A Prototype of Penetration Testing Device: Project EL-Padrino

Asyraaf Ayob
Forensic and Cyber Security Research Centre
Asia Pacific University of Technology
and innovation (APU)
Kuala Lumpur, Malaysia
asyraafayoob23@gmail.com

Nor Azlina Abd Rahman
Forensic and Cyber Security Research Centre
Asia Pacific University of Technology
and innovation (APU)
Kuala Lumpur, Malaysia
nor_azlina@apu.edu.my

***Abstract*— The paper is the continuity research on the Project El Padrino framework that being proposed based on the research being done on several malware and USB attacks. The research and literature review being done for several devices such as Rubber Ducky, Malduino and Digispark USB Development Board in identifying on improvement needed for project El Padrino implementation. Taking consideration on several aspects of the improvement, this paper is focusing on the prototype design and development of penetration testing device that named it as project El Padrino. Discusses about the development of Project El Padrino such as the challenges faced, the solutions and the implementations for the device and the application. The paper shows all the initial and final design of the device and how it was improvised. The paper focuses on the main features in the device and the application and how it was built successfully without facing errors.**

Keywords—CLI-based application, Ducky Script, NTLM Hash Cracker, Project El Padrino, SSH server, system design.

I. INTRODUCTION

In order to build this device, proper knowledge of malware scripting is needed so that the scripts can run properly. For the malware scripting part, knowledge of Ducky Script is needed since all the exploits that will be developed in this project, works in Ducky Script. For the application part in Kali Linux, the expertise in Python programming language is needed to develop a proper CLI application. Even though this device has a certain feature that the penetration testing devices in the market do not have, but some users will still prefer to buy the original device due to the brand and popularity of the device.

Therefore, Project El Padrino is hard to compete with the devices that are already in the market for so many years. Besides that, a basic understanding of electronics is needed to build a penetration testing device. The researcher needs to wire the pins correctly so that the whole device works. Basic soldering skill is also needed to build a device like Project El Padrino. The researcher needs to solder the pins correctly without damaging any pins of the hardware.

A. Rationale

Project El Padrino will help cybersecurity students to gain a new experience while using this penetration testing device and learn about malware and Bad-USB attacks at the same time.

As for beginners in cybersecurity, this project will be a starter pack for them to start learning about penetration testing. This project will also give cybersecurity students the opportunity to build their own penetration testing device rather than buying a premade device. Project El Padrino also will include a user manual and a step-by-step learning process for cybersecurity students to learn how the malware scripts work.

B. Potential Benefits

Looking into the benefits of this project in detail, the benefits are divided into tangible and intangible benefits as follow:

1) *Tangible Benefits*
   a) Project El Padrino is cost-saving compared to other penetration testing device that is similar to it.
   b) Helps cybersecurity students to save time by compiling the list of malware scripts once, instead of compiling the scripts individually.

2) *Intangible Benefits*
   a) Provide good user satisfaction as it gives cybersecurity students the options of malware scripts that they can choose and run.
   b) This project is also user-friendly as it is easy to use and the user manual is provided as well.
   c) Cybersecurity students will have exposure to malware and Bad-USB attacks while using this device and learning from the user guide provided.
   d) The beginners of cybersecurity will be motivated to learn more about penetration testing after using this device.
   e) This device is portable and cybersecurity students can carry this device anywhere they want.
   f) Cybersecurity students will be motivated to build their own penetration testing device after using this device.

II. SYSTEM DESIGN

The Use Case diagram as shown in Fig 1. begins with an actor, User. User can move down and up the list of malware scripts in the device using the buttons provided. User can use ‘Enter’ button to exploits a malware script at a time. Users can also use the features in application to integrate with the device.

