

一次钓鱼到快速打穿

背景

21年中旬的一个对某互联网公司的红队项目，为期5天。客户要求最终能获取堡垒机权限，客户安全部门全部人员投入对网络进行防守。

前期信息收集

fofa、quake搜寻子域名、证书、高命中C段，导出信息寻找可以快速getshell漏洞，theharvester邮箱搜寻。最终在高命中C段资产中找到一个gitlab并开放注册功能，在公共仓库中找到了一个邮箱账户密码但是密码已经失效不过知道了客户工作人员使用的是第三方163企业邮，后在hackerone上找到了cve-2021-22205漏洞，该漏洞当时国内各大公众号并未发文利用，在21年10月中下旬国内各大安全公众号开始披露报告，打了一个信息时间差。



The screenshot shows a HackerOne report page for CVE-2021-22205. The page is in Chinese and displays a timeline of events. A red arrow points to the date '5月15日 (9个月前)' next to the event '这份报告已经披露。' (This report has been disclosed).

hackerone.com/reports/1154542

登录

解决方案 ∨ 产品 ∨ 伙伴 ∨

我的自托管 Ultimate 许可证已过期，我可以续订还是新的？ :)

干杯，威尔

时装 亚博体育app工作人员 发表评论。 5月12日 (9个月前)

嗨@vakzz,

我申请了新的许可证，它应该发送到您的@wearehackerone.com 电子邮件地址。如果你一周内没有收到，请告诉我。

谢谢， Dominic GitLab 安全团队

时装 亚博体育app工作人员 要求披露这份报告。 5月15日 (9个月前)

专业 同意披露这份报告。 5月15日 (9个月前)

这份报告已经披露。 5月15日 (9个月前)

项目过程

利用cve-2021-22205 ExifTool打进gitlab，打的过程中弹shell发现弹不出来（在这个时候其实已经

被设备捉住了），后来利用openssl加密弹的shell，使用linux的C2走CDN控住以求隐蔽控制并权限控死。


ab

Projects ▾ Groups ▾ More ▾

Search or jump to...

Write a comment or drag your files here...

Markdown is supported



500 Internal Privoxy Error

Privoxy encountered an error while processing your request:

Could not load template file `connection-timeout` or one of its included components.

Please contact your proxy administrator.

If you are the proxy administrator, please put the required file(s) in the `(confdir)/templates` directory. The location of the `(confdir)` directory is specified in the main Privoxy `config` file. (It's typically the Privoxy install directory, or `/etc/privoxy/`).

[Try again](#) or [attach a new file.](#)

```

bond0    Link encap:Ethernet  HWaddr 80:18:44:EE:26:2A
          inet (        ) Bcast:      Mask:255.255.255.192
          inet6 addr: fe80::8218:44ff:feee:262a/b4 Scope:Link
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:12268541595 errors:0 dropped:4 overruns:0 frame:4
          TX packets:51915881 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:808681201397 (753.1 GiB)  TX bytes:8784978298 (8.1 GiB)

bond1    Link encap:Ethernet  HWaddr 80:18:44:EE:26:28
          inet addr:      Bcast:10      Mask:255.255.255.0
          inet6 addr: fe80::8218:44ff:feee:2628/b4 Scope:Link
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:223349153645 errors:0 dropped:14 overruns:0 frame:64
          TX packets:1434298555 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:298704674265139 (271.6 TiB)  TX bytes:185430182736 (172.6 GiB)

eth0     Link encap:Ethernet  HWaddr 80:18:44:EE:26:28
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:187112515992 errors:0 dropped:0 overruns:0 frame:0
          TX packets:717080535 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:251292561430593 (228.5 TiB)  TX bytes:92696065366 (86.3 GiB)
          Interrupt:41

eth1     Link encap:Ethernet  HWaddr 80:18:44:EE:26:28
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:36236637653 errors:0 dropped:14 overruns:0 frame:64

