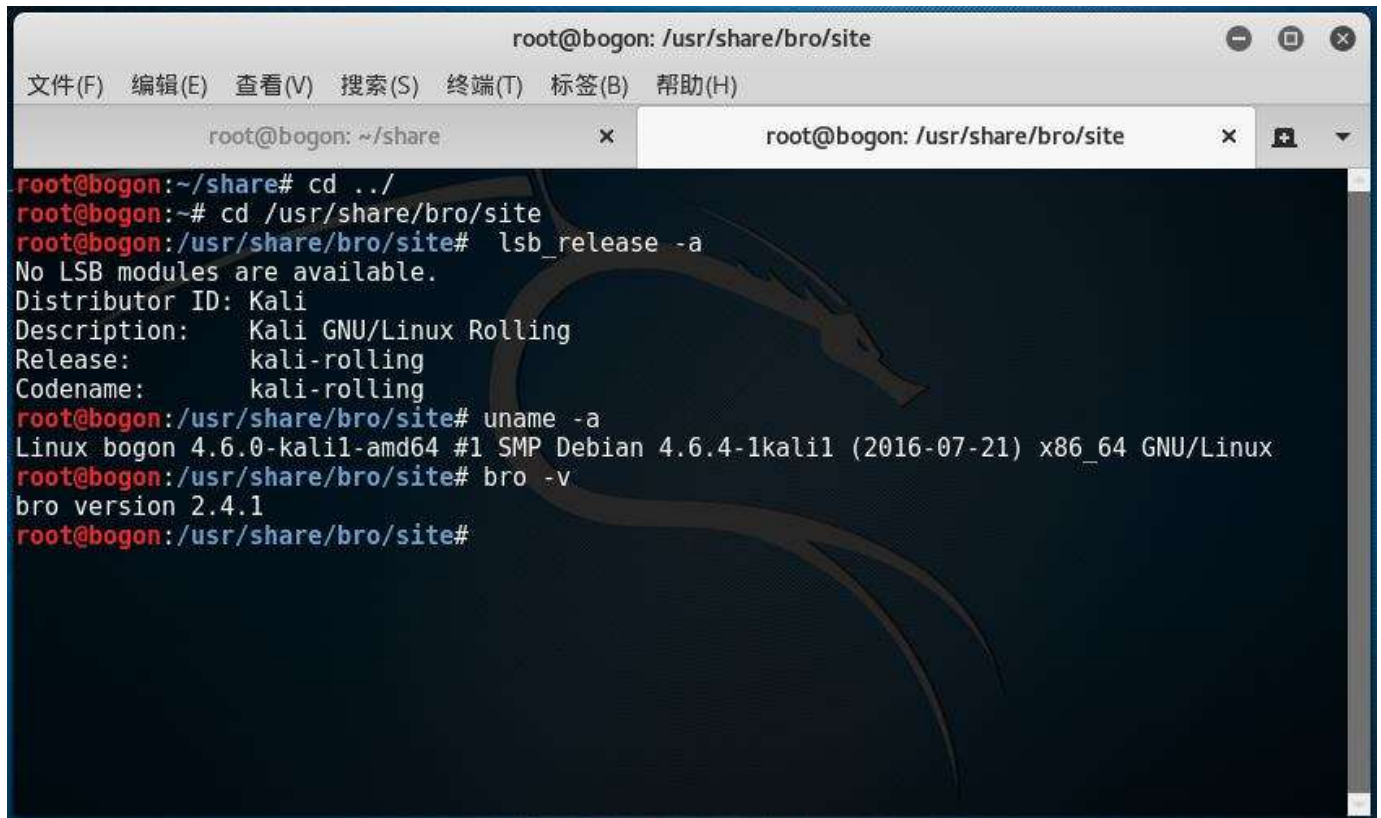


计算机入侵取证

原始数据：

包含可疑流量的pcap数据包（attack-trace.pcap）

实验环境：



```
root@bogon: /usr/share/bro/site
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B) 帮助(H)
root@bogon: ~/share x root@bogon: /usr/share/bro/site x
root@bogon:~/share# cd ../
root@bogon:~# cd /usr/share/bro/site
root@bogon:/usr/share/bro/site# lsb_release -a
No LSB modules are available.
Distributor ID: Kali
Description:    Kali GNU/Linux Rolling
Release:        kali-rolling
Codename:       kali-rolling
root@bogon:/usr/share/bro/site# uname -a
Linux bogon 4.6.0-kali1-amd64 #1 SMP Debian 4.6.4-1kali1 (2016-07-21) x86_64 GNU/Linux
root@bogon:/usr/share/bro/site# bro -v
bro version 2.4.1
root@bogon:/usr/share/bro/site#
```

