aDvens

Security for the greater good

Renseignement sur les menaces
Bulletin du mois de juin 2024

2024-06-20 | **TLP:CLEAR** | CERT aDvens - CTI
Advens - 16 Quai de la Mégisserie - 75001 Paris

Sommaire

1. Virologie : analyse d'un échantillon Gomir (APT Kimsuky)

1.1. Une porte dérobée sophistiquée

Découvert en mai 2024, Gomir est une porte dérobée utilisée par l'APT Kimsuky (Corée du Nord). Développé en **Go** au format **ELF 32**, celle-ci est conçue pour être utilisée sur le système d'exploitation **Linux**.

Gomir a été distribué lors d'une campagne de cyber-espionnage ciblant des organisations localisées en Corée du Sud et aux Etats-Unis.

L'analyse de Gomir révèle la sophistication de ce maliciel ainsi que des similitudes avec Gobear (destinée aux environnements Microsoft), une autre porte dérobée intégrée à l'arsenal de l'APT Kimsuky.

1.2. Fonctionnalités

Ci-dessous, les principales fonctionnalités du logiciel malveillant Gomir.

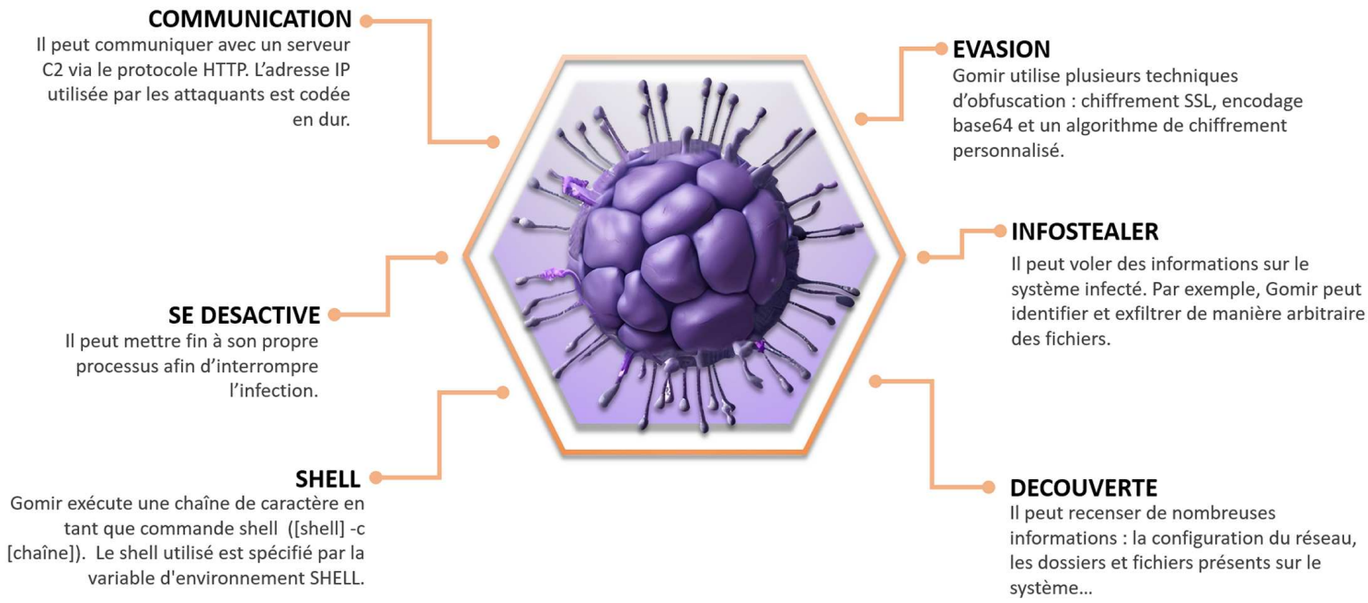
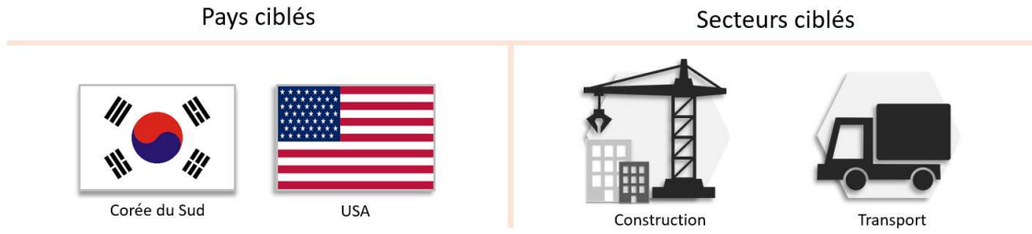


Figure 1. Les fonctionnalités de Gomir : une porte dérobée multifonction.

