Honeynet Challenge #27

Before anything, I downloaded the file sotm27.tgz from the Honeynet Projet Website and check the MD5 signature, which was correct. Then, we could begin trying to answer the questions...

Beginner Questions:

What is IRC?

IRC stands for "Internet Relay Chat". This protocol, defined by the RFC1459, allows clients to connect to interconnected IRC servers, and to send and receive text messages to and from others connected clients, in real time: it's a chat protocol, distributed over multiple servers. Latest developements allow users to send and receive files, with other improvements.

What message is sent by an IRC client when it asks to join an IRC network?

The RFC 1459 states the USER command is used when a client wants to join an IRC network. When a client issue this command, the parameters used are: <username> <realname>. The "username" provides the name the user will be known with by the server, and the "realname" was originally created to keep track of the identity behind the nicknames of users. As explained in the RFC, the user can easily lie on the "realname" parameter, so its TRUC is only practical.

What is a botnet?

A botnet is generally a blackhat-controlled IRC network of interconnected servers and clients, where he can manage the machines he compromised. Actually, the hacker installs an IRC client script on each box he gains access. This script is an IRC controlled trojan, which automatically connects to predefined IRC servers and channels. Then, by issuing commands in the channel or to the main server, the hacker can send specific commands to his connected "bots" (robots).

What are botnets commonly used for?

Botnets are commonly used by blackhats to launch attacks, particularly Distributed Denial of Service attacks, using the numerous hosts he manages through the IRC channel. Depending on the trojan installed on compromised machines, possible attacks range from IRC flood to Teardrop or ICMP DoS.

What TCP ports does IRC generally use?

The IANA official TCP/IP port assignement (http://www.iana.org/assignments/port-numbers) states that ports 194, 529, 994, and the 6665-6669 range, both TCP and UDP, are reserved for IRC use, with different variants. Actually, the 6666 and 6667 ports are commonly used by clients to connect to IRC servers.

What is a binary log file and how is one created?

A binary log file is log file using the pcap format, also known as "tcpdump format". This type of log file allow applications to quickly dump network packets in a single binary file, which can be interpreted and presented in text format by almost all network sniffers like tcpdump, ethereal, or snort.

What IRC servers did the honeypot, which has the IP address 172.16.134.191, communicate with?

We choose to use the Snort package to parse the binary log file, because of its IDS capabilities and ease of use. Issuing the command: "snort -dev -r sotm27 src

172.16.134.191 and dst port 6666-6669" gives us all the packets sent from the honeypot to IRC servers, as we learned above that the IRC server port range goes from 6666-6669. Moreover, we know that in order to connect to a server, a client or another server will issue a USER command, generally after a NICK command. Piping the output of the previous command to a "grep -A 5 -B 5 NICK", we now have all the connection requests issued by the honeypot.
We thus can see three connections to IRC Servers on the Internet:
The first on March 6th, at 5:56:16.794916, to the server 63.241.174.144 (no DNS entry), the honeypot issued the following commands:
NICK eohisou
USER eohisou localhost localhost :eohisou

Then, a few seconds later, at 5:56:37.569359 to the server 217.199.175.10 (ns2.caralarmuk.com):
NICK rgdiuggac
USER rgdiuggac localhost localhost :rgdiuggac

Finally, the same day at 6:23:19.767469 to the server 209.196.44.172 (ipdwbc0271atl2.public.registeredsite.com):
NICK rgdiuggac
USER rgdiuggac localhost localhost :rgdiuggac

Note the random username taken by the server. The very short time between the two first connections let us think the first server closed the connection, whatever the reason. The output of the "snort -dev -r sotm27 src 63.241.174.144 and dst 172.16.134.191" command confirm this, as we can see a closing connection packet, at 5:56:36.406108:

```
63.241.174.144:6667 -> 172.16.134.191:1133 TCP TTL:52 TOS:0x0 ID:29429 IpLen:20
DgmLen:111 DF
***AP*** Seq: 0x9267193E  Ack: 0xDBDA75FE  Win: 0x7B68  TcpLen: 20
45 52 52 4F 52 20 3A 43 6C 6F 73 69 6E 67 20 4C  ERROR :Closing L
69 6E 6B 3A 20 5B 65 6F 68 69 73 6F 75 40 32 35  ink: [eohisou@25
35 2E 32 35 35 2E 32 35 35 2E 32 35 35 5D 20 28  5.255.255.255] (
43 6F 6E 6E 65 63 74 69 6F 6E 20 54 69 6D 65 64  Connection Timed
20 4F 75 74 29 0D 0A                              Out)..
```

