

Sécurité des systèmes d'information

CVE 2022-21907

Réalisé par :
KAMAL MAROUANE
NABILA RIMAQUI
ISMAIL ZARKTOUNI

Encadré par :
M. SÉBASTIEN VIARDOT

Table des matières

Introduction Générale	2
1 Présentation de la faille	3
1.1 Généralités	3
1.2 Score CVSS	3
1.3 Programme compromis	4
1.4 Type de compromission	4
1.5 Impact sur les Systèmes Clients et Serveurs Windows	5
1.6 Explication de la vulnérabilité et du mécanisme d'exploitation	5
1.6.1 Préparation à l'Analyse	5
1.6.2 Identification des Modifications Clés	6
1.6.3 Analyse de l'Interaction Fonctionnelle	7
1.6.4 Mécanisme de Déclenchement du Crash du Système	8
1.6.5 Exécution de code à distance	9
1.6.6 Conclusion de l'exploitation	9
1.7 Architecture de l'exploitation de la faille	9
1.7.1 Exploitation - Déni de service	9
1.7.2 Exploitation - Execution de code a distance	11
2 Exploitation de la vulnérabilité	13
2.1 Configuration de l'Environnement d'Exploitation	13
2.1.1 Configuration de la machine virtuelle Windows 10 (Victime)	13
2.1.2 Configuration de la machine virtuelle Kali Linux (Attaquante)	20
2.2 Développement du Code d'Attaque	22
2.3 Preuve de concept	22
3 Préconisations de Sécurité	25
3.1 Préconisations pour Limiter l'Impact	25
3.2 Préconisations pour Empêcher l'Exploitation	25
3.3 Bonnes Pratiques de Sécurité	26
3.4 Amélioration des Pratiques d'Initialisation en Développement pour Prévenir les Failles	26
3.5 Cible de sécurité	27
3.5.1 Objectif	27
3.5.2 Criticité de la faille	27
3.5.3 Actions préventives et curatives	27
Bibliography	29

Table des figures

1.1	CVSS Score	3
1.2	Les fonctions principalement modifiées	6
1.3	Comparaison des deux versions vulnérable et corrigée de la fonction http !Ul- pAllocateFastTracker() (vulnérable à gauche)	6
1.4	Deux appels à http !UlPAllocateFastTracker() depuis http !UlFastSendHtt- pResponse()	7
1.5	Strcuture MDL	7
1.6	Fonction http !UlFastSendHttpReponse() décompilée	8
1.7	Nous controlons la valeur de MDL->MappedSystemVa	8
1.8	Architecture d'exploitation causant un BDOS	10
1.9	Architecture d'exploitation pour l'exécution de code à distance	11
2.1	Ouverture de l'outil Services dans Windows via la commande Exécuter	13
2.2	Liste des services Windows avec 'Windows Update' sélectionné	14
2.3	Menu contextuel de 'Windows Update' avec l'option 'Propriétés' sélectionnée	14
2.4	Propriétés du service 'Windows Update' avec le type de démarrage affiché .	15
2.5	Modification du type de démarrage du service 'Windows Update' pour le désactiver	15
2.6	Fenêtre de l'outil Exécuter pour ouvrir l'Éditeur du Registre avec la com- mande 'regedit'	16
2.7	Navigation dans l'Éditeur du Registre pour localiser la clé de Windows Update	16
2.8	Création d'une nouvelle valeur DWORD dans l'Éditeur du Registre pour la configuration de Windows Update	17
2.9	Nouvelle valeur DWORD 'NoAutoUpdate' dans l'Éditeur du Registre pour désactiver les mises à jour automatiques	17
2.10	Modification de la valeur 'NoAutoUpdate' pour désactiver les mises à jour automatiques	17
2.11	Modification de la valeur du Registre pour désactiver les mises à jour au- tomatiques	18
2.12	Recherche du Panneau de configuration dans Windows pour la désinstalla- tion de programmes	18
2.13	Activation de fonctionnalités Windows depuis le Panneau de configura- tion	19
2.14	Activation des services Internet Information Services dans Windows	19
2.15	Vérification de la version de Python sur Kali Linux	21
2.16	Environnements de bureau Windows et Kali Linux côté à côté	22
2.17	Configuration IP d'une machine Windows affichée dans l'invite de commandes	23
2.18	Changement de répertoire vers le dossier d'exploitation CVE-2022-21907 sur Kali Linux	24
2.19	Écran bleu de la mort sur une machine Windows indiquant un crash système	24

Glossaire

CVE (Common Vulnerabilities and Exposures) : Un référentiel international qui fournit des informations claires et concises sur les vulnérabilités de sécurité publiquement connues.

RCE (Remote Code Execution) : Un type de vulnérabilité qui permet à un attaquant d'exécuter à distance du code arbitraire sur un système compromis.

DoS (Denial of Service) : Une attaque informatique visant à rendre une ressource informatique indisponible à ses utilisateurs prévus.

CVSS (Common Vulnerability Scoring System) : Un cadre standardisé pour évaluer la gravité des vulnérabilités de sécurité informatique.

http.sys : Un pilote de mode noyau dans Windows qui reçoit et traite les requêtes HTTP.

IIS (Internet Information Services) : Un ensemble de services Internet flexibles et gérables pour le système d'exploitation Windows Server.

Buffer Overflow : Un défaut de programmation où un programme écrit des données au-delà des limites d'un tampon alloué, ce qui peut conduire à des vulnérabilités de sécurité.

MDL (Memory Descriptor List) : Une structure utilisée dans les systèmes Windows pour décrire les tampons de mémoire dans les opérations d'entrée/sortie.

BinDiff : Un outil d'analyse binaire qui permet de comparer différentes versions d'un fichier binaire pour détecter les changements ou les similitudes.

Ghidra : Un outil d'analyse de logiciels qui aide les chercheurs en sécurité à comprendre le code binaire.

memset() : Une fonction en C utilisée pour remplir un bloc de mémoire avec une valeur particulière, souvent utilisée pour initialiser la mémoire.

BSOD (Blue Screen of Death) : Un écran d'erreur affiché sur un ordinateur Windows après un crash du système.

PSSI (Politique de Sécurité des Systèmes d'Information) : Un ensemble de directives visant à protéger les informations et les systèmes informatiques d'une organisation contre les cybermenaces.

Introduction Générale

Dans ce rapport, nous abordons une vulnérabilité critique qui peut impacter sévèrement le fonctionnement normal des systèmes informatiques et compromettre la confidentialité et l'intégrité des données. Cette étude se concentre sur la CVE-2022-21907, une faille majeure dans la pile de protocole HTTP de Microsoft Windows, découverte et rendue publique en début d'année 2022.

Cette vulnérabilité, qui affecte le pilote noyau http.sys et par extension les Services d'Informations Internet (IIS) de Windows, permet à un attaquant d'exécuter du code à distance et de provoquer des conditions de déni de service. Qualifiée de " wormable " par Microsoft, cette faille souligne la potentialité d'une attaque se propageant de manière autonome d'un système à l'autre au sein d'un réseau.

Notre analyse se propose de dévoiler les intrications techniques de la vulnérabilité CVE-2022-21907, en s'attachant à en décrire la nature, les vecteurs d'exploitation possibles ainsi que les mesures correctives appropriées. Le document sera articulé autour de trois chapitres principaux : le premier présentera une vue d'ensemble de la vulnérabilité, exposant en détail son fonctionnement et son architecture sous-jacente ; le second se focalisera sur une mise en application pratique de la vulnérabilité, illustrant la manière dont elle peut être exploitée à travers un exemple de simulation d'attaque ; et le troisième détaillera les stratégies de mitigation à privilégier pour en atténuer les risques. Cette exploration vise à enrichir notre compréhension de la CVE-2022-21907, en mettant en exergue son rôle critique dans l'écosystème de la cybersécurité et en élaborant sur les pratiques optimales pour se défendre face à de telles menaces.

Chapitre 1

Présentation de la faille

1.1 Généralités

Les CVE (Common Vulnerabilities and Exposures) sont des identifiants attribués aux failles de sécurité informatique, permettant une coordination efficace pour les résoudre. Supervisé par MITRE et soutenu par la CISA, le programme CVE est crucial pour la gestion des vulnérabilités en offrant une référence claire aux professionnels de la cybersécurité. Les attaques d'exécution de code à distance (RCE) sont particulièrement préoccupantes, permettant aux cyberattaquants d'infecter à distance un système, compromettant les données ou utilisant le réseau à des fins malveillantes. Les attaques de déni de service (DoS), souvent associées aux vulnérabilités RCE, visent à rendre une ressource informatique indisponible, aggravant les dommages des cyberattaques.

L'exemple de CVE-2022-21907, publié le 11 janvier 2022, met en lumière une vulnérabilité RCE dans la pile de protocole HTTP de Windows. Elle permet à un attaquant d'exécuter du code arbitraire ou de réaliser une attaque DoS, soulignant l'importance de la vigilance, de la réactivité, de la détection rapide et de la mise en œuvre des correctifs dans le domaine de la cybersécurité.

1.2 Score CVSS

CVSS scores for CVE-2022-21907						
Base Score	Base Severity	CVSS Vector	Exploitability Score	Impact Score	Source	
10.0	HIGH	AV:N/AC:L/Au:N/C:C/I:C/A:C	10.0	10.0	nvd@nist.gov	
9.8	CRITICAL	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	3.9	5.9	secure@microsoft.com	

FIGURE 1.1 – CVSS Score

La méthodologie CVSS (Common Vulnerability Scoring System) est un outil crucial pour évaluer l'impact des vulnérabilités de sécurité informatique, attribuant un score de 0 à 10 en fonction de la sévérité. Développé par le FIRST, le score CVSS comprend une composante de base mesurant l'impact théorique, un score temporel variant avec les exploits ou correctifs, et un score environnemental tenant compte du contexte utilisateur. Dans le cas de CVE-2022-21907, le score CVSS atteint 9.8, signalant une vulnérabilité extrêmement grave. Ceci est dû au fait que la faille ne requiert pas d'authentification, facilitant l'attaque, et son exploitation peut entraîner une exécution de code à distance (RCE) ainsi qu'une possibilité de déni de service (DoS). Ce score souligne l'urgence d'appliquer des correctifs et de mettre en place des stratégies de mitigation pour protéger les systèmes vulnérables.

1.3 Programme compromis

La vulnérabilité CVE-2022-21907 a un impact considérable sur une gamme étendue de systèmes d'exploitation Windows, soulignant la gravité et l'ampleur de la menace qu'elle représente. Les systèmes affectés comprennent plusieurs versions de Windows 10, notamment les versions 1809, 21H1, 20H2 et 21H2, dans leurs déclinaisons 32 bits, x64 et ARM64. De plus, cette vulnérabilité touche également Windows 11, à la fois pour les systèmes x64 et ARM64, ainsi que plusieurs versions de Microsoft Windows Server, incluant Windows Server 2019, Windows Server 2022, et la version Core Installation de Windows Server 20H2.