1.3. Victimologie



1.4. Infectiologie

1.4.1. Chaine d'infection : synthèse

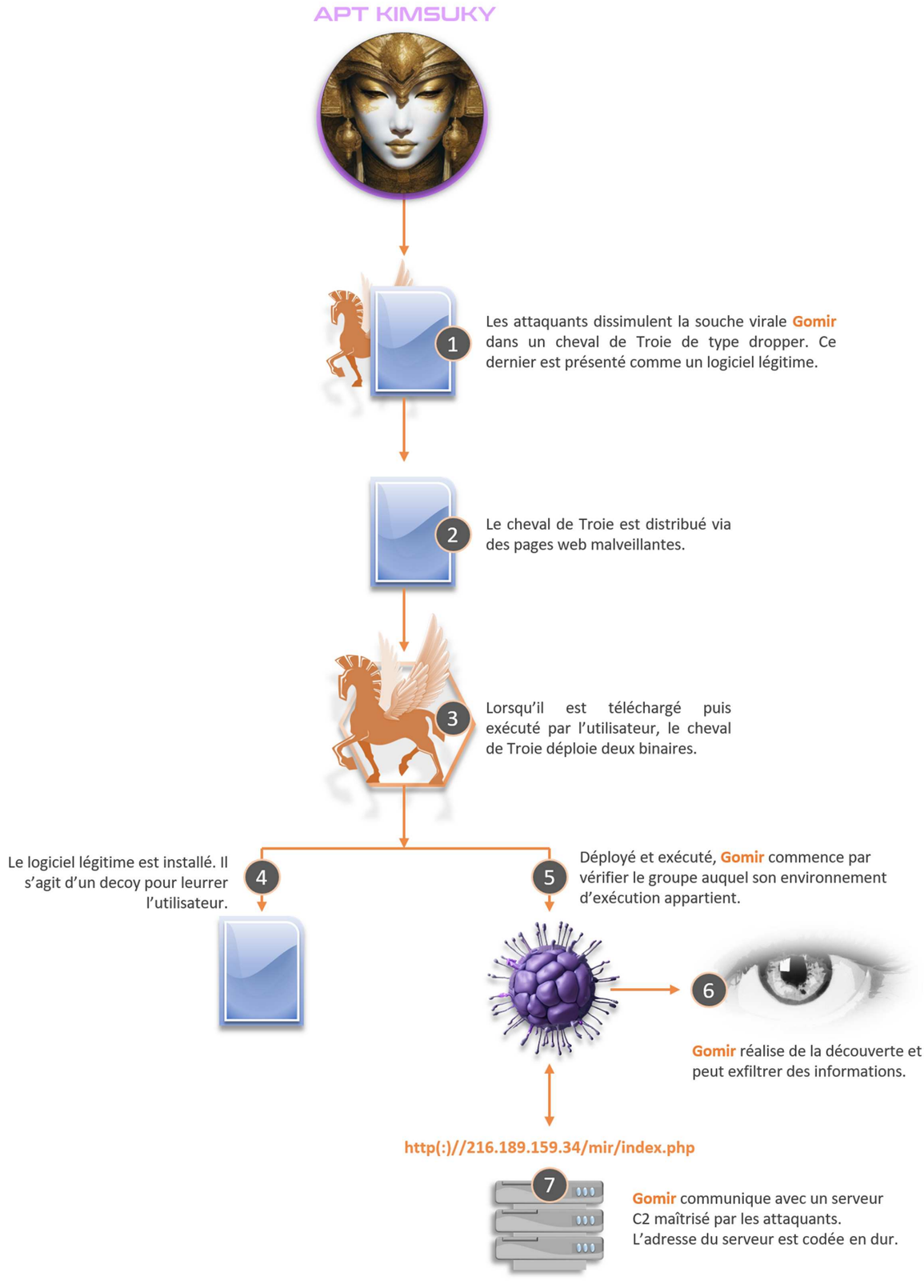


Figure 2. Synthèse infographie de la chaine d'infection.

