Kobalos : HPC complex yet tiny malware

Résumé sommaire

Découvert par les chercheurs de l'entreprise ESET, entreprise de sécurité informatique européenne, Kobalos est un malware multiplateforme complexe ciblant les systèmes Linux, FreeBSD et Solaris. Ce malware n'a pas été répandu au hasard car il cible seulement des organisations puissantes tel que de grands Instituts de recherches, des agences de marketing, des universités ou même encore le gouvernement.

Une fois déployé, ce malware donne accès au système de fichier de l'hôte et permet d'accéder à un terminal à distance, permettant à l'attaquant d'éxécuter des commandes à distances.

L'hôte infecté par Kobalos peut agir passivement mais aussi comme un bot activement connecté un serveur C&C (Command and Control). Ces mêmes serveurs C&C sont eux aussi infectés par Kobalos, le code de lancement de serveur est aussi présent dans le malware.

Les chercheurs ne sont pas parvenus à savoir quand Kobalos a été écrit mais ils ont pu remonter à la première activité fin 2019. Le groupe derrière Kobalos est resté actif jusqu'au milieu de 2020.

Point clefs

Kobalos est une backdoor multiplateforme qui fonctionne sur Linux, FreeBSD et Solaris. Il y a plusieurs artefacts qui indiquent que des variants de ce malware existent pour les systèmes AIX (IBM os) et Windows.

Kobalos visent principalement :

- des HPC (High-Performance Computer) d'universités
- des endpoints de compagnie de sécurité
- fournisseur de services internets

Kobalos a été retrouvé sur des serveurs situés en Europe, en Amérique du Nord et en Asie.

Kobalos utilise un "complex obfuscation mechanism" qui rend l'analyse plus que compliquée.

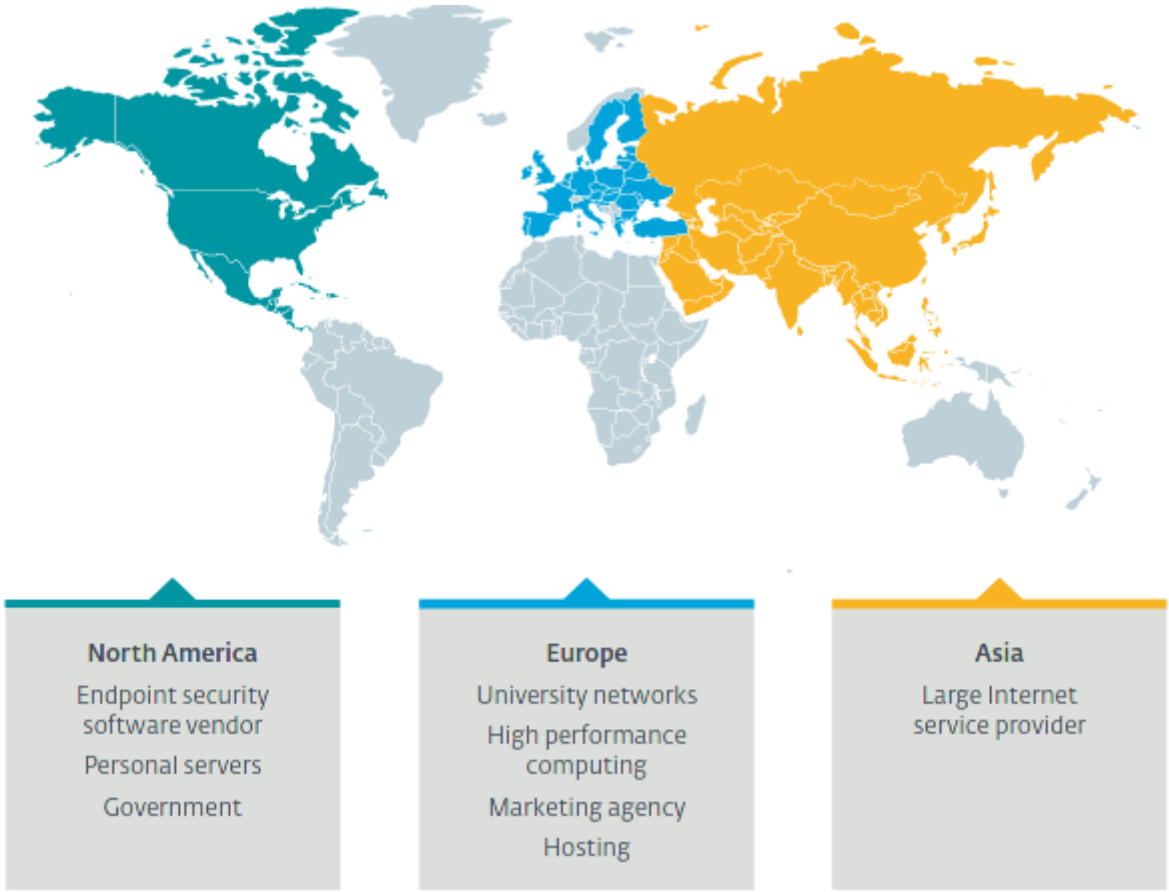
N'importe quel système infecté par Kobalos peut-être transformé en un C&C pour les autres hôtes compromis par Kobalos. Le code est intégré dans le malware et peut-être activé par un opérateur à n'importe quel moment.

