Caveat: it's been a couple years since I have attended a security conference (Shmoocon/Defcon/Blackhat). My opinions are not formed from recent first-hand experience, but through previous stimuli to my cerebrum that have been confirmed by 2nd hand experience over the last couple of years. When I first went to Blackhat/Defcon, it was with the wide-eyed anticipation of, "I'm going to go listen to all of the talks that I can, soak up all of the information possible, and become a super-1337-haxxor." What a let-down of an experience that was. You find the most interesting topics and briefings, wait in lines to get a seat, and find yourself straining your ears to listen to someone that has basically nothing new to say. Most of the talks get hyped up exponentially past any amount of substance they actually provide, most of the "interactive sessions" end up in a "oh! woe is the state of the security industry!" chant, and leave the audience no better off than before.

If you want to learn crazy new things, more often than not, you won't find it at a talk in a con. Why not walk around NSA, find people in offices that do things you find interesting, and talk to them about how they do what they do (or find a mentor in that area)? Despite stereotypes of the kinds of people that work here, many people are kind and open enough to share their trade-craft with others. We have the luxury of working in a community that has some of the brightest, smartest, and most cutting edge people around, it would be a shame for people to constantly attend cons hoping to learn that "cool new thing", when there is exponentially more knowledge sitting around them every single day at work.

Granted, there are always a couple exceptional talks at the cons, but, in my humble opinion, they don't make up for the overall lack of content. So, what good are these conferences? My personal opinion is that their utility is mainly for social interaction and meeting *relatively* like-minded individuals. It's the ability to kick back for a weekend and geek-out with other people. For some, this makes the cost of the con completely worth it, others may be severely disappointed...it all depends on what you expect to get out of it.

Current Mood: 😞okay
So, SIGINT is down right cool. As much as we complain about our "Big Data Problem", collection/processing issues, dismal infrastructure/outdated browsers/OS's, our ability to pull bits out of random places of the Internet, bring them back to the mother-base to evaluate and build intelligence off of is just plain awesome!

One of the coolest things about it is **how much** data we have at our fingertips. If we *only* collected the data we knew we wanted...yeah, we'd fill some of our requirements, but this is a whole world of possibilities we'd be missing! It would be like going on a road-trip, but wearing a blindfold the entire time, and only removing it when you're at one of your destinations...yeah, you'll still see stuff, but you'll be missing out on the entire journey!

So I decided to write a short series (affectionately titled 'I hunt ...') on things that I'm trying to do with data that wouldn't normally be interesting by itself, but by thinking about it in a new way, makes it extremely valuable. My interests lately have been in using passive collect to identify/enable CNE efforts, so that's predominantly what the first few topics will be about.

If there are any topics someone wants to see specifically, let me know. As well, if any of the following information is useful, please let me know and I can put more out. **Part 2 - Hunting sys admins** coming very soon!
(U/FOUO) This post is meant to provide a background for *why* it's good to target sys admins in SIGINT. If you already know this, feel free to skip forward to the next sections.

(SI/REL) Being in SIG, our overall goal is to produce intelligence to give to decision-makers. How we go about doing that, is whenever a target uses technology to communicate, we collect it, analyze it, and write reports on it. Sounds simple enough...except for the fact that we have to be targeted in what networks we collect. We can't collect everything all the time, so if a target starts to communicate on a network where we are not collecting, there is some manual leg-work that has to be done to steer the SIGINT system in their direction. This is where I must introduce my loyal friend, the sys admin.

(SI/REL) Up front, sys admins generally are not my end target. My end target is the extremist/terrorist or government official that happens to be using the network some admin takes care of. Sys admins are a means to an end. For example, assume your target is using a CDMA device on a foreign network; there may be situations where we passively collect his phone call/SMS out in the wild, but it would be *really* nice if we had access to the local infrastructure where we could monitor which tower he's connected to at any given point in time, or monitor all phone calls/data traffic that his phone generates. Many times, it's difficult to directly target infrastructure...generally we'll need a fair amount of information going into an operation, such as:

- *topology of the network we are targeting*
- *credentials for infrastructure devices*
- *situational knowledge, such as access lists set up to only allow specific IP addresses to administer certain machines*
- *an overall knowledge of how the network is put together and configured*

(SI/REL) In order to get that, who better to target than the person that already has the 'keys to the kingdom'? Many times, as soon as I see a target show up on a new network, one of my first goals is, "Can we get CNE access to the admins on that network, in order to get access to the infrastructure that target is using?"

