

# BİLGİSAYAR AĞLARI VE TASARIMI

## GNS3 ile topoloji oluşturup, sanal makineler arası DDOS atağı ve trafik analizi

*Gazi Üniversitesi*

*Grup Üyeleri: Berke Güneş-181816064 && Selim Savaş-181816070*

### Özet

GNS3 uygulamasında basit bir ağ topolojisi oluşturup, bu topoloji içerisinde VirtualBox'un içine kurduğumuz sanal makinelerimiz arasında hping3 paketini kullanarak DDOS saldırısı yaptık ve saldırıya uğrayan makinemizde, sFlow ve nload paketlerini yardımıyla trafik analizini gerçekleştirdik.

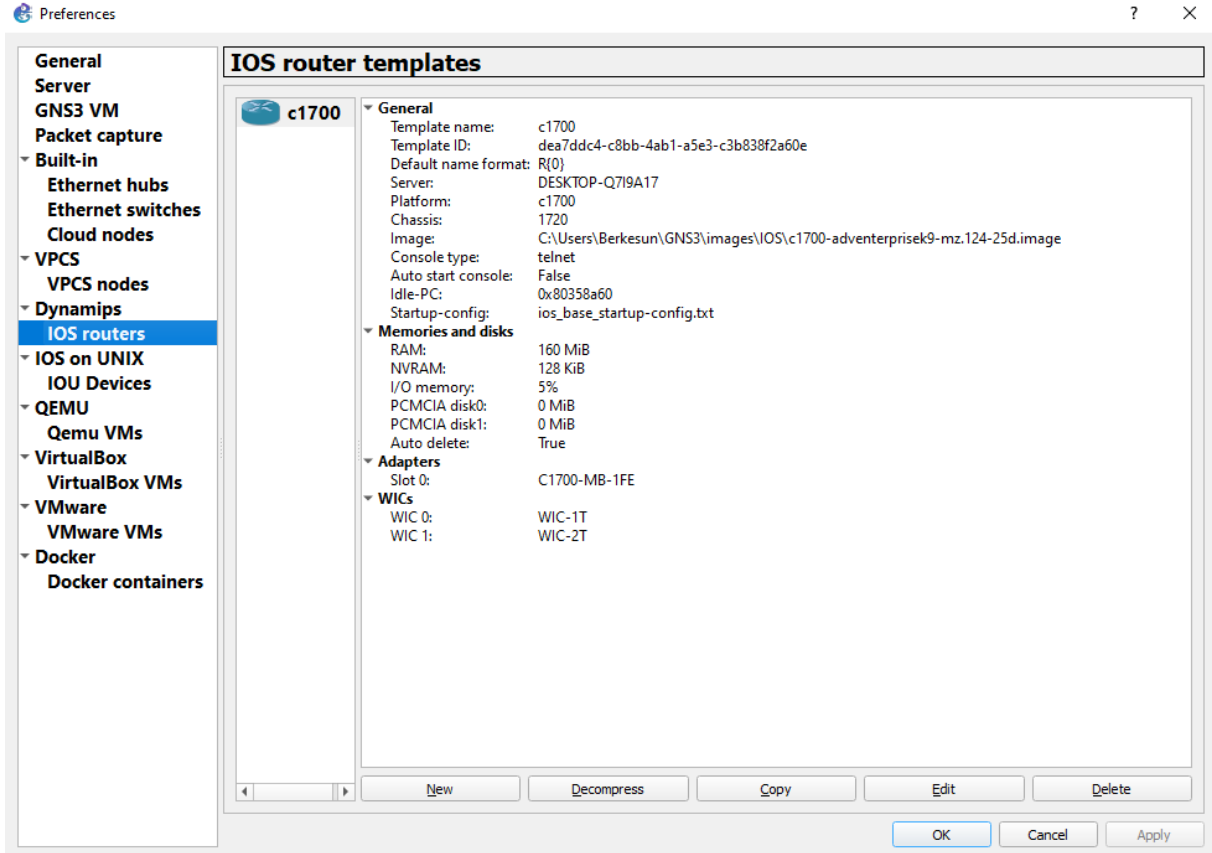
**Anahtar Kelimeler:** GNS3, VirtualBox, DDOS, hping3, sFlow, nload

