

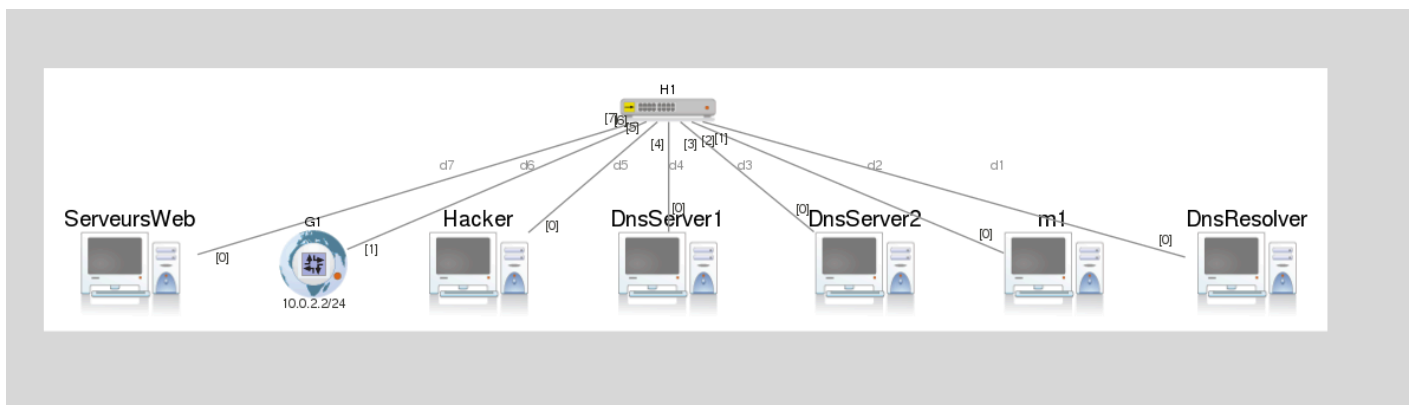
Compte rendu de la Sécurisation des Services : Sécurisation DNS

Méthodologie

Ce TP adopte une démarche pratique et expérimentale, structurée autour d'un processus itératif incluant la mise en place, la configuration, les tests et le dépannage. Cette approche favorise l'acquisition de compétences techniques sur les outils et protocoles réseaux tout en approfondissant la compréhension des menaces de sécurité actuelles et des stratégies de mitigation.

En recréant un environnement proche des conditions réelles et en explorant divers scénarios de dépannage, ce TP a pour objectif de préparer les participants à détecter et à répondre efficacement aux incidents de sécurité affectant les systèmes de réseau DNS.

Mise en place de la topologie du TP



Aperçu général

La topologie réseau mise en place pour ce TP est conçue pour reproduire un environnement réseau réaliste, intégrant des composants interconnectés qui simulent les rôles et interactions typiques d'un écosystème DNS. Grâce à l'outil Marionnet, il est possible de créer, configurer et manipuler cette topologie de manière flexible et contrôlée.

Paramétrage du Réseau
















Chaque machine virtuelle est associée à une interface réseau dédiée, configurée avec une adresse IP statique spécifique et une passerelle par défaut pour assurer la communication au sein du réseau simulé. Les paramètres réseau, tels que les adresses IP et les configurations de

routage, sont méticuleusement définis pour correspondre aux rôles assignés à chaque machine dans la topologie.

Initialisation avec root



L'attribution des adresses

Documents							
Nom	Type	Adresse MAC	MTU	Adresse IPv4	Passerelle IPv4	Adresse IPv6	Passerelle IPv6
▼ m1							
eth0		02:04:06:48:80:f6	1500	10.0.2.1	10.0.0.2		
▼ DnsResolver							
eth0		02:04:06:fa:d4:4b	1500	10.0.2.10	10.0.0.2		
▼ DnsServer1							
eth0		02:04:06:c9:93:92	1500	10.0.2.100	10.0.0.2		
▼ DnsServer2							
eth0		02:04:06:de:0b:0a	1500	10.0.2.110	10.0.0.2		
▼ Hacker							
eth0		02:04:06:60:47:e7	1500	10.0.2.5	10.0.0.2		
▼ ServeursWeb							
eth0		02:04:06:e9:3e:23	1500	10.0.2.200	10.0.0.2		
eth1		02:04:06:3e:e5:e7	1500	10.0.2.202	10.0.0.2		
eth2		02:04:06:60:66:bc	1500				
eth3		02:04:06:57:a8:f3	1500				

