"). The shell used is specified by the environment variable "SHELL" if present. Otherwise a fallback shell is configured by operation 10 below.
03	Reports about the current working directory.
04	Changes the current working directory and reports pathname of the new working directory.
05	Triggers the arbitrary probing of network endpoints for TCP connectivity.
06	Stops Gomir by terminating its own process.
07	Reports the executable pathname of its own process (the backdoor executable).
08	Collects statistics about an arbitrary directory tree: number of subdirectories - number of files - total size of files.
09	Reports the configuration details of the affected computer: hostname - username - CPU - RAM - network interfaces - listing each interface name - MAC - IP and IPv6 address
10	Configures a fallback shell to use when executing the shell command in operation 02. Initial configuration value is "/bin/sh".
11	Configures a codepage to use when interpreting output from the shell command in operation 02.
12	Pauses communication with the C&C server (arbitrary time).
13	Responds with the hardcoded message "Not implemented on Linux!" .
14	Connects to an arbitrary control endpoint in order to start a reverse proxy. The communication is encrypted (SSL protocol) and uses messages consistent with https(:)//github.com/kost/revsocks.git where the backdoor acts as a proxy client.
15	Reports about the control endpoints of the reverse proxy.
30	Creates an arbitrary file.
31	Exfiltrates an arbitrary file.

Github projects

The attackers seem to be exploiting several elements from different *Github* projects in Gomir.

Project: klauspost/cpuid

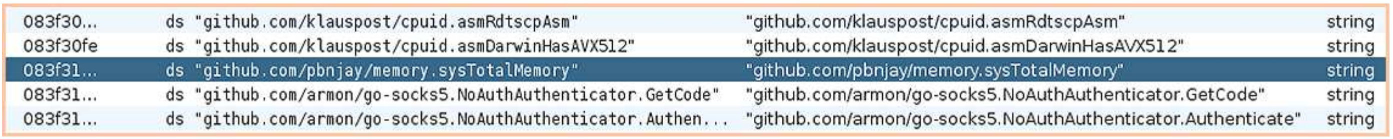


```
083f2b...   ds "github.com/klauspost/cpuid.glob..func1"               "github.com/klauspost/cpuid.glob..func1"               string
083f2b...   ds "github.com/klauspost/cpuid.init.0"                    "github.com/klauspost/cpuid.init.0"                    string
083f2b...   ds "github.com/klauspost/cpuid.initCPU"                   "github.com/klauspost/cpuid.initCPU"                   string
083f2b...   ds "github.com/klauspost/cpuid.Detect"                    "github.com/klauspost/cpuid.Detect"                    string
083f2b...   ds "github.com/klauspost/cpuid.(*flagSet).unset"          "github.com/klauspost/cpuid.(*flagSet).unset"          string
083f2bff    ds "github.com/klauspost/cpuid.CPUInfo.FeatureSet"        "github.com/klauspost/cpuid.CPUInfo.FeatureSet"        string
083f2c...   ds "github.com/klauspost/cpuid.(*flagSet).nEnabled"       "github.com/klauspost/cpuid.(*flagSet).nEnabled"       string
083f2c...   ds "github.com/klauspost/cpuid.(*CPUInfo).frequencies"    "github.com/klauspost/cpuid.(*CPUInfo).frequencies"    string
083f2c...   ds "github.com/klauspost/cpuid.maxFunctionID"             "github.com/klauspost/cpuid.maxFunctionID"             string
083f2ccd    ds "github.com/klauspost/cpuid.ParseFeature"              "github.com/klauspost/cpuid.ParseFeature"              string
083f2cf5    ds "github.com/klauspost/cpuid.flagSet.Strings"           "github.com/klauspost/cpuid.flagSet.Strings"           string
083f2d...   ds "github.com/klauspost/cpuid.(*flagSet).inSet"          "github.com/klauspost/cpuid.(*flagSet).inSet"          string
083f2d...   ds "github.com/klauspost/cpuid.brandName"                 "github.com/klauspost/cpuid.brandName"                 string
083f2d...   ds "github.com/klauspost/cpuid.maxExtendedFunction"       "github.com/klauspost/cpuid.maxExtendedFunction"       string
083f2d...   ds "github.com/klauspost/cpuid.threadsPerCore"            "github.com/klauspost/cpuid.threadsPerCore"            string
083f2d...   ds "github.com/klauspost/cpuid.logicalCores"              "github.com/klauspost/cpuid.logicalCores"              string
083f2df2    ds "github.com/klauspost/cpuid.familyModel"               "github.com/klauspost/cpuid.familyModel"               string
083f2e...   ds "github.com/klauspost/cpuid.physicalCores"             "github.com/klauspost/cpuid.physicalCores"             string
083f2e...   ds "github.com/klauspost/cpuid.vendorID"                  "github.com/klauspost/cpuid.vendorID"                  string
083f2e...   ds "github.com/klauspost/cpuid.cacheLine"                 "github.com/klauspost/cpuid.cacheLine"                 string
083f2e...   ds "github.com/klauspost/cpuid.(*CPUInfo).cacheSize"      "github.com/klauspost/cpuid.(*CPUInfo).cacheSize"      string
083f2e...   ds "github.com/klauspost/cpuid.(*CPUInfo).Has"            "github.com/klauspost/cpuid.(*CPUInfo).Has"            string
083f2e...   ds "github.com/klauspost/cpuid.hasSGX"                    "github.com/klauspost/cpuid.hasSGX"                    string
083f2f07    ds "github.com/klauspost/cpuid.support"                   "github.com/klauspost/cpuid.support"                   string
083f2f2a    ds "github.com/klauspost/cpuid.(*flagSet).setIf"          "github.com/klauspost/cpuid.(*flagSet).setIf"          string
083f2f56    ds "github.com/klauspost/cpuid.(*flagSet).set"            "github.com/klauspost/cpuid.(*flagSet).set"            string
083f2f80    ds "github.com/klauspost/cpuid.valAsString"               "github.com/klauspost/cpuid.valAsString"               string
083f2fa7    ds "github.com/klauspost/cpuid.addInfo"                   "github.com/klauspost/cpuid.addInfo"                   string
083f2fca    ds "github.com/klauspost/cpuid.FeatureID.String"          "github.com/klauspost/cpuid.FeatureID.String"          string
083f2ff6    ds "github.com/klauspost/cpuid.init"                      "github.com/klauspost/cpuid.init"                      string
083f30...   ds "github.com/klauspost/cpuid.CombineFeatures"           "github.com/klauspost/cpuid.CombineFeatures"           string
083f30...   ds "github.com/klauspost/cpuid.map.init.0"                "github.com/klauspost/cpuid.map.init.0"                string
083f30...   ds "github.com/klauspost/cpuid.asmCpuid"                  "github.com/klauspost/cpuid.asmCpuid"                  string
083f30...   ds "github.com/klauspost/cpuid.asmCpuidex"                "github.com/klauspost/cpuid.asmCpuidex"                string
083f30...   ds "github.com/klauspost/cpuid.asmXgetbv"                 "github.com/klauspost/cpuid.asmXgetbv"                 string
083f30...   ds "github.com/klauspost/cpuid.asmRdtscpAsm"              "github.com/klauspost/cpuid.asmRdtscpAsm"              string
```

Figure 12. Identified element.

- Source: https://github.com/klauspost/cpuid
- Use: a Golang library used to retrieve information about the microprocessor.

Project: pbnjay/memory

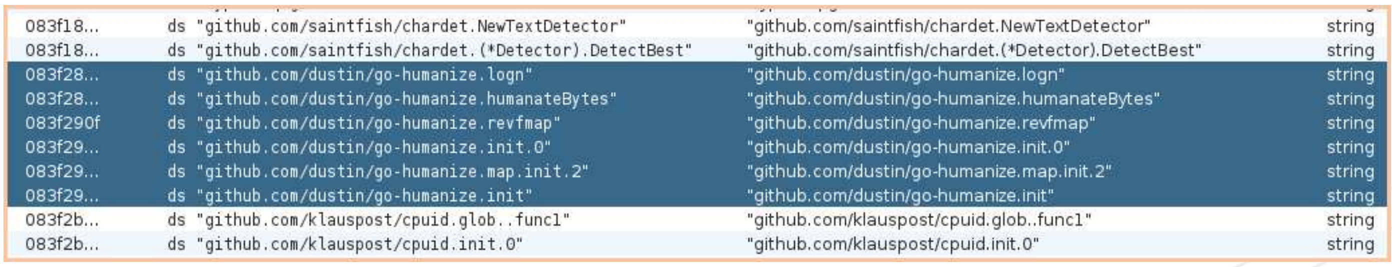


```
083f30...   ds "github.com/klauspost/cpuid.asmRdtscpAsm"                  "github.com/klauspost/cpuid.asmRdtscpAsm"                      string
083f30fe    ds "github.com/klauspost/cpuid.asmDarwinHasAVX512"            "github.com/klauspost/cpuid.asmDarwinHasAVX512"                string
083f31...   ds "github.com/pbnjay/memory.sysTotalMemory"                  "github.com/pbnjay/memory.sysTotalMemory"                      string
083f31...   ds "github.com/armon/go-socks5.NoAuthAuthenticator.GetCode"   "github.com/armon/go-socks5.NoAuthAuthenticator.GetCode"       string
083f31...   ds "github.com/armon/go-socks5.NoAuthAuthenticator.Authen...  "github.com/armon/go-socks5.NoAuthAuthenticator.Authenticate"  string
```

Figure 13. Identified element.

- Source: https://github.com/pbnjay/memory
- Use: a Golang library used to get information about the system's memory.

Project: go-humanize

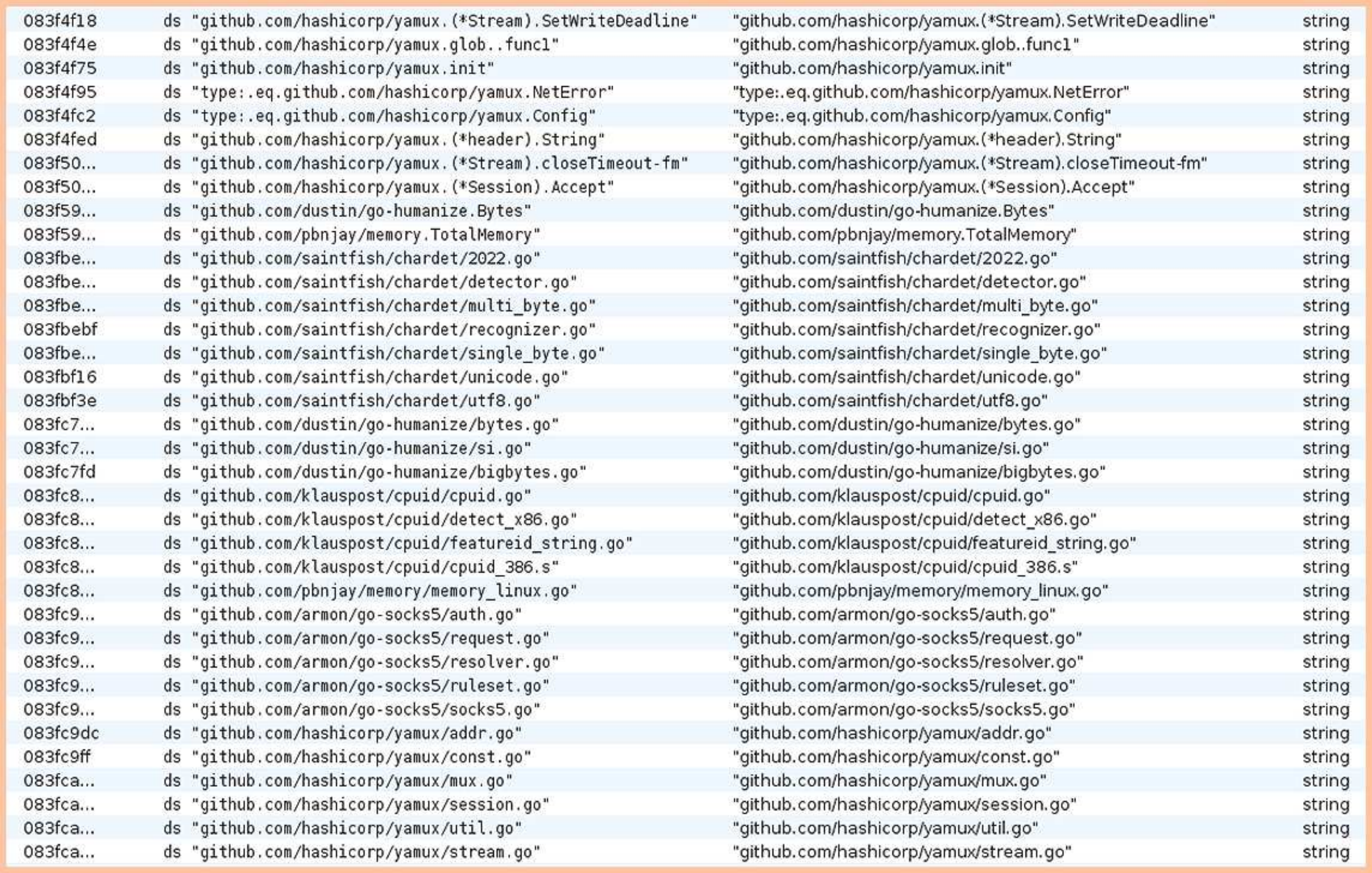


```
083f18...   ds "github.com/saintfish/chardet.NewTextDetector"            "github.com/saintfish/chardet.NewTextDetector"            string
083f18...   ds "github.com/saintfish/chardet.(*Detector).DetectBest"     "github.com/saintfish/chardet.(*Detector).DetectBest"     string
083f28...   ds "github.com/dustin/go-humanize.logn"                      "github.com/dustin/go-humanize.logn"                      string
083f28...   ds "github.com/dustin/go-humanize.humanateBytes"             "github.com/dustin/go-humanize.humanateBytes"             string
083f290f    ds "github.com/dustin/go-humanize.revfmap"                   "github.com/dustin/go-humanize.revfmap"                   string
083f29...   ds "github.com/dustin/go-humanize.init.0"                    "github.com/dustin/go-humanize.init.0"                    string
083f29...   ds "github.com/dustin/go-humanize.map.init.2"                "github.com/dustin/go-humanize.map.init.2"                string
083f29...   ds "github.com/dustin/go-humanize.init"                      "github.com/dustin/go-humanize.init"                      string
083f2b...   ds "github.com/klauspost/cpuid.glob..func1"                  "github.com/klauspost/cpuid.glob..func1"                  string
083f2b...   ds "github.com/klauspost/cpuid.init.0"                       "github.com/klauspost/cpuid.init.0"                       string
```

