

Defending the Borders of Internet Traffic

IIT Cyber Infrastructure Lab Protocols & Tools



Meet the team



James Jones
Project Lead



Thomas Cox
Technical Lead



Nathan Plenzler
Communication
Lead



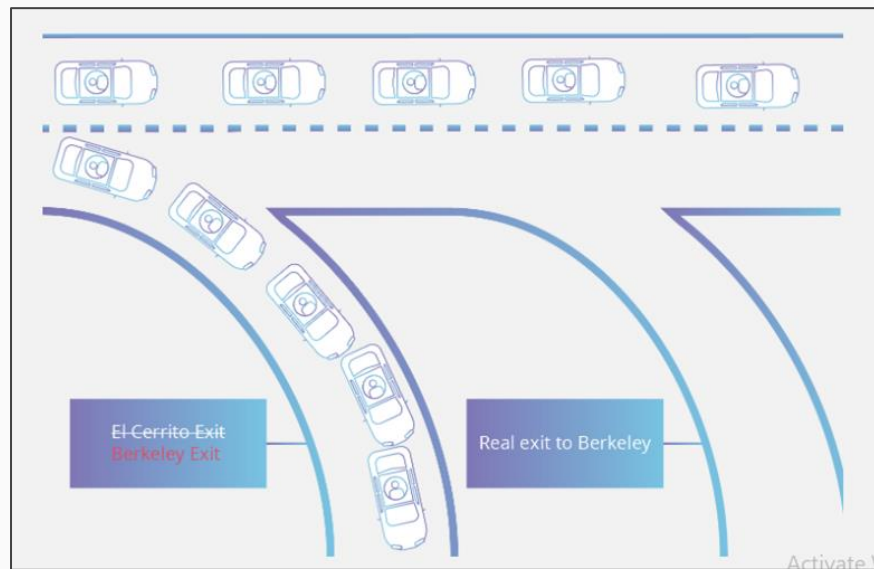
Chris Weidner
Quality Lead

Mission statement:

To create a instructional Lab's that teaches users how to emulate a fully functioning network of interconnected autonomous systems (AS) in order to study external Border Gate Protocol (BGP) hijacking.

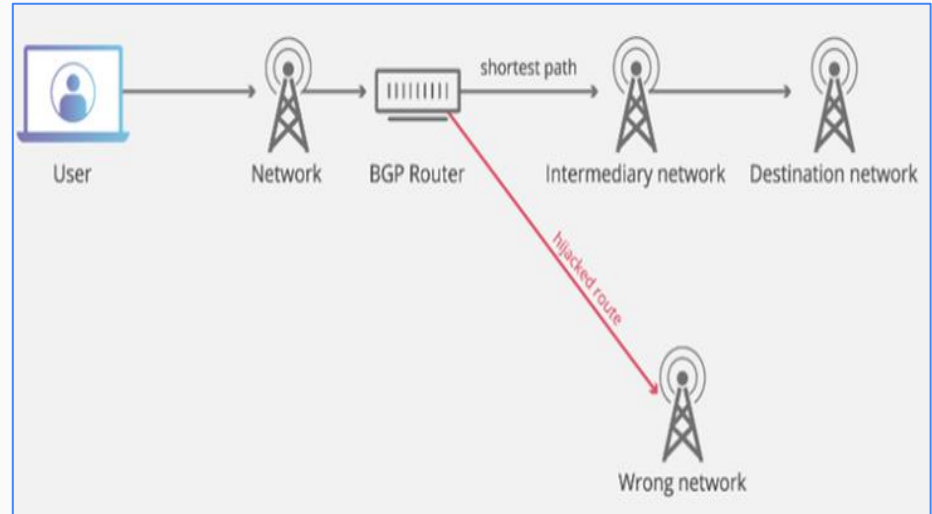
The problem

- BGP hijacking is when attackers maliciously reroute Internet traffic. Attackers accomplish this by falsely announcing ownership of groups of IP addresses, called IP prefixes, that they do not actually own, control, or route to.
- Imagine if someone were to change out all the signs on a stretch of freeway and reroute automobile traffic onto incorrect exits.



The problem

- Because BGP is built on the assumption that interconnected networks are telling the truth about which IP addresses they own, BGP hijacking is nearly impossible to stop





The solution

This is more difficult to answer.

