Denial of Service Attack Aftermath

(and what did Iran have to do with it?)

On Monday (Jan 4th), early afternoon U.S. East Coast time, isc.sans.org was hit by a small DoS attack. Luckily, it was easily blocked, but it was enough to be noticed. This is a summary of what was found investigating the attack. Thanks to the system administrators who helped investigate this attack. It was amazing how fast and comprehensive some of them responded to assist. The assisting ISPs and system administrator will not be named to prevent retaliatory attacks against them. This report is preliminary. At this point of the investigation there are enough details to make a summary worthwhile.

I Preparation

A number of systems (shell scripts, external services, internal systems like nagios) are used to monitor the health of the affected systems. Most things that can go wrong are monitored. Some aspects are monitored twice. In addition, a few scripts are maintained to quickly summarize and analyze logs. For example, one of these scripts summarizes the most popular user agent strings for the last few thousand access log entries. Another script is used to summarize the top referrers.

II Identification

As typical for a DoS attack, the first indication of a problem was an SMS message indicating a problem with the ISC web server. The web page responded slowly or not at all and this was confirmed with by accessing it with a browser manually.

Ssh did not respond at all initially but later responded sluggishly. The NOC did not report any network problems. Other systems on the same network responded fine. The database received a flood of inbound connections from the web server.

At this point two possibilities were considered:

1. A disk on the web server filled up, and caused the server to hang. This happened a couple months ago and the monitoring script didn't work at

the time. The monitoring script should be fixed now, but there is always a chance that the fix didn't work.

2. An intentional or accidental denial of service attack. It can happen that a site is flooded due to being featured on a popular site like Slashdot or Digg. Isc.sans.org is built to withstand these volumes and the last couple times it happened, did not have a problem.

The next priority was to get access to the system. After a few minutes, the ssh server responded, but sluggishly.

The system admin at the NOC was also able to log in first and noticed a system load in excess of 700, caused by a large number of httpd processes. The first recommendation was to turn off the web server to free up the system so we can investigate. At the same time, a “tail” of the web server’s access log showed that the "search.html" page was being flooded. As a quick test, the search.html page was removed and replaced with a static copy of the index page. The system load dropped immediately. The system started to respond again, and a more detailed analysis of the logs was now possible. After a few minutes, the load was back to normal.

III Containment

I looked at the search.html queries in more detail. I found that a large number of them searched for the same string, and they all used the user agent string, even though they came from very different source IPs. I used a 'grep' to filter out the logs and wrote them to a separate file to make it easier to work with. Next, I extracted the unique IP addresses. Luckily, I identified only about 40 different unique IPs. A quick shell script, and I had them all blocked via iptables. At this point I was able to move search.html back into place. I still watched the logs, and noticed that a few hits still came in via IPv6. So I repeated the same script for the IPv6 addresses using ip6tables.

In order to preserve some evidence, I extracted relevant logs and moved them to my home system for additional analysis. Unlike for a system compromise, a DoS attack has the advantage that the system integrity is not affected by the attack.

IV Eradication

This is a bit difficult with a denial of service attack It is not possible to "patch" a system against a flood of packets. But there are two things that can be done:

- Investigate steps to harden our system to be able to absorb more requests.
- Investigate the attack to trace it back to its source, and eliminate the source.

The second part is more interesting and will be discussed here in more detail.

Among the about 40 attacking IPs addresses, one stood out. It only sent a single request. That particular request was the first request of the attack. The IP address was part of AS48787 (95.82.48.0/21). According to RIPE “whois” data, it is located in Tehran, Iran. Did the attack originate in Iran? The corresponding log entry is shown below for further discussion. Note that logs are “wrapped” and some details may have been removed to make them more readable.

```
95.82.x.y  [04/Jan/2010:19:41:43 +0000]
"GET /search.html?cx=010041889075795008512%3Apdyz_nfupma&
     cof=FORID%3A10&q=dool&auto=y HTTP/1.1"
"Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Trident/4.0)"
__utma=211864204.2053237371.1199386868.1199386868.119938686
8.1;
__utmb=211864204;
__utmc=211864204;
__utmz=211864204.1199386868.1.1.utmccn=(direct)|utmcsr=(dir
ect)|utmcmd=(none)"
```