L'étendue des systèmes affectés met en évidence la vulnérabilité généralisée des infrastructures informatiques face à des failles de sécurité critiques. Les systèmes d'exploitation touchés par CVE-2022-21907 sont largement utilisés dans des environnements professionnels et personnels, ce qui rend cette vulnérabilité particulièrement préoccupante. Sa capacité à affecter des systèmes aussi variés souligne l'importance d'une mise à jour et d'une maintenance régulières des systèmes pour assurer la sécurité et la protection des données.

1.4 Type de compromission

L'impact de CVE-2022-21907 sur les systèmes informatiques est à la fois profond et diversifié, révélant la gravité de cette vulnérabilité d'exécution de code à distance (RCE) au sein de la pile de protocole HTTP de Windows. Cette vulnérabilité, de nature vermifuge, permet aux attaquants d'exécuter une série d'actions malveillantes à travers des réseaux, potentiellement se propageant à d'autres systèmes non sécurisés. Parmi les actions malveillantes possibles, on retrouve :

- Une attaque unique peut entraîner un crash ou un redémarrage du système Windows. Toutefois, des attaques répétées ou persistantes peuvent causer un crash complet, menant à un DoS de la machine virtuelle cible. .
- Les attaquants peuvent exploiter la faille pour installer des logiciels de scraping de données ou exécuter directement des commandes afin d'extraire et d'envoyer des informations sensibles. La capacité de cette vulnérabilité à se propager comme un ver augmente le risque de divulgation d'informations sur un large éventail de systèmes

connectés.

L'ampleur de l'impact de cette vulnérabilité dépend grandement de la machine cible, des services qu'elle héberge, et des données qu'elle traite. La facilité d'exécution de l'attaque, sans nécessiter d'interaction utilisateur, combinée à la difficulté de détecter une attaque active, rend cette vulnérabilité particulièrement attrayante pour les acteurs malveillants. Cette faille leur permet non seulement d'implanter du code arbitraire, mais aussi de créer une menace persistante qui peut se répandre au sein de réseaux d'applications.

1.5 Impact sur les Systèmes Clients et Serveurs Windows

La vulnérabilité CVE-2022-21907 impacte à la fois les machines clientes et serveurs, ciblant la pile de protocole HTTP (`http.sys`) de Microsoft Windows. Cette composante est fondamentale pour des services tels qu'Internet Information Services (IIS), et sa présence étendue la rend pertinente pour une variété de systèmes. Toute machine, qu'elle opère en tant que serveur ou client, est susceptible d'être vulnérable si le service `http.sys` y est actif. En conséquence, CVE-2022-21907 présente un risque potentiel d'exploitation sur un large éventail de systèmes Windows où `http.sys` joue un rôle crucial dans le traitement des requêtes HTTP.

1.6 Explication de la vulnérabilité et du mécanisme d'exploitation

L'analyse de la faille CVE-2022-21907 dans le composant `http.sys` de Windows a révélé un défi considérable. Face à la nature fermée du système Windows et l'absence d'accès au code source, notre analyse s'est appuyée sur l'investigation du fichier binaire `http.sys`. Pour cela, nous avons utilisé des outils d'analyse tels que BinDiff et Ghidra, nous lançant ainsi dans une exploration technique poussée.

La première étape a été de comprendre la nature de la faille. En consultant le site officiel de Windows, nous avons déduit qu'elle était liée au traitement des requêtes HTTP avec Trailers en mode "chunked". Les avis divergeaient entre un buffer overflow et un défaut de gestion de la mémoire. Face à cette incertitude, nous avons décidé de mener notre propre étude en nous inspirant des travaux d'experts.

1.6.1 Préparation à l'Analyse

La seconde étape a consisté à récupérer les versions vulnérable (10.0.19041.1387) et corrigée (10.0.19041.1466) du fichier `http.sys`. En sélectionnant des versions proches, nous avons assuré que notre analyse se concentrerait sur les changements directement liés à la vulnérabilité.

Avec BinDiff et Ghidra à notre disposition, nous avons plongé dans le dédale des binaires. BinDiff nous a permis de comparer les deux versions du fichier `http.sys`, mettant en lumière les différences subtiles mais essentielles. Ghidra, grâce à ses capacités de rétro-ingénierie, a offert une vue détaillée sur la structure interne et le fonctionnement du fichier, le rendant plus compréhensible.

1.6.2 Identification des Modifications Clés

Notre recherche s'est concentrée sur Windows 10, où deux fonctions principales ont montré des modifications significatives : http!UlpAllocateFastTracker() et http!UiFastSendHttpResponse().

Similarity	Confid	Change	EA Primary	Name Primary
0.90	0.99	GI-JEL-	00000001C00C0180	UiFastSendHttpResponse
0.97	0.98	-I--E--	00000001C00D4CDC	UlpAllocateFastTracker

FIGURE 1.2 – Les fonctions principalement modifiées

Voici une brève description de ces deux fonctions pour mieux comprendre le contexte :

- **http!UlpAllocateFastTracker()** est responsable de l'allocation et de la gestion d'un bloc de mémoire spécifique, nommé FastTracker. Le FastTracker est utilisé pour suivre et gérer l'état interne et les données importantes liées au traitement des requêtes HTTP.
- **http!UiFastSendHttpResponse()** est chargée d'envoyer les réponses HTTP aux clients. Elle intervient après le traitement d'une requête HTTP, gérant la préparation et l'envoi de la réponse appropriée.

Nous avons commencé l'analyse avec http!UlpAllocateFastTracker(). En comparant les versions vulnérables et corrigées, nous avons découvert l'ajout d'un double appel à memset comme le montre l'image 1.3. La fonction memset() est utilisé pour initialiser différents segments de la mémoire.

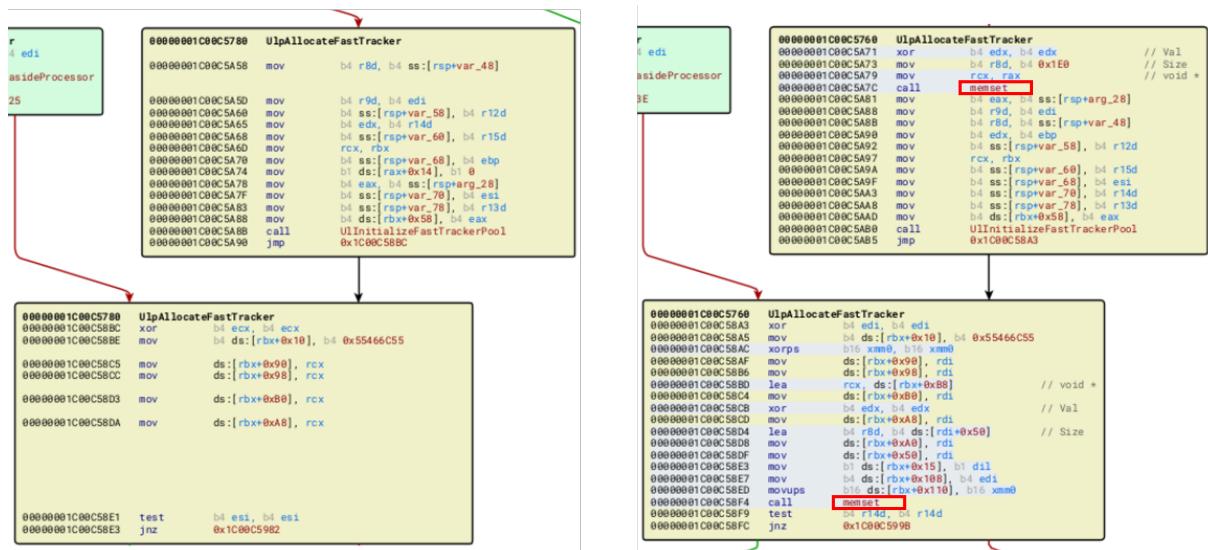


FIGURE 1.3 – Comparaison des deux versions vulnérable et corrigée de la fonction http!UlpAllocateFastTracker() (vulnérable à gauche)

Le premier appel initialisait 0x1e0 octets, et le second à partir de 0x2e pour 0x50 octets. L'appel double à memset dans le même tampon, en particulier pour initialiser un segment spécifique du Tracker (de 0x2E à 0x7E), était inattendu. Cela indiquait que le développeur avait identifié la nécessité de sécuriser spécifiquement cette partie du Tracker. Se pourrait-il que la vulnérabilité soit causée par ce manque d'initialisation ? (spoiler : la réponse est oui).

1.6.3 Analyse de l'Interaction Fonctionnelle

Poursuivant notre analyse, nous nous sommes intéressés à l'endroit où la fonction `http!UlpAllocateFastTracker()` était appelée. Il s'est avéré qu'elle était utilisée dans `http!UIFastSendHttpRspone()` (figure 1.4). Nous avons donc décidé d'examiner de plus près l'interaction entre ces deux fonctions, supposant que la faille se trouvait à proximité.

xrefs to UlpAllocateFastTracker			
Direction	Type	Address	Text
Up	o	.rdata:00000001C007AC78	RUNTIME_FUNCTION <rva Ulp
Up	o	.pdata:00000001C009535C	RUNTIME_FUNCTION <rva Ulp
Up	p	UIFastSendHttpRspone+2F0	call UlpAllocateFastTracker
Up	p	UIFastSendHttpRspone+E99	call UlpAllocateFastTracker

FIGURE 1.4 – Deux appels à `http!UlpAllocateFastTracker()` depuis `http!UIFastSendHttpRspone()`

En plaçant un point d'arrêt sur `http!UIFastSendHttpRspone`, nous avons observé son comportement lors de l'envoi de plusieurs requêtes Trailers. Nous avons constaté que `UlpAllocateFastTracker` était effectivement appelée, mais de manière discontinue. Ce comportement intermittent suggérait une relation complexe entre le type de requête reçue et l'activation de la vulnérabilité, bien que la raison exacte de cette intermittence nous échappait.

Cependant, sachant que certaines requêtes déclenchaient l'appel à `http!UlpAllocateFastTracker()`, nous avons porté notre attention sur le bloc de mémoire retourné par cette fonction, le FastTracker. Ce dernier, en tant que composant crucial contenant des pointeurs vers des structures MDL, s'est avéré être un élément clé pour comprendre la vulnérabilité. Les MDL (Memory Descriptor Lists) (dont la structure est visible sur la figure 1.5) sont fondamentales dans Windows pour le mappage entre adresses de mémoire virtuelle et physique. Une altération des données des MDL pourrait provoquer une instabilité du système et un crash, constituant ainsi une piste prometteuse pour notre analyse.

```
void MmUnmapLockedPages(
    [in] PVOID BaseAddress,
    [in] PMDL MemoryDescriptorList
);
```

FIGURE 1.5 – Structure MDL

Dans `http!UIFastSendHttpRspone`, nous avons également remarqué la fonction `MmUnmapLockedPages` (encadrée en rouge sur la figure 1.6), une routine du noyau Windows qui libère un mappage entre une adresse de mémoire virtuelle et une adresse de mémoire physique.

