Triaging "Potential Security Scan" Email Alerts

Security regularly analyzes the netflow data using flow-tools and flow-dscan to find hosts at SLAC that are engaged in suspicious activity such as port scanning, host scanning and flows with unusually high octets or packets. Typical scans are caused by peer to peer (P2P) applications such as Skype. When security discovers a scan that has a potential security problem (e.g. denial of service due to heavy use of a low performance network such as the the visitor network) then an alert email is sent to a group of network and security people at SLAC. This email contains up to 200 related flows. In the case of scanning hosts on the visitor network they are also placed in the "Penalty Box" that reduces the performance of all penalized hosts to share only 56kbps.

I have written two scripts that enable automating of the triage for the emails, and also to analyze many such emails in order to gain event type frequencies. The idea is to see how effective we can be in automatically identifying scan signatures without requiring manual expert attention. Thus we may be able to use less expert help in reviewing the emails and/or may be able to reduce the number of emails. We may also be able to improve the configuration of flow-dscan.

As part of this I also looked in detail at an alert generated when I had a long Skype conversation as part of a meeting with Pakistani collaborators. The analysis of this alert was performed in Excel. This assisted in identifying the characteristics of a Skype signature.

Scan.pl

This analyzes the 200 data lines from a single email. Usage information is available. It makes the following tests:

- Gets the name of the Source host.
- Optionally gets the names of the destination hosts (this is optional since it takes time)
- Checks whether the Source host is a well known host that often shows scans which are not security related (false positives). Examples include the pinger monitoring host, the ssrl host
- Checks which network the source host is on, the visitor subnet, the SLAC internal network, the SLAC VPN etc.
- Keeps count of the numbers of different TCP and UDP ports used.
- Identifies how often the major ports (e.g. SMTP, DNS, eDonkey ...) are used
- Keeps count of the number of times each unique destination IP address is seen.
- Keeps count of the number of broadcast IP addresses (255.255.255.255) seen.
- Analyses the occurrence of consecutively encountered ports
- It reports on:
    - Optionally (option Debug level >=0)the source host name (address), destination host name (address) number of flows and number of packets
        - If this option is taken then it reports the number of times hosts with names taht typically reflect a service are seen (e.g. hostnames begining with www, mx, mail, ntp, smtp, ftp, visserv, or containing comcast, slac, or whose IP address does not resolve to a name)
    - The number of unique destination hosts, unique source ports
    - The IP address of the source and the subnet it is located on (visitor, VPN, SLAC)
    - The protocol, number and name of the source port (if known), together with the number of flows and octets.
    - The number of flows by protocol (TCP, UDP. ICMP, GRE).
        - For ICMP reports the number of port unreachable messages (this is often encountered after a host shuts down Skype, and other hosts still thinks it is a supernode).
    - The number of destination broadcast addresses (if any) excluding those associated with bootp or Office_X
        - Microsoft Office_X for Mac broadcasts to see if any one is already running Office_X.
    - Whether it is a long term DHCP roaming host and if so the relevant information, if accessible
    - The destination port most often encountered
    - Flow size distributions, i.e. the frequencies of singlets (single packet flows), short flows (<=3 packets), messages (4-10 packets and < 2kBytes), file transfers (the rest).
        - Successful TCP connections require >= 4 packets (e.g. initiator to open sends SYN (gets SYN/ACK Back), sends ACK, then to terminate it is a 4 way handshake (2 per flow direction). Thus a large number of short TCP flows is anomalous.
    - The number of quintuplet flows (> 1), that have the same protocol_srcAddr:srcPort,dstAddr:dstPort, together with the number of associated octets and packets.
    - A rough guess at the likely cause of the alert, e.g. Skype, P2P file sharing, AFS, PING
        - Skype is the most likely cause. Its signature is a fixed unrecognized source UDP port with many short flows.
    - The median, 25%, 75%, inter-quartile range, min and max of the octets/packet, and the number of different octets/packet values.
    - The observed conditional port probabilities of the Source and destination ports and the numbers of different ports
    - The elapsed time taken for the script to complete.
    - The timestamp of the first flow reported.

