Rapport technique externe de l'analyse des causes profondes (RCA) — Channel File 291

INTRODUCTION

Ce rapport développe les informations précédemment partagées dans notre rapport préliminaire post-incident (PIR), en approfondissant les constatations, les mesures de remédiation prises, les détails techniques et l'analyse des causes profondes de l'incident. En date du 29 juillet à 17 heures PDT, en utilisant une comparaison d'une semaine sur l'autre, ~99 % des agents Windows étaient en ligne par rapport leur état d'avant la mise à jour du contenu. Nous observons généralement une variation de ~1 % d'une semaine à l'autre dans les connexions des agents.

Tout au long de cette analyse RCA, nous avons utilisé une terminologie générale pour décrire la plateforme CrowdStrike Falcon afin d'en faciliter la lecture.

QUE S'EST-IL PASSE ?

L'agent CrowdStrike Falcon fournit de puissants modèles d'IA et de Machine Learning (ML) pour protéger les systèmes des clients en identifiant et en remédiant aux dernières cybermenaces avancées. Ces modèles sont tenus à jour et renforcés grâce aux enseignements tirés de la dernière télémétrie des cybermenaces de l'agent et des renseignements humains des ingénieurs en détection des cybermenaces des équipes Falcon Adversary OverWatch, Falcon Complete et CrowdStrike. Cet ensemble de télémétrie de sécurité commence par le filtrage des données et leur agrégation sur chaque agent dans un graphe local

Chaque agent corrèle le contexte de son graphe local avec l'activité en temps réel du système pour trouver des comportements et des indicateurs d'attaque (IOA) dans un processus d'amélioration continu. Ce processus inclut un Sensor Detection Engine combinant le Sensor Content au Rapid Response Content diffusé depuis le cloud. Le Rapid Response Content est utilisé pour collecter des données de télémétrie, identifier les indicateurs de comportement des cyberadversaires et ajouter de nouvelles détections et préventions sur l'agent sans besoin de modifier son code. Le Rapid Response Content utilise de l'analyse comportementale heuristique, séparé et distinct des fonctionnalités en IA de prévention et de détection sur l'agent CrowdStrike.

Le Rapid Response Content est diffusé via des Channel Files et interprété par le « Content Interpreter » de l'agent, à l'aide d'un moteur basé sur des expressions régulières. Chaque Channel File de Rapid Response Content est associé à un « Template Type » spécifique intégré dans la version de l'agent. Le « Template Type » fournit au « Content Interpreter »

des données d'activité et un graphe de contexte à comparer au « Rapid Response Content ».

Avec la sortie de la version 7.11 de l'agent en février 2024, CrowdStrike a introduit un nouveau « Template Type » pour permettre la visibilité et la détection de nouvelles techniques d'attaque qui abusent des « Named Pipes » et d'autres mécanismes de communication interprocessus (« IPC ») de Windows. Comme indiqué dans le PIR, le nouveau « Template Type » IPC a été développé et testé conformément à nos processus standards de développement du « Sensor Content » des agents et a été intégré à l'agent pour être utilisé sur le terrain. Les Template Instances IPC sont fournies sous forme de « Rapid Response Content » aux agents via un Channel File correspondant numéroté 291.

Le nouveau « Template Type » définissait un paramètre de 21 champs de données, mais le code d'intégration qui faisait appel au « Content Interpreter » avec les « Template Instances » du Channel File 291 ne fournissait que 20 champs à comparer. Cette inadéquation du nombre dans les paramètres est passé inaperçu à de multiples niveaux de validation et de tests, car elle n'a pas été découverte lors du processus de test de lancement sur l'agent, ni lors des tests de résistance (stress tests) du « Template Type » (à l'aide d'une « Template Instance » de test) ni lors des premiers déploiements réussis de Template Types IPC sur le terrain. Cela était dû en partie à l'utilisation de critères de correspondance génériques pour le 21ème champ lors des tests et dans les Template Instances IPC initiaux.