Une grande partie des machines compromises possédaient aussi un "credentials stealer" d'OpenSSH, ce qui peut expliquer comment Kobalos a pu se répandre si facilement.

Pour le moment, les intentions des auteurs du malware restent inconnus. Les chercheurs n'ont trouvé aucune intention de voler des informations confidentielles, un but lucratif ou qu'ils étaient après quelques choses d'autres.

Mode opératoire

Suite au scan organisé par ESET pour trouver les victimes de Kobalos. Il y a eu très peu de victime à travers le monde, mais elles ne sont pas choisis au hasard, principalement des HPC faisant partie de réseau académique et de recherche. L'un des ordinateurs infecté possède 512Go de RAM et quasi un petabyte d'espace. Ces ressources n'ont pas été utilisé pour miner des cryptomonnaies ou autres activités illicites.



Source : ESET Research Paper, Les industries et les régions compromisent

La source d'infection

Il n'y a pas pu avoir une trace sûr d'où provenait le Kobalos, plusieurs hypothèses sont possibles quant à son arrivé :

- Sur les machines cibles, il y avait un récupérateur de clefs SSH, ça peut-être un des moyens par lequel Kobalos s'est répandu. Étant des ordinateurs universitaires, des étudiants, professeurs ont pu se connecter dessus et avoir leur clef volée auparavant
- Un autre possibilité est que les machines compromises possèdaient des logiciels/OS obsolètes, plus supportés ou pas à jour. Les attaquants ont alors pu utiliser ces failles pour s'immiscer dans les systèmes.

Fonctionnement

Kobalos possède de nombreuses fonctionnalités lui permettant de récupérer les données. Voici comment il procède :

1. Tout d'abord, il possède de nombreuses commandes basiques pour lire et écrire sur le système de fichier et aussi pour ouvrir un terminal permettant d'exécuter n'importe quelle autre commande. Cependant rien ne montre précisément les intentions de l'auteur. il l'utiliserai sûrement en ouvrant un terminal et en tappant à distance les commandes dont il a besoin.

2. Ensuite, il y a beaucoup de fonctionnalités permettant d'établir une connexion entre un serveur infecté et l'opérateur. Dans toutes ces fonctionnalités on retrouve 3 façon d'établie cette connexion :

    - En ouvrant un port TCP et en attendant une connexion externe (aussi appelé "Passive Backdoor")
    - En se connectant à une autre instance Kobalos configuré en tant que serveur C&C
    - En attendant la connexion d'un autre service provenant d'un port TCP source précis

