Comprendre le programme First Responder (Secure Firewall Edition)

Contenu

Introduction

Ce document décrit l'utilisation et la mise en oeuvre du programme First Responder pour Cisco Secure Firewall.

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Components Used

Ce document est basé sur les produits Cisco Secure Firewall.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations générales

Le programme First Responder a été créé par le TAC pour faciliter et accélérer la fourniture de données de diagnostic pour les cas ouverts. Le programme se compose de deux éléments principaux :

E-mail automatisé

Cet e-mail est envoyé au début du dossier avec des instructions sur la façon de collecter et de télécharger des données de diagnostic pour l'analyse du TAC. Plusieurs technologies tirent parti de ce système, et chaque e-mail est associé à la « technologie » et à la « sous-technologie » choisies lors de la création du dossier.

Script / Commandes

Chaque mise en oeuvre du programme des premiers intervenants a sa propre façon de gérer la collecte et la livraison des données. Pour ce faire, la mise en oeuvre du pare-feu sécurisé utilise le script python firepower.py créé par le TAC. Le processus d'e-mail automatisé génère une commande d'une ligne, unique dans ce cas spécifique, qui peut être copiée et collée dans l'interface de ligne de commande des périphériques Secure Firewall à exécuter.

Raison de cet e-mail

Certaines technologies sont activées pour le programme des premiers intervenants. Cela signifie que chaque fois qu'un dossier est ouvert pour l'une de ces technologies activées, un e-mail de premier répondant est envoyé. Si vous recevez un e-mail de premier répondant et que vous ne pensez pas que la demande de données est pertinente, n'hésitez pas à ignorer la communication.

Pour l'exemple d'utilisation Secure Firewall, le programme de premier intervenant est limité au logiciel Firepower Threat Defense (FTD). Si vous exécutez un ASA (Adaptive Security Appliance) à base de code, ignorez cet e-mail. Étant donné que ces deux produits fonctionnent sur le même matériel, il est généralement observé que les cas ASA sont créés dans l'espace technologique Secure Firewall, qui génère l'e-mail du premier répondant.

E-mail automatisé

Voici un exemple d'e-mail automatisé envoyé dans le cadre de ce programme :

```
From: first-responder@cisco.com <first-responder@cisco.com>
Sent: Thursday, September 1, 2022 12:11 PM
To: John Doe <john.doe@cisco.com>
Cc: attach@cisco.com
Subject: SR 666666666 - First Responder Automated E-mail

Dear John,

In an effort to resolve your case faster it may be necessary to collect some diagnostic data
from your environment.
Based on the problem statement you provided, below are a few pieces of data that would help
speed the resolution and the steps to collect them:

*** Troubleshoot File ***

* Connect to the device using SSH
* Issue the command expert, skip this step for FMC version 6.4.x and earlier
* Issue the command sudo su
* When prompted for the password, enter your password.
* For FMC 6.4 or FTD 6.7 and later issue the command
curl -s -S https://cxd.cisco.com/public/ctfr/firepower.py | python - -c 666666666 -t
aBcDeFgHiJkLmNoP --auto-upload &

* For FMC 6.3 or FTD 6.6 and earlier issue the command
curl -k -s -S https://cxd.cisco.com/public/ctfr/firepower.py | python - -c 666666666 -t
aBcDeFgHiJkLmNoP --auto-upload &

For more information on what this command does, or to understand why you are receiving this e-
mail - please refer to
<LINK_TO_THIS_ARTICLE>

For 6.3 and earlier versions we recommend confirming cxd.cisco.com resolves to <CURRENT_CXD_IP1>
or
<CURRENT_CXD_IP2>. Furthermore, we recommend validating the SHA checksum of the file by running
url -s -k https://cxd.cisco.com/public/ctfr/firepower.py | shasum which should output
<CURRENT_SHA>.

If you are unable to upload troubleshooting files (or would prefer not to), please let us know
what
hardware and software version ou are running if you have not already.

Sincerely, First Responder Team
```

Les courriers électroniques automatisés destinés au programme de premier répondeur sont divisés en deux parties appelées bloc d'introduction et bloc de demande de données.