Figure 14. Identified element.

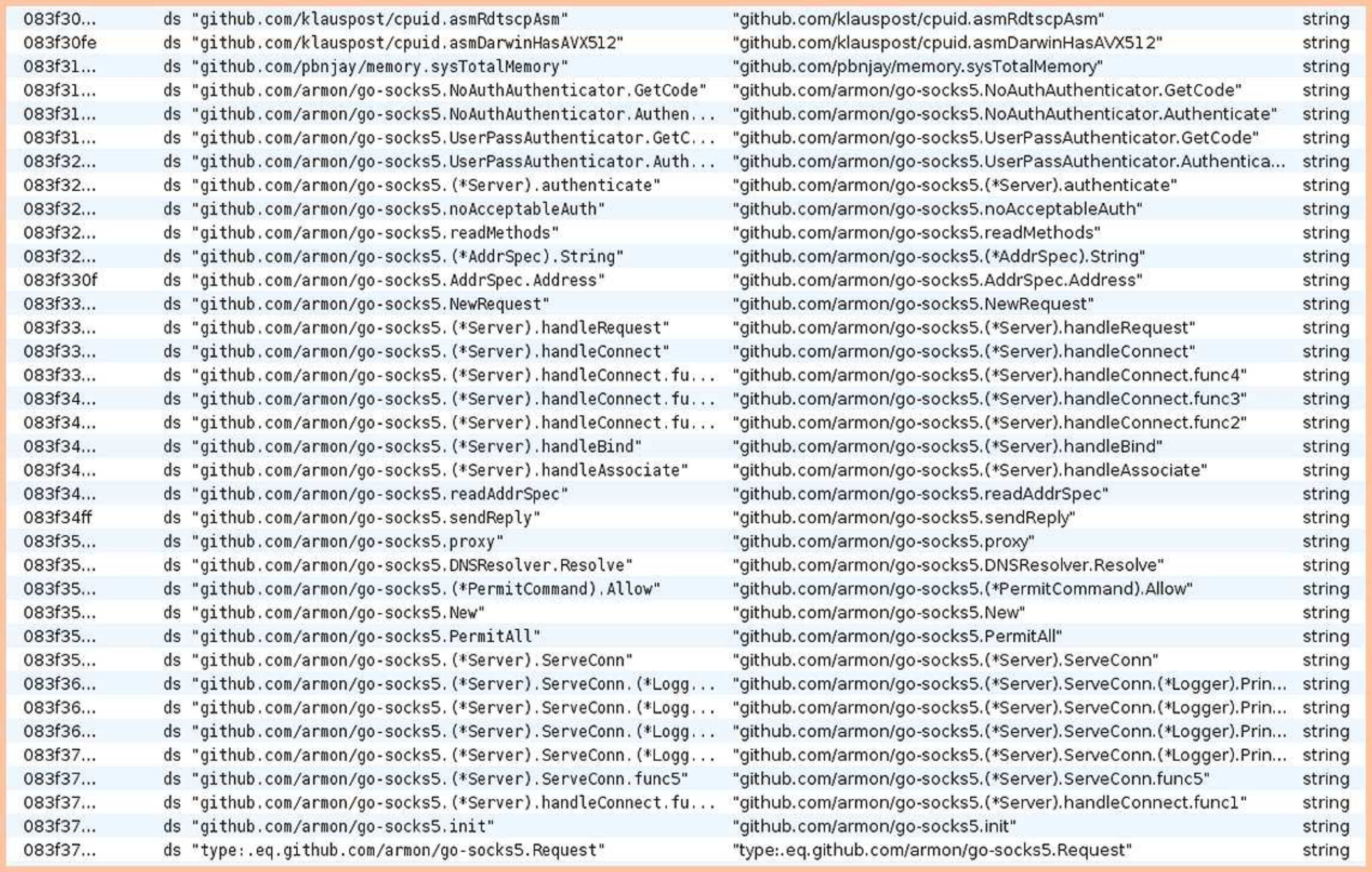
- Source: https://github.com/dustin/go-humanize
- Use: set of functions used to format information on the size of the file system.

Below, the libraries from various github repositories which are integrated in Gomir:



```
083f4f18    ds "github.com/hashicorp/yamux.(*Stream).SetWriteDeadline"    "github.com/hashicorp/yamux.(*Stream).SetWriteDeadline"    string
083f4f4e    ds "github.com/hashicorp/yamux.glob..func1"                   "github.com/hashicorp/yamux.glob..func1"                   string
083f4f75    ds "github.com/hashicorp/yamux.init"                          "github.com/hashicorp/yamux.init"                          string
083f4f95    ds "type:.eq.github.com/hashicorp/yamux.NetError"             "type:.eq.github.com/hashicorp/yamux.NetError"             string
083f4fc2    ds "type:.eq.github.com/hashicorp/yamux.Config"               "type:.eq.github.com/hashicorp/yamux.Config"               string
083f4fed    ds "github.com/hashicorp/yamux.(*header).String"              "github.com/hashicorp/yamux.(*header).String"              string
083f50...   ds "github.com/hashicorp/yamux.(*Stream).closeTimeout-fm"     "github.com/hashicorp/yamux.(*Stream).closeTimeout-fm"     string
083f50...   ds "github.com/hashicorp/yamux.(*Session).Accept"             "github.com/hashicorp/yamux.(*Session).Accept"             string
083f59...   ds "github.com/dustin/go-humanize.Bytes"                      "github.com/dustin/go-humanize.Bytes"                      string
083f59...   ds "github.com/pbnjay/memory.TotalMemory"                     "github.com/pbnjay/memory.TotalMemory"                     string
083fbe...   ds "github.com/saintfish/chardet/2022.go"                     "github.com/saintfish/chardet/2022.go"                     string
083fbe...   ds "github.com/saintfish/chardet/detector.go"                 "github.com/saintfish/chardet/detector.go"                 string
083fbe...   ds "github.com/saintfish/chardet/multi_byte.go"               "github.com/saintfish/chardet/multi_byte.go"               string
083fbebf    ds "github.com/saintfish/chardet/recognizer.go"               "github.com/saintfish/chardet/recognizer.go"               string
083fbe...   ds "github.com/saintfish/chardet/single_byte.go"              "github.com/saintfish/chardet/single_byte.go"              string
083fbf16    ds "github.com/saintfish/chardet/unicode.go"                  "github.com/saintfish/chardet/unicode.go"                  string
083fbf3e    ds "github.com/saintfish/chardet/utf8.go"                     "github.com/saintfish/chardet/utf8.go"                     string
083fc7...   ds "github.com/dustin/go-humanize/bytes.go"                   "github.com/dustin/go-humanize/bytes.go"                   string
083fc7...   ds "github.com/dustin/go-humanize/si.go"                      "github.com/dustin/go-humanize/si.go"                      string
083fc7fd    ds "github.com/dustin/go-humanize/bigbytes.go"                "github.com/dustin/go-humanize/bigbytes.go"                string
083fc8...   ds "github.com/klauspost/cpuid/cpuid.go"                      "github.com/klauspost/cpuid/cpuid.go"                      string
083fc8...   ds "github.com/klauspost/cpuid/detect_x86.go"                 "github.com/klauspost/cpuid/detect_x86.go"                 string
083fc8...   ds "github.com/klauspost/cpuid/featureid_string.go"           "github.com/klauspost/cpuid/featureid_string.go"           string
083fc8...   ds "github.com/klauspost/cpuid/cpuid_386.s"                   "github.com/klauspost/cpuid/cpuid_386.s"                   string
083fc8...   ds "github.com/pbnjay/memory/memory_linux.go"                 "github.com/pbnjay/memory/memory_linux.go"                 string
083fc9...   ds "github.com/armon/go-socks5/auth.go"                       "github.com/armon/go-socks5/auth.go"                       string
083fc9...   ds "github.com/armon/go-socks5/request.go"                    "github.com/armon/go-socks5/request.go"                    string
083fc9...   ds "github.com/armon/go-socks5/resolver.go"                   "github.com/armon/go-socks5/resolver.go"                   string
083fc9...   ds "github.com/armon/go-socks5/ruleset.go"                    "github.com/armon/go-socks5/ruleset.go"                    string
083fc9...   ds "github.com/armon/go-socks5/socks5.go"                     "github.com/armon/go-socks5/socks5.go"                     string
083fc9dc    ds "github.com/hashicorp/yamux/addr.go"                       "github.com/hashicorp/yamux/addr.go"                       string
083fc9ff    ds "github.com/hashicorp/yamux/const.go"                      "github.com/hashicorp/yamux/const.go"                      string
083fca...   ds "github.com/hashicorp/yamux/mux.go"                        "github.com/hashicorp/yamux/mux.go"                        string
083fca...   ds "github.com/hashicorp/yamux/session.go"                    "github.com/hashicorp/yamux/session.go"                    string
083fca...   ds "github.com/hashicorp/yamux/util.go"                       "github.com/hashicorp/yamux/util.go"                       string
083fca...   ds "github.com/hashicorp/yamux/stream.go"                     "github.com/hashicorp/yamux/stream.go"                     string
```