Le 19 juillet 2024, deux « Template Instances » IPC supplémentaires ont été déployées. L'une d'entre elles a introduit un critère de correspondance non générique pour le 21ème paramètre de données. Ces nouvelles « Template Instances » ont donné naissance à une nouvelle version du Channel File 291 obligeant l'agent à inspecter le 21ème paramètre. Jusqu'à ce que ce Channel File soit livré aux agents, aucune Template Instance IPC dans les versions Channel précédentes n'avait utilisé le 21ème champ de donnée. Le Content Validator a évalué les nouvelles Template Instances, mais est parti du principe que le Template Type IPC serait fourni avec 21 champs.

Les agents qui ont reçu la nouvelle version de Channel File 291 comportant le contenu problématique ont été exposés à un problème latent de lecture hors limites dans le « Content Interpreter ». Lors de la notification IPC suivante émanant du système d'exploitation, les nouvelles Template Instances IPC ont été évaluées, en spécifiant une comparaison avec le 21ème champ. Le « Content Interpreter » n'attendait que 20 valeurs. Par conséquent, la tentative d'accès à la 21ème valeur a produit une lecture de mémoire hors limites au-delà de la fin du tableau de données et a entraîné une panne du système.

En résumé, c'est l'association de ces problèmes qui a entraîné un crash du système : l'inadéquation entre les 21 champs validés par le Content Validator et les 20 fournis au Content Interpreter, le problème latent de lecture hors limites dans le Content Interpreter, et

l'absence d'un test spécifique pour les critères de correspondance non génériques dans le 21ème champ. Bien que ce scénario avec le Channel File 291 ne puisse plus se reproduire, il explique également les améliorations des processus et les mesures de remédiation que CrowdStrike est en train de déployer afin de renforcer encore sa résilience.

CONSTATATIONS ET MESURES DE REMEDIATION

1. Le nombre de champs dans le Template Type IPC n'a pas été validé au moment de la compilation de l'agent

Constatations : Au moment de l'incident, le code de l'agent pour le Template Type IPC décrivait 20 sources d'entrée différentes à utiliser par la Template Instance. Cela signifie que lorsque l'agent voulait prendre une décision de détection basée sur le Template Type IPC, le code de l'agent allait fournir 20 champs de données différents au Content Interpreter. Cependant, la définition du Template Type IPC dans le fichier de définitions des Template Types indiquait qu'il attendait 21 champs. Cette définition a donné lieu à des Template Instances dans le Channel File 291 qui s'attendaient à fonctionner sur 21 champs. Cette non-concordance n'a pas été détectée lors du développement du Template Type IPC. Les tests et le Rapid Response Content utilisés pour valider le Template Type IPC n'ont pas déclenché d'erreur pendant le développement de la fonctionnalité ou pendant les tests de la version 7.11 de l'agent.

Remédiation : Valider le nombre de champs dans le Template Type au moment de la compilation de l'agent

Un correctif pour le Sensor Content Compiler qui valide le nombre de données fournies par un Template Type a été développé le 19 juillet 2024 et est entré en production le 27 juillet 2024, dans le cadre des outils de création internes de CrowdStrike. Le patch du Sensor Content Compiler a également permis de vérifier qu'aucun autre Template Type ne fournissait un nombre de champs incorrect, quelle que soit la plateforme.

2. Une vérification manquante des limites des tableaux à l'exécution pour les champs du Content Interpreter sur le Channel File 291

Constatations : Le Rapid Response Content pour le Channel File 291 a indiqué au « Content Interpreter » de lire le 21ème champ du pointeur de tableau. Toutefois, le Template Type IPC ne générait que 20 champs. Par conséquent, une fois le Rapid Response Content livré avec critère de correspondance non générique pour le 21ème champ, le « Content Interpreter » a effectué une lecture hors limites du tableau de données.

Il ne s'agit pas d'un problème d'écriture arbitraire de mémoire et cela a été examiné indépendamment.

Remédiation : ajouter des vérifications des limites du tableau à exécution du Content Interpreter pour le Rapid Response Content dans le Channel File 291

Le 25 juillet 2024, la vérification des limites a été ajoutée à la fonction du « Content Interpreter » qui reçoit des chaînes de caractères. Une vérification supplémentaire selon laquelle la taille du tableau de données correspond au nombre de champs attendu par le Rapid Response Content a été ajoutée en même temps. Ces correctifs sont rétro-portés vers toutes les versions 7.11 et supérieures des agents Windows par le biais d'un correctif du logiciel des agents. Cette version sera disponible d'ici le 9 août 2024 (GA).