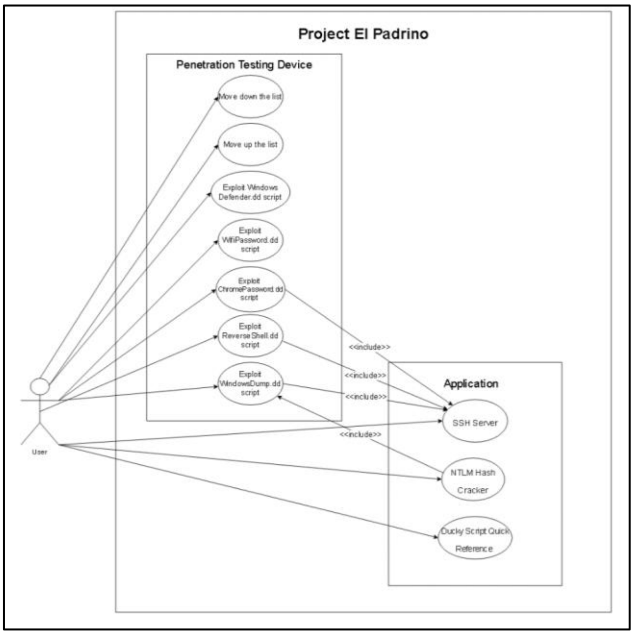
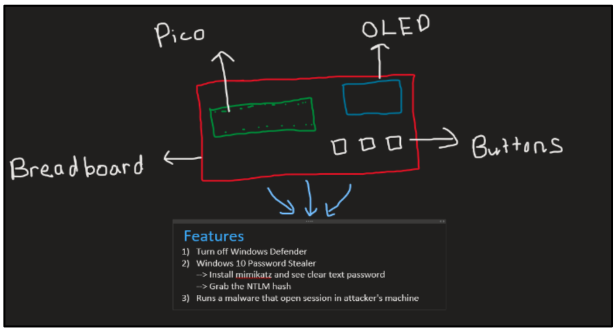

Project El Padrino
Penetration Testing Device
Move down the list
Move up the list
Exploit Windows Defender.dd script
Exploit WifiPassword.dd script
Exploit ChromePassword.dd script
Exploit ReverseShell.dd script
Exploit WindowsDump.dd script
<<include>>
<<include>>
<<include>>
<<include>>
Application
SSH Server
NTLM Hash Cracker
Ducky Script Quick Reference
User

Fig. 1. Project El Padrino use case diagram

Fig 2. Shows the first sketch done by the researcher. The first idea is to include all the hardware on a small breadboard. The researcher also had the ideas of the features that would be implemented in the development stage. However, the researcher faces issue with the initial plan after the buying the hardware. The issue was there is no space to put all the hardware on a single breadboard. Therefore, the second design of the device includes two small breadboards as seen in the Fig 3.



Features

1) Turn off Windows Defender
2) Windows 10 Password Stealer
   --> Install mimikatz and see clear text password
   --> Grab the NTLM hash
3) Runs a malware that open session in attacker's machine

Fig. 2. The first sketch of Project EL Padrino

Fig 3. Shows the second sketch, the researcher has connected two small breadboards to place all the hardware together. However, another concern arises, where the researcher needs to wire the OLED display and the three push buttons to the appropriate pins of Raspberry Pi Pico.

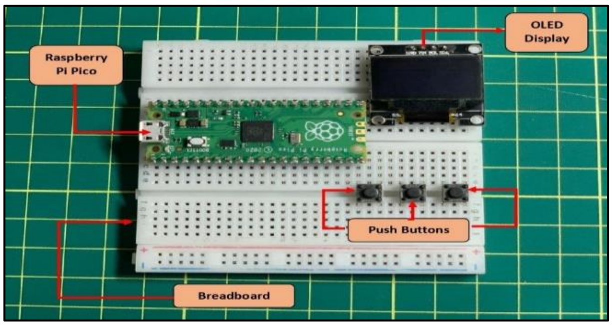


Fig. 3. The second design of the device

Wiring is one of the important steps for a well-designed device. If there is mistake in the wiring, the hardware might damage the Pico's pins with high voltage. Fig 4. shows the wiring is completed and all the hardware is connected to Pico. However, the jumper wires used is messy and can interrupt users when using the buttons. Due to this, the researcher has decided to build a final and new design for this device.

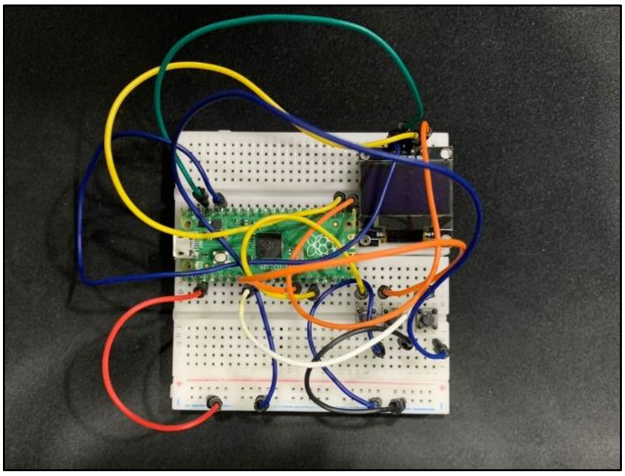


Fig. 4. The second device design with jumper wires

Fig 5. shows the final design of the device invented by the researcher. The researcher replaces the small breadboard to a mini breadboard for the OLED display and has adjusted the position of the hardware to allow users to use the buttons comfortably. The researcher also uses jumper wires that are not messy and attachable to the breadboard. The wiring of the hardware is shown in Table I.