1.4.2. Chaine d'infection : analyse détaillée

Vecteur d'infection

Le principal vecteur d'infection utilisé par les attaquants est la distribution de chevaux de Troie de type dropper. Ces derniers sont présentés comme étant des logiciels légitimes, ils embarquent deux binaires :

- Le logiciel légitime : celui-ci est déployé puis exécuté pour leurrer (decoy) l'utilisateur.
- La souche virale Gomir : une porte dérobée sophistiquée qui est installée de manière discrète.

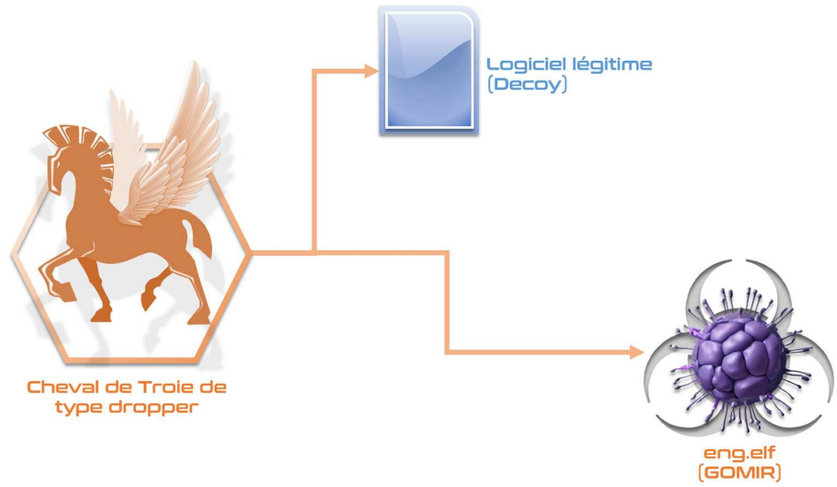


Figure 3. Principal vecteur d'infection : cheval de Troie dropper.

Depuis le début de l'année 2024, différents logiciels légitimes ont été exploités de manière malveillante par l'APT Kimsuky pour dissimuler les souches virales (Troll Stealer,Gobear et Gomir).

- **TrustPKI** et **NX_PRNMAN** de la société *SGA Solutions*.
- **Wizvera VeraPort** de la société *Wizvera*.
- **Humetro** de la société *Humetro Busan Kr*.

Par ailleurs, la chaine d'approvisionnement de *Wizvera VeraPort* a été victime de multiples cyberattaques attribuées au groupe nord coréen APT Lazarus en 2020.

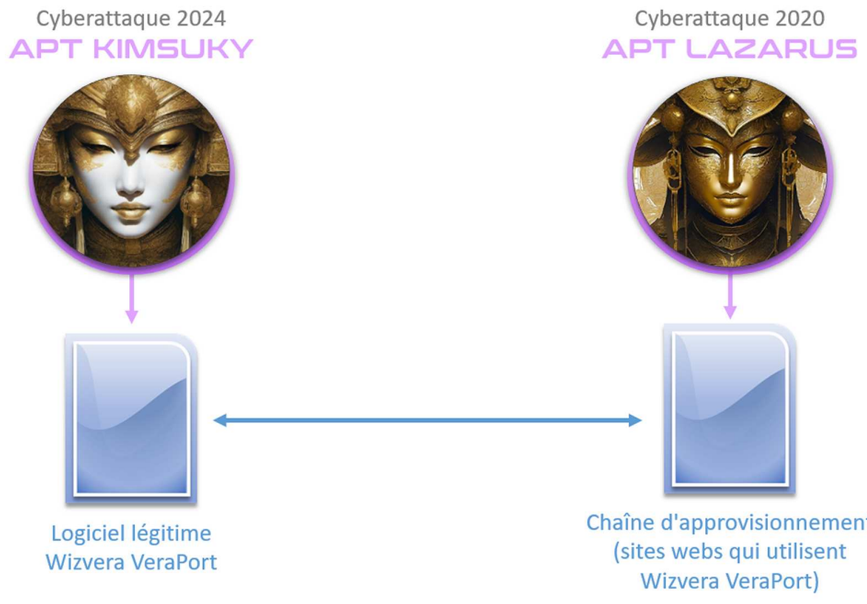


Figure 4. Wizvera VeraPort : une cible répétée de la menace nord-coréenne.

