

**IOT Honeynet Framework**

**Timothy McCann**

**Jason Swords**

**Raigridas Bartkus**

*Submitted in part fulfilment for the degree of* *B.Sc. (Hons) Digital Forensics and Cyber Security, School of Informatics Department of Informatics and Engineering Technology University Dublin,*

*Blanchardstown Dublin 15.*

**May 2019**

# **Declaration**

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Degree of Honours B.Sc. in Digital Forensics and Cyber Security in the Technological University Dublin – Blanchardstown Campus, is entirely my own work except where otherwise stated, and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

Signed:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dated: \_\_\_\_/\_\_\_\_\_/\_\_\_\_\_

Signed:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dated: \_\_\_\_/\_\_\_\_\_/\_\_\_\_\_

Signed:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dated: \_\_\_\_/\_\_\_\_\_/\_\_\_\_\_

# **Summary**

# **Acknowledgements**

# **Abstract**

**Contents**

[**Declaration** 2](#_Toc7018451)

[**Summary** 3](#_Toc7018452)

[**Acknowledgements** 4](#_Toc7018453)

[**Abstract** 5](#_Toc7018454)

[**List of Tables** 7](#_Toc7018455)

[**List of Figures** 7](#_Toc7018456)

[**Chapter 1: Introduction** 8](#_Toc7018457)

[**Chapter 2: Current Technologies** 9](#_Toc7018458)

[**Chapter 3: Design** 9](#_Toc7018459)

[**3.1 Hosting Service** 9](#_Toc7018460)

[**3.1.1 Understanding Amazon EC2 Features** 9](#_Toc7018461)

[**3.1.1.1 Setting Up Amazon EC2** 10](#_Toc7018462)

[**3.1.1.2 Virtual Private Cloud (VPC’s)** 13](#_Toc7018463)

[**3.1.1.3 AMI’s and Instances** 13](#_Toc7018464)

[**3.1.1.4 Regions and Availability Zones** 14](#_Toc7018465)

[**3.1.1.5 Amazon EC2 Key Pairs** 14](#_Toc7018466)

[**3.1.1.6 Security Groups** 15](#_Toc7018467)

[**3.1.2 VPC Flowlogs and CloudWatch** 15](#_Toc7018468)

[**Chapter 4: Implementation** 15](#_Toc7018469)

[**AWS** 15](#_Toc7018470)

[**Bibliography** 16](#_Toc7018471)

# **List of Tables**

# **List of Figures**

# **Chapter 1: Introduction**

# **Chapter 2: Current Technologies**

# **Chapter 3: Design**

## **3.1 Hosting Service**

Amazon Web Services (AWS) is a subsidiary of Amazon Inc. and describes itself as a “secure cloud services platform, offering compute power, database storage, content delivery and other functionality” [2]. It was decided to use AWS’s Amazon Elastic Compute Cloud (Amazon EC2) as the hosting service for the project. Amazon EC2 describes itself as being able to provide “scalable computing capacity in the Amazon Web Services (AWS) cloud” [3]. Amazon EC2 allows developers to deploy as many or as few virtual servers as is needed, configure their networking and security, and manage their storage.

Amazon EC2 was chosen due to its flexibility, making it possible to deploy and manage virtual servers on demand. Other hosting providers such as Digiweb, Microsoft Azure and DigitalOcean were considered but Amazon EC2’s flexibility and prestige as being an industry leader in providing cloud hosting services resulted in it being chosen for this project.

### **3.1.1 Understanding Amazon EC2 Features**

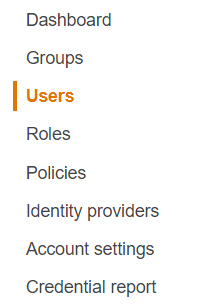
As previously mentioned, Amazon EC2 describes itself as being able to provide scalable computing capacity in the Amazon Web Services (AWS) cloud. Amazon EC2 instances can be provisioned under the AWS 12-month free tier to facilitate the frameworks need to deploy multiple instances. The features that make up Amazon EC2 and their relevance to the project are explored in further detail in the following sections.