Figure 15. Elements of Github projects used by Gomir: example 1.



```
083f30...   ds "github.com/klauspost/cpuid.asmRdtscpAsm"                  "github.com/klauspost/cpuid.asmRdtscpAsm"                                string
083f30fe    ds "github.com/klauspost/cpuid.asmDarwinHasAVX512"            "github.com/klauspost/cpuid.asmDarwinHasAVX512"                          string
083f31...   ds "github.com/pbnjay/memory.sysTotalMemory"                  "github.com/pbnjay/memory.sysTotalMemory"                                string
083f31...   ds "github.com/armon/go-socks5.NoAuthAuthenticator.GetCode"   "github.com/armon/go-socks5.NoAuthAuthenticator.GetCode"                 string
083f31...   ds "github.com/armon/go-socks5.NoAuthAuthenticator.Authen...  "github.com/armon/go-socks5.NoAuthAuthenticator.Authenticate"            string
083f31...   ds "github.com/armon/go-socks5.UserPassAuthenticator.GetC...  "github.com/armon/go-socks5.UserPassAuthenticator.GetCode"               string
083f32...   ds "github.com/armon/go-socks5.UserPassAuthenticator.Auth...  "github.com/armon/go-socks5.UserPassAuthenticator.Authentica...          string
083f32...   ds "github.com/armon/go-socks5.(*Server).authenticate"        "github.com/armon/go-socks5.(*Server).authenticate"                      string
083f32...   ds "github.com/armon/go-socks5.noAcceptableAuth"              "github.com/armon/go-socks5.noAcceptableAuth"                            string
083f32...   ds "github.com/armon/go-socks5.readMethods"                   "github.com/armon/go-socks5.readMethods"                                 string
083f32...   ds "github.com/armon/go-socks5.(*AddrSpec).String"            "github.com/armon/go-socks5.(*AddrSpec).String"                          string
083f330f    ds "github.com/armon/go-socks5.AddrSpec.Address"              "github.com/armon/go-socks5.AddrSpec.Address"                            string
083f33...   ds "github.com/armon/go-socks5.NewRequest"                    "github.com/armon/go-socks5.NewRequest"                                  string
083f33...   ds "github.com/armon/go-socks5.(*Server).handleRequest"       "github.com/armon/go-socks5.(*Server).handleRequest"                     string
083f33...   ds "github.com/armon/go-socks5.(*Server).handleConnect"       "github.com/armon/go-socks5.(*Server).handleConnect"                     string
083f33...   ds "github.com/armon/go-socks5.(*Server).handleConnect.fu...  "github.com/armon/go-socks5.(*Server).handleConnect.func4"               string
083f34...   ds "github.com/armon/go-socks5.(*Server).handleConnect.fu...  "github.com/armon/go-socks5.(*Server).handleConnect.func3"               string
083f34...   ds "github.com/armon/go-socks5.(*Server).handleConnect.fu...  "github.com/armon/go-socks5.(*Server).handleConnect.func2"               string
083f34...   ds "github.com/armon/go-socks5.(*Server).handleBind"          "github.com/armon/go-socks5.(*Server).handleBind"                        string
083f34...   ds "github.com/armon/go-socks5.(*Server).handleAssociate"     "github.com/armon/go-socks5.(*Server).handleAssociate"                   string
083f34...   ds "github.com/armon/go-socks5.readAddrSpec"                  "github.com/armon/go-socks5.readAddrSpec"                                string
083f34ff    ds "github.com/armon/go-socks5.sendReply"                     "github.com/armon/go-socks5.sendReply"                                   string
083f35...   ds "github.com/armon/go-socks5.proxy"                         "github.com/armon/go-socks5.proxy"                                       string
083f35...   ds "github.com/armon/go-socks5.DNSResolver.Resolve"           "github.com/armon/go-socks5.DNSResolver.Resolve"                         string
083f35...   ds "github.com/armon/go-socks5.(*PermitCommand).Allow"        "github.com/armon/go-socks5.(*PermitCommand).Allow"                      string
083f35...   ds "github.com/armon/go-socks5.New"                           "github.com/armon/go-socks5.New"                                         string
083f35...   ds "github.com/armon/go-socks5.PermitAll"                     "github.com/armon/go-socks5.PermitAll"                                   string
083f35...   ds "github.com/armon/go-socks5.(*Server).ServeConn"           "github.com/armon/go-socks5.(*Server).ServeConn"                         string
083f36...   ds "github.com/armon/go-socks5.(*Server).ServeConn.(*Logg...  "github.com/armon/go-socks5.(*Server).ServeConn.(*Logger).Prin...        string
083f36...   ds "github.com/armon/go-socks5.(*Server).ServeConn.(*Logg...  "github.com/armon/go-socks5.(*Server).ServeConn.(*Logger).Prin...        string
083f36...   ds "github.com/armon/go-socks5.(*Server).ServeConn.(*Logg...  "github.com/armon/go-socks5.(*Server).ServeConn.(*Logger).Prin...        string
083f37...   ds "github.com/armon/go-socks5.(*Server).ServeConn.(*Logg...  "github.com/armon/go-socks5.(*Server).ServeConn.(*Logger).Prin...        string
083f37...   ds "github.com/armon/go-socks5.(*Server).ServeConn.func5"     "github.com/armon/go-socks5.(*Server).ServeConn.func5"                   string
083f37...   ds "github.com/armon/go-socks5.(*Server).handleConnect.fu...  "github.com/armon/go-socks5.(*Server).handleConnect.func1"               string
083f37...   ds "github.com/armon/go-socks5.init"                          "github.com/armon/go-socks5.init"                                        string
083f37...   ds "type:.eq.github.com/armon/go-socks5.Request"              "type:.eq.github.com/armon/go-socks5.Request"                            string
```

Figure 16. Elements of Github project used by Gomir: example 2.

3.5. Virological lineage

3.5.1. Similarities within APT KIMSUKY's arsenal

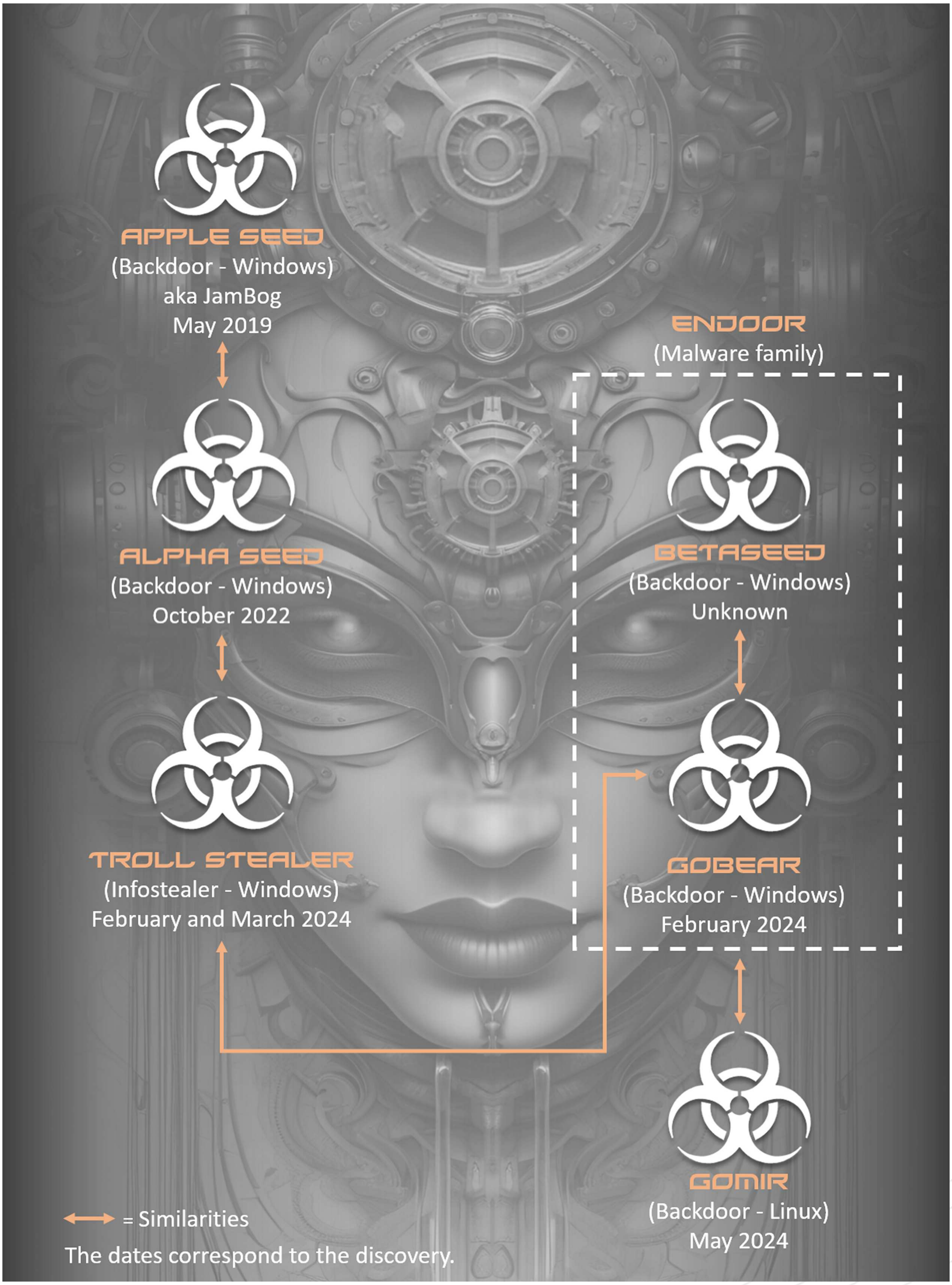


Figure 17. Non-exhaustive infographic summary: APT Kimsuky's arsenal.

Several similarities have been identified between different viral strains belonging to APT Kimsuky's arsenal.

Troll Stealer and Apple Seed

- The location and title of the virus strain are identical. Furthermore, the names of the mutex and several functions are similar.

Troll Stealer and Alpha Seed

- Data encryption and decryption are the same.

Gobear and Gomir

- The two viral strains are almost structurally identical.

Gobear and Troll Stealer

- They have the same certificate *D2innovation Co.,LTD*

Gobear and Betaseed

- Some functions have the same names.

Apple Seed and Alpha Seed

- Alpha Seed is a version of Apple Seed written in Go.

3.5.2. Gomir: deployed by Chalubo in 2023?

Gomir has been the subject of analyses and a few reports that were published during the month of May 2024. The exact date of the emergence of this backdoor is unknown. However, an interesting piece of information was discovered during our open source research: Gomir was allegedly deployed by the Chalubo Trojan during a devastating cyberattack in October 2023.

According to a report published by *Black Lotus Lab* of *Lumen Technologies*, more than 600,000 routers in the United States of America were rendered inoperable during a cyberattack which took place from 25 to 27 October 2023. Unknown attackers used the Chalubo Trojan as a primary infection malware. Additional malwares were deployed on routers to carry out a sabotage operation. Among the additional payloads, one has the SHA256 of Gomir.

The sha256 30584f13c0a9d0c86562c803de350432d5a0607a06b24481ad4d92cdf7288213 corresponds to the Gomir sample deployed during both cyber-attacks that took place in 2023 and 2024.

It is possible that Gomir is **older than it appears** and that its use **is not limited to cyber-espionage only but also to cyber-sabotage**.

Other fingerprints (Gobear and Troll Stealer) were also identified during the cyber-sabotage which took place in October 2023.

The hypothesis that **APT Kimsuky is the author of this sabotage is probable**.

3.6. APT Kimsuky - TTP Evolution

3.6.1. Trojan Droppers with decoy

Since the beginning of the year 2024, APT Kimsuky seems to have added **a new technique to deploy its arsenal**. Attackers create and distribute **Trojan Droppers** via malicious web pages. The Trojans deploy a decoy (legitimate application) and the backdoor payload on the user's system. **No phishing emails** appear to have been used in the distribution of Troll Stealer, Gobear and Gomir during the cyber-espionage campaigns against South Korea.

