HW防守之日志分析 一

LemonSec 今天

转载自：https://www.freebuf.com/column/202350.html
笔者都是作为CTF解题思路来讲述的日志分析方式，其实在真实的网络攻击中，日志分析方式大同小异，这里引荐笔者的文章。

概念：
首先，咱们还是老规矩，先介绍一下什么是日志分析。
日志分析—-计算机、网络和其他IT系统生成审计跟踪记录或记录系统活动的日志。日志分析是对这些记录的评估，帮助公司缓解各种风险并满足合规性法规。
在当下的CTF大赛中，多以流量分析的形式出现，但是在个别比赛中依然会出现一题关于日志分析类的题目，一般的题目都是会让我们通过日志找线索，不会将flag写在日志，因此我们需要通过分析日志来判断，flag可能存在的位置，再通过类似的方式获取flag。
日志分析主要分成两种：
●Web日志分析
●系统日志分析
本期主要给大家带来Web日志分析。

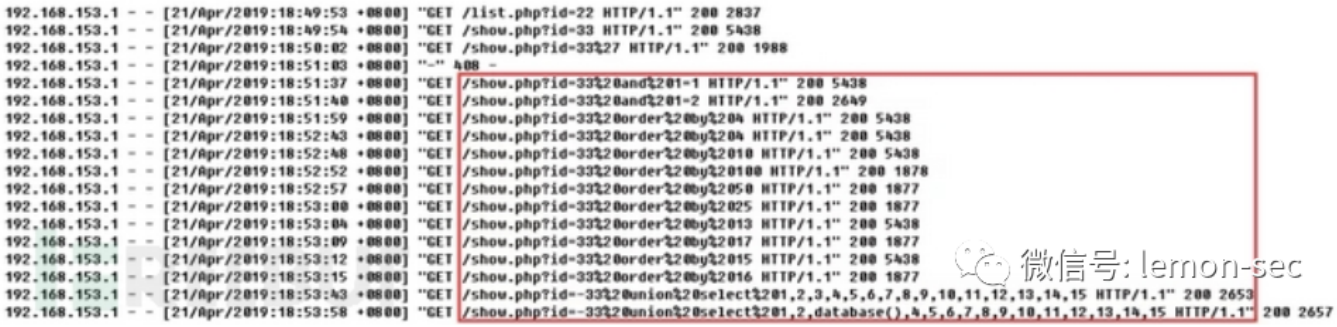
日志格式类型：
既然要进行分析日志，首先我们得先了解一下日志的格式到底有哪些？
目前在比赛中比较常见的WEB日志格式主要有两类：
●Apache的NCSA日志格式，NCSA格式分为：
NCSA普通日志格式（CLF）
NCSA扩展日志格式（ECLF）
●IIS的W3C日志格式
除了格式不同之外，一般的分析方法基本相似，因此接下来以NCSA普通日志格式进行演示。
为了可以更好的演示，这边使用的是NCSA普通日志格式，它的格式如下：

远程主机 IP	- (E-mail)	- (登录名)	请求时间	方法+资源+协议	状态代码	发送给客户端的字节数
192.168.153.1	-	-	[21/Apr/2019:18:48:29 +0800]	POST /admin/login.action HTTP/1.1	[illegible]	78

常用日志分析方法：
常见的日志分析方法有两种：
1.特征字符分析
2.访问频率分析
特征字符分析：

特征字符分析法：顾名思义，就是根据攻击者利用的漏洞特征，进行判断攻击者使用的是哪一种攻击。
常见的类型有以下：SQL注入、XSS跨站脚本攻击、恶意文件上传、一句话木马连接等。
SQL注入：
漏洞特征：存在SQL注入语句
常见的SQL注入语句有：
●通过报错注入、布尔盲注、时间盲注判断是否存在注入：
⊙字符型
■ 参数后加单引号，报错：sql1.php?name=admin’
■ 参数后加’ and ‘1’=’2和’ and ‘1’=’2，访问正常：sql1.php?name=admin’ and ‘1’=’1 /sql1.php?name=admin’ and ‘1’=’2
■ 参数后加’ and sleep(3) –，是否延迟3秒打开：sql1.php?name=admin’ and/or sleep(3)–
⊙数字型
■ 参数后加单引号，报错:sql2.php?id=1’
■ 参数后加and 1=1和and 1=2，访问正常：sql2.php?id=1 and 1=1/sql2.php?id=1 and 1=2
■ 参数后加and sleep(5)，是否延迟3秒打开：sql2.php?id=1 and sleep(5)
● 通过各种注入语句进行SQL注入攻击：
⊙联合查询注入
■ union select
■ order by
⊙报错注入(常见报错注入函数)
■ floor()
■ extractvalue()
■ updatexml()
■ geometrycollection()
■ multipoint()
■ polygon()
■ multipolygon()
■ linestring()
■ multilinestring()
■ exp()
⊙常见数据库类型判断
■ ACCESS
and (select count (*) from sysobjects)>0返回异常
and (select count (*) from msysobjects)>0返回异常
■ SQLSERVER
and (select count (*) from sysobjects)>0返回正常
and (select count (*) from msysobjects)>0返回异常
and left(version(),1)=5%23参数5也可能是4
■ MYSQL

