blog.erratasec.com

We scanned the Internet for port 22

5-6 minutes

Yesterday (Sept. 12) we scanned the entire Internet for port 22 -- the port reserved for "SSH", the protocol used by sysadmins to remotely log into machines. Unlike our normal scans of port 80 or 443, this generated a lot more "abuse" complaints, so I thought I'd explain the scan.

Firstly, we'll happily add you to our "blacklist", so that we won't scan you ever again (barring accidents on our part). Our current blacklist is hundreds of entries long. However, please consider adding our scanner (71.6.151.167) to your "whitelist". We are well-known cyber-sec researchers, we aren't trying anything nefarious or evil, and we are being as transparent as possible about our scans.

Our scanner was just checking banners. It didn't complete the connection, nor did it try any passwords. Several abuse complaints assumed that we were trying to "login", but we weren't. Yes, hackers are constantly trying to login into SSH servers, so it's a good assumption to make, it's just that in this case, it doesn't apply to us.

Here are the top 20 "unique" banners that we got back:

```
 1730887 SSH-2.0-OpenSSH_4.3
```

```
1562709 SSH-2.0-OpenSSH_5.3
1067097 SSH-2.0-dropbear_0.46
 824377 SSH-2.0-dropbear_0.51
 483318 SSH-2.0-dropbear_0.52
 348878 SSH-2.0-OpenSSH_5.9p1 Debian-5ubuntu1
 327841 SSH-1.99-Cisco-1.25
 320539 SSH-2.0-OpenSSH_5.5p1 Debian-6+squeeze3
 318279 SSH-2.0-OpenSSH_5.9p1 Debian-5ubuntu1.1
 307028 SSH-2.0-ROSSSH
 271614 SSH-2.0-OpenSSH_5.5p1 Debian-6+squeeze2
 233842 SSH-2.0-OpenSSH_5.1p1 Debian-5
 225095 SSH-2.0-OpenSSH_5.1
 224991 SSH-2.0-OpenSSH_5.1p1 FreeBSD-20080901
 213201 SSH-2.0-OpenSSH_4.7
 209023 SSH-2.0-OpenSSH_6.0p1 Debian-4
 195977 SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu7
 140809 SSH-2.0-dropbear_0.50
 135821 SSH-2.0-OpenSSH
 132351 SSH-2.0-Cisco-1.25
```

In other words, the top result of 1,730,887 systems on the Internet show an SSH banner of "SSH-2.0-OpenSSH_4.3". (Note: this is actually only 60% of the Internet, I've got corruption in the files for 40% of the results that I need to fix).

Note that these counts are a bit off. Some networks have a router that forwards all connections of a certain port to a single machine. Maybe "OpenSSH_4.3" is most popular banner, or maybe the national ISP of Elbonia just reroutes all port 22 requests. It takes a lot of manual investigation of the results to figure stuff out. I'm just showing raw results above so that people get a sense of why we are scanning.

We'll be scanning SSH again in October. This time, we'll complete more of the SSH connection in order to grab the public keys, in an effort to see how many people use "weak" keys or "duplicate" keys. Again, this isn't an attempt to hack the systems, but to do research and produce results like those above. We won't be trying to log in.

The source code we used for the scan is at https://github.com/robertdavidgraham/masscan/releases/tag/v1. As you can see from the source code, there's no ability to complete the SSH connection and login. It compiles and runs on Mac/Win/Linux, so it'd be a useful tool to run within your own private network.

A common question in the abuse complaints was of the form "Why did you target my network?". The answer is that we targeted *everyone*, the range 0.0.0.0/0. We throttle the scanner to only about 100,000 packets/second, and it takes about 10 hours to complete. We actually only hit 3.5 billion addresses, the remaining 800,000,000 addresses in the 32-bit address space are blacklisted.

Right now, we regularly scan port 80 and 443. Curiously, we don't get abuse complaints for those ports like we do for port 22. Even automated systems don't bother generating complaints for those ports.

We are going to be extending this to more ports, such as FTP and SMTP. Soon, we should have weekly scans going for about 10 ports. I'm moving slowly forward to resolve abuse complaints, like this one generated for port 22. We plan on publishing the results, such as the anonymous counts above, in a nice weekly report for the public.

Finally, the scanner will actually do 10-million packets/second. We are currently running only at 1% maximum capacity. If you've got a fast network, and can deal with the fact you'll get about 20 abuse complaints, we'd love to try a scan from your network at that speed :). Remember, it randomizes the targets, so it never hits any destination network very hard.

If you have more questions, leave a comment below, or contact me via my twitter handle @ErrataRob.

Update: by the way, we got 58 abuse complaints, mostly automated. We replied to each one. We got 4 replies to our replies asking us exclude their ranges from future scans (which we'll do), and a few replies from universities that they'll add us to their "whitelist", which I presume means that they won't allow us through their firewall, but that'll stop triggers from the drops.