实验目标：

通过分析数据样本包，获取包含的TCP并发会话数量、攻击者和靶机的IP地址、靶机在此次安全事件中被利用的漏洞等信息。

方法一：基于bro分析

一、什么是bro？

Bro is a powerful network analysis framework（网络入侵检测系统）。While focusing on network security monitoring（安全监控），Bro provides a comprehensive platform for more general network traffic analysis as well（流量分析）。Today, it is relied upon operationally in particular by many scientific environments for securing their cyberinfrastructure.

二、实验步骤

1、配置bro

- 操作：编辑/etc/bro/site/local.bro文件，尾部追加

```
@load frameworks/files/extract-all-files
@load mytuning.bro
```

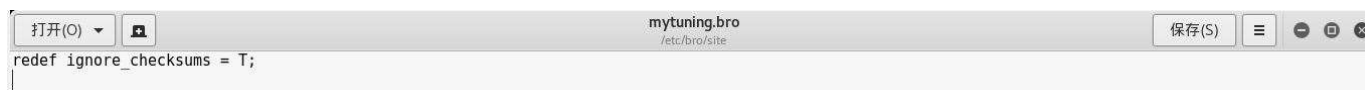
- 说明：加载库指令，以@load来加载module中定义的命名空间。命令1表示使用bro提取所有文件，命令2表示加载mytuning.bro中我们自己编写的指令

```
@load frameworks/files/extract-all-files
@load mytuning.bro
```