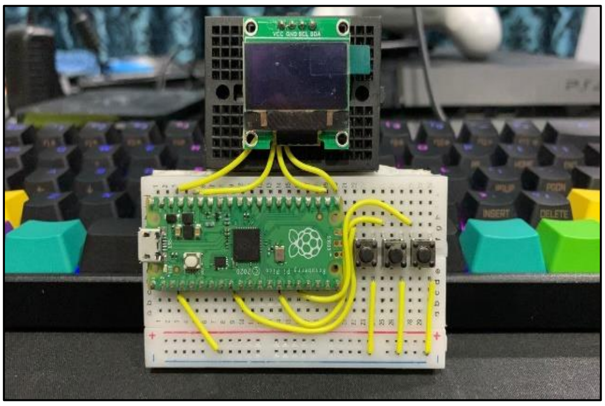


Fig. 5. Final device design

TABLE I. STYLES HARDWARE WIRING

OLED Display	Raspberry Pi Pico
GND	GND
VCC	3V3(OUT)
SCL	GP17
SDA	GP16
Buttons	**Raspberry Pi Pico**
Move Up	GP6
Move Down	GP10
Enter	GP12
Buttons	**Raspberry Pi Pico**
Move Up GND	GND
Move Down GND	GND
Enter GND	GP17

III. PROJECT EL PADRINO PENETRATION TESTING DEVICE

Fig 6. shows the Project El Padrino penetration testing device invented by the researcher. The device consists of a white breadboard that hardware such as Raspberry Pi Pico and three push buttons. The researcher also included a mini black breadboard to hold the OLED display. All the hardware in the Fig 6 is wired and connected by yellow jumper wires. The purpose of this device is to run any automated script. The researcher has provided five malware scripts. Each script has their own functions, for example:

- WifiPassword.dd → Grabs all the password of the WIFI that the victim machine has connected to before and sends a zip file of it to attacker's personal email. This email can be changed accordingly to user's personal email.
- WindowsDefender.dd→ Turns off the Windows Defender Virus Protection to make sure the other exploits run successfully without getting blocked by Windows Defender.
- ChromePassword.dd→ Grabs all the website, username and passwords that are saved in Google Chrome and sends the credentials text file to attacker's personal email. This email can be changed accordingly to user's personal email.
- WindowsDump.dd→Dumps the password for Windows machine and sends the text file to attacker's personal email. This email can be changed accordingly to user's personal email.
- ReverseShell.dd→Execute elpadrino.exe reverse shell malware which is retrieve from the server and connect to the listener in the application.

The first push button from the left is for moving up the list. The middle button is for moving down the list. And the final push button is used as an exploit button to run the script.

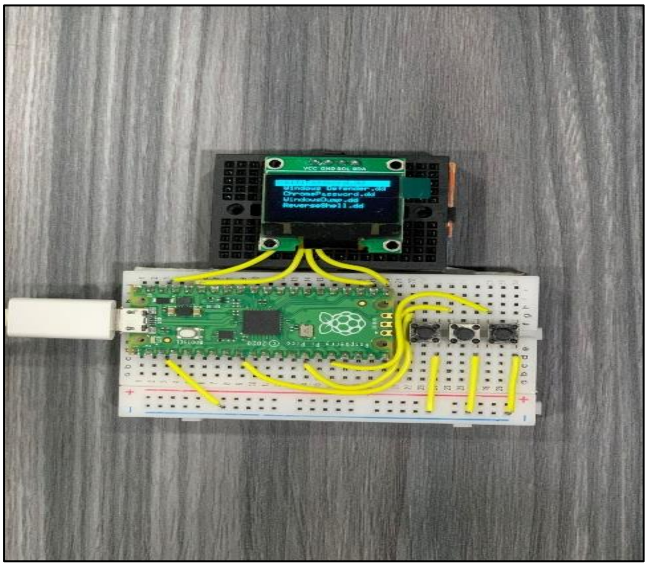
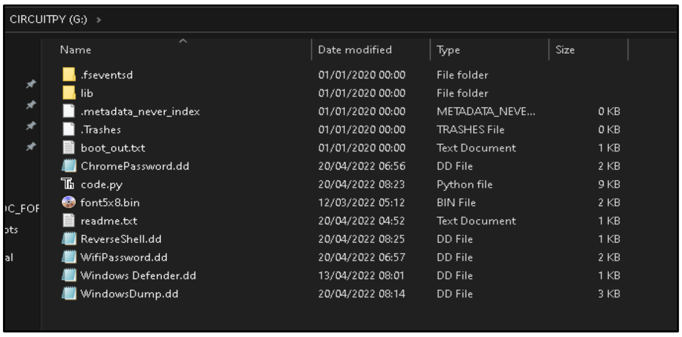