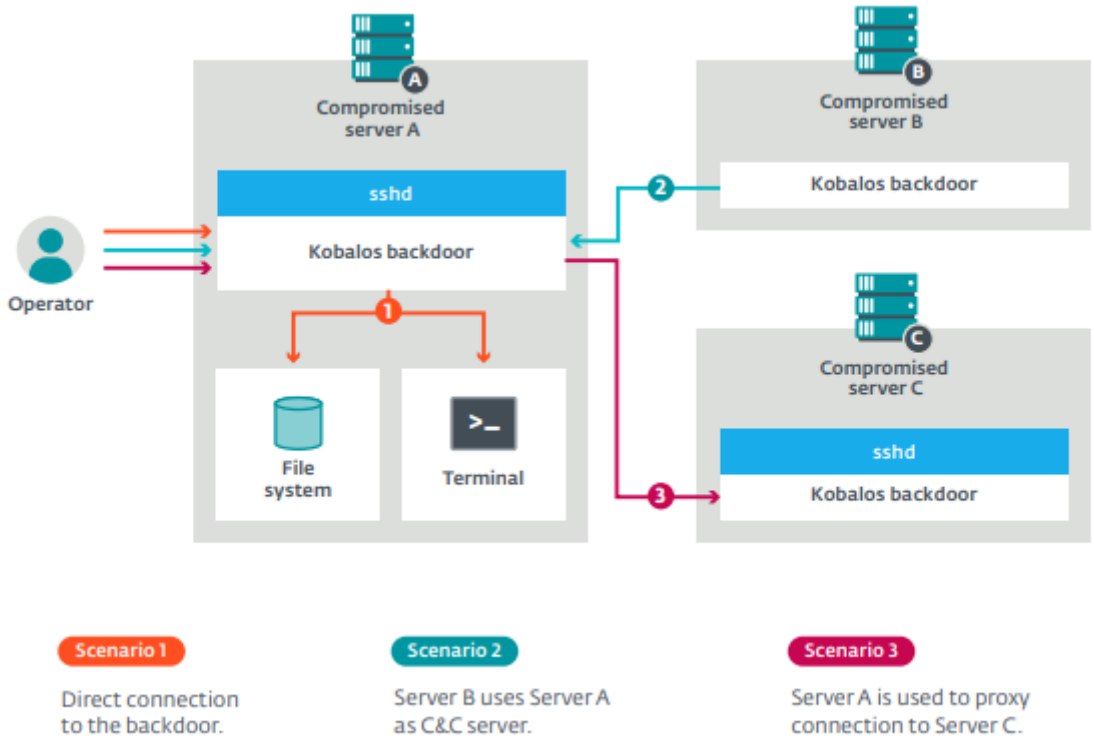
    Pour la dernière façon de fonctionner, il est nécessaire de remplacer le service en fonctionnement par un avec le code Kobalos. Quand les opérateurs ont choisis d'utiliser cette méthodes ils ont toujours modifier le serveur OpenSSH en fonctionnement. Pour cela ils remplaçaient le fichier `sshd` pour que le malware reste persistent sur le service ou après un redémarrage système.

3. Pour activer une backdoor, il faut qu'un client s'authentifie avec une clef privé RSA-512 et un mot de passe. Une fois que les deux sont validés, Kobalos génère et encrypte 2 clefs de 16 octets avec la clef publique RSA-512 et les envoient à l'attaquant. Ces clefs seront ensuite utilisées pour encrypter les données entrantes et sortantes.

4. Pendant la phase d'authentification, l'opérateur distant peut choisir de poursuivre la communication sur un autre port TCP. S'il choisit de le faire, Kobalos va alors écouter sur ce port et déplacer toutes les communications vers ce port pour que les données soient encryptées avec les clefs RC4 échangées auparavant.

5. Enfin, Kobalos peut aussi être utiliser comme un Proxy pour connecter d'autres serveurs infectés par celui-ci. Mais ce n'est pas un simple Proxy car il attend que la connexion soit encapsulée avec des paquets spécifiques au malware. Bien-sûr ce fonctionnement supporte aussi le choix alternatif de port mentionné précédemment. Pour cela, il faut envoyer une commande au Proxy pour changer la connexion de port. Bien entendu, les proxy peuvent s'enchaîner pour que l'opérateur puisse accéder à une machine infectée via d'autres machines infectées.

En combinant tout ce qui a été expliqué ci-dessus, on obtient un fonctionnement très complexe à déceler. Voici à quoi il ressemble :



Source : ESET Research Paper, Overview of Kobalos features and ways to access them

Analyse technique

Le 1er échantillon de Kobalos analysé a été un serveur OpenSSH pirate. La taille du code et de ses données est assez basse : approximativement 25 kB pour une architecture x86-64. Une chose qui rend spécial Kobalos est que l'ensemble du code est empaqueté dans une seule fonction. Il n'y a qu'un seul appel à cette fonction dans le code légitime d'OpenSSH.

