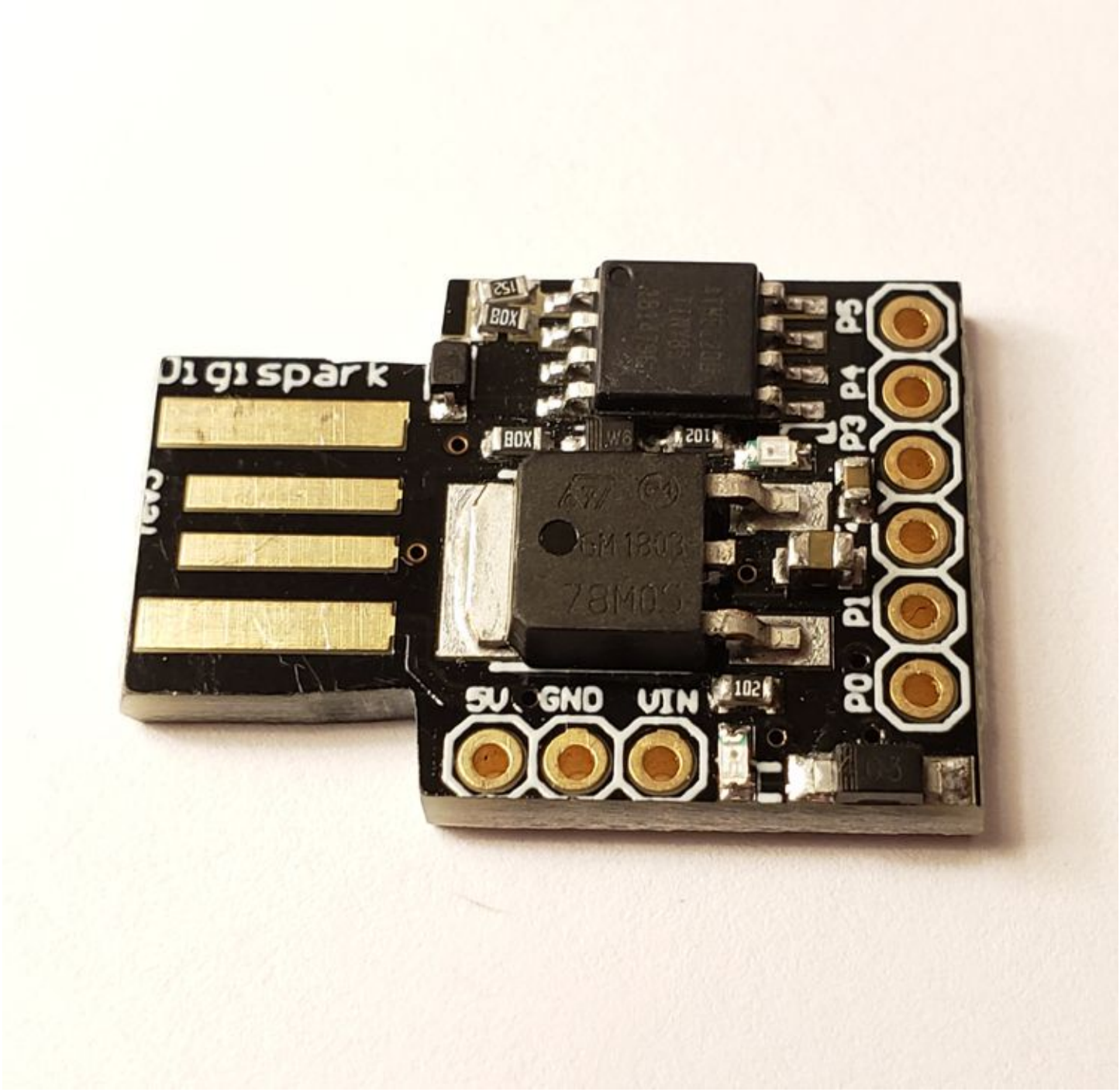
Setup Instructions for the ATtiny85 (a.k.a. The Poor Man’s Rubber Ducky)