L'ajout du contrôle des limites empêche le Content Interpreter d'accéder hors limites au tableau de données et de faire crasher le système. La vérification supplémentaire ajoute un niveau supplémentaire de validation d'exécution indiquant que la taille du tableau de données correspond au nombre de champs attendu par le Rapid Response Content.

Nous avons achevé des tests de fuzzing sur le Template Type du Channel 291 et nous sommes en train de d'étendre cela à d'autres gestionnaires de Rapid Response Content intégrés à l'agent.

Remédiation : Corriger le nombre de champs fournis par le Template Type IPC

Le code de l'agent définissant le Template Type IPC a été mis à jour pour fournir le bon nombre de champs (21). Ce correctif est rétro-porté vers toutes les versions 7.11 et supérieures des agents Windows via un correctif logiciel de l'agent. Cette version sera disponible d'ici le 9 août 2024 (GA).

3. Les tests de Template Types devraient couvrir une plus grande variété de critères de correspondance

Constatations : Des tests manuels et automatisés ont été effectués pendant le développement du Template Type IPC. Ces tests se sont concentrés sur la validation fonctionnelle du Template Type, y compris le flux correct des données relatives à la sécurité et l'évaluation de ces données pour générer des alertes de détection appropriées basées sur les critères créés dans les tests de développement.

Les tests automatisés ont tiré parti d'outils internes et externes pour créer les données relatives à la sécurité requises pour utiliser le Template Type IPC sur toutes les versions de Windows prises en charge, dans le cadre d'un large ensemble de cas d'usage opérationnels possibles. Pour les tests automatisés, un ensemble statique de 12 cas de test a été sélectionné pour être représentatif des attentes opérationnelles générales et pour valider la

création d'alertes de télémétrie et de détection. Une partie de ces tests comprenait la définition d'un Channel File à utiliser dans les cas testés. La sélection des données dans le Channel File a été effectuée manuellement et incluait un critère de correspondance générique regex dans le 21ème champ pour toutes les Template Instances, ce qui signifie que l'exécution de ces tests pendant les versions de développement et de publication n'a pas montré de lecture hors limites dans le Content Interpreter lorsqu'il était fourni avec 20 plutôt que 21 champs de données.

Remédiation : Augmenter la couverture des tests pendant le développement du Template Type

Pour confirmer que nous validons tous les champs de chaque Template Type, des tests automatisés ont été créés qui utilisent des critères de correspondance autres que des caractères génériques pour chaque champ. Cette étape a été effectuée pour tous les Template Types existants et est requise pour tous les futurs Template Types. De plus, tous les futurs Template Types incluront des tests avec des scénarios supplémentaires qui reflètent mieux l'utilisation en production.

4. Le Content Validator contenait une erreur de logique

Constatations : Le Content Validator a évalué les nouvelles Template Instances. Cependant, il a basé son évaluation sur l'hypothèse selon laquelle le Template Type IPC serait fourni avec 21 champs. La Template Instance problématique a donc été envoyée au Content Interpreter.

Remédiation : créer des contrôles supplémentaires dans le Content Validator

Le Content Validator est en cours de modification pour ajouter de nouvelles vérifications afin de garantir que le contenu des Template Instances n'inclut pas de critères de correspondance qui correspondent à plus de champs que ceux fournis en entrée à le Content Interpreter. Ce correctif sera mis en production d'ici le 19 août 2024.

Remédiation : empêcher la création de Channel Files 291 problématiques

Le Content Validator a été modifié pour n'autoriser que les critères de correspondance par caractères génériques dans le 21ème champ, ce qui empêche l'accès hors limites aux agents qui ne fournissent que 20 champs.

5. La validation de la Template Instance doit être étendue pour inclure les tests dans le Content Interpreter

Constatations : Les nouveaux Template Types sont soumis à de nombreux tests et validations, tels que l'utilisation des ressources, l'impact sur les performances du système et le volume de détection. Pour de nombreux Template Types, y compris le Template Type IPC, une Template Instance spécifique est utilisée pour tester la résistance du Template Type en le comparant à toutes les données associées possibles afin d'identifier les interactions indésirables du système.