En conditions normales, ces opérations sont inoffensives, mais dans le contexte de la vulnérabilité, elles deviennent un vecteur d'attaque potentiel.

```

if ( FastTracker )
{
    if ( *(_QWORD *)(FastTracker + 0x50) )
    {
        md1 = *(PMDL *)(FastTracker + 0x80);
        if ( md1->MdlFlags & 1 )
            MmUnmapLockedPages(md1->MappedSystemVa, *(PMDL *)(FastTracker + 0x80));
    }
    UlpFreeFastTracker((PVOID)FastTracker);
    if ( v14R )

```

FIGURE 1.6 – Fonction http!UlFastSendHttpResponse() décompilée

Maintenant que nous avons défini l'ensemble des acteurs jouant un rôle dans la vulnérabilité, nous pouvons passer au mécanisme de l'exploitation.

1.6.4 Mécanisme de Déclenchement du Crash du Système

Le cœur de la vulnérabilité réside dans la manière dont les données de la requête HTTP influencent le FastTracker et, par conséquent, les structures MDL. Dans le scénario d'exploitation, nous envoyons d'abord une série de requêtes GET volumineuses pour remplir les tampons de traitement des données du serveur. Cette étape prépare le terrain pour l'exploitation en étendant les limites de la mémoire allouée et en créant un environnement favorable à un débordement de tampon.

Ensuite, une requête malformée, dépourvue de "HTTP/1.1", est envoyée. Cette requête, intentionnellement incorrecte, crée une condition anormale qui, couplée à l'état préparé du serveur, déclenche un buffer overflow. Ce débordement de tampon entraîne la corruption de la mémoire, avec des données de requête HTTP se retrouvant dans des emplacements mémoire inattendus, en particulier dans MDL->MappedSystemVa comme le montre la figure 1.7. En effet, la valeur hexadécimale "0x41414141.." représente "AAAA.." que nous avons envoyé dans nos requêtes.

```

1: kd> r
rax=0000000c000000d rbx=00000000000000c8 rcx=4141414141414141
rdx=fffffe089fb2f0bb0 rsi=0000000000000000 rdi=fffffe089fb2f08a0
rip=fffff8062cafe9ed rsp=fffffa30c7680ef80 rbp=fffffe089f4358100
r8=fffffe089fb2f0c28 r9=00000000c000000d r10=fffff8062f424b70
r11=0000000000000000 r12=00000000ffffffffff r13=000001b2a6c052c0
r14=fffffe089fb8a010 r15=00000000c000000d
iopl=0          nv up ei pl nz na pe nc
cs=0010  ss=0018  ds=002b  es=002b  fs=0053  gs=002b          efl=00040202
HTTP!UlFastSendHttpResponse+0x2e86d:
fffff806`2cafe9ed e87e619202      call     nt!MmUnmapLockedPages (fffff806`2f424b70)

```

FIGURE 1.7 – Nous contrôlons la valeur de MDL->MappedSystemVa

Lorsque MmUnmapLockedPages est appelée dans un contexte de données mémoire corrompues, elle tente de traiter des informations altérées, conduisant à un comportement anormal du système. Le résultat est souvent un crash, manifesté par un écran bleu de la

mort (BSOD) suivit du redémarrage du système. De plus, en répétant cette attaque, un attaquant peut facilement provoquer un déni de service (DoS).

1.6.5 Exécution de code à distance

L'exploitation de CVE-2022-21907 pour une exécution de code à distance implique une série d'étapes techniques complexes. Initialement, l'attaquant prépare le système en envoyant des requêtes GET volumineuses pour saturer les buffers de mémoire, créant ainsi un environnement propice à un débordement de tampon. Ensuite, une requête HTTP malformée est utilisée pour injecter un shellcode malveillant dans la mémoire du serveur. Lorsque le débordement de tampon se produit, le shellcode doit être positionné stratégiquement dans une zone de mémoire exécutable. L'attaquant doit ensuite manipuler les adresses de retour ou les pointeurs de fonction pour rediriger le flux d'exécution du programme vers ce shellcode. Cette manœuvre requiert une connaissance approfondie de l'architecture du système cible et une précision dans la manipulation des données et de la mémoire.

1.6.6 Conclusion de l'exploitation

En conclusion, l'analyse de la vulnérabilité CVE-2022-21907 révèle un scénario d'exploitation où un bombardement ciblé de requêtes GET volumineuses, suivi d'une requête HTTP malformée, conduit à un débordement de tampon. Ce débordement de tampon entraîne à son tour une corruption de la mémoire, provoquant un crash du système. Au cœur de cette vulnérabilité se trouve un problème d'initialisation de la mémoire, où certaines parties de la structure mémoire ne sont pas correctement initialisées.

La résolution de cette faille, en grande partie, a été obtenue grâce à l'introduction stratégique d'appels à `memset()` dans la fonction `http!UlpAllocateFastTracker()`. Ces appels garantissent que les segments critiques du FastTracker sont correctement initialisés à zéro, éliminant ainsi les données résiduelles ou non sécurisées qui pourraient être exploitées.

1.7 Architecture de l'exploitation de la faille

La compréhension détaillée des mécanismes sous-jacents à la faille CVE-2022-21907 est cruciale pour appréhender les risques associés et les stratégies de mitigation nécessaires. Dans cette section, nous explorons deux architectures distinctes d'exploitation, chacune représentant un vecteur d'attaque différent et les conséquences pour les systèmes affectés.

1.7.1 Exploitation - Déni de service

Cette première architecture illustre le scénario conduisant à un écran bleu de la mort (BSOD), un symptôme classique d'une erreur fatale au niveau du système d'exploitation. L'exploitation répétée de cette vulnérabilité peut entraîner un déni de service (DoS), où le système cible devient inopérant, nécessitant une intervention pour la récupération.

- 1 - Identification de la Cible : Les attaquants peuvent trouver l'adresse IP d'une cible par des moyens tels que la résolution DNS, l'analyse de réseau, l'ingénierie sociale, ou le balayage de plages d'adresses IP. Ils peuvent également exploiter des registres de connexion, des services en ligne, ou des techniques de phishing et de

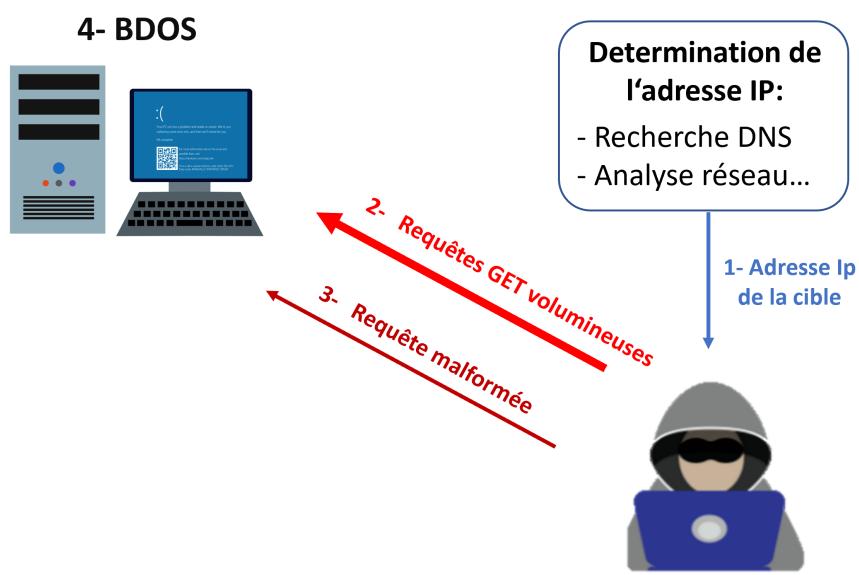


FIGURE 1.8 – Architecture d’exploitation causant un BDOS

malware pour obtenir ces informations.

- 2 - Envoi de Requêtes GET Volumineuses : L’attaquant commence par saturer le serveur cible avec une série de requêtes GET volumineuses. Cette approche est stratégique : elle vise à mettre sous pression les mécanismes de traitement des requêtes du serveur, en particulier le traitement des en-têtes HTTP et le mécanisme de transfert ‘chunked’ des données. L’objectif est de préparer le serveur à un état où il devient vulnérable à des manipulations supplémentaires, rendant ainsi possible un débordement de tampon.
- 3 - Envoi d’une Requête Malformée : Une fois que le serveur est préparé par la première vague d’attaques, une requête HTTP spécifiquement malformée est envoyée. Cette requête ne respecte pas le format standard attendu par le serveur (par exemple, en omettant le “HTTP/1.1” à la fin). La requête est intentionnellement conçue pour exploiter la vulnérabilité au sein du système en déclenchant un comportement erroné du serveur lors de la gestion des données de la requête.
- 4 - Exploitation et Conséquences : Suite à la réception de la requête malformée, le serveur tente de traiter les données, mais en raison de la vulnérabilité CVE-2022-21907, cette tentative échoue et provoque un débordement de tampon. Ce débordement entraîne une corruption de la mémoire et, en fin de compte, un écran bleu de la mort (BSOD). Si cette attaque est effectuée de manière répétitive, elle peut entraîner un déni de service (DoS), rendant le serveur inopérant et nécessitant un redémarrage ou une intervention technique.

1.7.2 Exploitation - Execution de code à distance

Cette seconde architecture met en lumière la possibilité d'exécuter du code à distance (RCE), exploitant la nature vermifuge de la faille. Ce scénario démontre comment un attaquant peut non seulement provoquer un crash du système mais également prendre le contrôle de la machine affectée, représentant ainsi une menace significative pour l'intégrité et la confidentialité des données de l'utilisateur.

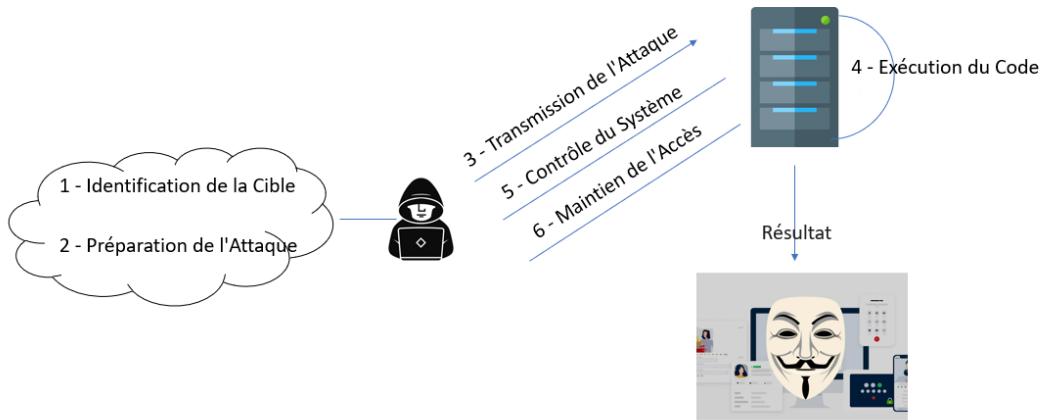


FIGURE 1.9 – Architecture d'exploitation pour l'exécution de code à distance

- 1 - Identification de la Cible : L'attaquant identifie un système vulnérable en se basant sur la version de Windows et la configuration de la pile HTTP.
- 2 - Préparation de l'Attaque : L'attaquant prépare un paquet ou une requête HTTP spécialement conçue pour exploiter la faille dans http.sys.
- 3 - Transmission de l'Attaque : L'attaquant envoie la requête malveillante au système cible via le réseau, souvent en se masquant pour éviter la détection.
- 4 - Exécution du Code : Lorsque la requête atteint le système cible, la faille dans http.sys est exploitée, permettant l'exécution du code malveillant.
- 5 - Contrôle du Système : Suite à l'exécution réussie, l'attaquant peut prendre le contrôle du système, accéder à des données sensibles, ou exécuter d'autres activités malveillantes.
- 6 - Maintien de l'Accès : L'attaquant peut installer des outils pour maintenir l'accès au système, facilitant des actions futures.