## 1. GİRİŞ

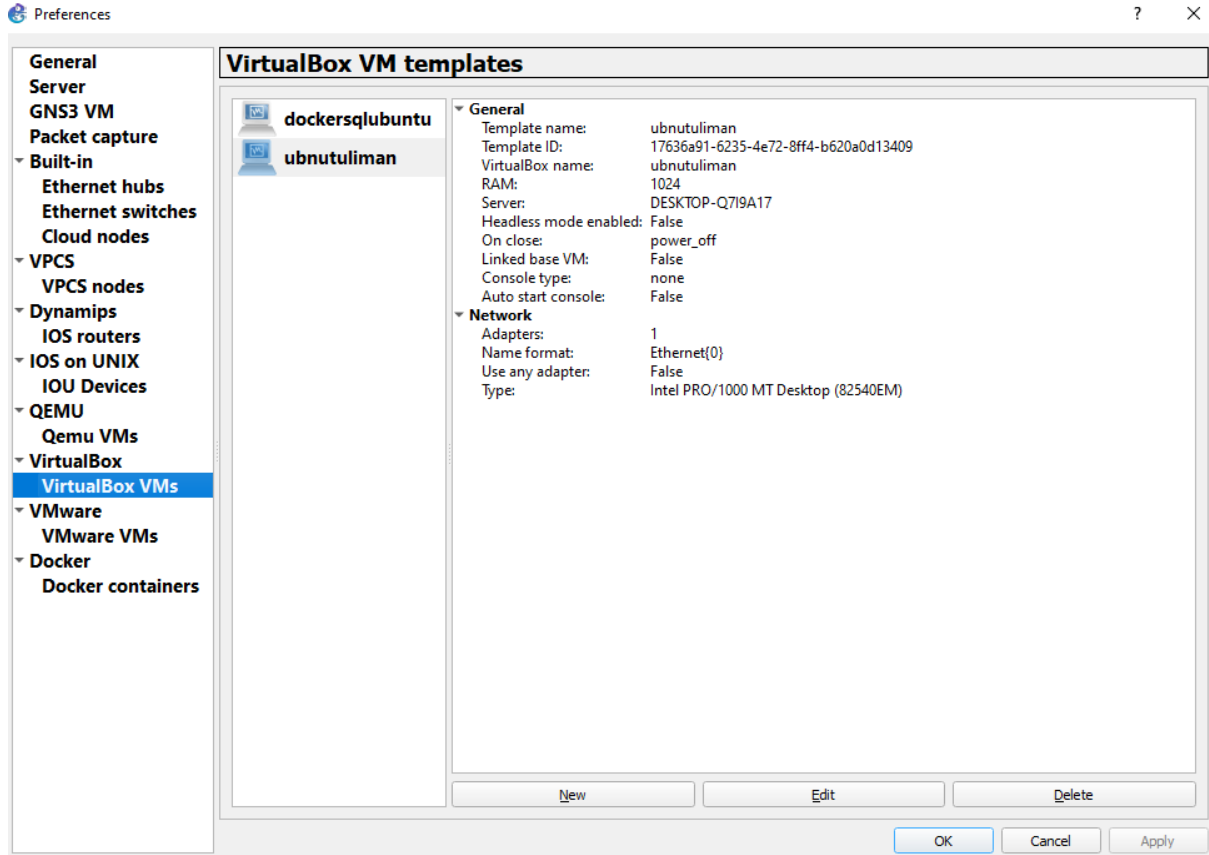
GNS3 uygulamasında gerçekleştirdiğimiz topolojinin içerisindeki router, Cisco Packet Tracer'a ait bir router ISO dosyasıdır. Router konfigürasyonlarının ardından bu router'ı kullanarak oluşturduğumuz topolojideki bilgisayarlar VirtualBox üzerinde kurduğumuz iki adet Ubuntu makineleridir. Saldırgan ve saldırıya uğrayan olmak üzere olan bu iki makinelerde statik ip ataması yapıp belirli paketler aracılığıyla DDOS saldırısı gerçekleştirdik ve bu saldırıyı belli paketler üzerinde görüntüleyip analiz etme fırsatı yarattık.