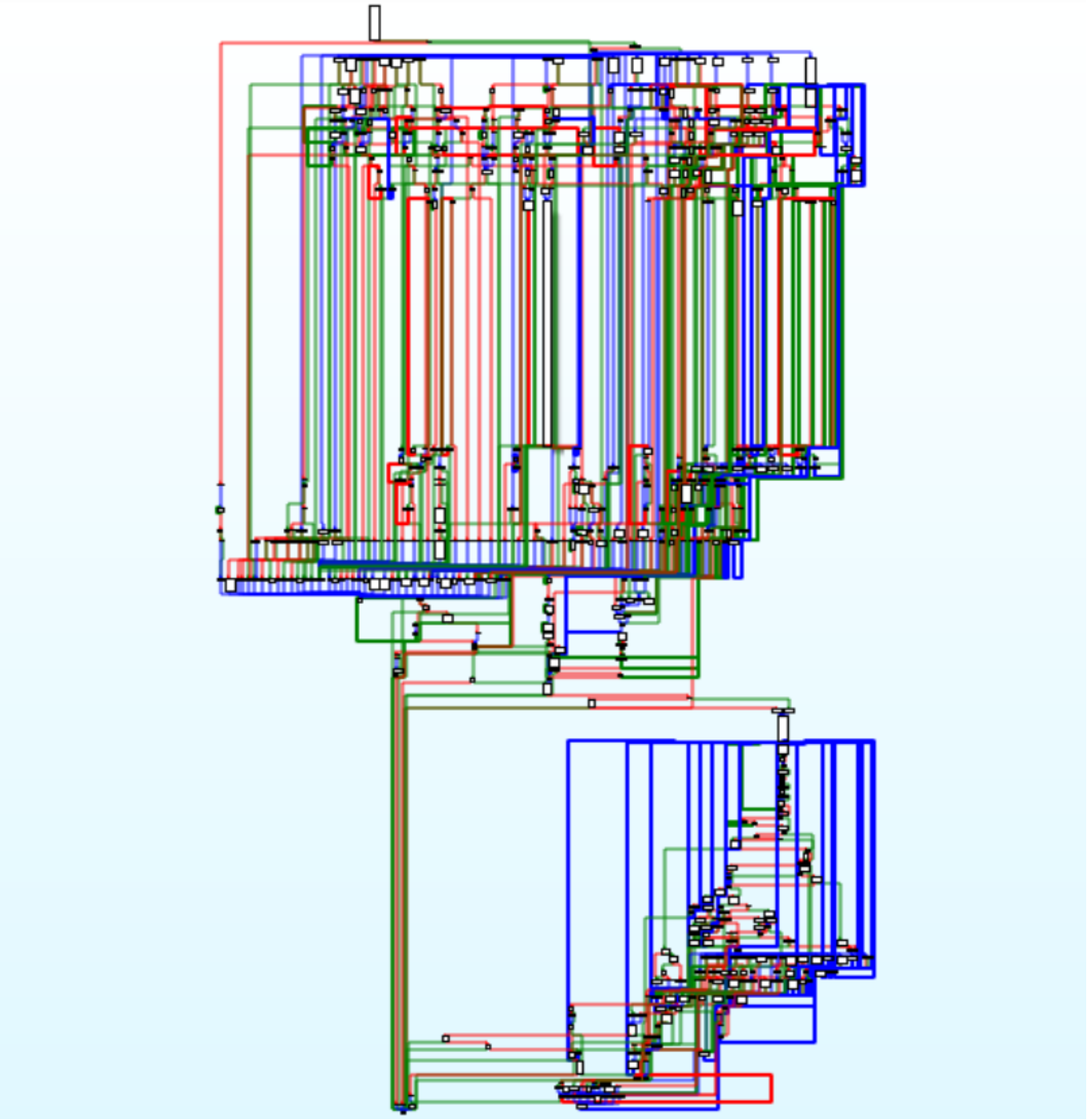
Kobalos est un malware complexe dans lequel, ses développeurs ont investit un temps et des ressources considérables. Ses auteurs ont développés plusieurs fonctionnalités et ont pris le temps d'implémenter un "obscurcissement" (obfuscation) propre à ce malware.

Obfuscation

Control flow

La fonction principale (et unique de Kobalos) s'appelle récursivement en modifiant ses paramètres d'appels pour réaliser la tâche voulu. La figure ci-dessous montre le "control flow graph" de Kobalos.