Test de la connectivité

De DnsResolver vers DnsServer2

```
[0 root@DnsResolver ~]$ ping 10.0.2.110
PING 10.0.2.110 (10.0.2.110) 56(84) bytes of data.
64 bytes from 10.0.2.110: icmp_req=1 ttl=64 time=2.79 ms
^C
--- 10.0.2.110 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.798/2.798/2.798/0.000 ms
[0 root@DnsResolver ~]$
```

De m1 vers hacker

```
No mail.
[0 root@m1 ~]$ ping 10.0.2.5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data.
64 bytes from 10.0.2.5: icmp_req=1 ttl=64 time=2.19 ms
64 bytes from 10.0.2.5: icmp_req=2 ttl=64 time=1.52 ms
^C
--- 10.0.2.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1009ms
rtt min/avg/max/mdev = 1.521/1.856/2.191/0.335 ms
[0 root@m1 ~]$
```

De DnsServer2 vers DnsResolver

```

[0 root@DnsServer2 ~]$ ping 10.0.2.10
PING 10.0.2.10 (10.0.2.10) 56(84) bytes of data.
64 bytes from 10.0.2.10: icmp_req=1 ttl=64 time=1.80 ms
64 bytes from 10.0.2.10: icmp_req=2 ttl=64 time=1.32 ms
^C
--- 10.0.2.10 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1004ms
rtt min/avg/max/mdev = 1.321/1.560/1.800/0.242 ms
[0 root@DnsServer2 ~]$

```

De DnsServer1 vers DnsResolver

```

[1 root@DnsServer1 ~]$ ping 10.0.2.10
PING 10.0.2.10 (10.0.2.10) 56(84) bytes of data.
64 bytes from 10.0.2.10: icmp_req=1 ttl=64 time=2.23 ms
^C
--- 10.0.2.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.237/2.237/2.237/0.000 ms

```

2.1 Configuration du Serveur DNS (DnsServer1)

La configuration du serveur DNS vise à établir une résolution de noms efficace et sécurisée pour le domaine simulé iut-villetaneuse.fr, essentielle pour le fonctionnement fiable du réseau.

Méthodes et Étapes

Initialisation:Création de l'environnement requis en configurant le répertoire `/var/named/master` destiné à héberger les fichiers de configuration des zones DNS

```

bash: cd: /var/named/master: no such file or directory
[1 root@DnsServer1 ~]$ sudo mkdir -p /var/named/master
[0 root@DnsServer1 ~]$ sudo chown -R bind:bind /var/named/master
[0 root@DnsServer1 ~]$ sudo chmod -R 750 /var/named/master
[0 root@DnsServer1 ~]$

```

Configuration BIND : Modification du fichier de configuration principal `/etc/bind/named.conf`. La configuration suivante a été établie :

```
DnsServer1 (debian-wheezy-08367)
GNU nano 2.2.6 File: /etc/bind/named.conf

options {
    directory "/var/named";
    listen-on { any; };
    allow-query { any; };
};

zone iut-villetaneuse.fr {
    type master ;
    file "master/iut-villetaneuse.fr" ;
    allow-update { none; };
    notify no;
};

[ Read 12 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Définition des options de BIND pour écouter sur toutes les interfaces (listen-on { any; };) et permettre les requêtes depuis n'importe quelle source (allow-query { any; };).

Spécification de la zone iut-villetaneuse.fr avec les paramètres appropriés pour désigner DnsServer1 comme le serveur DNS maître.

Fichier de Zone : Création et configuration du fichier de zone /var/named/master/iut-villetaneuse.fr :

Ensuite, il faut créer le fichier de zone suivant : /var/named/master/iut-villetaneuse.fr :

```
DnsServer1 (debian-wheezy-08367)
GNU nano 2.2.6 File: /var/named/master/iut-villetaneuse.fr

