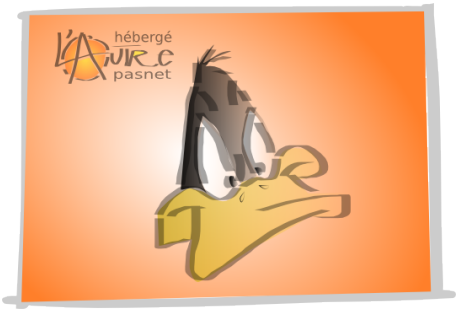
https://www.coincoin.fr.eu.org/?Accidental-DoS-during-an



Accidental DoS during an intentional DoS

- 6- Webographie -

Date de mise en ligne : vendredi 24 octobre 2014

Funny, I remember always liking DOS as a kid...

Anyway, on Tuesday, I took a day off, but ended up getting a call at home from my boss at 4:30pm or so. We were apparently causing a DoS attack, he said, and the upstream university had disabled our net connection. He was trying to conference in the central network (ITS) admins so we could figure out what was going on.

I sat down at my computer and was able to connect to my desktop at work, so the entire network wasn't shut down. It looked like what they had done was actually turn off out-bound DNS, which made me suspect that one of the machines on our network was performing a DOS as a kid...

Anyway, on Tuesday, I took a day off, but ended up getting a call at home from my boss at 4:30pm or so. We were apparently causing a DoS attack, he said, and the upstream university had disabled our net connection. He was trying to conference in the central network (ITS) admins so we could figure out what was going on.

I sat down at my computer and was able to connect to my desktop at work, so the entire network wasn't shut down. It looked like what they had done was actually turn off out-bound DNS, which made me suspect that one of the machines on our network was performing a DOS as a kid...

Anyway, on Tuesday, I took a day off, but ended up getting a call at home from my boss at 4:30pm or so. We were apparently causing a DoS attack, he said, and the upstream university had disabled our net connection. He was trying to conference in the central network (ITS) admins so we could figure out what was going on.

I sat down at my computer and was able to connect to my desktop at work, so the entire network wasn't shut down. It looked like what they had done was actually turn off out-bound DNS, which made me suspect that one of the machines on our network was performing a DOS as a kid...

Anyway, on Tuesday, I took a day off, but ended up getting a call at home from my boss at 4:30pm or so. We were apparently causing a DoS attack, he said, and the upstream university had disabled our net connection. He was trying to conference in the central network (ITS) admins so we could figure out what was going on.

I sat down at my computer and was able to connect to my desktop at work, so the entire network wasn't shut down. It looked like what they had done was actually turn off out-bound DNS, which made me suspect that one of the machines on our network was performing a DNS reflection attack, but this was just a sign of my not thinking straight. If that had been the case, they would have shut down inbound DNS rather than outbound.

After talking with them, they saw that something on our network had been initiating a denial of service attack on DNS servers using hundreds of spoofed source IPs. Looking at graphite for that time, I suspect you'll agree when I say, "yep" :
[PNG - 55.8Â kio]

Initially, the malware was spoofing IPs from all kinds of IP ranges, not just things in our block. As it turns out, I didn't have the sanity check on my egress ACLs on my gateway that said, "nothing leaves that isn't in our IP block", which is my bad. As soon as I added that, a lot of the traffic died. Unfortunately, because the university uses private IP space in the 10.x.x.x range, I couldn't block that outbound. And, of course, the malware quickly caught up to speed and started exclusively using 10.x addresses to spoof from. So we got shut down again.

Over the course of a day, here's what the graph looked like :
[PNG - 52.8Â kio]

Now, on the other side of the coin, I'm sure you're screaming "SHUT DOWN THE STUPID MACHINE DOING THIS", because I was too. The problem was that I couldn't *find it*. Mostly because of my own ineptitude, as we'll see.

Alright, it's clear from the graph above that there were some significant bits being thrown around. That should be easy to track. So, lets fire up graphite and figure out what's up.

Most of my really useful graphs are thanks to the ironically named Unhelpful Graphite Tip #6, where Jason Dixon describes the "mostDeviant" function, which is pure awesome. The idea is that, if you have a BUNCH of metrics, you probably can't see much useful information because there are so many lines. So instead, you probably want the few weirdest metrics out of that collection, and that's what you get. Here's how it works.