Typical output appears as:

```
195cottrell@pinger:~>bin/scan.pl -f scan/edonkey -D -1
(1/113)Src_port=ICMP_0, occurs=3,
(60/113)Src_port=TCP_4662, occurs=2, P2P:eDonkey:TCP_4662
(63/113)Src_port=TCP_4665, occurs=2, P2P:eDonkey:TCP_4665
(111/113)Src_port=UDP_123, occurs=1, NTP
(107/129)Dst_port=UDP_123, occurs=1, NTP
(121/129)Dst_port=UDP_53, occurs=3, DNS
----------------SUMMARY from scan.pl (debug=-1)-----
Fri Sep  7 17:26:23 2007 scan.pl took 0 secs to analyze 201 lines in scan/edonkey
Timestamp of first data record=0906.09:45:23
Found 110 unique Dst addresses and 113 source ports (TCP=107, UDP=5, ICMP=1, Unk=0)
Flows: TCP=167, udp=29, icmp=3, others=0
TCP flows: short(<=3pkts)=129, message(4-10 pts & < 2kB)=38, file_transfer=0
UDP flows: short(<=3pkts)=25, message(4-10 pts & < 2kB)=4, file_transfer=0
dhcpvisitor218162.slac.stanford.edu(198.129.218.162) on visitor subnet, run penalty.pl -a 198.129.218.162
Likely SKYPE supernode - source port UDP_55747 occurs=21, times out of 201 records (>=60)
Max Dst port use (5 flows) is for TCP_1257
Flows for source port=UDP_55747: Median(oct/pkt)=59, IQR=4, 25%=56, 75%=60, min=48, max=63, #different oct/pkt
vals=21
Src Conditional Run Probability(# unique ports=113) occurs= 0.42,
Dst Conditional Run Probability(# unique ports=129) occurs= 0.01,
```

The hostname option is obtained by not specifying a Debug level (i.e. Debug >=0) where the numbers in parentheses are the host number encountered and the total number of different host addresses.

```
196cottrell@pinger:~>bin/scan.pl -f scan/edonkey
...
(92/110)Src=dhcpvisitor218162.slac.stanford.edu(198.129.218.162),Dst=cable-9-147.cgates.lt(80.240.9.147),
records=2, pkts=4
...
Special Dst hosts by name: web=0, ftp=0, mail=1, comcast=9, visserv=1, slac=0, time=1, blizzard=0,
no_dns_name=13
...
```

The usage information is:

```
Usage:   scan.pl [opts]
        Opts:
        -f scan_file
        -p if set to -1 the do not prompt for stdin input
        -v print this USAGE information
        -D debug_level (default=0), 1..3 give increased debugging output
           (Nb -D -1 does not try to name hosts and does not call whereis or
            or try and get the relevant DHCP records - since need AFS access)

    where:
       scan_file location of scan mail file (default /u/sf/cottrell/scan/icmp)
       debug_level is -1..3, default 0 for interactive, -1 for cronjob
         larger values give more diagnostoc output
         -1 removes the printing of all unique Src names (time consuming
            reverses name resolution)

Purpose:
  The script tries to analyze a scan in terms of number of remote hosts scanning
  Type of scan, use of P2P ports, usage of consecutive ports etc.,
Input:
  A Typical scan file entry appears from /u/sf/cottrell/scan/icmp as:
       Start         SrcIP           SrcP  DstIP           DstP  Prot   Pkts     Octets
        0809.14:14:44 198.129.217.211 0     84.27.199.173   771   ICMP      1         56
Output:

Examples:
 scan.pl -v
 scan.pl -f - #read scan file from STDIN, use ctrl-d to end the input
 scan.pl -f - -D -2
 scan.pl -f scan/icmp -p -1
 scan.pl -f scan/skype

Bugs:
 When calculating the consecutive source port numbers it combines TCO & UDP they should
 be separate.

Version=1.0, 9/8/07, Cottrell
```

On a 3MHz Pentium 4 running Linux it takes about 1/10 second to run if the Debug is not set to enable resolving the destination hostnames, otherwise it takes about 2.5 minutes.