$TTL 180
@ IN SOA DnsServer1.iut-villetaneuse.fr. root.iut-villetaneuse.fr. (
20230102
;Serial
120 ;Refresh
60 ;Retry
300 ;Expire
180 ;Minimum TTL
)
DnsServer1 IN NS DnsServer1.iut-villetaneuse.fr.;
DnsServer1 IN A 10.0.2.100
DnsServer2 IN A 10.0.2.110
www IN A 10.0.2.200
```

Enregistrement SOA (Start of Authority) indiquant DnsServer1 comme source d'autorité principale et définissant les paramètres de rafraîchissement et de réessai.

Enregistrements NS (Name Server) établissant DnsServer1 comme serveur de noms pour le domaine.

Enregistrements A (Address) attribuant les adresses IP aux noms DnsServer1, DnsServer2, et www, garantissant que les requêtes seront résolues vers les bonnes adresses.

Validation : Utilisation de named-checkconf pour valider la syntaxe du fichier de configuration /etc/bind/named.conf et de named-checkzone pour vérifier le fichier de zone

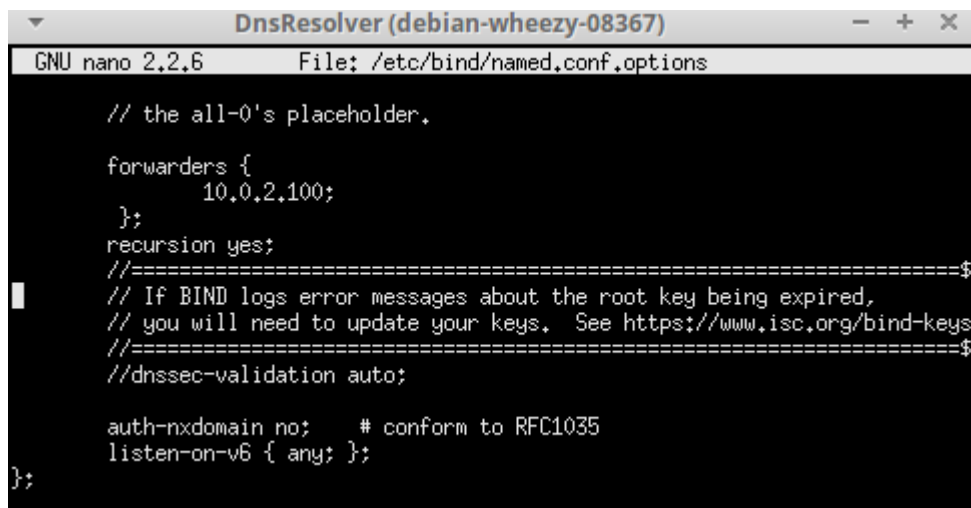
```
[0 root@DnsServer1 /var/named/master]$ named-checkzone zone iut-villetaneuse.fr
zone zone/IN: loaded serial 20230102
OK
[0 root@DnsServer1 /var/named/master]$ named-checkconf /etc/bind/named.conf
[0 root@DnsServer1 /var/named/master]$
```

Démarrage de BIND : Commande `service named start` ou `/etc/init.d/bind9 start` exécutée pour lancer le service DNS.

```
[0 root@DnsServer1 /var/named/master]$ /etc/init.d/bind9 restart
[....] Stopping domain name service...: bind9rndc: connect failed: 127.0.0.1#953
: connection refused
. ok
[ ok ] Starting domain name service...: bind9.
[0 root@DnsServer1 /var/named/master]$
```

2.1.2 Configuration du serveur resolver/forwarder :

Pour le serveur DnsResolver, on a uniquement besoin de modifier le fichier `/etc/bind/named.conf.options`:



```
DnsResolver (debian-wheezy-08367)
GNU nano 2.2.6 File: /etc/bind/named.conf.options

// the all-0's placeholder.

forwarders {
    10.0.2.100;
};
recursion yes;
//=====
// If BIND logs error messages about the root key being expired,
// you will need to update your keys. See https://www.isc.org/bind-keys
//=====
//dnssec-validation auto;

auth-nxdomain no;    # conform to RFC1035
listen-on-v6 { any; };
};
```

2.1.3 Test de résolution de nom :

Avec `nameserver 10.0.2.1`

```

[0 root@m1 ~]$ dig www.iut-villetaneuse.fr

; <<> DiG 9.8.4-rpz2+rl005.12-P1 <<> www.iut-villetaneuse.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12682
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.iut-villetaneuse.fr.      IN      A

;; ANSWER SECTION:
www.iut-villetaneuse.fr. 180     IN      A      10.0.2.200

;; AUTHORITY SECTION:
iut-villetaneuse.fr.    180     IN      NS      DnsServer1.iut-villetaneuse.fr.

;; ADDITIONAL SECTION:
DnsServer1.iut-villetaneuse.fr. 180 IN  A      10.0.2.100

;; Query time: 204 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Mon Jan 13 11:38:42 2025
;; MSG SIZE rcvd: 98

[0 root@m1 ~]$ █

```

Avec NS

```

[0 root@m1 ~]$ dig iut-villetaneuse.fr NS

; <<> DiG 9.8.4-rpz2+rl005.12-P1 <<> iut-villetaneuse.fr NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56431
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;iut-villetaneuse.fr.      IN      NS

;; ANSWER SECTION:
iut-villetaneuse.fr.    105     IN      NS      DnsServer1.iut-villetaneuse.fr.

