

COMMENT RENDRE LES « SCANNERS WEB » FOUS

DE QUEL GENRE SONT VOS ASSAILANTS ?



LA DURE VÉRITÉ

- Peu de choses risquent d'arrêter une attaque ciblée
- Les hack-opportunistes...
 - vont au plus facile
 - n'ont que faire de qui vous êtes
 - utilisent des outils
- Tous utilisent des outils

COMMENT SE PROTÉGER ?

1. Corriger les vulnérabilités
2. Détecter et bloquer les attaques
3. “Décourager” les attaquants

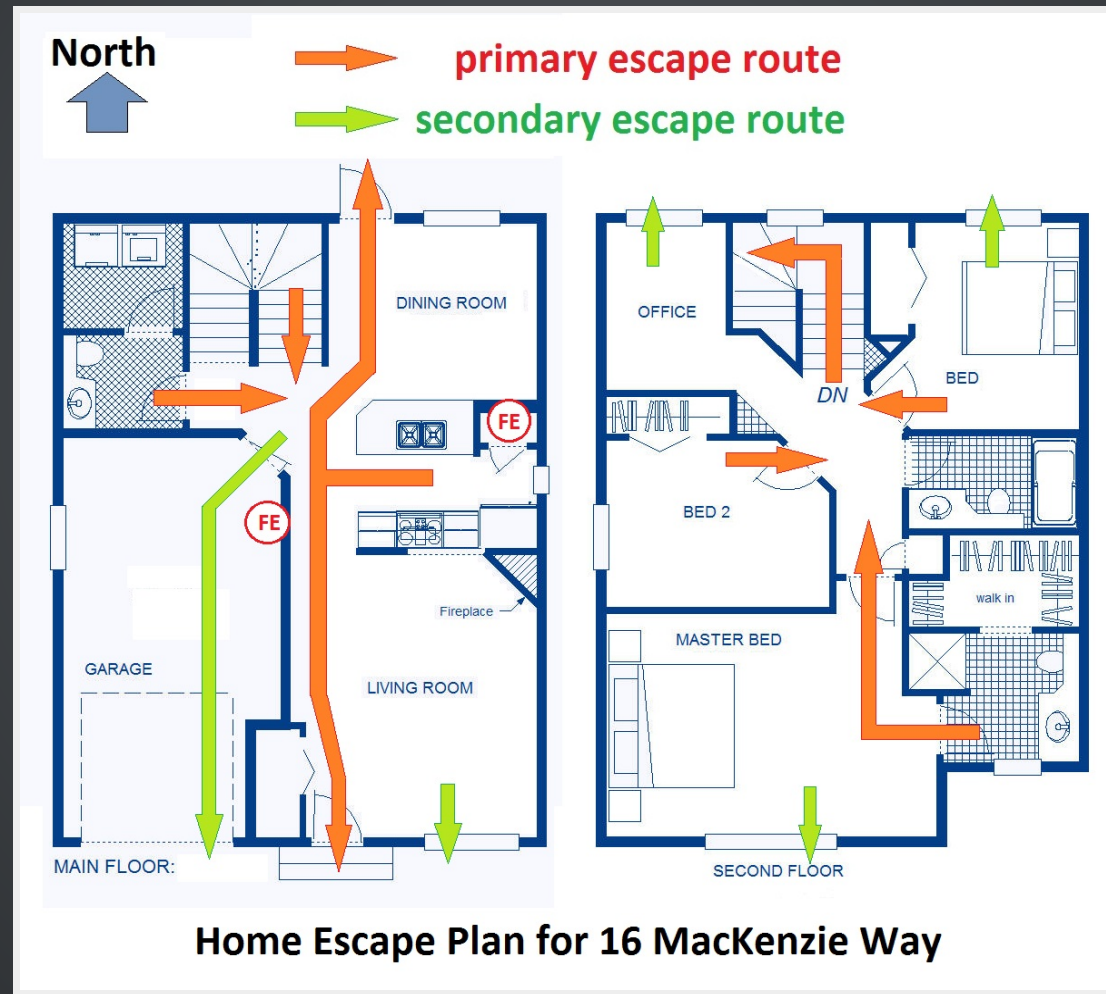
**SOLUTION :
RENDRE LES OUTILS MOINS EFFICACES**

- En les ralentissant,
- En les rendant plus facilement détectables,
- En les rendant plus spécifiques

THÈMES ABORDÉS

- Suivez le guide !
- C'est moi qui décide !
- Un combat inégal !
- Entrez, faites comme chez vous !