```

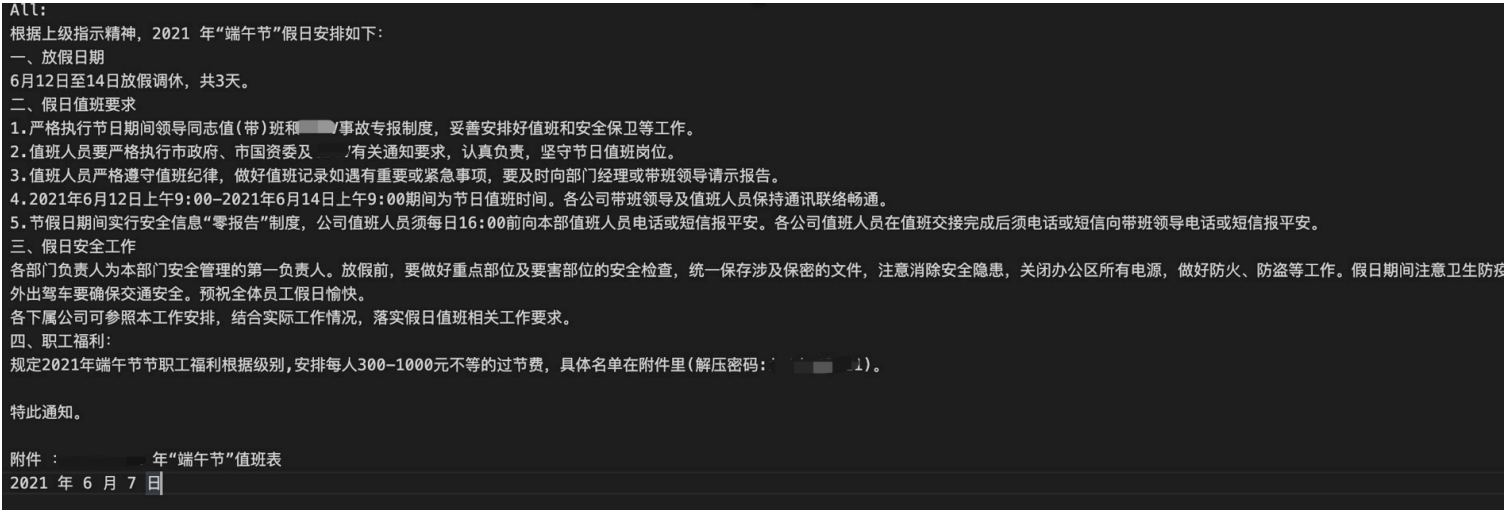
拿到权限后直接进入gitlab的data目录，该目录存放了全部代码。利用grep快速查找关键词在该目录搜寻到了一个代码中存放的邮件账户密码，该密码有效登录了163企业邮，但是在搜寻邮箱中邮件的时候发现gitlab已经被下线，C2中的session也直接断掉了，机器已断网。从控制gitlab服务器到下线中间时间不超过5分钟，至此权限丢失。足以可见防守方没有在进行摸鱼也在积极的与攻击队对抗。

外网除了这个gitlab以外无在发现能进入内网的点，没有进入内网的点后我们将目光转移到了手上的邮箱账户上，该邮箱为监控服务状态邮箱，会定时接收应用状态error信息以及服务器运行状态等，除了群发的通告邮件外无与工作人员单独来往邮件。但是该邮箱可以看到全部通讯录，我们导出了开发/运维人员的邮箱准备进行钓鱼。

时间临近端午节，就以端午节展开文案进行钓鱼，找写马的同事要了写好的马 免杀+捆绑+ico替

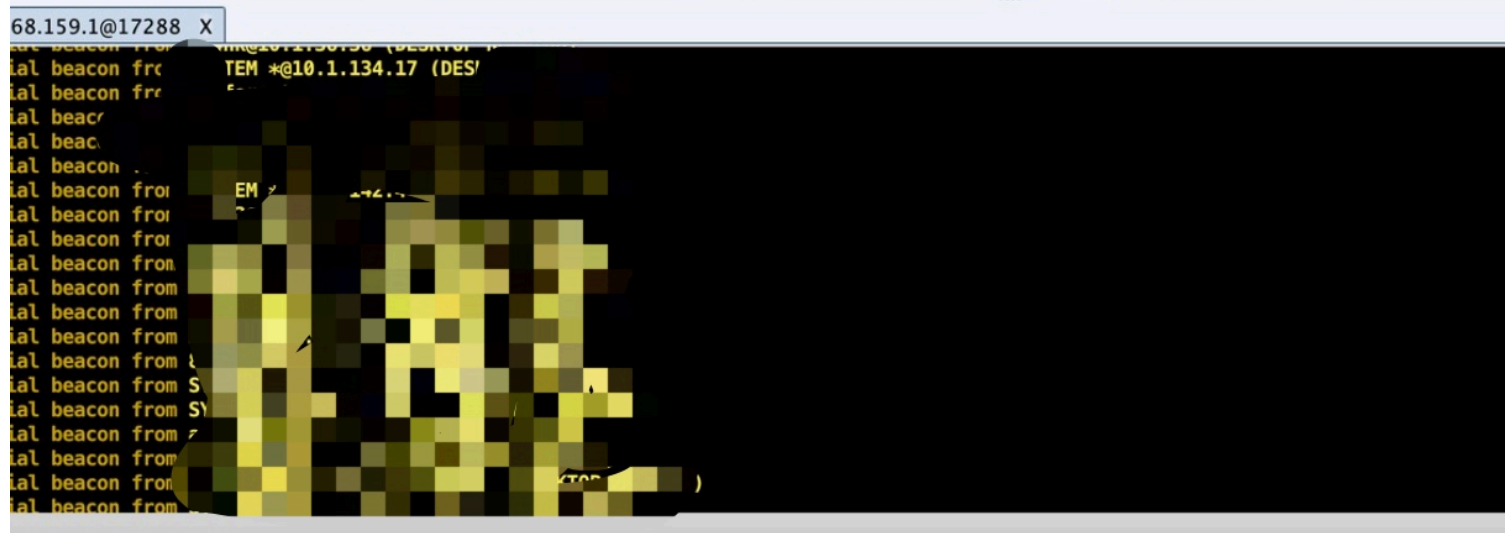
换。不捆绑的话对方点击后无弹出有效文件对方就会警觉，毕竟钓的是开发/运维也属于专业人士了。马采用winrar压缩加密并且采用超长文件名，超长文件名在WinRAR的默认显示中不会显示后缀，只会在后面状态中显示为应用程序，大部分人会直接在WinRAR中直接点开邮件信息。（现自建邮箱大部分会有现邮件网关并且会自动搜寻邮件正文中的明文对邮件进行解压放入沙盒，但是163企业邮好像并无该功能）