Conséquences de l'Exploitation

Les conséquences d'une telle exploitation sont graves : elles vont de la violation de données sensibles à la perturbation des opérations commerciales, et peuvent même inclure des répercussions réglementaires et juridiques pour les organisations touchées.

Chapitre 2

Exploitation de la vulnérabilité

Dans ce chapitre, nous détaillerons les étapes clés et les choix stratégiques faits pour configurer un environnement d'exploitation réaliste centré autour de la vulnérabilité CVE-2022-21907.

2.1 Configuration de l'Environnement d'Exploitation

Pour commencer, nous établissons un réseau NAT personnalisé, crucial pour assurer que les machines attaquante et victime soient dans le même réseau, conformément à l'exploitation de la faille CVE-2022-21907. La commande utilisée est :

```
1 VBoxManage natnetwork add --netname MyCustomisedNet --network
2 "192.168.100.0/24" --enable --dhcp on
```

2.1.1 Configuration de la machine virtuelle Windows 10 (Victime)

La machine virtuelle Windows 10 servira de cible dans notre scénario. Nous détaillerons sa configuration en soulignant les aspects la rendant vulnérable à l'attaque ciblée. Ces aspects incluent la version du système d'exploitation (Windows 10 Version 20H2 pour systèmes basés sur ARM64) et toute modification pertinente. Nous désactivons les mises à jour automatiques pour maintenir la version vulnérable, illustré par des captures d'écran à insérer ici. Les étapes pour désactiver les mises à jour automatiques de Windows via **services.msc** et l'Éditeur du Registre **regedit**.

1 - Ouvrez la boîte de dialogue Exécuter en appuyant sur Win + R et tapez services.msc pour ouvrir l'outil Services.

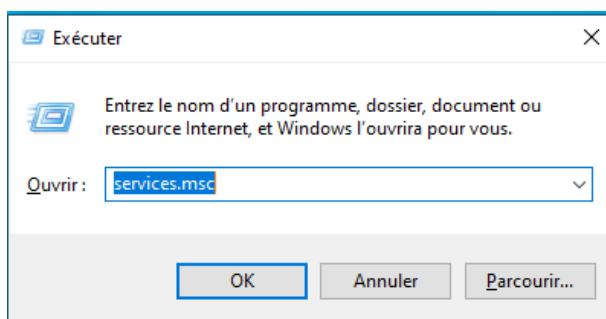


FIGURE 2.1 – Ouverture de l'outil Services dans Windows via la commande Exécuter

2 - Dans la liste des services, localisez Windows Update et ouvrez ses propriétés.

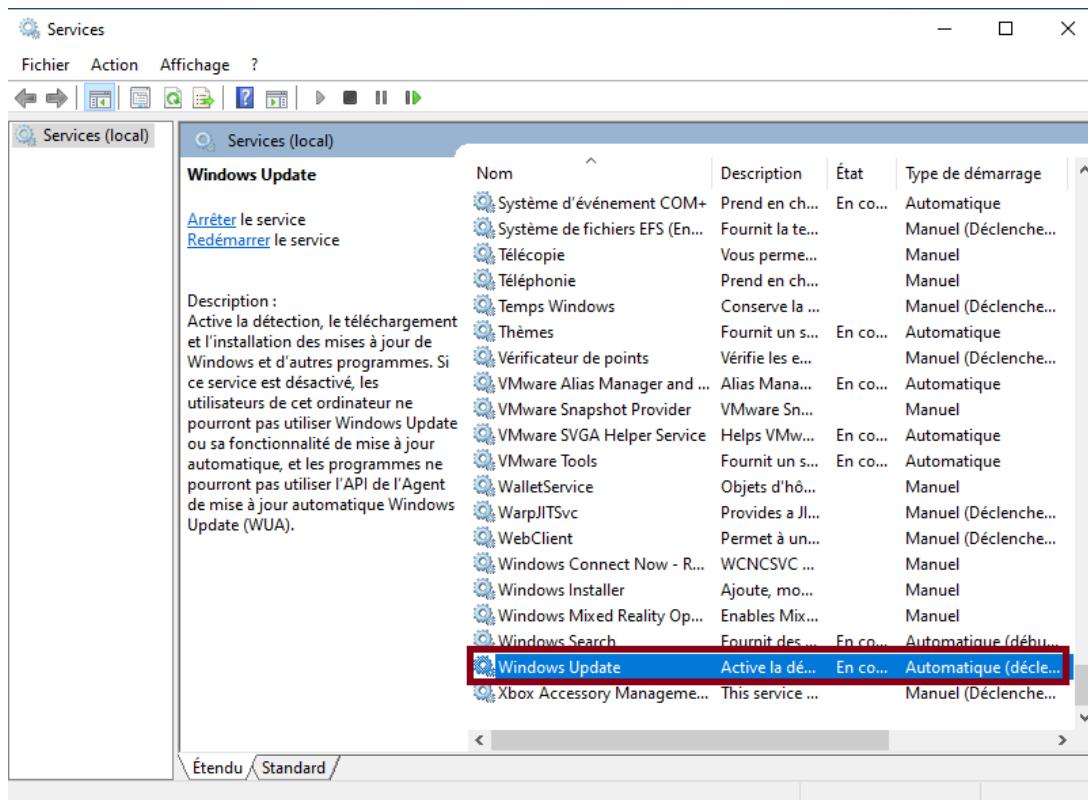


FIGURE 2.2 – Liste des services Windows avec 'Windows Update' sélectionné

3 - Changez le type de démarrage à Désactivé et arrêtez le service si nécessaire.

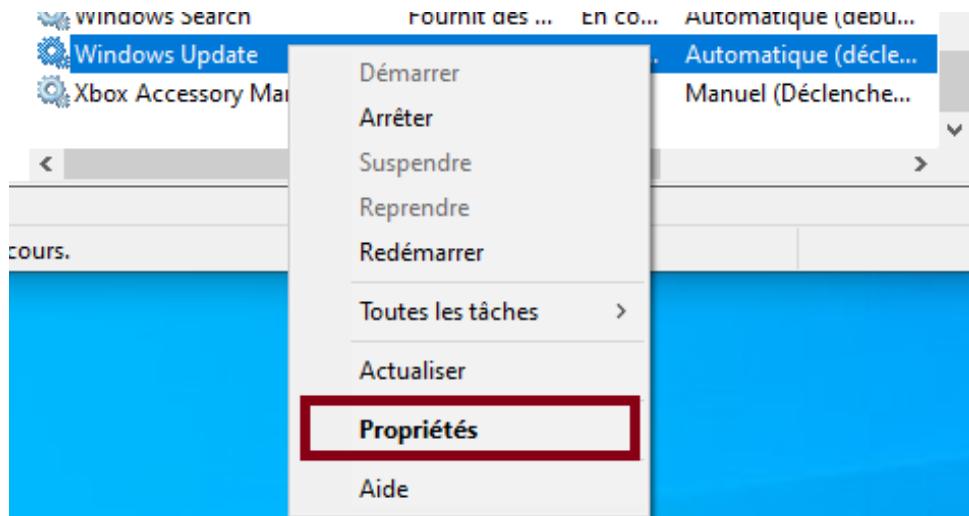


FIGURE 2.3 – Menu contextuel de 'Windows Update' avec l'option 'Propriétés' sélectionnée

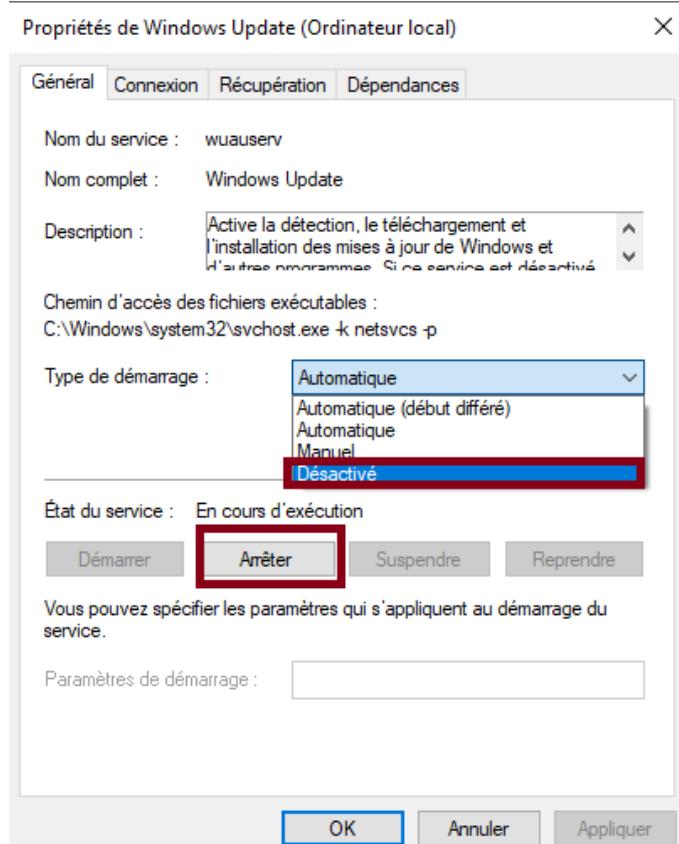


FIGURE 2.4 – Propriétés du service 'Windows Update' avec le type de démarrage affiché

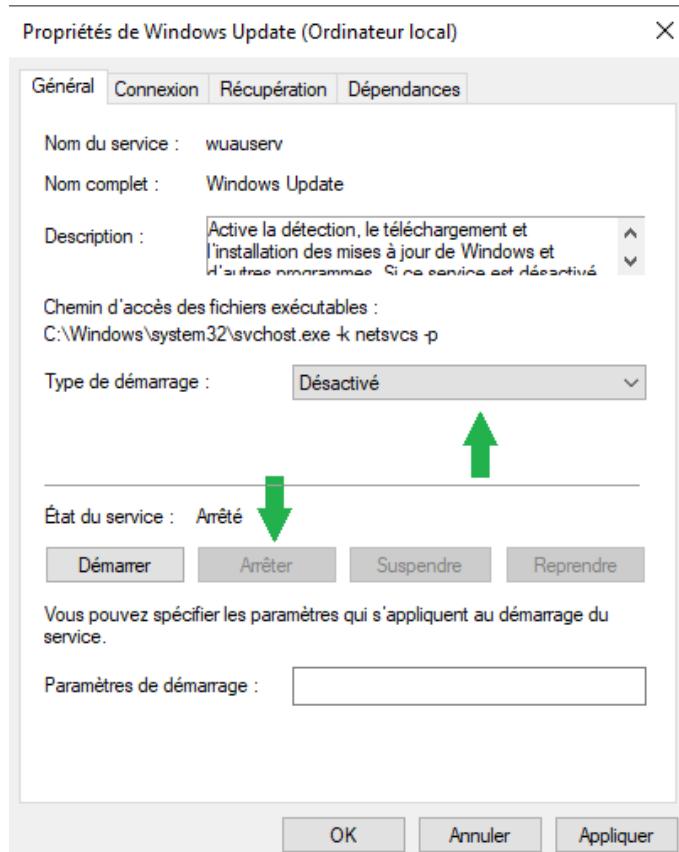


FIGURE 2.5 – Modification du type de démarrage du service 'Windows Update' pour le désactiver

4 - Ouvrez à nouveau la boîte de dialogue Exécuter et tapez regedit pour ouvrir l'Éditeur du Registre.