1.4.3. Analyse de la souche virale

Exécution et vérification du groupe

Lorsqu'il est déployé et exécuté par le cheval de Troie de type dropper, Gomir commence par vérifier le groupe auquel appartient son processus d'exécution. Pour cela, c'est la fonction suivante qui est utilisée:

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

Ce qui permet de résoudre la fonction **getegid32()**.

Si son processus appartient au groupe 0 (privilège root), la porte dérobée Gomir garantit sa persistance lors de l'installation via **systemd**. Dans le cas contraire, elle sera maintenue via le planificateur de tâches **crontab**.

Installation avec persistance via Systemd

Si son processus appartient au groupe 0 (privilège root), alors Gomir se copie dans le dossier :

```
/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 : l'instruction LEA de la fonction 082de32d charge le registre EAX avec les données 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 : les données DAT_083471c7, celles-ci correspondent à /var/log/syslogd.

Gomir crée un fichier dans le dossier suivant :

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

Ce dernier contient les informations ci-dessous :

```
[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]
         8f 35 08
082de3e2 f7 d9           NEG       ECX
082de3e4 90              NOP
```

Figure 7. GHIDRA - CodeBrowser : l'instruction LEA de la fonction 082de3dc charge le registre ECX avec les données 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 : les données DAT_08358f4b, celles-ci correspondent au contenu de l'artéfact syslogd.service

Le fichier **syslogd** permet sur les environnements linux de configurer la journalisation des évènements. Afin que les modifications soient prises en compte, Gomir exécute les commandes suivantes :

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

Lorsque ce service est exécuté, Gomir interrompt son processus et se supprime pour ne pas laisser de trace.

Installation avec persistance via crontab

Si son processus n'appartient pas au groupe 0 (privilège root), Gomir utilise crontab pour établir sa persistance. Pour cela, le fichier **cron.txt** est créé dans le dossier où Gomir est présent.

Le fichier **cron.txt** contient le code suivant :

```
@reboot [Chemin_du_processus]
```

```
                    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 : l'intitulé de l'artéfact (cron.txt) est codé en dur dans les données DAT_08343e99.

Gomir tente de répertorier toutes les entrées crontab existantes, puis les concatène dans le fichier **cron.txt**. Après le chargement de la nouvelle configuration de crontab (commande ci-dessous), le fichier est supprimé.

```
/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 : la chaine de caractère de la commande (crontab cron.txt) est codée en dur dans les données DAT_083471d7.

Identifiant de la victime

Lors de l'infection du système, Gomir génère un identifiant de la victime via la fonction **generate_infection_id** :

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

Cet identifiant est utilisé lors de la communication avec le serveur C2.

Communication avec le serveur C2

Gomir communique avec un serveur C2 dont l'adresse est codée en dure. La communication est réalisée via des requêtes HTTP.