Bloc introduction

Le bloc d'introduction est une chaîne statique incluse dans chaque e-mail de premier répondant. Cette phrase introductive sert simplement à mettre en contexte le ou les blocs de demande de données. Voici un exemple de bloc d'introduction :

```
Dear <NAME>,

In an effort to resolve your case faster it may be necessary to collect some diagnostic data
from your environment.
Based on the problem statement you provided, below are a few pieces of data that would help
speed the resolution
and the steps to collect them:
```

bloc de demande de données

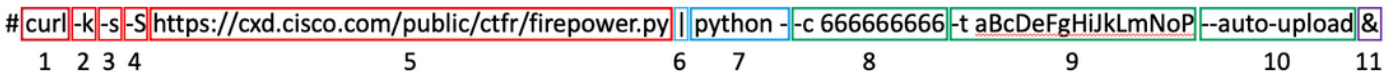
Les blocs de demande de données sont au coeur du programme de premier répondeur. Chaque bloc est un ensemble prédéfini d'étapes pour collecter des données pour une technologie donnée. Comme indiqué dans la section **Informations de base**, chaque bloc de demande de données est mappé à une technologie spécifique. Il s'agit de la même technologie que celle choisie pour ouvrir un dossier d'assistance. Généralement, le courrier électronique automatisé contient un seul bloc de demande de données. Cependant, si la technologie choisie a plus d'un bloc de demande de données mappé, alors plusieurs demandes de données sont incluses dans l'e-mail. Voici un exemple de format de bloc de demande de données avec plusieurs demandes de données :

```
*** <REQUEST NAME 1> ***

<REQUEST 1 STEPS>

*** <REQUEST NAME 2> ***

<REQUEST 2 STEPS>
```

Par exemple, dans le cas du pare-feu sécurisé, plusieurs blocs de demande de données sont souvent inclus lorsqu'une demande d'assistance est émise pour des problèmes de VPN d'accès à distance (RA-VPN) avec Firepower Threat Defense (FTD), car la technologie VPN dispose également d'un bloc de demande de données mappé configuré pour l'assistance pour la collecte des ensembles DART.

Commande générée

Pour l'exemple d'utilisation Secure Firewall en particulier, une commande unique d'une ligne est générée pour chaque cas dans le cadre de l'e-mail automatisé. Voici une répartition de la structure de la commande à une ligne :



1. La commande curl permet de télécharger la dernière version du script firepower.py
2. L'indicateur **-k** est une option permettant à curl d'ignorer les erreurs de certificat pendant la connexion.
3. L'indicateur **-s** est une option permettant à curl de s'exécuter en mode silencieux. Ceci est utilisé pour supprimer la sortie de boucle normale car elle est bruyante.
4. L'indicateur **-S** est une option permettant à curl d'afficher des erreurs. Ceci est utilisé pour forcer la boucle à toujours afficher les erreurs de sortie même avec l'option silent activée.
5. URL où la dernière version du script firepower.py est hébergée. Ce chemin indique à la commande curl d'extraire le dernier corps du script à exécuter.
6. Il s'agit d'un canal Linux, qui passe la sortie de la commande curl (le contenu d'un script python) à une instruction d'exécution dans l'étape suivante.
7. Àcette étape, le binaire python sur le périphérique est appelé avec un "-" supplémentaire. Cela indique à python que la source est tirée de stdin (puisque le contenu du script est redirigé depuis curl).
8. L'indicateur **-c** est un argument d'entrée pour le script firepower.py, qui indique le numéro de cas vers lequel les données doivent être téléchargées. La valeur 666666666 après cette option est l'exemple de numéro de cas.

9. L'indicateur **-t** est un argument d'entrée pour le script firepower.py, qui indique un jeton unique (mot de passe) qui a été généré pour ce cas particulier. La valeur aBcDeFgHiJkLmNoP après cette option est l'exemple de jeton pour ce cas.
10. L'indicateur **—auto-upload** est un argument spécial pour le script firepower.py, qui indique le script à exécuter en mode automation. Vous trouverez plus d'informations à ce sujet dans la section spécifique au script.
11. Le **&** demande à toute cette commande de s'exécuter en arrière-plan, ce qui permet à l'utilisateur de continuer à interagir avec son shell pendant l'exécution du script.

Note: L'indicateur -k est requis pour toute version de FMC antérieure à 6.4 et toute version de FTD antérieure à 6.7 puisque le certificat racine utilisé par CXD n'était pas approuvé par les périphériques Firepower jusqu'à la version 6.4 de FMC et la version 6.7 de FTD, ce qui entraîne l'échec de la vérification du certificat.