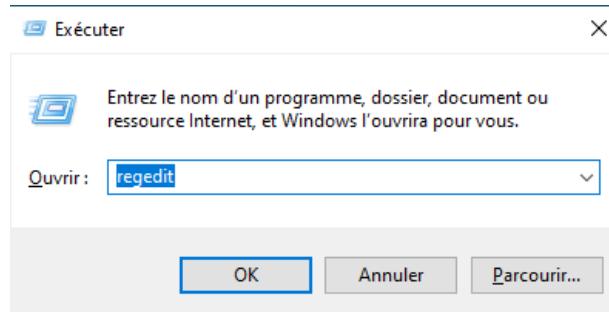


FIGURE 2.6 – Fenêtre de l'outil Exécuter pour ouvrir l'Éditeur du Registre avec la commande 'regedit'

5 - Naviguez jusqu'à la clé de registre Utilisez : "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU" pour accéder à la clé de registre (Assurez-vous de créer d'abord les répertoires WindowsUpdate et AU).

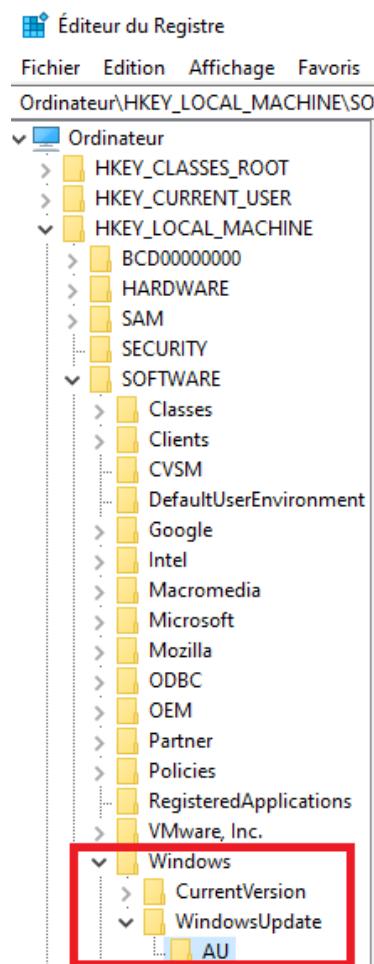


FIGURE 2.7 – Navigation dans l'Éditeur du Registre pour localiser la clé de Windows Update

6 - Créez une nouvelle valeur DWORD (32 bits) appelée NoAutoUpdate et définissez sa valeur sur 1 pour désactiver les mises à jour automatiques.

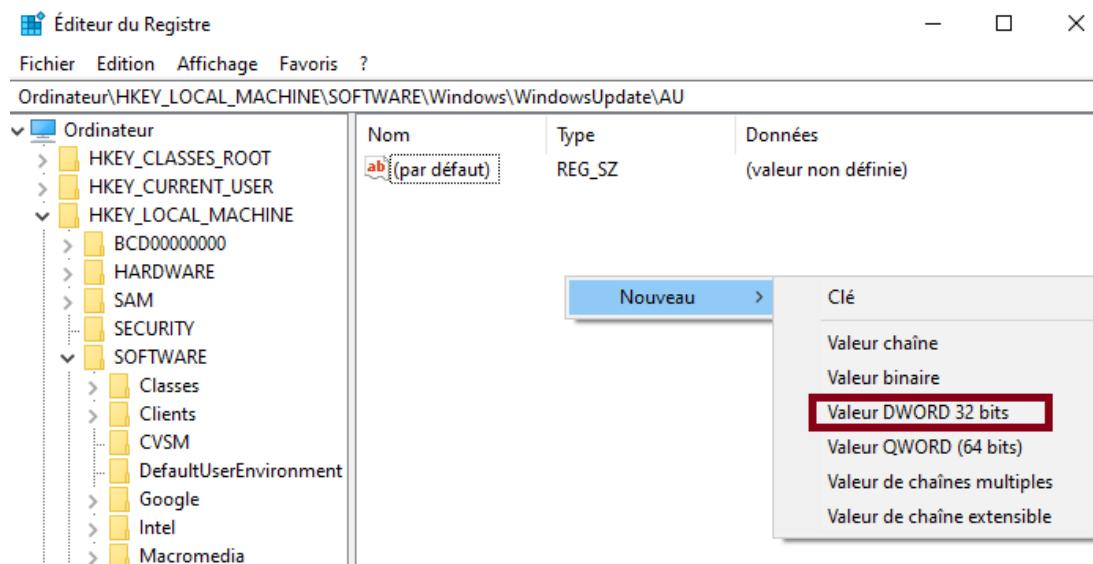


FIGURE 2.8 – Création d’une nouvelle valeur DWORD dans l’Éditeur du Registre pour la configuration de Windows Update

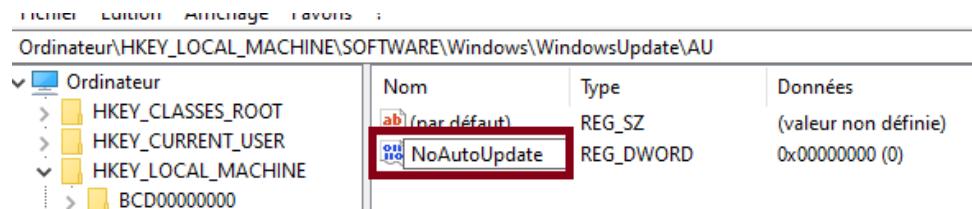


FIGURE 2.9 – Nouvelle valeur DWORD 'NoAutoUpdate' dans l’Éditeur du Registre pour désactiver les mises à jour automatiques

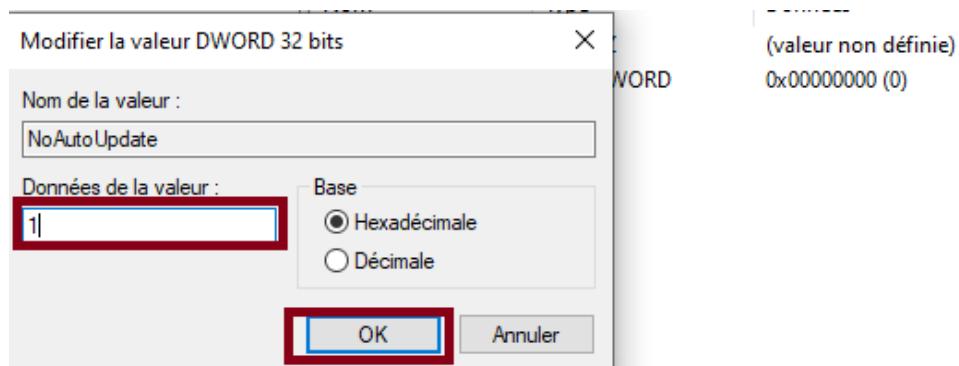


FIGURE 2.10 – Modification de la valeur 'NoAutoUpdate' pour désactiver les mises à jour automatiques

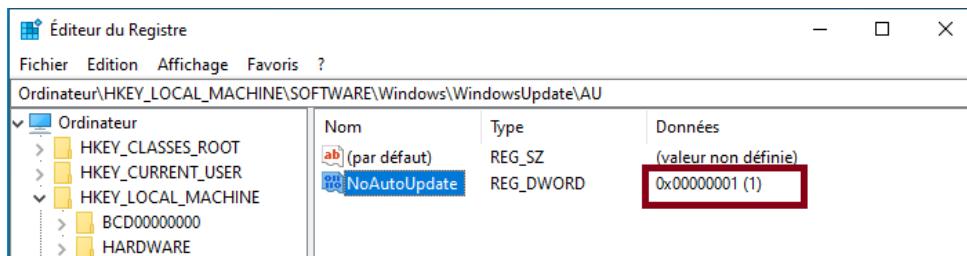


FIGURE 2.11 – Modification de la valeur du Registre pour désactiver les mises à jour automatiques

Passons maintenant à l'activation des services Internet Information Services (IIS) sur la machine virtuelle Windows 10, une étape clé pour simuler une vulnérabilité alignée sur la CVE-2022-21907. Cette vulnérabilité, affectant spécifiquement le traitement des requêtes HTTP/2 par IIS, nécessite l'activation des services Web pour permettre une reproduction fidèle et une analyse approfondie de l'exploitation de la faille. Cette configuration nous permet d'étudier comment des requêtes HTTP/2 mal formées peuvent entraîner un crash du système, offrant une compréhension précise des mécanismes et des impacts de l'attaque.

7 - Utilisez la fonction de recherche de Windows pour trouver et ouvrir le "Panneau de configuration" et sélectionnez "Désinstaller un programme" sous la catégorie Programmes.

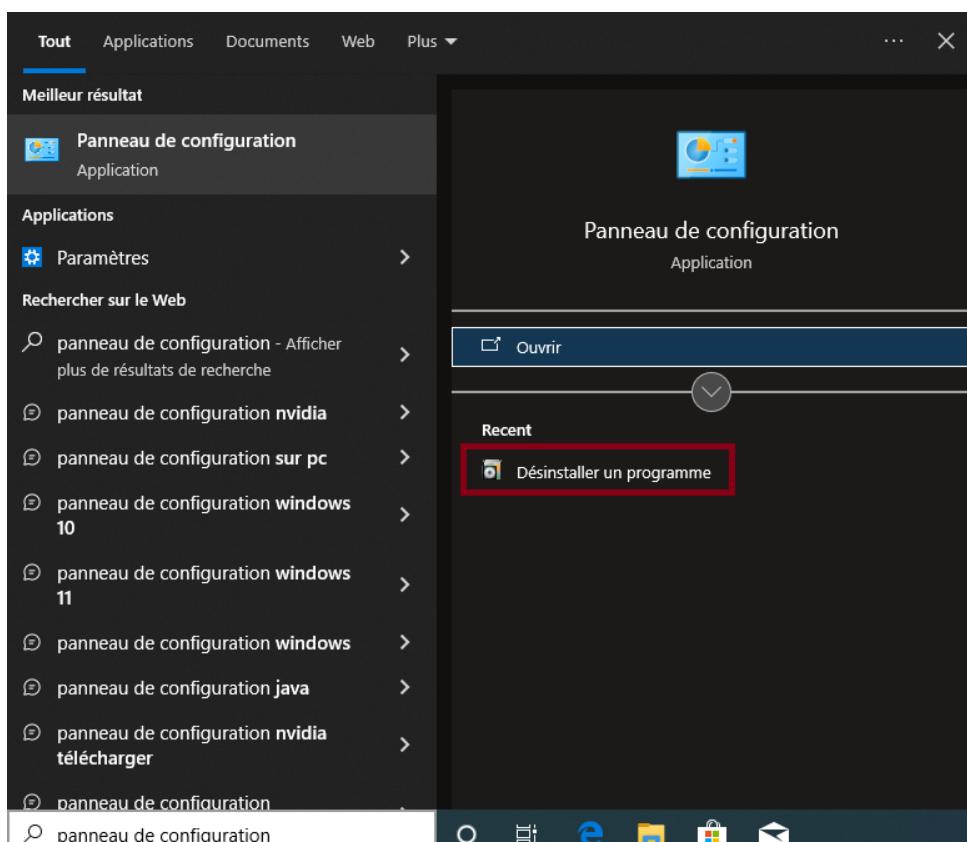


FIGURE 2.12 – Recherche du Panneau de configuration dans Windows pour la désinstallation de programmes

8 - Sur la gauche, cliquez sur "Activer ou désactiver des fonctionnalités Windows".

