Networking Basics

For starters; we need to talk about what a network is. A computer network by definition is any setup where two or more computers are able to communicate with each other. This can be done in 2 ways (we are starting simple here). Peer-to-peer is the easiest way, and you just need a single ethernet cable with a Class A and B CAT-5 on opposing ends. You plug the Class A cable in one, and the Class B in another. This allows us to transfer data between the two machines without the need for a router, or the internet. The second type requires routers, switches, and modems. This is the main ingrediant for most home networks today, and it is required to connect to the internet. For a wired network in the US we would use the Class A CAT-5 cable, and for the UK we'd use the Class B; you can use either if you want to make them yourself, but this is generally the standard companies will use in each respected country.

What's the difference between Class A & B? Nothing really; the main difference is how the cables are wired together, we generally don't need to understand how this works for our purposes, but it's still good to know so you're grabbing the right cables when setting it up yourself.

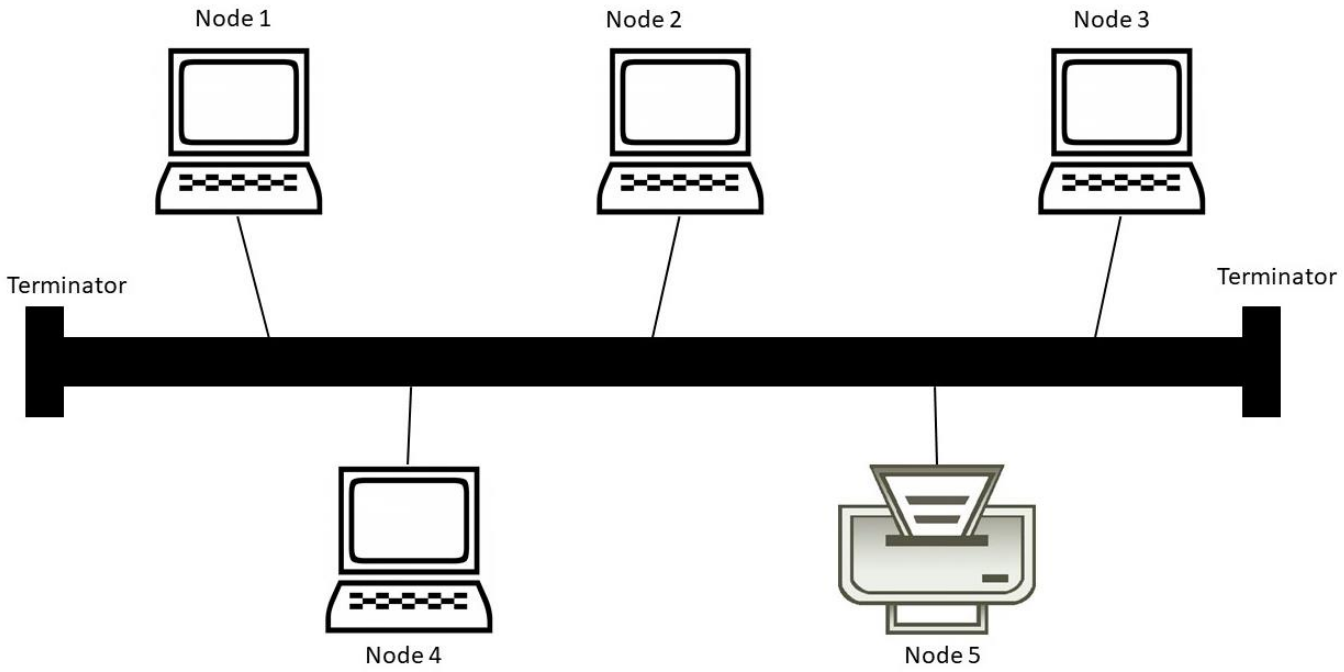
We also need to identify a node; this is just a fancy word for a machine on the network. We say machine because it can be anything from a server, to a router, to even a laptop or workstation (desktop PC).