;; ADDITIONAL SECTION:
DnsServer1.iut-villetaneuse.fr. 105 IN  A      10.0.2.100

;; Query time: 84 msec
;; SERVER: 10.0.2.10#53(10.0.2.10)
;; WHEN: Mon Jan 13 11:39:57 2025
;; MSG SIZE rcvd: 78

[0 root@m1 ~]$ █

```

Analyse du trafic DNS

7	10.00897800(10.0.2.1	10.0.2.10	DNS	83 Standard query 0xf3ac
8	10.00901000(10.0.2.10	10.0.2.1	ICMP	111 Destination unreachab
72	529.5799780(10.0.2.1	10.0.2.10	DNS	83 Standard query 0xf91c
75	529.6265310(10.0.2.10	10.0.2.100	DNS	94 Standard query 0x92a9
76	529.6265440(10.0.2.10	10.0.2.100	DNS	70 Standard query 0x7c7e
77	529.6710330(10.0.2.100	10.0.2.10	DNS	151 Standard query respon
80	529.7153140(10.0.2.10	10.0.2.1	DNS	140 Standard query respon

Dans le cadre de ce TP, une analyse du trafic réseau a été réalisée pour étudier les interactions entre le client, le serveur DNS et le serveur web. L'outil de capture a enregistré plusieurs types de paquets, notamment des requêtes et réponses DNS, ainsi que des messages ICMP signalant des destinations inatteignables.

Les requêtes DNS observées sont des requêtes standard où le client demande la résolution d'un nom de domaine, identifiables par des paquets marqués "Standard query" accompagnés d'un identifiant unique. En réponse, le serveur DNS envoie des paquets marqués "Standard query response", attestant de la résolution réussie du nom de domaine.

Par ailleurs, des paquets ICMP portant le message "Destination unreachable" ont été détectés, indiquant des tentatives de communication échouées. Ces échecs peuvent être attribués à une configuration réseau incorrecte, à un service cible hors ligne, ou à l'absence de réponse du service visé.

2.2 Configuration du service web :

2.2.1 Création du site web réel/légitime :

```
[0 root@ServeursWeb ~]$ mkdir /var/www/RealWebsite.com
[0 root@ServeursWeb ~]$
```

Avec votre éditeur de texte favori, créez un nouveau fichier html nommé « index.html » dans le répertoire du site web (la page d'accueil du site web) :


```
GNU nano 2.2.6 File: index.html
<!DOCTYPE html>
<html>
<head>
  <meta charset="0"0"0utf-8"0"0">
  <title>Real Webpage</title>
</head>
<body>
  <center>
    <h1>Real Webpage</h1>
    <table border="1">
      <tr><td>
        <table>
          <tr>
            <td>name:</td>
            <td><input name="username"/></td>
          </tr>
          <tr>
            <td>password:</td>
            <td><input name="password" type="password"/></td>
          </tr>
          <tr>
            <td/>
            <td align="center"><input type="button"
value="login"/></td>
          </tr>
        </table>
      </td></tr>
    </table>
  </center>
</body>
</html>
```

Créons le hôte virtuel correspondant (avec le bonne adresse IP : 10.0.2.200) dans le répertoire dédié aux sites web d'Apache.

```
GNU nano 2.2.6 File: /etc/apache2/sites-available/RealWebSite
<VirtualHost 10.0.2.200:80>
  DocumentRoot /var/www/RealWebSite.com
  ServerName www.RealWebSite.com
  ServerAlias RealWebSite
  <Directory /var/www/RealWebSite.com/>
    Options FollowSymLinks
    AllowOverride None
  </Directory>
</VirtualHost>
```

Une fois la configuration faite pour le site, il reste à l'activer pour qu'Apache le prenne en compte. Pour faire utiliser l'outil a2ensite :

```
[0 root@ServeursWeb ~]$ a2ensite RealWebSite
Site RealWebSite already enabled
```

Puis on redémarre le service apache:

```
[0 root@ServeursWeb ~]$ sudo service apache2 restart
```

2.2.2 Création du faux site web (mis en place par l'attaquant) :

On procèdera de la même manière pour créer le faux site web que l'attaquant a mis en place pour son

attaque.