"Yeah, that pretty much makes sense, but how are you 'just gonna get CNE access' on an admin?"

(SI/REL) Good question, thanks for asking. Most of the time I’m going to rely on QUANTUM to get access to their account *(yeah, you could try spam, but people have been getting smarter over the last 5-10 years...it's not as reliable anymore)*. So, in order to work our QUANTUM-magic on an admin, we’ll need some sort of webmail/facebook selector for them.

"You know, you *could* just look up the 'point of contact' in the registry information associated with their IP space/domain names..."

(SI/REL) Yeah, you could do that. Personally, I haven't had a huge amount of luck with it, because most of the time I end up running across their *official* e-mail address that's hosted on their own network. That's generally not a recipe for success in the QUANTUM world, what we’d really like is a personal webmail or facebook account to target. There’s a couple ways you could try this: dumpster-dive for alternate selectors in the big SIGINT trash can, or pull out your wicked Google-fu to see if they've posted on any forums and list both their official and non-official e-mails in a signature block...but what if there was another way to do it?

(SI/REL) Other fun (read:useful) things to get off of a sys admin (from my point of view):

- *network maps off of their hard drive*
- *credentials from text files (or from our key-loggers...potato potato)*
- *full lists of customers (along with associated dedicated IP allocations is a bonus)*
- *e-mail with upstream providers detailing how your network is connected to the bigger Internetz. For example, if I see they use certain fiber cables to connect to the world, I'll go look in SSO's collect for their traffic. If they use VSAT's, I'll go look for their network in FORNSAT's environment.*
- *pictures of cats in funny poses with amusing captions*

(SI/REL) But all of this boils down to getting an admin's webmail/facebook account in order to QUANTUM it and get CNE access to their box. Next section will detail targeting admins who use telnet...
[advertiser]/local-post - #3 ext 1.2.3.4

**LogLevel** Assuming 1.2.3.4 actually means learning on your own through the Swahili hardtops, and the server will send you back some packet and something like this may pop up:

> "Fehn, but that wouldn't work after all against any sys admin that uses SSH, because we would never see the content of the config! It's encrypted!"

**[SysAdmin]** That's absolutely correct, however. **[OptionsBase]** I think it's still possible to identify a given system's SSH configuration with one quick and easy method. First off if you're at all familiar with SSH, please feel free to find out what the resource graciously posted by one of my favorite friends and homebrewed by myself.

**[SysAdmin]** Before we get into the methodology, let's talk about SSH for a minute, and some of the great new tools that are available to us now. Let me assume that you're familiar with SSH, and that you're going to make up from now on: I'm assuming we're looking for SSH on port 22. I'm sure there's ways to find it on non-standard ports, but just for the sake of simplicity, I'll just lump them all together into port 22. Let's assume that you're using 0.0DHA to connect to some server:

```
   |advertiser/local-post - #3 ext 1.2.3.4
```

**Enter username:**

**Enter password:**

**[SysAdmin]** Okay, here's where the rubber meets the road, one of two options exist: Either you have the right password, or you don't. Let's explore what would happen under each of the circumstances:

**[OptionsBase]** If you pretend you do NOT have the right password, you try **admin** for the password and press enter.

**[SysAdmin]** If you're password guessing, you'll probably get it right before the server tells you the connection, and you have to re-enter and try again. However, if you do have the right password, you'll be able to log in and the server will send you the output for all of the commands that you run.

**[SysAdmin]** I know, sounds ubersimple, but here's why it matters. Think about all of the other sys administrators out there, we're looking for a *very* easy way to identify that you have a live server, and then we can proceed to attack the server, and try to connect to a given username, sending a password, and sending commands (assuming the client successfully logged in). It's too much work for people to keep doing that. That depends.

**[SysAdmin]** If the client does not have the right credentials, we expect the server to direct connection to exit.

1) sending the response's banner
2) sending the prompt for a password
3) sending something for a username
4) sending another prompt for a password
5) sending BACKSPACE to clear the buffer
6) killing the connection, and forcing the client to restart and try again.