[Network Topology:]

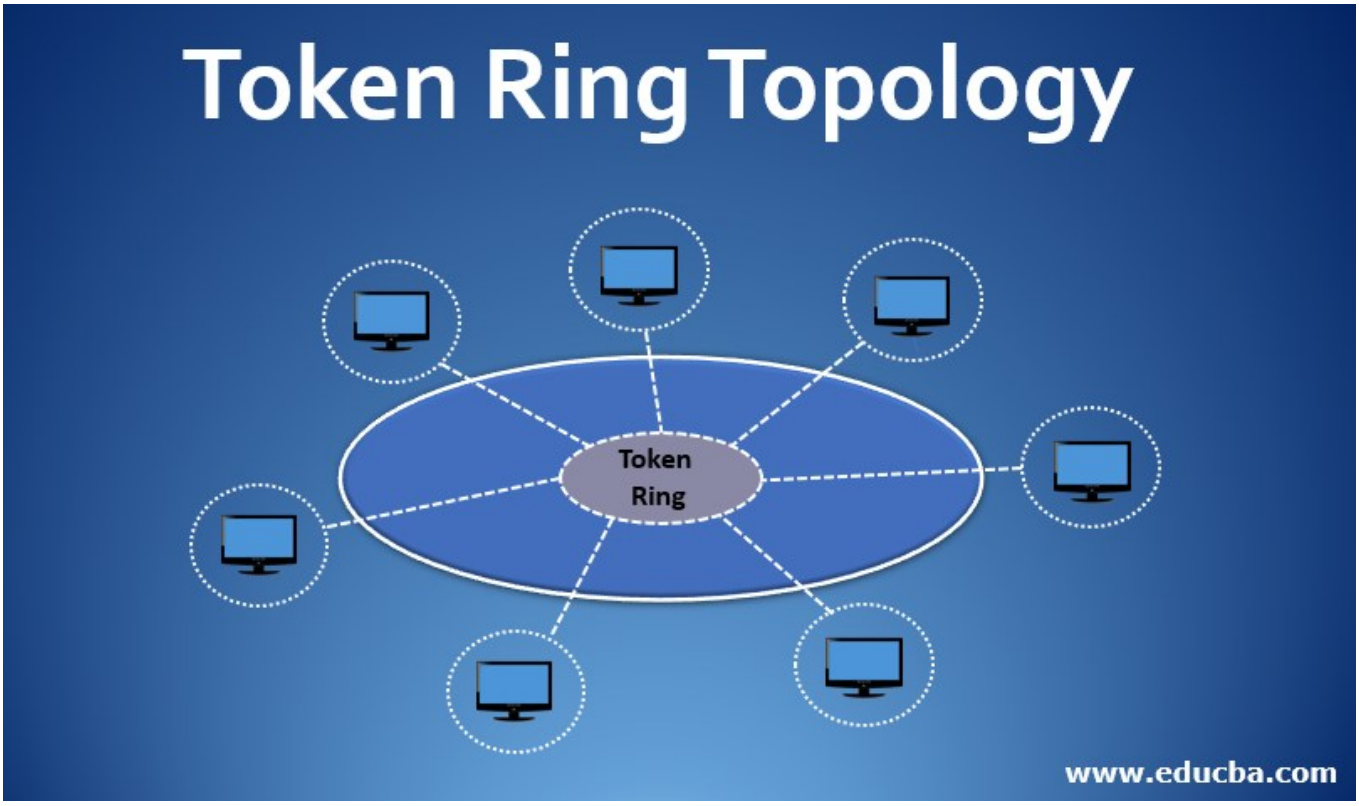
This is the actual structure of a network, and we can learn a lot about a network's topology from a simple OSINT map. We most likely won't need to do this as hackers, but it's still good to know you can if you really need to. (OSINT is getting a whole notebook of it's own...)

There's 6 types of network topologies: the ring, star, bus, star-bus hybrid, star-ring hybrid, and mesh.

