

## **Setting up a trusted Command and Control Server (C2 / C&C)**

**Author:** Emilio Revelo  
[jose@emiliorevelo.com](mailto:jose@emiliorevelo.com)

# Introduction

This paper covers one way to create a simple command and control server with a trusted certificate emitted by a free CA, using Empire with that certificate to encrypt all the data trasmitted between agents and listener.


hh

This can be very useful for all of those who wants to perform red team activities with a limited budget.

## Initialize an Instance

There are many options (even free) that we could use to create a virtual server with a static public IP, in this case I used EC2 from AWS (I decided to use my 1 year of free tier).

---

 **Ubuntu Server 18.04 LTS (HVM), SSD Volume Type** - ami-0f65671a86f061fcd (64-bit x86) / ami-0f2057f28f0a44d06 (64-bit Arm)

Free tier eligible

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).  
Root device type: ebs      Virtualization type: hvm

In this example I used Ubuntu Server with apache2, then we just install it and start it:

```
# apt-get install apache2  
  
# service apache2 start
```

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Fri 2019-01-18 18:30:18 UTC; 1h 15min ago
```

Then we just go to Instances/Description and check the public IP address.

## Getting a Domain Name and link with your public IP

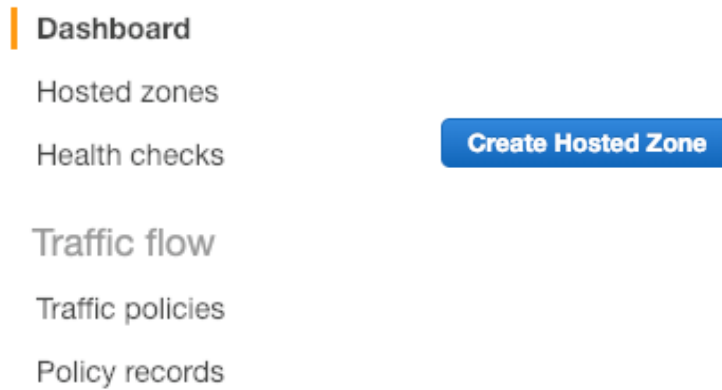
We going to use Let's encrypt that is a free, automated, and open Certificate Authority, but only can be used with domain names not with raw IP address.

There are many providers that sell cheap domain names, for example:

- Godday
- 1&1 (Actually IONOS)
- Namecheap
- Register
- Moniker
- etc...

Assuming that you already have a domain name with the provider that you prefer, then we go to our AWS console and do the follow to link our domain name with our EC2 IP:

1- Go to **Route 53** option, select **Hosted Zones** and **Create Hosted Zone**:



Once there, in the right column you need to specify your domain name and click on create button.

A screenshot of the 'Create Hosted Zone' form in the AWS Route 53 console. The form has a title bar 'Create Hosted Zone'. Below it, there is a descriptive text: 'A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.' The form contains three fields: 'Domain Name:' with a text input containing 'example.org' (highlighted with a green border), 'Comment:' with an empty text input, and 'Type:' with a dropdown menu showing 'Public Hosted Zone'. Below the 'Type:' dropdown, there is a note: 'A public hosted zone determines how traffic is routed on the Internet.'

After that, we then get the name servers that we going to use to redirect our site.

<input type="text" value="Record Set Name"/>	<input type="button" value="X"/>	<input type="button" value="Any Type ↕"/>	<input type="checkbox"/> Aliases Only	<input type="checkbox"/> We
<input type="checkbox"/>	Name	Type	Value	
<input type="checkbox"/>	example_ex.org.	NS	ns-857.awsdns-43.net. ns-1303.awsdns-34.org. ns-110.awsdns-13.com. ns-1715.awsdns-22.co.uk.	
<input type="checkbox"/>	example_ex.org.	SOA	ns-857.awsdns-43.net. awsdns-hostmaster.amazon.	

Finally we must select the option Create a Record Set to specify the IP address that will be linked with our domain name.

- 2- When your NS created, you must configure them in your domain name provider's console. After a couple of minutes or hours this change will be applied.

## Emiting a trusted certificate by a CA

We could use Let's Encrypt to create our trusted certificate, first we need to install certbot<sup>1</sup> by selecting Apache as a web server (or whichever you want) and the system OS, once that we have selected that options, we could see the instructions to install certbot (This is quite easy).

It's important to mention that Let's Encrypt<sup>2</sup> only issues certificates related to domain names, not for raw IP address, that's the reason why we must already have a domain name.

---

<sup>1</sup> <https://certbot.eff.org/>

<sup>2</sup> <https://letsencrypt.org/>



## Setting up Empire with the previous certificate

Once that your trusted certificate is done, you must copy the next two files (To another path) located in */etc/letsencrypt* and change their names as follows:

```
privkey1.pem -> empire-priv.key  
fullchain1.pem -> empire-chain.pem
```

We must setting up an HTTPS with the next parameters:

```
Host: https://[domain name or IP]:Port  
Port: 8080, 443, 80, etc...  
CertPath: Path where you copied certificate's files above.
```

And then execute the listener and start to connect clients.

## References:

- [1] <https://certbot.eff.org/>
- [2] <https://www.powershellempire.com/>
- [3] <https://letsencrypt.org/>