Un test de résistance (stress test) du Template Type IPC avec une Template Instance de test a été exécuté dans notre environnement de test, qui comprend une variété de systèmes d'exploitation et de workloads. Le Template Type IPC a réussi le test de résistance et a été validé pour son utilisation, et une Template Instance a été mise en production dans le cadre d'une mise à jour du Rapid Response Content.

Cependant, la Template Instance testée par le Content Validator n'a pas remarqué qu'un nombre inégal de champs provoquait une panne du système lorsqu'il était fourni au Content Interpreter par le Template Type IPC.

Remédiation : mettre à jour les procédures de test du système de configuration du contenu

Le système de configuration de contenu a été mis à jour avec de nouvelles procédures de test pour garantir que chaque nouvelle Template Instance est testée, peu importe que la Template Instance initiale soit testée avec le Template Type lors de la création. Cela fournit aux Template Instances des tests supplémentaires avant le déploiement en production.

6. Les Template Instances devraient être déployées par étapes

Constatations : chaque Template Instance doit être déployée dans le cadre d’un déploiement par étapes.

Remédiation : le système de configuration de contenu a été mis à jour avec des étapes de déploiement et des contrôles de validation supplémentaires.

Le déploiement par étapes atténue l'impact si une nouvelle Template Instance provoque des pannes comme des crash du système, des pics de volume de détection de faux positifs ou des problèmes de performances. Les nouvelles Template Instances qui ont réussi les tests Canary doivent être poussées successivement vers des phases de déploiement plus larges ou annulées si des problèmes sont détectés. Chaque étape est conçue pour identifier et atténuer les problèmes potentiels avant un déploiement plus large. Le passage d'une Template Instance vers le niveau suivant est suivi d'un temps de préparation supplémentaire, au cours duquel la télémétrie est collectée pour déterminer l'impact global de la Template Instance sur l'endpoint.

Remédiation : Fournir au client un contrôle sur le déploiement des mises à jour du Rapid Response Content

La plateforme Falcon a été mise à jour pour offrir aux clients un contrôle accru sur la fourniture du Rapid Response Content. Ils peuvent choisir où et quand les mises à jour du Rapid Response Content seront déployées. Nous continuons à améliorer cette fonctionnalité afin de fournir un contrôle plus précis sur les déploiements du Rapid Response Content, ainsi que sur les informations relatives à la mise à jour du contenu via les notifications de publication, auxquelles les clients peuvent s'abonner.

ANALYSE PAR UN FOURNISSEUR TIERS INDEPENDANT

CrowdStrike a engagé deux fournisseurs de logiciels de sécurité tiers indépendants pour procéder à un examen plus approfondi du code de l'agent Falcon à la fois pour la sécurité et l'assurance qualité. De plus, nous menons un contrôle de qualité indépendant sur la totalité des processus d'élaboration de nos logiciels, du développement au déploiement. Les deux fournisseurs ont entamé leurs analyses en se concentrant dans l’immédiatsur le code et le processus impactés le 19 juillet.

DETAILS TECHNIQUES

Contexte et terminologie

CrowdStrike fournit des mises à jour de configuration de contenu de sécurité à nos agents de deux manières : le contenu de l'agent, directement livré avec notre agent, et le Rapid Response Content, conçu pour répondre à l'évolution des cybermenaces, le plus rapidement possible.

Le traitement du Rapid Response Content basé sur le regex de l'agent implique plusieurs éléments :

- Le Content Interpreter **:** Partie du code C++ de l'agent qui peut tester les les flux de données entrants par rapport au regex.
- **Template Types :** contiennent des champs prédéfinis que les ingénieurs en détection des cybermenaces peuvent exploiter dans le Rapid Response Content. Les Template Types sont exprimés en code et compilés dans l'agent au moment du Build.
- **Fichier de définitions des Template Types :** définit les paramètres de chaque Template Type. Les définitions de ce fichier incluent des informations sur le Channel File qui fournira le Rapid Response Content pour chaque Template Type, le nombre

de champs que le Template Type est censé utiliser et le type de données requis pour chaque champ.