To compare, here is the first entry that is attributed to the actual “bot net”:

```
209.217.203.201 - - [04/Jan/2010:19:43:47 +0000]
   "GET
/search.html?cx=010041889075795008512:pdyz_nfupma&cof=FORID
```

```
:10&q=dool&auto=y HTTP/1.1"
"http://isc.sans.org/search.html?cx=010041889075795008512:p
dyz_nfupma&cof=FORID:10&q=dool&auto=y"
   "Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US;
rv:1.9.1.2) Gecko/20090729 Firefox/3.5.2 (.NET CLR
3.5.30729)"
```

There are a couple of notable differences:

1. the first request sent cookies! So this appears to be more like a normal browser that visited the site before.
2. the user agent is different.
3. the referrer is different

The '95.82.x.y' request appears to be triggered manually. The other request uses some kind of simple tool (more about that later). While there is no hard evidence, it looks like the user agent is faked for the first request as well. There is no .Net add on in the user agent which is unusual but possible for a Windows system.

The first request from 95.82.x.y came in a couple minutes earlier:

```
95.82.x.y - - [04/Jan/2010:19:40:44 +0000] "GET
/diary.html?storyid=6601 HTTP/1.1" 200 11854 "-"
"Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Trident/4.0)" "-"
```

Interestingly, the story that this user looked at talked about an Apache DoS tool. This diary was first published last June, and it receives about one hit each hour these days. Interesting that there was a second hit just before "Iran" (95.82.x.y) hit the page:

```
77.100.x.y - - [04/Jan/2010:19:40:18 +0000] "GET
/diary.html?storyid=6601 HTTP/1.1"
200 12253 "-" "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-
US; rv:1.9.1.6) Gecko/20091201 Firefox/3.5.6 (.NET CLR
3.5.30729)"
"__utma=211864204.1647979335.1261969153.1261969153.12619691
53.1;
__utmz=211864204.1261969153.1.1.utmccn=(referral)|utmcsr=ds
lreports.com|
 utmcct=/forum/r23550070-Microsoft-IIS-0Day-Vulnerability-
in-Parsing-Files|utmcmd=referral"
```

77.100.x.y is an IP assigned to Virginmedia in the UK, a provider for residential cable modem service. It requested the same page, 26 seconds earlier. Google is nice enough to remind us via its tracking cookie that this user originally came via a link from DSL Reports. It looks like this person was looking for more information on the recent IIS 0 Day Vulnerability.

The IP address reverse resolves to *.perr.cable.virginmedia.com . Virignmedia appears to use four letter codes for service areas (e.g. "nott" for Nottingham). It is not clear what area “perr” stands for. The original access to the page via DSL Reports occurred on December 28th.

```
77.100.x.y - - [28/Dec/2009:02:59:20 +0000] "GET
/diary.html?storyid=7816 HTTP/1.1"
"http://www.dslreports.com/forum/r23550070-Microsoft-IIS-
0Day-Vulnerability-in-Parsing-Files"
"Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US;
rv:1.9.1.6) Gecko/20091201 Firefox/3.5.6 (.NET CLR
3.5.30729)" "-"
```

This is the original hit from this particular user coming to us from DSL Reports. I found other hits from this IP address, reaching back to November:

```
77.100.x.y - - [06/Nov/2009:18:16:50 +0000] "GET
/diary.html?storyid=7141 HTTP/1.1"
"http://www.google.co.uk/url?sa=t&source=web&ct=res&cd=1&ve
d=0CAcQFjAA&url=http%3A%2F%2Fisc.sans.org
%2Fdiary.html%3Fstoryid%3D7141&rct=j&q=smb2+exploit&ei=AWj0
SqbxNcWk4Qbvx5ncAw&usg=
AFQjCNEWcCdSfu-mXlYqSOhACAIQDm7EbA"
"Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US;
rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5 (.NET CLR
3.5.30729)"
__utma=211864204.410146471.1257531392.1257531392.1257531392
.1; __utmb=211864204; __utmc=211864204; __utmz=21186420
4.1257531392.1.1.utmccn=(organic)|utmcsr=google|utmctr=smb2
+exploit|utmcmd=organic"
```