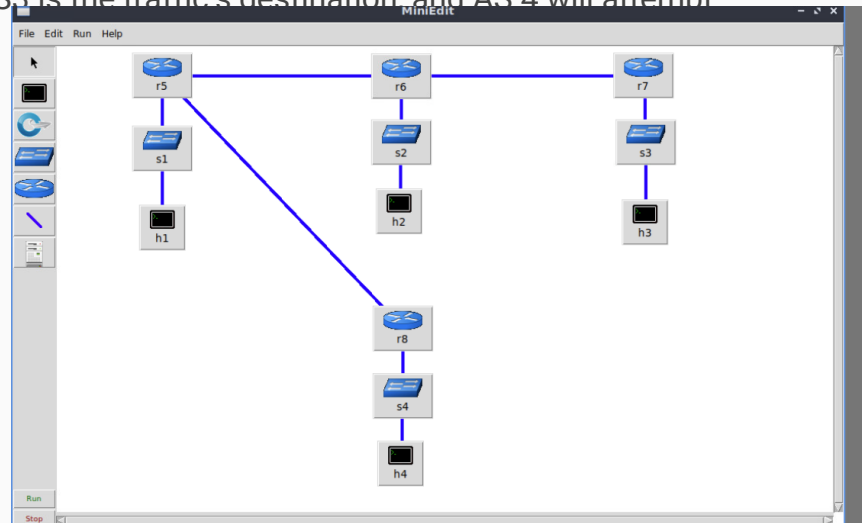
Since BGP is based on a interconnected network of routers that accept updates to IP address from other trusted routers, any change to network security will have to be adopted by every other connected network to totally eliminate the issue of BGP hijacking.

The best a network administrator can do is to install protocols that will discriminate against IP addresses that were not assigned by a ISP. To understand this, our team will first start with showing the user how to conduct a BGP attack

How it works

Step 1

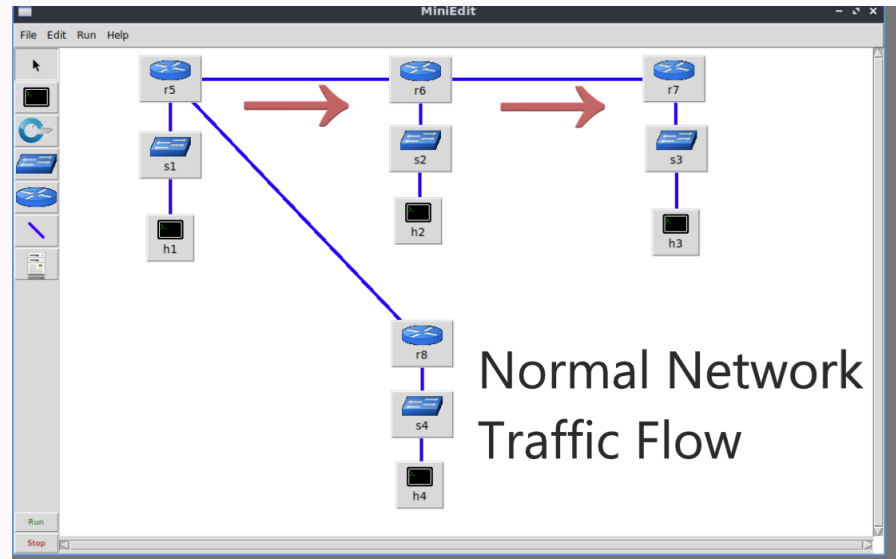
Using a Mininet Virtual Machine on <http://netlab.cec.sc.edu/>, the user will create a network topology that includes 4 networks or Autonomous Systems (AS). AS1 is where the traffic will start, AS2 is the network that routes local traffic to its destination, AS3 is the traffic's destination, and AS 4 will attempt to steal the traffic by hijacking the route



How it works

Step 2

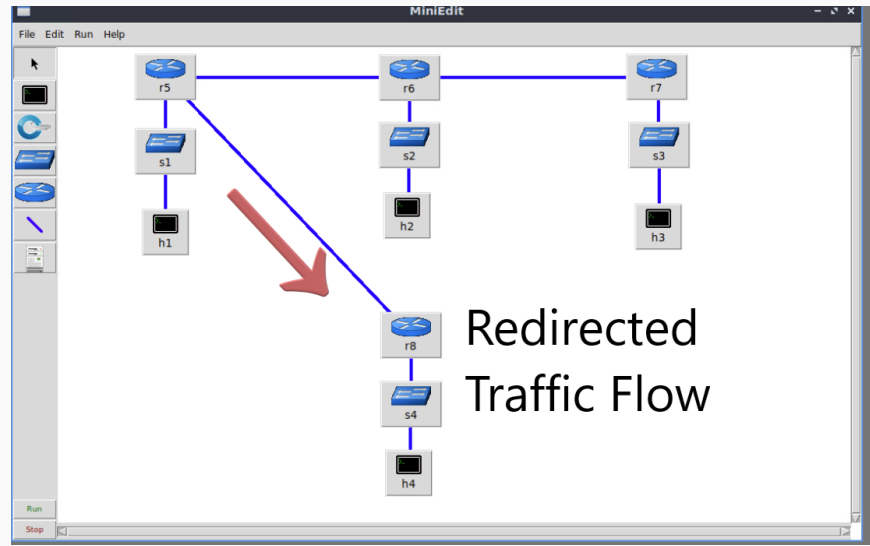
AS1 - AS3 will already be preconfigured for the user. The user will observe the route for the network traffic as intended, before AS4 attempts to steal it.



