Documentation Projet Solarwinds CyberRange

Introduction

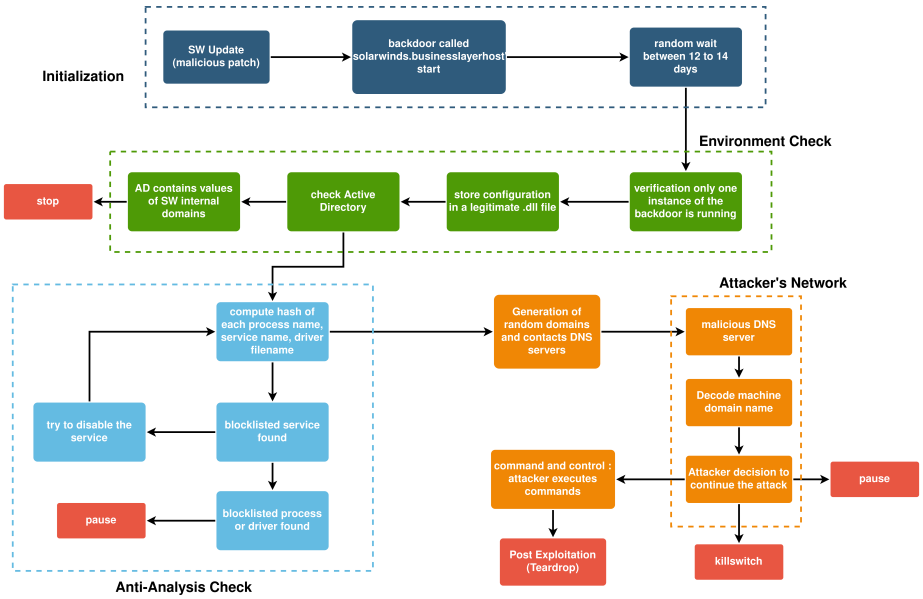


Figure 1: Description générale de Sunburst

Ce repository possède des dossiers pour chaque grande étape de l'attaque :

- 1-Initialization
- 2-EnvironmentCheck
- 3-Anti-AnalysisCheck
- 4-CommunicationC2Server

Les slides de la soutenance finale sont présentes dans le dossier *annex*. Notre implémentation est uniquement en python, nous avons créé des .exe de ce code python avec `auto-py-to-exe` avant d'exécuter le tout sur l'outil CyberRange d'Airbus.

Nous avons travaillé uniquement sur **Sunburst** et non sur la compromission initiale de SolarWind (cf slides de la soutenance finale).

Description technique

1-Initialization

Ce dossier possède 2 sous-dossiers :

- Obfuscation : ce dossier contient les icônes, licences et fonctions permettants d'obfusquer au mieux l'exécution du malware afin de le faire passer pour un logiciel légitime.
- OrionUpdate : cette partie correspond à la mise à jour initiale de Orion avec la version compromise. Pour cela, le code recherche le fichier

OrionUpdateRequest.conf qui contient l'IP d'un serveur distant puis télécharge le contenu et extrait les fichiers qui contiennent toutes les étapes suivantes du malware.

2-EnvironmentCheck

Ce dossier possède 2 sous-dossiers :

- CheckMalware : cela correspond à la vérification qu'une seule instance du malware est en exécution et que le fichier *SolarWinds.Orion.Core.BusinessLayer.dll.config* (qui sert à stocker des paramètres du malware) est bien présent. Un fichier *README.md* dans ce dossier apporte plus de précisions.
- DomainDiscovery : le malware vérifie que l'active directory présent ne contient pas certaines valeurs internes à Solarwinds afin d'être certain que le malware s'exécute chez un client de Solarwinds.