There are two reasons to believe that this is the same individual:

- the search is for a simple script kiddy exploit issue
- the browser uses "en-US" as language, not "en-UK" as you would expect for most UK users

Which gets us back to the real question: Who is this person?

Interestingly, a Google search for the IP address leads to weblogs from a Persian blog. The Google translation of the blog is not very clear. It appears to be benign (not a hacker or political blog). The blogs with links to this IP address are: sherry.persianblog.ir, babosh.persianblog.ir, sherry.persianblog.ir and sayehayehtardid.persianblog.ir . Any help from someone with more language skills would be appreciated.

Initially about ten different abuse contacts and technical contacts (based on “whois” data) were contacted. Four of them responded with personal messages offering assistance. This response rate is very much appreciated, and some of the responses led to valuable input. Two identified the script responsible for the attack. One offered remote access to the system.

It appears that the systems attacking isc.sans.org were Windows servers. Everything from Windows 2000 up to 2008 (the single affected 2008 server appeared to be clean to the investigating administrator). These systems where part of other DoS attacks as well. It appears that the script was uploaded via unprotected FTP accounts. The script itself is a rather simple ASP script. There is no "channel" the script connects to. Instead, the script waits for commands to be delivered to it via HTTP GET requests.

Here is a typical GET request sent to an infected server:

```
/Bot.aspx
Job=MyDDOS&U=http://isc.sans.org/search.htmlw.w.h.h.a.a.t.t
cx=010041889075795008512%3A
pdyz_nfupmaa.a.n.n.d.dcof=FORID%3A10a.a.n.n.d.dq=doola.a.n.
n.d.dauto=y&Q=10000&T=10&C=
```

The system used to control the attack (which sent these GET requests) was in 165.252.77.x . Sadly, nobody from that network responded yet. This "bot net" was used to attack a number of other sites. The actual "Bot.aspx" file may be discussed in more detail in another report. It is very simple and doesn't really qualify as a bot. Maybe it should be called an HTTP backdoor?

V Recovery

Once the root cause was identified in the hits against search.html, recovery was pretty quick. Removing search.html gave back the breathing room to look at things in more detail. Ultimately, blocking the small number of attacking IP addresses was the solution to get back to normal. "search.html" is now back and the IPs are still blocked at the firewall in case the culprit decides to come back using the remnants of this network.

For the hosts participating in the attack, recovery will be a bit more difficult. At the very least, the Bot.aspx file needs to be removed. But then the actual cause (weak FTP passwords?) needs to be fixed and there are possibly other scripts that got uploaded. In one case a process called “dns.exe” was running as 'SYSTEM' that appeared to be linked to the attack (per results from netstat and process viewer). So far the binary has not been sent to us for analysis.

VI Lessons Learned

Bash scripting rules! All this analysis was done with simple grep/cut/sort/uniq commands, including applying the iptables commands to block the attack. Anything else would have added too much additional load to the system. Gnuplot was used to visualize the attack volume (see Appendix) but only much after the fact for the purpose of this report.

It was worthwhile to contact the sources of the attack. Initially we picked smaller companies / ISPs out of the list and that choice proved to be right. We got a lot of help and that help is highly appreciated. The attack was not large enough to request filtering from our own ISP.

We need to look into better ways to limit requests to the server. There is some application logic right now to prevent load issues, but most of them are put in place to prevent accidental DoS conditions or data harvesting. I am looking at some Apache modules right now to see which will work best.

For additional information, please contact Johannes Ullrich, jullrich@sans.edu, or submit information via https://isc.sans.org/contact.html.