- **Sensor Content :** Détermine comment combiner les données pertinentes pour la sécurité avec le Rapid Response Content afin de prendre certaines décisions de détection. Le Sensor Content comprend l'IA sur l'agent et les modèles de Machine Learning ainsi que les Template Types. Il est compilé dans le cadre de la mise à disposition de la version de l'agent.
- **Template Instances :** Critères de correspondance développés par les ingénieurs en détection. Les Template Instances consistent en un contenu regex destiné à être utilisé avec un Template Type spécifique. Les Template Instances identifient des données spécifiques à utiliser dans les opérations de sécurité. Les Template Instances sont définies à l'aide d'une interface utilisateur pilotée par le fichier de définition des Template Types.
- **Rapid Response Content :** Consiste en plusieurs Template Instances regroupées ensemble. Le Rapid Response Content est livré par Channel File.
- **Content Validator** : vérifie la validité des Channel Files par rapport à leur définition dans le fichier de définitions des Template Types.
- **Système de configuration de contenu** : Utilisé pour créer des Template Instances, qui sont validées et déployées sur l'agent via un mécanisme appelé **Channel Files**.

Utilisation du Kernel Driver dans un produit de sécurité

Comme l'a souligné David Weston sur le blog Microsoft Security, les produits de sécurité de l'écosystème Windows, y compris l'agent Falcon, exploitent généralement les kernel drivers en tant que composants essentiels d'une offre de sécurité robuste.

La présence dans le kernel offre une ample visibilité sur les activités liées à la sécurité à l'échelle du système, telles que la création de processus et de threads, ou les fichiers en cours d'écriture, de suppression et de modification sur le disque. Les interfaces exposées par le kernel permettent aux drivers de CrowdStrike d'appliquer des contrôles critiques pour un produit de sécurité, tels que la prévention en ligne des processus malveillants ou le blocage des fichiers malveillants en cours d'écriture sur le disque.

Le kernel driver de CrowdStrike est chargé dès le démarrage du système pour permettre à l'agent d'observer et de se défendre contre les malwares lancés avant le démarrage des processus en mode utilisateur.

L'ajout de contenu de sécurité actualisé (par exemple, le contenu Rapid Response de CrowdStrike) à ces fonctionnalités du kernel permet à l'agent de défendre les systèmes contre les cybermenaces en évolution rapide sans modifier le code du kernel. Le Rapid Response Content est constitué de données de configuration ; il ne s'agit pas d'un code ou d'un kernel driver.

CrowdStrike certifie chaque nouvelle version d'un agent Windows par le biais du programme Windows Hardware Quality Labs (WHQL), qui comprend des tests approfondis parmi tous ceux requis dans les kits Windows Hardware Lab (HLK) et Windows Hardware Certification Kit (HCK) de Microsoft. Le processus de certification WHQL marque la fin d'une série complète de tests internes comprenant des tests fonctionnels, des tests de longévité, des tests de résistance avec injection de défauts, des tests de fuzzing et des tests de performance. Lors des tests requis pour le programme WHQL, les agents utilisent les dernières versions des Channel Files au moment de la certification.

Alors que les nouvelles versions de Windows prennent en charge un plus grand nombre de ces fonctions de sécurité dans l'espace utilisateur, CrowdStrike met à jour son agent pour qu'il utilise ce support. Il reste encore beaucoup à faire pour que l'écosystème Windows puisse proposer un produit de sécurité robuste qui ne repose pas sur un kernel driver pour certaines de ses fonctionnalités. Nous nous engageons à collaborer directement et en continu avec Microsoft, à mesure que Windows ajoute davantage de support des besoins en matière de produits de sécurité dans l'espace utilisateur.

Analyse de Crash Dump

Pour illustrer comment les Template Instances dans le Channel File 291 ont entraîné une panne du système, examinons brièvement un Crash Dump du kernel d'un système affecté par le contenu problématique. Cela s'étend à l'analyse de crash partagé par David Weston sur le blog de sécurité Microsoft.