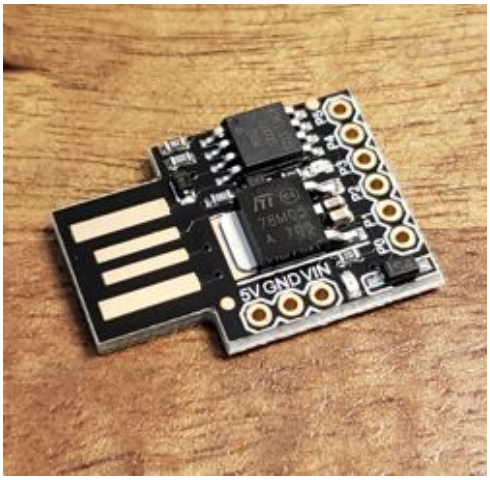
Developed for the “Microcontrollers and Single Board Computers for Hacking, Fun and Profit” Workshop at Day of Shecurity



In the Beginning...

Thank you for taking the time to attend the “Microcontrollers and Single Board Computers for Hacking, Fun and Profit” Workshop. I truly hope that you will enjoy this workshop as much as I enjoyed creating it. The genesis that sparked the creation of the Poor Man’s Rubber Ducky started with a common security best practice that we all see way too often; unlocked laptops in public areas. As security technologists we know all too well the risks of such a best practice being ignored. Yet our fellow employees, friends, family and colleagues may not. To help demonstrate the risks of this, I created the PwnMeUSB. while the PwnMeUSB worked well, it was limited in what it could do and only usable on the Apple OSX operating system. To truly demonstrate the risks of leaving one’s laptop unlocked in public, I needed something cheaper, more flexibility and cross-platform support. Enter the Poor Man’s Rubber Ducky, Arduino. Arduino-based laptop exploitation was nothing new; other engineers have created code repositories with their particular approach. However, having used this as a tool for onboarding and security best practice education was something that had not quite been socialized.

“Microcontrollers and Single Board Computers for Hacking, Fun and Profit” was first presented at DEF CON 26 to a very enthusiastic audience. It has subsequently been presented regionally for the Bay Area OWASP and Cybersecurity Career Accelerator Expo in Sacramento, CA.



In this workshop you will learn to program the ATtiny85 USB development board to inject automated keyboard strokes. This will invoke a reverse shell on the “victim’s” computer. You will then use a cloud-based command and control server to capture the shell request. Lastly you will remotely connect to the C2 server and begin issuing commands to the “victim’s” computer. As you will see, it is very easy to mimic basic behaviors of the operating system to trick users into providing passwords, read supposedly protected files or event troll relentlessly.

This document details the setup instructions you need to have completed before the workshop. This document is thorough and detailed. If you find that something doesn’t make sense or isn’t quite clear, please feel free to reach out to me for clarification. Chances are likely that if you’re confused, someone else is as well.

Once again, thank you for your time, your attention and your interest in the “Microcontrollers and Single Board Computers for Hacking, Fun and Profit” Workshop.

Enjoy!

Matt Torbin

Table of Contents

In the Beginning... 2

Operating System Support 4

Conventions Used in This Document 4

Technology Required 5

Hardware 5

Software 5

Download and Install the Official Arduino IDE 6

Downloading the IDE 6

Installing the IDE 6

Apple OSX 6

Linux Distributions 6

Microsoft Windows 7

Additional Driver Update for Windows 8

Configuring the Environment for the IDE 8

Apple OSX 8

Linux Distributions 8

Microsoft Windows 8

Setting up the IDE for the ATtiny85 9

Download and Install Oracle VirtualBox 9

Downloading and Installing VirtualBox 9

Apple OSX / Microsoft Windows 9

Linux 10

Installing a Linux Image in VirtualBox 10

Installing Additional “Guest Additions” Features 11

Setting up the Arduino IDE Environment 11

Setting up Amazon AWS 11

Creating an EC2 Instance 12

Properly Configuring Security Groups 12

Remotely Logging Into Your Virtual Machine 13

Running Netcat Remotely on your AWS EC2 Instance 14

Preparing the ATtiny85 15

Conclusion 16

Operating System Support

Below is the operating system support

OS	Installation Tested	ATtiny85 Micronucleus	Reverse Shell Support Tested
	✔	✔	✔
	✔		✔
	✔	✔	

Conventions Used in This Document

The following conventions are used throughout this document:

Syntax Example	Description
`$ terminal command`	Exact commands to be typed into the terminal are displayed in this manner.
`FieldValue`	Exact values to be entered into the text fields or navigational elements are displayed in this manner.
`[VALUE]`	A variable placeholder to be replaced with a user-specific value is displayed in this manner.