Script Firepower.py

L'objectif principal du script est de générer et de télécharger un ensemble de diagnostics à partir du périphérique Secure Firewall appelé « dépannage ». Pour générer ce fichier de dépannage, le script firepower.py appelle simplement le script intégré sf_troubleshoot.pl qui est responsable de la création de ce lot. Il s'agit du même script qui est appelé lorsque nous générons un dépannage à partir de l'interface graphique utilisateur. Outre le fichier de dépannage, le script peut également collecter d'autres données de diagnostic qui ne sont pas incluses dans l'ensemble de dépannage. À l'heure actuelle, les seules données supplémentaires qui peuvent être recueillies sont les fichiers de base, mais elles pourront être élargies à l'avenir si nécessaire. Le script peut être exécuté en mode "Automation" ou "Interactive" :

Automatisation

Ce mode est activé lorsque nous utilisons l'option "—auto-upload" lors de l'exécution du script. Cette option désactive les invites interactives, active la collecte des fichiers de base et télécharge automatiquement les données sur le boîtier. La commande d'une ligne générée par l'e-mail automatisé inclut l'option "—auto-upload".

Interactif

Il s'agit du mode d'exécution par défaut du script. Dans ce mode, l'utilisateur reçoit des invites lui demandant de confirmer s'il doit ou non collecter des données de diagnostic supplémentaires telles que des fichiers de base. Quel que soit le mode d'exécution, une sortie significative est imprimée à l'écran et consignée dans un fichier journal pour indiquer la progression de l'exécution des scripts. Le script lui-même est largement documenté via des commentaires de code en ligne et peut être téléchargé / révisé à l'adresse https://cxd.cisco.com/public/ctfr/firepower.py.

Résultat attendu du script

Voici un exemple d'exécution réussie du script :

```
root@ftd:/home/admin# curl -k -s -S https://cxd.cisco.com/public/ctfr/firepower.py | python - -c
666666666 -t aBcDeFgHiJkLmNoP --auto-upload &
[1] 26422
```

```
root@ftd:/home/admin#
`/var/common/first_responder_notify` successfully uploaded to 666666666
Running sf_troubleshoot.pl command to create a troubleshoot file...
Troubleshoot file successfully generated at /ngfw/var/common/results-08-30-2022--135014.tar.gz
Attempting to upload troubleshoot to case...
##################################################################################################
############################## 100.0%
`/ngfw/var/common/results-08-30-2022--135014.tar.gz` successfully uploaded to 666666666
Found the following core files:
(0 B) - /ngfw/var/common/core_FAKE1.gz
(0 B) - /ngfw/var/common/core_FAKE2.gz
(0 B) - /ngfw/var/common/core_FAKE3.gz
Successfully created /ngfw/var/common/cores_666666666-1661867858.tar.gz
Attempting core file upload...
##################################################################################################
############################## 100.0%
`/ngfw/var/common/cores_6666666660-1661867858.tar.gz` successfully uploaded to 666666666
FINISHED!
```

Veuillez noter que cet exemple de résultat inclut les téléchargements de fichiers de base. Si aucun fichier de base n'est présent sur votre périphérique, un message "No core files found. Skipping core file processing" est présenté à la place.

Problèmes courants

Voici quelques problèmes courants que vous pouvez rencontrer (dans l'ordre du processus / exécution) :

Sécurité de la messagerie électronique / Réécriture des URL

Souvent, il est observé que l'utilisateur final dispose d'un certain niveau de sécurité de la messagerie qui réécrit l'URL. Cette opération modifie la commande d'une ligne générée dans le cadre de l'e-mail automatisé. L'exécution échoue car l'URL d'extraction du script a été réécrite et n'est pas valide. Voici un exemple de la commande à une ligne attendue :

```
curl -s -S https://cxd.cisco.com/public/ctfr/firepower.py | python - -c 666666666 -t
aBcDeFgHiJkLmNoP --auto-upload &
```

Étapes à résoudre

Si l'URL dans la commande du courrier électronique est autre que "https://cxd.cisco.com/public/ctfr/firepower.py", alors l'URL a probablement été réécrite en cours de transfert. Pour résoudre ce problème, remplacez simplement l'URL avant d'exécuter la commande.

Échec DNS

Cette erreur de boucle est souvent visible lorsque le périphérique ne parvient pas à résoudre l'URL pour télécharger le script :

```
curl: (6) Could not resolve host: cxd.cisco.com
```

Étapes à résoudre

Pour résoudre ce problème, vérifiez les paramètres DNS sur le périphérique afin de vous assurer qu'il peut résoudre correctement l'URL pour continuer.