How it works

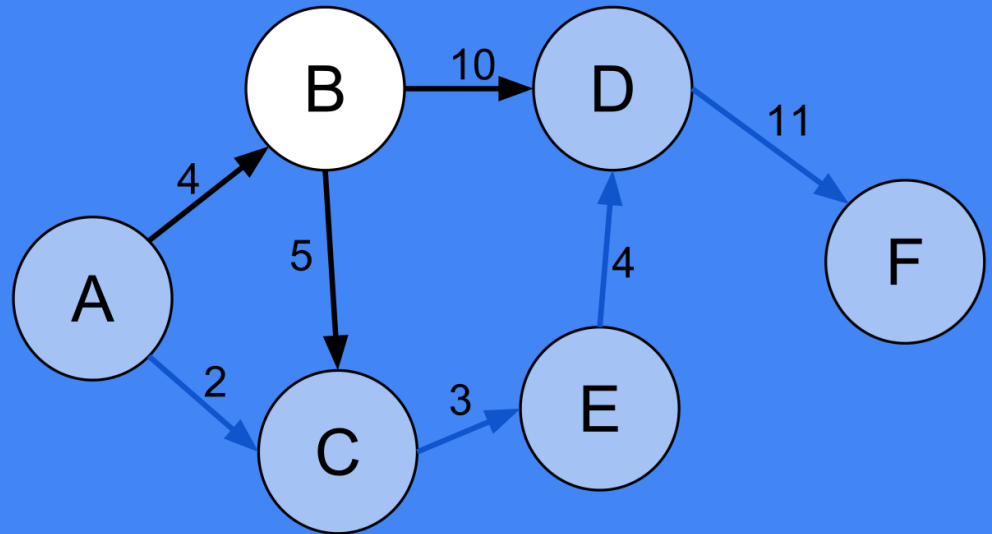
Step 3

Finally the user will be instructed how to configure AS4 so that the network traffic intended for AS3, is hijacked and routed to AS4.



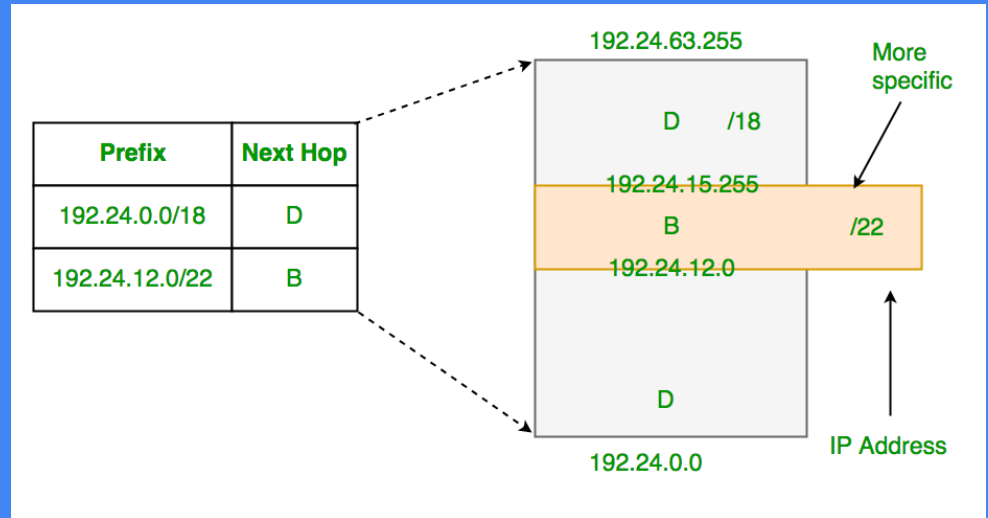
Because of the way the algorithm of BGP works, network traffic will always be routed to the IP address that has the shortest route...