## 2. GNS3'de topoloji oluşturma

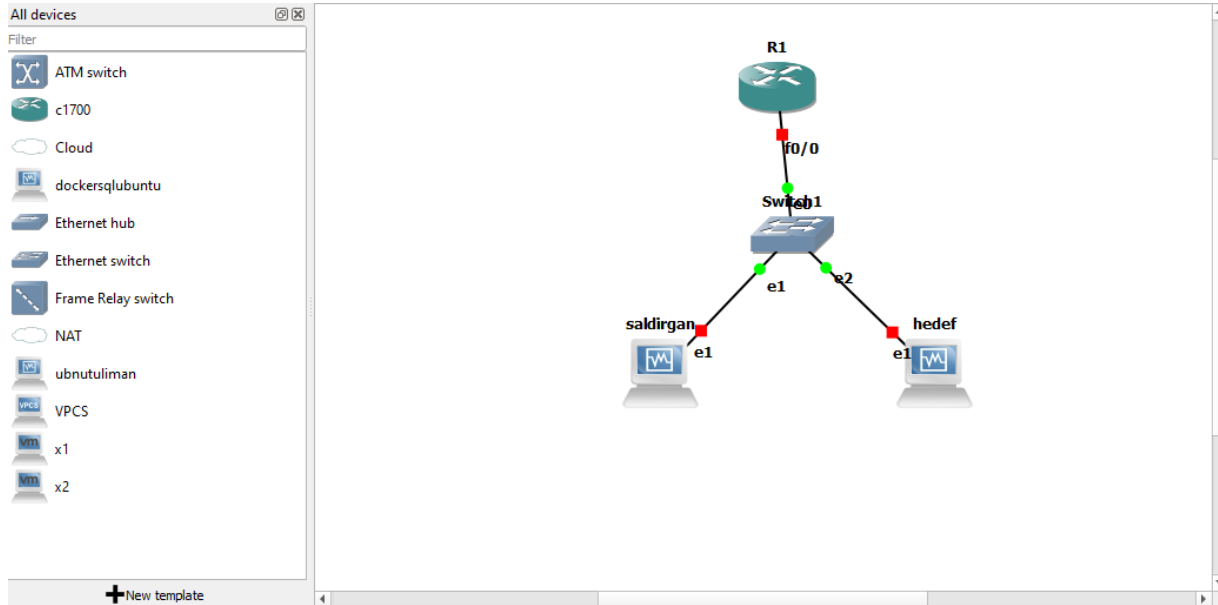
Daha önceden indirdiğimiz Router ISO'sunu[1] aşağıdaki ekrandan Edit —> Preferences kısmından ekliyoruz.



Yine aynı şekilde Edit —> Preferences kısmından VirtualBox üzerinden oluşturduğumuz sanal makinelerimizi tanımlıyoruz.



Gerekli eklemeleri yaptıktan sonra topolojiyi oluşturmaya başlayabiliriz.



Şekilde gördüğümüz gibi Router, Switch ve sanal makinelerimizi ekledikten sonra ilgili portlarla kablo bağlantımızı yaptıktan sonra gerekli Router ve Switch konfigürasyonlarına göz atalım.

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface fastethernet0
R1(config-if)#ip add
R1(config-if)#ip address 192.168.43.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#end
R1#
*Mar  1 00:02:42.827: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

Router için ayarladığımız IP konfigürasyonu bu şekilde sırada Switch portları için ayarlamalara bakalım.