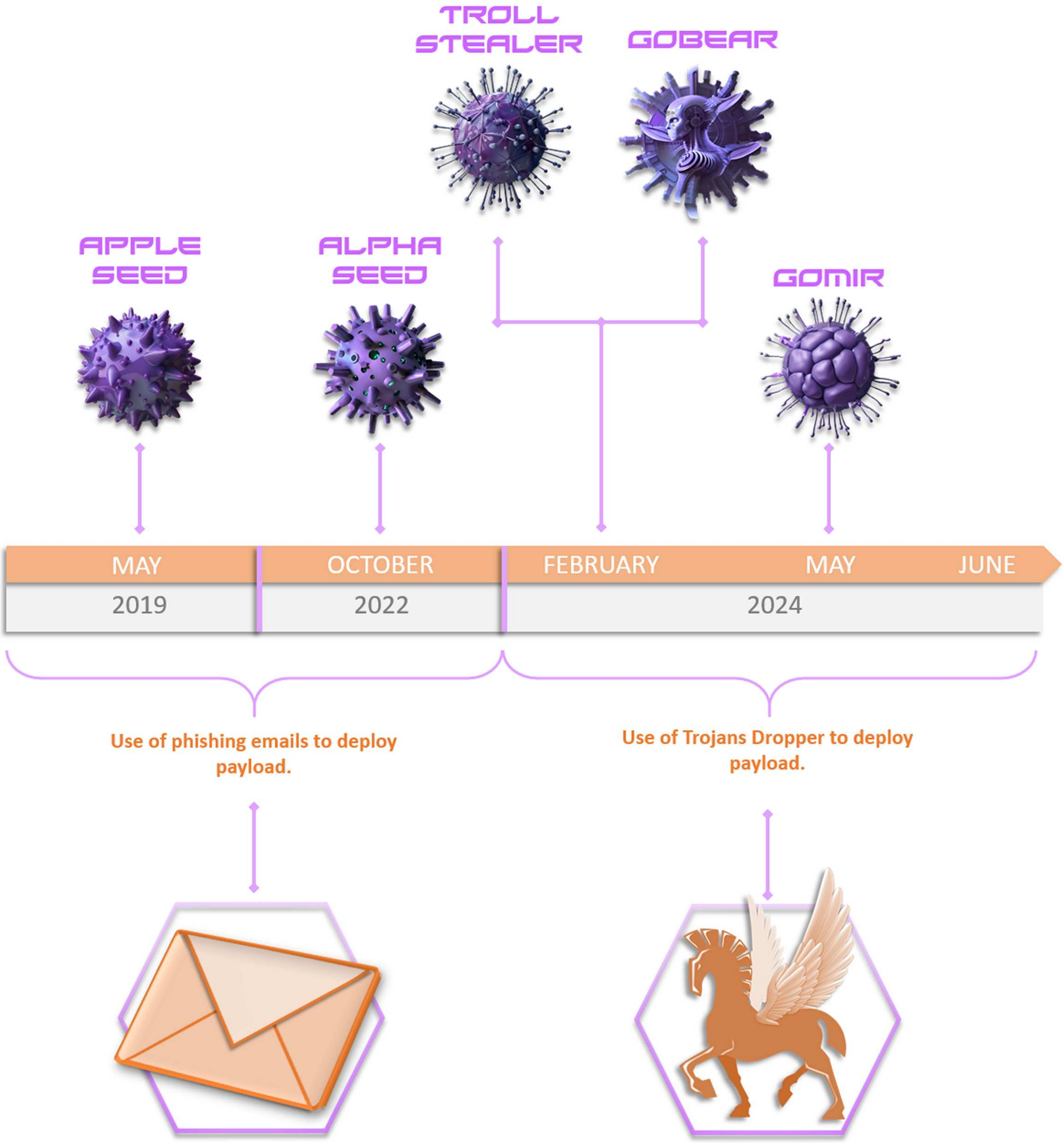

TROLL STEALER
GOBEAR
APPLE SEED
ALPHA SEED
GOMIR
MAY
OCTOBER
FEBRUARY
MAY
JUNE
2019
2022
2024
Use of phishing emails to deploy payload.
Use of Trojans Dropper to deploy payload.

Figure 18. APT Kimsuky : TTP evolution.

3.7. APT Kimsuky - Diamond Model

APT Kimsuky (aka APT 43, TA406, Thallium, Black Banshee, Velvet Chollima…) is a North Korean state-sponsored advanced and persistent threat.

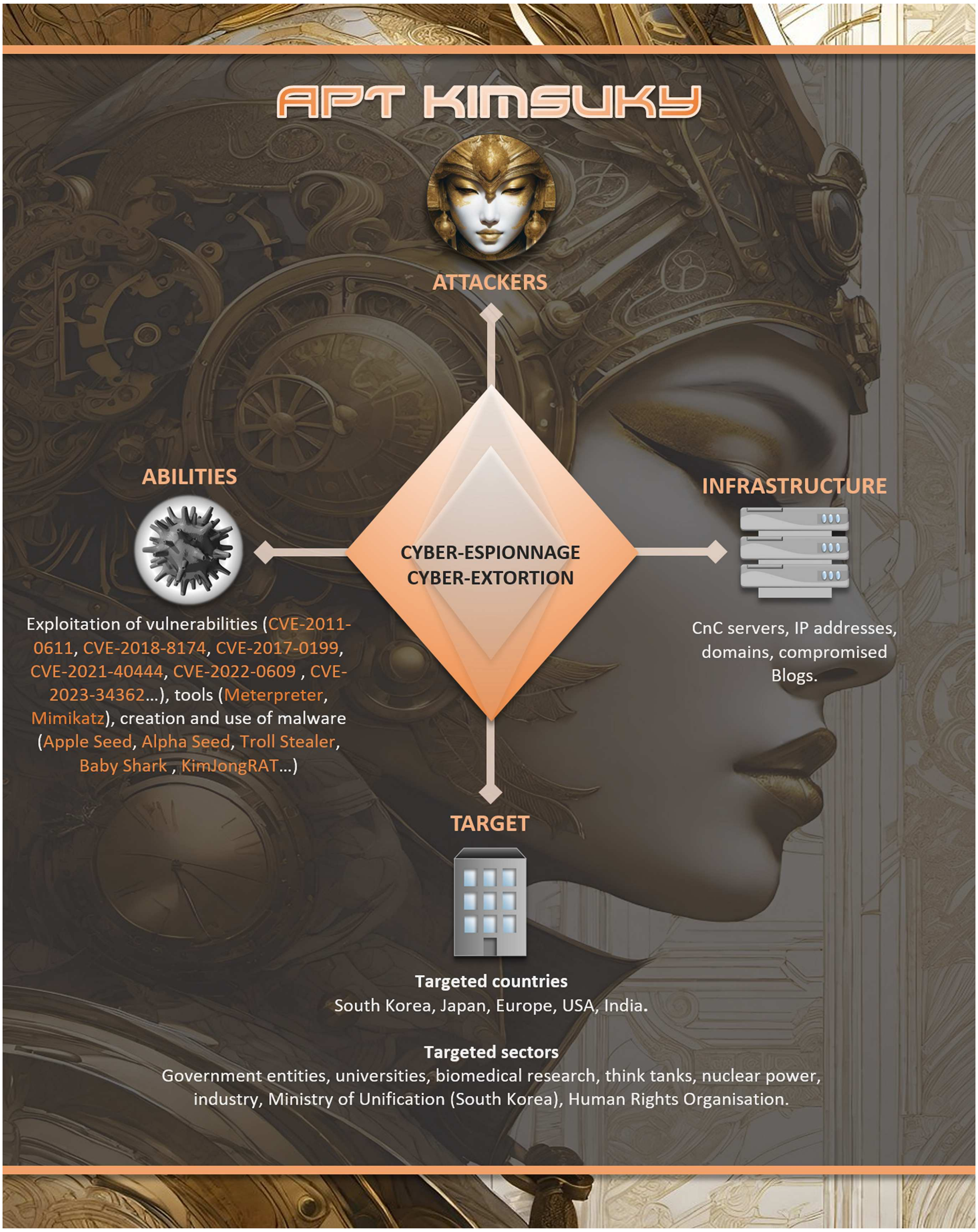
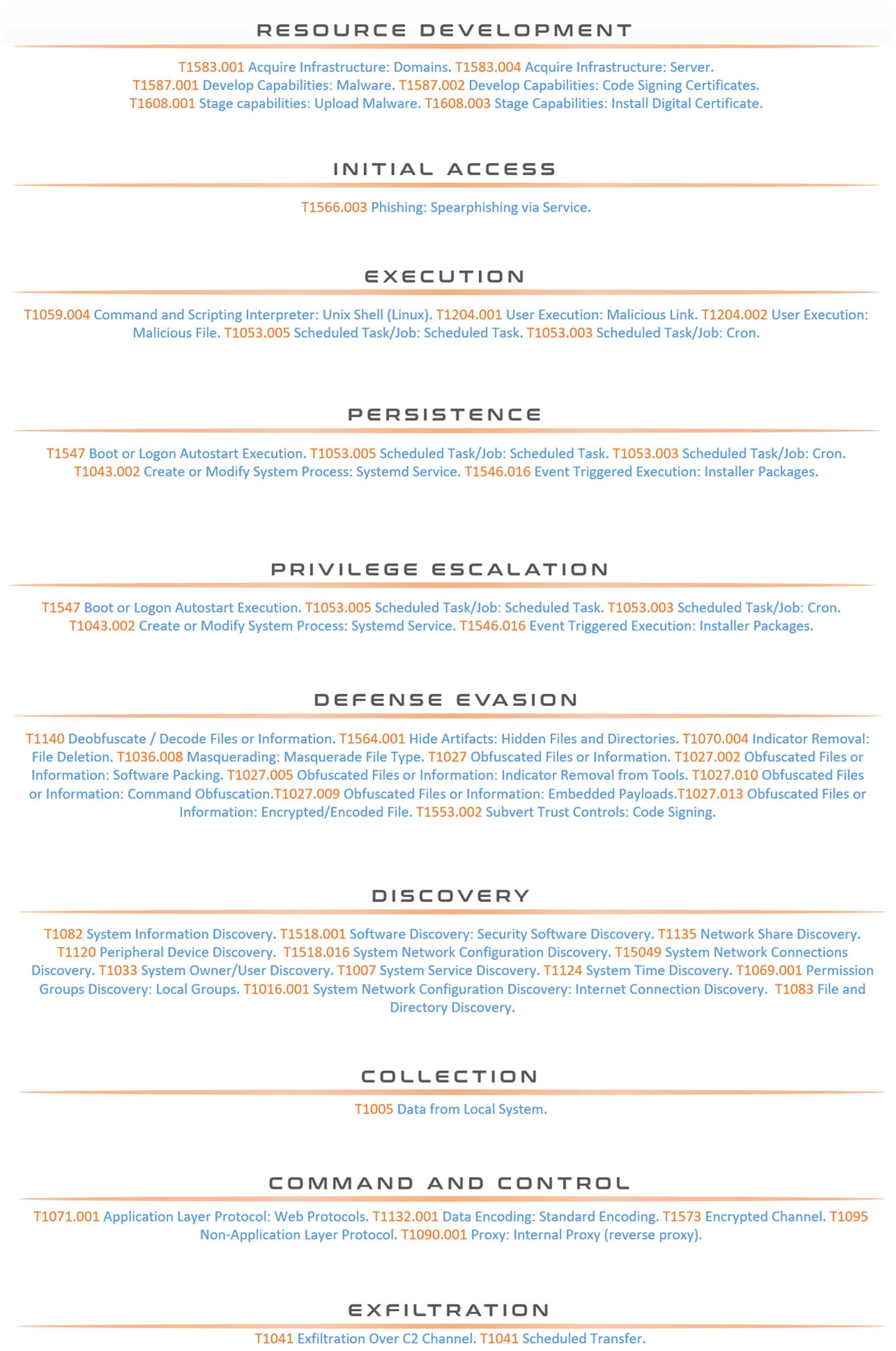

APT KIMSUKY
ATTACKERS
ABILITIES
INFRASTRUCTURE
CYBER-ESPIONNAGE
CYBER-EXTORTION
Exploitation of vulnerabilities (CVE-2011-0611, CVE-2018-8174, CVE-2017-0199, CVE-2021-40444, CVE-2022-0609 , CVE-2023-34362…), tools (Meterpreter, Mimikatz), creation and use of malware (Apple Seed, Alpha Seed, Troll Stealer, Baby Shark , KimJongRAT…)
CnC servers, IP addresses, domains, compromised Blogs.
TARGET
Targeted countries
South Korea, Japan, Europe, USA, India.
Targeted sectors
Government entities, universities, biomedical research, think tanks, nuclear power, industry, Ministry of Unification (South Korea), Human Rights Organisation.

Figure 19. APT Kimsuky Diamond Model.

3.8. MITRE ATT&CK



RESOURCE DEVELOPMENT

T1583.001 Acquire Infrastructure: Domains. T1583.004 Acquire Infrastructure: Server. T1587.001 Develop Capabilities: Malware. T1587.002 Develop Capabilities: Code Signing Certificates. T1608.001 Stage capabilities: Upload Malware. T1608.003 Stage Capabilities: Install Digital Certificate.

INITIAL ACCESS

T1566.003 Phishing: Spearphishing via Service.

EXECUTION

T1059.004 Command and Scripting Interpreter: Unix Shell (Linux). T1204.001 User Execution: Malicious Link. T1204.002 User Execution: Malicious File. T1053.005 Scheduled Task/Job: Scheduled Task. T1053.003 Scheduled Task/Job: Cron.

PERSISTENCE

T1547 Boot or Logon Autostart Execution. T1053.005 Scheduled Task/Job: Scheduled Task. T1053.003 Scheduled Task/Job: Cron. T1043.002 Create or Modify System Process: Systemd Service. T1546.016 Event Triggered Execution: Installer Packages.