SUIVEZ LE GUIDE !



PRINCIPE 1

LES SITES WEB SONT PRÉVISIBLES

LA CONVIVIABILITÉ D'ABORD !

- Structure de répertoire

```
/_layouts/AdminRecycleBin.as  
/_layouts/bpcf.aspx  
/_layouts/create.aspx  
/_layouts/listfeed.aspx  
/_layouts/managefeatures.as
```

LA CONVIVIABILITÉ D'ABORD !

- Chaînes de requêtes

```
/eBayISAPI.dll?ViewFeedback2&userid=halu_games&ftab=AllFeedback&rt=nc&mywo
```

LA CONVIVIABILITÉ D'ABORD !

- Formulaires

```
<td>
<input id="email" name="email" class="inputtext" type="text" tabindex="1" v
</td>
<td>
<input id="pass" name="pass" class="inputtext" type="password" tabindex="2"
</td>
```

RENDRE LES RESSOURCES « ANONYMES »

La façon de nommer les ressources n'a aucun impact sur les utilisateurs !

Des identifiants génériques peuvent être utilisés pour :

- Les noms de répertoire
- Les paramètres de chaîne de requêtes
- Les paramètres de formulaire
- Les identifiants d'éléments HTML
- Etc.

EXEMPLES IDENTIFIANTS GÉNÉRIQUES

```
/_layouts/bpcf.aspx  
/_layouts/create.aspx
```



```
/726fca03/a6ea4cc7.aspx  
/726fca03/859d26fb.aspx
```


EXEMPLES IDENTIFIANTS GÉNÉRIQUES

userid=halu_games



9ca69116=halu_games

EXEMPLES IDENTIFIANTS GÉNÉRIQUES

```
<td>  
value="" name="email">  
</td>  
<td>  
name="pass">  
</td>
```



```
<td>  
value="" name="6852e570">  
</td>  
<td>  
name="c3ac1511">  
</td>
```

ALLONS PLUS LOIN !

Requête 1 :

```
username=' OR User LIKE 'a%25'&password=bidon
```

Requête 2 :

```
username=' OR User LIKE 'b%25'&password=bidon
```

CHAMPS DE FORMULAIRE DYNAMIQUES

Requête 1:

```
5134008b=user&a5ee2f12=password
```

Requête 2:

```
1a59a2ac=user&d54e31c6=password
```

CSRF TOKEN VS ID DYNAMIQUES?

Oblige les outils à être encore plus spécifiques.

```
var cpt=0;
for(var i in document.forms[0].children)
{
    if(cpt == 2 &&
        document.forms[0].children[i].type == 'input' &&
        document.forms[0].children[i].getAttribute("type") == "text")
    {
        document.forms[0].children[i].value = "FUZZVALUE";
    }else if(document.forms[0].children[i].type == 'input' &&
        document.forms[0].children[i].getAttribute("type") == "text")
    {
        cpt++;
    }
}
```

RENDRE LES RESSOURCES « ANONYMES »

- Empêche l'énumération des ressources
- Peut être fait à la compilation
- Très utile dans le cas de gestionnaires de contenu

C'EST MOI QUI DÉCIDE !