Fig. 6. Five malware scripts in the device

The Raspberry Pi Pico runs with Circuit Python; therefore, all the files are stored in the flash memory as shown in Fig 7. The code.py is the brain of these project where it contains all the functions for the device to run. The libraires for OLED are also located in the flash memory just like all the other libraries used in the code.py.



Name	Date modified	Type	Size
.fseventsd	01/01/2020 00:00	File folder	
lib	01/01/2020 00:00	File folder	
.metadata_never_index	01/01/2020 00:00	METADATA_NEVE...	0 KB
.Trashes	01/01/2020 00:00	TRASHES File	0 KB
boot_out.txt	01/01/2020 00:00	Text Document	1 KB
ChromePassword.dd	20/04/2022 06:56	DD File	2 KB
code.py	20/04/2022 08:23	Python file	9 KB
font5x8.bin	12/03/2022 05:12	BIN File	2 KB
readme.txt	20/04/2022 04:52	Text Document	1 KB
ReverseShell.dd	20/04/2022 08:25	DD File	1 KB
WifiPassword.dd	20/04/2022 06:57	DD File	2 KB
Windows Defender.dd	13/04/2022 08:01	DD File	1 KB
WindowsDump.dd	20/04/2022 08:14	DD File	3 KB

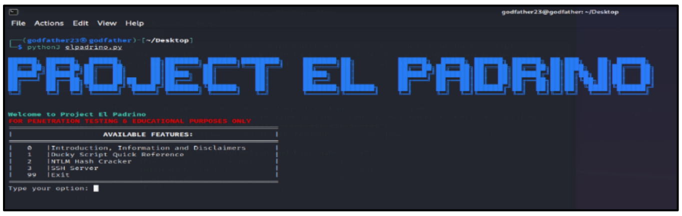
Fig. 7. The files located in the flash memory for Circuit Python

IV. PROJECT EL PADRINO CLI-BASED APPLICATION

A. *MAIN MENU*

Fig 8 shows Project El Padrino CLI-based application that was developed and coded in Python by the researcher. Fig 8. also shows that the 'elpadrino.py' file was executed with python3 in Kali Linux. The application begins with the tittle of this project along with a welcome message which is meant to be forwarded to the users of this application. The researcher also included a one-lined disclaimer to ensure that the users are aware of the ethics and rules of a penetration testing tool which can only be used for ethical penetration testing and for educational purposes only. Moving on, the primary function of the main menu is to list down all the available features that the users can use in this application. Beside the Introduction, Information and Disclaimers, the main features of this application are as follows:

- Ducky Script Quick Reference
- NTLM Hash Cracker
- SSH Server

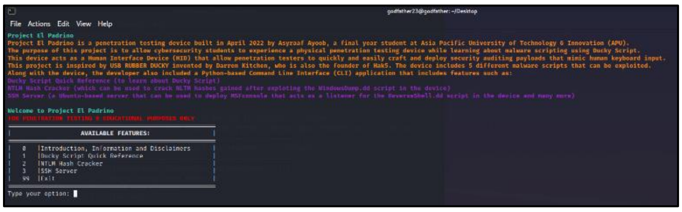


```
┌──(godfather23㉿godfather)-[~/Desktop]
└─$ python3 elpadrino.py
PROJECT EL PADRINO

Welcome to Project El Padrino
FOR PENETRATION TESTING & EDUCATIONAL PURPOSES ONLY
|                AVAILABLE FEATURES:                |
|   0   |Introduction, Information and Disclaimers  |
|   1   |Ducky Script Quick Reference               |
|   2   |NTLM Hash Cracker                          |
|   3   |SSH Server                                 |
|   99  |Exit                                       |
Type your option:
```

Fig. 8. Application main menu

B. *INTRODUCTION, INFORMATION AND DISCLAIMERS*

Fig 9. shows the Introduction, Information and Disclaimers section of this project. The researcher has given a short introduction on Project El Padrino and the features that are available in the device and the application.