id=2 and version()>0返回正常
id=2 and length(user())>0返回正常
id=2 CHAR(97, 110, 100, 32, 49, 61, 49)返回正常
■ Oracle
and length (select user from dual)>0返回正常
由于文章长度有限，只列举部分，一般出现有上述内容，则可判断此处可能存在SQL注入。

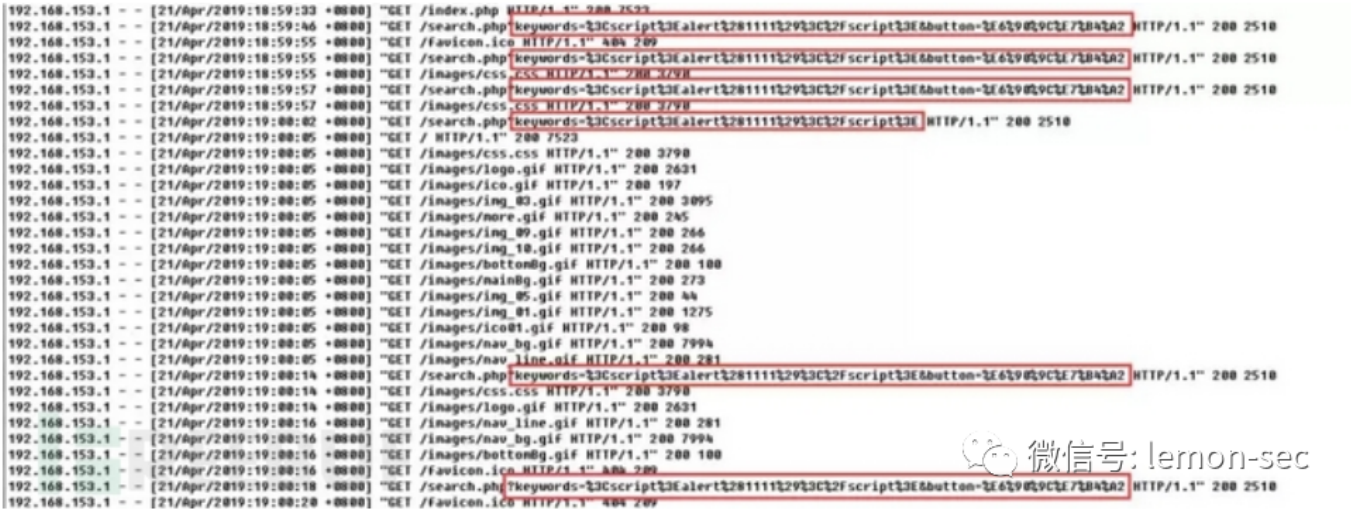


如图，可以很明显地发现红色框中有很明显的SQL注入语句，如布尔盲注、union select联合注入。

XSS跨站脚本攻击：

漏洞特征：明显的js恶意执行代码
常见的XSS跨站脚本攻击中存在的一些代码：
● 标签
■ <script>
■<body>
■<input>
■<img>
■<a>
■<svg>
■<BGSOUND>
■<LINK>
■<META>
■<TABLE>
■<DIV>
■<IFRAME>
■<FRAMESET>
■<STYLE>
■<OBJECT>
■ ……
● 常用触发事件
■ oninput
■ onload
■ oncut
■ onclick
■ onerror
■ onmouseover

■ onfocus
■ onblur
■ poster
■ onscroll
■……
● 常用恶意代码
■ prompt
■ confirm
■ alert
■ javascript
■ eval
■ expression
■ window.location
■……
斗哥只给出部分常见的js代码，有兴趣的同学，自行查阅资料，将其补全，因为比赛是瞬息万变的，偶尔也会出现一些比较偏门的也不一定。



如图，可以很明显地发现红色框中有很明显的js恶意执行代码，如<script>标签、alert语句，但是由于apache日志的特性，如果是通过Post请求，则无法准确判断出是否存在XSS跨站脚本攻击

