Contournement des systèmes de filtrage HTTP en ligne

Paul Such

Sommaire

- Les systèmes de filtrage en ligne
- Pourquoi et quand les contourner
- Comment ?
- Quelques possibilités
- ..et d'autres non encore explorées en détail

Les systèmes de filtrage en ligne

Les systèmes de filtrage en ligne

- De nombreux équipements proposent de filtrer le contenu en transit : proxy, firewall, UTM, IPS/IDS,...
  - BluecXXX
  - FortinXX
  - JunipXX
  - McAf...
  - …
- Pour différents protocoles : HTTP(s), FTP, Mail, IM....

Exemples

[redacted] : filtrer les contenus Web

Firewall - [redacted]

Basés sur une architecture proxy de pointe, tous les équipements [redacted] se comportent comme un « médiateur invisible » entre les utilisateurs et les applications. Plus particulièrement, ces équipements fonctionnent comme serveur envers le client et comme client envers le serveur – de manière parfaitement fluide. Les équipements [redacted] forment ainsi un point de contrôle essentiel sur la base de règles permettant de filtrer les contenus Web (y compris les contenus SSL cryptés), de bloquer les spywares et autres programmes malveillants et de contrôler le trafic MI, Skype, P2P et de flux multimédias.

[redacted]

Exemples

complète incluant l'antivirus, la prévention des intrusions, l'anti-espion, le filtrage des contenus Internet et l'anti-spam

Protection permanente contre les menaces les plus récentes

Protection automatique contre les contenus indésirables

suivants :

- Pare-feu assurant le contrôle des personnes et des éléments qui accèdent au réseau
- Système de prévention des intrusions (pare-feu à action approfondie) permettant de bloquer les attaques au niveau de l'application
- Antivirus (avec Anti-Phishing, Anti-Spyware, Anti-Adware) permettant de bloquer les virus, les chevaux de Troie et d'autres logiciels malveillants
- Anti-Spam permettant de bloquer les spammeurs et les hameçonneurs connus
- Filtre Web permettant de contrôler l'accès vers les sites de téléchargement malveillants ou

Leur utilisation

- Empêcher les utilisateurs de télécharger du contenu non souhaité
- Avoir un deuxième niveau de protection (ou pas)
- Assurer une liberté contrôlée à l'utilisateur

Les contourner, quand ?

Quand ?

- Dans notre cas : tests d'intrusion internes
  - Depuis un poste standard-client
  - Depuis une machine contrôlée

Nombreux outils « utiles » détectés comme des « malwres »

  - Nmap, Cain, Cachedump,....

Les contourner

Quelques possibilités

Toujours une question d'outils à disposition

- De nombreuses possibilités
- Encryption
- Archives : trouver le bon format...
  - Jar
  - ARJ
  - CAB
  - 7Zip (standard ou légèrement modifié...)