PRIVILEGE ESCALATION

T1547 Boot or Logon Autostart Execution. T1053.005 Scheduled Task/Job: Scheduled Task. T1053.003 Scheduled Task/Job: Cron. T1043.002 Create or Modify System Process: Systemd Service. T1546.016 Event Triggered Execution: Installer Packages.

DEFENSE EVASION

T1140 Deobfuscate / Decode Files or Information. T1564.001 Hide Artifacts: Hidden Files and Directories. T1070.004 Indicator Removal: File Deletion. T1036.008 Masquerading: Masquerade File Type. T1027 Obfuscated Files or Information. T1027.002 Obfuscated Files or Information: Software Packing. T1027.005 Obfuscated Files or Information: Indicator Removal from Tools. T1027.010 Obfuscated Files or Information: Command Obfuscation.T1027.009 Obfuscated Files or Information: Embedded Payloads.T1027.013 Obfuscated Files or Information: Encrypted/Encoded File. T1553.002 Subvert Trust Controls: Code Signing.

DISCOVERY

T1082 System Information Discovery. T1518.001 Software Discovery: Security Software Discovery. T1135 Network Share Discovery. T1120 Peripheral Device Discovery. T1518.016 System Network Configuration Discovery. T15049 System Network Connections Discovery. T1033 System Owner/User Discovery. T1007 System Service Discovery. T1124 System Time Discovery. T1069.001 Permission Groups Discovery: Local Groups. T1016.001 System Network Configuration Discovery: Internet Connection Discovery. T1083 File and Directory Discovery.

COLLECTION

T1005 Data from Local System.

COMMAND AND CONTROL

T1071.001 Application Layer Protocol: Web Protocols. T1132.001 Data Encoding: Standard Encoding. T1573 Encrypted Channel. T1095 Non-Application Layer Protocol. T1090.001 Proxy: Internal Proxy (reverse proxy).

EXFILTRATION

T1041 Exfiltration Over C2 Channel. T1041 Scheduled Transfer.

Figure 20. TTPs linked to GOMIR (APT KIMSUKY)

3.9. IOCs

GOMIR

TLP	TYPE	VALEUR	COMMENTAIRE
TLP:CLEAR	SHA256	30584f13c0a9d0c86562c803de350432d5a0607a06b24481ad4d92cdf7288213	GOMIR (Souche virale)
TLP:CLEAR	SHA1	93edc15a20aac8b5193e5b22e35dbb09848e2ca0	GOMIR (Souche virale)
TLP:CLEAR	MD5	e562cf30d17d47347c7e6ffd249fc190	GOMIR (Souche virale)
TLP:CLEAR	IP	216(.)189.159.34	C2 GOMIR

GOBEAR

TLP	TYPE	VALEUR	COMMENTAIRE
TLP:CLEAR	SHA256	7BD723B5E4F7B3C645AC04E763DFC913060EAF6E136EECC4EE0653AD2056F3A0	Trojan Dropper GOBEAR
TLP:CLEAR	SHA1	1DD417D7373DF9B8F5B76E7EB8FE87B7C37F0CC8	Trojan Dropper GOBEAR
TLP:CLEAR	MD5	B74EFD8470206A20175D723C14C2E872	Trojan Dropper GOBEAR

TROLL STEALER

TLP	TYPE	VALEUR	COMMENTAIRE
TLP:CLEAR	SHA256	d7f3ecd8939ae8b170b641448ff12ade2163baad05ca6595547f8794b5ad013b	Troll Stealer (Souche virale)
TLP:CLEAR	SHA256	36ea1b317b46c55ed01dd860131a7f6a216de71958520d7d558711e13693c9dc	Troll Stealer (Souche virale)
TLP:CLEAR	MD5	19c2decfa7271fa30e48d4750c1d18c1	Trojan Dropper NX_PRNMANS.EXE
TLP:CLEAR	SHA1	e6be97ca9e79b45c671c6531908f70b353d47994	Trojan Dropper NX_PRNMANS.EXE
TLP:CLEAR	SHA256	6eebb5ed0d0b5553e40a7b1ad739589709d077aab4cbea1c64713c48ce9c96f9	Trojan Dropper NX_PRNMANS.EXE
TLP:CLEAR	MD5	7b6d02a459fdaa4caa1a5bf741c4bd42	Trojan Dropper NXTPKIENT.exe
TLP:CLEAR	SHA1	4eea45c22881a092ac7a8b0a5379076d5803e83e	Trojan Dropper NXTPKIENT.exe
TLP:CLEAR	SHA256	f8ab78e1db3a3cc3793f7680a90dc1d8ce087226ef59950b7acd6bb1beffd6e3	Trojan Dropper NXTPKIENT.exe
TLP:CLEAR	MD5	27ef6917fe32685fdf9b755eb8e97565	Trojan Dropper XOWizmxM6U.exe
TLP:CLEAR	SHA1	6d531b021b20febf1dafa730582944eb82d9c6f3	Trojan Dropper XOWizmxM6U.exe
TLP:CLEAR	SHA256	2e0ffaab995f22b7684052e53b8c64b9283b5e81503b88664785fe6d6569a55e	Trojan Dropper XOWizmxM6U.exe
TLP:CLEAR	MD5	7457dc037c4a5f3713d9243a0dfb1a2c	Troll Stealer (Souche virale)
TLP:CLEAR	SHA1	4c8b7d968806f8108ccde6ac07a37b8174ac44bf	Troll Stealer (Souche virale)
TLP:CLEAR	SHA256	ff3718ae6bd59ad479e375c602a81811718dfb2669c2d1de497f02baf7b4adca	Troll Stealer (Souche virale)
TLP:CLEAR	MD5	c8e7b0d3b6afa22e801cacaf16b37355	Troll Stealer (Souche virale)
TLP:CLEAR	SHA256	955cb4f01eb18f0d259fcb962e36a339e8fe082963dfd9f72d3851210f7d2d3b	Troll Stealer (Souche virale)
TLP:CLEAR	MD5	88f183304b99c897aacfa321d58e1840	Troll Stealer (Souche virale)

TLP	TYPE	VALEUR	COMMENTAIRE
TLP:CLEAR	SHA256	bc4c1c869a03045e0b594a258ec3801369b0dcabac193e90f0a684900e9a582d	Troll Stealer (Souche virale)
TLP:CLEAR	URL	hxxp(:)//ai.kostin.p-e(.)kr/index.php	
TLP:CLEAR	URL	hxxp(:)//ar.kostin.p-e(.)kr/index.php	
TLP:CLEAR	URL	hxxp(:)//ai.negapa.p-e(.)kr/index.php	
TLP:CLEAR	URL	hxxp(:)//ol.negapa.p-e(.)kr/index.php	
TLP:CLEAR	URL	hxxp(:)//ai.limsjo.p-e(.)kr/index.php	
TLP:CLEAR	URL	hxxp(:)//qi.limsjo.p-e(.)kr/index.php	
TLP:CLEAR	URL	hxxp(:)//coolsystem(.)co.kr/admin/mail/index.php	
TLP:CLEAR	Domaine	ai.kostin.p-e(.)kr	
TLP:CLEAR	Domaine	ar.kostin.p-e(.)kr	
TLP:CLEAR	Domaine	ai.negapa.p-e(.)kr	
TLP:CLEAR	Domaine	ol.negapa.p-e(.)kr	
TLP:CLEAR	Domaine	ai.limsjo.p-e(.)kr	
TLP:CLEAR	Domaine	qi.limsjo.p-e(.)kr	
TLP:CLEAR	IP	216.189.159(.)197	C2 TROLL STEALER

3.10. YARA

3.10.1. YARA 1

YARA - ShadowStackre
Source : https://www.shadowstackre.com/analysis/gomir

```
rule GomirBackdoor {
meta:
        description = "Rule to detect Gomir Backdoor"
        author = "ShadowStackRe.com"
        date = "2024-05-22"
        Rule_Version = "v1"
        malware_type = "backdoor"
        malware_family = "gomir"
        License = "MIT License, https://opensource.org/license/mit/"
        Hash = "30584f13c0a9d0c86562c803de350432d5a0607a06b24481ad4d92cdf7288213"
strings:
        $strCronText = "cron.txt"
        $strHttpResPathMIR = "mir/"
        $strSystemDSvc = "syslogd.service"
        $strSocksList = "Socks list"
        $strCmdPath = "CmdPath:"
        $strCodePage = "Codepage:"
        $strNextConnTime = "Next Connection Time:"
        $strTCPOpenedIndicator = {
        C7 44 24 29 5B 2B 5D 20
        C7 44 24 2C 20 4F 70 65
        C7 44 24 30 6E 65 64 2E
        }
condition:
        all of them and filesize < 6MB
}
```

3.10.2. YARA 2

YARA - aDvens

```
rule GOMIR_Specific_strings {
meta:
author = "aDvens-CTI"
source = "aDvens"
status = "RELEASED"
sharing = "TLP:CLEAR"
malware = "GOMIR"
description = "Yara_rule_that_detects_GOMIR_Backdoor_June_2024."
info = "GOMIR_Backdoor_malware_used_by_APT_KIMSUKY"
strings:
$GOMIR_string1 = "cron.txt"
$GOMIR_string2 = "/var/log/syslogd"
$GOMIR_string3 = "216.189.159.34"
condition:
$GOMIR_string1 and $GOMIR_string2 and $GOMIR_string3
}
```

4. Olympic Games 2024: Analysis of the AcidPour threat

On 26 July 2024, France inaugurates the Olympic Games with a Parade of Nations on the Seine, expected to welcome 600,000 spectators. This event is seen as an opportunity for France to shine on the world stage, to invoke a peaceful truce and the spirit of brotherhood of Pierre de Coubertin.

The result in public discourse is a media consensus advocating a strict separation between geopolitics and sport. However, the modern Olympic Games, since their rehabilitation in 1894, have served as a showcase for nations to display their power and transmit political messages. The Olympic ideal must not obscure the threats weighing on the organisation of the Games and France. It is essential to deconstruct the idea that political games have no place in sports games and to remain aware of the cyber threats that hover over the Paris 2024 edition.

4.1. Geopolitics in sports

In the modern version of the Olympic Games, the competition no longer pits Greeks against each other within a single sanctuary, but different nations, often antagonistic, who welcome them successively. The modern Olympic Games will then naturally follow the way of thinking of people and societies, especially as sport takes on increasing importance in domestic life and public space.

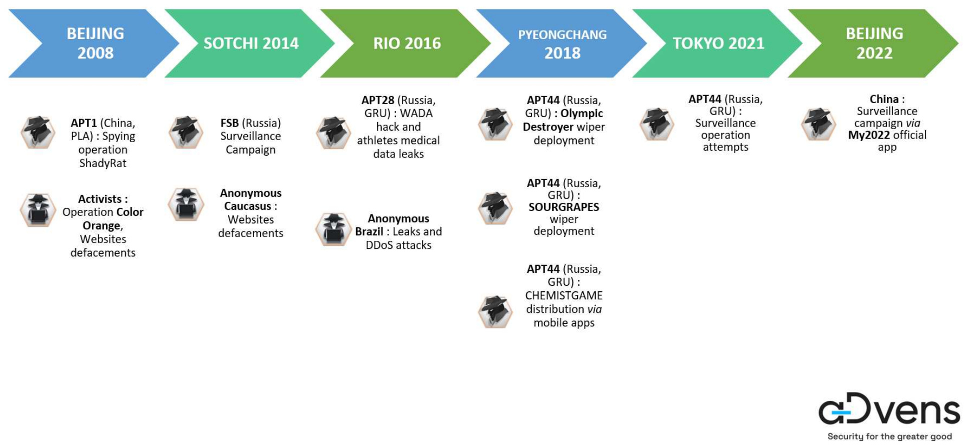
Over the last century, 4 editions stand out and show the inseparable link between global sporting competition and geopolitics:

- The Berlin Games of 1936, there is no need to remind the setting and the context,
- The 1980 Moscow Games, boycotted by the Western Bloc countries,
- The following Games, Los Angeles in 1984, boycotted by the Soviet Bloc countries,
- More recently, the 2008 Beijing Games allowed China to publicly display, particularly during its grandiose opening ceremony, its return to the world stage as an economic and political giant, a competitor to the United States of America.