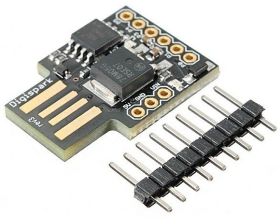
The following iconography are used throughout this document:

Icon Example	Description
	A topic, set of instructions or a point that is specific to OSX-based systems.
	A topic, set of instructions or a point that is specific to Linux-based systems.
	A topic, set of instructions or a point that is specific to Windows-based systems.

Technology Required

If you're reading this document outside of a workshop environment, then chances are you will need to acquire the proper hardware and software. The following list, while not exhaustive, is the minimum needed to successfully set up an environment to program and test an ATtiny85 or "Poor Man's Rubber Ducky".

Hardware

Item	Description\
	**10 in. Bastard-Cut Mill File:** Basically any file will work, though this size tends to work well. As you will see from the presentation, you only need to file down the sides a *very* little bit so whatever file you have around the house should work fine, so long as it's not too big or coarse.
	**ATtiny85 Development Board:** The official "brand name" version is the Digispark Kickstarter ATTINY85 Arduino General Micro USB Development Board. However, there are many copies made which are not brand name and will work just fine. For the purposes of this workshop, you will want to make sure that you are acquiring one that is the "USB development board". This means that it has a male USB 2.0 end. This board is easily acquired at both Amazon and eBay.
	**(OPTIONAL) USB-A to USB-C Adapter:** To exploit systems that have transitioned to the USB-C socket, you will need a proper adapter. These adapters are easily obtained on Amazon, eBay and Fry's. While name brand is typically the way to go, I've found with this particular exploit that a generic version tends to be less scrutinizing than say, an Apple-branded adapter (and way cheaper as well).

Software

- **Amazon AWS Account:** We will be creating our command and control (henceforth referred to as C2) server using AWS. While any cloud or publicly-exposed endpoint will work, AWS is friendly to new users. Accounts are free to set up and the instance we will be building will cost only a few dollars to maintain.
- **Official Arduino IDE:** This is the official development environment for the Arduino platform. While there are other ways to program an Arduino, this software is free, cross-platform and works rather well. We will be installing it next.
- **Oracle VirtualBox:** To perform all of the actions necessary on one system, we will use Oracle's VirtualBox to house the attacker's machine.

Download and Install the Official Arduino IDE

The first step is to download, install and set up the official Arduino IDE. The Arduino IDE software can be downloaded directly from the Official Arduino web site. As mentioned earlier, there are a number of ways to program for the Arduino platform. If you feel more comfortable using a different software application, feel free to skip this section.

Downloading the IDE

1. Open your browser of choice.
2. Navigate to https://www.arduino.cc.
3. From the navigational menu on the top of the page, hover over `Software` and then click `Downloads`.
4. Scroll down to the section entitled “Download the Arduino IDE”. On the right-hand side of the first panel, you will see multiple platform options available for download. Select the one that is right for your operating system.

Figure 1: The download options panel as it appears at the time of this writing.

Installing the IDE

The install process for each of the three major operating systems should be fairly consistent. However, to abate any confusion we will cover each of them individually below.

Apple OSX

1. Find the Arduino IDE installer zip file that was downloaded. Double-click it to decompress it.
2. Once the application has been decompressed, drag it to your Applications folder.

Linux Distributions

1. Open the Terminal application within your operating system and navigate to your Downloads folder:

```
$ cd ~/Downloads
```

2. Find the XZ file that was downloaded and decompress it. The example below is for the latest version of the Arduino IDE, which at the time of this writing was version 1.8.9:

```
$ tar -xvf arduino-1.8.9-linux64.tar.xz
```

The file will begin to decompress into a directory. In the case of the example above, the directory is named `arduino-1.8.9`.