3-Anti-AnalysisCheck

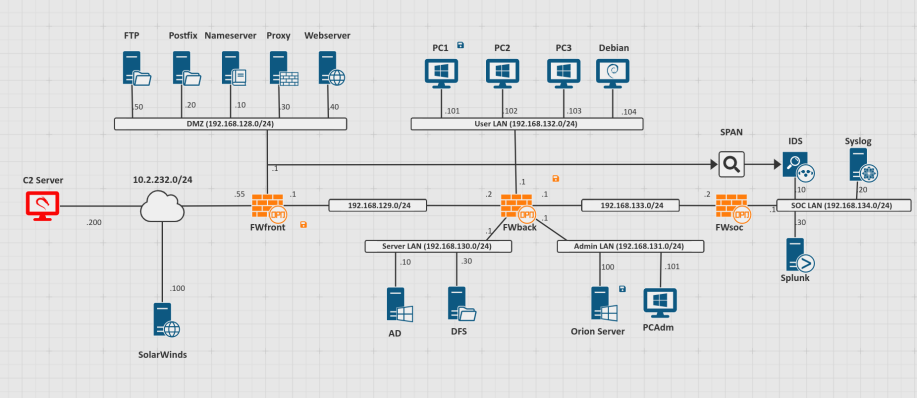
Ce dossier contient la partie chargée de détecter si un processus ou un service représente une menace pour le bon déroulement de l'attaque. Le programme liste les processus et services en cours sur l'hôte, et possède une liste hardcodée de programmes que le malware tentera d'éviter en tentant de les arrêter ou en stoppant l'attaque.

4-CommunicationC2Server

Ce dossier possède 3 sous-dossiers :

- Backdoor : ce code permet la communication entre le malware et le serveur C2 de l'attaquant à l'aide de requête DNS puis de requêtes HTTP. Toute la partie de commuunication par requêtes DNS a été commenté car nous n'avons pas implémenté cela dans Cyberrange (nous avons hard-codé l'IP de l'attaquant sur Cyberrange). Le serveur de l'attaquant est un serveur Flask et ce dernier précise les commandes qu'ils souhaite exécuter à distance dans le fichier *commands.csv* puis récupère les informations dans le fichier *c2_log.txt*. Les slides 19 à 26 de la soutenance finale apportent davantage de précisions. Un fichier *README.md* est également présent dans ce dossier *Backdoor*.
- DGA : ce dossier correspond à l'algorithme de génération des noms de domaine. Le domaine généré permet la première communication avec l'attaquant avec des requêtes DNS. Ce code n'a pas été utilisé dans notre implémentation sur Cyberrange.
- MinimalBackdoor : ce code nous a permis de tester qu'une backdoor minimaliste était possible sur Cyberrange.

Topologie sur Cyberrange



Dans cette modélisation nous avons choisi une topologie d'entreprise basée sur un Active Directory. Un serveur Orion est déployée sur un réseau dit *admin*.

L'attaque

Nous avons essayé de représenter fidèlement la logique de l'intrusion des attaquants dans le réseau de leurs victimes, en prenant en compte leur attention particulière à la discrétion. Puis nous avons implémenté nous-même une attaque réaliste une fois la backdoor créée.

Scénario

Une fois en possession d'une backdoor vers le serveur Orion de l'entreprise, l'attaque lance une reconnaissance sur le réseau sur lequel il est présent.

Implémentation des actions sur CyberRange

Afin de créer un scénario sur l'outil CyberRange, nous avons découpé l'attaque de manière logique afin de montrer visuellement et temporellement la chaîne de décision ainsi que l'attaque. Dans l'espace CyberRange dédié à l'attaque vous trouverez nos actions qui constitueront le coeur du scénario.