Échec d'ouverture/création du fichier journal

L'une des premières choses que le script tente de faire est de créer (ou d'ouvrir, s'il existe déjà) un fichier journal nommé **first-responder.log** dans le répertoire de travail courant. Si cette opération échoue, une erreur indiquant un problème d'autorisation simple s'affiche :

```
Permission denied while trying to create log file. Are you running this as root?
```

Dans le cadre de cette opération, toutes les autres erreurs sont identifiées et imprimées à l'écran dans le format suivant :

```
Something unexpected happened while trying to create the log file. Here is the error:
----------
```

```
----------
```

Étapes à résoudre

Pour corriger cette erreur, exécutez simplement le script en tant qu'utilisateur administratif, par exemple « admin » ou « root ».

Échec d'ouverture/écriture du fichier de notification

Dans le cadre de l'exécution du script, un fichier de 0 octet nommé « first_responder_notify » est créé sur le système. Ce fichier est ensuite téléchargé sur le boîtier dans le cadre de l'automatisation de ce programme. Ce fichier est écrit dans le répertoire "/var/common". Si l'utilisateur qui exécute le script ne dispose pas des autorisations suffisantes pour écrire des fichiers dans ce répertoire, le script affiche l'erreur suivante :

```
Failed to create file -> `/var/common/first_responder_notify`. Permission denied. Are you running as root?
```

Étapes à résoudre

Pour corriger cette erreur, exécutez simplement le script en tant qu'utilisateur administratif, par exemple « admin » ou « root ».

> **Note**: Si une erreur non liée aux autorisations est rencontrée, une erreur catch-all s'affiche à l'écran "Unexpected error while trying to open file -> `/var/common/first_responder_notify`. Please check first-responder.log file for full error". Le corps complet de l'exception se trouve dans le **fichier first-responder.log**.

Échec du verrouillage du fichier sf_troubleshoot.pid

Pour s'assurer qu'un seul processus de génération de dépannage est exécuté à la fois, le script de génération de dépannage tente de verrouiller le fichier /var/sf/run/sf_troubleshoot.pid avant de continuer. Si le script ne parvient pas à verrouiller le fichier, une erreur s'affiche :

```
Failed to run the `sf_troubleshoot.pl` command - existing sf_troubleshoot process detected.
Please wait for existing process to complete.
```

Étapes à résoudre

La plupart du temps, cette erreur signifie qu'une tâche de génération de dépannage distincte est déjà en cours. Parfois, cela est dû à des utilisateurs qui exécutent accidentellement la commande à une ligne deux fois de suite. Pour résoudre ce problème, attendez la fin du travail de génération de dépannage actuel et réessayez ultérieurement.

> **Note**: Si une erreur se produit dans le script sf_troubleshoot.pl, elle s'affiche à l'écran **"Unexpected PROCESS error while trying to run `sf_troubleshoot.pl` command. Please check first-responder.log file for full error".** Le corps complet de l'exception se trouve dans le **fichier first-responder.log**.

Problèmes de téléchargement

Il existe une fonction de téléchargement commune dans le script qui est responsable de tous les téléchargements de fichiers tout au long de l'exécution des scripts. Cette fonction est simplement un wrapper python pour exécuter une commande curl upload pour envoyer les fichiers au boîtier. De ce fait, toute erreur rencontrée lors de l'exécution est renvoyée sous forme de code d'erreur curl. En cas d'échec du téléchargement, cette erreur s'affiche à l'écran :

```
[FAILURE] Failed to upload `/var/common/first_responder_notify` to 666666666. Please check the
first-responder.log file for the full error
```

Consultez le fichier **first-responder.log** pour voir l'erreur complète. En général, le fichier first-responder.log ressemble à ceci :

```
08/29/2022 06:51:57 PM - WARNING - Upload Failed with the following error:
----------
Command '['curl', '-k', '--progress-bar',
'https://666666666:aBcDeFgHiJkLmNoP@cxd.cisco.com/home/',
'--upload-file', '/var/common/first_responder_notify']' returned non-zero exit status 6
----------
```

Étapes à résoudre

Dans ce cas, curl a retourné un état de sortie de **6** qui signifie **" Impossible de résoudre l'hôte "**. Il s'agit d'une simple erreur DNS pendant que nous essayons de résoudre le nom d'hôte **cxd.cisco.com** . Reportez-vous à la documentation curl pour décoder les états de sortie inconnus.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.