It is important to understand the different features provided by Amazon EC2 in order to comprehend the design of the IOT Honeynet Framework and all of its associated features which help researchers and users to seamlessly scan devices and deploy them as honeypots quickly and efficiently. There are a number of requirements that must be satisfied in order to allow such an IOT Honeynet Framework to function properly. It is for this reason, that the features of Amazon EC2 must be explored and understood.

#### **3.1.1.1 Setting Up Amazon EC2**

Using a free tier EC2 instance first involves setting up an AWS user account. Once created, an account has access to all AWS services automatically including Amazon EC2 [1]. With an account now setup, the next step is to create an IAM user. Services in AWS, like Amazon EC2, require users to provide credentials when accessing them so it can be determined whether a user has permission to access that service or not [1]. It is not recommended that an AWS service be accessed using the default AWS account login credentials but rather by using the AWS Identity and Access Management instead, which is necessary to allow services to accessed programmatically [1].

Adding a user involves a few simple steps. Once logged into an AWS account, launch the Identity and Access Management console. Click *“Users”* on the navigation menu on the left of the screen as seen in the below figure.



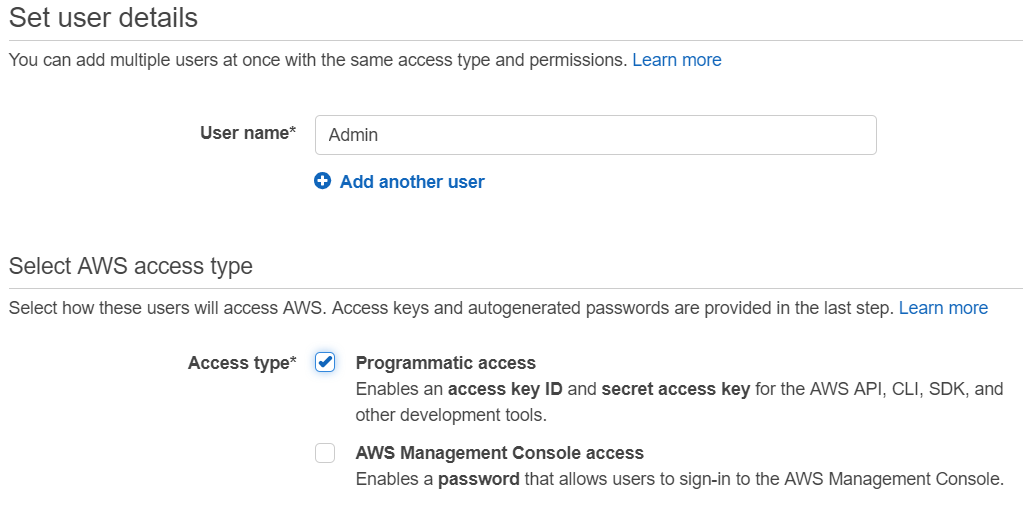
*Figure : IAM Navigation Menu.*

Next, in the popup window choose *“Add User”*.



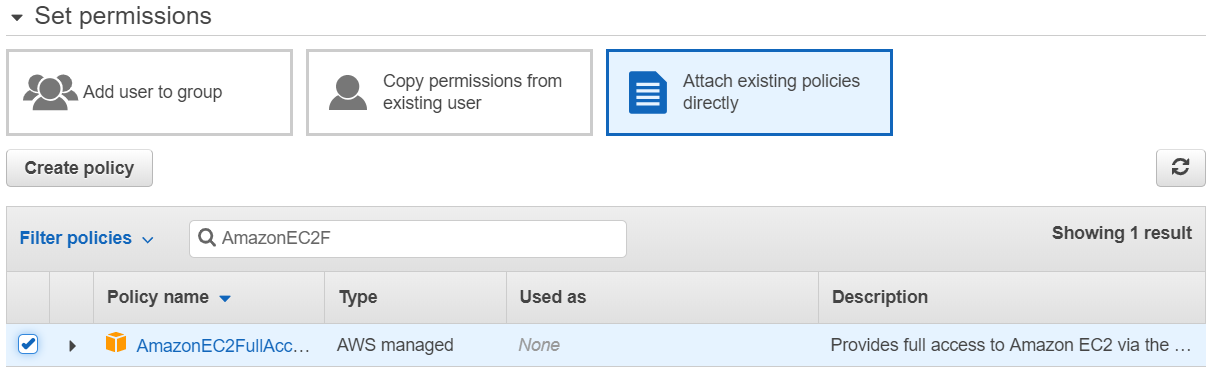
*Figure : IAM Add User.*

This will present a new window that requests a user name and the type of access this user should have. The type of access this user is allowed is important for when a user is working through the IOT Honeynet Framework. Programmatic Access is essential in order to use the AWS CLI, a core feature of the IOT Honeynet Framework that will be explored in more detail later. For the purposes of this project it is necessary for the user with administrator privileges to have programmatic access, which is the option selected as can be seen in the figure below.