```
http(:)//216.189.159.34/mir/index.php
```

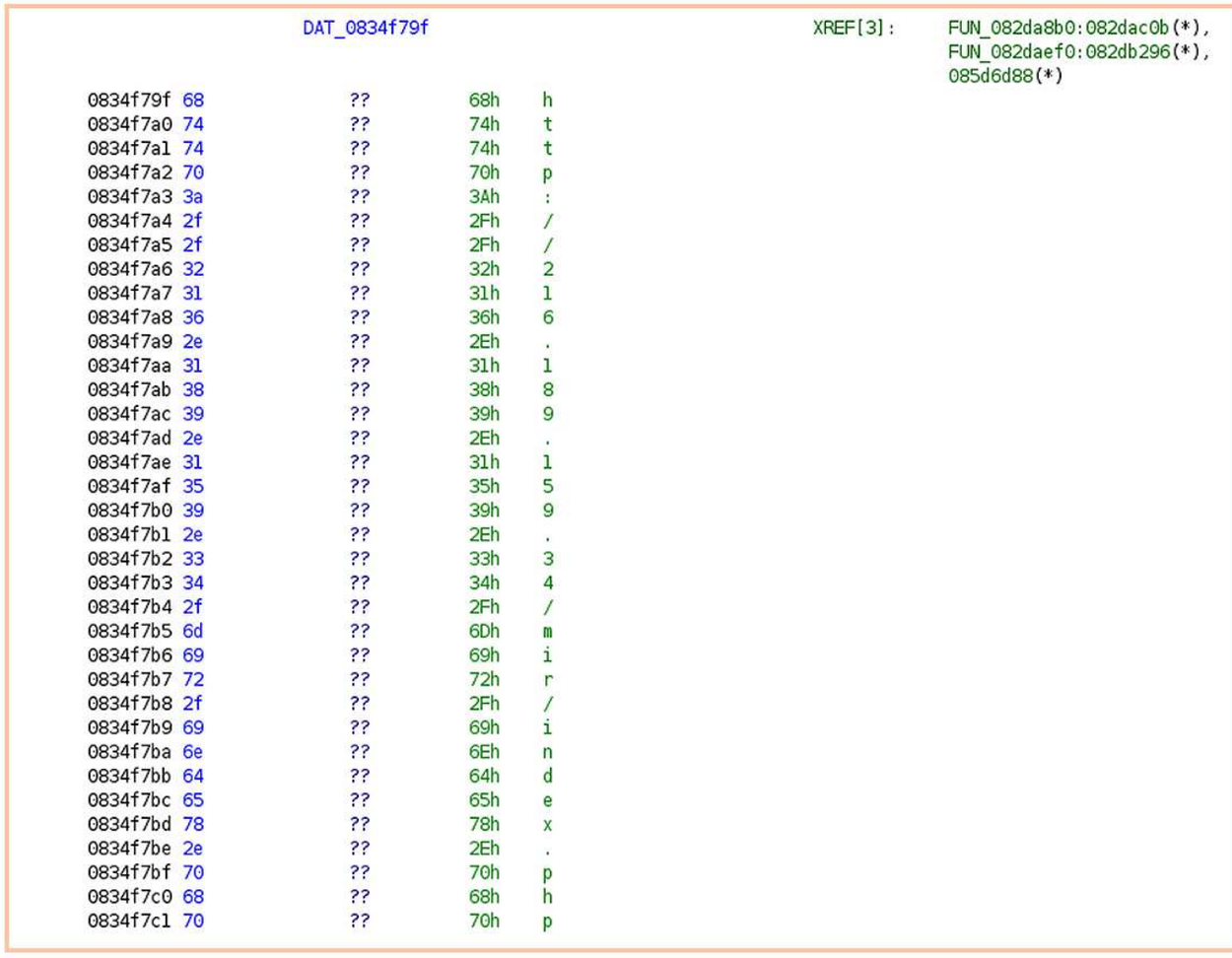


Figure 11. GHIDRA - CodeBrowser : l'adresse du serveur C2 est codée en dur dans la souche virale. Données DAT_0834f79f.

Afin de recevoir de nouvelles instructions, Gomir envoie une requête HTTP de type POST au serveur C2. Le corps de la requête est structuré de la manière suivante :

