Cours d'introduction au reverse-engineering et mécanismes de sécurités des OS Windows
Évaluation

Alexandre Gazet, Julien Lenoir, Sylvian Peyrefitte, Raphaël Rigo

TLS-SEC 2020/2021

Abstract. La blue team de la société MoonBreeze vient de détecter un serveur compromis sur laquelle des artefacts ont pu être collecté. En tant que nouveau membre de leur équipe CERT et jeune professionnel surentraîné, il vous est demandé d'analyser les evidences et définir une réponse ciblée.

Souvenez-vous qu'il vous faudra regarder le code droit dans les yeux pour en percer les mystères.

Note importante: **toutes vos réponses doivent être justifiées. Pensez par exemple à expliciter l'emploi des outils qui vous paraissent appropriés et à décrire votre démarche. L'analyse et les manipulations de haute précision peuvent-être réalisées dans la VM fournie durant le cours.**

Cette évaluation est possible seul(e) ou en binôme. Vous nous ferez part de votre joie immense à résoudre ce challenge dans un rapport élégant, empreint d'une touche sobre et professionnelle. Ce dernier, ainsi que d'éventuels scripts/PoC développés par vos soins, sera à renvoyer avant le dimanche 14 Février 2021 minuit, dernier délai.

1 Ordre de mission

Prérequis: installez les redistribuables Visual Studio 2019 dans votre environnement Windows, disponibles ici[1].

Pour la suite de l'analyse, le binaire principale se nomme `apt-secure-up.exe`, vous le trouverez dans le répertoire `artefacts`.

1.1 Préliminaires

- Quel est le *timestamp* contenu dans l'en-tête du binaire ?
- Combien de sections sont présentes dans le binaire ? Combien contiennent du code exécutable ?
- Proposez une analyse rapide des fonctions **importées** par le binaire. Peut-on en déduire quelque chose ?
- Une des bibliothèque utilise un mécanisme d'import par ordinal, laquelle? Pouvez-vous résoudre ces fonctions (c'est à dire retrouver leur nom) ?
- De la même manière, proposez une analyse rapide des fonctions **exportées** par le binaire.

1.2 Reverse-engineering

Posez les bases et analysez les fonctions:

- `starter01` `starter02` en proposant un pseudo-code (ou réimplémentation en `C`/`Python` par exemple.)
- Quelles sont toutes les options supportées par la ligne de commande?
- Pour chacune d'entre elles indiquez le nombre d'arguments attendus ainsi que leurs sémantiques.

Tip: pour la suite, aidez-vous au maximum des informations présentes:

- De nombreux messages de debug/log semblent présents.
- Certains symboles (noms de fonctions) n'ont pas été retirés.
- Une fonction apparaissant en violet/rose dans `IDA` est une API exposée par une bibliothèque, donc documentée. Recherchez les constantes pour bien comprendre la sémantique des appels.

1.3 Analyse de la fonction `apt_func_001`

- Que fait la fonction `apt_func_001`?
- Quelles sont ses entrées/sorties?
- Donnez son prototype (avec des types explicites)

[1] https://support.microsoft.com/en-us/topic/the-latest-supported-visual-c-downloads-2647da03-1eea-4433-9aff-95f26a218cc0

1.4 Analyse de la fonction `apt_func_002`

Tip: Rappelez-vous de la question sur l'import par ordinal..

- En analysant les fonctions appelées depuis `apt_func_002`, donnez une description haut-niveau de `apt_func_002`.
- Quelles sont ses entrées/sorties?
- Donnez son prototype (avec des types explicites)

1.5 Analyse de la fonction `apt_func_003`

- Donnez une description haut-niveau de `apt_func_003`.
- La fonction externe `RAND_set_rand_method` semble du plus haut intérêt. Expliquez ce qu'elle fait.
- Quel argument prend elle ? À partir de la définition de son prototype (cf. la documentation), proposez un nom pour chacune des fonctions de la structure passée en argument.

1.6 Analyse du chiffrement

Le cœur du logiciel semble être la fonctionnalité d'archive chiffrée ou enveloppe (en mémoire). La seule information dont nous disposons est que tous les algorithmes de chiffrement utilisés semblent standards.

Détaillez la fonction `envelope_seal`:

1. Quels sont les algorithmes utilisés?
2. Comment est générée la clé de chiffrement?
3. L'algorithme de chiffrement met-il en œuvre un vecteur d'initialisation (IV)?
4. Création d'une archive chiffrée. Des metadata sont stockées dans un en-tête (ou header):
   - Quel est le format de l'en-tête ?
   - Proposez une structure `C` le décrivant.

Tip: pour l'analyse du mécanisme cryptographique, toutes les fonctions utilisées proviennent d'une bibliothèque bien connue et parfaitement documentée.
Tip2: pour reconstruire une structure on s'intéresse souvent à initialisation. Taille de l'allocation initiale puis les différents offsets accédés. *cf.* "IDA: Structures Tutorial"[2].

[2] `https://hex-rays.com/products/ida/support/tutorials/structs.shtml`

1.7 Data exfiltration

- Détaillez le mécanisme d'exfiltration des données.
- Afin d'aider le SOC à détecter toute présence de ce group d'attaquant sur le réseau et à analyser les logs de ces derniers mois collectez tous les artefacts réseau possible (identifiants, port, protocole, nom de domaine, etc.)
- Quel mécanisme est utilisé par l'attaquant pour rendre son activité plus furtive?
- Si la compromission initiale a eu lieu le 01.01.2020, quelle trace réseau devons nous rercherher dans les logs de ce jour ?

2 Strike back

Des rumeurs sont parvenues jusqu'à vous selon lesquelles les services secrets de certains pays seraient en mesure d'accéder au contenu de tout ou une partie des commnications réalisées au moyen de cet outil.