PRINCIPE 2

**LE PROTOCOLE HTTP OFFRE UNE
GRANDE SURFACE D'ATTAQUE**

LA RFC 2616

```
Method      = "OPTIONS"           ; Section 9.2
              | "GET"              ; Section 9.3
              | "HEAD"             ; Section 9.4
              | "POST"            ; Section 9.5
              | "PUT"             ; Section 9.6
              | "DELETE"          ; Section 9.7
              | "TRACE"           ; Section 9.8
              | "CONNECT"         ; Section 9.9
              | extension-method
extension-method = token
```

LA RFC 2616

```
Status-Code      =
    "100" ; Section 10.1.1: Continue
    | "101" ; Section 10.1.2: Switching Protocols
    | "200" ; Section 10.2.1: OK
    | "201" ; Section 10.2.2: Created
    | "202" ; Section 10.2.3: Accepted
    | "203" ; Section 10.2.4: Non-Authoritative Information
    | "204" ; Section 10.2.5: No Content
    | "205" ; Section 10.2.6: Reset Content
    | "206" ; Section 10.2.7: Partial Content
    | "300" ; Section 10.3.1: Multiple Choices
    | "301" ; Section 10.3.2: Moved Permanently
    | "302" ; Section 10.3.3: Found
    | "303" ; Section 10.3.4: See Other
    | "304" ; Section 10.3.5: Not Modified
    | "305" ; Section 10.3.6: Use Proxy
    | "307" ; Section 10.3.8: Temporary Redirect
    | "400" ; Section 10.4.1: Bad Request
    | "401" ; Section 10.4.2: Unauthorized
    | "402" ; Section 10.4.3: Payment Required
    | "403" ; Section 10.4.4: Forbidden
    | "404" ; Section 10.4.5: Not Found
    | "405" ; Section 10.4.6: Method Not Allowed
    | "406" ; Section 10.4.7: Not Acceptable
```

LA RFC 2616

```
request-header = Accept           ; Section 14.1
                | Accept-Charset  ; Section 14.2
                | Accept-Encoding ; Section 14.3
                | Accept-Language ; Section 14.4
                | Authorization   ; Section 14.8
                | Expect          ; Section 14.20
                | From            ; Section 14.22
                | Host            ; Section 14.23
                | If-Match        ; Section 14.24
```

Fielding, et al.

Standards Track

[Page 38]

RFC 2616

HTTP/1.1

June 1999

```
| If-Modified-Since ; Section 14.25
| If-None-Match     ; Section 14.26
| If-Range          ; Section 14.27
| If-Unmodified-Since ; Section 14.28
| Max-Forwards       ; Section 14.31
| Proxy-Authorization ; Section 14.34
| Range              ; Section 14.35
| Referer            ; Section 14.36
| TE                 ; Section 14.39
| User-Agent         ; Section 14.43
```

LA RFC 2616

```
foo://example.com:8042/over/there?name=ferret#nose
  \  /      \  /      \  /      \  /      \  /
  |          |          |          |          |
scheme      authority  path      query    fragment
  |          |          |          |          |
  / \      / \      / \      / \      / \
urn:example:animal:ferret:nose
```


RESTREINDRE LE PROTOCOLE HTTP

- Utiliser seulement ce dont on a besoin
- Ce fait déjà partiellement

LIMITER LES MÉTHODES HTTP

- Recommandation vieille comme le monde
- 99 % du temps deux méthodes suffisent : GET et POST

LIMITER LES CODES DE STATUT

Requête :

```
GET /878926y17t123.html HTTP/1.1
```

Réponse :

```
HTTP/1.1 200 OK
[...]
<div>
    La ressource que vous avez demandé est inexistante
</div>
[...]
```

LIMITER LES CODES DE STATUT

Requête :

```
GET /private/docs/ HTTP/1.1
```

Réponse :

```
HTTP/1.1 200 OK
[...]
<a href="/private/docs/EF53FA82/">EF53FA82</a>
<a href="/private/docs/EF53A82/">EF53A82</a>
<a href="/private/docs/EF3F82/">EF3F82</a>
<a href="/private/docs/EF2/">EF2</a>
<a href="/private/docs/3/">3</a>
[...]
```

LIMITER LES ENTÊTES HTTP

```
Content-Type : application/www-form-urlencoded  
Expect : 100-continue  
From : user@example.com  
If-Match : "737060cd8c284d8af7ad3082f209582d"  
Max-Forwards: 10  
User-agent : Mozilla/5.0 (Windows NT 6.2; WOW64; rv:24.0) Gecko/20100101 Fire
```