Node properties

### Switch1 configuration

General

Name: Switch1

Console type: none

Settings

Port: 3

VLAN: 1

Type: access

QinQ EtherType: 0x8100

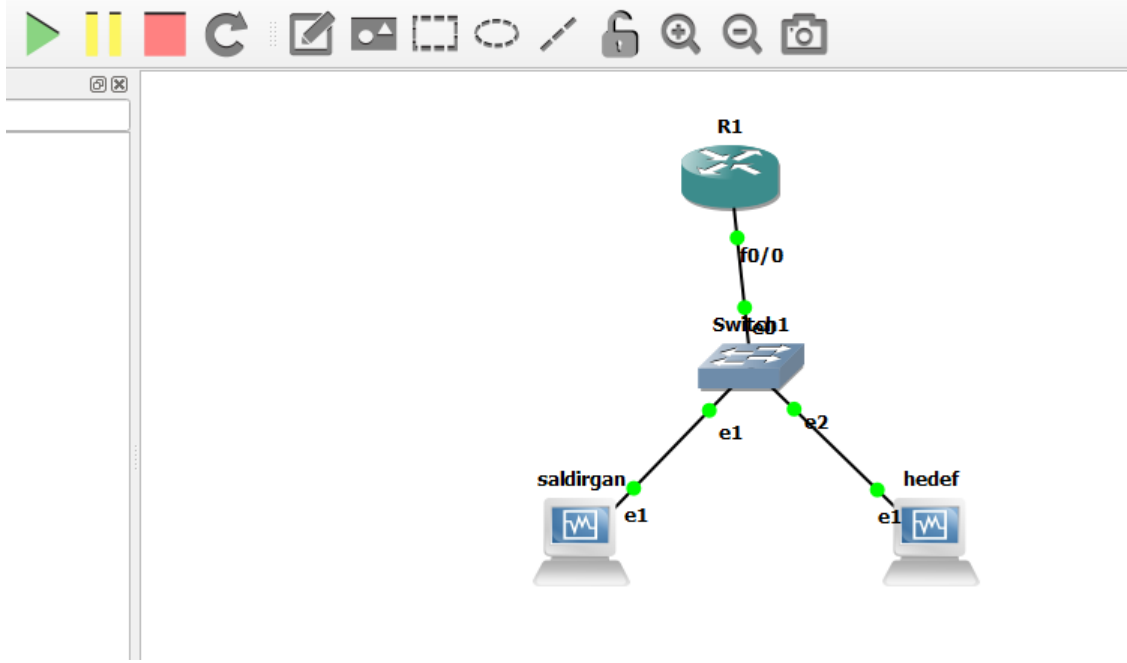
Ports

Port	VLAN	Type	EtherType
0	1	access	
1	1	qinq	0x88A8
2	1	qinq	0x88A8

Buttons: Add, Delete, Reset, OK, Cancel, Apply

Sanal makinelere bağlı olan Switch portlarını qinq şeklinde düzenledik.(Makineler arası ping alışverişi sağlanabilmesi için.)

Gerekli ayarlamaları yaptıktan sonra ağ topolojisi çalışmaya başlıyor.



Şimdi sanal makinelerimizde bulunan interface ayarlarına göz atalım.

### 3. Sanal Makinelerde IP konfigirasyonları

Bu uygulamada daha önce kurulumu gerçekleştirdiğimiz iki adet Ubuntu işletim sistemli sanal makinelerimizi kullanacağız.

```
PS C:\Users\Berkesun> ssh berkesun@192.168.43.3
berkesun@192.168.43.3's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-91-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Thu 06 Jan 2022 11:22:33 PM UTC

System load:          0.32
Usage of /:            70.5% of 8.79GB
Memory usage:         49%
Swap usage:           0%
Processes:            147
Users logged in:      1
IPv4 address for br-5c5ca2ce6c71: 172.18.0.1
IPv4 address for docker0: 172.17.0.1
IPv4 address for enp0s3: 192.168.43.3

11 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

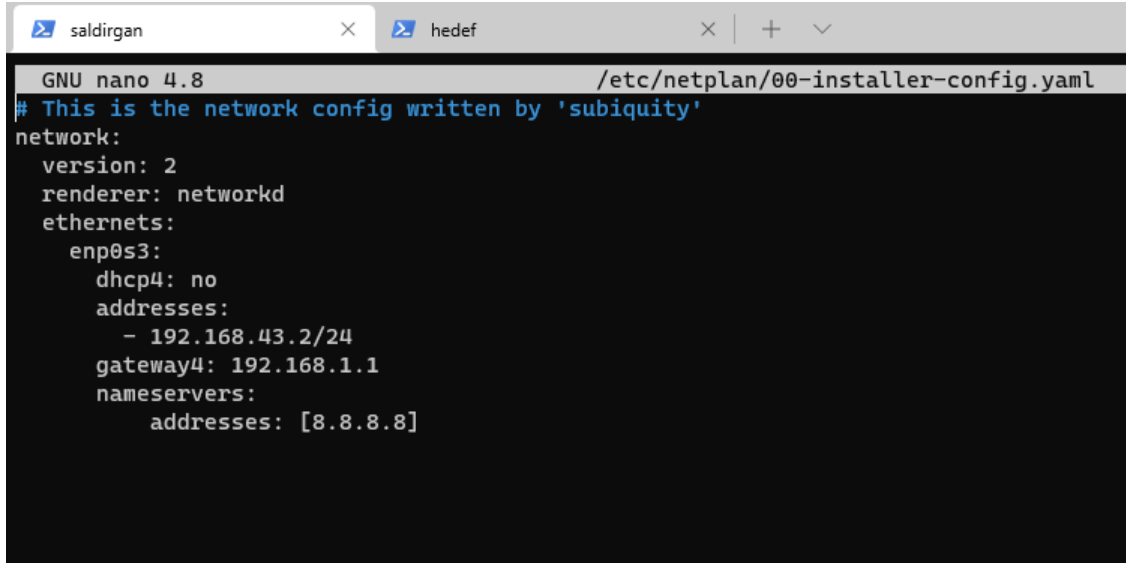
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Thu Jan  6 23:21:25 2022
berkesun@sysadmin:~$
```