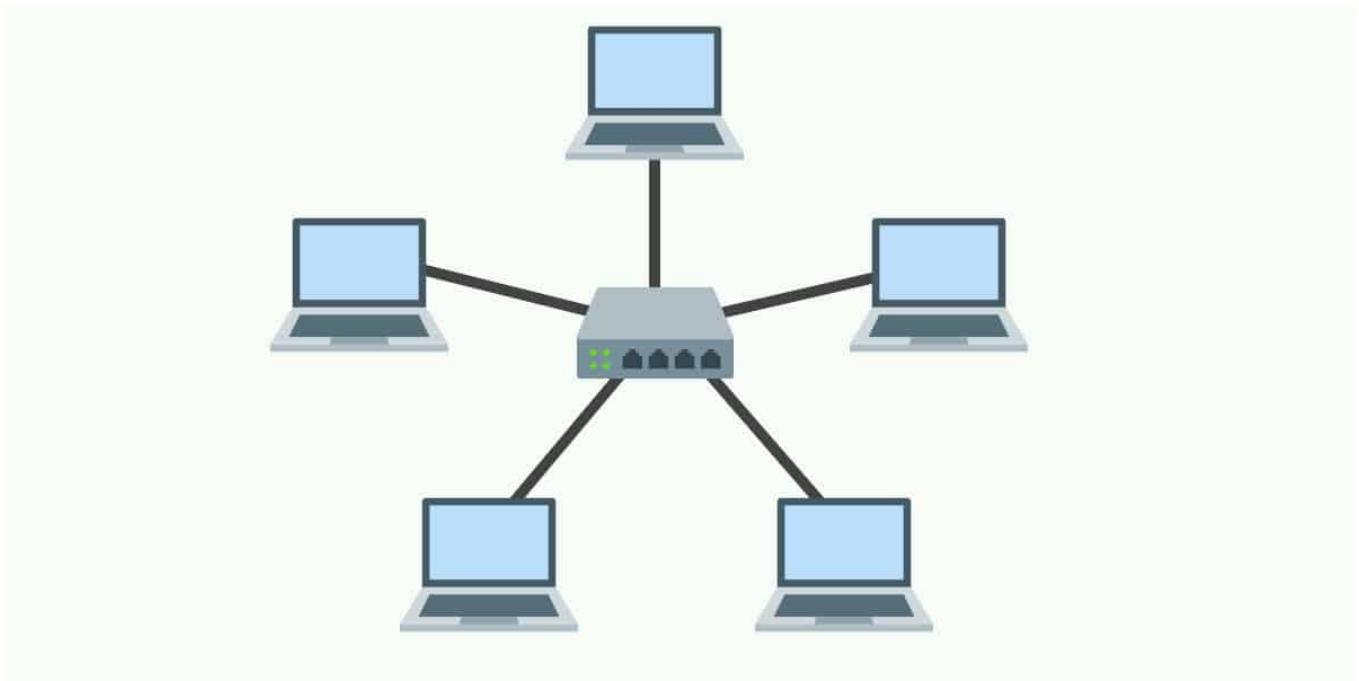
[Bus Topology:]
This is like a giant chain network with cables being strung from one network node to the next. I.e. a laptop connecting to a server would do so with a cable rather than a router. Each packet sent is then detected by all nodes on the segment; each segment has a start terminator and an end terminator.



[Ring Topology:]
The ring topology is very interesting; it's a giant ring of continuous dataflow that doesn't require any terminators for it to run since it doesn't have a specified start or end. It works by starting at any node, and going to the destination; it then keeps going unil it goes back to the starting node also known as the source node. This ring is powered by a token; a token is a random string of characters used to authenticate the data transmitions being sent. This is used to avoid collisions, the parts for these networks are super expensive! As a result they are rare and have less room for scalability.



[Star Topology:]
These are the oldest network types in existance: they go back to telephone switching systems. They are usually connected via a hub or switch in the center with all the nodes being connected directly though the switch or hub. (A switch is just an updated hub this is the easiest way to understand this even if it's not exactly true hubs are just slowly being outdated). This leads us to a single point of failure within the architecture of the system; if the centeral router/swtich/hub goes out; you do not have a network...