Contourner : 7Zip

Fichier 7zip légèrement modifié

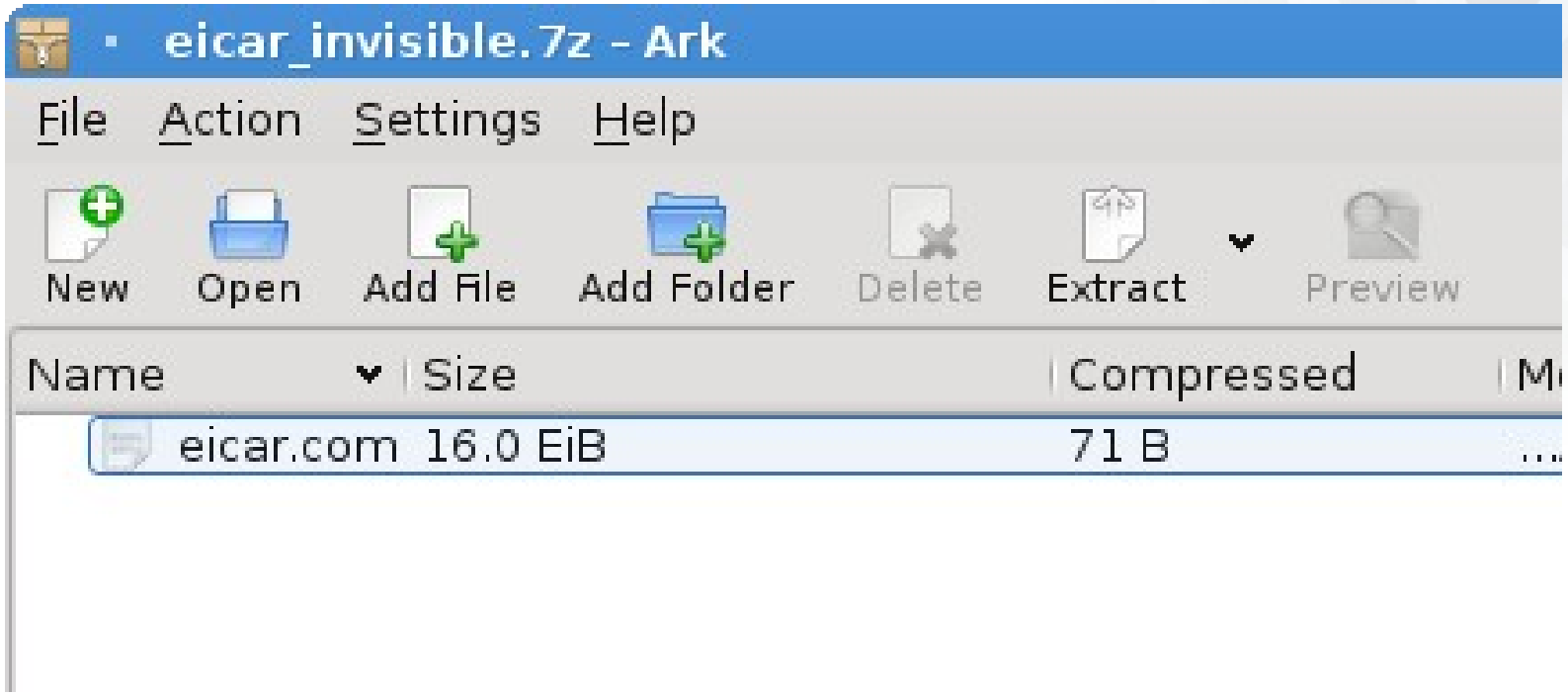
Modification de la structure du fichier 7Zip

```
REAL_UINT64 means real UINT64.

UINT64 means real UINT64 encoded with the following scheme:

  Size of encoding sequence depends from first byte:
  First_Byte  Extra_Bytes        Value
  (binary)
  0xxxxxxx                : ( xxxxxxx                )
  10xxxxxx    BYTE y[1]   : (  xxxxxx << (8 * 1)) + y
  110xxxxx    BYTE y[2]   : (   xxxxx << (8 * 2)) + y
  ...
  1111110x    BYTE y[6]   : (       x << (8 * 6)) + y
  11111110    BYTE y[7]   :                       y
  11111111    BYTE y[8]   :                       y
```

Contourner : 7Zip



Contourner : 7Zip

File **eicar_invisible.7z** received on **2010.05.28 08:51:37 (UTC)**
Current status: **finished**
Result: **2/41 (4.88%)**

Compact

Print results

Antivirus	Version	Last Update	Result
a-squared	4.5.0.50	2010.05.10	-
AhnLab-V3	2010.05.28.00	2010.05.28	-
AntiVir	8.2.1.242	2010.05.28	-
Antiy-AVL	2.0.3.7	2010.05.26	-
Authentium	5.2.0.5	2010.05.28	-
Avast	4.8.1351.0	2010.05.28	-
Avast5	5.0.332.0	2010.05.28	-
AVG	9.0.0.787	2010.05.27	-

Contourner : la quantité

Une archive dans une archive dans une...