```
a\w{9}=2&b\w{9}=[Identifiant victime]1&c\w{9}=
```

Instructions C2

Gomir peut recevoir 17 instructions du serveur C2 :

OEPARATION	INSTRUCTIONS
1	Suspend la communication avec le serveur C&C pour une durée arbitraire.
2	Exécute une chaîne de caractères en tant que commande shell ([shell] -c [chaine]). Le shell utilisé est spécifié par la variable d'environnement "SHELL". Si cette variable n'est pas disponible un shell de secours est configuré via l'opération 10.
3	Indique le répertoire de travail actuel.
4	Modifie le répertoire de travail actuel et précise le nouveau chemin d'accès du répertoire de travail.
5	Teste la connectivité TCP de points de terminaison du réseau.
6	Arrête son processus : interrompant ainsi la porte dérobée Gomir
7	Indique le chemin de l'exécutable de son propre processus.
8	Collecte des statistiques sur une arborescence de répertoires et génère des rapports sur le nombre total de sous-répertoires et de fichiers ainsi que les tailles totales des fichiers
9	Collecte des détails de configuration de l'ordinateur compromis (nom d'hôte, nom d'utilisateur,CPU, RAM ainsi que des informations sur la configuration du réseau).
10	Configure un shell de secours à utiliser lors de l'exécution de l'opération 02. La valeur de configuration initiale est "/bin/sh".
11	Configure une page de codes pour interpréter la sortie de la commande shell de l'opération 02.
12	Suspend la communication avec le serveur C2 jusqu'à une date/heure arbitraire.
13	Répond par le message "Not implemented on Linux!"
14	Démarre un proxy inversé en se connectant à un terminal de contrôle arbitraire. La communication avec ce terminal de contrôle est chiffrée à l'aide du protocole SSL. La porte dérobée agit comme un client proxy. Cela permet à l'attaquant distant d'initier des connexions à des terminaux arbitraires sur le réseau de la victime.
15	Indique les terminaux de contrôle du proxy inverse.
30	Crée un fichier arbitraire.
31	Exfiltre un fichier arbitraire.

Eléments Github

Les attaquants semblent exploiter via Gomir plusieurs éléments issus de différents projets *Github*.

Projet : 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. Elément identifié.

- Source : https://github.com/klauspost/cpuid
