Malicious Traffic Analysis

Project Presentation

Park Kittipatkul

Attila Csokai

Kevin Tropeck

Woon Chiat Sim

Agenda

- Problem Statement
- Definitions
- Approach / Results
- Conclusion

Problem Statement

- flow data entering and leaving a network looks the same
- hard to distinguish malicious traffic from normal
- define various forms of malicious traffic & its characteristics
- isolate those traffic patterns entering and leaving an organization's network using SiLK filters
- attempt to match these malicious traffic patterns to known malware behaviors

Definitions

DoS attack

- goal of attack is to exhaust the resources of the target by overwhelming its communication requests
- there are many forms of DoS attacks, a more common one is the **SYN Flood**
- attacker sends multiple single packet SYN requests, usually using a spoofed source IP
- typically have the SYN flag set only
- the response to the spoofed source address is never responded to, leaving return SYN/ACK packets unanswered

Definitions

Port Scans

- looks for open and/or listening ports on hosts small 1-3 packets/flow with a 40 bytes/packet high ratio of SYN requests versus ACK responses and follow-up connections
- **Horizontal Scan** will scan multiple destination IP addresses with a single or few destination ports
- **Vertical Scan** will scan a single destination IP address and multiple destination ports
- **General Scan** will scan just a few destination IP addresses/ ports and may have to rely on the timing between sent SYN packets

Definitions

SPAM

- traffic characteristics of spamming traffic are to send a high volume of email in a short period of time
- we attempt to differentiate between SMTP clients and servers by looking at whether they respond to communication attempts on port 25

Definitions

Worms

- worms will try to copy themselves to other hosts and network shares without user interaction
- worms will use horizontal port scans as one method to locate other hosts and network shares
- worm scan will have a constant packet size and or flow
- type of flow data generated is usually very worm specific, depending on what, who and how it is trying to infect

Definitions

Beaconing / Command and Control

- beaconing is a signaling method by which a host indicates its presence
- for our interest, beaconing is used by infected hosts to signal their presence and availability to the master signaling characteristics include regular timings per source and destination IP address pairs using DNS, IRC, and HTTP protocols
- on average, the beaconing data will be fairly constant in terms of byte size and flow
- Command and Control (C&C), is command script from a worms' author to the worms instructing them to perform certain tasks

SYN Flood

- Look for large number of single packet SYN requests to a small number of destination IP/ports.
- Filter used:

```
rwfilter /afs/andrew/course/95/855/ext3/2008* --
   syn=1 --ack=0 --fin=0 --proto=6 --dport=80 --
   pass=stdout | rwuniq --fields=sip --all-counts --
   flows=500 --dip-distinct
```

SYN Flood

- Results:

```
rwfilter /afs/andrew/course/95/855/ext3/2008* --syn=1 --ack=0 --fin=0 --proto=6 --dport=80 --pass=stdout | /
rwuniq --fields=sip --all-counts --flows=460 --dip-distinct
rwuniq: Warning: Using default temporary directory /tmp
```

sIP	Bytes	Packets	Records	min_sTime	max_eTime	Unique_DIP
153.59.117.10	35100	585	585	2008/06/16T05:00:28	2008/06/16T05:06:37	6
192.120.244.27	45248	902	474	2008/10/17T05:00:03	2008/10/17T05:14:40	6
174.50.237.46	82116	1582	1511	2008/10/17T05:00:12	2008/10/17T05:10:25	10
201.38.44.55	31680	528	528	2008/09/16T05:00:02	2008/09/16T05:15:00	1
218.41.233.62	1335600	27825	24638	2008/10/17T05:00:01	2008/10/17T05:15:00	1
55.57.79.114	1976400	41175	34723	2008/10/08T05:00:01	2008/10/08T05:15:00	1
195.120.85.122	22888	477	476	2008/05/15T05:01:19	2008/05/15T05:14:25	476
9.128.167.132	1522224	31713	27765	2008/10/17T05:00:01	2008/10/17T05:15:00	1
138.32.194.147	38016	792	792	2008/05/15T05:00:02	2008/05/15T05:14:59	792
44.104.112.174	54360	906	505	2008/07/16T05:01:18	2008/07/16T05:14:00	3
17.230.154.180	84600	1410	655	2008/09/16T05:00:51	2008/09/16T05:14:32	3
209.210.16.203	109536	2282	1066	2008/07/16T05:00:06	2008/07/16T05:15:00	101
134.24.122.225	765400	15946	15943	2008/08/19T05:00:00	2008/08/19T05:15:00	2
6.200.124.228	1638096	34127	34104	2008/10/17T05:00:01	2008/10/17T05:15:00	1