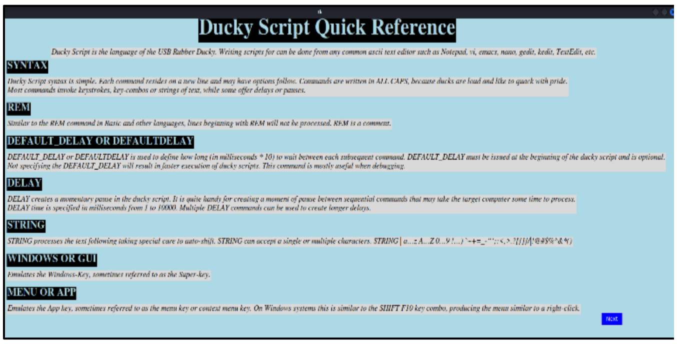


```
Project El Padrino
Project El Padrino is a penetration testing device built in April 2022 by Asyraaf Ayoob, a final year student at Asia Pacific University of Technology & Innovation (APU).
The purpose of this project is to allow cybersecurity students to experience a physical penetration testing device while learning about malware scripting using Ducky Script.
This device acts as a Human Interface Device (HID) that allow penetration testers to quickly and easily craft and deploy security auditing payloads that mimic human keyboard input.
This project is inspired by USB RUBBER DUCKY invented by Darren Kitchen, who is also the founder of Hak5. The device includes 5 different malware scripts that can be exploited.
Along with the device, the developer also included a Python-based Command Line Interface (CLI) application that includes features such as:
Ducky Script Quick Reference (to learn about Ducky Script)
NTLM Hash Cracker (which can be used to crack NLTM hashes gained after exploting the WindowsDump.dd script in the device)
SSH Server (a Ubuntu-based server that can be used to deploy MSFconsole that acts as a listener for the ReverseShell.dd script in the device and many more)

Welcome to Project El Padrino
FOR PENETRATION TESTING & EDUCATIONAL PURPOSES ONLY
|              AVAILABLE FEATURES:              |
|   0   |Introduction, Information and Disclaimers |
|   1   |Ducky Script Quick Reference              |
|   2   |NTLM Hash Cracker                         |
|   3   |SSH Server                                |
|   99  |Exit                                      |
Type your option:
```

Fig. 9. The Introduction, Information and Disclaimers for the application

C. *DUCKY SCRIPT QUICK REFERENCE*

Fig 10. shows the first main feature of this application which is the Ducky Script Quick Reference. This GUI-based reference was developed using Tkinter GUI library. The researcher has provided some references for the users to refer and understand on how the commands are used in the malware scripts provided. This is only the first part of the reference and to move to the second page, the user can click the 'Next' button below.



Ducky Script Quick Reference

Ducky Script is the language of the USB Rubber Ducky. Writing scripts for can be done from any common ascii text editor such as Notepad, vi, emacs, nano, gedit, kedit, TextEdit, etc.

SYNTAX

Ducky Script syntax is simple. Each command resides on a new line and may have options follow. Commands are written in ALL CAPS, because ducks are loud and like to quack with pride.
Most commands invoke keystrokes, key-combos or strings of text, while some offer delays or pauses.

REM

Similar to the REM command in Basic and other languages, lines beginning with REM will not be processed. REM is a comment.

DEFAULT_DELAY OR DEFAULTDELAY

*DEFAULT_DELAY or DEFAULTDELAY is used to define how long (in milliseconds * 10) to wait between each subsequent command. DEFAULT_DELAY must be issued at the beginning of the ducky script and is optional.*
Not specifying the DEFAULT_DELAY will result in faster execution of ducky scripts. This command is mostly useful when debugging.

DELAY

DELAY creates a momentary pause in the ducky script. It is quite handy for creating a moment of pause between sequential commands that may take the target computer some time to process.
DELAY time is specified in milliseconds from 1 to 10000. Multiple DELAY commands can be used to create longer delays.

