Getting Started with Assets
By Joseph Orsetto

How can you secure something if you don't know what you have? Meaning, how can you claim to be secure if you do not have an idea of what you are securing. The goal of our work here is to be able to state with confidence and proof that we know exactly what is in our environment. Oh yeah and also WHERE it is. We need to be able to remediate any issues arising such as failed equipment, physical migrations or even unauthorized workstations plugged-in. It is not enough to say, "Hey there is a bad guy on the network!". If you don't believe me imagine yourself in this situation.

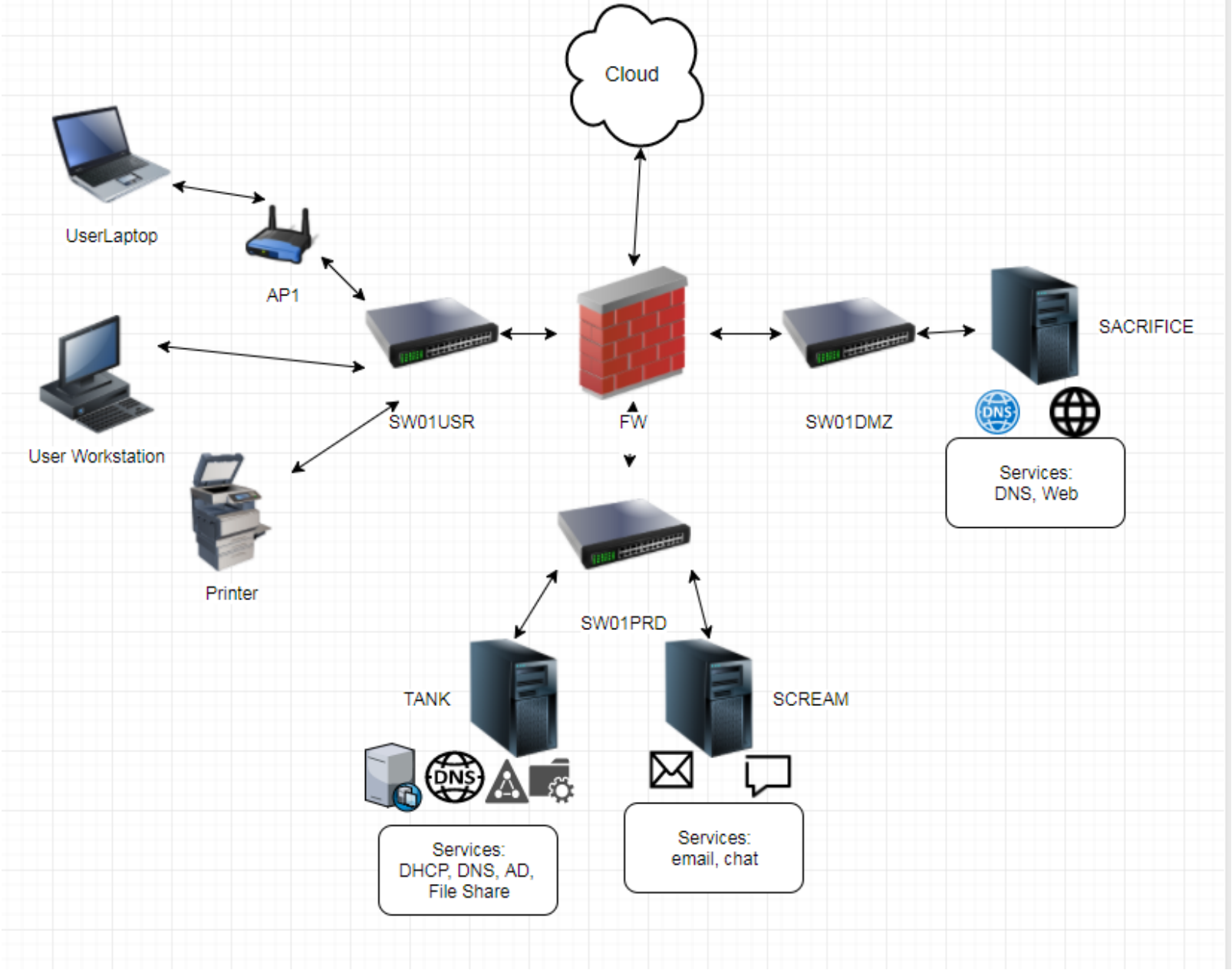
You get a call stating that the user network is slow. From here you log into the firewall and notice that your interface that connects to your ISP is saturated. Great, now how do we figure out who is causing it? IF you already have all of your networking equipment routed and logging properly then maybe you can get a "top talkers" off the firewall. With this report, you find the IP address is 10.0.0.234 it the top talker. Great now how do we know who this is? You grab the MAC address linked to that IP address off the firewall. From here you start to dump ARP tables on your networking gear. You trace it down to Switch 2 with an IP address of 10.0.0.12. Now that we know the switch that the device is connected to, we log into the switch... what was the account to login again? Locate what switch port has the MAC address listed on it. There it is port Gi1/0/24! From there we administratively down the interface to go have a conversation with the user. Go find the system that was causing the issue, remediate and get back to the work you were doing before this whole mess. But where is Switch 2 physically? Where does port Gi1/0/24 go to? What user is on the other side of that port? Was it the CEO? This is all assuming that you have the proper logging and reports available on the firewall or router.