En ouvrant le crash dump dans le débogueur du kernel Windows et en utilisant la commande standard !analyze -v pour un résumé rapide, nous voyons qu'une faille de mémoire (également connue sous le nom de « violation d'accès ») s'est produite. *(Remarque : les détails de débogage non liés sont omis par souci de concision, et un fichier d'erreur représentatif est analysé ici. Il existe des variantes de ce fichier, en fonction des détails de l'état de la machine).*

```
1: kd> !analyze -v
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

PAGE_FAULT_IN_NONPAGED_AREA (50)
Invalid system memory was referenced.  This cannot be protected by try-except.
Typically the address is just plain bad or it is pointing at freed memory.
Arguments:
Arg1: ffffd6030000006a, memory referenced.
Arg2: 0000000000000000, X64: bit 0 set if the fault was due to a not-present PTE.
        bit 1 is set if the fault was due to a write, clear if a read.
        bit 3 is set if the processor decided the fault was due to a corrupted PTE.
        bit 4 is set if the fault was due to attempted execute of a no-execute PTE.
        - ARM64: bit 1 is set if the fault was due to a write, clear if a read.
```

```
        bit 3 is set if the fault was due to attempted execute of a no-execute PTE.
Arg3: fffff8020ebc14ed, If non-zero, the instruction address which referenced the bad memory
        address.
Arg4: 0000000000000002, (reserved)

READ_ADDRESS:  ffffd6030000006a Paged pool

MM_INTERNAL_CODE:  2

IMAGE_NAME:  csagent.sys

MODULE_NAME: csagent

FAULTING_MODULE: fffff8020eae0000 csagent

PROCESS_NAME:  System

TRAP_FRAME:  ffffae035f57eca0 -- (.trap 0xffffae035f57eca0)
NOTE: The trap frame does not contain all registers.
Some register values may be zeroed or incorrect.
rax=ffffae035f57f280 rbx=0000000000000000 rcx=0000000000000003
rdx=ffffae035f57f250 rsi=0000000000000000 rdi=0000000000000000
rip=fffff8020ebc14ed rsp=ffffae035f57ee30 rbp=ffffae035f57ef30
 r8=ffffd6030000006a  r9=0000000000000000 r10=0000000000000000
r11=0000000000000014 r12=0000000000000000 r13=0000000000000000
r14=0000000000000000 r15=0000000000000000
iopl=0         nv up ei ng nz na po nc
csagent+0xe14ed:
fffff802`0ebc14ed 458b08          mov     r9d,dword ptr [r8] ds:ffffd603`0000006a=????????
Resetting default scope

STACK_TEXT:
ffffae03`5f57ea78 fffff802`05add2da     : 00000000`00000050 ffffd603`0000006a 00000000`00000000
ffffae03`5f57eca0 : nt!KeBugCheckEx
ffffae03`5f57ea80 fffff802`05947efc     : ffffd603`000ed454 00000000`00000000 00000000`00000000
ffffd603`0000006a : nt!MiSystemFault+0x1bc19a
ffffae03`5f57eb80 fffff802`05a2707e     : 00000000`00000000 ffffd603`e33a019e ffffae03`5f57f0a0
ffffae03`5f57f0a0 : nt!MmAccessFault+0x29c
ffffae03`5f57eca0 fffff802`0ebc14ed     : 00000000`00000000 ffffae03`5f57ef30 ffffd603`f208200c
ffffd603`f207a05c : nt!KiPageFault+0x37e
ffffae03`5f57ee30 fffff802`0eb9709e     : 00000000`00000000 00000000`e01f008d ffffae03`5f57f202
fffff802`0ed6aaf8 : csagent+0xe14ed
ffffae03`5f57efd0 fffff802`0eb98335     : 00000000`00000000 00000000`00000010 00000000`00000002
ffffd603`f207a01c : csagent+0xb709e
ffffae03`5f57f100 fffff802`0edd20c7     : 00000000`00000000 00000000`00000000 ffffae03`5f57f402
00000000`00000000 : csagent+0xb8335
ffffae03`5f57f230 fffff802`0edcec44     : ffffae03`5f57f6e8 fffff802`060abae0 ffffd603`ed408580
00000000`00000003 : csagent+0x2f20c7
ffffae03`5f57f4b0 fffff802`0eb47a31     : 00000000`0000303b ffffae03`5f57f770 ffffd603`edc908a0
ffffc189`7fcd4098 : csagent+0x2eec44
ffffae03`5f57f670 fffff802`0eb46aee     : ffffd603`edc908a0 fffff802`0ebf1e7e 00000000`00006820
fffff802`0ed3f8f0 : csagent+0x67a31
ffffae03`5f57f7e0 fffff802`0eb4685b     : ffffae03`5f57fa58 ffffd603`edc97830 ffffd603`edc908a0
ffffc189`7f90f4b8 : csagent+0x66aee
ffffae03`5f57f850 fffff802`0ebe99ea     : 00000000`f047f4ef ffff49ac`ca0f55d4 00000000`00000000
ffffd603`ec18fc30 : csagent+0x6685b
ffffae03`5f57f8d0 fffff802`0eb3efbb     : 00000000`00000000 ffffae03`5f57fad9 ffffc189`7f90f010
ffffc189`7f7ea470 : csagent+0x1099ea
```

```
ffffae03`5f57fa00 fffff802`0eb3edd7     : ffffc189`7ab79000 00000000`00000000 ffffc189`7f90f010
ffffc189`00000001 : csagent+0x5efbb
ffffae03`5f57fb40 fffff802`0ebde681     : 00000000`00000000 00000000`00000000 ffffc189`7f5a97d0
ffffc189`7f7ea470 : csagent+0x5edd7
ffffae03`5f57fb70 fffff802`05879ca7     : ffffc189`7faa8040 00000000`00000080 fffff802`0ebde510
00000000`00000000 : csagent+0xfe681
ffffae03`5f57fbb0 fffff802`05a1af64     : ffffe601`bcf51180 ffffc189`7faa8040 fffff802`05879c50
00000000`00000000 : nt!PspSystemThreadStartup+0x57
ffffae03`5f57fc00 00000000`00000000     : ffffae03`5f580000 ffffae03`5f579000 00000000`00000000
00000000`00000000 : nt!KiStartSystemThread+0x34
```