Spam

- Look for:
  - SMTP client sending to lots of different servers
  - "Speaks" mostly SMTP
  - Gets RSTs from some of its intended peers.
- Filters used:
  - To identify SMTP clients:

```
rwfilter ../200809161400 --pass=stdout --flags-
  all=SAF/SAF --packets=5- --dport=25 | rwuniq --
  fields=sip --flows=10 --no-title --delimited=" " |
  awk '{print $1}' | rwsetbuild stdin
  200809161400.smtp.clients.set
```

Spam

- To identify SMTP servers:

```
rwfilter ../200809161400 --pass=stdout –packets=3- --
  flags-initial=SA/SFA --sport=25  | rwuniq --
  fields=sip,sport --flows=5 --no-title --delimited=" "
  | awk '{print $1}' | rwsetbuild stdin
  00809161400.smtp.servers.set
```

- To identify SMTP clients who are not SMTP servers:

```
rwsettool --difference 200809161400.smtp.clients.set
  00809161400.smtp.servers.set --
  output=200809161400.possible.spammers.set
```

- To see the distinct destinations a “suspect” connects to:

```
rwfilter ../200809161400 --pass=stdout --flags-
  all=SAF/SAF --packets=5- --dport=25 --
  saddress=192.117.128.200 | rwuniq --fields=dip
```

Spam

- To see the major protocols used:

```
rwfilter ../200809161400 --pass=stdout --flags-
  all=SAF/SAF --saddress=192.117.128.200 | rwstats –
  dport –count=10
```

- See how many RST a “suspect” received:

```
rwfilter ../200809161400 --pass=stdout --flags-
  initial=R/SAFR --daddress=192.117.128.200 --sport=25|
  rwcut --fields=sip,dip
```

Spam

- Results (from find_spammers.py):

```
---> Possible spammer's IP: 207.98.32.161
            dIP|   Records|
207.175.206.133|         1|
 207.16.229.140|         1|
   48.32.84.215|         1|
   194.108.6.70|         1|
221.186.211.202|         1|
   76.162.84.78|         1|
  82.255.24.111|         1|
   195.200.19.4|         1|
 192.192.123.53|         1|
    48.3.228.56|         1|
 192.192.123.52|         1|
  66.193.199.46|         1|
  82.255.24.152|         1|
   <snip>
```

Spam

```
Protocol distribution:
            25|                  86| 97.727273| 97.727273|
         22942|                   1|  1.136364| 98.863636|
          4490|                   1|  1.136364|100.000000|

RSTs received from:
            sIP|
221.186.155.119|
221.186.155.119|
   48.32.84.215|
  220.209.81.60|
  220.209.81.60|
    71.40.6.150|
 48.216.152.218|
221.186.155.119|
 202.40.249.140|
  66.193.199.46|
 67.116.241.209|
  211.43.166.30|
 <snip>
```

Worm Traffic

- Look for traffic from high-port to high-port, scanning for and connect to well-known backdoor ports.
- Filter Used:

```
rwfilter ./2008* --sport=1024- --dport=1024- --
  pass=stdout | rwstats --dip --top --count=50
                dip/sip/dport/sport

rwfilter ./2008* --sport=3127 --dport=1024- --
  pass=stdout | rwstats --sip --top --count=50
```

