Task 5.2C

Student Name

Institutional Affiliation

**Question A:**

Investigate what is “syn cookie” and explain this in your own words. (1 reference is expected, word count is up to 300 words).

SYN cookies is a technique for mitigating attacks whereby the server responds to TCP SYN queries, without adding any new records into its SYN queue, using crafted SYN-ACKs. A new record is only introduced when the customer answers this crafted reply. This technology protects the SYN Queue server from TCP SYN flooding.

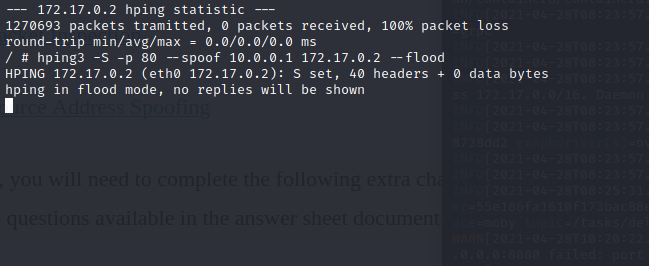
The client sends a SYN during normal service, and the server responds with a SYN+ACK request, and the server keeps state information in the TCP stack while awaiting the client ACK message. SIN packets create a single flood using suitable software that consumes all available TCP memory since the server needs to keep state for all half-open connections. A simple SYN flood is created. As this table is finite, the server no longer accepts new TCP connections and consequently fails or denies the user's operation (Tshang & Liu, 2019). It is highly leveraged since a very few bandwidths and CPUs can use a wide range of servers to exhaust resources.

The server does not need to maintain this status table by explicitly calculating the TCP sequence number with a special secret math function in the SYN-ACK response. The TCP sequence number is tested on reception of the ACK by the customer to decide if the response is valid. If the check is successful, the server creates the TCP session and the user login continues as normal. The TCP sequence number is usually a random choice at the beginning of a TCP sequence.

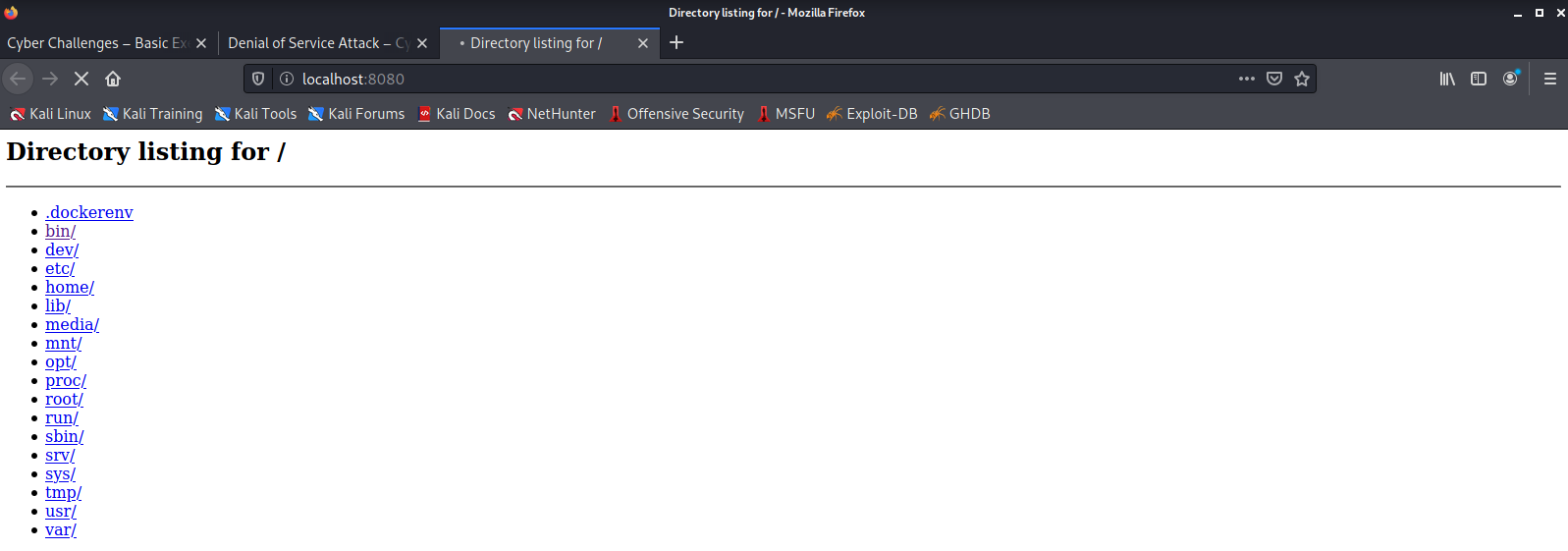
To protect from a SYN flood you don't have to use SYN cookies because the state table is monitored and disabled by most modern firewalls once a high water mark is reached. Smarter firewalls, of course, will examine SYN packets per protocol a second and launch an attack plus purge half-open links to insure the availability of resources. But sometimes they don't have smart routines and can actually discard good TCP sessions, particularly with high-volume attacks), causing a degraded service as the attack goes on.