In 2022 and 2023, the events of the Football World Cup in Qatar, or the ban on the participation of Russian and Belarusian athletes in the Paris 2024 games, demonstrates that sport can be both a driving force and a pretext for political issues.

4.2. Cyber campaigns from previous editions

Consequently, it is expected that *cyber* threats will be present this summer, as a vector for relaying the political ambitions of the participating countries, or those not officially participating for that matter. Since it became an efficient tool for nations, and a credible threat, there has not been an Olympic Games without cyber attacks since Beijing in 2008, with various reasons in mind: destabilisation, sabotage, espionage or greed.


BEIJING 2008
APT1 (China, PLA) : Spying operation ShadyRat
Activists : Operation Color Orange, Websites defacements
SOTCHI 2014
FSB (Russia) Surveillance Campaign
Anonymous Caucasus : Websites defacements
RIO 2016
APT28 (Russia, GRU) : WADA hack and athletes medical data leaks
Anonymous Brazil : Leaks and DDoS attacks
PYEONGCHANG 2018
APT44 (Russia, GRU) : Olympic Destroyer wiper deployment
APT44 (Russia, GRU) : SOURGRAPES wiper deployment
APT44 (Russia, GRU) : CHEMISTGAME distribution via mobile apps
TOKYO 2021
APT44 (Russia, GRU) : Surveillance operation attempts
BEIJING 2022
China : Surveillance campaign via My2022 official app
aDvens
Security for the greater good

4.3. Presentation of the threat

Any important event, frequently with media coverage, is both a cause and a pretext for cyber attacks. The Paris Olympic Games are potentially the subject of attack campaigns, whatever the purpose, and the pretext of initial access vectors, such as mass phishing emails.

Preventing this scenario is extremely complex due to the muny actors involved: Operators of Vital Importance (OIVs), infrastructures, competition sites, local authorities that are hosting events, partner companies, media relays, subcontractors, etc. ..

To add to this complexity, the current state global geopolitics is tense, with several ongoing armed operations and conflicts, on which France maintains official positions. These draw different stakeholders, hostile, unfriendly, opportunistic, active or in ambush.

4.3.1. Global Context

- **Ukrainian war**: France has firmly positioned itself in the conflict between Ukraine and Russia. The announcements of potentially sending French troops to the front and ban of Russian and Belarusian athletes, being "unwelcome", have led to very significant tensions between France and Russia, as well as some explicit threats from certain decision-makers in Moscow.

Antagonist	*Russian Federation*	Threat	*Very high*

- **Pacific situation**: China pursues its ambition every day to compete with the United States on economic, military and diplomatic aspects. Even though France remains in the background in this conflict between powers, China fiercely watches over speeches in France concerning human rights, the Uighurs, and especially Taiwan. France does not recognise the sovereignty of the island of "Chinese Taipei", however language inaccuracies or certain media positions during the event may arouse the ire and reprisals of Beijing, as was the case during the Tokyo Olympic Games in 2022. Finally, if China is not accustomed to vast campaigns of destabilisation, it is on the other hand an opportunistic and almost systematic actor in matters of espionage.

Antagonist	*People's Republic of China*	Threat	*High*

- **Middle Eastern conflict**: Hamas' attacks against Israeli civilians in October 2023, and the Israeli response in the Gaza Strip caused a stir worldwide. The recent involvement of Iran in April 2024 has further complicated the confrontation. France maintained a neutral position by calling for a truce, and was little exposed to declarations of reprisals.

Antagonist	*Arab nationalist or Muslim activist groups*	Threat	*Moderate*

4.3.2. Cybercrime

As the different groups making up the underground landscape of cybercrime are all in competition with each other, a global event is an opportunity not to be missed in order to build or strengthen one's reputation. In addition, these private and lucrative groups are possess an opportunistic mentality and the Games can be used as the subject and theme for many phishing emails and decoy documents.

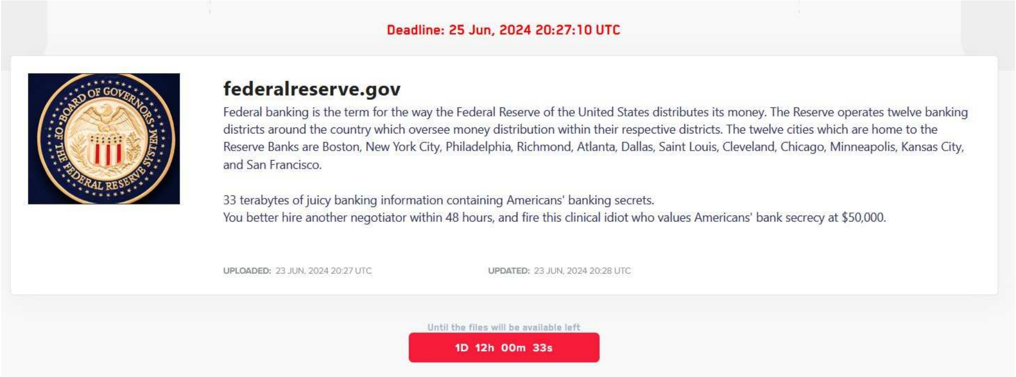
It should be observed that in terms of impact, these cartels represent the first threat to all sectors combined. Their operators are competent, mature, sophisticated and rely on effective infrastructures and proven methods. Finally, the majority of these criminal groups come from the Russian Federation. The targeting of their victims, oriented towards the rest of Asia and the West, is part of an informal agreement with the Russian executive, and therefore meets the interests of Moscow. In addition, their residence beeing within the borders of the Russian Federation protects them from police operations and possible arrests.

Lockbit

The case of the Lockbit group is a good example. On 19 February 2024, the international police operation Cronos dismantled a part of the ransomware group's infrastructure that was the most active since 2022. The publications on Lockbit's showcase website by the police were a very serious blow to the group's reputation, and therefore to its economic model. However, its founder LockbitSupp, a Russian resident, and the main developers of the brand were not worried about the operation. Although the seizure of the infrastructure brought activity to a sudden halt, the product Lockbit3.0 nevertheless remains a reference brand for many cybercriminal affiliates. After a phase of silence and disorganisation, the ransomware has returned in force and aggression. Major attacks have hit French victims:

- 04/30/2024: Cannes Hospital Centre,
- 06/05/2024: Ile-de-France Green Spaces Agency.

At the time of this writing, Lockbit felt confident enough to claim, on 24 June 2024, the exfiltration of 33TB of banking data from the Federal Reserve of the USA. These were published on 26 June, and ultimately turned out to belong to Evolve Bank and Trust, which received a cease and desist order from the Federal Bank of the United States. France, whose National Gendarmerie participated in operation Cronos, could be heavily targeted by the group during the period of the Olympic Games.



Deadline: 25 Jun, 2024 20:27:10 UTC

federalreserve.gov

Federal banking is the term for the way the Federal Reserve of the United States distributes its money. The Reserve operates twelve banking districts around the country which oversee money distribution within their respective districts. The twelve cities which are home to the Reserve Banks are Boston, New York City, Philadelphia, Richmond, Atlanta, Dallas, Saint Louis, Cleveland, Chicago, Minneapolis, Kansas City, and San Francisco.

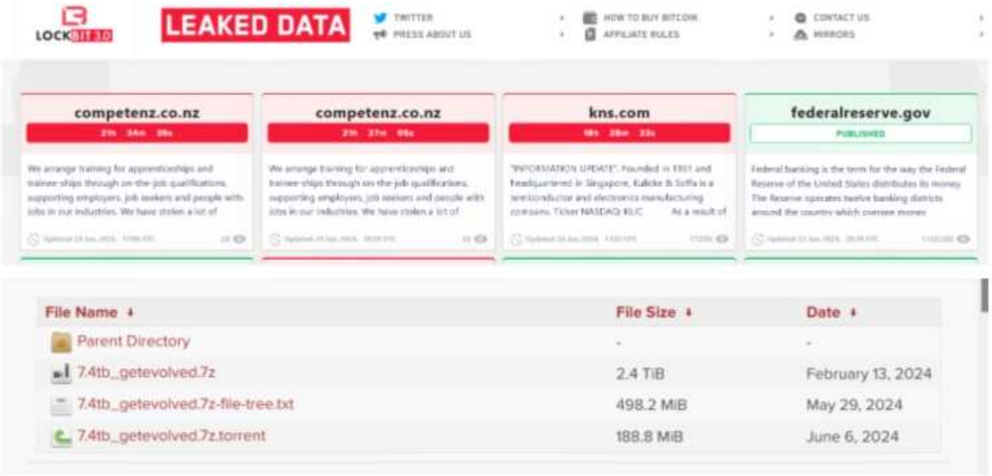
33 terabytes of juicy banking information containing Americans' banking secrets.
You better hire another negotiator within 48 hours, and fire this clinical idiot who values Americans' bank secrecy at $50,000.

UPLOADED: 23 JUN, 2024 20:27 UTC   UPDATED: 23 JUN, 2024 20:28 UTC

Until the files will be available left

1D 12h 00m 33s

Figure 21. Source: Lockbit.



competenz.co.nz

We arrange training for apprenticeships and trainee-ships through on-the-job qualifications, supporting employers, job seekers and people with jobs in our industries. We have stolen a lot of

competenz.co.nz

We arrange training for apprenticeships and trainee-ships through on-the-job qualifications, supporting employers, job seekers and people with jobs in our industries. We have stolen a lot of

kns.com

"INFORMATION UPDATE". Founded in 1951 and headquartered in Singapore, Kulicke & Soffa is a semiconductor and electronics manufacturing company. Ticker NASDAQ: KLIC As a result of

federalreserve.gov

PUBLISHED

Federal banking is the term for the way the Federal Reserve of the United States distributes its money. The Reserve operates twelve banking districts around the country which oversee money

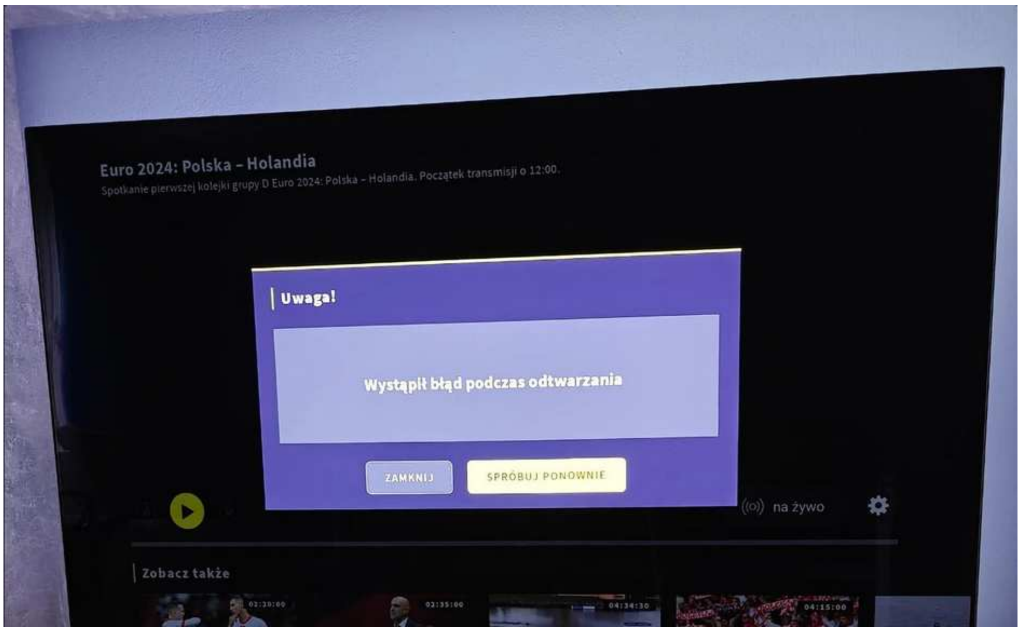
File Name ↓	File Size ↓	Date ↓
Parent Directory	-	-
7.4tb_getevolved.7z	2.4 TiB	February 13, 2024
7.4tb_getevolved.7z-file-tree.txt	498.2 MiB	May 29, 2024
7.4tb_getevolved.7z.torrent	188.8 MiB	June 6, 2024

- **Targets:** Large cities, medium and small municipalities, ministries and public institutions, tourism operating companies, partner companies, private actors from all sectors.

4.3.3. Hacktivism

Hacktivist groups operate for political reasons, often nationalist or religious in the case of groups targeting France. Their modus operandi is to cause DDoS attacks against websites or certain platforms. Although the material impact is zero, the temporary inaccessibility of online resources and the media coverage of these attacks are significant, potentially having a significant psychological impact on populations.