STRING

STRING processes the text following taking special care to auto-shift. STRING can accept a single or multiple characters. STRING | a...z A...Z 0...9 !...) `~+=_-"';:<,>.?[{]}/|!@#$%^&()*

WINDOWS OR GUI

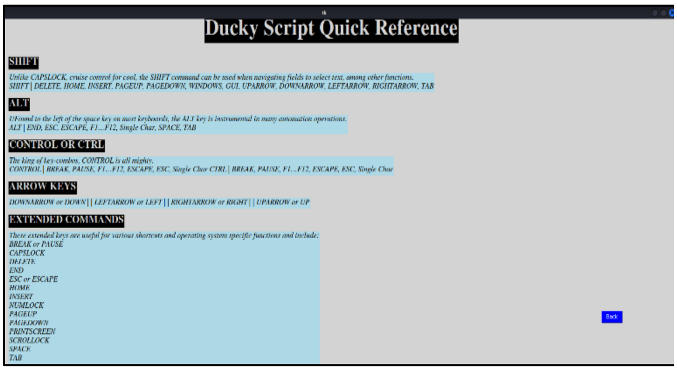
Emulates the Windows-Key, sometimes referred to as the Super-key.

MENU OR APP

Emulates the App key, sometimes referred to as the menu key or context menu key. On Windows systems this is similar to the SHIFT F10 key combo, producing the menu similar to a right-click.

Fig. 10. GUI-based Ducky Script Quick Reference in application

Based on the Fig 11., users are redirected to the second page after clicking the 'Next' button in the first page. The continuation of the reference is seen in the Fig 11. and if users decided to go back to the first page, they can click the 'Back' button. To close the Ducky Script Quick Reference, users can hit the 'x' on the top right as any other program. Subsequently, the main menu will prompt again.



Ducky Script Quick Reference

SHIFT

Unlike CAPSLOCK, cruise control for cool, the SHIFT command can be used when navigating fields to select text, among other functions.
SHIFT | DELETE, HOME, INSERT, PAGEUP, PAGEDOWN, WINDOWS, GUI, UPARROW, DOWNARROW, LEFTARROW, RIGHTARROW, TAB

ALT

UFound to the left of the space key on most keyboards, the ALT key is instrumental in many automation operations.
ALT | END, ESC, ESCAPE, F1...F12, Single Char, SPACE, TAB

CONTROL OR CTRL

The king of key-combos, CONTROL is all mighty.
CONTROL | BREAK, PAUSE, F1...F12, ESCAPE, ESC, Single Char CTRL | BREAK, PAUSE, F1...F12, ESCAPE, ESC, Single Char

ARROW KEYS

DOWNARROW or DOWN || LEFTARROW or LEFT || RIGHTARROW or RIGHT || UPARROW or UP

EXTENDED COMMANDS

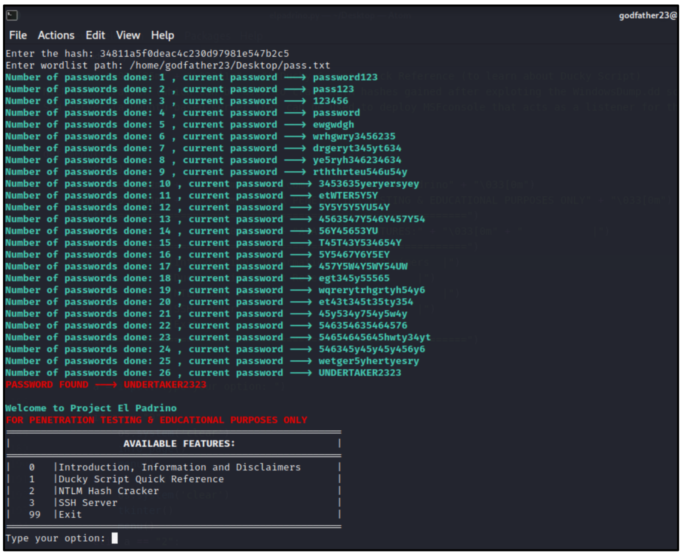
These extended keys are useful for various shortcuts and operating system specific functions and include:
BREAK or PAUSE
CAPSLOCK
DELETE
END
ESC or ESCAPE
HOME
INSERT
NUMLOCK
PAGEUP
PAGEDOWN
PRINTSCREEN
SCROLLOCK
SPACE
TAB