3. Once the directory has been fully decompressed, change into this directory and run the `install.sh` shell script to begin the installation process:

```
$ cd arduino-1.8.9/
$ ./install.sh
```

4. Once the installation process is complete you can close the terminal.

Microsoft Windows

1. Find the Arduino IDE installer executable that was downloaded and double-click it to launch it.
2. When prompted with “Do you want to allow this app to make changes to your device?” click the `Yes` button.
3. Review the license agreement and once done, click the `I Agree` button.
4. At the Installation Options pane, ensure that all four of the checkboxes have been checked and then click the `Next` button.
5. At the Installation Folder pane, select where you would like to install the Arduino IDE. Typically the default suggestion is usually best. When ready, click the `Next` button. At this point the Arduino IDE will begin to install. This process can take a few minutes.
6. When prompted with “Would you like to install this software?” with regards to the Adafruit Industry LLC Ports, click the `Install` button.
7. When prompted with “Would you like to install this software?” with regards to the Arduino USB Driver, click the `Install` button. There will be two of these requests. Repeat the same actions for the second request.
8. At this point the installation is complete. You can now click the `Close` button and close the Arduino IDE installer executable.

Additional Driver Update for Windows

To quote the Digistump wiki:

> *If using Arduino 1.6.6 or higher and windows - you will need to download and install the drivers manually. Download, unzip and run "Install Drivers" (on 32bit systems) or "DPInst64" (on 64bit systems). If you get stuck, try following the steps shown in this YouTube video. The driver files are located here: https://github.com/digistump/DigistumpArduino/releases/download/1.6.7/Digistump.Drivers.zip*

1. **Download the drivers above.**
2. **Install the proper bit version for your OS** *(in the extracted folder, either click on* `DPinst` *or* `DPinst64`*).*
3. **Ensure via the Device Manager that your ATtiny85 shows up.**
4. **If you have questions, watch the YouTube video.** Trust me, it helps.

Configuring the Environment for the IDE

The only operating system that requires additional configuration is Linux. Those instructions are below:

Apple OSX

Congratulations! There is no additional configuration necessary!

Linux Distributions

It may happen that the user you're logged in with has not yet been added to the dialoout group and as such, you might get "Error opening serial port …" errors when you are working with sketches. To correct this, add your current user to the dialout group:

```
$ sudo usermod -a -G dialout [YOUR_USER]
```

You will have to log out and then log back in again. For further detailed help, please refer to the document entitled "Install the Arduino Software (IDE) on Linux" on the Official Arduino web site.

Please note that there are some very specific troubleshooting suggestions for Linux. Instead of duplicating efforts and including them here, I will link you to the Digistump Wiki Linux Troubleshooting guide.

Microsoft Windows

Congratulations! There is no additional configuration necessary!

Setting up the IDE for the ATtiny85

In order to properly use the Arduino IDE with the ATtiny85, we need to add some additional boards to it. This process is outlined in the steps below. First, open the `Preferences…` dialog box. How you get to this particular pane might be slightly different depending upon the operating system that you're using.

Apple OSX

Select Arduino > Preferences

Linux Distributions

Select File > Preferences

Microsoft Windows

Select File > Preferences

Once you have the Preferences dialog box open, follow these steps:

1. In the textfield next to `Additional Board Manager URLs`, enter `http://digistump.com/package_digistump_index.json`. When done, click the OK button.
2. Under the `Tools` menu, select `Board:...` and then `Boards Manager…`. The `Board:...` menu item may have a different default board depending upon your setup but that doesn't matter. Your goal is to get to the Boards Manager.
3. With the Boards Manager dialog box visible, type `digistump` into the search filter textfield at the top.
4. When you see the option for `Digistump AVR Boards`, hover over that box and click the `Install` button. Once the installation is done, click the `close` button.

Download and Install Oracle VirtualBox

The next piece of software that will be needed is a virtual machine. While there are many available, both paid and free, for the purposes of this workshop we will be focusing on Oracle's VirtualBox. If you feel comfortable using another virtual machine, feel free to skip this section.