- 操作：在/etc/bro/site/目录下mytuning.bro，写入代码：

```
redef ignore_checksums = T;
```

- 说明：使bro能够在系统上分析本地捕获流量，忽略校验和验证。

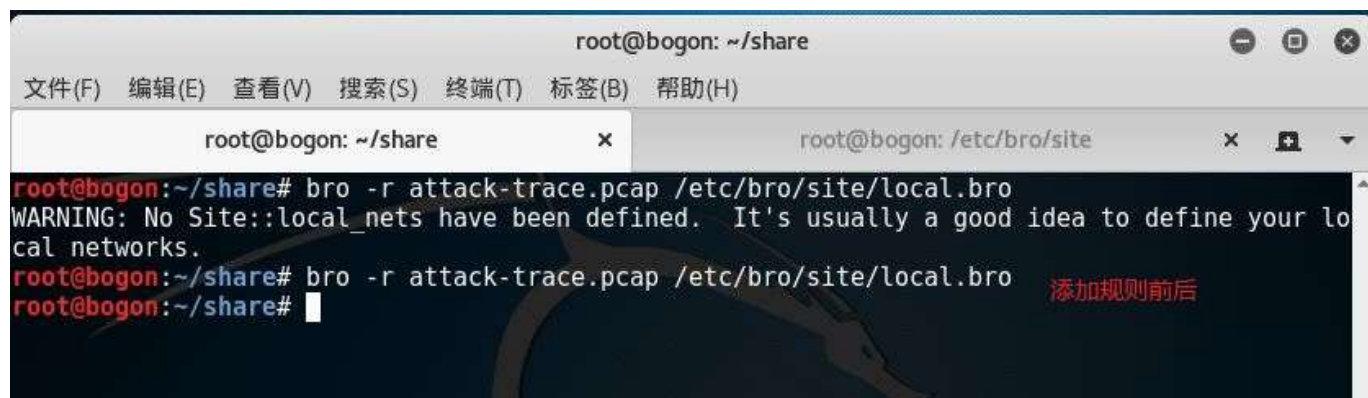


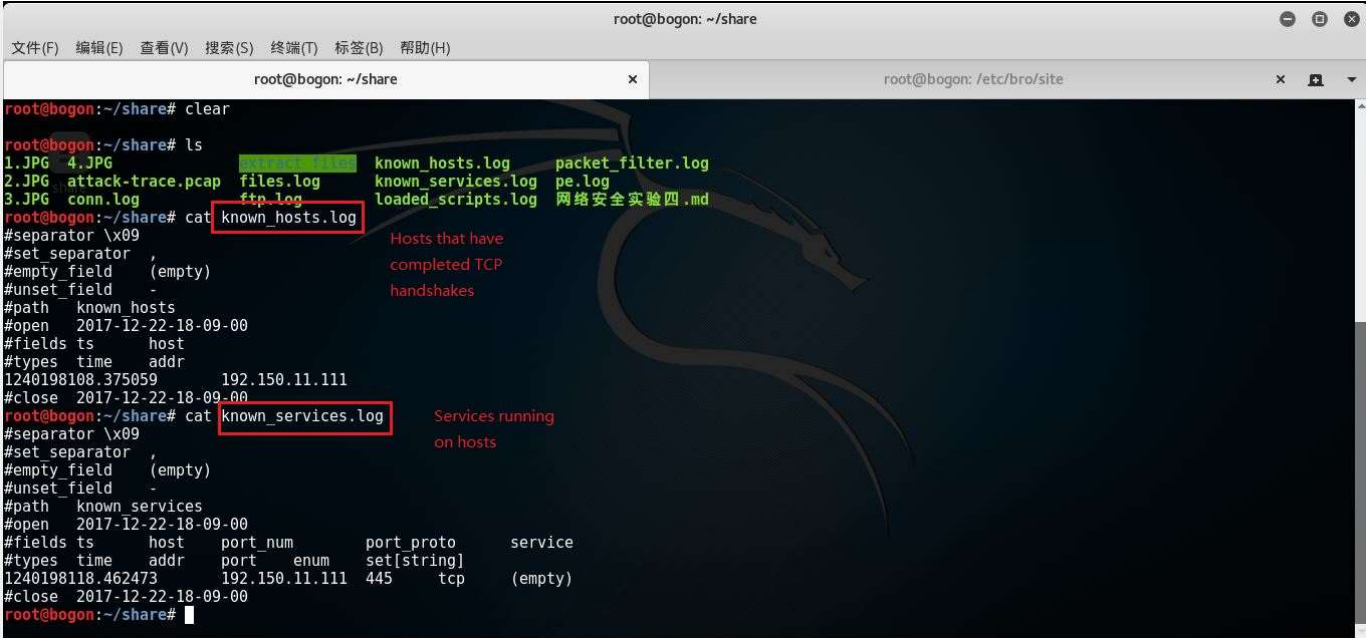
2、使用bro分析流量包

- 操作：命令行中输入 `bro -r attack-trace.pcap /etc/bro/site/local.bro`
- 说明：

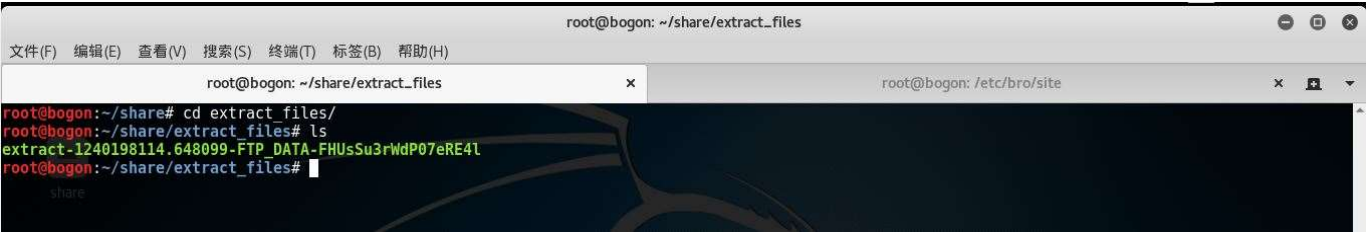
```
-r: 表示分析的为数据包
attack-trace.pcap: 待分析数据包
/etc/bro/site/local.bro:
```

- 注意：直接运行将会弹出警告信息，不影响分析结果，可在mytuning.bro中写入 `redef Site::local_nets = { 本地网络IP地址范围（因本数据包中假定靶机IP为192.150.11.1，故将此处填写为192.150.11.0/24）};` 消除。添加后会新增2个日志文件，报告待分析数据包中发现的本地网络IP和该IP关联的已知服务信息





3、在线分析extract_files目录下的文件，确定攻击类型



- 经过在线分析表明这是一个已知蠕虫病毒，接下来将寻找入侵线索。

4、阅读源代码分析入侵线索

- 操作：阅读/usr/share/bro/base/files/extract/main.bro
- 说明：通过阅读main.bro，可知文件名最后的id是file.log中文件唯一标识符


```
function on_add(f: fa_file, args: Files::AnalyzerArgs)
{
    if ( ! args?$extract filename )
        args$extract_filename = cat("extract-", f$last_active, "-", f$source,
            "-", f$id);

    f$info$extracted = args$extract_filename;
    args$extract_filename = build_path_compressed(prefix, args$extract_filename;
    mkdir(prefix);
}
```