- Utilisation : une bibliothèque Golang qui permet de récupérer des informations sur le microprocesseur.

Projet : 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. Elément identifié.

- Source : https://github.com/pbnjay/memory
- Utilisation : une bibliothèque Golang qui permet de récupérer des informations sur la mémoire du système.

Projet : 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. Elément identifié.

- Source : https://github.com/dustin/go-humanize
- Utilisation : ensemble de fonctions qui permettent de formater des informations sur la taille du système de fichiers.

Ci-dessous, les librairies provenant de divers dépôts github qui sont intégrées dans la souche virale 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. Intégration d'éléments de projets Github dans 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. Intégration d'éléments de projets Github dans Gomir : example 2.

1.5. Lignée virologique

1.5.1. Des similitudes au sein l'arsenal de Kimsuky

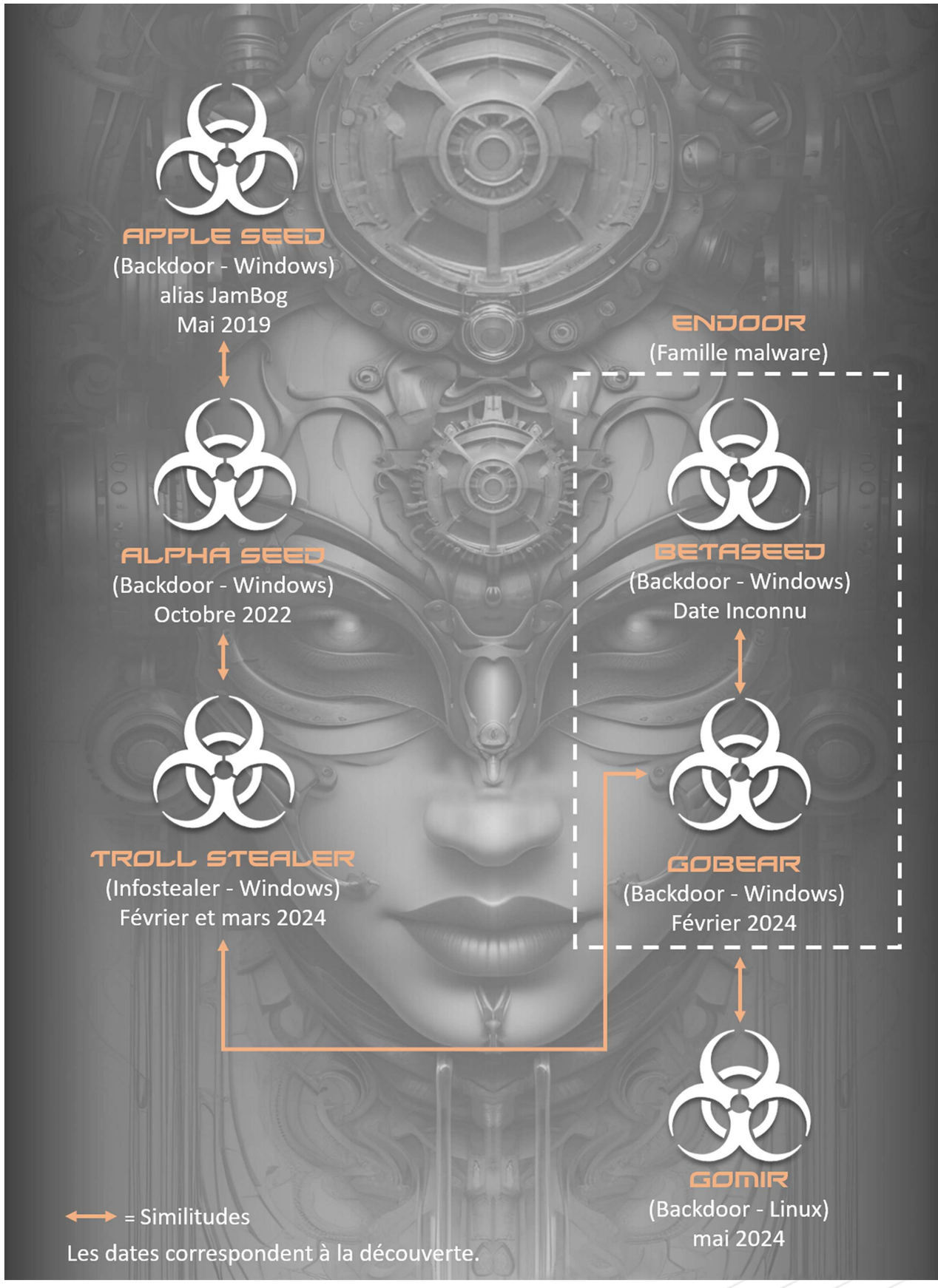


Figure 17. Infographie non exhaustive des similitudes avec l'arsenal d'APT Kimsuky.

Plusieurs similitudes ont été identifiées entre différentes souches virales appartenant à l'arsenal de l'APT Kimsuky.

Troll Stealer et Apple Seed

- L'emplacement et l'intitulé de la souche virale sont identiques. Par ailleurs, le nommage du mutex et plusieurs fonctions sont semblables.

Troll Stealer et Alpha Seed

- Le chiffrement et le déchiffrement des données sont identiques

Gobear et Gomir

- Les deux souches sont structurellement presque identiques.

Gobear et Troll Stealer

- Ils ont le même certificat *D2innovation Co.,LTD*

Gobear et Betaseed

- Certaines fonctions ont le même intitulé.

Apple Seed et Alpha Seed

- Alpha Seed est une version développée en Go d'Apple Seed.

1.5.2. Gomir : déployé par Chalubo en 2023 ?

Gomir a fait l'objet d'analyses et publications au cours du mois de mai 2024. La date exacte de l'émergence ne semble pas connue. Cependant, une information intéressante a été relevée lors d'une recherche en sources ouvertes : Gomir aurait été déployé par le cheval de Troie Chalubo lors d'une cyberattaque d'envergure en octobre 2023.