Les différentes actions :

- `Initialisation Orion` : Cette action est un pré-requis au scénario à jouer sur l'hôte qui servira de machine victime. Elle crée le répertoire `Orion` et `Orion\modules` à l'emplacement `C:\"Program Files (x86)"\Orion\`
- `Installation Orion` : Cette action est un pré-requis au scénario à jouer sur l'hôte qui servira de machine victime. Elle upload le fichier `Installation Orion.tar` et le désarchive dans le `C:\"Program Files (x86)"\Orion\`. Ne possédant pas la licence *Orion*, cette archive contient un programme qui jouera le rôle de couche graphique simulant le bon fonctionnement du logiciel ciblé. Cette archive contient :

  - `orion.exe` : un 'dummy program' dont le but est de symboliser le logiciel légitime de Solarwinds
  - un faux fichier de licence
  - le logo de SolarWinds
  - un faux fichier de configuration `OrionUpdateRequest.conf`
  - `update.exe` : programme légitime permettant de mettre à jour Orion

- `Update Orion` : Cette action est le départ de l'attaque. Elle exécute `orionUpdate.exe` qui télécharge les fichiers suivants dans le répertoire `C:/Program Files (x86)/Orion/modules/`
  - `environmentCheck.exe`
  - `DomainDiscovery.exe`
  - `antiAnalysisCheck.exe`
  - `Dns_backdoor.exe`
- `Environment Checking`: L'action exécute EnvironmentCheck.exe qui se situe dans le repertoire `C:/Program Files (x86)/Orion/modules/envCheck.exe`
- `Domain Discovery`: L'action exécute EnvironmentCheck.exe qui se situe dans le repertoire `C:/Program Files (x86)/Orion/modules/DomainDiscovery.exe`
- `Anti-Analysis Check`: L'action exécute Anti-AnalysisCheck.exe qui se situe dans le repertoire `C:/Program Files (x86)/Orion/modules/anti_analysis.exe`
- `Dns_backdoor`: L'action exécute Dns_backdoor.exe qui se situe dans le repertoire `C:/Program Files (x86)/Orion/modules/Dns_backdoor.exe`

Faire marcher la machine Orion :

Ci-dessous figurent les diverses instructions concernant le bon déroulement technique de l'attaque sur notre topologie.

Pare-feu : par manque de temps, les pare-feu ont été simplement désactivés dans la topologie fournie. Pour une meilleure représentation de la réalité, il serait préférable d'ajouter des règles de NAT ainsi que de whitelister les échanges provenant du serveur Orion.

Se connecter aux machines : il est possible de se connecter aux machines avec le compte administrateur de l'Active Directory en utilisant les credentials suivant :

- User : `SIGEN\Administrator`
- Password : `SIGEN\Administrator`

Redémarrage de l'AD : pour des raisons inconnues, les machines AD et DFS s'éteignent toutes les 2h30 environ. Pour que l'attaque fonctionne, la machine AD doit être rédémarrée et il faut se connecter avec les credentials ci-dessus.

Redémmarrage d'Orion : la machine Orion a aussi tendance a reboot sans raison particulière. Pour que l'attaque fonctionne, il faut procéder aux manipulations suivantes (il serait préférable à terme d'automatiser ces opérations) :

- Démarrer un shell Powershell
- Lancer la commande `Test-NetConnection sigen.net -port 9389`

- Vérifier que la connexion est opérationnelle avec la commande `Get-ADDomain -Identity sigen.net`
- *En cas d'échec* : Lancer la commande `Set-ExecutionPolicy RemoteSigned`

Commandes au LDAP : pour envoyer des commandes au LDAP, il faut :

- Avoir installé le module activedirectory sur la machine Orion (c'est déjà le cas sur la mâchine fournie). Sinon, lancer la commande `import-module activedirectory` dans un Powershell.
- Préciser l'identité à contacter en rajoutant `-Identity sigen.net`.

Bibliographie : principaux articles

- https://www.mandiant.com/resources/sunburst-additional-technical-details
- https://github.com/CyberSecOps/SolarWinds-Sunburst-Solorigate-Supernova-FireEye
- https://www.mandiant.com/resources/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor
- https://blog.cloudflare.com/a-quirk-in-the-sunburst-dga-algorithm/
- https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/solarwinds-unique-dga