文件名组成: extract-最近活跃时间-来源-id

提取出的文件名

名称	修改日期	类型	大小
extract-1240198114.648099-FTP_DATA-FHUsSu3rWdP07eRE4l	2017/12/22 18:57	648099-FTP_DAT...	155 KB

- 操作: 阅读file.log
- 说明: 获得对应conn_uids

```
root@bogon: ~/share
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B) 帮助(H)

root@bogon: ~/share x root@bogon: /etc/bro/site x

root@bogon:~/share# ls
1.JPG 3.JPG 5.JPG 7.JPG 9.JPG conn.log files.log known_hosts.log loaded_scripts.log pe.log
2.JPG 4.JPG 6.JPG 8.JPG attack-trace.pcap ftp.log known_services.log packet_filter.log 网络安全实验四.md
root@bogon:~/share# cat files.log
#separator \x09
#set_separator ,
#empty_field (empty)
#unset_field -
#path files
#open 2017-12-22-18-57-09
#fields ts fuid tx_hosts rx_hosts conn_uids source_depth analyzers mime_type filename duration l
ocal_orig is_orig seen_bytes total_bytes missing_bytes overflow_bytes timedout parent_fuid md5 sha1 sha256 extra
cted
#types time string set[addr] set[addr] set[string] string count set[string] string string interval bool boolc
count count count count count count count count count count count count count count count count count count
1240198114.648099 FHUsSu3rWdP07eRE4l 98.114.205.102 192.150.11.111 CaekM71dMupq7iJ2C9 FTP_DATA 0 SHA1,PE,MD5,EXTRACT a
pplication/x-dosexec - 9.76/306 F T 158720 - 0 0 F 14a09a48ad23fe0ea5a180bee8cb750a a
c3cdd673f5126bc49faa72fb52284f513929db4 - extract-1240198114.648099-FTP_DATA-FHUsSu3rWdP07eRE4l
#close 2017-12-22-18-57-09
root@bogon:~/share#
```

- 操作: 阅读conn.log
- 说明: 获得攻击者ip地址

```
root@bogon:~/share# cat conn.log
#separator \x09
#set_separator ,
#empty_field (empty)
#unset_field -
#path conn
#open 2017-12-22-18-57-09
#fields ts uid id.orig_h id.orig_p id.resp_h id.resp_p proto service duration orig_bytes resp_bytes c
onn_state local_orig local_resp missed_bytes history id_resp_p orig_ip_bytes resp_ip_bytes resp_bytes c
s
#types time string addr port addr port enum string interval count count string bool bool count string count
count count count count count count count count count count count count count count count count count count
1240198108.374595 CGu4dhISPhyX2bs28 98.114.205.102 1821 192.150.11.111 445 tcp - 0.238169 0 0 SF F
T 0 ShAFaf 4 168 3 128 (empty)
1240198110.466428 ClBl1H2N4xj9bfgtCg 98.114.205.102 1924 192.150.11.111 1957 tcp - 2.980258 133 2 SF F
T 0 ShAdDaFf 6 381 6 250 (empty)
1240198108.509145 ChwUoT2s01Xe3PnPv6 98.114.205.102 1828 192.150.11.111 445 tcp - 4.938123 4209 902 RSTOF
T 0 ShAdadR 14 4777 17 1590 (empty)
1240198113.457215 CoQRu2aPtZqdmqnsL 192.150.11.111 36296 98.114.205.102 8884 tcp ftp 11.136591 77 214 SF T
F 0 ShAdDFaRf 15 841 12 850 (empty)
1240198114.516921 CaekM71dMupq7iJ2C9 98.114.205.102 2152 192.150.11.111 1080 tcp ftp-data 9.954513 158720 0 S
F 0 ShAdAff 159 165088 112 4488 (empty)
#close 2017-12-22-18-57-09
root@bogon:~/share#
```

5、log文件查看技巧

查看conn.log中所有可用的“列名”: `grep ^#fields conn.log | tr '\t' '\n'`

按照“列名”输出conn.log中我们关注的一些“列”: `bro-cut ts id.orig_h id.orig_p id.resp_h id_resp_p proto < conn.log`