Worm Traffic

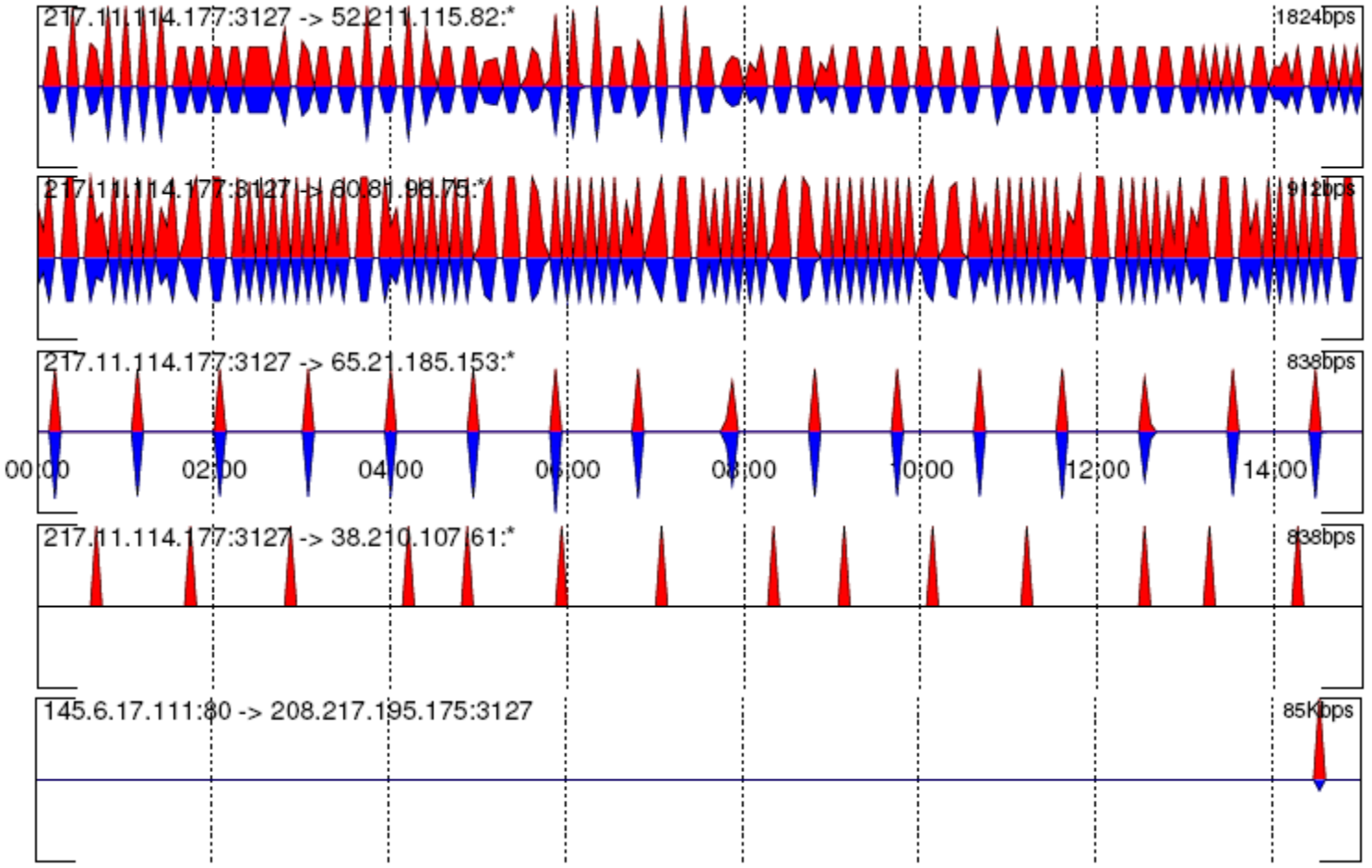
```
rwfilter ./2008* --sport=1024- --dport=1024- --pass=stdout | rwstats --dip --top --count=50
INPUT SIZE: 3242607 records for 1100719 unique keys
DESTINATION IP Key: Top 50 flow counts
            dIP|             Records|%_of_total|   cumul_%|
    163.7.48.83|              106360|  3.280077|  3.280077|
  160.87.172.28|               51999|  1.603617|  4.883694|
 209.22.177.208|               40744|  1.256520|  6.140214|
    95.7.189.72|               35037|  1.080519|  7.220733|
 217.11.114.177|               34087|  1.051222|  8.271955|
208.172.168.147|               33236|  1.024978|  9.296933|
208.172.168.153|               28782|  0.887619| 10.184552|
 192.120.254.26|               25672|  0.791709| 10.976261|
  8.179.206.104|               24500|  0.755565| 11.731826|

rwfilter ./2008* --sport=1024- --dport=1024- --daddress=160.87.172.28 --pass=stdout | rwstats --dport --
    top --count=10
INPUT SIZE: 51999 records for 263 unique keys
DESTINATION PORT Key: Top 10 flow counts
          dPort|             Records|%_of_total|   cumul_%|
           3127|               23650| 45.481644| 45.481644|
           3128|               23034| 44.297006| 89.778650|
           3124|                4399|  8.459778| 98.238428|
          12000|                 144|  0.276928| 98.515356|
           2121|                  75|  0.144234| 98.659590|
           8080|                  45|  0.086540| 98.746130|
           4121|                  37|  0.071155| 98.817285|
```

Worm Traffic

```
rwfilter ./2008* --sport=3127 --dport=1024- --pass=stdout | rwstats --dip --top --count=50
INPUT SIZE: 152318 records for 3667 unique keys
DESTINATION IP Key: Top 50 flow counts
            dIP|             Records|%_of_total|   cumul_%|
  8.179.206.104|               24492| 16.079518| 16.079518|
     95.7.189.72|               17568| 11.533765| 27.613283|
 209.22.177.208|               16331| 10.721648| 38.334931|
    9.9.247.197|                9609|  6.308512| 44.643443|
    73.16.75.75|                4883|  3.205793| 47.849236|
   36.178.31.73|                4104|  2.694363| 50.543600|
 213.85.234.101|                3096|  2.032590| 52.576189|
 217.88.135.184|                2213|  1.452881| 54.029071|
 220.129.136.52|                2136|  1.402329| 55.431400|
```

Worm Traffic



Conclusion

- Saw various malicious traffic within 15 mins intervals.
- Gave a sense of the magnitude of malicious traffic on the Internet.
- Flow analysis provided a better insight to such traffic, in comparison to other methods.
- Need to know what we are looking for.
- Successful implementation requires staffing with right skill set.

Conclusion

- CERT/NetSA is onto something.

Questions?