选择时间在中午13:50发送，该时间点为大部分人刚午休结束，还没进入工作状态。脑袋转的不是那么的快大部分还属于刚睡醒懵逼状态，对邮件这种通知信息会掉下戒备心并且随意点开看看信息就忘了此事。



邮件发送后陆续上线10余台机器，均为运维/开发价值很高

internal	listener	user	computer	note
10.1.1.1	CDN	admin	DESKTOP-DI2...	Ver: 6.2
10.1.1.2	CDN	admin	DESKTOP-DI2...	Ver: 10.0
10.1.1.3	CDN	admin	DESKTOP-DI2...	Ver: 10.0
10.1.1.4	CDN	admin	LAPTOP-7...	Ver: 6.2
10.1.1.5	CDN	admin	DESKTOP-U...	Ver: 10.0
10.1.1.6	CDN	admin	DESKTOP-U...	Ver: 6.2
10.1.1.7	CDN	admin	DESKTOP-U...	Ver: 10.0
10.1.1.8	CDN	admin	DESKTOP-U...	Ver: 6.2
10.1.1.9	CDN	admin	DESKTOP-U...	Ver: 10.0
10.1.1.10	CDN	admin	DESKTOP-F...	Ver: 6.2
10.1.1.11	CDN	admin	DESKTOP-T...	Ver: 6.2
10.1.1.12	CDN	admin	DESKTOP-I...	Ver: 10.0
10.1.1.13	CDN	admin	DESKTOP-I...	Ver: 10.0
10.1.1.14	CDN	admin	DESKTOP-I...	Ver: 6.2
10.1.1.15	CDN	admin	DESKTOP-I...	Ver: 10.0
10.1.1.16	CDN	admin	DESKTOP-I...	Ver: 6.2
10.1.1.17	CDN	admin	DESKTOP-I...	Ver: 6.2
10.1.1.18	CDN	admin	DESKTOP-I...	Ver: 10.0
10.1.1.19	CDN	admin	DESKTOP-I...	Ver: 10.0
10.1.1.20	CDN	admin	DESKTOP-I...	Ver: 6.2
192.168.1.1	CDN	administrator	DESKTOP-I2V...	Ver: 10.0
192.168.1.2	CDN	admin	DESKTOP-I2V...	Ver: 6.2



上线后不能胡乱操作，先确定好该机器的防护情况，不然一个误操作可能导致员工感到异常，机器下线上线。后拿了处理过的frp直接上线进入内网(去特征+CDN)，我们对其中一个机器进行了导出浏览器凭据，获取了dap认证的账号密码，后发现该内网无域，采用了openldap。