SIMPLIFIER LA SYNTAXE DES URL

`example.com:8080/messages/usermsg.php?type=1`

`example.com/8080/messages/usermsg.php/type/1`

`example.com/8080/messages/usermsg/1`

`example.com/e87598de`

RESTREINDRE LE PROTOCOLE HTTP

- Empêche d'exécuter des actions non désirées
- Limite les informations retournées aux outils
- Rend la différenciation des ressources plus difficile

UN COMBAT INÉGAL



PRINCIPE 3 :

IL EST FACILE DE DIFFÉRENCIER DEUX RÉPONSES HTTP

EXAMPLE SQL INJECTION

Invalid name or password
Please enter your name and password

name:

password:

EXAMPLE SQL INJECTION

Access Granted

Welcome: **jake**

You are an authorised user. [Log out](#)

TEMPS DE COMPARAISON

Cas d'une réponse HTTP de 58 Ko

- Chercher une chaîne de caractères \Rightarrow 1 milliseconde
- Calculer l'empreinte MD5 \Rightarrow 1 milliseconde
- Chercher une expression régulière \Rightarrow 2 millisecondes

COMPLEXIFIER LA COMPARAISON

1. Rendre des pages Web identiques "très différentes"
2. Rendre des pages Web différentes "très similaires"

TRANSFORMER LE CODE HTML

```
<b>Hello</b>
```

TRANSFORMER LE CODE HTML

```
<div class="myBoldStyleClass">Hello</div>
```

TRANSFORMER LE CODE HTML

```
<form name="form1">  
<input type="text" name="textbox1" />  
</form>  
  
...  
<form name="form2">  
<input type="text" name="textbox1" />  
</form>
```

TRANSFORMER LE CODE HTML

```
<form name="form1_2">  
<input type="text" name="form1_textbox1" />  
...  
<input type="text" name="form2_textbox1" />  
</form>
```

LE JAVASCRIPT NOTRE ALLIÉ

```
<div>  
    <b>Hello</b> World  
</div>
```

LE JAVASCRIPT NOTRE ALLIÉ

```
<script>
    document.write('<div>');
    document.write('<b>');
    document.write('Hello');
    document.write('</b>');
    document.write('World');
    document.write('</div>');
</script>
```

TRANSFORMER LE CODE JAVASCRIPT

```
var _0xaabd=["\x3C\x64\x69\x76\x3E","\x77\x72\x69\x74\x65","\x3C\x62\x3E","\x48\x65\x6  
document[_0xaabd[1]](_0xaabd[0]);  
document[_0xaabd[1]](_0xaabd[2]);  
document[_0xaabd[1]](_0xaabd[3]);  
document[_0xaabd[1]](_0xaabd[4]);  
document[_0xaabd[1]](_0xaabd[5]);  
document[_0xaabd[1]](_0xaabd[6]);
```


TRANSFORMER LE CODE JAVASCRIPT

```
eval(function(p,a,c,k,e,d){
    e=function(c){return c};
    if(!''.replace(/^/,String))
    {while(c--){d[c]=k[c]||c}k=[function(e){return d[e]}};e=function(){return '\\w
    while(c--){if(k[c]){p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c])}}
    return p
}('1.0(\\'<3>\\');1.0(\\'<2>\\');1.0(\\'4\\');1.0(\\'</2>\\');1.0(\\'5\\');1.0(\\'</3>\\');','6,6,')
```

TRANSFORMER LE CODE JAVASCRIPT

```
$=~[];  
$={____:++$, $$$$: (![]+"") [$], __$:++$, $__$: (![]+"") [$], _$:++$, $__$: ({ }+"") [$], $$__: ($ [$  
$. $ _= ($. $ _=$+"") [$.$ _$]+ ($. $ _=$.$ _[$.___$]) + ($. $$= ($. $+"") [$.___$]) + ((!$)+"") [$.___$$]+ ($.  
$. $$=$.$+ (! ""+"") [$.___$$]+ $. __+ $. _+ $. $+ $. $$;  
$. $= ($. __) [$.$ _] [$.$ _];  
$. $ ($.$ ($.$ ($.$+"\""+ "\""+ "\"\\\""+ $. __$+ $. __$+ $. $$ _$+ $. _$+ $. $$ __+ $. _+ "\"\\\""+ $. __$+ $. $ _$+ $. $ _$+ $. $ _$+ $. $$
```

OFFUSQUER NE SUFFIT PAS

Un offuscateur de code retourne toujours le même résultat.