Control flow graph : représentation graphique, sous forme d'un graphe, de tous les chemins possibles qui peuvent être traversés dans l'éxécution du programme.



Source : ESET Research Paper about Kobalos, CFG de Kobalos **

Le 1er paramètre à cette fonction est l'action à effectuer. Il y a un total de 37 actions à effectuer et cette fonction et s'occupe en plus de gérer les signaux ***SIGCHLD*** et ***SIGALARM*** qui servent respectivement à laisser finir un processus fils gracieusement et ***SIGALARM*** pour gérer un évènement de type "connection timeout".

D'un point de vue code source, une transformation est faite, exemple de la transformation :

Before

```
int add(int a, int b) {
    return a + b;
}

int mul(int a, int b) {
    return a * b;
}

int square(int a) {
    return mul(a, a);
}

int get_magic(void) {
    return add(square(59), 56);
}

int main(void) {
    return get_magic();
}
```

After

```
int f(int action, int a, int b) {
    int ret;
    switch(action) {
        case 1000:
            ret = a + b;
            break;
        case 1001:
            ret = a * b;
            break;
        case 1002:
            ret = f(1001, a, a);
            break;
        case 1003:
            ret = f(1002, 59, 0);
            ret = f(1000, ret, 56);
            break;
    }
    return ret;
}


int main(void) {
    return f(1003, 0, 0);
}
```

Source : ESET Research Paper, obfuscation du code source

Une partie de cette obfuscation peut-être automatisée avec le compilateur mais demande une retouche manuelle ou un outil personnalisé pour assigner des identifiants numériques pour chaque fonction et pour pouvoir gérer le même nombre d'arguments dans toutes les fonctions.

Anti-forensics

Pour éviter que le malware laisse des traces lors d'une erreur, le signal ***RLIMIT_CORE*** est mis à zéro pour éviter la génération d'un *core dump file.*

Il ignore aussi la plupart des autres signals pour rendre plus difficile l'arrêt du programme.

Le timestamp des fichiers tels que *ssh (pour ajouter le crédential stealer)* ou *sshd (pour déployer Kobalos)* est faussé pour être le moins suspect possible.

Le programme contenait quelques chaines de caractères encryptés avec RC4 (ARC4). Elles sont décryptées juste après la communication initiale mais avant l'authentification.

1. %s %s
2. /dev/ptmx
3. ptem
4. ldterm
5. ttcompat
6. /dev/tty
7. %s
8. %d
9. /
10. \
11. %d.%d
12. win3.11
13. win95
14. winNT
15. win??
16. \\.\pipe\2
17. %s %s.%s
18. /dev/ptc

Source : ESET Research Paper, encrypted strings

Seules les variables 1, 6-9 ont été utilisé dans les programmes Kobalos analysés. Les autres, quant à elles, ont pu être utilisé dans d'autre variante : les variables [10, 12-16] sont spécifiques à Windows, la 18 pour système AIX et 3-5 pour un système Solaris (Oracle)

OpenSSH Credential Stealer

Sur la plupart des systèmes infectés par Kobalos, les chercheurs ont retrouvé un malware permettant de voler les identifiants SSH grâce à un client SSH utilisé comme “Cheval de Troie (Trojan)”. Plusieurs variantes de ce malware ont été trouvées dont certaines pouvant fonctionner sous Linux ou OpenBSD. Contrairement à Kobalos, aucun système d’obfuscation n’a été utilisé sur ce Trojan.

Sa fonctionnalité première est de voler les hostname, ports, noms de compte et mots de passe permettant de se connecter en SSH à la machine infectée. Ensuite, ces données sont enregistrées dans des fichiers qui sont ensuite encryptés.

Des versions plus récentes permettent aussi d’extraire ces identifiants dans des communications UDP, mais les chercheurs n’ont pas trouvé cette fonctionnalité activée dans leurs échantillons analysés.

Ce malware est tellement atypique qu’ils n’ont pas réussi à le cataloguer dans une famille de malware déjà connu. C’était un tout nouveau genre de backdoors OpenSSH.