It is currently kept in ~cottrell/scan.pl. It runs successfully on visser1. If run there it cannot provide CANDO information on the source host. Below is how it appears on visserv1.

```
2cottrell@visserv1:~>bin/scan.pl -f scan/bittorrent -D -1
(1/65)Src_port=ICMP_0, occurs=10,
(60/65)Src_port=TCP_6881, occurs=21, P2P:BitTorrent:TCP_6881
(65/65)Src_port=UDP_6881, occurs=84, P2P:BitTorrent:UDP_6881
(18/120)Dst_port=TCP_4662, occurs=1, P2P:eDonkey:TCP_4662
(44/120)Dst_port=TCP_6881, occurs=17, P2P:BitTorrent:TCP_6881
(94/120)Dst_port=UDP_53, occurs=8, DNS
(111/120)Dst_port=UDP_6881, occurs=12, P2P:BitTorrent:UDP_6881
----------------SUMMARY from scan.pl (debug=-1)-----
Fri Sep  7 18:03:36 2007 scan.pl took 0 secs to analyze 201 lines in scan/bittorrent
Timestamp of first data record=0828.10:15:23
Found 123 unique Dst addresses and 65 source ports (TCP=59, UDP=5, ICMP=1, Unk=0)
Flows: TCP=94, udp=95, icmp=10, others=0
TCP flows: short(<=3pkts)=85, message(4-10 pts & < 2kB)=7, file_transfer=2
UDP flows: short(<=3pkts)=90, message(4-10 pts & < 2kB)=4, file_transfer=1
dhcpvisitor216193.slac.stanford.edu(198.129.216.193) on visitor subnet, run penalty.pl -a 198.129.216.193
Likely P2P:BitTorrent:UDP_6881 file sharing used port 84 times (>30) in 201 records
Max Dst port use (17 flows) is for TCP_6881
Flows for source port=UDP_6881: Median(oct/pkt)=119, IQR=20, 25%=99, 75%=119, min=44, max=293, #different oct
/pkt vals=105
Src Conditional Run Probability(# unique ports=65) occurs= 0.28,
```

Scans.pl

This uses scan.pl to analyze multiple potential scan emails that have been saved in a file. Usage information is available. A typical full email will appear as:

```
From:   owner-net-eng@lists1.slac.stanford.edu on behalf of sec-group@slac.stanford.edu
Sent:   Wednesday, August 15, 2007 9:20 AM
To:     sec-group; net-eng
Subject:        Potential scanning host 134.79.80.232

The automated scan-detect tool has detected a potential
scanning host with source ip address 134.79.80.232

No NBTSTAT information returned

The first 200 flows with this source ip address are:

Start         SrcIP           SrcP  DstIP           DstP  Prot   Pkts     Octets
0815.09:00:46 134.79.80.232   0     67.161.10.179   0     ICMP      3        441
0815.09:02:03 134.79.80.232   0     76.21.1.105     0     ICMP      3        420
0815.09:03:18 134.79.80.232   7409  86.156.75.30    52351  TCP      3        120
...
815.09:04:43 134.79.80.232   7409  193.88.8.59     12350  UDP      1         79
0815.09:04:42 134.79.80.232   7409  77.101.59.60    37227  UDP      1         58
```

Scans.pl uses the output from scan.pl to gather statistics on reasons for the emails. A typical call to scan.pl is:

```
scan.pl -f - -D -1 -p -1 < /tmp/scan_file
```

Where -f - means read the input from STDIN (in this case redirected from the file /tmp/scan_file), -p -1 says do not prompt for STDIN , -D says no debugging output and avoid using things that require AFS.

Scans.pl includes counters for the number of emails:

- That use various well known ports used for P2P file sharing applications such as BitTorrent, Gnutella.
- That are likely caused by Skype, AFS, ICMP
- That are caused by well know source hosts
- That include extended runs of consecutive ports.
- And their distribution by day of week