```
GNU nano 2.2.6
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Fake Webpage</title>
</head>
<body>
  <center>
    <h1>Fake Webpage</h1>
    <table border="1">
      <tr><td>
        <table>
          <tr>
            <td>name:</td>
            <td><input name="username"/></td>
          </tr>
          <tr>
            <td>password:</td>
            <td><input name="password" type="password"/></td>
          </tr>
          <tr>
            <td/>
            <td align="center"><input type="button"
value="login"/></td>
          </tr>
        </table>
      </td></tr>
    </table>
  </center>
</body>
</html>
```

Créons le hôte virtuel correspondant (avec le bonne adresse IP : 10.0.2.202) dans le répertoire dédié aux sites web d'Apache.

```
Marionnet - /home... H1 G1 Téléchargements -... TP- Sécurisation D... ServeursWeb (deb... Hacker (debian-wh...
ServeursWeb (debian-wheezy-08367)
GNU nano 2.2.6 File: /etc/apache2/sites-available/FakeWebSite
<VirtualHost 10.0.2.202:80>
  DocumentRoot /var/www/FakeWebSite.com
  ServerName www.FakeWebSite.com
  <Directory /var/www/FakeWebSite.com/>
    Options FollowSymLinks
    AllowOverride None
  </Directory>
</VirtualHost>
```

```
[0 root@m1 ~]$ curl http://10.0.2.202
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Fake Webpage</title>
</head>
<body>
  <center>
    <h1>Fake Webpage</h1>
    <table border="1">
      <tr><td>
        <table>
          <tr>
            <td>name:</td>
            <td><input name="username"/></td>
          </tr>
          <tr>
            <td>password:</td>
            <td><input name="password" type="password" value="login"/></td>
          </tr>
          <tr>
            <td/>
            <td align="center"><input type="button" value="login"/></td>
          </tr>
        </table>
      </td></tr>
    </table>
  </center>
</body>
</html>
[0 root@m1 ~]$
```

2.3 Test fonctionnement de bout en bout

```
</html>
[0 root@m1 ~]$ curl http://www.iut-villetaneuse.fr
```

on voit apparaître la page réelle correspondant à la page "index" du RealSite

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Real Webpage</title>
</head>
<body>
  <center>
    <h1>Real Webpage</h1>
    <table border="1">
      <tr><td>
        <table>
          <tr>
            <td>name:</td>
            <td><input name="username"/></td>
          </tr>
          <tr>
            <td>password:</td>
            <td><input name="password" type="password"
"/></td>
          </tr>
          <tr>
            <td/>
            <td align="center"><input type="button"
value="login"/></td>
          </tr>
        </table>
      </td></tr>
    </table>
  </center>
</body>
</html>

```

3- Mise en place de l'attaque DNS spoofing (poisoning attack)

3.1 Mise en place de scapy

Simulation d'Attaque DNS Spoofing Préparation Scapy a été installé depuis une archive pour contourner les limitations de performance des dépôts Debian.

Une modification a été apportée au code de Scapy pour gérer une particularité des tables de routage de Marionnet

Exécution

Un script sniffing_attack.py a été écrit et exécuté pour intercepter les requêtes DNS et injecter de fausses réponses. Explication du script :

from scapy.all import *: Importe toutes les fonctions de la bibliothèque Scapy.

Les variables Authoritative_DNS_IP, Recursive_DNS_IP, et Malicious_IP stockent les adresses IP respectives.

DNS_Responder est une fonction fermée qui crée une autre fonction, getResponse, qui sera appelée pour chaque paquet capturé qui correspond aux critères définis.

getResponse analyse les paquets pour trouver ceux qui sont des requêtes DNS sans réponse (ancount == 0) destinées au serveur autoritaire depuis le serveur de résolution.

Si la requête concerne le domaine ciblé ("iut-villetaneuse.fr"), le script crée une réponse DNS spoofée pointant vers l'adresse IP malicieuse définie par Malicious_IP.

La réponse falsifiée est envoyée en utilisant la fonction send de Scapy.

La fonction sniff est appelée avec le paramètre prn, qui spécifie une fonction de rappel (DNS_Responder) pour chaque paquet capturé.

Commandes de Configuration et de Lancement

```
[0 root@Hacker ~/scapy-2.0.0.10]$ python sniffing_attack.py
WARNING: No route found for IPv6 destination :: (no default route?)
Don't care
Don't care
Don't care
Don't care
Don't care
Don't care
+
Sent 1 packets.
Spoofed DNS Response Sentiut-villetaneuse.fr.
Don't care
```