根据对上线机器的信息收集获取到了内网多个应用，其中我们将目光放到了wiki上。

```

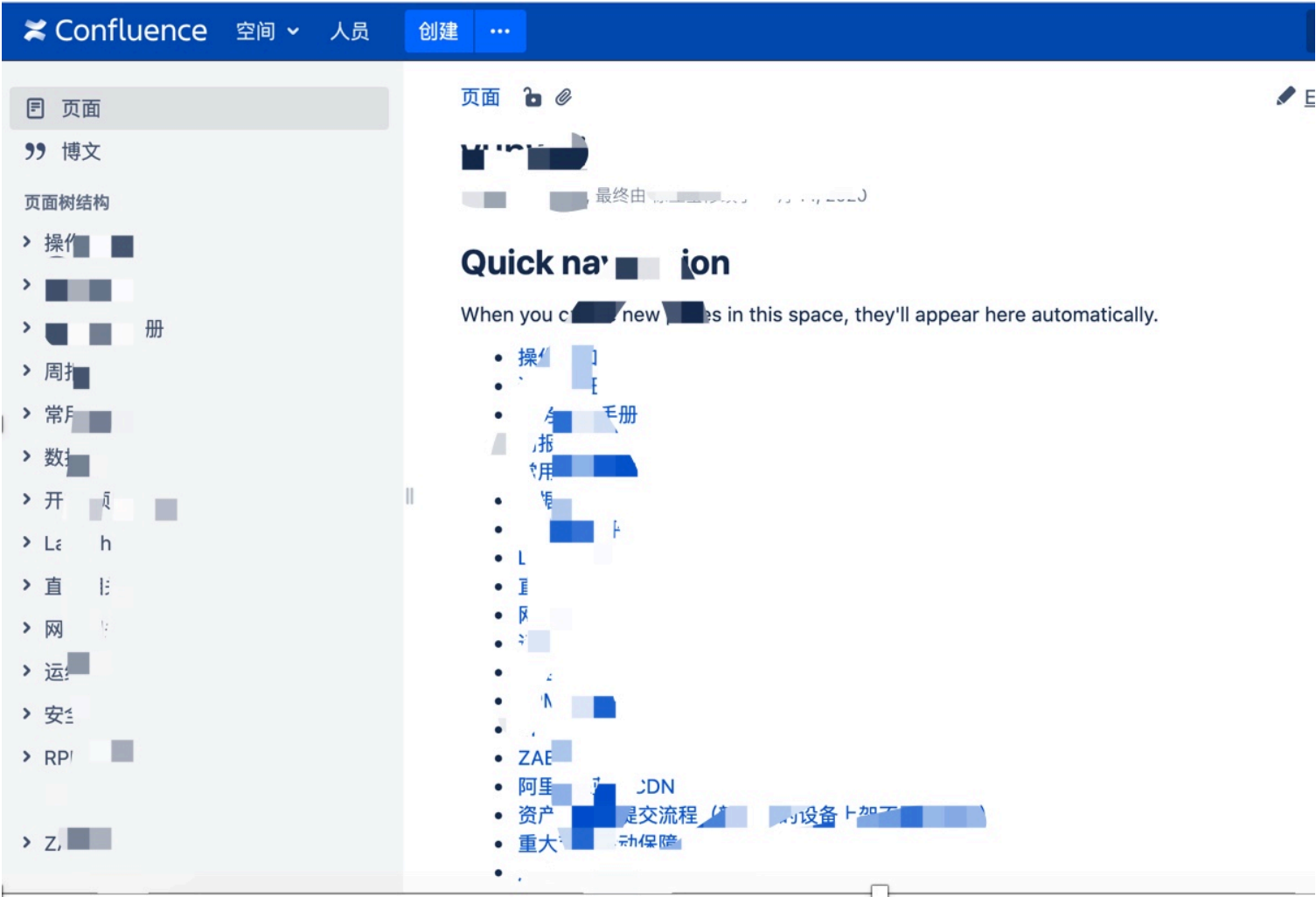
URL -> http://wiki.g.../login.action
USERNAME -> ...
PASSWORD -> ...

```

其实看到wiki以及/login.action的时候我就知道这是个confluence，当时脸上已经开始乐了。众所周知，confluence为互联网公司用的比较多，像运维/开发等空间又往往存放了大量的敏感信

息，资产拓扑，密码等。

并且大部分企业在使用confluence的时候没有做到合理的鉴权，随意一个通过认证的账号就可访问各大目录空间，比如你登录一个市场部的用户正常来说你能访问的目录空间应该只限于市场部，但是没有做鉴权就可以访问任何部门的空间。



在安全部门空间定位到了几个管网络安全设备的人。

2021

=====

本周工作：（5.31~6.4）

1) 定期端口探测情况确认：

2) 攻防演练服务跟进：

本周已正式启动项目，现已正式开始进行攻击阶段；

3) 网安检查：已完成整改计划文档，周五下午报送网安大队；

4) 完善UTS增采论证报告以及立项报告，并已发起采购流程；

5) HIDS告警处理确认：

5.1) 反26 10.87.130 进程存在向 10.87.130 端口的反向连接行为，经确认为运维在用salt推送配置，已

0.4/ 2.0 运行任务/白，已删除定时任务/白

6) WAF项目跟进：

安徽和北京亦庄的WAF已完成部署安装，且安徽已将部分线上流量切入WAF进行加检测；

7) DDOS采购跟进：

因合同生效期拟定在6.15-7.14，故暂无法全量测试，需先确认部分域名进行绑定hosts测试。

下周计划：



21

20

- 1.入职，熟悉工作环境
- 2.检测外网域，发现若干中低风险，形成报告

- 1.熟悉工作环境，安全设备，防病毒，日志审计等
- 2.陕西移动安全扫描报告风险确认，并给出解决方案，证书类型调研
- 3.外网主机漏洞扫描
- 4.跟进处理日志审计日志占硬盘空间过大问题
- 5.内网安全风险测试发现若干高中危风险，汇总