Topolojiyi çalıştırdıktan sonra sanal makinelerimiz açıldı ben daha kolay çalışabilmek adına ssh bağlantısı kullanarak Powershell üzerinden gerekli komutları ve ayarları sizlerle paylaşacağım.

İlk önce Saldırgan makinesindeki IP ayarlarına göz atalım.

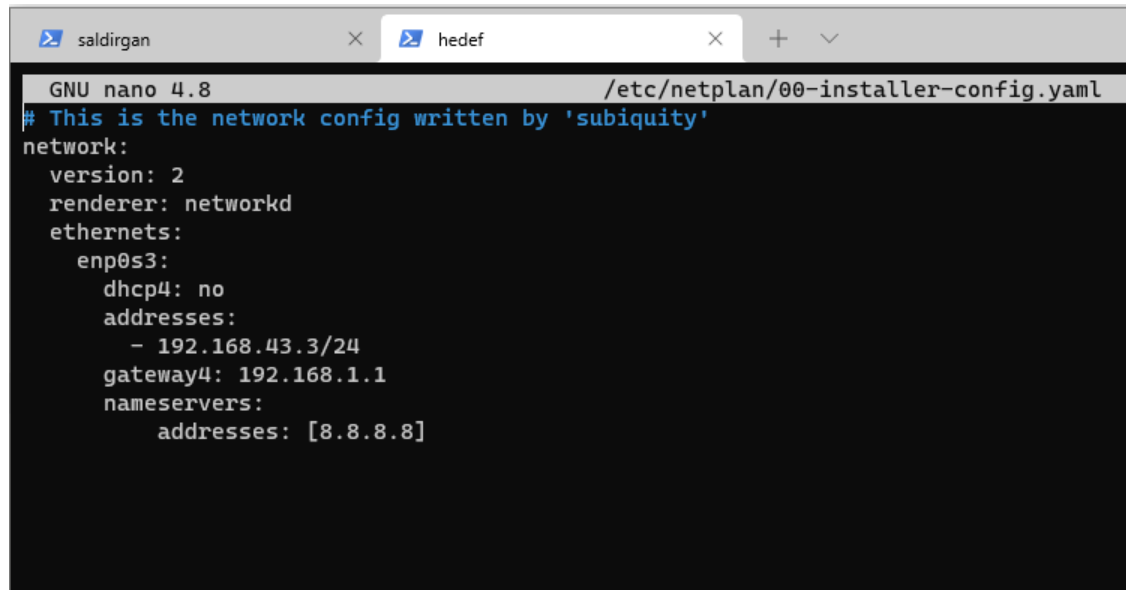
```
sudo nano /etc/netplan/00-installer-config.yaml
```



The screenshot shows a terminal window with two tabs: 'saldirgan' and 'hedef'. The 'saldirgan' tab is active, displaying the contents of the file `/etc/netplan/00-installer-config.yaml` using GNU nano 4.8. The configuration is as follows:

```
GNU nano 4.8 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.43.2/24
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8]
```

Hedef makinemizin IP ayarları ise,



The screenshot shows a terminal window with two tabs: 'saldirgan' and 'hedef'. The 'hedef' tab is active, displaying the contents of the file `/etc/netplan/00-installer-config.yaml` using GNU nano 4.8. The configuration is as follows:

```
GNU nano 4.8 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.43.3/24
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8]
```