**[SysAdmin]** If the client does have the right credentials, we might expect the server to direct connection to exit.

1) sending the prompt for a password
2) sending the prompt for a username
3) sending the prompt for a password
4) 4 thru 6 sending back the output for whatever commands the client has run.

**[SysAdmin]** So, purely based off of the above assessment, we would expect unsuccessful logins to be identified, but regardless, in the number of sessions, the successful login will be of variable length, but probably considerably longer in size (in bytes) because of the additional commands that are executed. Saving the successful logins will be of variable length, but probably considerably longer in size (in bytes) because of the additional commands that are executed.

You can question whether an SSH session was successful or not **MUST** based off of the size of the egression in the server-to-client direction.

**[SysAdmin]** So, imagine that you do some analysis, and determine that 1500 bytes is roughly what you're looking for, okay? After we have a list of the IP addresses that have been in contact with the server, we can look for a specific pattern in the session below that splice is probably an attempt, and any server-IP pair that has that size is probably a successful attempt. What could you do, either, with information like that?

1) You could create lists of client IP addresses that consistently are unsuccessful in SSH to multiple servers. Then you could test if these are 1,268 packets, probably brute force.
2) You could use filters on IP addresses that appear to be in successful in having access to other IP's. (Leaky jeeps, here is where we can have fun.)

**[SysAdmin]** The previous two points, you can do with that sort of data, but more likely interested in #2 at the moment. Based purely off of:

* session size is greater than 1500 bytes
* then infer that:
  * 0 IP = admin
  * 1 IP = possible shenanigans
  * 2 or more IP's = possible vulnerable

The admin appears to have successful access to the server "Aassadov"

**[SysAdmin]** From here on, all I have to do is recycle my methodology from last cent! I can trace ALL OF SIGNIFICANT for sessions that meet the above criteria, harness a list of "admin" IPs, use that as a filter on the server to do multiple searches. Then you put this data into a, probably packet, probably probably brute force.

**[SysAdmin]** You could do something similar with IP addresses that appear to be in successful in having access to other IP's. (Leaky jeeps, here is where we can have fun.)

**[SysAdmin]** One of the reasons why you'd only do login with Admin Login within a certain timeframe of a list that is in the hands of someone with admin, and not some other random person who happened to receive that some IP via DHCP the day before.

**[SysAdmin]** The previous few posts have gone into some fun ways that we can harvest the power of the system. It's a topic that could be expanded further, which is why we can't catch someone who either the legitimation sys admin has to access to a system.

**[SysAdmin]** I'm sure there are other fun and innovative ways to go about this, so let's finish to share them if you think you can, or also feel free to take any of the above thoughts and use them for your own purposes. I'm just maintaining a list of IPs that we're successfully guessing in SSH.

As always, if you have any questions, or want more info, feel free to drop me an email or leave a comment.
Happy Friday my esteemed and valued Intelligence Community colleagues! There has been a topic of conversation that has started to rumble beneath the surface of the Cyber-scene lately, it's about router hacking (for this post, I'm not talking about your home ADSL router, I'm talking about bigger routers, such as Ciscos/Junipers/Huawei's used by ISPs for their infrastructure). Hacking routers has been good business for us and our 5-eyes partners for some time now, but it is becoming more apparent that other nation states are honing their skillz and joining the scene. Before I get into it too much, let's go over some of the things that someone could do if they hack a router:

* You could add credentials, allowing yourself to log in any time you choose
* You could add/change routing rules
* You could set up a packet capture capability...imagine running Wireshark on an ISP's infrastructure router...like a local listening post for any credentials being passed over the wire(!)
* You could weaken any VPN encryption capabilities on the router, forcing it to create easily decryptable tunnels
* You could install a dorked version of the Operating System with whatever functionality you want pre-built in

There are a plethora of things you could do once you get CNE access to a router...suffice it to say, getting access to a router is very good for the actor, and very bad for the victim. So, we would obviously love to know which countries/actors have access to what other routers (especially if it's our routers). Then the question comes down to:

"How would you identify the fact that someone has CNE access to a router?"