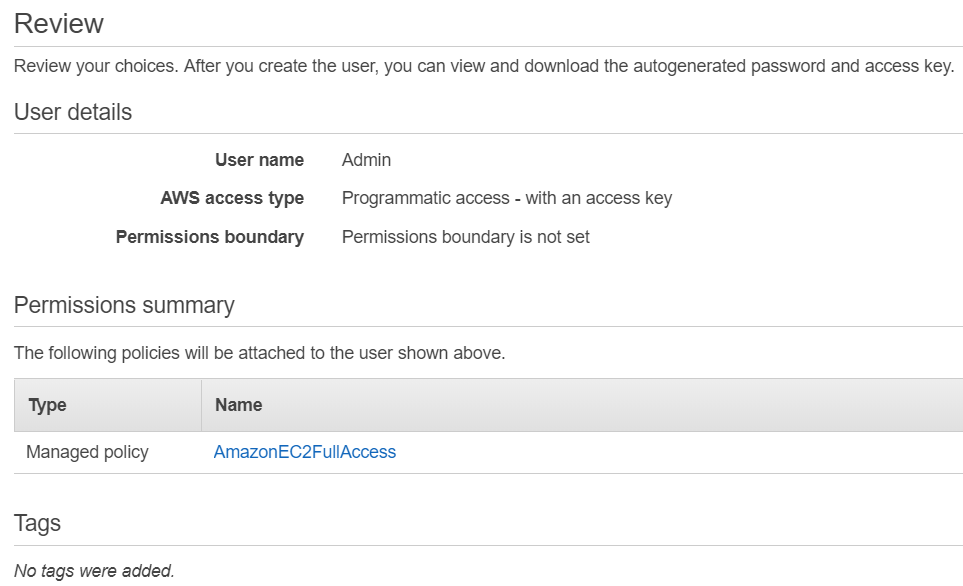
*Figure : IAM Username and Access Type.*

The next step is to determine what permissions the user should have. It is possible to create different groups to divide users based on what access privileges they are allowed to have. For the purposes of this project it was decided that this would be left to the user to decide. It was decided for the purposes of demonstrating this process, an existing policy would be attached to the user that is being created. The policy titled *“AmazonEC2FullAccess”* was selectedand attached to the user which can be seen in the figure below.



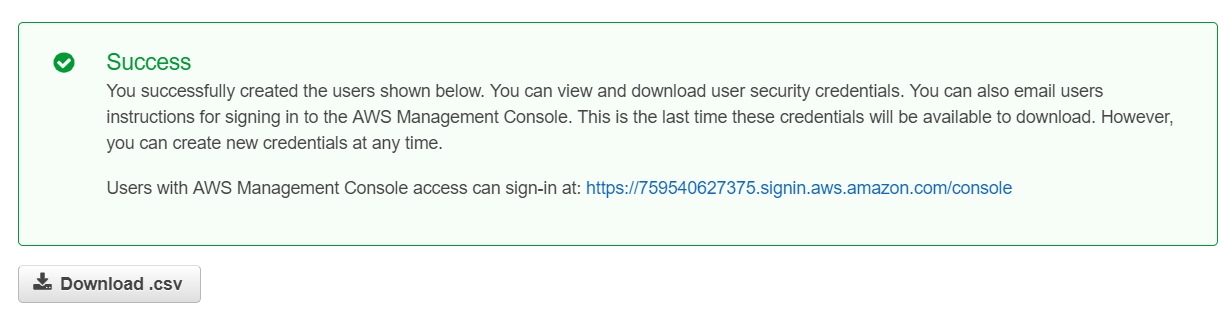
*Figure : IAM user permissions.*

Finally, the optional choice to add tags to the user is presented which can be used to assign metadata in order to help track, organize and control access for the user [2]. After this, the user details and permission levels can be reviewed before finally creating the user.