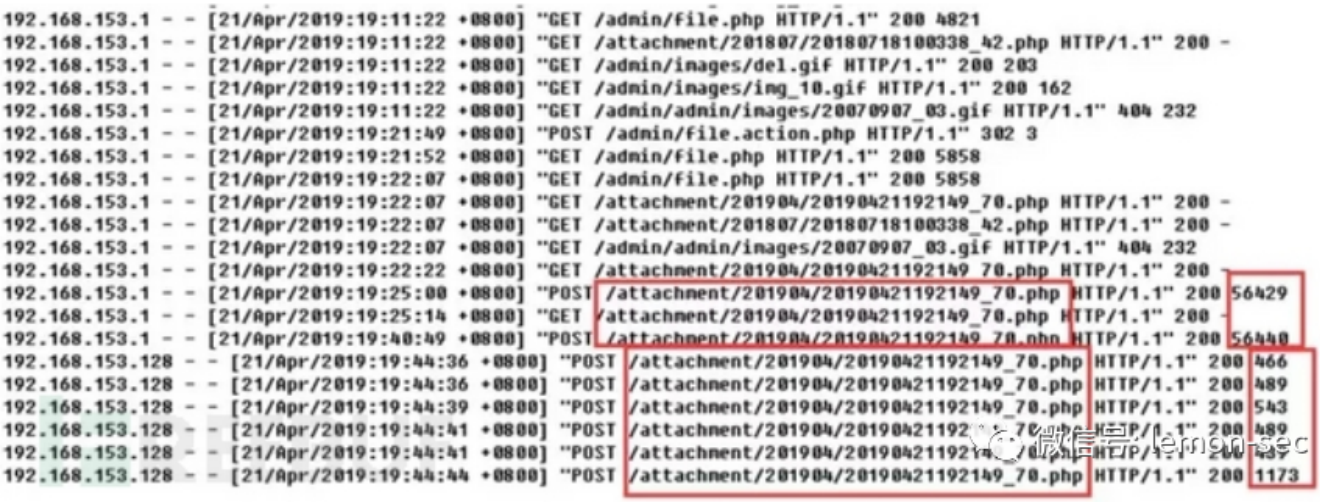
恶意文件上传：

通常存在于upload、file等出现类似字样的文件，均可能存在恶意文件上传，具体还需结合日志进行判断，一般是判断后续是否有出现Webshell等一些可以的web操作，可通过查看下图，发现在file.php页面的前后日志中，有存在一个带着日期的php页面，很可能就是利用file.php上传的文件，服务器自动生成名字，因此判断此处可能存在恶意文件上传。

一般地，如果Post请求的数据未被显示出来，则需要我们通过访问的链接以及上下文的访问详情确认此处是否存在恶意文件上传

一句话木马（Webshell）：

一般名字可疑的文件，如带日期字样的页面(.php、.asp、.aspx、.ash、.jsp等)、一串随机值的页面等，并且是通过Post请求，同时会返回一定的数据，此时可判断可能存在一句话木马、webshell等恶意文件，有些日志可能还有post请求参数，可结合参数，更准确地判断出是否存在一句话木马、webshell等恶意文件。



