

Configuration Manual

MSc Research Project
Masters in Cybersecurity

Rishabh Vyas
Student ID: 21108714

School of Computing
National College of Ireland

Supervisor: Ross Spelman

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Rishabh Vyas
Student ID: 21108714
Programme: MSc. in Cybersecurity **Year:** 2021-2022
Module: MSc Research Project/Internship (MSCCYB1)
Lecturer: Ross Spelman
Submission Due Date: 15/12/2022
Project Title: Implementation of Blockchain Technology in IoT environment to mitigate DDoS attacks
Word Count: 1740 **Page Count:** 3

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

Date:

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Rishabh Vyas
Student ID: 21108714

1 VirtualBox VM

The first part for the configuration of the project requires the installation of a VirtualBox VM. In the project, I have used the Oracle VirtualBox VM for the setup of Ubuntu 6.1 (VirtualBox.org, n.d.) which is in addition to being a powerful x86 and AMD64/Intel64 virtualization platform, VirtualBox can be used both at home and in the enterprise. Once the installation of Ubuntu is complete, we move onto the next section.

2 Software's Used

For the configuration of the project, we have used multiple software's, of which, some are cloned from the github website such as:

- Mininet (Mininet: An Instant Virtual Network on your Laptop, 2022): This is used to simulate a virtual SDN domain with controllers and hosts. We can also create multiple SDN domain comprising of numerous controllers as well as hosts.
- sFlow-rt (inmon: sFlow-RT, 2022): sFlow-rt is connected to the mininet via a sflow agent which helps in monitoring the flow of packets and for graphical purposes. In the sFlow-rt “app” section we have added apps such as browse-metrics, browse-flows, browse-drop and most importantly mininet-dashboard which helps in viewing the topology of the domain as well as the graphical representation of the flow of packets.
- Floodlight (Floodlight Controller, 2018): Floodlight controller is also connected to the sflow-rt and mininet.
- Ryu controller (component-based software defined networking framework Build SDN Agilely, 2017): Ryu controller and its repositories are connected to the domain to help in the mitigation process of the DDoS attack. This controller gives a detailed view of the packet flow with each of the controller within the domain and also notifies when an attack is detected and detailed view of which of the host is the attacker and which is the victim. (The pre-requisites for Ryu controller should be installed in the python 3 library).
- Truffle-framework (Truffle Suite: Sweet Tools for Smart Contracts, 2022): This software was used to write the Collaboration.sol smart contract that was created for the project.
- Remix-IDE (Vermouth, et al., 2022): An online version of the remix-IDE was used to test the functionalities of the smart contract.
- Ganache (Ganache, 2022): This software was used for noting the gas used for the deployment, creation as well as the functions within the smart-contract on a private blockchain.
- Görli faucet (Alchemy Insights Inc, 2022) : A website which helps in sending fake ETH to a personal wallet, to test the deployment of the smart contract on the official Görli testnet Network

- Metamask (metamask.io, 2022): Metamask was the wallet used to store the ETH.
- Infura: This website was used to help in the deployment of the smart-contract to the testnet.

3 Setup-procedure

Once the installation of all the software and their pre-requisites are done, we can now move onto setting up the project. First, we created the smart-contract using the solidity (Solidity, 2022) language on the truffle-framework. We can view the smart-contract (Collaboration.sol) that was created as provided in the zip file. Once our smart contract is created, we then move onwards to the testing of the functionalities of the smart contract using Remix-IDE which helps in compiling and testing of the smart contract.

Moving on towards the deployment of the smart contract on the official Görli testnet Network, we will be utilising Metamask as well as Görli faucet website and infura. First we send no value ether or fake ETH using the Görli faucet website to our virtual wallet which was created using Metamask. We then create a new project on the infura website so as to get started with the deployment of the smart-contract. Here, once we have selected the Görli as our endpoint we are provided with a link to connect with the truffle-framework for the deployment in the testnet network. The link is then pasted in the truffle-config.js file under the networks function that is present in the file. For the testing purpose we enter our mnemonic key in the same file, although this key should not be shared, optionally we can create a .env file and enter our secret key there and link it to the config file. We must also specify the truffle as our HDWalletProvider in the config file as well to finally complete the configuration for the link. We then install the @truffle/hdwalletprovider using the integrated terminal so that the deployment procedure may begin. We then use the command `truffle --migrate Görli` network to successfully compile and deploy the smart-contract on the testnet network. We can use the value in the transaction hash to verify the deployment is successful on the following URL: <https://goerli.etherscan.io/>. Here we can note the gas used for the creation and deployment of the smart-contract. We then test the functionalities of the smart-contract using our private blockchain simulator Ganache and Remix-IDE. Before the testing of the smart-contract can be performed we must first configure and connect ganache to our contract using the truffle-config.js file. Here again, under the network function we must specify our host IP address port and network id in order to connect with ganache. We must firstly, open our Collaboration.sol file in the Remix-IDE, then if there possess no error in the building and compilation of the smart contract, we can see the transaction for deployment/creation of smart contract on the private blockchain, also, test and compare the gas used. Once we notice a successful transaction on ganache, we can test the various functions of the smart contract using Remix-IDE and monitoring the transactions that occur on the Ganache. This is helpful in noting the gas used for each of the function to be executed and then calculating the total cost that will be required to deploy the project on a public blockchain.