Downloading and Installing VirtualBox

Apple OSX / Microsoft Windows

1. Open your browser of choice.
2. Navigate to https://www.virtualbox.org.
3. Click the giant button in the center of the page which reads `Download VirtualBox`. At the time of this writing, the most current version was 6.0.
4. Scroll down to the section entitled "VirtualBox 6.0.10 platform packages". Click the link for the package that is right for your operating system.
5. When the installer downloads successfully, double-click the installer file and follow the prompts to install VirtualBox.

Linux

1. Install virtualbox from the command line using apt:

```
$ sudo apt install virtualbox
```

Installing a Linux Image in VirtualBox

In the field, this type of exploit would be done with a completely separate machine. However, for the purposes of this workshop, we will be using a Linux virtual machine to act as the attacker. There are lots of operating systems that can be chosen and if you have one in particular that you feel most comfortable with, you are welcome to use that in place of the following instructions. However, if do not have a preference, you are also welcome to follow along with the instructions below. Please note that none of the tooling used below is unique to the operating system that we have chosen. For the purposes of this example we will be using Ubuntu Mate as it's quite small and fairly easy to work with:

1. Go to https://ubuntu-mate.org/download/ and choose `32-bit` (for the purposes of our example, speed is not a concern).
2. When presented with a release, choose the most recent version. At the time of this writing, the most recent version was 18.04.3.
3. When presented with a download option, you may choose whatever works best for you, though for this workshop we will be choosing the ISO file type (which will download over HTTP/S).
4. Once the file has been downloaded, launch VirtualBox.
5. When VirtualBox opens, select the New icon.
6. In the Name and operating system pane, enter the following information:
    a. **Name:** Pick a name for your virtual machine.
    b. **Machine Folder:** It's probably best to leave this in its default state unless you are comfortable with VirtualBox.
    c. **Type:** Select `Linux` from the dropdown menu.
    d. **Version:** Select `Debian (32-bit)` from the dropdown menu.
7. In the Memory pane enter `3072` for the memory (roughly 3GB) and click `Continue`.
8. In the Hard disk pane, leave the radio button for Create a virtual hard disk now selected and click `Continue`.
9. In the Hard disk file type pane, leave the default selection for VDI (VirtualBox Disk Image) alone and click `Continue`.
10. In the Storage on physical hard disk pane, leave the default selection for Dynamically allocated alone and click `Continue`.
11. In the File location and size pane, leave the file location alone but change the size of the virtual hard disk to `16.00 GB`. When you are done, click `Create`.
12. Specific to Ubuntu Mate, there is one additional setting that we must make:
    a. Click the `Settings` icon.
    b. Select the `System` icon and then the `Processor` tab in the pane.
    c. Check the box for the option `Enable PAE/NX`. When you are done, click `OK` to save your changes.

At this point your empty virtual machine will be created. However, you still will need to install the operating system that you chose. To do that, we will power on the machine by clicking the `Start` icon. The moment your virtual machine boots up for the first time, it will ask you to find the installation image that you downloaded. Click on the file icon and find your installation disk image. Follow the prompts to install the operating system into your virtual machine. It is beyond the scope of this document to walk through the specific installation steps for a given operating system. However, if you require help with this, please reach out to info@dayofshecurity.com.

Installing Additional “Guest Additions” Features

You will notice that under the Devices menu in VirtualBox you have an option for `Install Guest Additions CD Image…` which you should go ahead and install. These features are required to interact with your virtual machine in a way that will be helpful, specifically in moving your PEM file over to your virtual machine.

Setting up the Arduino IDE Environment

The setup for all three operating systems is identical and is detailed below:

1. Open the Arduino IDE and on the menu bar, select `File > Preferences`.
2. Find the text field with the label “Additional Board Manager URLs” and enter the following:

   ```
   http://digistump.com/package_digistump_index.json
   ```

3. Click the OK button when done.
4. Under the `Tools` menu on the menu bar, select the `Board > Board Manager`. The board manager dialog box will appear.
5. Select the “Digistump AVR Boards” package and click the “Install” button.
6. Once the installation has finished, under the `Tools` menu on the menu bar, select the `Board` submenu and then `Digispark (Default 16.5mhz)` from the select list. This is the board that we will be programming with.