The second connection was also quickly closed, with the following packet, but apparently didn't reconnect before the third attempt at 6:23:

```
03/06-05:56:38.001313 0:E0:B6:5:CE:A -> 0:5:69:0:1:E2 type:0x800 len:0x96
217.199.175.10:6667 -> 172.16.134.191:1139 TCP TTL:45 TOS:0x0 ID:43467 IpLen:20
DgmLen:136 DF
***AP*** Seq: 0x97EE9F11  Ack: 0xDC32A128  Win: 0x7D78  TcpLen: 20
45 52 52 4F 52 20 3A 43 6C 6F 73 69 6E 67 20 4C  ERROR :Closing L
69 6E 6B 3A 20 72 67 64 69 75 67 67 61 63 5B 7E  ink: rgdiuggac[~
72 67 64 69 75 67 67 61 63 40 32 35 35 2E 32 35  rgdiuggac@255.25
35 2E 32 35 35 2E 32 35 35 5D 20 28 53 6F 72 72  5.255.255] (Sorr
79 2C 20 73 65 72 76 65 72 20 69 73 20 66 75 6C  y, server is ful
6C 20 2D 20 74 72 79 20 6C 61 74 65 72 29 0D 0A  l - try later)..
```

During the observation period, how many distinct hosts accessed the botnet associated with the server having IP address 209.196.44.172?

Each time a host accessed the botnet, it tried to join a channel, apparently named "#x.......x", from which the black hat seems to operate. At this time, the server 209.196.44.172 issued a message to the others servers of the botnet, including the honeypot, to warn them that a new user joined the channel. The command used to perform this is the JOIN command. By typing the following on our

lab box: "snort -dev -r sotm27 src 209.196.44.172 and dst 172.16.134.191 | grep -A 5 -B 5 JOIN" we can see packets sent by the main IRC server of the botnet to inform the other servers that a new user has joined the channel. Replacing the "grep -A 5 -B 5 JOIN" by a "grep -c JOIN" gives us the count of matching line: the result is 7135.

Assuming that each botnet host has a 56 kbps network link, what is the aggregate bandwidth of the botnet?