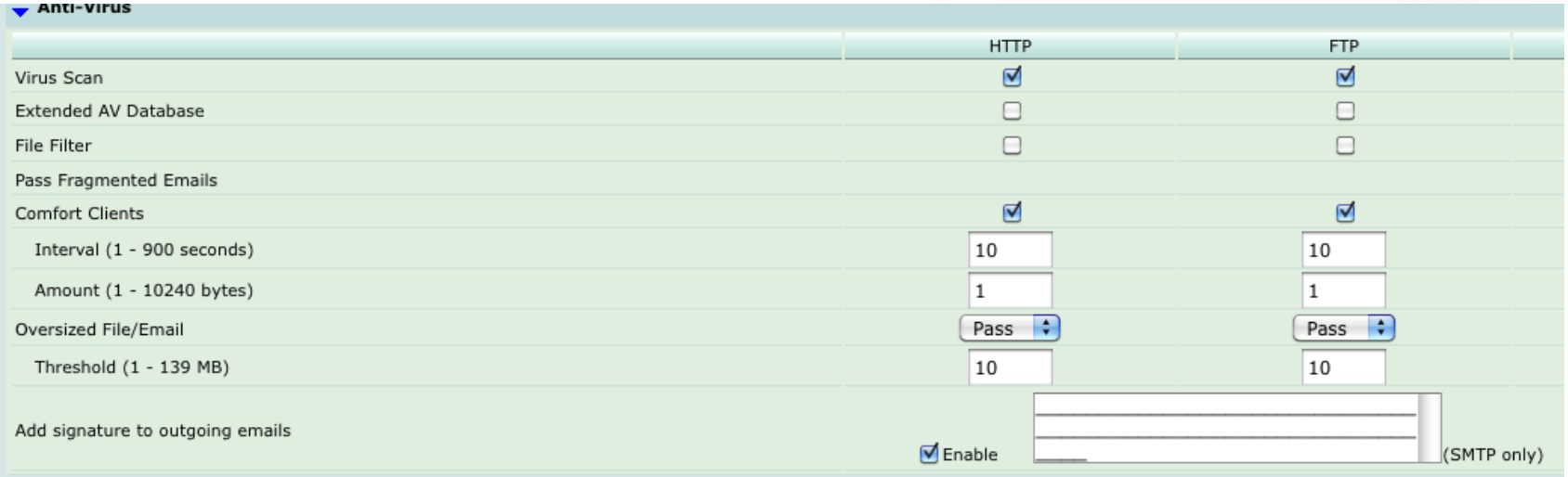
Jouer avec les limites

- Mode par défault... Pass
- Valeur maximale : 255
- Valeur « commune » : 16/31

Contourner : la taille...

La taille a une importance..

- Default : 10Mo
- Compressed / Uncompressed



Contourner : jouer avec la RFC

- Idée : les RFC sont très complètes et sont souvent partiellement implémentées
  - Ex : plusieurs centaines de pages pour la RFC 2616 (HTTP)
  -

Contourner : jouer avec la RFC

- Utiliser la RFC 2616
  - Content-Range header
- Récupérer des « chunks » (morceaux) de données

14.16 Content-Range

The Content-Range entity-header is sent with a partial entity-body to specify where in the full entity-body the partial body should be applied. Range units are defined in section 3.12.

```
Content-Range = "Content-Range" ":" content-range-spec

content-range-spec      = byte-content-range-spec
byte-content-range-spec = bytes-unit SP
                          byte-range-resp-spec "/"
                          ( instance-length | "*" )

byte-range-resp-spec = (first-byte-pos "-" last-byte-pos)
                                | "*"
instance-length           = 1*DIGIT
```

Contourner : jouer avec la RFC

```
# -*- coding: utf-8 -*-
import urllib2, sys
index=0
chunks=20
URL=sys.argv[1]
fName=URL[URL.rfind('/')+1:]
req=urllib2.Request(URL)
req.add_header("Range","bytes=0-0")
data=urllib2.urlopen(req)
if data.getcode()!=206:
  print "Server does not support chunking. Exiting..."
  sys.exit(0)
length = data.info().getheader("Content-Range")
length=int(length[length.rfind('/')+1:])
print "File length is "+str(length)+" bytes"
chunkSize = int(length/chunks)
output=open(fName,'wb')
while index < length:
  req=urllib2.Request(URL)
  req.add_header("Range","bytes="+str(index)+"-"+str(index+chunkSize))
  data=urllib2.urlopen(req)
  #print "bytes="+str(index)+"-"+str(index+chunkSize) + " - " + str((index*100)/length)+"%"
  print str((index*100)/length)+"%"
  output.write(data.read())
  index=index+chunkSize+1
output.close()
print 'done'
```