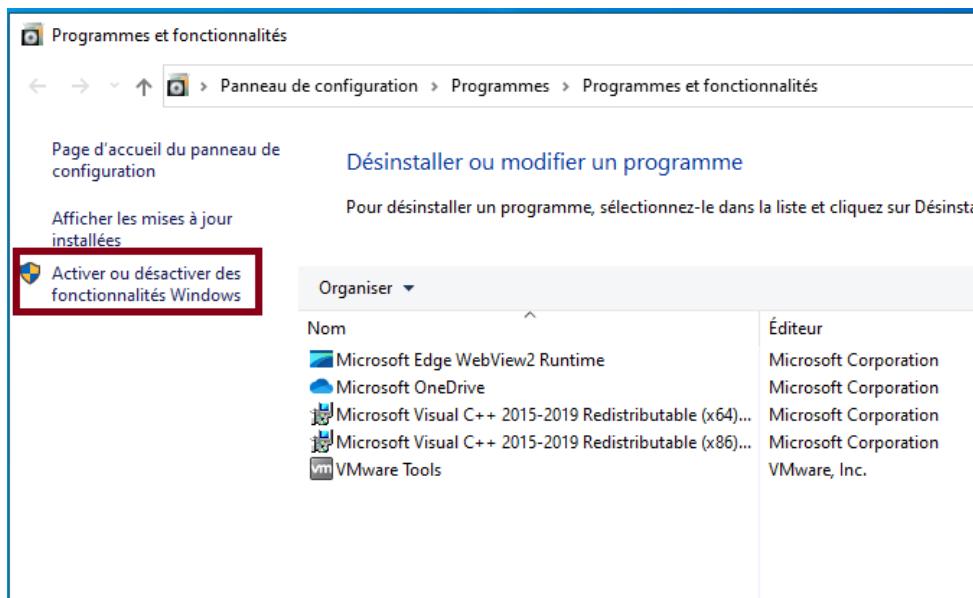


FIGURE 2.13 – Activation de fonctionnalités Windows depuis le Panneau de configuration

9 - Dans la fenêtre qui s'ouvre, vous pouvez cocher ou décocher les fonctionnalités que vous devez activer. Vous devez activer "Internet Information Services" et ses sous-composants comme "Outils d'administration Web" et "Services World Wide Web".

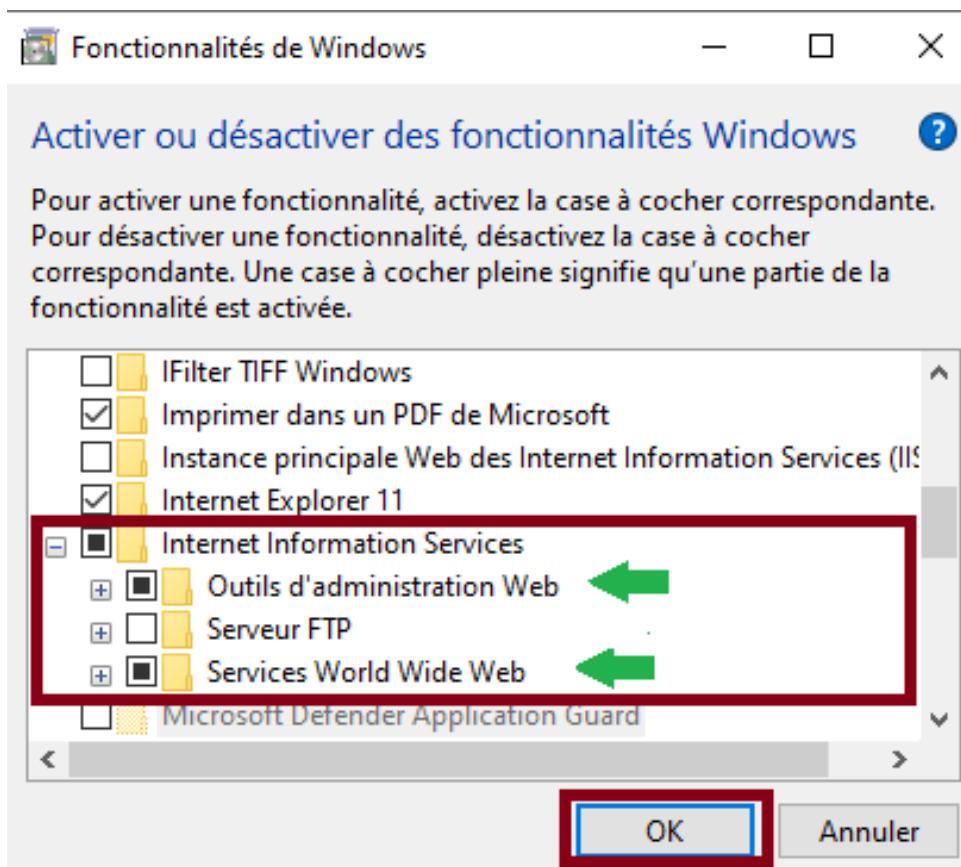


FIGURE 2.14 – Activation des services Internet Information Services dans Windows

Après avoir finalisé la configuration de la machine virtuelle, celle-ci est désormais prête pour le téléversement sur Vagrant Cloud. Pour utiliser cette machine, il n'est pas nécessaire

de refaire toute la configuration manuellement. Il suffit simplement de copier le contenu du fichier Vagrantfile ci-dessous :

```
1 Vagrant.configure("2") do |config|
2   config.vm.box = "marouaneekamal001/Win10_2004_x64"
3   config.vm.box_version = "1.0.0"
4   config.vm.box_url = "https://vagrantcloud.com/marouaneekamal001/
5   Win10_2004_x64"
6   config.vm.define "WindowsVul" do |hq|
7     hq.vm.box = "marouaneekamal001/Win10_2004_x64"
8     hq.vm.network "private_network", type: "dhcp",
9     virtualbox__intnet: true
10    hq.vm.provider "virtualbox" do |vb|
11      vb.name = "WindowsVul"
12      vb.cpus = 2
13      vb.customize ["modifyvm", :id, "--nic3", "natnetwork"]
14      vb.customize ["modifyvm", :id, "--nat-network3", "
15      MyCustomisedNet"]
16    end
17  end
18 end
```

Une fois copié, exécutez la commande :

```
1 vagrant up
```

dans votre terminal ou invite de commande, et attendez la fin de l'installation. Votre machine virtuelle sera alors prête à l'emploi. N'oubliez pas de vérifier que vous avez bien Vagrant et VirtualBox installés sur votre système avant de lancer la commande.

2.1.2 Configuration de la machine virtuelle Kali Linux (Attaquante)

Pour préparer la machine virtuelle Kali Linux qui servira d'attaquante dans notre environnement de test, la configuration requise est relativement directe. Il est impératif d'avoir Python 3 installé, car c'est la version de Python avec laquelle notre script d'exploitation est compatible. Python 3 est souvent préinstallé sur Kali Linux ; cependant, si ce n'est pas le cas, il peut être facilement installé via le gestionnaire de paquets. En plus de Python 3, il est essentiel d'avoir accès à un terminal, qui sera utilisé pour exécuter notre script.

En résumé, la machine attaquante doit être configurée avec les éléments suivants :

- Python 3 : Assurez-vous que Python 3 est installé et prêt à l'emploi. Vous pouvez vérifier cela en exécutant `python3 --version` dans le terminal.
- Terminal : Un environnement de ligne de commande où vous exécuterez le script d'exploitation.
- Dépendances Python : les bibliothèques tierces comme `requests` ou `loguru` doivent être installées en utilisant `pip`, le gestionnaire de paquets Python, via la commande `pip3 install requests loguru`.

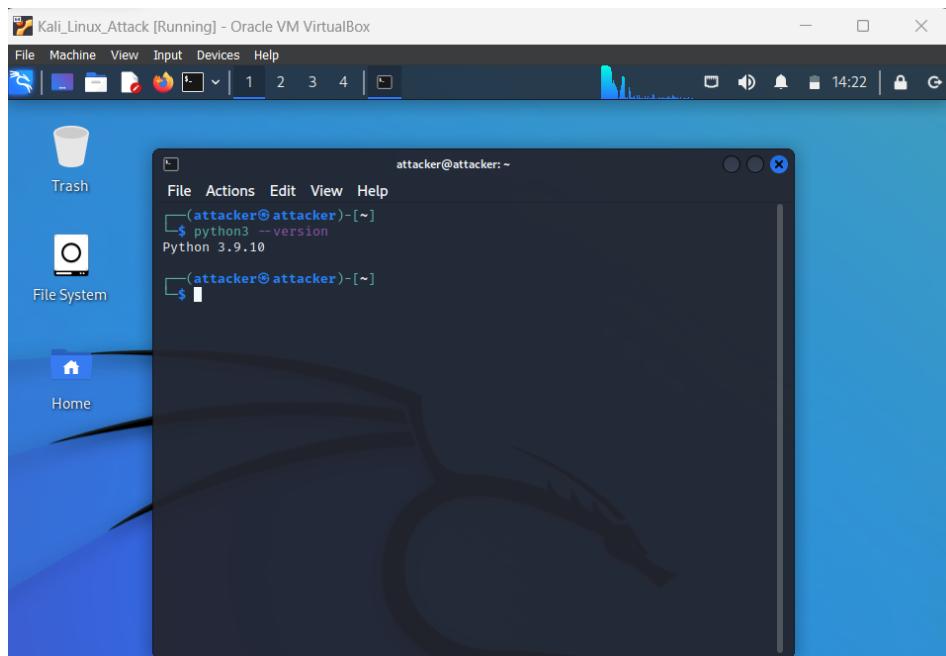


FIGURE 2.15 – Vérification de la version de Python sur Kali Linux

La mise en place de la machine virtuelle Kali Linux destinée à l’attaque se fait avec une simplicité équivalente à celle de la machine précédente. Aucune configuration manuelle complexe n’est nécessaire pour vous. Vous aurez simplement besoin du fichier Vagrantfile suivant pour définir et démarrer votre environnement :

```

1 Vagrant.configure("2") do |config|
2   config.vm.box = "base"
3   config.vm.box_version = "1.0.0"
4   config.vm.box_url = "https://vagrantcloud.com/marouane_kamal001/
Kali_Linux_x64"
5   config.vm.define "Kali_Linux_Attack" do |hq|
6     hq.vm.box = "marouane_kamal001/Kali_Linux_x64"
7     hq.vm.network "private_network", type: "dhcp",
virtualbox__intnet: true
8     hq.vm.provider "virtualbox" do |vb|
9       vb.name = "Kali_Linux_Attack"
10      vb.cpus = 2
11      vb.customize ["modifyvm", :id, "--nic3", "natnetwork"]
12      vb.customize ["modifyvm", :id, "--nat-network3", "
MyCustomisedNet"]
13    end
14  end
15 end

