Nirbhay Sharma (B19CSE114)

Embedded Systems - Assignment - 2

```
$ Programming Language chosen - Python, Java
$ Two attacks are modelled -
    - Dos attack
    - Parity change attack
```

1. **Modelling of attacks**
    - *DOS attack*
        1. the first attack is DOS attack which is modelled using client server architecture, since here we are talking about peripherals and cpu type of interaction, so we can consider client as peripheral and server as cpu which takes data from client.
        2. the DOS attack is modelled by providing many client request to server which eventually increase load on the system and makes it slow,
    - *Parity change attack*
        1. This is the second attack that has been modelled in Java using classes and objects concept
        2. the attack goes like this - suppose there is one distribution channel which is used to transfer data from source to destination, and suppose that mallory/adversary has got control over the channel and now he/she can change any bit of the message but at the sender side we can check for change in bit using parity bit which is calculated as xor of all the bits.
        3. if the attack has made server will respond by providing a warning message
2. **Explanation of attack through code output**
    - *DOS attack*

> Before attack cpu's condition

```
sharma406@LAPTOP-N4BIN1J0:/mnt/c/Users/Nirbhay sharma$ iostat
Linux 5.10.16.3-microsoft-standard-WSL2 (LAPTOP-N4BIN1J0)       02/18/22        _x86_64_        (12 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.19    0.03    0.15    0.01    0.00   99.61

Device             tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
sda               4.60         1.40      1961.38         0.00        225     315920          0
sdb              44.55      1375.21       471.92       326.12     221505      76012      52528
```

> After attack cpu's condition

```
sharma406@LAPTOP-N4BIN1J0:/mnt/c/Users/Nirbhay sharma$ iostat
Linux 5.10.16.3-microsoft-standard-WSL2 (LAPTOP-N4BIN1J0)       02/18/22        _x86_64_        (12 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.82    0.02    0.62    0.01    0.00   98.53

Device             tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
sda               4.53         1.17      1962.15         0.00        225     377420          0
sdb              37.31      1152.74       395.18       273.09     221729      76012      52528
```

- *Parity change attack*

```
$ javac ParityProtection.java ; java ParityProtection
sender ready to send data
Destination is waiting for message
enter msg to send: 000111
someone has changed the message
enter msg to send: 00111
message received
enter msg to send: 00111100111
message received
enter msg to send: 00101010011
message received
enter msg to send: 0101010010101
someone has changed the message
enter msg to send:
```

- In first case we can see the cpu utilization after attack, since the programme is sending so many requests to the server and each time we are creating new thread for the different client the cpu utilization increase (we can see cpu utilization by %user column which is increased from 0.19% to 0.82%)
- In the second attack we can see that sender is sending the data to destination via a channel which is hacked by adversary and he/she can alter the msg (randomly) but due to parity protection bit we are able to get catch any alteration in message.

3. **I/O for the above attacks**

- **For DOS**

```
---Input---
> for first program ubuntu is required
> First run the server2.py file on terminal
> Then run the Load server file on new terminal

$ python3 server2.py 8895
$ python3 LoadServer.py 127.0.0.1 8895 3000

where syntax for running server2 file is:
> python3 server2.py <PORT>
syntax for running LoadServer.py fiel is:
> python3 LoadServer.py <IP> <PORT> <Iterations> (To load the server)

---Output---
on server2.py side we can see a very huge queries from various clients
and the cpu uitlization got increased.
we can use iostat to see the cpu utilization as well (%user column will
tell us that how much cpu is utilized by user program)
```

- **For Parity change attack**

```
---Input---
> Run the java file provided using the following command

$ javac ParityProtection.java ; java ParityProtection

then enter the input only in binary format (like 1010101 etc.)

---Output---
We can see the response from destination ,that msg is altered or not.
```

4. **Practicality of above attacks**

   - **DOS**

     - The attack is practical in real world scenarios, here just a high level modelling is given but in real scenarios actual servers can get crashed by this type of attack, here also the number of iterations are kept less (around 3000) which can be easily handled by cpu's now a days but if we could give a huge number like 3000000 then there are very high chances that CPU utilization increases very much.
     - The modelling is not 100% accurate as compared to real world scenarios but is very close to accurate as client server functionality is there and an outsider is trying to overload the server with many clinet requests. So design is very close to accurate modelling.

   - **Parity change attack**

     - This type of attack is possible when we are transfering the data from one place to another via a channel that we consider as secure but it is actually not, and hence there are very high chance of message alteration attack by adversary in real scenarios as well.
     - The modelling is not 100% accurate as compared to real world scenarios but it is mimicing that sender, channel, receiver and also the parity bits are used to check for the correctness of message. Hence modelling is mimicing real world scenarios nicely.