将UNIX时间戳格式转换成人类可读的时间: `bro-cut -d < conn.log`

方法二：tshark与snort配合使用

一、什么是snort？

一个强大的网络入侵检测系统。它具有实时数据流量分析和记录IP网络数据包的能力，能够进行协议分析，对网络数据包内容进行搜索/匹配。它能够检测各种不同的攻击方式，对攻击进行实时报警。此外，Snort是开源的入侵检测系统，并具有很好的扩展性和可移植性。

二、实验步骤

1、统计数据包内流量信息

- 操作：使用 `tshark -r attack-trace.pcap -z ip_hosts,tree -qn` 或 `wireshark` 端点统计工具
- 说明：统计数据包内信息

-r：后跟待分析流量包

-z：设置统计参数

```
root@bogon: ~/share
root@bogon: /etc/bro/site

root@bogon:~/share# tshark -r attack-trace.pcap -z ip_hosts,tree -qn
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
[string "/usr/share/wireshark/init.lua"]:44: dofile has been disabled due to running Wireshark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges for help in running Wireshark as an unprivileged user.

=====
IPv4 Statistics/All Addresses:
Topic / Item      Count      Average      Min val      Max val      Rate (ms)      Percent      Burst rate      Burst start
-----
All Addresses     348         0.0215       0.0215       0.0215       100%          0.0800        6.395
98.114.205.102    348         0.0215       0.0215       0.0215       100.00%       0.0800        6.395
192.150.11.111    348         0.0215       0.0215       0.0215       100.00%       0.0800        6.395
=====

root@bogon:~/share#
```

Wireshark · Endpoints · attack-trace

Ethernet · 2IPv4 · 2IPv6TCP · 9UDP

Address	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Latitude	Longitude
98.114.205.102	348	183 k	195	174 k	153	9439	—	—
192.150.11.111	348	183 k	153	9439	195	174 k	—	—

☐ 解析名称☐ 显示过滤器的限制

Endpoint 类型 ▾

复制 ▾映射CloseHelp

2、发现攻击者ip

- 操作：使用命令 `tshark -r attack-trace.pcap -Y "tcp.flags==0x02" -n`
- 说明：筛选建立tcp连接的数据包

```
root@bogon:~/share# tshark -r attack-trace.pcap -Y "tcp.flags==0x02" -n
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
[string "/usr/share/wireshark/init.lua"]:44: dofile has been disabled due to running Wireshark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges for help in running Wireshark as an unprivileged user.

1  0.000000 98.114.205.102 → 192.150.11.111 TCP 62 1821-445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
5  0.134550 98.114.205.102 → 192.150.11.111 TCP 62 1828-445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
36 2.091833 98.114.205.102 → 192.150.11.111 TCP 62 1924-1957 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
50 5.082620 192.150.11.111 → 98.114.205.102 TCP 74 36296-8884 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=4055633882 TSecr=0 WS=128
68 6.142326 98.114.205.102 → 192.150.11.111 TCP 62 2152-1080 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1

root@bogon:~/share#
```

3、查看攻击持续时间

- 操作：capinfos attack-trace.pcap
- 说明：计算捕获到数据包的起始时间

```

root@bogon: ~/share
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B) 帮助(H)

root@bogon: ~/share x root@bogon: /etc/bro/site

root@bogon:~/share# capinfos attack-trace.pcap
File name:          attack-trace.pcap
File type:          Wireshark/tcpdump/... - pcap
File encapsulation: Ethernet
File timestamp precision: microseconds (6)
Packet size limit:  file hdr: 65535 bytes
Number of packets:  348
File size:          189 kB
Data size:          183 kB
Capture duration:   16.219218 seconds
First packet time:  2009-04-20 11:28:28.374595
Last packet time:   2009-04-20 11:28:44.593813
Data byte rate:     11 kBps
Data bit rate:      90 kbps
Average packet size: 527.33 bytes
Average packet rate: 21 packets/s
SHA1:               d261a70feeeabcd49a6cfd33087989b472fd80d
RIPEMD160:          39bfcee293ca77ddc9e3a9d24b81a080739e596c
MD5:                6ad68928fe8062632c12c432ce785ac5
Strict time order:  True
Number of interfaces in file: 1
Interface #0 info:
  Encapsulation = Ethernet (1/1 - ether)
  Capture length = 65535
  Time precision = microseconds (6)
  Time ticks per second = 1000000
  Number of stat entries = 0
  Number of packets = 348
root@bogon:~/share#