More complex metamorphic viruses and permutation techniques

The Win32/Evol virus appeared in early July, 2000. The virus implements a metamorphic engine. Evol is capable to run on any major Win32 platform. Figure 6 shows an example code fragment as mutated to a new form in a new generation of the same virus.

a. An early generation:

```
C7060F000055  mov     dword ptr [esi],5500000Fh
C746048BEC5151  mov     dword ptr [esi+0004],5151EC8Bh
```

b. And one of its later generations:

```
BF0F000055     mov     edi,5500000Fh
893E           mov     [esi],edi
5F             pop     edi
52             push    edx
B640           mov     dh,40
BA8BEC5151     mov     edx,5151EC8Bh
53             push    ebx
8BDA           mov     ebx,edx
895E04         mov     [esi+0004],ebx
```

c. And yet another generation with recalculated ("encrypted") "constant" data.

```
BB0F000055     mov     ebx,5500000Fh
891E           mov     [esi],ebx
5B             pop     ebx
51             push    ecx
B9CB00C05F     mov     ecx,5FC000CBh
81C1C0EB91F1   add     ecx,F191EBC0h ; ecx=5151EC8Bh
894E04         mov     [esi+0004],ecx
```

Figure 6: Example of code metamorphosis of Win32/Evol

Tiré de Hunting For Metamorphic de Symantec


MODIFIER LE CODE DYNAMIQUEMENT


MODIFIER LE CODE DYNAMIQUEMENT

```
_ $$=~[];  
_ $$={0D:++_ $$,$$:(![ ]+"")[_ $$],$_ :++_ $$,00:(![ ]+"")[_ $$],$_ $:++_ $$,$$$$_:({}+"")[_ $$],  
_ $$.$_ $=( _ $$.$_ $=_ $$+"")[_ $$.$_ ]+( _ $$.$$_=_ $$.$_ $[_ $$.$_ ])+( _ $$.$_=( _ $$.$$_+"")[_ $  
_ $$.$_=$$_.$$_+(!" "+"" )[_ $$.$_ $$]+_ $$.$$_$_+_ $$.$_.$$_+_ $$.$$_+_ $$.$$_+_ $$.$$_+_ $$.$$_+_ $$.0;  
_ $$.$$_=( _ $$.$_0D)[ _ $$.$_ $_][ _ $$.$_ $_];  
_ $$.$$_( _ $$.$$_( _ $$.$_+""\""+ _ $$.$_+_ $$.$$_+_ $$.$$_$_+_ $$.$$_+""\""+ _ $$.$_+_ $$.$$_+_ $$.$$_+_ $$.0
```

MODIFIER LE CODE DYNAMIQUEMENT

Country Computation service

USERNAME 

PASSWORD 

LOGIN

MUTER LE CODE ?

- Difficile à bien faire
- Certains pièges à éviter
 - Les permutations
 - Les répétitions
 - Etc.