4 SDN-Domain creation

We use the mininet and the Ubuntu terminal for the creation of a virtual SDN topology consisting of two main switches, sA, sB, which are then connected to two sub-switches each named sAA, sAB, sBA, sBB respectively. Each of these sub-switches consist of at least two hosts making four host in domain one connected to controller cA and four in the other connected to the controller cB. Once our virtual SDN domain is created and we have started the sFlow-rt as well as ryu controller, we can use the Controller1.py & Controller2.py file to start the monitoring the packet flow between domain using the Ubuntu terminal. These files

also contain the codes written for the mitigation of inter-domain DDoS attacks. Once we run the file the file helps the Ryu controller in identifying the attacker as well as the victim and when the attacking is taking place it also blocks the host from attacking, once the attack has stopped it again sends a signal to remove the blocking filter on the attacker. The file Controller1.py & Controller2.py also contain method to send the list of the attacker and victim's IP addresses to the smart-contract, but it is not possible to test this feature on a private blockchain network. We can also see the graphical representation of the packet flow in our localhost sFlow-rt in the mininet-dashboard, browse-flow, etc.

5 DDoS Attack & Mitigation

Once our topology is set up, and we have sFlow-rt and Ryu controller monitoring the packet flows we can test for the inter-domain DDoS attacks. For the testing purpose we have only secured the hosts AAh1 & AAh2 under the sub-switch sAA. This is because to test whether the DDoS attack is not mitigated on other hosts. We can simulate a UDP flood DDoS attack using the Hping3 script (Tool Documentation, 2022) onto the host AAh1 originating from host BAh1. As we monitor our Ryu controller within few seconds, we can note that the controller has successfully detected the attacker as BAh1 and added a filter to block all packets that are originating from the attacker. As soon as we, stop the simulation of the DDoS attack we can also note that within a few seconds the Ryu controller has been notified and the filter is removed from the attacker.

6 References

- Alchemy Insights Inc. (2022). *Goerli FAUCET*. Retrieved from Alchemy Goerli FAUCET: <https://goerlifaucet.com/>
- component-based software defined networking framework Build SDN Agilely*. (2017). Retrieved from Ryu-SDN: <https://ryu-sdn.org/>
- Floodlight Controller*. (2018). Retrieved 08 07, 2022, from <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/overview?homepageId=1343545>
- Ganache*. (2022). Retrieved 08 07, 2022, from <https://trufflesuite.com/docs/ganache/inmon:sFlow-RT>. (2022). Retrieved 08 07, 2022, from <https://sflow-rt.com/reference.php>
- metamask.io. (2022). *A crypto wallet & gateway to blockchain apps*. Retrieved from Metamask: <https://metamask.io/>
- Mininet: An Instant Virtual Network on your Laptop*. (2022). Retrieved 08 07, 2022, from <http://mininet.org/>
- Solidity*. (2022). Retrieved 08 07, 2022, from <https://docs.soliditylang.org/en/develop/>
- Tool Documentation*. (2022). Retrieved 08 07, 2022, from <https://www.kali.org/tools/hping3/>
- Truffle Suite: Sweet Tools for Smart Contracts*. (2022). Retrieved 08 07, 2022, from <https://trufflesuite.com/>
- Vermouth, A., Aniket, Disu, D., Mertens, F., Izang, J., Husikyan, L., . . . Levreau, Y. (2022). *Remix Project*. Retrieved from Remix: <https://remix-project.org/>
- VirtualBox.org, O. (n.d.). *Welcome to VirtualBox.org*. Retrieved from VirtualBox: <https://www.virtualbox.org/>