**Question B:**

Include a screenshot of the output in the first Terminal showing the huge amount of flooding packets after the attack is executed against the server.



The website starts hanging and cannot move to another tab:



A common interview question these days is about Mirai Botnet. Investigate this Botnet and in your own words explain how it worked and what was its impact. Is this Botnet still affecting IoT devices? Please ensure that you use references for your answer. The suggested word count is 400-500 words.

On 12 October 2016, most of the Internet was left inaccessible on the U.S. East coast by a major distributed denial of service (DDoS) attack. The assault, the work of a hostile nation-state, that the authorities feared initially, was in fact the work of the botnet of Mirai.

This assault, which at first had much less goals, became more effective than ever possible its developers dreamed of. The first wave of attacks on Mirai occurred on 19 September 2016 and was used against OVH which hosted common tool Minecraft servers use to combat DDoS attacks, as it was later seen (Jerman-Blazic & Klobucar, 2020). A few days later, the Mirai botnet code was published by "Anna-Senpai." Mirai had infected more than 65 thousand IoT devices by the end of the first day. On the second day, Mirai was already responsible, as seen in the figure above, for half of all Internet telnet scans that our honeypots have observed. Mirai infected more than 600,000 IoT devices at its peak in November 2016 (Fujita & Selamat, 2019).

Mirai, in its centre, is a self-propagating worm, which is a malicious program, which is reproduced by searching for and targeting compromised IoT devices and infecting them. The infected device is also considered a botnet, as it is managed through the central control servers (C&C). These servers say which locations to attack the infected devices next. Mirai consists, overall, of two main parts: the module of replication and the module of attack.

The replication module enslaves as many vulnerable IoT devices as possible to increase botnet size. This is achieved (allegedly) by searching and attacking the whole Internet for viable aims. The module reports it onto C&C servers once it compromises a compromised system to infect the new Mirai payload, as seen in the diagram above. The initial version from Mirai relied solely on a fixed collection of 64 widely used IoT devices for compromising devices. While this attack was very low technology, it proved highly successful and resulted in over 600,000 devices being compromised.

The attack module is responsible for performing DDoS attacks on the C&C server targets. The majority of DDoS codes, including HTTP flooding, UDP flooding and all TCP flood options, are implemented in this module (Burkart & McCourt, Patrick Burkart; Tom McCourt). Mirai has carried out volumetric attacks, layer attacks and TCP attacks in terms of exhaustion across this broad variety of methods.

It is highly likely that the proliferation of uncertain IoT devices on the Internet will be the key cause for DDoS attacks for the near future. If IoT vendors begin to adopt simple best safety practises, the Mirai and subsequent IoT botnets can be prevented.

# **References**

Burkart, P., & McCourt, T. (Patrick Burkart; Tom McCourt). *Why Hackers Win: Power and Disruption in the Network Society.* Univ of California Press.

Fujita, H., & Selamat, A. (2019). *Advancing Technology Industrialization Through Intelligent Software Methodologies, Tools and Techniques.* IOS Press.

Jerman-Blazic, B., & Klobucar, T. (2020). *Advanced Communications and Multimedia Security.* Springer.

Tshang, L., & Liu, H. (2019). *Journal of Communications and Networks.* Korean Institute of Communication Sciences.