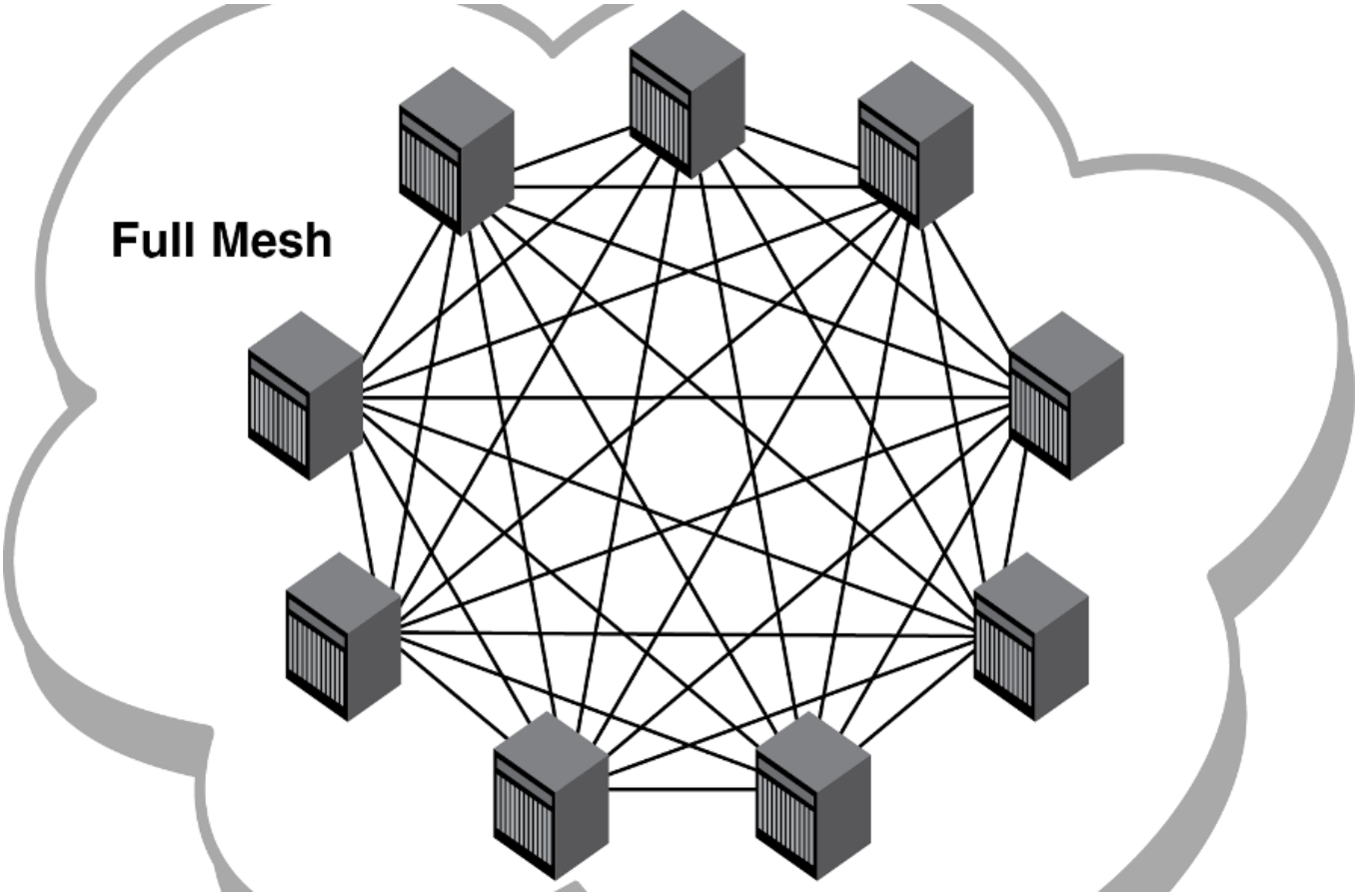


[Star-Bus Hybrid Topology:]
This adds a backbone to the star network topology; a backbone is a high-capacity communications medium that joins networks and centeral network devices on the same floor in a building, on different floors, and across long distances. It also allows us to have a separate bus segmant on each node of the star; but it is now limited to 3 per segment. You can also conenct multiple hubs, swtiches, or routers to these networks as well (expansion opurtunities).