This attack scope could be even greater in the case of the Paris 2024 Games with, for example, the targeting of television channels, video-on-demand platforms and online ticketing. On 19 June, the Polish channel TV Spot suffered an attack during the broadcast of Poland's match against the Netherlands, depriving spectators of the first half. Poland officially blamed Russia for the attack.



Euro 2024: Polska – Holandia

Spotkanie pierwszej kolejki grupy D Euro 2024: Polska – Holandia. Początek transmisji o 12:00.

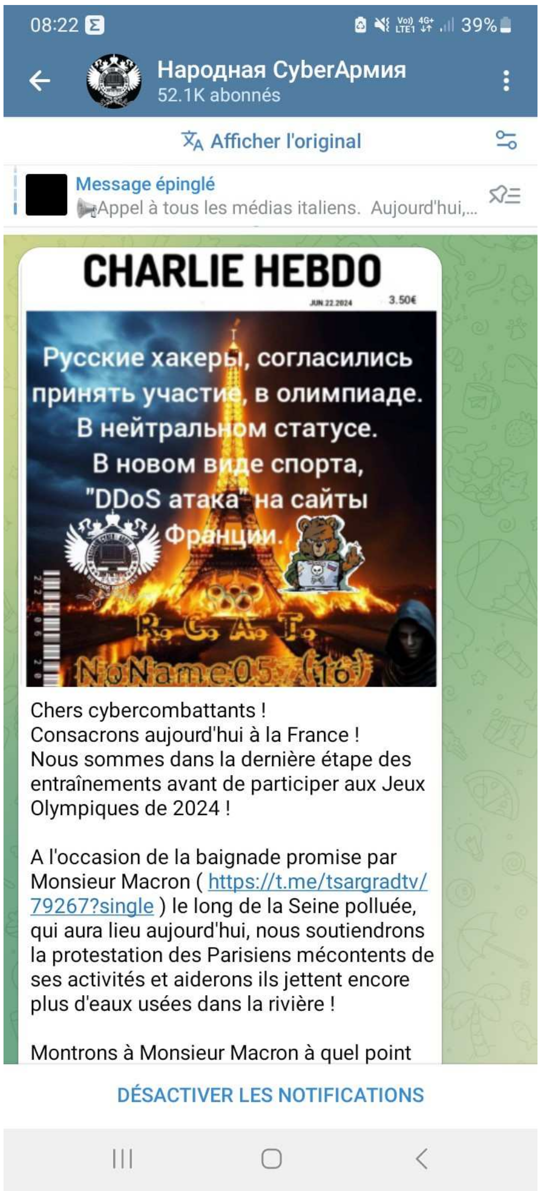
Uwaga!

Wystąpił błąd podczas odtwarzania

Figure 22. Source: Natemat.pl.

NoName057(16)

France has regularly been targeted by pro-Russian groups since the military offensive in Ukraine in 2022. The latest attack was orchestrated by the group NoName057(16) and targeted around fifteen government websites on 15 June 2024. This collective is very active and is a main actor with the development of the DDoSia project, a distributed denial of service (DDoS) attack toolbox, usable by any affiliate.


08:22
39%
CHARLIE HEBDO
JUN 22 2024
3.50€
Русские хакеры, согласились принять участие, в олимпиаде. В нейтральном статусе. В новом виде спорта, "DDoS атака" на сайты Франции.
R.C.A.T.
NoName057(16)
DÉSACTIVER LES NOTIFICATIONS

Народная CyberАрмия
52.1K abonnés

Afficher l'original

Message épinglé
Appel à tous les médias italiens. Aujourd'hui,...

Chers cybercombattants !
Consacrons aujourd'hui à la France !
Nous sommes dans la dernière étape des entraînements avant de participer aux Jeux Olympiques de 2024 !

A l'occasion de la baignade promise par Monsieur Macron (https://t.me/tsargradtv/79267?single) le long de la Seine polluée, qui aura lieu aujourd'hui, nous soutiendrons la protestation des Parisiens mécontents de ses activités et aiderons ils jettent encore plus d'eaux usées dans la rivière !

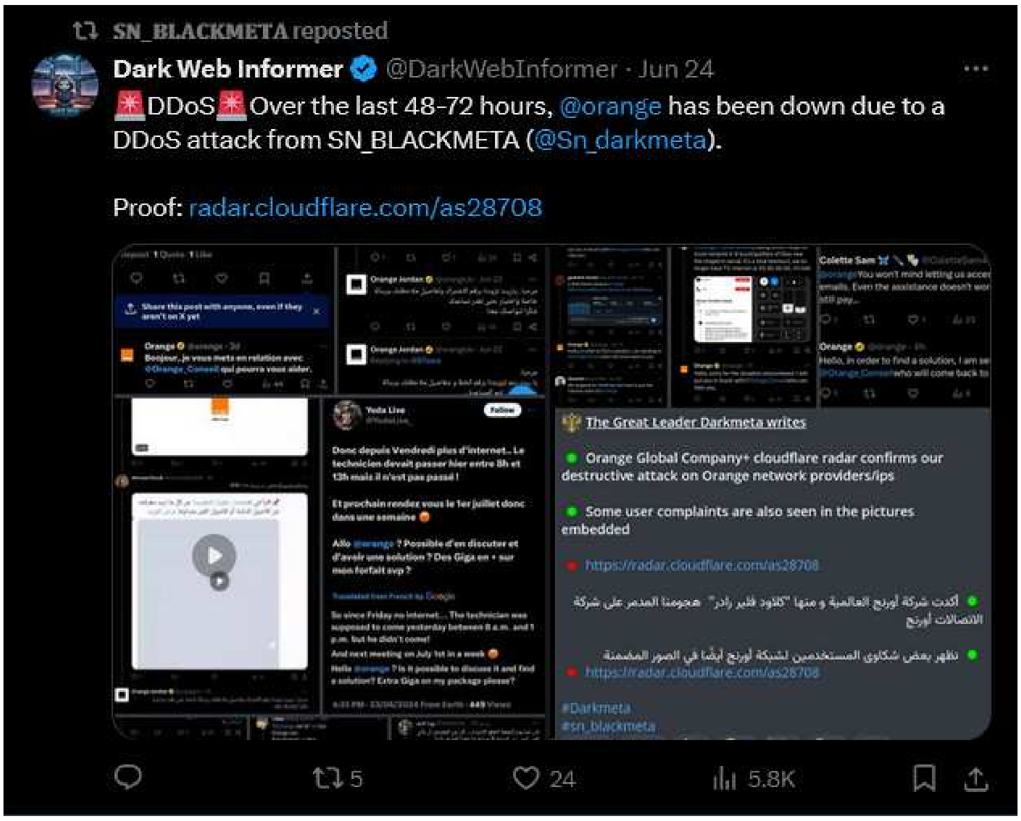
Montrons à Monsieur Macron à quel point

Figure 23. Source: CyberArmyofRussia_Reborn.

Other groups with a pro-Palestine beliefs have targeted France in recent months, for example:

- LulzSec Muslims: collective inspired by Killnet,
- Türk Hack Team: pro-Turkey group coordinating attacks against countries sympathetic to the Kurdish cause.

In the Middle East, the intensity of the conflict has not drastically decreased. In France, where the State of Palestine is not officially recognised, this conflict is followed through the political and media coverage. It would be a suprise if these latter groups take part in destabilising France during the Olympic Games as a result.



SN_BLACKMETA reposted

Dark Web Informer @DarkWebInformer · Jun 24

🚨DDoS🚨Over the last 48-72 hours, @orange has been down due to a DDoS attack from SN_BLACKMETA (@Sn_darkmeta).

Proof: radar.cloudflare.com/as28708

Yoda Live

Donc depuis Vendredi plus d'internet… Le technicien devait passer hier entre 8h et 13h mais il n'est pas passé !

Et prochain rendez vous le 1er juillet donc dans une semaine

Allo @orange ? Possible d'en discuter et d'avoir une solution ? Des Giga en + sur mon forfait svp ?

Translated from French by Google

So since Friday no internet… The technician was supposed to come yesterday between 8 a.m. and 1 p.m. but he didn't come!
And next meeting on July 1st in a week
Hello @orange ? Is it possible to discuss it and find a solution? Extra Giga on my package please?

The Great Leader Darkmeta writes

🟢 Orange Global Company+ cloudflare radar confirms our destructive attack on Orange network providers/ips

🟢 Some user complaints are also seen in the pictures embedded

🔴 https://radar.cloudflare.com/as28708

🟢 أكدت شركة أورنج العالمية و منها "كلاود فلير رادر" هجومنا المدمر على شركة الاتصالات أورنج

🟢 تظهر بعض شكاوى المستخدمين لشبكة أورنج أيضًا في الصور المضمنة
🔴 https://radar.cloudflare.com/as28708

#Darkmeta
#sn_blackmeta

Figure 24. Source: DDoS attack against Orange on 24/06/2024 by the SN_BLACKMETA group (source: X).

- **Targets:** Public services, Olympic ticket offices and committees, television channels and video-on-demand services.

4.3.4. Disruptions and sabotage

Russia

Russia maintains a constant cyber pressure on its adversaries on the global stage. The various APTs attached to the Russian intelligence apparatus are capable of carrying out acts of war, destruction, espionage, surveillance and destabilisation. France's CERT-FR recalled in a report on 19 June 2024 the recurring attack attempts by the NOBELIUM group, affiliated with the Russian foreign intelligence services (SVR), against French Ministries, notably the Ministries of Culture and Foreign Affairs.

Olympic Destroyer

In the case of the Olympic Games, Russia has already distinguished itself with the use of wipers on host infrastructures. The opening ceremony of the 2018 Pyeongchang Games in South Korea was disrupted by a wiper, which would later be named Olympic Destroyer. This took the official games website offline, deactivated the stadium's WiFi network, the video surveillance system, as well as several drones used for capturing images.

The malware was distributed by spear phishing emails and deployed two tools dedicated to stealing passwords stored in browsers and those used on the system. The malware destroys backups and shadow copies kept by the system, and disables the Windows Recovery Tool. Olympic Destroyer then removes its traces, and deactivates all Windows-related services before turning off the machine, which can then no longer restart.

The military campaign in Ukraine that began in 2022 is notable for the massive and unprecedented use of numerous *wipers* against Ukrainian infrastructure: WhisperGate, HermecticWiper, HermeticRansom (false *ransomware*), AcidRain, IsaacWiper, DesertBlade, CaddyWiper, DoubleZero, ArguePatch , Industroyer 2, Prestige (fake *ransomware*), NikoWiper, Somnia, RansomBoggs (fake *ransomware*), sDelete, AWFULSHRED, BidSwipe, SwiftSlicer. Russia has therefore been able to take advantage of its conflict in Ukraine to improve their capabilities concerning this type malware since Pyeongchang in 2018.

- **Targets:** Infrastructure.

4.4. AcidPour Analysis

4.4.1. AcidRain, the big brother

One of the feared scenarios in the context of these Olympic Games is the reuse of one of these weapons of destruction, or a variant, capable of paralysing and annihilating infrastructures (physical or network).

Among these wipers, AcidPour was identified in March 2024 by security researchers from SentinelOne. The latter is itself a variant of the infamous AcidRain, used on 24 February 2022, the day of the offensive in Ukraine. The attack targeted the KA-SAT network of the operator Viasat (Eutelsat) and affected the communications of several thousand customers in Ukraine but also in Europe. Among the collateral damage was the loss of remote access to 5,800 wind turbines in Germany. The attackers had used an access to the Skylogic VPN, before lateralising and executing legitimate commands on SurfBeam modems. These destructive commands overwrote the data in the flash memory of these modems.

If the *malware* is run with a *root* account, the disk devices (/dev/sdX, /dev/loopX,/dev/block/mtdblockX,/dev/block/mmcblkX.) are erased. Memory devices '/dev/mtdX' are erased via the *MEMWRITEOOB* and ioctl utilities. At the end of these deletions, a restart of the device is triggered.

4.4.2. AcidPour

AcidPour, first uploaded on 16 March 2024 in Ukraine, shares some similarities, such as the paths targeted on infected machines, and 30% of the code with AcidRain. This proximity is evident in the restart mechanism, its directory wiping logic and the clearing mechanism based on the IOCTL function used by both AcidRain and the VPNFilter “dstr” plugin. On the other hand, if AcidRain can target Linux systems with the MIPS architecture, AcidPour can now target Linux systems with an x86 architecture, in addition to its new embed features.