- 1.登录长亭WAF，查看是否有误报
- 2.主机安全扫描，分析扫描安全风险，提醒整改
- 3.内外网安全风险检测，发现若干高中危风险，汇总
- 4.参加新员工培训
- 5.青藤云处理未分组主机

并且获得到了jumpserver堡垒机、一个疑似自研堡垒机、青藤云服务器

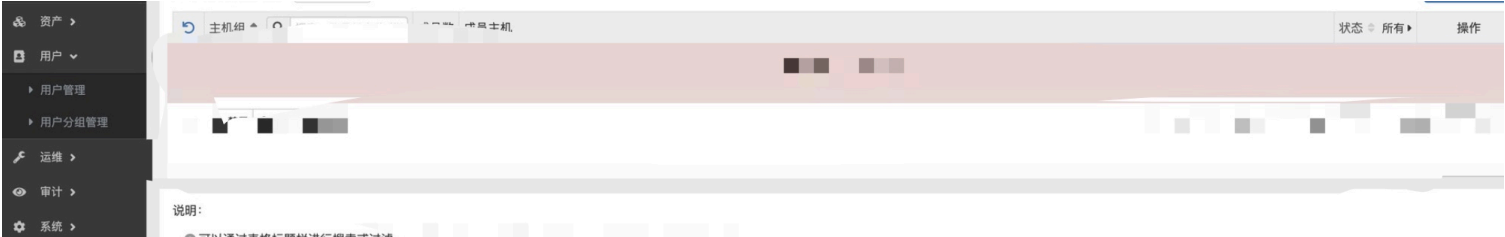
的登录地址。其中我们在运维空间发现了OA的登录地址以及OA服务器登录密码、OA数据库账户密码。后与客户复盘得知OA是几个人在运维因密码复杂，为了方便就将密码放在了wiki里面。

	1	oa.	at	ME
	2	oa.		
	4			

有了数据库的账号密码那就不用登录机器了，直接远程登录数据库，该OA是ekp蓝凌，用户账户密码存放在 ekp--dbo--sys_org_person中。

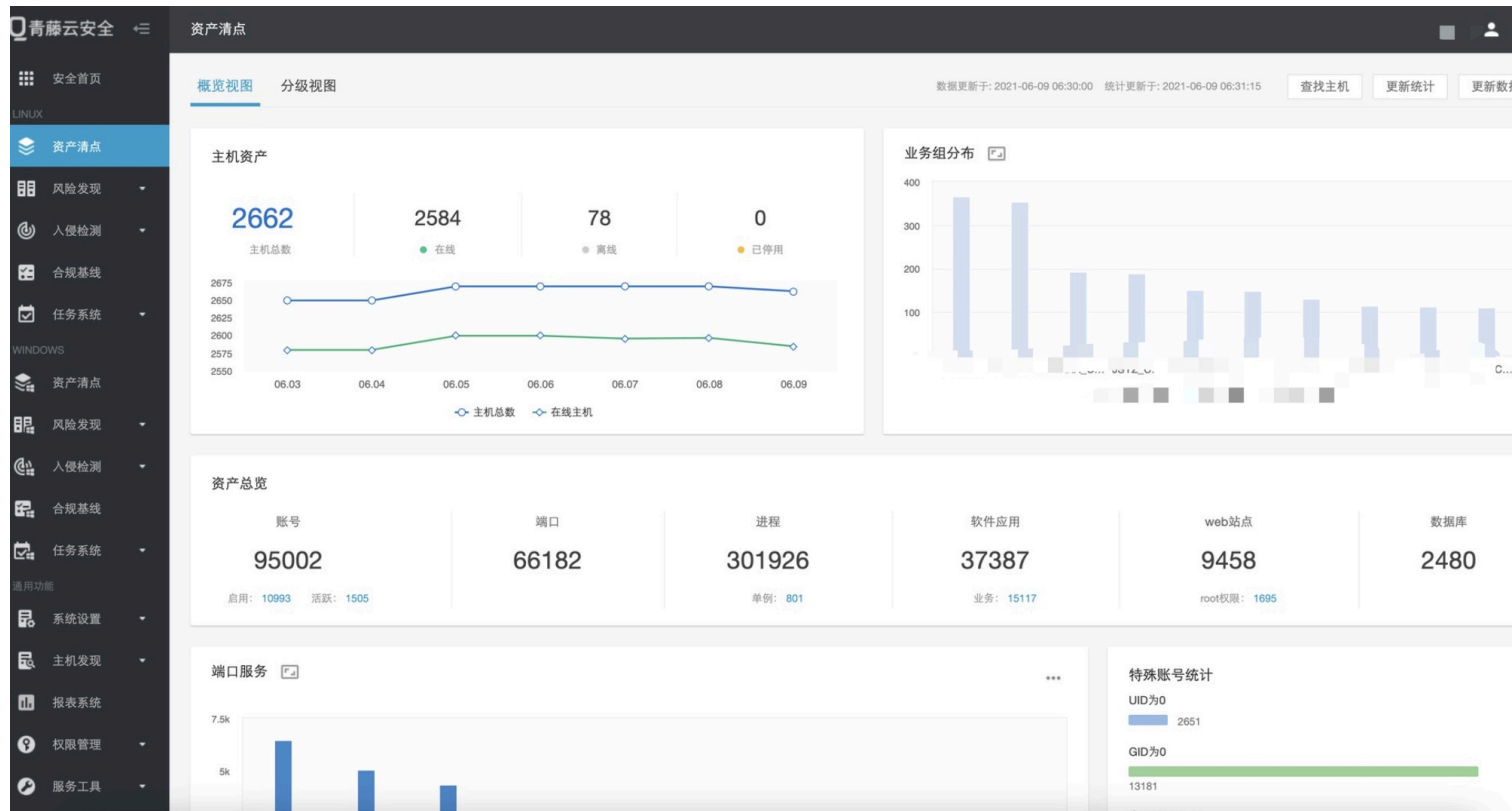
运维人员其中一人解密出hash后登录自研堡垒机发现机器少的可怜，当时判断可能机器依旧还在jumpserver上管控，后续可能会全部上线到自研堡垒机中。但是该账户登录jumpserver的时候发现为密码错误。