Fig. 11. Second part of Ducky Script Quick Reference

D. NTLM HASH CRACKER

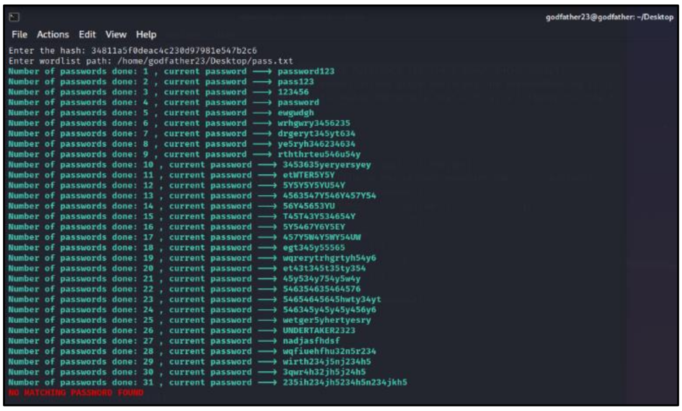
NTLM hash cracker are developed to work along with the WindowsDump.dd script. Besides that, it can be used to crack NTLM hashes for future encounters. Fig 12. shows the NTLM Hash Cracker in the application. The feature begins with an input where users need to input the NTLM hash that they want to crack. The second input will be the destination of the user's wordlist. If the hash provided matches with the password in the wordlist the application will prompt and highlight that the password is found. Not only that, but the system will also list the correct passwords. Fig 12. shows an example of a successful hash cracking. Looking into the details before the password is cracked, the system will list down the number of passwords done and also the current password. The purpose of this function is to track the passwords if the user provided a huge wordlist.

The NTLM hash cracker will prompt 'NO MATCHING PASSWORD FOUND' if there are no matching hashes with the password wordlist used by the user as shown in Fig.13.



```
Enter the hash: 34811a5f0deac4c230d97981e547b2c5
Enter wordlist path: /home/godfather23/Desktop/pass.txt
Number of passwords done: 1 , current password ⟶ password123
Number of passwords done: 2 , current password ⟶ pass123
Number of passwords done: 3 , current password ⟶ 123456
Number of passwords done: 4 , current password ⟶ password
Number of passwords done: 5 , current password ⟶ ewgwdgh
Number of passwords done: 6 , current password ⟶ wrhgwry3456235
Number of passwords done: 7 , current password ⟶ drgeryt345yt634
Number of passwords done: 8 , current password ⟶ ye5ryh346234634
Number of passwords done: 9 , current password ⟶ rththrteu546u54y
Number of passwords done: 10 , current password ⟶ 3453635yeryersyey
Number of passwords done: 11 , current password ⟶ etWTER5Y5Y
Number of passwords done: 12 , current password ⟶ 5Y5Y5Y5YU54Y
Number of passwords done: 13 , current password ⟶ 4563547Y546Y457Y54
Number of passwords done: 14 , current password ⟶ 56Y45653YU
Number of passwords done: 15 , current password ⟶ T45T43Y534654Y
Number of passwords done: 16 , current password ⟶ 5Y5467Y6Y5EY
Number of passwords done: 17 , current password ⟶ 457Y5W4Y5WY54UW
Number of passwords done: 18 , current password ⟶ egt345y55565
Number of passwords done: 19 , current password ⟶ wqrerytrhgrtyh54y6
Number of passwords done: 20 , current password ⟶ et43t345t35ty354
Number of passwords done: 21 , current password ⟶ 45y534y754y5w4y
Number of passwords done: 22 , current password ⟶ 546354635464576
Number of passwords done: 23 , current password ⟶ 54654645645hwty34yt
Number of passwords done: 24 , current password ⟶ 546345y45y45y456y6
Number of passwords done: 25 , current password ⟶ wetger5yhertyesry
Number of passwords done: 26 , current password ⟶ UNDERTAKER2323
PASSWORD FOUND ⟶ UNDERTAKER2323

Welcome to Project El Padrino
FOR PENETRATION TESTING & EDUCATIONAL PURPOSES ONLY
═══════════════════════════════════════════════════════
|                 AVAILABLE FEATURES:                 |
═══════════════════════════════════════════════════════
|   0   |Introduction, Information and Disclaimers     |
|   1   |Ducky Script Quick Reference                  |
|   2   |NTLM Hash Cracker                             |
|   3   |SSH Server                                    |
|   99  |Exit                                          |
═══════════════════════════════════════════════════════
Type your option: █
```