Cette commande de triage automatique identifie `csagent.sys` comme le driver effectuant l'accès à la mémoire hors limites. `csagent.sys` est le driver de CrowdStrike filtrant le système de fichiers, un type de kernel driver qui s'enregistre sur des composants du système d'exploitation Windows pour recevoir des notifications en temps réel sur les activités du système liées à la sécurité.

Parmi les notifications que le kernel de CrowdStrike reçoit, il y a une notification pour la création de Named Pipes. Lorsque le driver reçoit une notification de Named Pipe, ces données sont combinées avec d'autres informations contextuelles sur le système. Ces données combinées sont présentées pour évaluation par rapport aux Template Instances transmises dans le Channel File 291.

Pour examiner de plus près ce processus, nous visualisons l'état du registre au moment de la lecture de la mémoire hors limites en rétablissant la trame et en désassemblant les instructions précédentes pour nous orienter. *(Remarque : cette liste de désassemblage a été modifiée par rapport à la sortie standard du débogueur afin d'annoter le code avec des noms de symboles illustratifs).*

```
1: kd> .trap 0xffffae035f57eca0
NOTE: The trap frame does not contain all registers.
Some register values may be zeroed or incorrect.
rax=ffffae035f57f280 rbx=0000000000000000 rcx=0000000000000003
rdx=ffffae035f57f250 rsi=0000000000000000 rdi=0000000000000000
rip=fffff8020ebc14ed rsp=ffffae035f57ee30 rbp=ffffae035f57ef30
 r8=ffffd6030000006a  r9=0000000000000000 r10=0000000000000000
r11=0000000000000014 r12=0000000000000000 r13=0000000000000000
r14=0000000000000000 r15=0000000000000000
iopl=0         nv up ei ng nz na po nc
csagent+0xe14ed:
fffff802`0ebc14ed 458b08          mov     r9d,dword ptr [r8]
ds:ffffd603`0000006a=????????

1: kd> u @rip-16 L0n10
csagent!TemplateGetString+0xe:
```

```
fffff802`0ebc14d7 4e8b04d8        mov     r8,qword ptr [rax+r11*8]
fffff802`0ebc14db 750b            jne     csagent!TemplateGetString+0x1f
(fffff802`0ebc14e8)
fffff802`0ebc14dd 4d85c0          test    r8,r8
fffff802`0ebc14e0 7412            je      csagent!TemplateGetString+0x2b
(fffff802`0ebc14f4)
fffff802`0ebc14e2 450fb708        movzx   r9d,word ptr [r8]
fffff802`0ebc14e6 eb08            jmp     csagent!TemplateGetString+0x27
(fffff802`0ebc14f0)
fffff802`0ebc14e8 4d85c0          test    r8,r8
fffff802`0ebc14eb 7407            je      csagent!TemplateGetString+0x2b
(fffff802`0ebc14f4)
fffff802`0ebc14ed 458b08          mov     r9d,dword ptr [r8]
fffff802`0ebc14f0 4d8b5008        mov     r10,qword ptr [r8+8]
```

