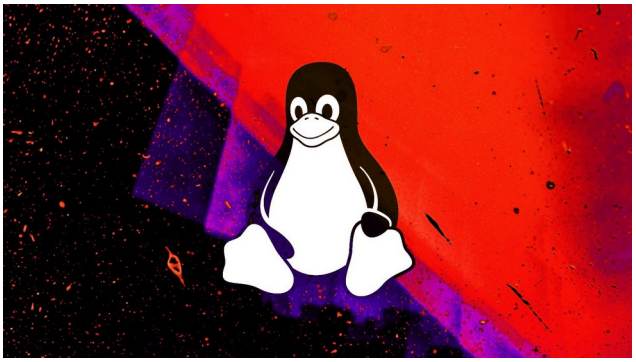
Les pirates chinois ciblent Linux avec un nouveau logiciel malveillant WolfsBane

Une nouvelle porte dérobée Linux appelée "WolfsBane" a été découverte, considérée comme un port de logiciels malveillants Windows utilisé par le groupe chinois de piratage de "Gelsemium".



Les chercheurs en sécurité d'ESET qui ont analysé WolfsBane rapportent que WolfsBane est un outil de logiciels malveillants complet avec un compte-gouttes, un lanceur et une porte dérobée, alors qu'il utilise également un rootkit open source modifié pour échapper à la détection.

Les chercheurs ont également découvert «FireWood», un autre logiciel malveillant Linux qui semble être lié au logiciel malveillant Windows du projet Wood.

Cependant, FireWood est plus probablement un outil partagé utilisé par plusieurs groupes APT chinois plutôt qu'un outil exclusif/privé créé par Gelsemium.

ESET dit que les deux familles de logiciels malveillants, apparaissant toutes deux sur VirusTotal au cours de l'année écoulée, s'inscrivent dans une tendance plus large où les groupes APT ciblent de plus en plus les plateformes Linux en raison de la sécurité de Windows.

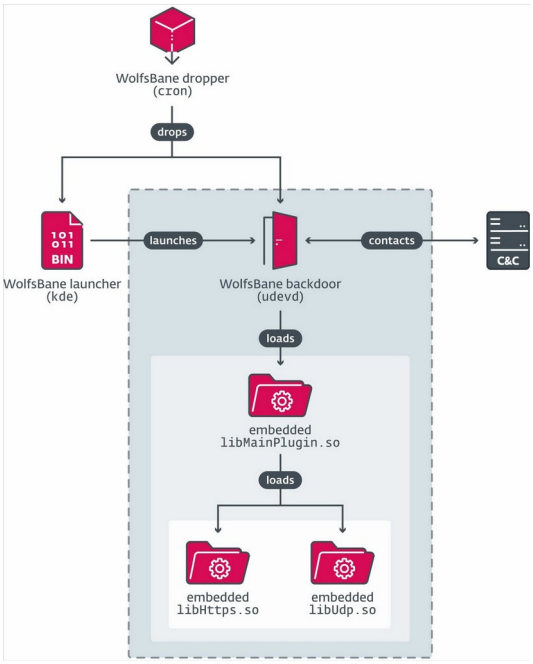
> *"La tendance des groupes APT se concentrant sur les logiciels malveillants Linux devient de plus en plus visible. Nous pensons que ce changement est dû à des améliorations de la sécurité des e-mails et des points d'extrémité Windows, telles que l'utilisation généralisée des outils de détection et de réponse des points d'extrémité (EDR) et la décision de Microsoft de désactiver les macros Visual Basic for Applications (VBA) par défaut. Par conséquent, les acteurs des menaces explorent de nouvelles pistes d'attaque, en mettant de plus en plus l'accent sur l'exploitation des vulnérabilités dans les systèmes orientés Internet, dont la plupart fonctionnent sur Linux.»*
>
> *ESET*

Le hurlement furtif de WolfsBane

WolfsBane est introduit dans les cibles via un compte-gouttes nommé «cron», qui laisse tomber le composant du lanceur déguisé en composant de bureau KDE.

En fonction des privilèges avec lesquels il s'exécute, il désactive SELinux, crée des fichiers de service système, ou modifie les fichiers de configuration utilisateur pour établir la persistance.

Le lanceur charge le composant de logiciel malveillant de confidentialité, «udvd», qui charge trois bibliothèques cryptées contenant sa fonctionnalité principale et sa configuration de communication de commande et de contrôle (C2).



Flux d'exécution de WolfsBane
Source : ESET

Enfin, une version modifiée du rootkit userland BEURK est chargée via '/etc/ld.so.preload' pour l'accrochage à l'échelle du système pour aider à dissimuler les processus, les fichiers et le trafic réseau liés aux activités de WolfsBane.

«Le rootkit WolfsBane Hider obtene de nombreuses fonctions de base de la bibliothèque C de normes telles que **l'ouverture**, la **statistique**, **la lecture dir** et **l'accès»,** explique ESET.

"Si ces fonctions accrochées invoquent les fonctions d'origine, elles filtrent tous les résultats liés au logiciel malveillant WolfsBane."

L'opération principale de WolfsBane est d'exécuter des commandes reçues du serveur C2 en utilisant des mappages de fonction de commande prédéfinis, qui est le même mécanisme que celui utilisé dans son homologue Windows.