Selon un rapport publié par *Black Lotus Lab* de la société *Lumen Technologies*, plus de 600 000 routeurs aux États-Unis d'Amérique ont été mis hors-service lors d'une cyberattaque qui s'est déroulée du 25 ou 27 octobre 2023. Dans un premier temps, des attaquants inconnus ont utilisé le cheval de Troie Chalubo comme logiciel malveillant de primo-infection. Des implants additionnels ont été déployés sur les routeurs pour réaliser une opération de sabotage. Parmi les souches additionnelles identifiées, l'empreinte de l'une d'elles correspond au SHA256 de Gomir.

L'empreinte (sha256) 30584f13c0a9d0c86562c803de350432d5a0607a06b24481ad4d92cdf7288213 correspond à l'implant Gomir déployé lors de campagnes d'attaque en 2023 et 2024.

Il est possible que Gomir soit **plus ancien qu'il n'y parait** et que son utilisation **ne se limite pas qu'au cyber-espionnage mais aussi au cyber-sabotage**.

D'autres empreintes (Gobear et Troll Stealer) ont aussi été identifiées lors du cyber-sabotage qui s'est déroulé en octobre 2023.

L'hypothèse que l'**APT Kimsuky soit potentiellement l'auteur de ce sabotage est considéré comme probable**.

1.6. APT Kimsuky - Evolution de TTP

Chevaux de Troie avec decoy

Depuis le début de l'année 2024, l'APT Kimsuky utiliserait **une nouvelle technique pour déployer son arsenal**. Les attaquants forgent et distribuent via des pages web malveillantes **des chevaux de Troie de type dropper**: ces derniers déploient un decoy (application légitime) et la souche virale sur le système de l'utilisateur. **Aucun courriel d'hameçonnage** ne semble avoir été utilisé pour le déploiement de Troll Stealer, Gobear et Gomir lors des campagnes de cyber-espionnage à l'encontre de la Corée du Sud.

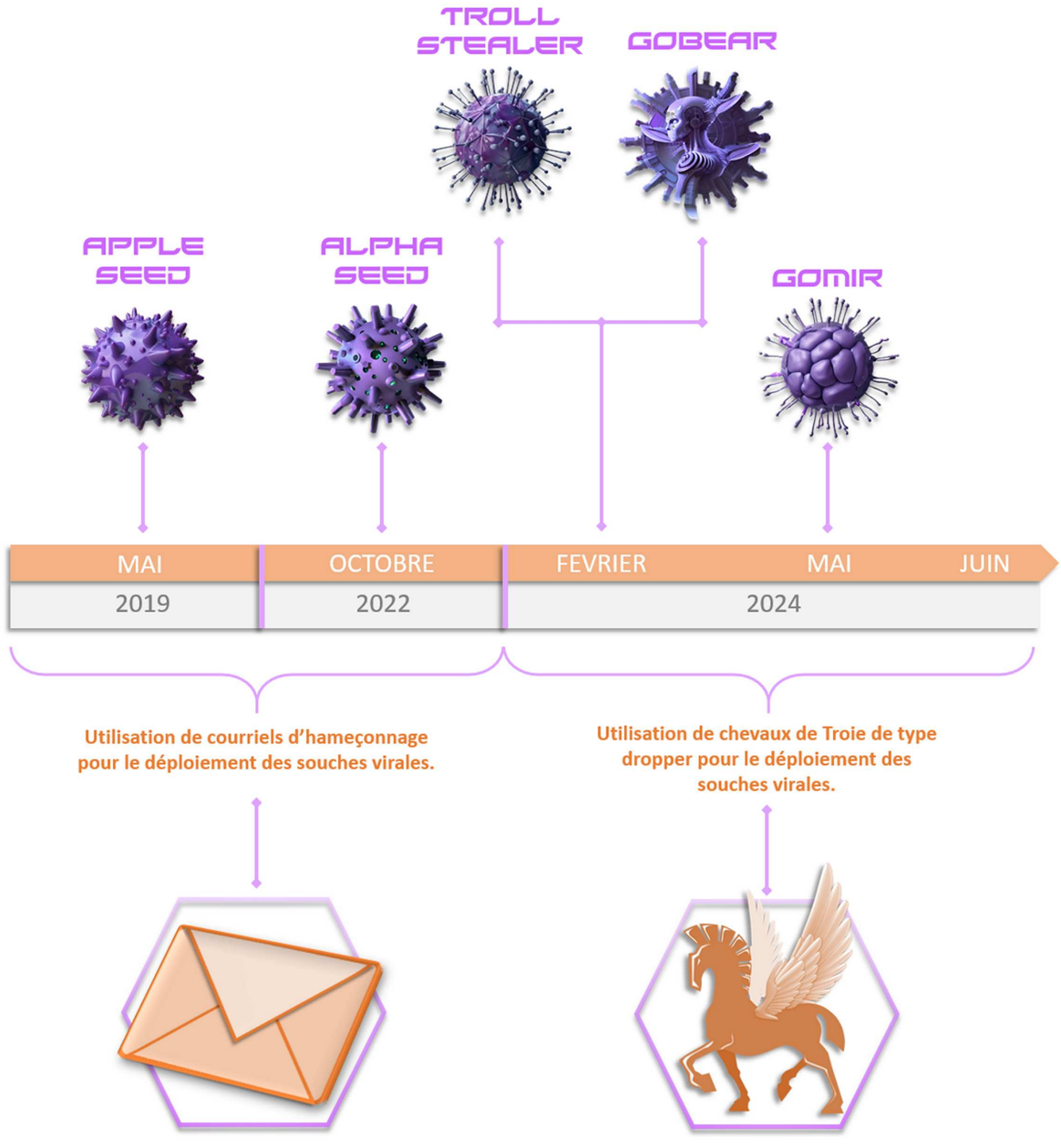


Figure 18. APT Kimsuky : évolution de TTP.

1.7. APT Kimsuky - Modèle diamant

L'APT Kimsuky (alias APT 43, TA406, Thallium, Black Banshee, Velvet Chollima…) est une menace avancée et persistante d'origine nord-coréenne

Figure 19. Modèle diamant de l'APT Kimsuky.

1.8. 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 GOMIR (APT KIMSUKY)

1.9. 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

1.10. YARA

1.10.1. 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
}
```

1.10.2. 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
}
```