Evaluez la robustesse des mécanismes cryptographiques employés.

Une porte dérobée est-elle possible ?

Si oui, vous appliquerez vos talents de casseur de code à déchiffrer l'artefact *intercepted/data.dmp* et à en retrouver le contenu original.

3 Exploit in the wild

Récemment une vulnérabilité critique a touché les Active Directory, éléments centraux dans un réseau d'entreprises, se nommant ZeroLogon (aka CVE-2020-1472). Pouvez-vous expliquer les erreurs d'implémentation qui ont conduit à cette vulnérabilité.

Un exploit publique existe. Analysez-le, et explicitez son mode de fonctionnement.

Quelles sont les stratégies de détection qui peuvent être mis en place?

4 Outro

Vous nous ferez part de votre joie immense à résoudre ce challenge dans un rapport élégant, empreint d'une touche sobre et professionnelle. Ce dernier (ainsi que les scripts/PoC développés par vos soins) sera à renvoyer avant le 14 Février 2021 minuit, dernier délai.

Remember[3]:

[3] (Credit: Shutterstock/Fotokon)

Fig. 1. Keep fresh and praise the sun

A Tip: Ghidra

Quelques pointeurs pour Ghidra qui peuvent vous aider

- vous pouvez utiliser un serveur Ghidra pour travailler en parallèle et en collaboration
- pensez à bien redéfinir le type des fonctions pour améliorer l'analyse (y compris *varargs*)
- l'aide est très complète, *data structure editor* peut vous être utile

B Tip: Let Me Google That For You

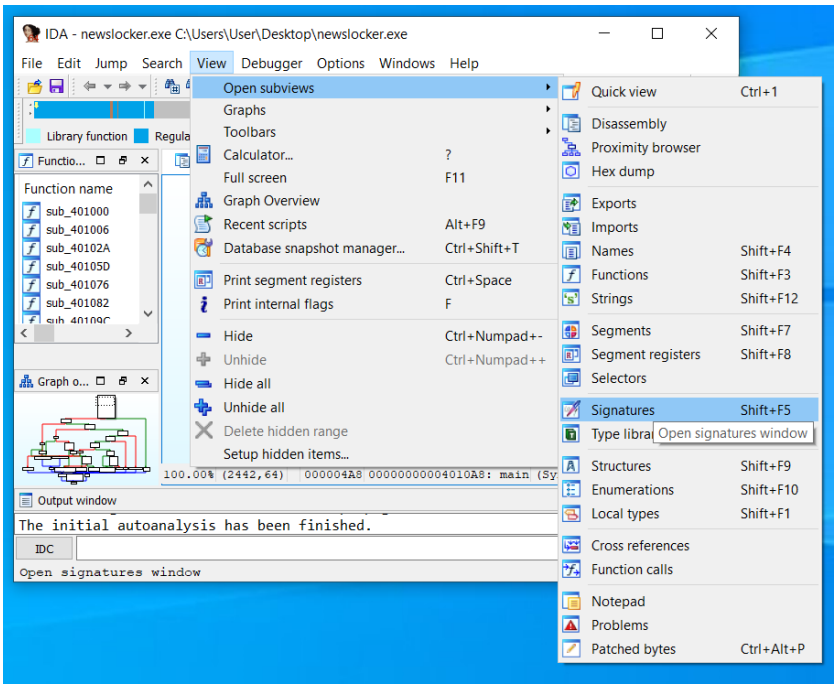
Les primitives cryptographiques utilisées dans le binaire sont standards et parfaitement documentées. N'hésitez pas à chercher sur Internet de exemples d'utilisation proches/similaires pour vous aider dans la compréhension du code, une fois une ou des fonction(s) intéressante(s) identifiée(s). Prenons les fonctions "`EVP_*`", une recherche "*Evp encrypt example*" aide grandement.

C'est la même méthodologie que pour les constantes spécifiques du `MD5` vues en TP.

C Tip: Signature

Bien que la version *free* d'`IDA Pro` que vous utilisez dispose d'un nombre restreint de banques de signatures de fonction, vous pouvez charger le fichier *vc64ucrt* afin de reconnaitre de permettre à `IDA` de reconnaitre quelques fonctions (elle apparaissent alors en bleu clair).

Enfin rappelez-vous la propriété de cohérence spatiale du code généré, si des fonctions ne sont pas reconnues mais qu'elles sont situées au milieu d'autres fonctions reconnues, il y a de très fortes chances pour que tout ce petit monde appartiennent à la même bibliothèque, et dans notre cas au même runtime.


IDA - newslocker.exe C:\Users\User\Desktop\newslocker.exe
File Edit Jump Search View Debugger Options Windows Help
Open subviews
Graphs
Toolbars
Calculator... ?
Full screen F11
Graph Overview
Recent scripts Alt+F9
Database snapshot manager... Ctrl+Shift+T
Print segment registers Ctrl+Space
Print internal flags F
Hide Ctrl+Numpad+-
Unhide Ctrl+Numpad++
Hide all
Unhide all
Delete hidden range
Setup hidden items...
Quick view Ctrl+1
Disassembly
Proximity browser
Hex dump
Exports
Imports
Names Shift+F4
Functions Shift+F3
Strings Shift+F12
Segments Shift+F7
Segment registers Shift+F8
Selectors
Signatures Shift+F5
Type libra
Open signatures window
Structures Shift+F9
Enumerations Shift+F10
Local types Shift+F1
Cross references
Function calls
Notepad
Problems
Patched bytes Ctrl+Alt+P
Function name
sub_401000
sub_401006
sub_40102A
sub_40105D
sub_401076
sub_401082
Library function
Output window
The initial autoanalysis has been finished.
IDC
Open signatures window