Ces commandes incluent les opérations de fichiers, l'exfiltration des données et la manipulation du système, donnant au Gelsemium un contrôle total sur les systèmes compromis.

```
aLoadedPluginsC db 'loaded_plugins_command::response_read_command',0
                                        ; DATA XREF: init(void)+8E↑o
                align 8
aLoadedPluginsC_0 db 'loaded_plugins_command::response_write_data',0
                                        ; DATA XREF: init(void)+14C↑o
                align 8
aCommonInfoComm db 'common_info_command::response_read_command',0
                                        ; DATA XREF: init(void)+20C↑o
                align 8
aCommonInfoComm_0 db 'common_info_command::response_write_data',0
                                        ; DATA XREF: init(void)+2CC↑o
                align 8
aKeepAliveComma db 'keep_alive_command::response_read_command',0
                                        ; DATA XREF: init(void)+385↑o
                align 8
aKeepAliveComma_0 db 'keep_alive_command::response_write_data',0
                                        ; DATA XREF: init(void)+43C↑o
aDisablePluginC db 'disable_plugin_command::response_read_command',0
                                        ; DATA XREF: init(void)+4F5↑o
                align 20h
aDisablePluginC_0 db 'disable_plugin_command::response_write_data',0
                                        ; DATA XREF: init(void)+5AC↑o
                align 10h
aLoadPluginComm db 'load_plugin_command::response_read_command',0
                                        ; DATA XREF: init(void)+665↑o
                align 20h
aLoadPluginComm_0 db 'load_plugin_command::response_write_data',0
                                        ; DATA XREF: init(void)+71C↑o
                align 10h
aMainPluginSett db 'main_plugin_setting_command::response_read_command'
                                        ; DATA XREF: init(void)+7DC↑o
                align 8
aMainPluginSett_0 db 'main_plugin_setting_command::response_write_data'
                                        ; DATA XREF: init(void)+89C↑o
                align 20h
aConnectSetting db 'connect_setting_command::response_read_command',0
                                        ; DATA XREF: init(void)+955↑o
                align 10h
aConnectSetting_0 db 'connect_setting_command::response_write_data',0
                                        ; DATA XREF: init(void)+A0C↑o
                align 20h
aUninstallClien db 'uninstall_client_command::response_read_command',0
                                        ; DATA XREF: init(void)+AC5↑o
aUninstallClien_0 db 'uninstall_client_command::response_write_data',0
                                        ; DATA XREF: init(void)+B79↑o
```

```
; char aMainPluginSett_0[49]
aMainPluginSett_0 db 'main_plugin_setting_command::response_write_data'
                                        ; DATA XREF: DllMain(x,x,x)+841
                align 10h
; char aMainPluginSett[51]
aMainPluginSett db 'main_plugin_setting_command::response_read_command'
                                        ; DATA XREF: DllMain(x,x,x)+788
                align 4
; char aLoadPluginComm_0[41]
aLoadPluginComm_0 db 'load_plugin_command::response_write_data',0
                                        ; DATA XREF: DllMain(x,x,x)+6D8
                align 10h
; char aLoadPluginComm[43]
aLoadPluginComm db 'load_plugin_command::response_read_command',0
                                        ; DATA XREF: DllMain(x,x,x)+631
                align 4
; char aDisablePluginC_0[44]
aDisablePluginC_0 db 'disable_plugin_command::response_write_data',0
                                        ; DATA XREF: DllMain(x,x,x)+588
; char aDisablePluginC[46]
aDisablePluginC db 'disable_plugin_command::response_read_command',0
                                        ; DATA XREF: DllMain(x,x,x)+4D5
                align 4
; char aKeepAliveComma_0[40]
aKeepAliveComma_0 db 'keep_alive_command::response_write_data',0
                                        ; DATA XREF: DllMain(x,x,x)+423
; char aKeepAliveComma[42]
aKeepAliveComma db 'keep_alive_command::response_read_command',0
                                        ; DATA XREF: DllMain(x,x,x)+36E
                align 4
; char aCommonInfoComm_0[41]
aCommonInfoComm_0 db 'common_info_command::response_write_data',0
                                        ; DATA XREF: DllMain(x,x,x)+2BB
                align 4
; char aCommonInfoComm[43]
aCommonInfoComm db 'common_info_command::response_read_command',0
                                        ; DATA XREF: DllMain(x,x,x)+205
                align 4
; char aLoadedPluginsC_0[44]
aLoadedPluginsC_0 db 'loaded_plugins_command::response_write_data',0
                                        ; DATA XREF: DllMain(x,x,x)+14E
; char aLoadedPluginsC[46]
aLoadedPluginsC db 'loaded_plugins_command::response_read_command',0
                                        ; DATA XREF: DllMain(x,x,x)+AE↑
                align 10h
```

Noms de commande sur Linux (à gauche) et portes d'appui arrière Windows (à droite)
Source : ESET

Vue d'ensemble de FireWood

Bien que seulement vaguement lié au gésemium, FireWood est une autre porte dérobée Linux qui pourrait permettre des campagnes d'espionnage polyvalentes à long terme.

Ses capacités d'exécution de la commande permettent aux opérateurs d'effectuer des opérations de fichier, de l'exécution de la commande de shell, du chargement/déchargement de la bibliothèque et de l'exfiltration des données.

ESET a identifié un fichier nommé 'usbdev.ko', qui est suspecté d'être fonctionner comme un rootkit au niveau du noyau, fournissant à FireWood la capacité à cacher les processus.

Le logiciel malveillant définit sa persistance sur l'hôte en créant un fichier de démarrage automatique (gnome-control.desktop) dans '.config/autostart/', alors qu'il peut également inclure des commandes dans ce fichier pour les exécuter automatiquement au démarrage du système.

Une liste complète d'indicateurs de compromis associés aux deux nouvelles familles de logiciels malveillants Linux et les dernières campagnes de Gelsemium sont disponibles sur ce dépôt GitHub.