后把信息安全部门员工的hash扣出来去解密。成功解密了两人，其他人hash未解出明文。



fd_id	fd_mobile_no	fd_email	fd_login_name	fd_password
001c47a78b2d9	(NULL)	(NULL)	admin	
39d	(NULL)	(NULL)	everyone	
16/ 3baze00	590	(NULL)	anonymous	0
16/ 018f 71758bd	5			5849b
16/ 05e90	5572	ha		4
16/ f8d4	3131	wa		18
16/ 7104	0794	an		e08
16/ 66e	4240	ji		2d
16/ 732	0f			e
16/ :f64	c			c
16/ 3c1	f0			
16/ 324	18			
16/ 104				
16/ 14				
16/ 0b	18			
16/ 4				5
16/ 54	8c			
16/ f4	6			
16/ 3	65			
1/ v	26	j		
1/ 1	3	c		
1/	26	f		
1/	1	v		
1/	29	c		
1/	5	z		
1/ 8c	7	l		2
1/ 0	0	h		8
1/ c	2	h		cb
1/ 8l	36	v		28f
1/ 5f	0	k		f88
1/)	8	y	lcr	394
1/ ld	6	q	lcr	462
1/ i2	3	cl	an	b2c
1/)	11	su		5a
1/ 2	2963	he		e
1/ 49ddafo	499	lu		
1/ 4cd52b9	1	s		
1/ 34d06238c	3	w		
1/ 34d9441c48f				
1/ 534e9160f65a945		cn		8
16/ 053e09465af205aa43e	1	ch		338