```

Pour lancer la machine, exécutez simplement la commande :

```
1 vagrant up
```

dans votre terminal. Patientez jusqu’à ce que le processus d’installation soit complété. Une fois terminé, vous disposerez d’une machine Kali Linux prête à l’emploi, préconfigurée pour l’exploitation de la faille. Cette machine sera également équipée de toutes les dépendances nécessaires, y compris le script d’exploitation, vous permettant ainsi de commencer vos tests sans délai supplémentaire. Après le démarrage de la machine attaquante, vous serez invité à entrer des identifiants pour accéder à la machine. Voici les informations de connexion nécessaires :

```

1 username : attacker
2 password : att

```

2.2 Développement du Code d'Attaque

Le script sera exécuté sur la machine attaquante et il faudra lui passer en argument l'adresse IP de la machine victime. Il est essentiel de s'assurer que les deux machines se trouvent dans le même réseau pour permettre au script d'interagir avec la machine cible. Voici un résumé explicatif du fonctionnement du script :

Première Poignée de Main (First Handshake) : Le script tente d'abord de faire une requête HTTP GET standard au serveur cible pour vérifier si le serveur est vivant et répond normalement.

Poignée de Main de la POC (POC Handshake) : Ensuite, il envoie une requête GET avec une valeur d'en-tête 'Accept-Encoding' personnalisée conçue pour déclencher la vulnérabilité. Cette valeur d'en-tête est délibérément malformée et comprend une chaîne de caractères excessivement longue, qui serait susceptible de provoquer un débordement de tampon si le serveur est vulnérable.

Vérification de la Poignée de Main (Verification Handshake) : Après l'envoi de la requête malveillante, le script tente de faire une autre requête GET standard pour vérifier si le serveur a planté et redémarré (un signe que l'exploit a réussi et a causé un BSOD).

Boucle pour la Vérification Déterministe : Le script entre dans une boucle où il continue à vérifier si le serveur est devenu non réactif, ce qui indiquerait une exploitation réussie menant à un crash du système (BSOD).

La valeur de l'en-tête 'Accept-Encoding' dans le script est considérablement plus longue et plus complexe que ce qui serait typiquement utilisé dans un trafic légitime. Cet aspect du PoC est conçu pour exploiter la vulnérabilité spécifique dans la manière dont le serveur analyse l'encodage de transfert 'chunked', qui pourrait conduire à un déni de service (DoS) via BSOD ou potentiellement à une exécution de code à distance (RCE), selon la nature spécifique de la vulnérabilité et comment elle est exploitée.

2.3 Preuve de concept

Pour exploiter la vulnérabilité, veuillez suivre attentivement les étapes suivantes :

1 - Lancement des Machines Virtuelles : Commencez par ouvrir vos machines virtuelles en cours d'exécution.

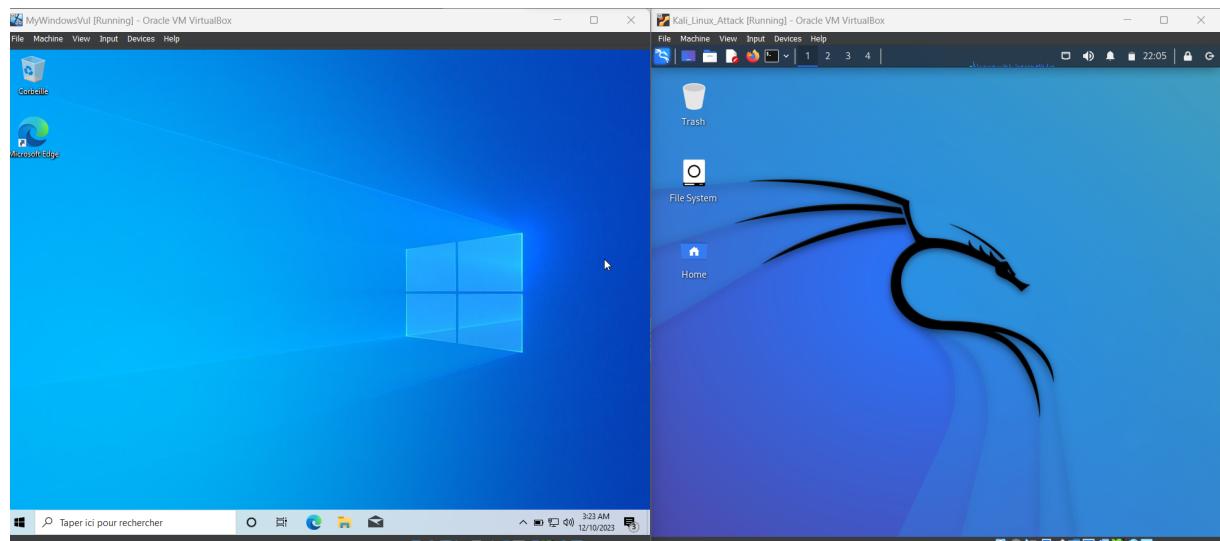


FIGURE 2.16 – Environnements de bureau Windows et Kali Linux côté à côté

2 - Obtention de l'Adresse IP de la Machine Victime : * Sur la machine Windows 10, ouvrez l'invite de commande (cmd) et tapez ipconfig pour afficher les informations réseau, notamment l'adresse IPv4 configurée préalablement.

```
C:\Users\victim>ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet :

    Suffrage DNS propre à la connexion. . . . . : 
    Adresse IPv6 de liaison locale. . . . . : fe80::4d4e:2fdb:b18f:ff93%9
    Adresse IPv4. . . . . : 10.0.2.15
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : 10.0.2.2

Carte Ethernet Ethernet 2 :

    Suffrage DNS propre à la connexion. . . . . : 
    Adresse IPv6 de liaison locale. . . . . : fe80::2dae:888:2031:9016%13
    Adresse d'autoconfiguration IPv4 . . . . . : 169.254.144.22
    Masque de sous-réseau. . . . . : 255.255.0.0
    Passerelle par défaut. . . . . : 

Carte Ethernet Ethernet 3 :

    Suffrage DNS propre à la connexion. . . . . : 
    Adresse IPv6 de liaison locale. . . . . : fe80::1577:d495:5b8c:d58a%6
    Adresse IPv4. . . . . : 192.168.100.4 ←
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : 192.168.100.1

C:\Users\victim>
```

FIGURE 2.17 – Configuration IP d'une machine Windows affichée dans l'invite de commandes

- Notez l'adresse IPv4 affichée. Elle a été établie via la commande VBoxManage mentionnée précédemment et sera utilisée ultérieurement.
 - Attention ! L'adresse IP sur votre machine virtuelle peut différer de celle utilisée dans cet exemple. Assurez-vous d'utiliser l'adresse IP obtenue après avoir exécuté la commande ipconfig. Notez que le Gateway par défaut doit être 192.168.100.1, comme défini par la commande VBoxManage. Ainsi, votre adresse IP sur la machine victime devrait être de la forme 192.168.100.X.

3 - Préparation de l'Attaque sur la Machine Kali Linux :

• Naviguez jusqu'au répertoire contenant le script d'exploitation en utilisant la commande :

1 cd CVE-2022-21907-Exploit

```

attacker@attacker: ~/CVE-2022-21907-Exploit
File Actions Edit View Help
└─(attacker㉿attacker)─[~]
$ cd CVE-2022-21907-Exploit
└─(attacker㉿attacker)─[~/CVE-2022-21907-Exploit]
$ 

```

FIGURE 2.18 – Changement de répertoire vers le dossier d’exploitation CVE-2022-21907 sur Kali Linux

4 - Exécution du Script d’Exploitation :

Sur la machine Kali Linux, exécutez le script Python pour déclencher le crash de la machine victime :

```
1 python3 CVE-2022-21907-exploit.py -i VICTIM_MACHINE_IP
```

Remplacez VICTIM_MACHINE_IP par l’adresse IP que vous avez notée précédemment sur la machine victime.

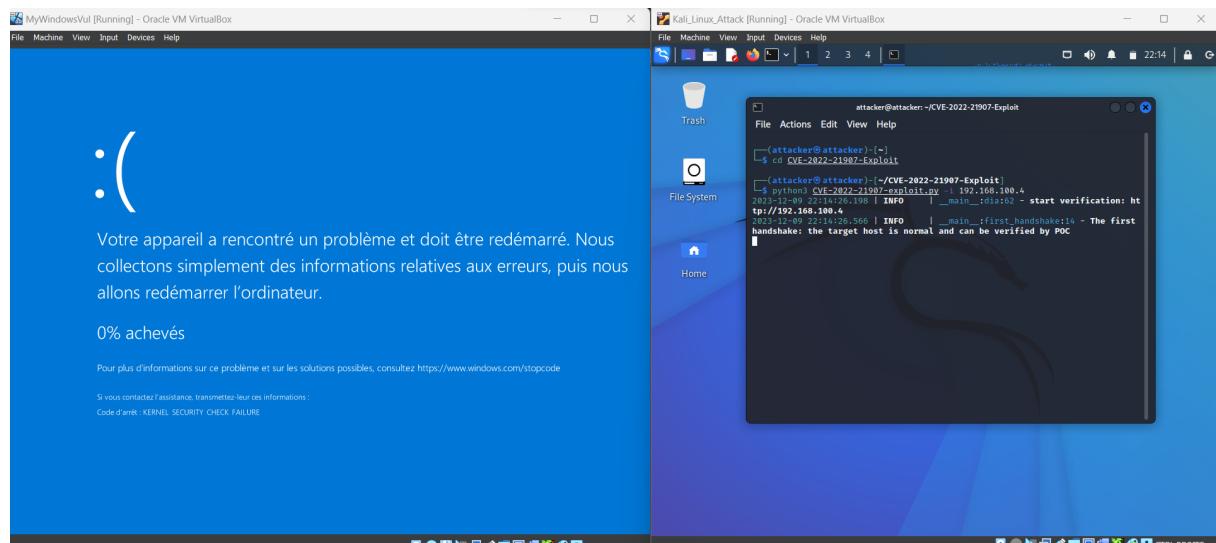


FIGURE 2.19 – Écran bleu de la mort sur une machine Windows indiquant un crash système

Félicitations ! Si tout se passe comme prévu, la machine victime devrait maintenant être en état de crash (écran bleu). Ces étapes fournissent une procédure claire et méthodique pour l’exploitation de la vulnérabilité CVE-2022-21907, en prenant soin de souligner les points critiques comme l’adresse IP correcte et le déroulement précis de l’attaque.