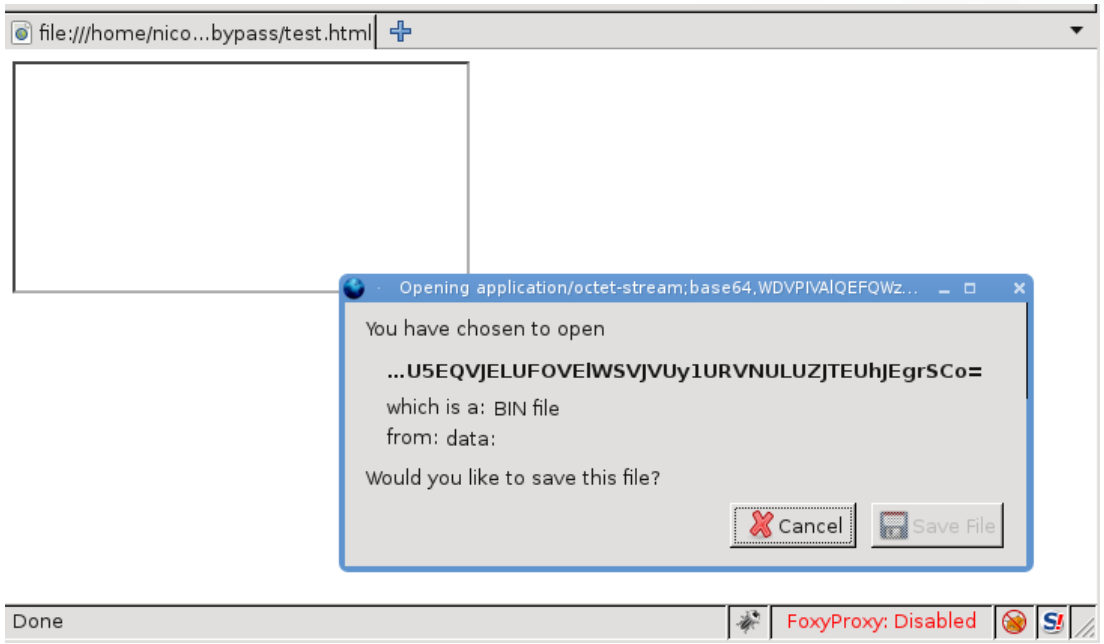
Credits : Nicolas Oberli - SCRT

Data URLs

- Le contenu d'une page HTML n'est pas vérifié
- HTML autorise l'inclusion de données directement dans le source de la page
  -
  - Utilisé pour inclure une image par exemple
- N'importe quel type de donnée peut être spécifié
  - "data:application/octet-stream;base64,data"

Data URLs - Suite

- Il est donc possible d'inclure un fichier binaire encodé en Base64
- Une fois inclus dans une iframe par exemple :



Inclusion dans un PDF

- Le format PDF permet d'inclure des fichiers dans un document PDF
  - Metasploit permet de le faire facilement
  - Utilisation du tag *EmbeddedFiles*
- De plus en plus d'analyseurs réseau recherchent maintenant à l'intérieur de ces fichiers

XMLHTTPrequest

- XMLHttpRequest est un objet ActiveX ou Javascript qui permet d'obtenir des données au format XML, JSON, mais aussi HTML, ou encore texte simple à l'aide de requêtes HTTP.
- On explique le succès récent de l'objet et la très grande utilisation qui en est faite actuellement (parfois au détriment de l'accessibilité d'un site) par la simple création du nom AJAX.

XMLHTTPrequest

- Les requêtes de type XMLHttpRequest permettent de récupérer un fichier distant
- Il est possible de spécifier des headers qui seront transmis
  - En utilisant la méthode de chunking, il est possible de passer outre un filtrage réseau

Ajax - Suite

- Exemple

```
<script>
  var req = new XMLHttpRequest();
  req.open("GET","http://");
  req.setRequestHeader("Range","bytes=0-20");
  req.onreadystatechange = function (aEvt) {
    if (req.readyState == 4) {
      if(req.status == 206)
        alert(req.responseText);
      else
        dump("Error loading page\n");
    }
  };
req.send(null);
</script>
```

- « Problème » : Sur le site contenant le fichier distant, il faudra mettre en place un fichier crossdomain.xml

Ajax - Suite

- Une fois les données récupérées, il n'est normalement pas possible de les faire télécharger au navigateur
- L'utilisation de data URLs permet de passer outre cette limitation
  - document.write('iframe src='data...

Application

- Application web
  - L'application télécharge un fichier sur Internet
  - L'utilisateur choisit une méthode de téléchargement
  - L'application prépare le téléchargement
    - Inclusion dans un PDF
    - Encodage en Base64
    - Archivage multiple
  - L'utilisateur télécharge le fichier
  - Le filtrage est contourné

D'autres possibilités...

- Java applet
- HTTP request truncating
- ...

Merci

Plus d'info : http://blog.scrt.ch