```

4、利用snort进行行为分析

- 操作：修改/etc/snort/snort.conf，将var HOME_NET后修改为[192.150.11.0/24]（本实验中靶机地址假定为192.150.11.1）；
执行 `sudo snort -q -A console -c /etc/snort/snort.conf -r attack-trace.pcap`
- 说明：-q：不显示状态报告 -A：信息输出到控制台 -c：指定使用的配置文件 -r：待分析数据包

```

Fatal Error, Quitting..
root@bogon:~/share# sudo snort -q -A console -c /etc/snort/snort.conf -r attack-trace.pcap
04/20-11:28:29.447746  [**] [1:2466:7] NETBIOS SMB-DS IPC$ unicode share access [**] [Classification: Generic Protocol Command Decode] [Priority: 3]
{TCP} 98.114.205.102:1828 -> 192.150.11.111:445
04/20-11:28:30.172468  [**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS DsRolerUpgradeDownlevelServer exploit attempt [**] [Classification: Attempted Admini
nistrator Privilege Gain] [Priority: 1] {TCP} 98.114.205.102:1828 -> 192.150.11.111:445
root@bogon:~/share#

```

危险动作：尝试漏洞利用

5、统计会话进行攻击过程划分

- 操作：执行 `tshark -r attack-trace.pcap -qnz conv,tcp`
- 说明：以表格形式统计tcp会话

```

root@bogon:~/share# tshark -r attack-trace.pcap -qnz conv,tcp
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
[string "/usr/share/wireshark/init.lua"]:44: dofile has been disabled due to running Wireshark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges for help in running Wireshark as an unprivileged user.
=====
TCP Conversations
Filter:<No Filter>

```

		<-		->		Total		Relative	Duration
		Frames	Bytes	Frames	Bytes	Frames	Bytes	Start	
98.114.205.102:2152	<-> 192.150.11.111:1080	112	6056	159	167332	271	173388	6.142326000	10.0719
98.114.205.102:1828	<-> 192.150.11.111:445	17	1828	14	4997	31	6825	0.134550000	4.9381
192.150.11.111:36296	<-> 98.114.205.102:8884	12	1018	15	1051	27	2069	5.082620000	11.1366
192.150.11.111:1957	<-> 98.114.205.102:1924	6	483	6	334	12	817	2.091833000	3.1000
98.114.205.102:1821	<-> 192.150.11.111:445	3	170	4	242	7	412	0.000000000	0.3543

猜测：扫描/枚举、漏洞利用、执行攻击指令、FTP会话、下载恶意代码

5、利用wireshark追踪tcp流，线上分析tcp报文

- 操作：打开wireshark，点击统计——>对话——>TCP，选中一条tcp会话应用为过滤器，而后跟踪tcp流，获取完整报文信息。将报文信息存储到文件中，进行线上分析

Wireshark · Conversations · attack-trace

Ethernet · 1IPv4 · 1IPv6TCP · 5UDP

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A	
98.114.205.102	1821	192.150.11.111	445	7	412	4	242	3	170	0.000000	0.3543	5464		3838
98.114.205.102	1828	192.150.11.111	445	31	6825	14	4997	17	1828	0.134550	4.9381	8095		2961
98.114.205.102	2152	192.150.11.111	1080	271	173 k	159	167 k	112	6056	6.142326	10.0719	132 k		4810
192.150.11.111	1957	98.114.205.102	1924	12	817	6	334	6	483	2.091833	3.1000	861		1246
192.150.11.111	36296	98.114.205.102	8884	27	2069	15	1051	12	1018	5.082620	11.1366	754		731

☐ 解析名称☐ 显示过滤器的限制☐ Absolute start time

Conversation 类型

复制Follow Stream...Graph...CloseHelp

检出率

SHA256

分析时间

Tags

用户标记

2 / 25

f67f5ea265250942e6d7e5a9f545249fef16a1c0777c4f2c06d48073c0abbf7a

2017-12-22 20:25:21 (6分钟前)

正常文件(0)

恶意文件(0)

 添加用户标签

检测结果

静态信息

行为分析

网络活动

可视分析

用户标签

反病毒软件	结果	病毒库日期
NANO	Exploit.Raw.Cve-2009-3129.ufuxb	2017-12-22
Avast	MPPT97:ShellCode-O	2017-12-22

参考材料

教材、课件

<http://www.freebuf.com/articles/system/135843.html>

<https://www.jianshu.com/p/113345bbf2f7>