Bottom Line



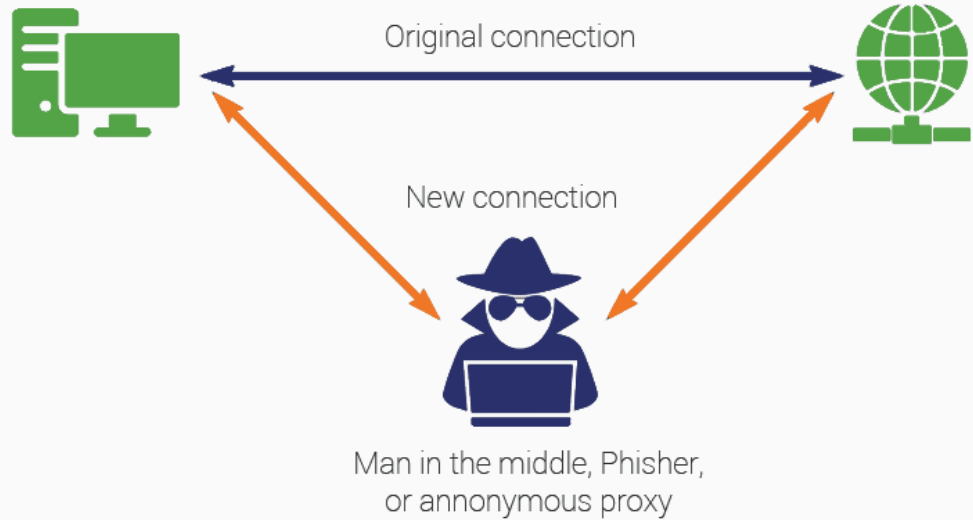
...and the most specific IP prefix.

Bottom Line



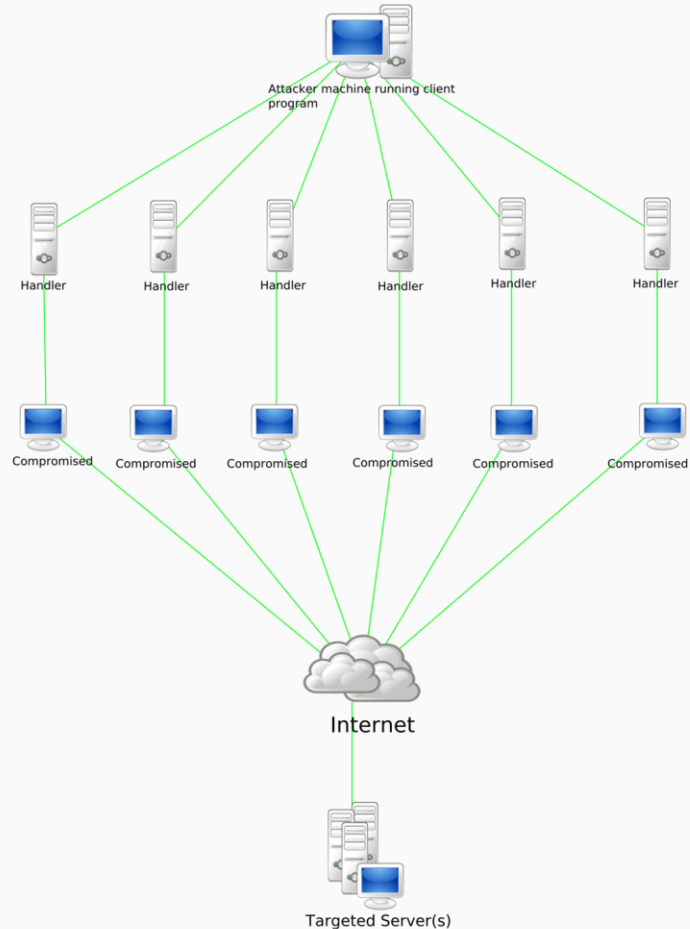
Man-In-the-Middle

Once a successful BGP hijack has occurred, the unsuspecting user can now be conned into delivering confidential information to the bad actor, much like what occurred in the following case study.