Did I just trigger your PTSD? Trust me you do not want to be in this situation, let alone in it with business side users yelling at you. So how do we limit the stress for when this inevitable occurs? Documentation is key and is used to ensure that you are able to expedite the remediation and know what is connecting where. What is even more important than documenting itself is ensuring that the documentation is also available to those who may need to manage or access it during an outage. Always try to plan for a failure!

Where do you start when you want to document an environment/network? I would say start with what you know. Now if you say you have no idea about any device other than the one you are currently working on we can still get some work done. I will cover a bit more of the process a little later, for now I want to show a bit of detail on our diagram examples. The following diagrams I created using draw.io, you just need to use a tool that works for you. The tool does not matter, we just have to make sure that the documentation lays out the desired information. We will create two diagrams to show the needed information. A physical and a logical network diagram. How the cables are run (physical), and how the data flows through the environment(logical). From there we will look at recording some of our more detailed items for use later. If you would like more detail I am only covering the topics as quickly as possible so feel free to read through here: http://networkdiagram101.com/

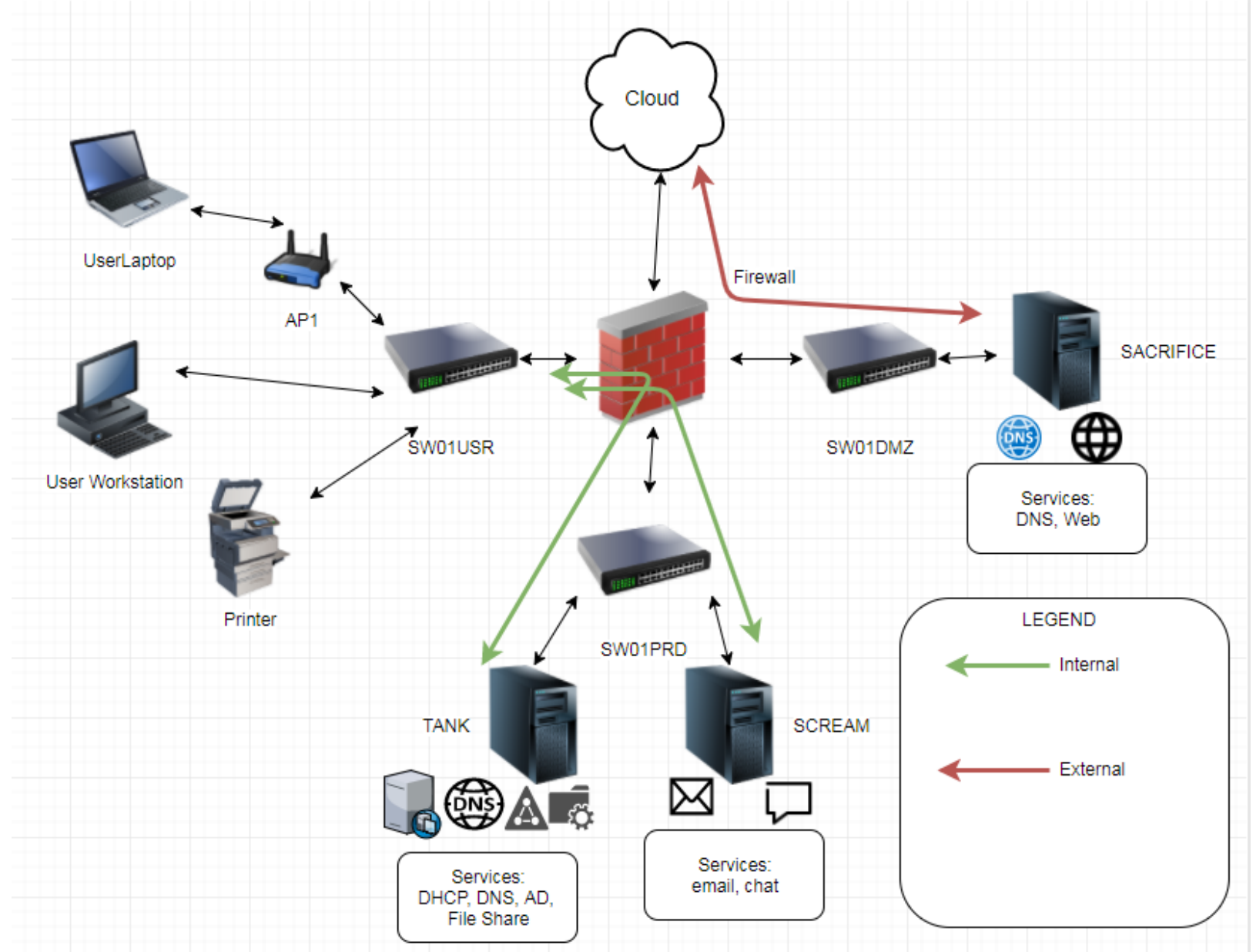
Before we get started I should mention that the inventory will be evolving as time goes on. We are just getting a good start here and it can/will be added to. As such make sure you have your documentation handy.

Step 1.1 - Document the network - Physical Topology
Draw out how each service providing device is connecting, and provide an example of where a client could live (You do not need to have an example of every client in the environment). We are looking for an infrastructure focus for a main diagram. However, in order to prevent the panic in an incident it may be good to document the patch to switch link. A quick example is below:



Great you made it this far! What did I not include in the diagram? How could it be improved? What information do you think you would want to have if you needed to go and correct an issue?

Step 1.2 - Document the network - Logical Topology (Data flow)



Now that we have an understanding of where devices are located, and how we believe they are communicating we can consider verifying our thinking. I would recommend having this data stored in a database to make usage easier down the road. If you are really starting with no information this is where we need to start to dig. We will perform our verification of the physical topology based on the process below:

Step 1.0 - Identify where you are
Since no information is provided we need to look at gathering details from our workstation out. We simply can look at our local interface and see what the networking details are. From there we can expand outward. On a windows machine we can run **ipconfig /all**. The output will give us the IP address of:

- Our workstation
  - Hostname - Can tell if there is a naming convention, and if so what it is.
- Default gateway
- DHCP Server address
- DNS Servers
  - DNS Suffix
- IPv6 usage (Or rather lack thereof)

Now we have a good foundation, but let us try and build on this.

Step 1.1 - Network Scan.
Ok before we get cracking on this let us take a moment to level set and ensure we are on the same page. Some of the action we will be performing have the ability to impact production systems. I would recommend getting approval

before completing the following steps. If you take a system down you want to make sure everything is kosher. I would start off with a ping sweep to see what is online. Ideally you would store this information in a database.
nmap -sn 10.0.0.0/24

From here you will get your hosts that are responding to ICMP. I typically will assume that anything on the same network will not have much in terms of network restrictions. Because of this a ping sweep scan will work to get us started most of the time. After the scan is completed you will have some results, lets poke around and see what we have available, I would then scan the found IP Addresses with a -sS to see what services are open.
nmap -sS 10.0.0.0/24