The output appears as:

```
44cottrell@iepm-bw:~>bin/scans.pl -f scan/scans.txt \| more
scans.pl starting on Fri Aug 24 13:52:34 2007(epoch=1187988754) to analyze scan/scans.txt (debug=-1)
Executing /u/sf/cottrell/bin/scan.pl -f - -D -1 -p -1 < /tmp/scan_file for 200 with data lines
Fri Aug 24 13:52:34 2007 (0 secs so far) scans.pl processed (1 mails, 213 lines)
  Sent:  Wednesday, August 15, 2007 9:20 AM
================= scans.pl: Mail(1) Sent:       Wednesday, August 15, 2007 9:20 AM ===============
(1/16)Src_port=ICMP_0, occurs=3,
 (167/176)Dst_port=UDP_7009, occurs=1, AFS
 ----------------SUMMARY (debug=-1)-----------------------------------------------
 Fri Aug 24 13:52:34 2007 scan.pl took 0 secs to analyze 200 lines in -
   finding 161 unique addresses and 16 source ports (TCP=14, UDP=1, ICMP=1, Unk=0)
 ilc-gwhite.slac.stanford.edu(134.79.80.232)CANDO information:
 DEVICE NAME:  ILC-GWHITE
 MODEL:        DELL-PRECISION-390
 OPER. SYSTEM: LINUX
 BUILDING:     213
 ROOM:         8
 DEVICE STATUS:ACTIVE
 PRIMARY USER: WHITE, GLEN RUSSELL
 IP HOSTNAME:  ILC-GWHITE
 SUBNET NAME:  PUB2
 HARDWARE:     00-1A-A0-04-D7-14    FOR IP: 134.79.80.232
 ABOVE ADDRESS LAST SEEN 23-AUG-2007  ON: RTR-CORE1, 9 INTERFACE
 INTERNET:     134.79.80.232
 PC NUMBERS:
 Likely SKYPE supernode - source port UDP_7409 occurs=172, times out of 200 records (>=60)
 Max Dst port use (4 flows) is for TCP_52429
 Flows for source port=UDP_7409: Median(octets/pkt)=60, IQR=8, 25%=55, 75%=63, min(0)=39, max(181)=526,
    #different octets/pkt values=182
 Src Conditional Run Probability(# unique ports=16) occurs= 0.00,
 Dst Conditional Run Probability(# unique ports=176) occurs= 0.00,
>>>>>>>>(mail=1)Signature=SKYPE(172) for mail Sent:     Wednesday, August 15, 2007 9:20 AM
Executing /u/sf/cottrell/bin/scan.pl -f - -D -1 -p -1 < /tmp/scan_file for 200 with data lines
Fri Aug 24 13:52:34 2007 (0 secs so far) scans.pl processed (2 mails, 426 lines)
  Sent:  Wednesday, August 15, 2007 9:20 AM
...
================= scans.pl: Mail(59) Sent:      Thursday, August 09, 2007 11:36 AM ===============
(142/188)Dst_port=UDP_53, occurs=3, DNS
 ----------------SUMMARY (debug=-1)-----------------------------------------------
 Fri Aug 24 13:48:19 2007 scan.pl took 0 secs to analyze 200 lines in -
   finding 191 unique addresses and 9 source ports (TCP=5, UDP=4, ICMP=0, Unk=0)
 dhcpvisitor21799.slac.stanford.edu(198.129.217.99) on visitor subnet, run penalty.pl -a 198.129.217.99
 Likely SKYPE supernode - source port UDP_18352 occurs=191, times out of 200 records (>=60)
 Max Dst port use (4 flows) is for TCP_993
 Flows for source port=UDP_18352: Median(octets/pkt)=65, IQR=80, 25%=60, 75%=140,
   min(0)=46, max(190)=596, #different octets/pkt values=191
 Src Conditional Run Probability(# unique ports=9) occurs= 0.25,
 Dst Conditional Run Probability(# unique ports=188) occurs= 0.00,
Fri Aug 24 13:48:19 2007 (8 secs so far) scans.pl processed (59 mails, 12106 lines)
  Sent:       Thursday, August 09, 2007 11:36 AM
>>>>>>>>(mail=59)Signature=SKYPE(191) for mail Sent:    Thursday, August 09, 2007 11:36 AM
~~~~~~~~~~~~~~~~scans.pl overall summary~~~~~~~~~~~~~~~~~~~~~~
BitTorrent>=20 occurs 2 times in 59 mails
Conditional Run Probability>50% & >100 unique ports  occurs 2 times in 59 mails
GRE occurs 2 times in 59 mails
Gnutella>=20 occurs 1 times in 59 mails
KNOWN occurs 5 times in 59 mails
```

Results

Unable to render {include} The included page could not be found.