In the graphite box, set the time frame that you're looking for :
[PNG - 24Â kio]

Then add the graph data that you're looking for. Wildcards are super-useful here. Since the uplink graph above is a lot of traffic going out of the switch (tx), I'm going to be looking for a lot of data coming *into* the switch (rx). The metric that I'll use is :

CCIS.systems.linux.Core*.snmp.if_octets-Ethernet*.rx

That metric, by itself, looks like this :
[PNG - 82.3Â kio]

There's CLEARLY a lot going on there. So we'll apply the mostDeviant filter :
[PNG - 30.2Â kio]

and we'll select the top 4 metrics. At this point, the metric line looks like this :

mostDeviant(4,CCIS.systems.linux.Core*.snmp.if_octets-Ethernet*.rx)

and the graph is much more manageable :
[PNG - 58.3Â kio]

Plus, most usefully, now I have port numbers to investigate. Back to the hunt !

As it turns out, those two ports are running to...another switch. An old switch that isn't being used by more than a couple dozen hosts. It's destined for the scrap heap, and because of that, when I was setting up collectd to monitor the switches using the snmp plugin, I neglected to add this switch. You know, because I'm an idiot.

So, I quickly modified the collectd config and pushed the change up to the puppet server, then refreshed the puppet agent on the host that does snmp monitoring and started collecting metrics. Except that, at the moment, the attack had stopped...so it was a waiting game that might never actually happen again. As luck would have it, the attack started again, and I was able to trace it to a port :
[PNG - 30.5Â kio]

Gotcha !

(notice how we actually WERE under attack when I started collecting metrics ? It was just so tiny compared to the full on attack that we thought it might have been normal baseline behavior. Oops)

So, checking that port led to...a VM host. And again, I encountered a road block.

I've been having an issue with some of my VMware ESXi boxes where they will encounter occasional extreme disk latency and fall out of the cluster. There are a couple of knowledgebase articles ([1] [2]) that sort-of kind-of match the issue, but not entirely. In any event, I haven't ironed it out. The VMs are fine during the disconnected phase, and the fix is to restart the management agents through the console, which I was able to do and then I could manage the host again.

Once I could get a look, I could see that there wasn't a lot on that machine - around half a dozen VMs. Unfortunately, because the host had been disconnected from the vCenter Server, stats weren't being collected on the VMs, so we had to wait a little bit to figure out which one it was. But we finally did.

In the end, the culprit was a NetApp Balance appliance. There's even a knowledge base article on it being vulnerable to ShellShock. Oops. And why was that machine even available to the internet at large ? Double oops.

I've snapshotted that machine and paused it. We'll probably have some of the infosec researchers do forensics on it, if they're interested, but that particular host wasn't even being used. VM cruft is real, folks.

Now, back to the actual problem...

The network uplink to the central network happens over a pair of 10Gb/s fiber links. According to the graph, you can see that the VM was pushing 100MB (800Mb/s). This is clearly Bad(tm), but it's not world-ending bad for the network, right ? Right. Except...

Upstream of us, we are going through an in-line firewall (that, like OUR equipment, was not set to filter egress traffic based on spoofed source IPs - oops, but not for me, finally !). We are assigned to one of five virtual firewalls on that one physical piece of hardware...despite that, the actual physical piece of hardware has a limit of around a couple hundred thousand concurrent sessions.

For a network this size, that is probably(?) reasonable, but a session counts as a stream of packets between a source IP and a destination IP. Every time you change the source IP, you get a new session, and when you spoof thousands of source IPs...guess what ? And since it's a per-physical-device limit, our one rogue VM managed to take out the resources of the big giant firewall.

In essence, this one intentional DoS attack on a couple of hosts in China successfully DoS'd our university as sheer collateral damage. Oops.

So, we're working on ways to fix things. A relatively simple step is to prevent egress traffic from IPs that aren't our own. This is done now. We've also been told that we need to block egress DNS traffic, except from known hosts or to public DNS servers. This is in place, but I really question its efficacy. So we're blocking DNS. There are a lot of other protocols that use UDP, too. NTP reflection attacks are a thing. Anyway, we're now blocking egress DNS and I've had to special-case a half-dozen research projects, but that's fine by me.

In terms of things that will make an actual difference, we're going to be re-evaluating the policies in place for putting VMs on publicly-accessible networks, and I think it's likely that there will need to be justification for providing external access to new resources, whereas in the past, it's just been the default to leave things open because we're a college, and that's what we do, I guess. I've never been a fan of that, from a security perspective, so I'm glad it's likely to change now.

So anyway, that's how my week has been. Fortunately, it's Friday, and my sign is up to "It has been [3] Days without a Network Apocalypse".
HTML - 1.6Â kio[HTML - 1.6Â kio]