*Figure : Review IAM User Details.*

When the user is created, a window is presented which shows the user’s access key id and access secret key. These are only available once, so it’s important to download and save them safely in a secure location.



*Figure : IAM User Created Successfully.*

Now that an IAM user is created, it is possible to programmatically access Amazon EC2 service, which is essential in order to allow the IOT Honeynet Framework to remotely access and manage EC2 services. This will be explored more later in the Implementation Chapter.

#### **3.1.1.2 Virtual Private Cloud (VPC’s)**

Amazon Virtual Private Cloud (VPC) is defined as “creating a virtual network in your own logically isolated area within the AWS cloud, known as a virtual private cloud (VPC)” [9]. Amazon EC2 instances can be launched in a VPC which acts similarly to a traditional network but with the added bonus of being able to use scalable infrastructure from AWS [9]. When an AWS account is created, a default VPC is created in the region specified when setting up the AWS account, enabling users to instantly launch instances.

It’s important to understand the concepts of VPC as it is the networking layer for Amazon EC2 instances. Amazon kindly provides every user who sets up an account with a default VPC which is configured and ready for use. The default VPC conveniently includes an internet gateway, and each default subnet is a public subnet [10]. Each EC2 instance that is launched on a default subnet has a private IPv4 address and a public IPv4 address. These instances can communicate with the internet through the internet gateway. An internet gateway enables your instances to connect to the internet through the Amazon EC2 network edge [10].

It is possible to create nondefault VPC’s but it was decided to work of the default VPC since it satisfies the requirements needed by the IOT Honeynet Framework, which is to allow honeypots access to the internet and to collect logging information, something that will be discussed in more detail later.

#### **3.1.1.3 AMI’s and Instances**

According to Amazon, an Amazon Machine Image (AMI) “is a template that contains a software configuration (for example, an operating system, an application server, and applications)” [4]. It is from this image that an instance is launched, which is a copy of a virtual server running in the cloud. There are multiple AMI’s that can be selected, with some being a part of the AWS 12-month free tier, while others being more expensive.

An important consideration to make when launching an instance is the type of instance that is being launched. The instance type that is specified determines the hardware of the host computer used for your instance, offering different compute, memory, and storage capabilities [6]. These instance types are often grouped based on their capabilities and can dramatically impact the performance of the application or software being deployed on the instance.

For the base operating system of each honeypot, it was decided to only use the free tier AMI’s and allow the user to select which operating system they would like each honeypot to run. This approach allows the user more options while also keeping in line with the AWS 12-month free tier contract. The different instance types that were allowed for a user to select from were kept to a small t2.micro general-purpose instance type. This was chosen based on the minimum resources that were required for each honeypot to operate in the Honeynet framework, meaning more honeypots could be deployed without unnecessarily using up resources.

#### **3.1.1.4 Regions and Availability Zones**

According to Amazon, Amazon EC2 can be hosted over multiple locations world-wide, with these locations being composed of Regions and Availability Zones [5]. Each Amazon region is designed to be completely isolated from other Amazon EC2 Regions. Each region is then comprised of isolated locations known as Availability Zones [5]. This is meant to achieve fault tolerance and stability, two very important requirements for the hosting service of a Honeynet as any downtime could result in valuable data being lost for researchers.

The choice of what region to deploy the Amazon EC2 instances could greatly impact the type of results and data gained by a researcher. It is an important consideration to make when determining what region to deploy Amazon EC2 services in. The goal of the IOT Honeynet Framework is not to make these decisions for the researcher but to rather support the ease of deploying and managing instances for the researcher. It is with this in mind that it was decided to allow the user to make this decision when initially setting up an AWS account.

#### **3.1.1.5 Amazon EC2 Key Pairs**

A key pair refers to the public and private key used to encrypt and decrypt data, which is used by Amazon EC2 to encrypt and decrypt login information [7]. This enables secure remote access into an EC2 instance. Amazon stores the public key while the private key is kept by the user which is why it is important to keep the private key secure as anyone with access to the key can decrypt the login information of any instance associated with that particularly private key. The keys that Amazon EC2 uses are 2048-bit SSH-2 RSA keys and it is possible to have up to five thousand key pairs per Region [7].