PAS BESOIN D'ÊTRE PARFAIT

COMPLEXIFIER LA COMPARAISON

- Permet :
 - De ralentir les outils en les forçant à interpréter le contenu
 - De rendre les outils « efficace » plus spécifiques
- À termes :
 - La comparaison sera toujours possible à partir du contenu interprété

```
document.documentElement.innerHTML
```

**ENTREZ, FAITES COMME
CHEZ VOUS !**

PRINCIPE 4

LE PROTOCOLE HTTP EST STATELESS

AUCUN ORDRE

```
GET /documents/FinancialReport2010.pdf HTTP/1.1  
POST /login.php HTTP/1.1  
POST /auth/message.php HTTP/1.1  
GET /file.php?path=/root/tmp HTTP/1.1
```

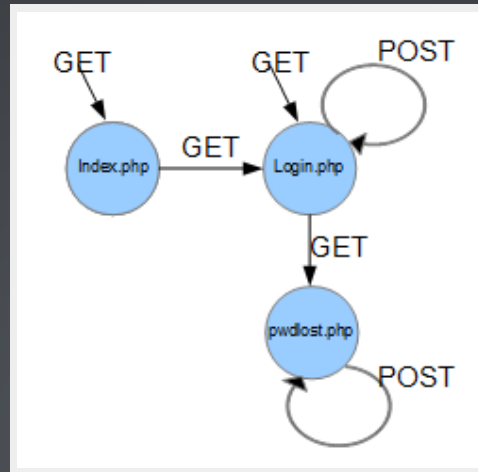
AUCUN ORDRE

```
GET /login.php HTTP/1.1  
POST /login.php HTTP/1.1  
GET /auth/index.php HTTP/1.1  
GET /auth/documentslist.php HTTP/1.1  
GET /documents/FinancialReport2010.pdf HTTP/1.1
```

COMPLEXIFIER L'ACCÈS AUX RESSOURCES

1. Forcer les outils à adopter le comportement prévu lors du développement du site Web

UNE SOLUTION INFALLIBLE



UNE SOLUTION INFALLIBLE

Inconvénients :

- Le développement et la mise en place sont coûteux
- La maintenance et l'évolution des sites Web deviennent très complexes

UNE AUTRE APPROCHE

Il est toujours facile de connaître la dernière ressource accédée.

LE REFERER

Requête 1 :

```
GET /index.php HTTP/1.1  
[...]
```

Serveur :

```
Session["lastPage"] = req.url.pathname
```

LE REFERER

Requête 2 :

```
GET /login.php HTTP/1.1  
[...]  
Referer: http://site.com/index.php
```

Serveur :

```
if(req.headers["Referer"] != Session["lastPage"])  
    throw Exception("HackAttempt");
```

LE JETON ANTI CSRF

Requête 1 :

```
GET /login.php HTTP/1.1  
[...]
```

Serveur :

```
Session["CSRFToken"] = GenerateCSRFToken();
```

LE JETON ANTI CSRF

Requête 2 :

```
POST /login.php?token=986213ABF082EF9862C HTTP/1.1  
[...]
```

Serveur :

```
if(req.query["token"] != Session["CSRFToken"])  
    throw Exception("HackAttempt");
```

LE VIEWSTATE

Réponse requête 1 :

```
<a href="contact.php">Contact</a>
<a href="help.php?topic=1">Contact</a>
<form action="login.php" method="POST">
[...]
```

Viewstate :

```
{ "GET" : ["contact.php", "help.php?topic=1"],
  "POST" : ["login.php"] }
```

LE VIEWSTATE

Requête 2 :

```
GET /help.php?topic=1 HTTP/1.1  
[...]
```

Serveur :

```
if(!req.viewstate[req.method].contains(req.pathname))  
    throw Exception("HackAttempt");
```


FORCER LA SÉQUENCE

- Avantages :
 - Oblige les outils à être plus intelligents
 - Facilite la reconnaissance d'attaques
 - Peut être jumelé à aux comparaison complexes

ON RÉCAPITULE !



OBJECTIF

Rendre les « Scanners Web » moins efficace.

C'est-à-dire :

- Les ralentir
- Les rendre plus facilement détectables
- Les rendre plus spécifiques

SE RÉSUME À DEUX CHOSES

1. Complexifier la génération et l'exécution de requêtes
2. Complexifier l'analyse des réponses

CONCRÈTEMENT

- Forcez les outils à interpréter le contenu des pages Web
- Forcez une séquence d'actions lorsque possible
- Compilez vos sites Web !
- Supportez le moins de fonctionnalités du protocole HTTP possible

AJAX ?