Setting up Amazon AWS

For the purposes of this workshop we will be using Amazon AWS for our remote C2 server. If you feel more comfortable using a different publicly-available endpoint, please feel free to skip this section. If you do not currently have an AWS account, you will need to sign up for one. Please note that an Amazon store account is *not* the same as an Amazon AWS account. While you may use the same user, you still have to go through the process of setting that account up. If you need to create an Amazon AWS account, please do so before continuing.

Creating an EC2 Instance

The very first step is to create an EC2 instance in AWS. This server will act as our C2 server. The following steps will guide you through the creation of this server. When you first log in, you will see your AWS Management Console. Depending upon how you last used AWS, this display could be slightly different for each person. For that reason, we are going to start by accessing the EC2 services via the navigational menu at the top of the page.

1. Select Services > and then EC2 under the Compute heading.
2. On the page that loads, click the Launch Instance button under the Create Instance heading.
3. On the page for Step 1, select Ubuntu Server 18.04 LTS as the machine image to work with by clicking the Select button. As of this writing, the version of Ubuntu is 18.04. Things may be slightly different when you read this.
4. On the page for Step 2, select the option for the t2.micro. In some cases, this option may already be selected. When ready, click on the Next: Configure Instance Details button.
5. On the page for Step 3, select the dropdown for Auto-assign Public IP and select Enable. All other configurations are fine with their default values. When ready, click on the Next: Add Storage button.
6. On the page for Step 4, there are no changes to be made here unless you wish to make additional customizations. When ready, click on the Next: Add Tags button.
7. On the page for Step 5, there are no changes to be made here unless you wish to make additional customizations. When ready, click on the Next: Configure Security Group button.
8. On the page for Step 6, add a custom name in the Security Group Name field. Add a custom description for this EC2 instance in the Description field. Lastly, in the table below, select My IP under the Source column and add an appropriate description such as “My Home” or “My Office”. When ready, click the Review and Launch button.
9. On the page for Step 7, review the configurations and ensure that they are what you want. When ready, click on the Launch button.
10. On the panel that appears, select Create a new key pair, add a name to the key pair in the Key pair name textfield.
11. Click the Download Key Pair button to download your EC2 instance’s key pair. When ready, click the Launch Instances button.
12. You will now see a page which is informing you of your instance’s launch status. To go directly to your instances, click on the View Instances button. Make sure to note your instance’s Public DNS (IPv4) address as this will be needed later to log in.

Properly Configuring Security Groups

The next step is to properly configure the security groups for your EC2 instance. For our example here, we won’t be configuring any outbounds outside of the default settings. Our primary goal is to properly set up the inbound rules. To get to the Security Groups, there are a number of ways to get there. However, we will take the approach of assuming that you’re currently looking at the AWS Management Console:

1. In the Find Services textfield, type “security groups”.
2. When EC2 appears in the suggestion dropdown, click on it.
3. On the EC2 Dashboard page, select Security Groups from the navigational menu on the left-hand side; it’s under the NETWORK & SECURITY section.

At a minimum the following rules should be set up for inbound; their description is after the table:

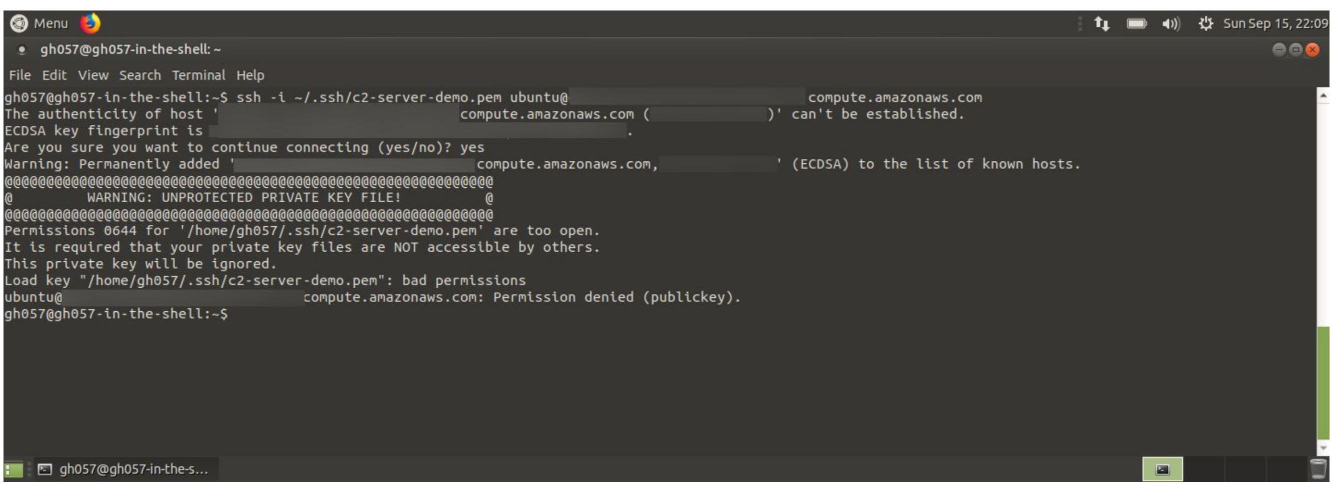
Type	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP	8080	0.0.0.0/0	IPV4 Inbound custom rule
Custom TCP Rule	TCP	8080	::/0	IPV6 Inbound custom rule
SSH	TCP	22	[YOUR_IP_ADDRESS]	[DESCRIPTIVE_LOCATION]

- **Custom TCP Rule #1:** This rule allows for all *IPV4* incoming traffic on port 8080. The port itself can be anything of your choosing but for this tutorial, we will be using 8080.
- **Custom TCP Rule #2:** This rule allows for all *IPV6* incoming traffic on port 8080. The port itself can be anything of your choosing but for this tutorial, we will be using 8080.
- **SSH Rule:** This rule allows for SSH connectivity between the machine located at `YOUR_IP_ADDRESS` and your EC2 instance. Without these types of rules, anyone could SSH into your EC2 instance which is not desirable.

Remotely Logging Into Your Virtual Machine

I've briefly outlined these steps below:

1. Create a directory called `.ssh` in your home directory if one does not already exist.
2. Move/Copy your PEM file that you downloaded from Amazon to this directory. There are many ways to accomplish this task. I personally found it most useful to turn on Shared Clipboard options and simply paste the contents of the PEM file into whatever command line text editor you choose to use.
3. Change the permissions of the PEM file to be more restrictive. Unfortunately, if you try to log in with the default file permissions that the operating system gives you, you will get an error like the following:



In order to correct this issue, you need to change the permissions of your PEM file to something more restrictive, say 400:

```
$ sudo chmod 400 ~/.ssh/YOUR_FILE_NAME.pem
```

Once this is done, you should be able to successfully log into your instance.

4. Log into your instance using the following command:

```
$ ssh -i ~/.ssh/YOUR_FILE_NAME.pem ubuntu@INSTANCE_PUBLIC_ADDRESS
```

If you remembered to note down your instance's `Public DNS (IPv4)` address, will replace `INSTANCE_PUBLIC_ADDRESS` above with that address. If not, you will have to log back into AWS to retrieve it.