It is important to create a key pair to securely access a honeypot instance from a remote location once it has been created. Once created it is necessary to change the mode of the key pair file to read-only otherwise it will be denied access. It is possible to create key pairs in the IOT Honeynet Framework which are stored in a file with a .pem extension and then later used when creating the instance. This will be demonstrated programmatically in the Implementation Chapter, but it is necessary to understand the importance of the EC2 key pair for logging in securely to newly created instances and is instrumental for setting up the python server used for the honeypot.

#### **3.1.1.6 Security Groups**

Amazon describes how a security group “acts as a virtual firewall that controls the traffic for one or more instances” [8]. When an instance is launched, one or more security groups can be assigned to it. A security group contains rules which dictates what traffic is allowed to and from an instance. Security groups are associated with network interfaces and when changing an instance's security groups, it changes the security groups associated with the primary network interface (eth0) [8].

The security group rules dictate what inbound traffic can reach an instance and what outbound traffic can leave an instance. There are a number of key characteristics that a security group has which are important in acknowledging when deploying an instance:

1. Security Groups allow all outbound traffic [8].
2. It is not possible to create rules that deny access [8].
3. Security groups are stateful meaning it tracks the operating state and characteristics of network connections traversing it. The firewall is configured to distinguish legitimate packets for different types of connections [8].
4. Rules can be added and removed at any time [8].

A rule in a security group is created with several parameters. When the IOT Honeynet Framework creates the security group, the protocol type, source and destination port, and the source address parameters are used. This will be demonstrated further in the Implementation Chapter. It is essential to know how to configure a security group to open ports for the honeypot based on what scan results are provided by the user.

### **3.1.2 VPC Flow Logs and CloudWatch**

VPC Flow Logs and CloudWatch are separate topics from Amazon EC2 and deserve their own section so as to better elaborate on their importance to the IOT Honeynet Framework. Logging is an important and vital function of any Honeynet Framework service. For this thesis, it was evident that if Amazon EC2 was being used by the IOT Honeynet Framework then VPC Flow Logs and Amazon CloudWatch were going to be used for collecting log information from each honeypot. It is therefore important to understand the process to setup each feature and how both relate to each other. It is also essential to be able to understand how to read the information that is being stored, hence why it is necessary to provide a separate section for both VPC Flow Logs and CloudWatch.

# **Chapter 4: Implementation**

## **AWS**

# **Bibliography**

1. Amazon.com. (2019). Setting Up with Amazon EC2 - Amazon Elastic Compute Cloud. [online] Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html [Accessed 24 Apr. 2019].
2. Amazon Web Services, Inc. (2019). What is AWS? - Amazon Web Services. [online] Available at: https://aws.amazon.com/what-is-aws/ [Accessed 22 Apr. 2019].
3. Amazon.com. (2019). What Is Amazon EC2? - Amazon Elastic Compute Cloud. [online] Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html [Accessed 22 Apr. 2019].
4. Amazon.com. (2019). Instances and AMIs - Amazon Elastic Compute Cloud. [online] Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instances-and-amis.html [Accessed 24 Apr. 2019].
5. Amazon.com. (2019). Regions and Availability Zones - Amazon Elastic Compute Cloud. [online] Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html [Accessed 24 Apr. 2019].
6. Amazon.com. (2017). Instance Types - Amazon Elastic Compute Cloud. [online] Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html [Accessed 24 Apr. 2019].
7. Amazon.com. (2019). Amazon EC2 Key Pairs - Amazon Elastic Compute Cloud. [online] Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html [Accessed 24 Apr. 2019].
8. Amazon.com. (2019). Amazon EC2 Security Groups for Linux Instances - Amazon Elastic Compute Cloud. [online] Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html [Accessed 24 Apr. 2019].
9. Amazon.com. (2013). Virtual Private Clouds - Amazon Elastic Compute Cloud. [online] Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-vpc.html [Accessed 24 Apr. 2019].
10. Amazon.com. (2013). What Is Amazon VPC? - Amazon Virtual Private Cloud. [online] Available at: https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html#what-is-connectivity [Accessed 24 Apr. 2019].