[Star-Ring Hybrid Topology:]

This is what modern ring topologies look like; it's a centeral hub, switch, or router that transmits data through a token ring as though it was on a ring, but there's no need for built-in terminators in this configuration (becuase it uses tokens).

[Mesh Topology:]
This is the best type of network becuase it provides fault tolerance; this means if a single node stops working the network can still function unlike the star topology. Each node is connected to every other node (including routers, switches, etc.) which is why it's more robust and gives us way more expansion room over the other types.



Now that we understand these concepts we can move onto the best parts (not really but we really need to know them as hackers).

[IP Addressing:]
This is a heavy load here, so we can spend some time figuring this one out... it's super important as most of our exploits will be attacking an IP address in some compacity.

This will cover how to create an IP address from binary (not as hard as it sounds), the whole way to subnetting a network with CIDR notation.

Things to know:

1. IP addresses come in 2 types: Public, and Private (you also have public/private ipv6 addresses) We will talk about this in the next section.
2. An IP address is split up into sections called octets, each octet can hold anywhere from 0-255.
3. An easy way to setup a binary conversion is writing it down right to left starting at 1, and doubling it until you hit 256.
   256 168 64 32 16 8 4 2 1
   If all of the bits are on, we consider it 255. otherwise you need to do some basic addition to figure it out.

Below will be an IP address cheet sheet; it'll help with subnetting a network (very useful skill to have!)

	A	B	C	D	E	F	G	H	I
1	Subnet x.0.0.0								
2	CIDR	/1	/2	/3	/4	/5	/6	/7	/8
3	Hosts	2,147,483,648	1,073,741,824	536,870,912	268,435,456	134,217,728	67,108,864	33,554,432	16,777,216
4	Class A	Subnet 255.x.0.0							
5	CIDR	/9	/10	/11	/12	/13	/14	/15	/16
6	Hosts	8,388,608	4,194,304	2,097,152	1,048,576	524,288	262,144	131,072	65,536
7	Class B	Subnet 255.255.x.0							
8	CIDR	/17	/18	/19	/20	/21	/22	/23	/24
9	Hosts	32,768	16,384	8,192	4,096	2,048	1,024	512	256
10	Class C	Subnet 255.255.255.x							
11	CIDR	/25	/26	/27	/28	/29	/30	/31	/32
12	Hosts	128	64	32	16	8	4	2	1
13									
14	Subnet Mask(replace x)	128	192	224	240	248	252	254	255
15									
16	Notes:								
17	Hosts always double								
18	Always subtract 2 from total								
19	- Network ID = First Address								
20	- Broadcast ID = Last Address								
21									

To read this graph there's a few things we need to note:

1. CIDR notation dictates how many hosts are allowed on the network, as well as the network class (depicted by each subnet.)
2. We always subtract 2 hosts from the total due to one being the Network ID and the other being the Broadcast ID.

3. We replace x with hte number under each column depending on the subnet configuration. For example: most home network subnets are: 192.168.1.x/24 by default. We can also feed this information into tools like nmap with the exact format. This will be discussed later in more detail.
   We can read more about CIDR notation here: https://whatismyipaddress.com/cidr
   I don't really remember much about it other than how it's used, so moving on...

Private Networks use private IP addresses and are called Local Area Networks. These can be anything from a simple home network, to a deep enterprise network they want to keep off the net. (These do exist)

Public Networks use public IP addresses and can be viewed by anyone; CAN, WAN, etc. (campus area network and wide area network) are connected to the internet for convience and ease of work. Students obviously need it for their studies, and some companies do a lot of their management through external systems and consider cloud storage a must have for disaster recovery plans.

Most attack vectors pretaining to IP addresses at the critical level are private; this is because it gives us direct system access. Your public IP can be used to get a general location of residence, but can't really do much other than cut off your internet access which we can circumvent entirely.