All.Net Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

Dem bugs Dem bugs Dey gonna rise again

This is the story of a minor incident in my internal network. For those of you who maintain networks on a regular basis, you will recognize a lot of the sorts of things involved, but for those who are new to this or whatever, I hope you enjoy the story.

It started with a Web page reload

I am debugging a server problem that is causing Apple Safari to not display the identical javascript portion of a Web page that it displays on another server. Every other browser I have tried works of course, but that's the world of the walled garden.

All of a sudden, a reload gave a bad certificate from the browser. That's strange, so I tried reloading the page and it worked – no problem. Then it happened again. And again. But only sporadically. I though perhaps it was the Web server having a process problem, but I checked, rebooted, etc. and found that it was not that.

So in the debugging process, I checked the DNS and found that the IP address of the server was changing every once in a while and then changing back. I checked the authoritative DNS servers, but they all looked right. Now it turns out this was happening for multiple domains sharing the same server – but not happening for other servers at other IP addresses. As it turns out, the timeout on these domains is only about 1 minute to allow things to be changed rapidly for occasions like this when I need to switch something over in a hurry.

At some point, I got to the point where I was trying to debug the DNS lookup process on my computer, so I checked other computers in my infrastructure and they had the same problem. Naturally, I then went to a sever at AWS and did the lookups from there, and they were stable and not changing the IP address of the server. So now I knew that it was something in my infrastructure that was going awry.

Debug debug debug and then I looked at the bad certificate that was showing up and lo and behold, I found that it was pointing the SSL to a certificate not from the actual site, but served by some server at ...

AHAH!!! COMCAST!!!

So Comcast was taking over my sessions and pointing the DNS to their own server for an apparent man in the middle attack including a redirect to one of their servers.

As it turns out, I had just (finally) left Comcast as an ISP…

Now the process of leaving Comcast is insane and waiting for a class action law suit. You have to request departure then they hassle you for a half hour then you have to full out a form that you can only get to during a weekday office hours, they they try to upsell you,etc.

I have redundant Internet – Comcase was one service, AT&T remains another, then there is my cellular provider (via Google Fi) which allows tethering and supports multiple carriers. So that's something like 5 alternatives – which I was cutting down to 4. I stopped paying Comcast a month ago, and it took them that long to shut the service down. But of course they didn't.

So disconnect Comcast – right?

So once I figured out that Comcast was interfering, I tried to figure out why that was happening. I knew it was DNS related, so I disconnected the router to Comcast from my side (I don't use theirs), and that would seem to eliminate the problem, except of course it didn't! So now I am not connected to Comcast anymore but I am still somehow sometimes using their DNS servers which are redirecting my SSL sessions.

So now I thought it might be my wireless and NAT gateways holding a cache or some such thing, so I rebooted (and upgraded) the routers and gateways and verified that they were no longer using Comcast DNS servers or otherwise doing anything that would create this problem. All to no effect. So I cleared a bunch of caches to make sure, and no effect.

OR IS IT AMAZON!

So now I dug in and started watching where my DNS lookups were going. Lo and behold, as the DNS rotated through connections, it went periodically to the WiFi EERO network I run. Of course this is all controlled by Amazon, so it won't run unless it talks to Amazon all the time. And there I found that, after taking Comcast offline and switching the cable of that portion of the network over to the AT&T feed, that it was still somehow looking things up at Comcast!

So I tested this out by changing the wireless over to other internal wireless networks and found that this was likely the source of the problem. Of course I had to go in and redo the EERO wireless network, and finally the problem stopped.

And then there was the ssh problem

It was now about 45 minutes later, yes it only took 45 minutes to do all this, and my network was again running (right?). Along the way to this problem, I had been encountering unexplained secure shell (ssh) session stoppages – after which sessions would restart and continue. This is extremely problematic when debugging by altering source code on servers and doing reloads. While the reload problem had not yet shown up, the ssh problem had shown up about a day earlier. I hadn't had the time to debug that problem because I was busy doing the javascript debugging between calls, so again, ahah! If I would have tried to debug it I would not likely have found it as quickly because I didn't have the key clue, which was that SSL redirect (it turns out they don't redirect ssh … as far as I am aware … yet).

Conclusions

As it turns out, even the simple infrastructure of my little network gets unnecessarily complicated because of the complex interdependencies unnecessarily created by the desire for companies to force you into and keep you within their walled gardens. I consider this an example of bad behavior by Comcast combined with bad behavior by Amazon. And of course Bad behavior by Apple that forces things to work differently in their systems than other ones.

I didn't get to the level of traffic sniffing in this case, but of course the actual debugging of the Web problem has already entered that realm. Apple injects an iframe into some Web pages in some cases, and of course it is not clear what those circumstances are, and I did not find it well documented on the first several hundred attempts to figure it out.

For clarity, this was not an attack by a hacker. It wasn't part of a global nation-state attack. It was merely the day to day malice of bad actors at so-called legitimate companies. A side effect of their desires to take advantage of their customers. And it started on Halloween!