访问频率分析：

访问频率分析：不难理解，就是通过查看攻击者访问的频率来判断攻击者使用的是哪一种攻击。

常见的类型有有以下：SQL盲注、敏感目录爆破、账号爆破、Web扫描。

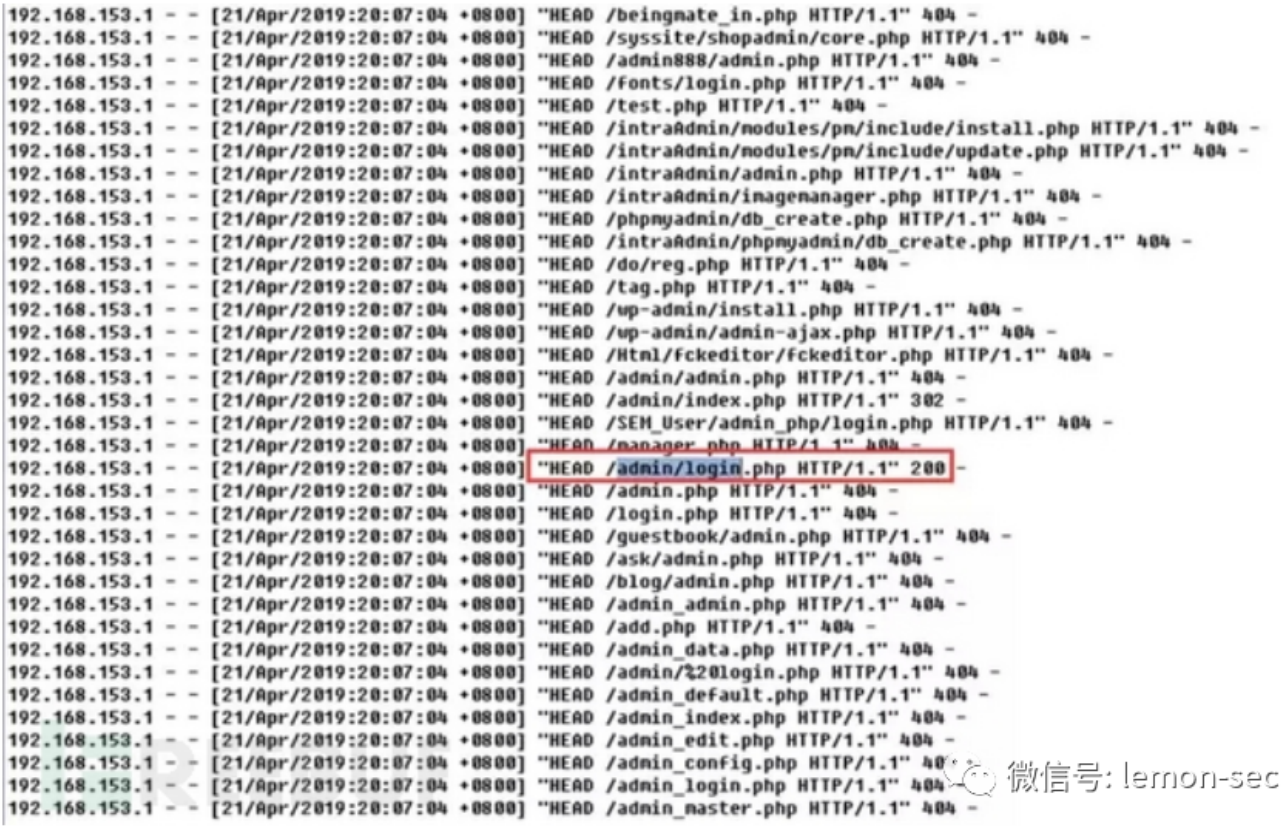
SQL盲注：

一般访问比较有规律，基本都包含SQL语句，并且大体都相似，有个别字符不同，具体情况可参考下图：

不难发现，上图中语句中除了包含SQL语句外，并且最后几个字符不同，其他都是一致的，因此可推测出此处为SQL盲注。

敏感目录爆破：

一般会有大量的探测目录，一般以Head方法为主进行探测，具体情况可参考下图：



发现在上图中，存在大量的HEAD请求方法，并且基本访问的页面返回的状态码为404。

账号爆破：

通常都是对一个登录页面进行大量post请求，并且返回值基本相似，具体情况可参考下图：

```
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:17 +0800] "POST /admin/login.action.php HTTP/1.1" 302 -
192.168.153.1 - - [21/Apr/2019:20:12:56 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:56 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:56 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:56 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:56 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:56 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:56 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:56 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
192.168.153.1 - - [21/Apr/2019:20:12:56 +0800] "POST /admin/login.action.php HTTP/1.1" 200 78
```

从上图可以很容易看出，明显是个登录页面，并且访问频率较高，都是通过post方法进行请求，访问值均一样，幸运的是此处出现了一个302状态码，说明攻击者已经爆破到账号密码了，故可判断此处应该为账号爆破。

Web扫描：

一般来说，访问的目标比较离散，并且来源地址相对固定，同时访问的结果大多数也都是失败的，并且在参数中有比较明显的扫描器特征字段

常见扫描器在url上的特征：

- AWVS 10.5或11

acunetix-wvs-test-for-some-inexistent-file
by_wvs
acunetix_wvs_security_test
acunetix
acunetix_wvs
acunetix_test
wvs_test

- Netsparker

netsparker
Netsparker
ns: netsparker

- Appscan

Appscan

- Webinspect

HP404
● Rsas
nsfocus
● Nessus
nessus
Nessus
● Sqlmap
sqlmap
此处借鉴了FREEBUFF上yiran4827大佬的文章常见扫描器或者自动化工具的特征（指纹）,具体内容可以点进去详看。
如下图：

根据之前讲到的条件，发现上图满足访问目标比较离散、并且源地址是固定，而且其中还带有wvstest字样。
总结:
现在CTF比赛中，单纯日志分析的题目会比较少，但也不可以说明它不重要,我们仍需要去学习它，去熟悉它。在本期中，大致给大家讲解了一些日志分析的方法，以及一些常见漏洞特征，这不仅可以用于CTF比赛中，也可以使用到日常的网站攻击溯源中的日志分析阶段。同时，也希望广大社区的小伙伴积极评论，因为斗哥的这份日志分析可能还不是很完整，还需要大家一起来完善它。最后，预告一下在本章之后还有一个章节，在下个章节中，会有系统的日志分析以及如何使用日志分析工具进行日志分析，敬请期待哟