Stockage des données

Le chemin de stockage des données dépend de la variantes de Credential Stealer. Cependant sur les échantillons analysés, les données étaient dans le dossier `/var/run` avec une extension en `.pid`. Voici quelques exemples de nom de fichiers et de chemins utilisés :

SHA-1	Target OS	Writes to
6616DE799B5105EE2EB83BBE25C7F4433420DFF7	RHEL	/var/run/nscd/ns.pid
E094DD02CC954B6104791925E0D1880782B046CF	RHEL	/var/run/udev/ud.pid
1DD0EDC5744D63A731DB8C3B42EFBD09D91FED78	FreeBSD	/var/run/udevd.pid
C1F530D3C189B9A74DBE02CFEB29F38BE8CA41BA	Arch Linux	/var/run/nscd/ns.pid
659CBDF9288137937BB71146B6F722FFCDA1C5FE	Ubuntu	/var/run/sshd/sshd.pid

Source : ESET Research Paper, Indicators of Compromise

Chiffrement

Tout les échantillons utilisaient le même chiffrement, très simple, consistant à ajouter `123` à chaque octets de données à sauvegarder.

Voici comment les attaquants ont procédés :

```
v23 = 'v';
v24 = 'a';
v25 = 'r';
LODWORD(v10) = v9;
v26 = '/';
v27 = 'r';
v28 = 'u';
v29 = 'n';
v30 = '/';
v31 = 'n';
v32 = 's';
v33 = 'c';
v34 = 'd';
v35 = '/';
v36 = 'n';
v37 = 's';
v38 = '.';
v39 = 'p';
v40 = 'i';
v41 = 'd';
v42 = 0;
__sprintf_chk(
  (__int64)&v19,
  1LL,
  1024LL,
  (__int64)"user: %.128s host: %.128s port %05d user: %.128s password: %.128s\n",
  v7,
  v6,
  v10,
  v8,
  (__int64)v5);
v11 = strlen(&v19);
if ( v11 > 0 )
{
  v12 = v20;
  v13 = &v19;
  v14 = &v20[v11 - 1];
  while ( 1 )
  {
    *v13 += 123;
    v13 = v12;
    if ( v12 == v14 )
      break;
    ++v12;
  }
}
v15 = "a";
v16 = fopen(&Filename, "a");
v17 = v16;
if ( v16 )
{
  v15 = (_BYTE *)(&n + 1);
  __fprintf_chk(v16, 1LL, "%s", &v19);
```

Source : ESET Research Paper, Encrypting and writing SSH credentials to a file

En ce qui concerne la variante présente sur des instances FreeBSD, le même chiffrement est appliqué mais différement. Par exemple, le chemin du fichier est encrypté dans le malware avec un XOR sur un octet.

Conclusion

Le nombre de fonctionnalités, la taille et la complexité du malware ainsi que les façons de détourner le réseau montrent que les auteurs de Kobalos ont bien plus de connaissances et de rigueur que les auteurs de malware communs sous Linux ou autres systèmes. De plus il a été très difficile de le repérer à cause de son empreinte laissée sur les serveurs et à la facilité qu'il avait à détourner le réseau à sa guise. De plus les auteurs ne voulaient sûrement pas toucher le plus grand publique car Kobalos a été déployé seulement sur des cibles de grande envergure possèdant des HPC.

Pour le moment la question de savoir quelles étaient les intentions des opérateurs de Kobalos n'est toujours pas répondu. Mise à part le Credential Stealer, aucun autre malware n'a été trouvé sur les serveurs, ce qui laisse penser que celui-ci servait plutôt de point d'entrée plutôt que d'un malware envoyé après que Kobalos ai infecté les serveurs. De plus il a été confirmé par les administrateurs des HPC que Kobalos ne servait pas à miner des cryptomonnaies.

Une dernière question se pose : depuis quand Kobalos est en service ? Des trâces liées à Windows 3.11 et Windows 95 sont présentes dans Kobalos. Or ceux-ci étaient en service il y a plus de 25 ans. Malgré cela, il a été découvert seulement entre 2019 et 2020.

Pour terminer, même si les auteurs sont très doués, Kobalos possèdait des faiblesses. La première concerne la cryptographie, certainement à cause du fait qu'il soit vieux et que les méthodes ont évolués depuis lors. Il serait possible de détecter les variants automatiquement, en regardant les ports TCP spécifiques qui sont utilisés par le programme. Sans ces erreurs, les chercheurs n'auraient sûrement pas pû identifier et contacter les autres victimes de Kobalos.

Sans ces erreurs, les chercheurs n'auraient sûrement pas pû identifier et contacter les autres victimes de Kobalos.