IP atamaları yapıldıktan sonra makineler ve router arası ping alışverişinin gerçekleşip gerçekleşmediğini kontrol edelim.

```
ping 192.168.43.1 //Router PING
ping 192.168.43.3 //Hedef makineye PING
```

```
berkesun@berkesun:/etc/netplan$ ping 192.168.43.1
PING 192.168.43.1 (192.168.43.1) 56(84) bytes of data.
64 bytes from 192.168.43.1: icmp_seq=1 ttl=64 time=4.65 ms
64 bytes from 192.168.43.1: icmp_seq=2 ttl=64 time=4.09 ms
64 bytes from 192.168.43.1: icmp_seq=3 ttl=64 time=13.7 ms
64 bytes from 192.168.43.1: icmp_seq=4 ttl=64 time=68.1 ms
^C
--- 192.168.43.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 4.094/22.644/68.130/26.536 ms
```

```
berkesun@berkesun:/etc/netplan$ ping 192.168.43.3
PING 192.168.43.3 (192.168.43.3) 56(84) bytes of data.
64 bytes from 192.168.43.3: icmp_seq=1 ttl=64 time=0.604 ms
64 bytes from 192.168.43.3: icmp_seq=2 ttl=64 time=0.394 ms
64 bytes from 192.168.43.3: icmp_seq=3 ttl=64 time=0.298 ms
64 bytes from 192.168.43.3: icmp_seq=4 ttl=64 time=0.288 ms
^C
--- 192.168.43.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3080ms
rtt min/avg/max/mdev = 0.288/0.396/0.604/0.127 ms
```

Makineler ve Router arasında haberleşme sağlanmakta. Şimdi ilgili paket kurulumlarına göz atalım.

Saldırgan makinemizde kullanacağımız hping3 paketi Ubuntu 20.xx sürümlerinde yüklü olarak gelmekte.

Eğer yüklü değilse:

```
sudo apt install hping3
```

Hedef makinemizde kullanacağımız sFlow[2] ve nload[3] paketlerinin kurulum komutlarına göz atalım.

sFlow:

```
wget https://github.com/sflow/host-sflow/releases/download/v2.0.25-3/hsflowd-ubuntu18_2.0.25-3_amd64.deb
sudo dpkg -i hsflowd-ubuntu18_2.0.25-3_amd64.deb
sudo systemctl enable hsflowd

sflow {
    collector { Makine-IP }
    pcap { speed=1G-1T }
    tcp { }
    systemd { }
}
/etc/hsflowd.conf
sudo systemctl restart hsflowd

sudo apt install openjdk-11-jre-headless
LATEST=`wget -qO - https://inmon.com/products/sFlow-RT/latest.txt`
wget https://inmon.com/products/sFlow-RT/sflow-rt_${LATEST}.deb
sudo dpkg -i sflow-rt_${LATEST}.deb
sudo /usr/local/sflow-rt/get-app.sh sflow-rt browse-metrics
sudo /usr/local/sflow-rt/get-app.sh sflow-rt browse-flows
sudo /usr/local/sflow-rt/get-app.sh sflow-rt prometheus
sudo systemctl enable sflow-rt
sudo systemctl start sflow-rt
```

```
sudo ufw allow 6343/udp
sudo ufw allow 8008/tcp
//usr/local/sflow-rt/conf.d/sflow-rt.conf
```

nload:

```
sudo apt install nload
```

İlgili paketlerin kurulumunun ardından sırada DDOS atma işlemine göz atalım.

#### 4. DDOS Saldırısı

Tüm gerekli ayarlamaları yazdıktan sonra DDOS saldırısı için yapmamız gereken bize uygun hping3 komutunu oluşturabilmek. Ben daha öncesinde taslak amaçlı hazırladığım komutu sizlerle paylaşacağım. hping3 ile ilgili gerekli komutları “hping3 ?” ile öğrenebilir kendinize göre DOS ve DDOS saldırıları için komutlar oluşturabilirsiniz.

```
hping3 -i u1 -S -p PORT --flood MAKINA-IP -o 64 -w 64 -c 64 -d 654 -t 64 --tcp-timestamp
```

Yukarıdaki komutu saldırgan bilgisayarımızda sudo yetkisi ile çalıştırıyoruz.

```
berkesun@berkesun:/etc/netplan$ sudo hping3 -i u1 -S -p 80 --flood 192.168.43.3 -o 64 -w 64 -c 64 -d 654 -t 64 --tcp-timestamp
HPING 192.168.43.3 (enp0s3 192.168.43.3): S set, 40 headers + 654 data bytes
hping in flood mode, no replies will be shown
```