It's tempting to do a simple 56x7135, but it would be incomplete: we forgot the already connected clients. This number is given by one of the first packet sent by the 209.196.44.172 machine upon connection of the honeypot to the botnet.

```
03/06-06:23:19.898919 0:E0:B6:5:CE:A -> 0:5:69:0:1:E2 type:0x800 len:0x436
209.196.44.172:6667 -> 172.16.134.191:1152 TCP TTL:52 TOS:0x0 ID:5886 IpLen:20
DgmLen:1064 DF
***AP*** Seq: 0xFE4C0163  Ack: 0xF544C60B  Win: 0x7D78  TcpLen: 20
3A 69 72 63 35 2E 61 6F 6C 2E 63 6F 6D 20 30 30  :irc5.aol.com 00
31 20 72 67 64 69 75 67 67 61 63 20 3A 57 65 6C  1 rgdiuggac :Wel
63 6F 6D 65 20 74 6F 20 74 68 65 20 49 6E 74 65  come to the Inte
72 6E 65 74 20 52 65 6C 61 79 20 4E 65 74 77 6F  rnet Relay Netwo
72 6B 20 72 67 64 69 75 67 67 61 63 0D 0A 3A 69  rk rgdiuggac..:i
72 63 35 2E 61 6F 6C 2E 63 6F 6D 20 30 30 32 20  rc5.aol.com 002
72 67 64 69 75 67 67 61 63 20 3A 59 6F 75 72 20  rgdiuggac :Your
68 6F 73 74 20 69 73 20 69 72 63 35 2E 61 6F 6C  host is irc5.aol
2E 63 6F 6D 5B 69 72 63 35 2E 61 6F 6C 2E 63 6F  .com[irc5.aol.co
6D 2F 36 36 36 37 5D 2C 20 72 75 6E 6E 69 6E 67  m/6667], running
20 76 65 72 73 69 6F 6E 20 32 2E 38 2F 68 79 62   version 2.8/hyb
72 69 64 2D 36 2E 33 2E 31 0D 0A 4E 4F 54 49 43  rid-6.3.1..NOTIC
45 20 72 67 64 69 75 67 67 61 63 20 3A 2A 2A 2A  E rgdiuggac :***
20 59 6F 75 72 20 68 6F 73 74 20 69 73 20 69 72   Your host is ir
63 35 2E 61 6F 6C 2E 63 6F 6D 5B 69 72 63 35 2E  c5.aol.com[irc5.
61 6F 6C 2E 63 6F 6D 2F 36 36 36 37 5D 2C 20 72  aol.com/6667], r
75 6E 6E 69 6E 67 20 76 65 72 73 69 6F 6E 20 32  unning version 2
2E 38 2F 68 79 62 72 69 64 2D 36 2E 33 2E 31 0D  .8/hybrid-6.3.1.
0A 3A 69 72 63 35 2E 61 6F 6C 2E 63 6F 6D 20 30  .:irc5.aol.com 0
30 33 20 72 67 64 69 75 67 67 61 63 20 3A 54 68  03 rgdiuggac :Th
69 73 20 73 65 72 76 65 72 20 77 61 73 20 63 72  is server was cr
65 61 74 65 64 20 53 75 6E 20 4A 61 6E 20 31 39  eated Sun Jan 19
20 32 30 30 33 20 61 74 20 31 39 3A 30 34 3A 30   2003 at 19:04:0
33 20 50 53 54 0D 0A 3A 69 72 63 35 2E 61 6F 6C  3 PST..:irc5.aol
2E 63 6F 6D 20 30 30 34 20 72 67 64 69 75 67 67  .com 004 rgdiugg
61 63 20 69 72 63 35 2E 61 6F 6C 2E 63 6F 6D 20  ac irc5.aol.com
32 2E 38 2F 68 79 62 72 69 64 2D 36 2E 33 2E 31  2.8/hybrid-6.3.1
20 6F 4F 69 77 73 7A 63 72 6B 66 79 64 6E 78 62   oOiwszcrkfydnxb
20 62 69 6B 6C 6D 6E 6F 70 73 74 76 65 0D 0A 3A   biklmnopstve..:
69 72 63 35 2E 61 6F 6C 2E 63 6F 6D 20 30 30 35  irc5.aol.com 005
20 72 67 64 69 75 67 67 61 63 20 57 41 4C 4C 43   rgdiuggac WALLC
48 4F 50 53 20 50 52 45 46 49 58 3D 28 6F 76 29  HOPS PREFIX=(ov)
40 2B 20 43 48 41 4E 54 59 50 45 53 3D 23 26 20  @+ CHANTYPES=#&
4D 41 58 43 48 41 4E 4E 45 4C 53 3D 32 30 20 4D  MAXCHANNELS=20 M
41 58 42 41 4E 53 3D 32 35 20 4E 49 43 4B 4C 45  AXBANS=25 NICKLE
4E 3D 39 20 54 4F 50 49 43 4C 45 4E 3D 31 32 30  N=9 TOPICLEN=120
20 4B 49 43 4B 4C 45 4E 3D 39 30 20 4E 45 54 57   KICKLEN=90 NETW
4F 52 4B 3D 58 4E 65 74 20 43 48 41 4E 4D 4F 44  ORK=XNet CHANMOD
45 53 3D 62 65 2C 6B 2C 6C 2C 69 6D 6E 70 73 74  ES=be,k,l,imnpst
20 45 58 43 45 50 54 53 20 4B 4E 4F 43 4B 20 4D   EXCEPTS KNOCK M
4F 44 45 53 3D 34 20 3A 61 72 65 20 73 75 70 70  ODES=4 :are supp
6F 72 74 65 64 20 62 79 20 74 68 69 73 20 73 65  orted by this se
72 76 65 72 0D 0A 3A 69 72 63 35 2E 61 6F 6C 2E  rver..:irc5.aol.
```

```
63 6F 6D 20 32 35 31 20 72 67 64 69 75 67 67 61  com 251 rgdiugga
63 20 3A 54 68 65 72 65 20 61 72 65 20 30 20 75  c :There are 0 u
73 65 72 73 20 61 6E 64 20 34 37 35 32 20 69 6E  sers and 4752 in
76 69 73 69 62 6C 65 20 6F 6E 20 34 20 73 65 72  visible on 4 ser
76 65 72 73 0D 0A 3A 69 72 63 35 2E 61 6F 6C 2E  vers..:irc5.aol.
63 6F 6D 20 32 35 32 20 72 67 64 69 75 67 67 61  com 252 rgdiugga
63 20 31 20 3A 49 52 43 20 4F 70 65 72 61 74 6F  c 1 :IRC Operato
72 73 20 6F 6E 6C 69 6E 65 0D 0A 3A 69 72 63 35  rs online..:irc5
2E 61 6F 6C 2E 63 6F 6D 20 32 35 34 20 72 67 64  .aol.com 254 rgd
69 75 67 67 61 63 20 34 20 3A 63 68 61 6E 6E 65  iuggac 4 :channe
6C 73 20 66 6F 72 6D 65 64 0D 0A 3A 69 72 63 35  ls formed..:irc5
2E 61 6F 6C 2E 63 6F 6D 20 32 35 35 20 72 67 64  .aol.com 255 rgd
69 75 67 67 61 63 20 3A 49 20 68 61 76 65 20 33  iuggac :I have 3
34 36 20 63 6C 69 65 6E 74 73 20 61 6E 64 20 31  46 clients and 1
20 73 65 72 76 65 72 73 0D 0A 3A 69 72 63 35 2E   servers..:irc5.
61 6F 6C 2E 63 6F 6D 20 32 36 35 20 72 67 64 69  aol.com 265 rgdi
75 67 67 61 63 20 3A 43 75 72 72 65 6E 74 20 6C  uggac :Current l
6F 63 61 6C 20 20 75 73 65 72 73 3A 20 33 34 36  ocal  users: 346
20 20 4D 61 78 3A 20 33 34 38 0D 0A 3A 69 72 63    Max: 348..:irc
35 2E 61 6F 6C 2E 63 6F 6D 20 32 36 36 20 72 67  5.aol.com 266 rg
64 69 75 67 67 61 63 20 3A 43 75 72 72 65 6E 74  diuggac :Current
```