Among these new features, AcidPour expands the scope of targeted devices to include *Unsorted Block Image* (UBI) and *Device Mapper* (DM) processes.

AcidRain supports the following devices:

- **/dev/sd**:* A generic block device,
- **/dev/mtdblock**:* Flash memory (common in routers and IoT devices),
- **/dev/block/mtdblock**:* Another potential way to access flash memory,
- **/dev/mtd**:* The device file for flash memory that supports file operations,
- **/dev/mmcblk**:* For SD/MMC cards,
- **/dev/block/mmcblk**:* Another potential way to access SD/MMC cards,
- **/dev/loop**:* Virtual block devices.

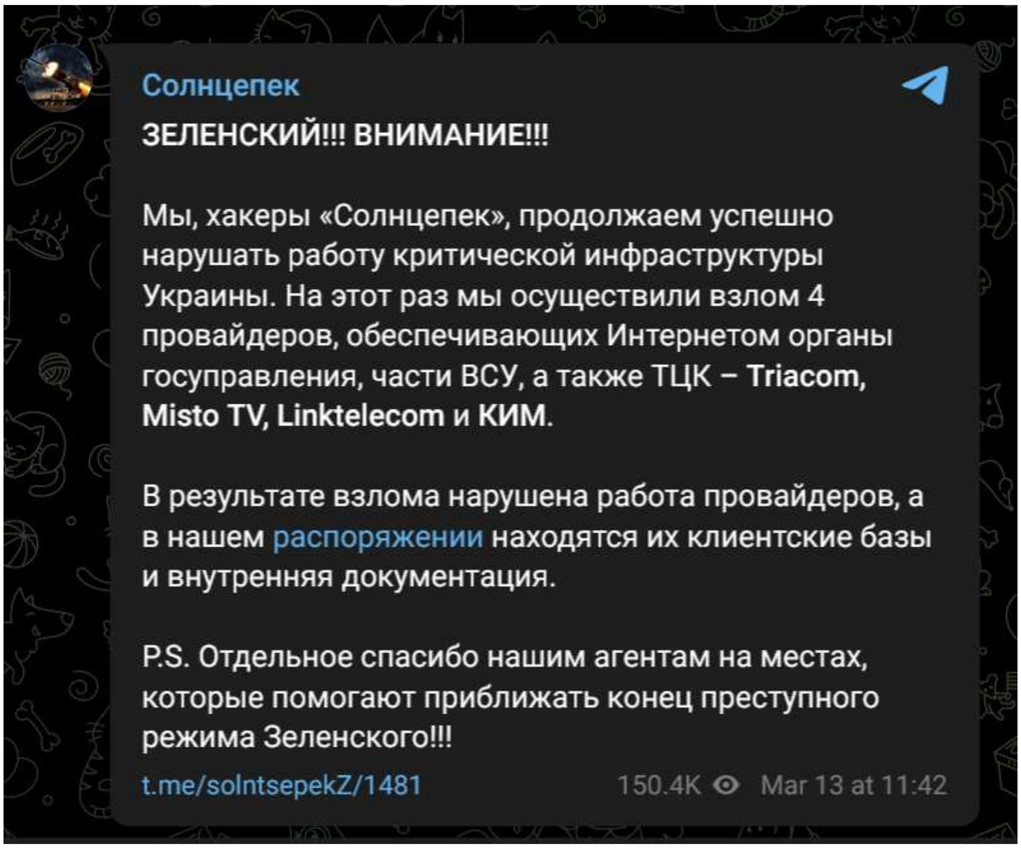
AcidPour extends these features and includes:

- **/dev/dm-XX:** Device mapping framework, making storage area networks (SAN) and network attached storage (NAS) vulnerable,
- **/dev/ubiXX:** The UBI interface is a flash memory wear management system. It is common in embedded systems like mobile devices, IoT, and even, sometimes, industrial control systems (ICS).
- **Self-delete:** This new version starts with a self-destruct feature, by mapping the original file into memory and then overwriting it with a sequence of bytes ranging from 0 to 255 followed by an "OK".

It is interesting to note that AcidPour is developed in C, like CaddyWiper, used against power plants in Ukraine (*see November 2023's monthly bulletin*) by Russian military intelligence. These new features seem to suggest it might be used against industrial systems, used in factories, power plants, or public infrastructures…

4.4.3. Attributions

CERT-UA has assigned the exploitation of AcidPour to UAC-0165, a subgroup of APT44 (ex -Sandworm). In addition, the discovery of AcidPour by SentinelOne security researchers coincides with an attack claimed by Solntsepek on 13 March 2024, i.e. 3 days before. This latest attack targeted 4 operators in Ukraine, Triacom, Misto TV, Linktelecom and КИМ, whose networks were paralysed for a week :



Солнцепек

ЗЕЛЕНСКИЙ!!! ВНИМАНИЕ!!!

Мы, хакеры «Солнцепек», продолжаем успешно нарушать работу критической инфраструктуры Украины. На этот раз мы осуществили взлом 4 провайдеров, обеспечивающих Интернетом органы госуправления, части ВСУ, а также ТЦК – **Triacom, Misto TV, Linktelecom и КИМ.**

В результате взлома нарушена работа провайдеров, а в нашем распоряжении находятся их клиентские базы и внутренняя документация.

P.S. Отдельное спасибо нашим агентам на местах, которые помогают приближать конец преступного режима Зеленского!!!

t.me/solntsepekZ/1481 150.4K Mar 13 at 11:42

Figure 25. Source: Solntsepek.

4.5. Conclusion

- Geopolitics is ever present in sporting competitions since the beginning of modern sports in the 20th century. The principle that politics has no place in sport may seem to be a response to the aggressive ambitions of countries opposed to France on the world stage, but this idea is unfounded.
- This posture, although seemingly noble, should not obscure the threat currently weighing on France and these Olympic Games, with adversaries openly asserting their hostility.
- One of the most feared scenarios is the use of one or more wipers against the Olympic Games' infrastructure, similar to what happened during the Pyeongchang Games. At that time, APT groups affiliated with the Russian military intelligence were not involved in the ongoing campaign in Ukraine. However, the proliferation of this destructive malware seen in this recent conflict raises concerns about its possible use during the Olympic Games. Additionally, the various GRU groups leveraged the experience gained during the conflict to develop a five-phase intrusion and attack model, designed for high-intensity offensive cyber operations aimed at increasing the speed, scale and intensity of attacks while minimising the risks of detection.

4.6. IoCs

TLP	TYPE	VALUE	Comments
TLP:CLEAR	File	tmphluyl8zn	AcidPour sample
TLP:CLEAR	SHA256	30584f13c0a9d0c86562c803de350432d5a0607a06b24481ad4d92cdf7288213	AcidPour sample
TLP:CLEAR	SHA1	b5de486086eb2579097c141199d13b0838e7b631	AcidPour sample
TLP:CLEAR	MD5	1bde1e4ecc8a85cffef1cd4e5379aa44	AcidPour sample
TLP:CLEAR	IP	185[.]61.137.155	Solntsepek Domain
TLP:CLEAR	IP	solntsepek[.]com	Solntsepek Domain
TLP:CLEAR	IP	solntsepek[.]info	Solntsepek Domain
TLP:CLEAR	IP	solntsepek[.]org	Solntsepek Domain
TLP:CLEAR	IP	solntsepek[.]ru	Solntsepek Domain
TLP:CLEAR	File	acid_rain.elf	AcidPour sample
TLP:CLEAR	SHA256	9b4dfaca873961174ba935fddaf696145afe7bbf5734509f95feb54f3584fd9a	AcidPour sample

5. Sources

CVEs

- https://community.zyxel.com/en/discussion/23278/zyxel-security-advisory-for-multiple-vulnerabilities-in-nas-products
- https://www.php.net/archive/2024.php#2024-06-06-2
- https://www.solarwinds.com/trust-center/security-advisories/cve-2024-28995

GOMIR (APT KIMSUKY)

- https://www.shadowstackre.com/analysis/gomir
- https://www.virustotal.com/gui/file/30584f13c0a9d0c86562c803de350432d5a0607a06b24481ad4d92cdf7288213/details
- https://www.joesandbox.com/analysis/1445737/0/html
- https://app.any.run/tasks/78586403-ddc5-4880-ac3f-2875a5bdd7d5?_gl=1*107l1ba*_gcl_au*Njc2MjgyOTU5LjE3MTQwMzAzMTc.*_ga*MTcwNjY5NzU3Ny4xNzE0MDMwMzE3*_ga_53KB74YDZR*MTcxNzQ5OTg0Ny4yLjEuMTcxNzUwMDExMC4wLjAuMTM1MjQ3OTk5/
- https://symantec-enterprise-blogs.security.com/threat-intelligence/springtail-kimsuky-backdoor-espionage
- https://thehackernews.com/2024/05/kimsuky-apt-deploying-linux-backdoor.html
- https://bazaar.abuse.ch/sample/30584f13c0a9d0c86562c803de350432d5a0607a06b24481ad4d92cdf7288213/
- https://www.welivesecurity.com/fr/2020/11/17/attaque-lazarus-coree-du-sud/
- https://any.run/report/7bd723b5e4f7b3c645ac04e763dfc913060eaf6e136eecc4ee0653ad2056f3a0/1d49cc21-50f1-4784-8216-decee3dbaf0d
- https://www.mphasis.com/content/dam/mphasis-com/global/en/home/services/cybersecurity/june-3-2-mysterious-threat-actor-used-chalubo-malware.pdf
- https://asec.ahnlab.com/en/61934/

Olympic games 2024 : AcidPour Analysis

- https://blog.sekoia.io/securing-gold-assessing-cyber-threats-on-paris-2024/#h-history-of-cyber-operations-impacting-olympic-games
- https://cloud.google.com/blog/topics/threat-intelligence/cyber-threats-2024-paris-olympics?hl=en
- https://www.france24.com/fr/info-en-continu/20240226-ni-r%C3%A9sign%C3%A9s-ni-d%C3%A9faitistes-r%C3%A9union-%C3%A0-paris-des-alli%C3%A9s-de-l-ukraine
- https://www.leparisien.fr/jo-paris-2024/jo-2024-les-athletes-russes-et-bielorusses-pas-les-bienvenus-a-paris-dit-anne-hidalgo-depuis-kiev-30-03-2024-MYSOLNPL5RBKLIZUPDD6ZCM2WE.php
- https://www.francetvinfo.fr/les-jeux-olympiques/jo-2021-a-tokyo-taiwan-ou-taipei-chinois-la-question-est-sensible-depuis-quarante-ans-et-pekin-veille-au-grain_4715377.html
- https://x.com/AlvieriD/status/1805074447130636320
- https://www.numerama.com/cyberguerre/1633112-les-polices-de-11-pays-dont-la-france-abattent-le-site-de-lockbit-le-plus-important-gang-de-hackers.html
- https://www.trendmicro.com/en_us/research/24/d/operation-cronos-aftermath.html
- https://www.lemondeinformatique.fr/actualites/lire-lockbit-30-revendique-un-vol-de-donnees-de-la-fed-94113.html
- https://x.com/DarkWebInformer/status/1805256295248769119
- https://next.ink/141115/une-attaque-ddos-aurait-vise-une-dizaine-de-sites-gouvernementaux-francais/
- https://www.numerama.com/cyberguerre/1764372-euro-2024-la-pologne-accuse-des-hackers-russes-davoir-perturbe-la-diffusion-dun-match.html
- https://www.cert.ssi.gouv.fr/cti/CERTFR-2024-CTI-006/
- https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/overview-of-the-cyber-weapons-used-in-the-ukraine-russia-war/
- https://www.bleepingcomputer.com/news/security/new-acidpour-data-wiper-targets-linux-x86-network-devices/
- https://www.sentinelone.com/labs/acidpour-new-embedded-wiper-variant-of-acidrain-appears-in-ukraine/
- https://cip.gov.ua/en/news/yak-zminyuyutsya-taktiki-cili-i-spromozhnosti-khakerskikh-grup-uryadu-rf-ta-kontrolovanikh-nim-ugrupovan-zvit
- https://www.wired.com/story/ukraine-kyivstar-solntsepek-sandworm-gru/
- https://cloud.google.com/blog/topics/threat-intelligence/gru-disruptive-playbook/?hl=en