Now we have a list of not only what system responds to ping, but also an idea of what services are running on each live system. To ensure it was noticed, I said live systems. Meaning if a device is powered off it will not respond to ping, and thus we will not have results for it. Now typically subnet addresses are assigned in a logical pattern. The best practice using the example network diagram above is to have the first address assigned to the router and the last address assigned to the firewall. So that means:

- 10.0.0.1 = Router
- 10.0.0.254 = Firewall

In order to move on to the next steps we need to have a starting place. Do you have a reply to the nmap scans on .1 or .254? Does your ipconfig /all output list a gateway address? I was able to see a response on 10.0.0.1, and not on 10.0.0.254, and this lines up with the gateway from the ipconfig /all output. I recorded everything in an excel sheet (or csv) and moved to the next step.

Step 1.2 - Review DHCP Server Configuration
DHCP servers contain different pieces of information that can assist with the overall identification of the environment. Here we look at the pools, reservations, active leases and lease duration. Are there any pools that should not be configured or active? Are there any devices that should have a reservation but do not? Are there exclusions within the pool that may cause issues? Are the pools sized properly to ensure that the users will not be getting disconnected? What about DHCP options? Are the DNS servers correct?

Step 1.3 - Review Firewall Configuration
Log into the gateway and review the different components listed below. I know it is a lot but it has to get done! Keep the focus on the task at hand, however do a quick scan over all areas as you may need to change tasks.

- Does the firewall perform DHCP?
- Does it have any VPN tunnels configured?
- What are the networks that are attached to the interfaces?
    - How are they connected? IS it a trunk port etc.
- What are the features that the firewall is capable of?
    - What are you licensed for?
- What does the rule base look like?
    - What are the inbound connections allowed from the internet?
    - What are the outbound connections allowed?
- What does the NAT policy look like?

Step 1.4 - ARP table dump. Or CDP/LLDP
As the CLI can be very different in terms of the command syntax I will just explain the theory here and leave the practical up to you. How I have approached mapping out the environment is a combination ARP/CDP/LLDP. If you are unfamiliar with CDP/LLDP all is well keep reading, if you are unfamiliar with ARP I recommend that you consider some networking foundations material. CDP and LLDP are protocols that can be used to identify other devices that are directly connected, however those devices must also support the protocol for it to work. More reading for CDP HERE and LLDP HERE. The process is as follows:

1. As we now know the IP address of the firewall. Let's list the interfaces, and dump the ARP table on the interfaces. Also, if you know your firewall supports CDP and LLDP try those as well. The combination or comparison can be used to validate the information that you currently have.

2. Focus on one interface at a time and review the MAC addresses for networking equipment there. If there are multiple MAC addresses listed on an interface it means that the port it used to provide access to other devices and is not terminating on a single node. (Note: this could also be a host of virtual devices. You should be able to tell based on the MAC address.)
3. On all switches and routers perform an ARP table dump (or CDP/LLDP).
4. Note any ports that have multiple addresses as possible links to more networking equipment. If CDP/LLDP information is available then compare. Else take note of the interfaces for further validation in step 1.5
5. Update the diagrams (Physical and logical) to have the new information.

Step 1.5 - Physical Validation... Yup you have to go for a walk, however we first need to get some documents!

1. Get a copy of the floor plans for the office from HR/Procurement/Facilities. They should have a digital copy for the fire route plans.
2. Update the diagram with the location of:
    1. Wiring closets. These also should have a naming convention that makes them easy to find. Say F10NC02 for Floor 10, North side of the floor, Closet 02.
        1. Take a look at what is in the room. If it is not appropriate or needed make a plan to remove it.
    2. Access Points/Wireless Routers.
    3. Power Rooms (Could be the same as the Wiring closets).
        1. While you are here take note of any UPS devices and their current health.
        2. Take a look at what is in the room. If it is not appropriate or needed make a plan to remove it.
    4. Validate if the patch cables are labeled and mapped properly.
        1. If not and you have the time to update properly great, if not we are here to get the most out of our time, and this can be a massive time sink.
3. Update the diagrams (Physical and logical) to have the new information.