Avant cet extrait de code, les données contextuelles de la notification de Named Pipe ont été préparées pour le Template Type IPC sous la forme d'un tableau de 20 pointeurs de données entrantes, chacun pointant vers une structure de chaîne contenant une adresse tampon et une valeur de taille. Cet extrait vise à sélectionner l'une des entrées de données pour renvoyer son adresse et sa taille tampon, selon un index spécifié par le Channel File 291.

Lorsque nous saisissons ce code, l'adresse du tableau de pointeurs à 20 entrées est conservée dans le registre rax, et le registre r11 indique que l'entrée à récupérer est à l'index 0x14, c'est-à-dire le 21ème élément.

En examinant le tableau, nous trouvons en effet un tableau de 20 pointeurs vers des structures de chaînes de données, suivi d'une 21ème valeur qui *ne* pointe pas vers une mémoire valide :

```
1: kd> dp @rax l0n21
ffffae03`5f57f280  ffffae03`5f57f320 ffffae03`5f57f330
ffffae03`5f57f290  ffffae03`5f57f340 ffffae03`5f57f350
ffffae03`5f57f2a0  ffffae03`5f57f360 ffffae03`5f57f370
ffffae03`5f57f2b0  ffffae03`5f57f380 ffffae03`5f57f390
ffffae03`5f57f2c0  ffffae03`5f57f3a0 ffffae03`5f57f3b0
ffffae03`5f57f2d0  ffffae03`5f57f3c0 ffffae03`5f57f3d0
ffffae03`5f57f2e0  ffffae03`5f57f3e0 ffffae03`5f57f3f0
ffffae03`5f57f2f0  ffffae03`5f57f400 ffffae03`5f57f410
ffffae03`5f57f300  ffffae03`5f57f420 ffffae03`5f57f430
ffffae03`5f57f310  ffffae03`5f57f440 ffffae03`5f57f450
ffffae03`5f57f320  ffffd603`0000006a
1: kd> !pte ffffd603`0000006a
                                           VA ffffd6030000006a
```

```
PXE at FFFFFE7F3F9FCD60    PPE at FFFFFE7F3F9AC060    PDE at FFFFFE7F3580C000
PTE at FFFFFE6B01800000
contains 0A00000107A00863  contains 0000000000000000
pfn 107a00    ---DA--KWEV  contains 0000000000000000
not valid
```

Après avoir lu ce pointeur non valide dans le registre r8, le flux de contrôle dans l'extrait de code ci-dessus effectue le premier saut pour aborder `fffff802'0ebc14e8`, effectue une vérification du pointeur NULL, puis tente une lecture à travers le pointeur invalide, ce qui entraîne une lecture hors limites et une vérification de bug ultérieure.

AUTRES RESSOURCES

Références et des liens vers des ressources techniques supplémentaires.
Hub de correction et de guidage : mise à jour du contenu Falcon pour les hôtes Windows
Blog : Détails techniques : Mise à jour du contenu Falcon pour les hôtes Windows
Centre de remédiation — Glossaire des termes

Ce document est une traduction de la version anglaise suivante https://www.crowdstrike.com/wp-content/uploads/2024/08/Channel-File-291-Incident-Root-Cause-Analysis-08.06.2024.pdf. Cette traduction est uniquement fournie afin d'en faciliter la compréhension. En cas de conflit ou d'ambiguïté, la version anglaise prévaudra toujours.