Running Netcat Remotely on your AWS EC2 Instance

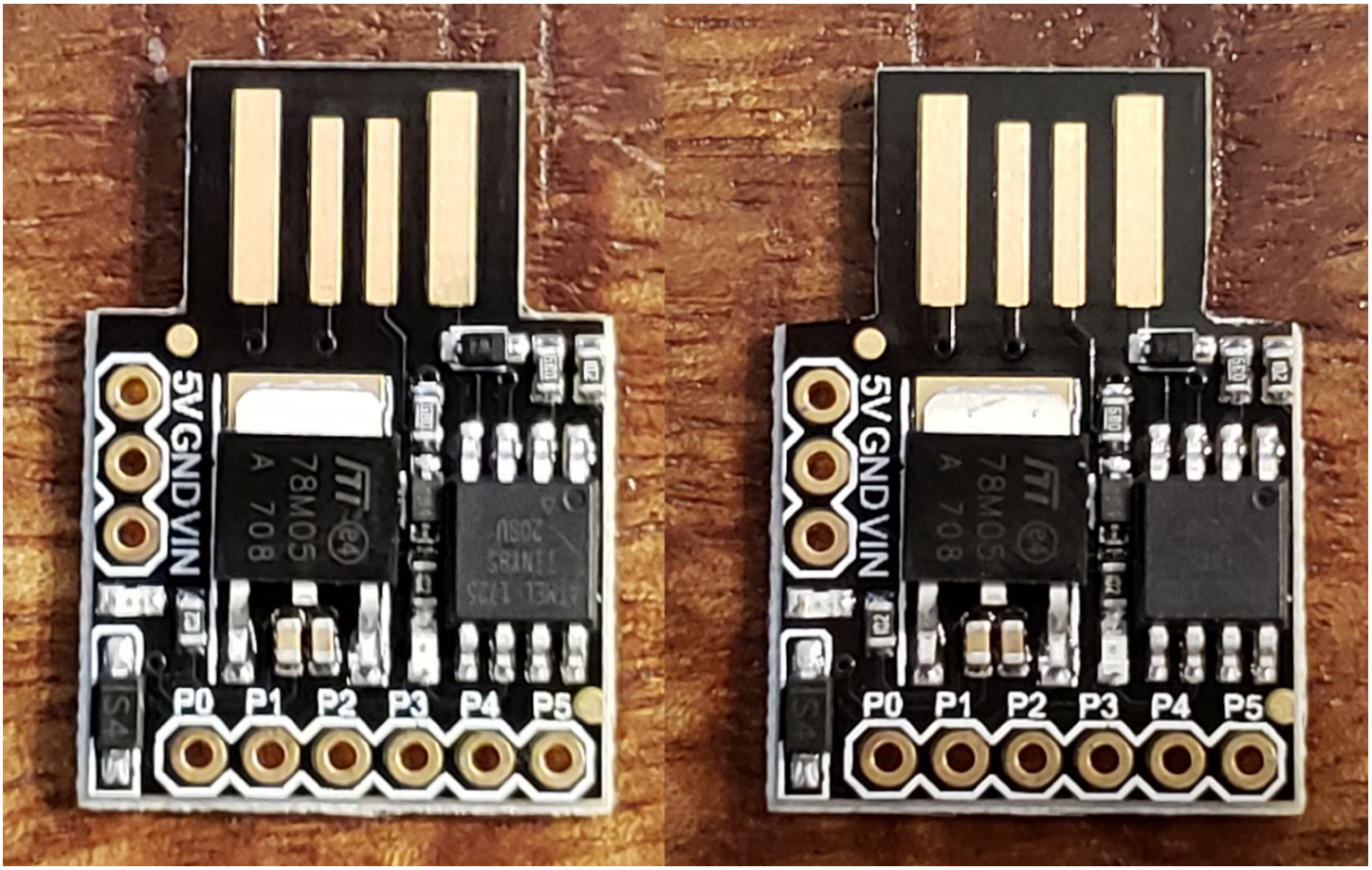
The next and final step for working with your AWS instance is to run a Netcat listener on port 8080, the port that we've chosen for this workshop. If Netcat isn't already installed in your instance, you should go ahead and install it now. For our purposes, we will use the apt package installer to do this for us:

```
$ sudo apt install nc
```

Once Netcat has been installed we are all good to go!

Preparing the ATtiny85

In many cases, the ATtiny85's that come from China have to be slightly filed down so that all of the leads will touch. Without this, it may appear to work, but in actuality only the power leads are touching which means the data leads are not actually connecting. There is no exact science about how much to file the edges down. See the photo below for a before and after shot of how much filing is required:



In general, I've found that I can file a board in under a minute and have it work just fine.

Conclusion

I want to thank you for supporting Day of Shecurity and attending my workshop. While I strive to have everything set up perfectly, sometimes things don’t always work as we plan. If there is any aspect of the documentation, workshop, hardware or presentation that you feel could be improved, please do not hesitate to reach out to me and share your thoughts. I want to ensure that this workshop continually gets better and that process is greatly helped by your input.