Step 1.6 - Automate scan for changes in the network.
Alright, as much fun as that was to go through manually, I figure if you made it this far through the wall of text you deserve some automation. Below is a bash script that I modified to work for me. Copy the text and modify for your environment.

```
#!/bin/sh
TARGET="192.168.1.0/24"
NMAP_OPTIONS="-v -T4 -F -sV"
DATE=`date +%F`
cd /home/joe/nmap
nmap $NMAP_OPTIONS $TARGET -oA scan-$DATE > /dev/null
if [ -e scan-old.xml ]
then
ndiff scan-old.xml scan-$DATE.xml > diff-$DATE
echo "------NDIFFed-RESULTS------"
cat diff-$DATE
echo
fi
echo "------SCAN-RESULTS------"
cat scan-$DATE.nmap
ln -sf scan-$DATE.xml scan-old.xml
```

Alright you are good to go, right? Just kidding let me break the code down for you, however sometimes it is nice to see the code without the comments to get an idea of what is being done. Now let us look at it with comments!!!

```
#!/bin/sh
#
# Variable declaration. These you need to make sure fit your environment.
### TARGET is specifying the network that will be scanned. Again, update as
needed.
TARGET="192.168.1.0/24"
```

```
### NMAP_OPTIONS we specify the flags that control how we will scan the
targets.
NMAP_OPTIONS="-v -T4 -F -sV"
### DATE will make it so the format is YYYY-MM-DD
DATE=`date +%F`
#
# Move over to the desired directory where files will be stored. Again, this
should be modified to fit the environment.
cd /home/joe/nmap
# Here is where the nmap command is made. This is where the work gets
done. We send our output to /dev/null so that we are not crowding the cli.
nmap $NMAP_OPTIONS $TARGET -oA scan-$DATE > /dev/null
# if statement, first we look to see if there is an old file
if [ -e scan-old.xml ]
then
# We compare two files, old and current and we drop them into a third file diff-
$DATE
ndiff scan-old.xml scan-$DATE.xml > diff-$DATE
# Place words on the string so we can see what is going on.
echo "------NDIFFed-RESULTS------"
# show the contents of the third file
cat diff-$DATE
# make a blank line on cli
echo
# now leaving the if loop.
Fi
# Place words on the string so we can see what is going on.... Again.
echo "------SCAN-RESULTS------"
# Display our full current results.
cat scan-$DATE.nmap
# copy the current scan over the old scan.
cp scan-$DATE.xml scan-old.xml
```

Now let's get the script to run at a desired frequency. We will make an entry in crontab that links to our script. Now remember when choosing when to run this script you need to make sure that the desired systems are online. So, if you are scanning a user network you need to make sure that you are scanning when your users are around. In this example, I will scan during the lunch break. In the CLI enter

 # crontab -e
0 12 * * * /home/joe/nmap/nmap_scann_diff.sh

For crontab there are

0 = 0 minutes after the hour.
12 = high noon, or the 12th hour of the day. Note this is in 24-hour time.
*** = All days of the month**
*** = All months of the year**
*** = All days of the week**
/home/joe/nmap/nmap_scann_diff.sh = File that is run

There now we have a script that will run every day at noon! There is a lot of expansion this script can handle and we will work on that in the future. For now I think we have done some good today, so let's recap.

Recap
After all this work we now have a much better idea of what is where. We also have a large amount of data from scans and manual verification. From here we need to make the information gathered usable, let's get to work!

- Physical Diagrams

    - Ensure all systems found that are related to networking are listed.
    - Ensure all host systems or general purpose are listed.
    - Ensure connections are listed correctly.

- Logical
    - Ensure all systems purposes as well as flow types are listed.
    - Ensure detail in terms of what traffic is going where is mentioned.
    - Ensure protocols are covered for each connection

- Device Tracking
    - For small deployments you can use Excel, though be warned this can quickly become hard to manage.
    - For large deployments you need to look into a CI manager.

- Define a maintenance process
    - If you don't define one what is the point in doing all this work in the first place? Here is where I would also recommend looking into getting automation in place to assist. Never underestimate the power of scripting and BASH!

At this point we have a great start to our network documentation. From here as more digging and system maintenance is performed these documents need to be managed. Depending on the size of the environment these are really the tip of the iceberg in terms of what will be needed.