Black Hole / Distributed Denial of Service

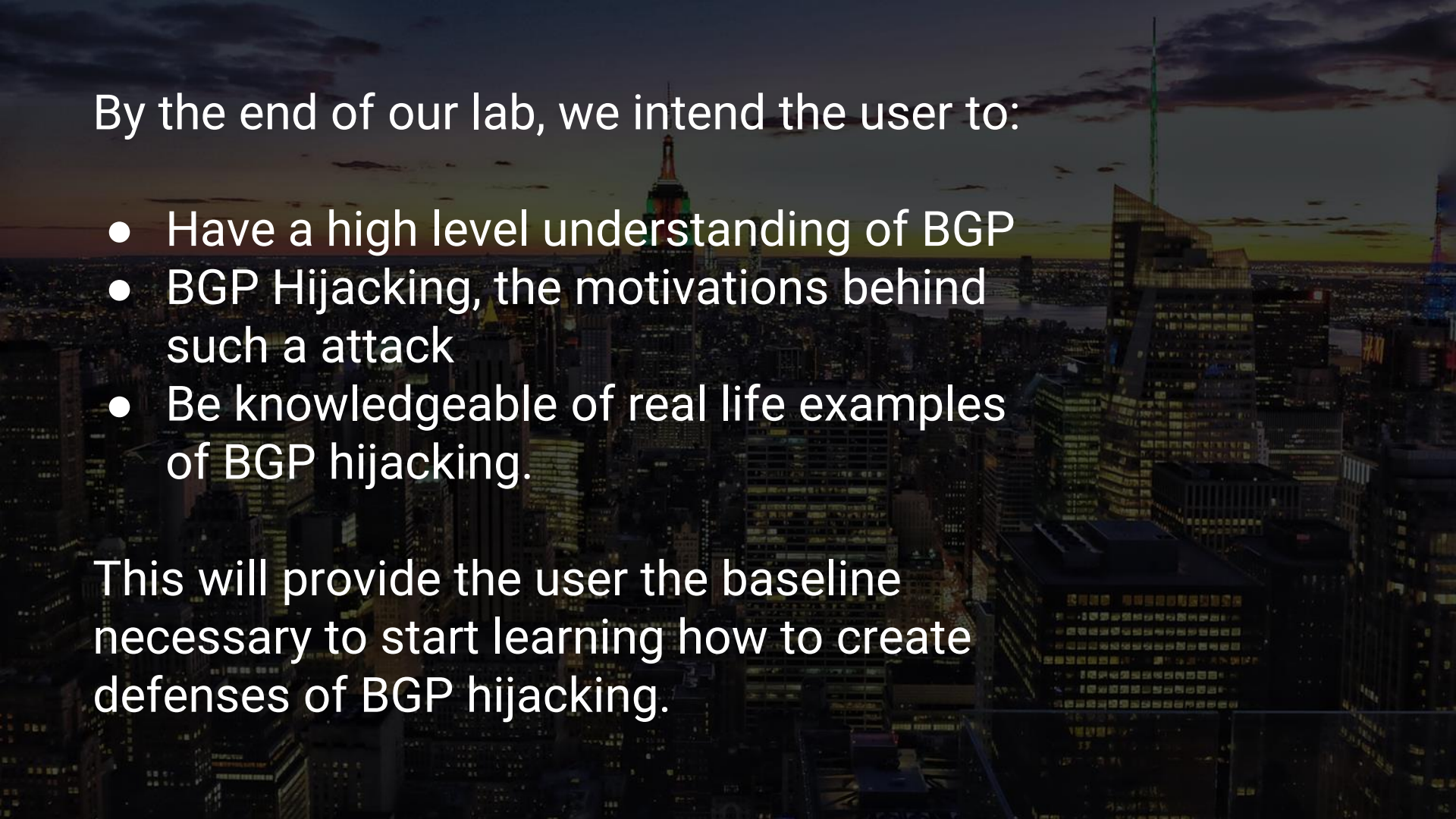
BGP hijacking doesn't always occur on purpose. If configured poorly, a router can accidentally broadcast a bogus IP prefix that hijacks network traffic, just like in the following case study.



Fortunately, conducting these types of hijacks, intentionally or otherwise is quite rare.

This is due to the necessity of attackers to need to control or compromise a BGP-enabled router that between one AS and another, not to mention the IP prefix of the destination AS.





By the end of our lab, we intend the user to:

- Have a high level understanding of BGP
- BGP Hijacking, the motivations behind such a attack
- Be knowledgeable of real life examples of BGP hijacking.

This will provide the user the baseline necessary to start learning how to create defenses of BGP hijacking.

References

- <https://www.cloudflare.com/learning/security/glossary/bgp-hijacking/>
- <https://www.teridion.com/blog/bgp-routing-mind-manrs/>
- https://en.wikipedia.org/wiki/Shortest_path_problem
- <https://www.geeksforgeeks.org/longest-prefix-matching-in-routers/>
- <https://www.thesststore.com/blog/man-in-the-middle-attack-2/>
- https://en.wikipedia.org/wiki/Denial-of-service_attack