我们拿着解出来的密码成功登录的青藤云server控制端

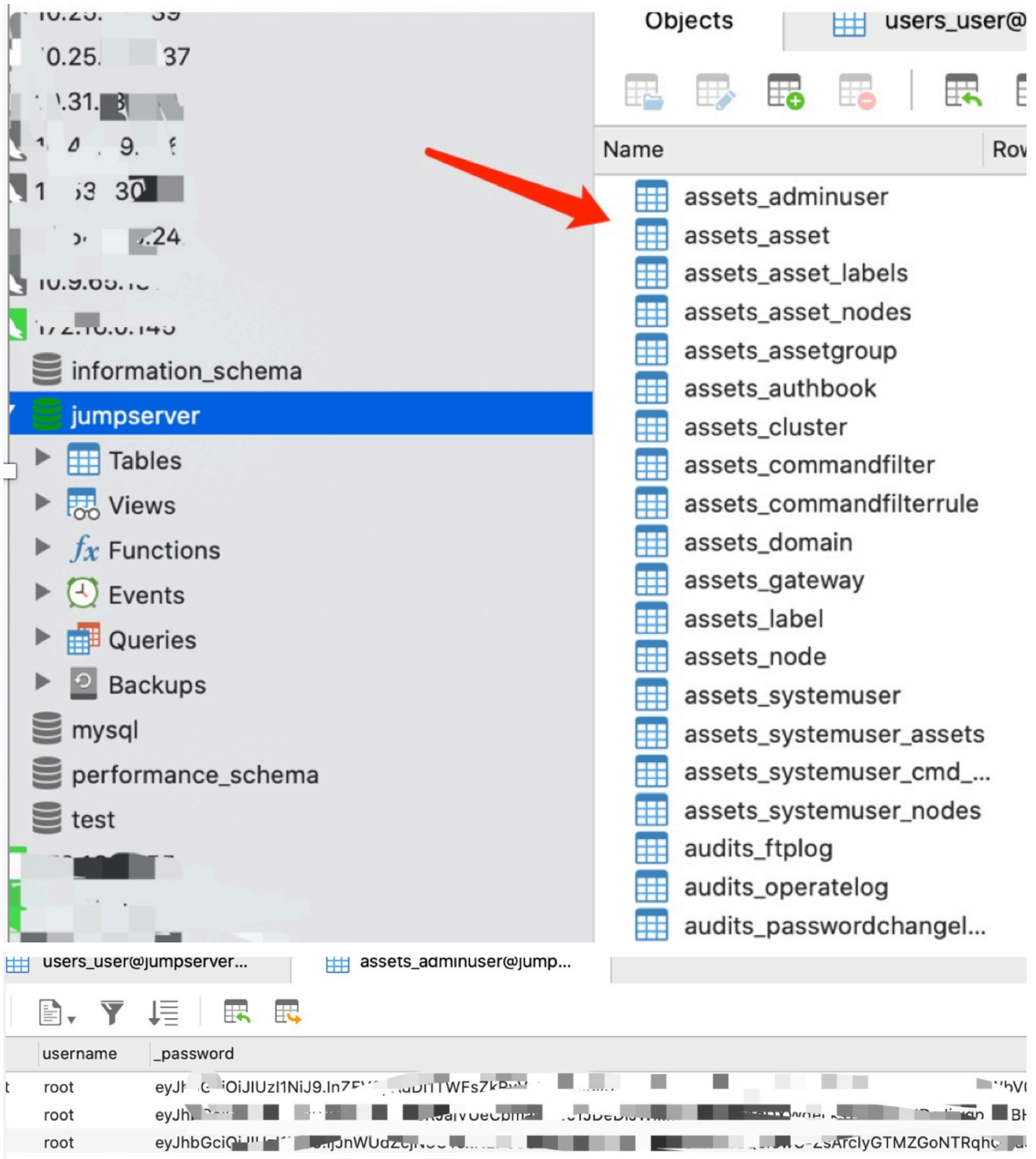


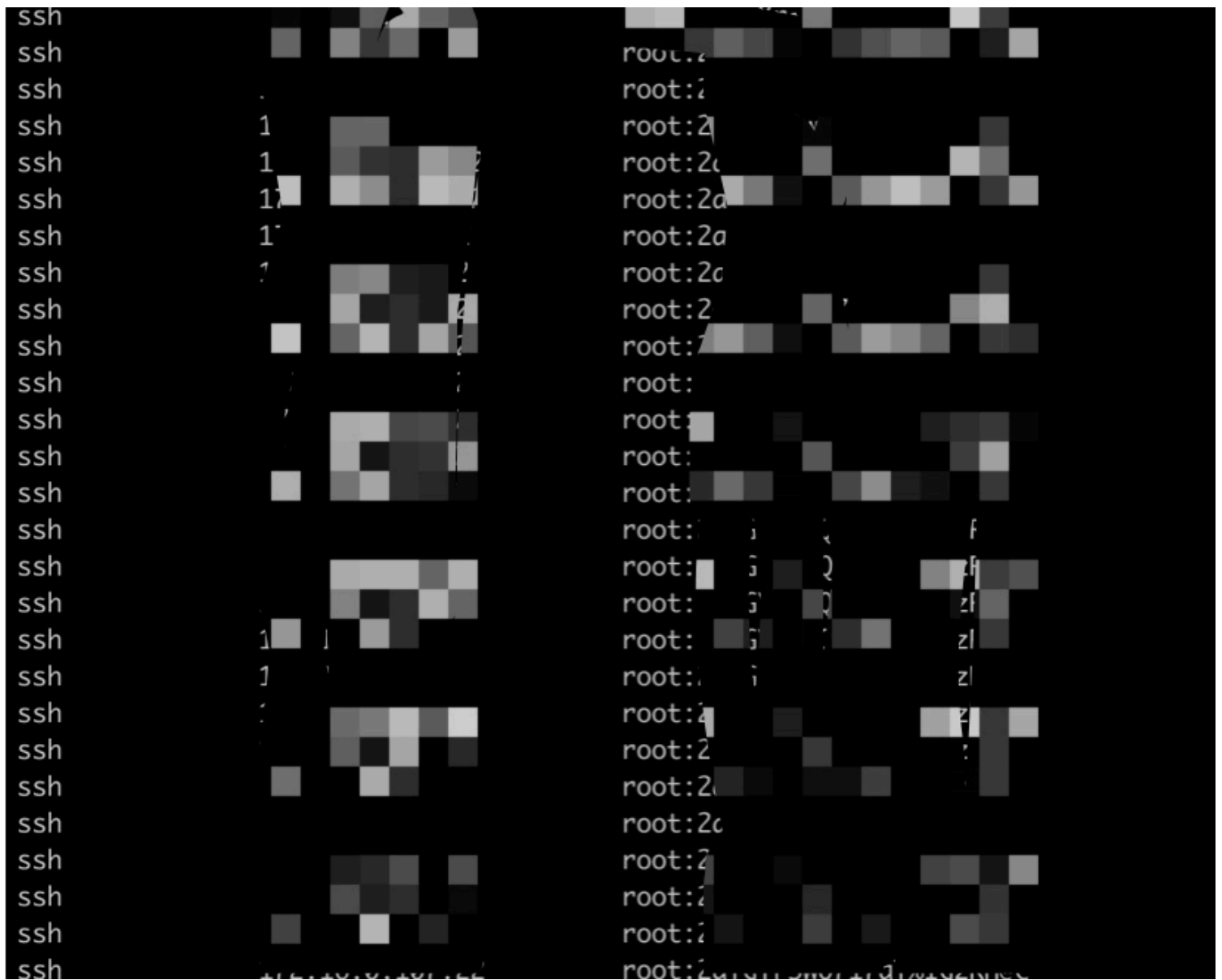
青藤云是可以对所管控主机下发命令的，但是在前台页面去掉了这些功能，需要他们的api文档，可惜的是我并没有这份文档，但是我在任务系统中发现了一个有意思的功能点，读取系统文件功能。我们利用该功能读取了jumpserver堡垒机的history文件，在该文件中读取到了一个关键信息，Jumpserver数据库的密码，运维在对jumpserver数据库运维的过程中在命令行使用 `mysqldump -u root -p xxxx`去备份数据库，在命令行中记录了明文，并且幸运的是该数据库支持外连。

```
{
  "read_status": "success",
  "content_lines": 1,
  "file_content": "1P3N7aC1v",
  "host_id": "1",
  "hostname": "1",
  "external_ip": "1",
  "internal_ip": "1",
  "display_ip": "1"
}
```

```
[文件查询]查看指定文件内容.json
top
#1605580126
vim /etc/ssh/sshd_config
#1605580675
exit
#1605581328
vim /etc/ssh/sshd_config
#1605581343
systemctl restart sshd
#1605581346
ps -ef | grep sshd
```