Special thanks to David Goldsmith from the SANS NOC team for his assistance during the attack and proofreading this report. Thanks to Rick Wanner for valuable corrections.

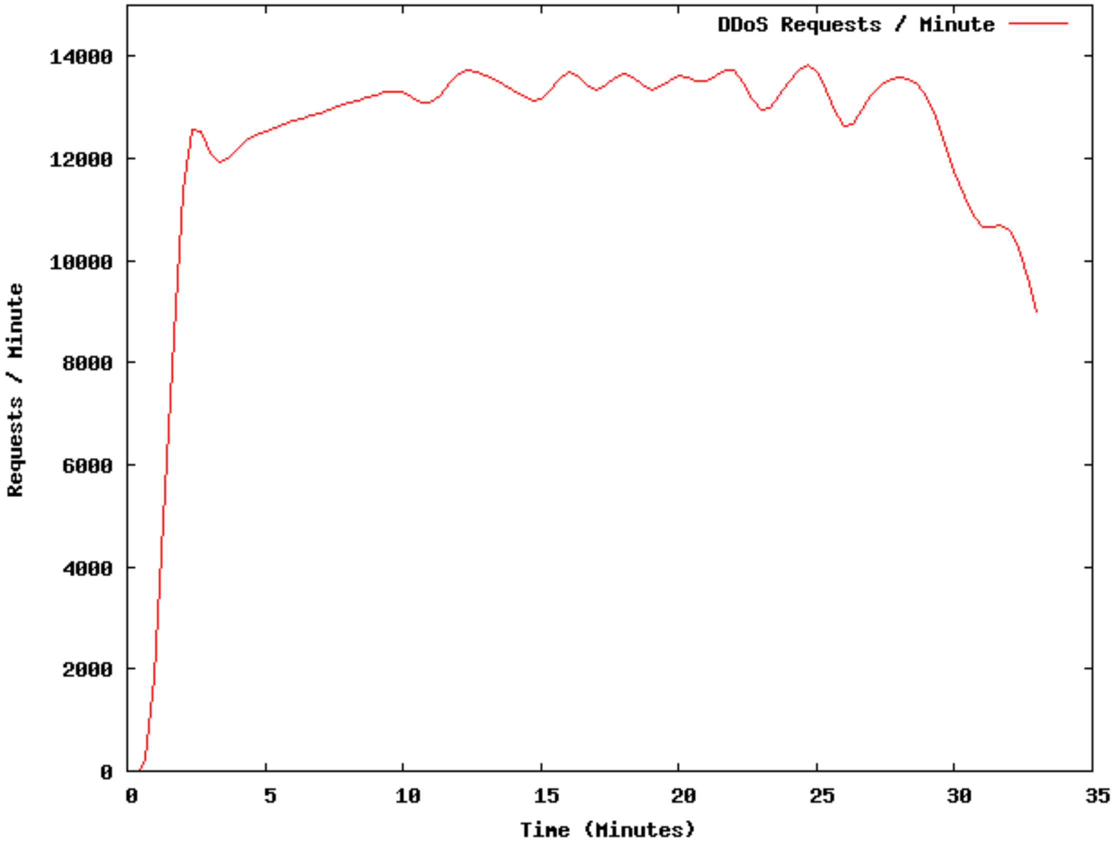
Appendix: IPv6

isc.sans.org is reachable via IPv6. Four of the participating IP addresses used IPv6. All of them used 6-over-4 tunnels. The IPv6 address is derived from the IPv4 address in this case using this scheme:

2002:IPv4-Address::IPv4Address.

For example, if your IPv4 address is 10.1.2.3, your IPv6 address would be 2002:0a01:0203::0a01:0203. If contacting the owner of such a system for help, it is usually best to include the IPv4 address, as the owner may not be aware of the fact that the system communicates via IPv6. These IPv6 tunnels can be configured automatically for systems with routable IPv4 address (the above example would not work, as 10/8 is not routable).

Appendix: Attack Traffic Over time


DDoS Requests / Minute
Requests / Minute
Time (Minutes)