We then see that 4752 clients are already connected at this time. Adding the 7135 ones which joined later, we find a number of 11887 hosts connected. 11887x56kbps=665,7Mbps (665.672kbps), or roughly 83MB/s. This is of course a very huge potential bandwidth, especially if used to flood a single server on the Internet. This clearly shows how this Distributed Denial of Service attacks can bring down even servers with a very high bandwidth connection.
If we do the same with the number provided in the presentation page (15,164), we then find an aggregate bandwidth of 106MBps. No comment.

Intermediate Questions

What IP source addresses were used in attacking the honeypot?

As the honeypot is not referenced in any way on the Internet, we can assume that every connection made to the box is suspicious. Thus, a tcpdump filter which search for packets with only the SYN bit checked could give us all the TCP connection attempts, from attackers, worms... Thus the command "snort -dev -r sotm27 tcp[13] == 2 and dst 172.16.134.191" give us the result. We need to filter this file in order to clean the multiple connections attempts from single hosts. Ideally, (= if I had the time to do so :), i would have wrote a script which would have count the number of distinct hosts. The reality is far less glorious: i used Excel and "grep -v" to manually clean the results, and I finally come to a number of 78 IP addresses, which had tried to connect to the honeypot. Obviously, this result doesn't take into account the UDP connections attempts. There are a lot of UDP connection, especially in the beginning of the log file, apparently caused by the propagation of the recent MSSQL Worm. Nevertheless, I didn't noticed any possible human activity using this protocol in the log file.

What vulnerabilities did attackers attempt to exploit?

Ouch! Friday the 25th? I desperatly need time!... By watching the file created from the output of the "snort -dev -r sotm27 tcp[13] == 2 and dst 172.16.134.191" command, we can quickly see the attacked ports. Using Excel, again, we can see the following ports accessed:
21
57

80
111
135
139
445
1080
1433
I voluntarly not mentionned the whole ports scanned by the IP address 24.197.194.106. The most important part is taken by attacks on port 139, and then on port 445, and after port 1433. We can say (without any evidence) the top vulnerabilities attempted by attackers to be exploited were Netbios and MS-SQL ones. Sorry for the lack of details, I'll do better next time :)

Which attacks were successful?

As far as I know, only the attack coming from the IP address 210.22.204.101 was successful. The attacker apparently used a method similar to a worm recently discovered, using the SMB protocol. It first connect to the 445 TCP port, and then try to connect to the victim host IPC$. The Ethereal output shows numerous "SMB Pipe TransactNmPipe Request", but I haven't found the time to learn more about SMB, thus I don't really know what's happening...:(The worm I referred earlier tries different passwords in order to successfully connect on the machine. I suspect the attacker to have done the same thing, given the administrator password was blank. I also saw attempts from the same attacker to exploit the IIS IDA vulnerability using different scans, nevertheless I couldn't find the successful attack that gave the attacker control of the box, but I later found using the "strings" command on the binary log file, some unusual strings in a network capture: "GetProcA", ".dll", "This program cannot be run in DOS mode". I tried to found this strings using snort and grep, and it seems the attacker succeeded in installing a remote control tool on the box. I didn't manage to correlate the IRC server stuff with this attack...

What did you learn about analysis as a result of studying this scan?

First, it's very time consuming!:) Seriously, I really wanted to do such a thing, and I'm very happy to had this opportunity. I learned not to focus on only one tool to do my analysis, and to firstly try to have a global view of the network capture and what happened, before digging to much one single event. I also learned to have a toolkit ready, and not begin downloading tools once the challenge is started :)

How do you anticipate being able to apply your new knowledge and skills?

... during the next Scan Of The Month challenge ! I think I need a lot more exercises in order to begin thinking using my skills in a professional way :)

How can we improve the SotM challenge? What would you like to see added? What would you like to see done differently?

I think the whole challenge is very well done .. Or perhaps you could improve the SotM challenge by asking my boss to give me some time to do it :)

E.Marchand