使用该密码登录了jumpserver数据库，进入jumpserver数据库中的assets_adminuser获取到了所管控制机器的加密密码。





后我们通过数据库的中的密码成功登录了jumpserver堡垒机的系统，通过jumpserver的Django中manager.py工具直接添加账户密码成功登录了堡垒机的web应用。通过堡垒机可直接管控6k余台机器，任务完成。

总结

- gitlab rce进入内网(信息时间差)>>>>>>>>>
- gitlab服务器获取到邮箱账户密码(权限掉线)>>>>>>>>>
- 通过邮件系统对高价值人员进行钓鱼>>>>>>>>>
- 上线后导出浏览器凭据获取内网多个关键应用ldap认证账户密码>>>>>>>>>
- 通过浏览器凭据登录confluence>>>>>>>>>
- confluence中发现关键应用登录地址、关键人员信息以及OA数据库账户密码>>>>>>>>>
- 登录OA数据库解密关键人员hash>>>>>>>>>
- 登录青藤云server端>>>>>>>>>

青藤云server读取堡垒机本地文件获取到堡垒机数据库密码>>>>>>>>>

解密堡垒机数据库中的机器密码成功登录内网大量机器>>>>>>>>>

通过解密密码登录jumpserver服务器，用manager.py成功添加用户登录jumpserver应用。

不足点：开始不应该弹shell，直接wget下个马回来执行就完事了，当时因为他dns不解析，dnslog没接到就傻逼呵呵的去弹一下试试了。

该项目时间为5天，攻击队两人一天时间打完，整个演练中除了开始的gitlab造成了告警以外，内网中仅仅依靠对所控主机信息收集，正常业务登录，安全设备应用功能使用 做到了内网无扫描、无告警在防守方依旧高度等待告警排查中打穿了内网。

未经本人同意禁止以任何形式转载此文章，感谢。