Important : Afin de tester efficacement l’exploitation de cette vulnérabilité, je vous recommande fortement de consulter le dépôt GitHub suivant. Vous y trouverez des informations détaillées sur la procédure à suivre, y compris les prérequis et les outils nécessaires. Pour accéder à ces ressources, veuillez visiter le lien suivant : [CVE-2022-21907 GitHub Repo](#)

Chapitre 3

Préconisations de Sécurité

3.1 Préconisations pour Limiter l'Impact

Afin de limiter l'impact de l'exploitation des vulnérabilités telles que CVE-2022-21907, plusieurs stratégies peuvent être mises en œuvre :

- Application immédiate des mises à jour de sécurité dès leur publication par Microsoft, en mettant l'accent sur la correction rapide des composants critiques tels que http.sys.
- Mise en place d'une stratégie de défense en profondeur comprenant l'utilisation de pare-feu, de systèmes de prévention d'intrusion, et d'autres mécanismes de filtrage du trafic pour contrôler strictement l'accès au serveur.
- Réalisation d'audits de sécurité réguliers pour identifier et corriger les mauvaises configurations susceptibles d'exposer les serveurs à des risques supplémentaires.
- Utilisation d'outils de surveillance et d'analyse du réseau pour détecter les activités suspectes, y compris les tentatives d'exploitation de vulnérabilités connues.
- Sensibilisation et formation continue du personnel pour les familiariser avec les menaces courantes et les bonnes pratiques à adopter en matière de cybersécurité.

3.2 Préconisations pour Empêcher l'Exploitation

Pour empêcher l'exploitation des vulnérabilités :

- Mettre en place un programme régulier de formation à la sécurité pour les développeurs, afin de les sensibiliser aux meilleures pratiques de codage sécurisé.
- Intégrer la sécurité dès les premières étapes du cycle de développement logiciel (SDLC), notamment via des revues de code et l'utilisation de scanners de vulnérabilités.
- Adopter une approche de programmation défensive, qui inclut la validation des entrées, la gestion appropriée des erreurs, et l'utilisation de techniques de sandboxing pour isoler les composants systèmes.
- Réduire la surface d'attaque en désactivant les services et les fonctionnalités inutiles, en limitant les priviléges des applications et des services, et en mettant en œuvre une séparation des priviléges.

3.3 Bonnes Pratiques de Sécurité

Les bonnes pratiques pour limiter la menace de CVE-2022-21907 et d'autres vulnérabilités similaires comprennent :

- Encourager l'utilisation de l'authentification multi-facteurs (MFA) pour tous les utilisateurs, en particulier ceux ayant accès à des informations sensibles.
- Mettre en œuvre des solutions de chiffrement pour sécuriser les données en transit et au repos, et assurer l'intégrité des données critiques.
- Établir des politiques de sauvegarde et de restauration claires pour permettre une récupération rapide en cas de compromission.
- Conduire des exercices de simulation d'attaques (red team exercises) pour tester la résilience des systèmes et des équipes face à des scénarios d'attaque réalistes.

3.4 Amélioration des Pratiques d'Initialisation en Développement pour Prévenir les Failles

La vulnérabilité CVE-2022-21907 met en lumière les conséquences d'une initialisation inadéquate des structures de données dans le développement logiciel. Pour limiter l'apparition de telles failles, il est crucial d'intégrer des pratiques d'initialisation rigoureuses dans les processus de développement :

- Initialisation Systématique des Variables : S'assurer que toutes les variables et structures de données sont correctement initialisées avant leur utilisation. Cela inclut l'initialisation à des valeurs sûres pour éviter les comportements imprévus ou l'exploitation de données non initialisées.
- Revue de Code Ciblée : Mettre en place des revues de code spécifiquement axées sur l'identification et la correction des problèmes d'initialisation, en particulier dans les composants critiques pour la sécurité.
- Tests Automatisés : Utiliser des outils de test automatique pour détecter les problèmes d'initialisation des variables et des structures de données, renforçant ainsi la détection des failles avant la mise en production.
- Formation des Développeurs : Sensibiliser et former les développeurs aux risques liés à l'initialisation inadéquate et aux meilleures pratiques pour une gestion sécurisée de la mémoire et des ressources système.
- Utilisation de Langages de Programmation avec Gestion de la Mémoire : Privilégier, lorsque c'est possible, l'utilisation de langages de programmation qui offrent une gestion automatique de la mémoire et qui réduisent les risques d'erreur d'initialisation.

En mettant l'accent sur une initialisation rigoureuse et sécurisée, les équipes de développement peuvent réduire significativement le risque de vulnérabilités dues à des erreurs d'initialisation, contribuant ainsi à renforcer la sécurité globale des logiciels développés.

3.5 Cible de sécurité

Pour contrer efficacement la CVE-2022-21907, il est essentiel d'adopter une Politique de Sécurité des Systèmes d'Information (PSSI) solide. La PSSI est un cadre stratégique définissant les normes, règles, et procédures pour garantir un haut niveau de sécurité informatique au sein d'une organisation. Son rôle est crucial dans la gestion proactive des risques liés à la cybersécurité, comme la faille CVE-2022-21907, permettant ainsi de déterminer des actions préventives et curatives, et de guider l'organisation dans la protection de ses systèmes d'information contre les menaces potentielles.

3.5.1 Objectif

Mettre en place des mesures de sécurité rigoureuses pour protéger les données sensibles et confidentielles. L'objectif est de maintenir l'intégrité et la confidentialité des informations clients et internes, en minimisant les risques d'accès non autorisés, tout en assurant la disponibilité des données pour un fonctionnement efficace.

3.5.2 Criticité de la faille

Le score CVSS élevé de CVE-2022-21907 indique une menace majeure. Les impacts potentiels incluent la compromission du système, l'accès non autorisé à des données sensibles et l'interruption des services critiques.

3.5.3 Actions préventives et curatives

Pour une gestion approfondie de CVE-2022-21907, on doit appliquer les correctifs de sécurité dès leur disponibilité, désactiver les configurations vulnérables, surveiller les systèmes pour détecter toute activité suspecte et isoler rapidement les systèmes affectés pour limiter la propagation.

Stratégies de Sécurité pour la Gestion de CVE-2022-21907 :

- Utilisateurs : Sensibiliser spécifiquement les utilisateurs aux risques liés à CVE-2022-21907, en mettant l'accent sur les bonnes pratiques de sécurité informatique.
- Biens à protéger : La protection doit se concentrer sur les serveurs et les ordinateurs personnels exécutant les versions de Windows affectées par CVE-2022-21907. Cela inclut des serveurs web utilisant IIS, des serveurs d'entreprise, ainsi que des postes de travail et des laptops dans l'environnement de l'entreprise.
- Menaces : Identifier et comprendre les risques spécifiques liés à CVE-2022-21907, y compris les attaques par déni de service et les intrusions malveillantes exploitant cette faille.
- Fonctions de sécurité : Implémenter des mises à jour de sécurité, une surveillance réseau efficace, des politiques de gestion des accès strictes, et des programmes de formation continue pour les employés.

Conclusion Générale

En conclusion, l'analyse approfondie de la vulnérabilité CVE-2022-21907 révèle non seulement les défis intrinsèques à la sécurisation des infrastructures informatiques modernes, mais aussi l'importance capitale de stratégies de cybersécurité adaptatives et proactives. Cette faille, caractérisée par un score CVSS de 9.8, met en exergue la criticité des vulnérabilités qui permettent l'exécution de code à distance et les dénis de service, et souligne la nécessité d'une vigilance constante ainsi que d'une réponse rapide et coordonnée face aux menaces.

La portée de CVE-2022-21907, affectant une vaste gamme de systèmes d'exploitation Windows utilisés tant au niveau des serveurs que des clients, illustre la propagation et l'impact significatif que peut avoir une seule vulnérabilité non adressée. Cela démontre l'importance d'une maintenance régulière des systèmes, comprenant l'application diligente de correctifs de sécurité et l'examen rigoureux des configurations système pour minimiser les surfaces d'attaque.

La découverte que la vulnérabilité est due à un défaut d'initialisation de la mémoire offre une leçon cruciale sur le rôle essentiel que joue la gestion de la mémoire dans le développement de logiciels sécurisés. La correction stratégique apportée par Microsoft, consistant en l'introduction d'appels à memset() dans la fonction http!UlpAllocateFastTracker(), ferme efficacement la faille et prévient les crashes systémiques qui en découlaient.

L'incident CVE-2022-21907 met en lumière la nécessité d'une Politique de Sécurité des Systèmes d'Information (PSSI) robuste, qui devrait inclure non seulement des mesures préventives comme la mise à jour des systèmes, mais aussi des stratégies de réponse aux incidents pour contenir et atténuer les attaques lorsqu'elles se produisent. La formation et la sensibilisation des utilisateurs aux risques et aux meilleures pratiques de cybersécurité constituent un autre pilier de la défense contre les cybermenaces.

En définitive, la cybersécurité ne doit pas être perçue comme une série de mesures réactives, mais plutôt comme une culture de vigilance continue, d'évaluation des risques et d'amélioration continue. La collaboration entre organisations, la recherche de cybersécurité, et la partage d'informations sont des éléments clés pour anticiper et contrer efficacement les menaces émergentes. Alors que les attaquants continuent d'innover, la communauté de la sécurité doit rester agile, apprendre de chaque incident et renforcer les infrastructures pour les défis de demain.

Bibliographie

- [1] HTTP Protocol Stack Remote Code Execution Vulnerability :
<https://orca.security/resources/blog/http-protocol-stack-remote-code-execution-security-vulnerability-cve-2022-21907>
- [2] How to exploit the HTTP.sys Remote Code Execution vulnerability (CVE-2022-21907) :
<https://pentest-tools.com/blog/exploit-http-sys-rce-vulnerability-cve-2022-21907>
- [3] Proof of Concept : CVE-2022-21907 HTTP Protocol Stack Remote Code Execution Vulnerability :
<https://www.coresecurity.com/core-labs/articles/proof-concept-cve-2022-21907-http-protocol-stack-remote-code-execution>
- [4] CVE-2021-31166 : A WORMABLE CODE EXECUTION BUG IN HTTP.SYS :
<https://www.zerodayinitiative.com/blog/2021/5/17/cve-2021-31166-a-wormable-code-execution-bug-in-httpsys>
- [5] Patch diffing CVE-2022-21907 :
<https://piffd0s.medium.com/patch-diffing-cve-2022-21907-b739f4108eee>
- [6] Vulnérabilité d'exécution de code à distance dans la pile du protocole HTTP :
<https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-21907>
- [7] Microsoft's January 2022 Patch Tuesday Addresses 97 CVEs (CVE-2022-21907) :
<https://www.tenable.com/blog/microsofts-january-2022-patch-tuesday-addresses-97-cves-cve-2022-21907>
- [8] HTTP.sys Remote Code Execution vulnerability (CVE-2022-21907) :
<https://crashtest-security.com/cve-2022-21907-http-vulnerability/>
- [9] CVE-2022-21907 Detail :
<https://nvd.nist.gov/vuln/detail/CVE-2022-21907>
- [10] Analysis of Microsoft CVE-2022-21907 :
<https://www.fortinet.com/blog/threat-research/analysis-of-microsoft-cve-2022-21907>