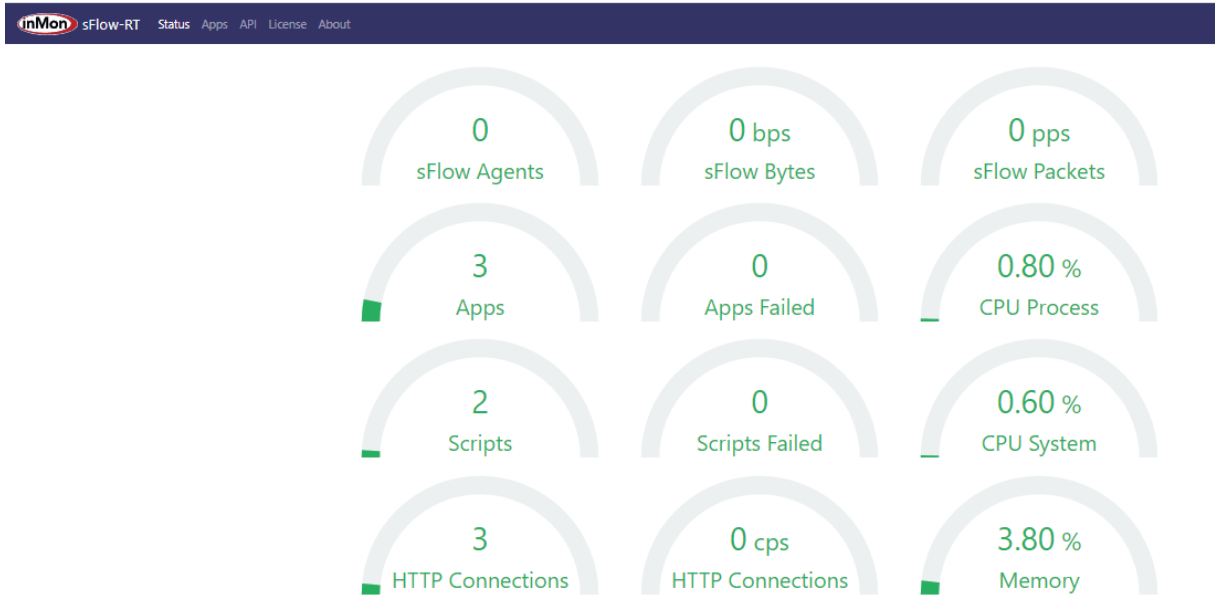
Görüldüğü üzere saldırı başladı şimdi hedef makinemizde trafik ve donanım hareketlerine göz atalım.

#### 5. Hedef Makinede Trafik Analizi

Trafik analizine başlamadan önce hedef makinede DDOS saldırısına uğramadan önceki trafik ve donanım kullanım hareketlerine göz atalım.

sFlow:





nload:

```
Device enp0s3 [192.168.43.3] (1/1):
=====
Incoming:

Curr: 944.00 Bit/s
Avg: 4.17 kBit/s
Min: 944.00 Bit/s
Max: 6.49 kBit/s
Ttl: 76.00 MByte

Outgoing:

Curr: 8.66 kBit/s
Avg: 17.72 kBit/s
Min: 8.41 kBit/s
Max: 27.30 kBit/s
Ttl: 6.47 MByte
```

Şimdi DDOS saldırısını başlatalım ve son duruma göz atalım.

sFlow:



## 6. SONUÇ

Görüldüğü üzere GNS3 üzerinde kurduğumuz topolojiyi kullanarak belirli paketler üzerinde makineler arası haberleşme ve DDOS saldırısında bulunduk. Sistem ve ağ tarafında bir çok özelliği kullandığımız bu uygulamada daha fazla verim alabilmek adına dilerseniz daha güçlü hping3 komutları, daha iyi görüntü alabilmek adına sFlow'a alternatif olarak netFlow kullanabilirsiniz. Ayrıca GNS3 üzerinde veya Linux işletim sistemleri üzerinde firewall oluşturup DDOS saldırısını koruyabilir, hangi portları güvence altına alabileceğinizi seçebilirsiniz.

## 7. KAYNAKÇA

[1][https://drive.google.com/file/d/1mbDaZ1Z1Kq\\_ik7coaVPVjvYGo6KclkbJ/view](https://drive.google.com/file/d/1mbDaZ1Z1Kq_ik7coaVPVjvYGo6KclkbJ/view)

[2]<https://blog.sflow.com/2020/03/ubuntu-1804.html>

[3]<https://www.tecmint.com/nload-monitor-linux-network-traffic-bandwidth-usage/>