Fig. 12. A successfully hash cracked

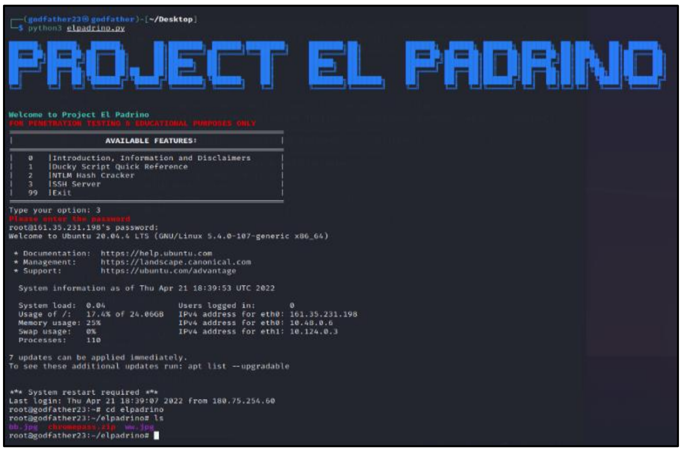


```
Enter the hash: 34811a5f0deac4c230d97981e547b2c6
Enter wordlist path: /home/godfather23/Desktop/pass.txt
Number of passwords done: 1 , current password ⟶ password123
Number of passwords done: 2 , current password ⟶ pass123
Number of passwords done: 3 , current password ⟶ 123456
Number of passwords done: 4 , current password ⟶ password
Number of passwords done: 5 , current password ⟶ ewgwdgh
Number of passwords done: 6 , current password ⟶ wrhgwry3456235
Number of passwords done: 7 , current password ⟶ drgeryt345yt634
Number of passwords done: 8 , current password ⟶ ye5ryh346234634
Number of passwords done: 9 , current password ⟶ rththrteu546u54y
Number of passwords done: 10 , current password ⟶ 3453635yeryersyey
Number of passwords done: 11 , current password ⟶ etWTER5Y5Y
Number of passwords done: 12 , current password ⟶ 5Y5Y5Y5YU54Y
Number of passwords done: 13 , current password ⟶ 4563547Y546Y457Y54
Number of passwords done: 14 , current password ⟶ 56Y45653YU
Number of passwords done: 15 , current password ⟶ T45T43Y534654Y
Number of passwords done: 16 , current password ⟶ 5Y5467Y6Y5EY
Number of passwords done: 17 , current password ⟶ 457Y5W4Y5WY54UW
Number of passwords done: 18 , current password ⟶ egt345y55565
Number of passwords done: 19 , current password ⟶ wqrerytrhgrtyh54y6
Number of passwords done: 20 , current password ⟶ et43t345t35ty354
Number of passwords done: 21 , current password ⟶ 45y534y754y5w4y
Number of passwords done: 22 , current password ⟶ 546354635464576
Number of passwords done: 23 , current password ⟶ 54654645645hwty34yt
Number of passwords done: 24 , current password ⟶ 546345y45y45y456y6
Number of passwords done: 25 , current password ⟶ wetger5yhertyesry
Number of passwords done: 26 , current password ⟶ UNDERTAKER2323
Number of passwords done: 27 , current password ⟶ nadjasfhdsf
Number of passwords done: 28 , current password ⟶ wqfiuehfhu32n5r234
Number of passwords done: 29 , current password ⟶ wirth234j5nj234h5
Number of passwords done: 30 , current password ⟶ 3qwr4h32jh5j24h5
Number of passwords done: 31 , current password ⟶ 235ih234jh5234h5n234jkh5
NO MATCHING PASSWORD FOUND
```

Fig. 13. No Matching Password found for the hash provided

E. SSH SERVER

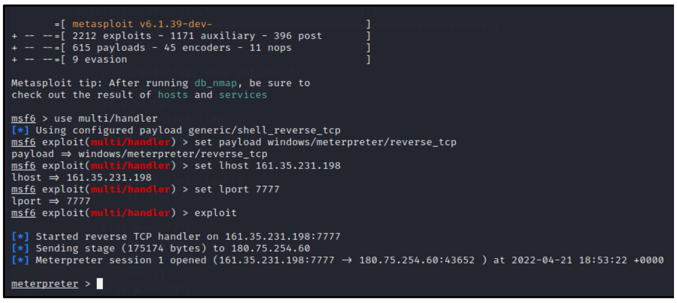
The SSH server is an Ubuntu-based server is created by the researcher using Digital Ocean as shown in Fig 14.,. The main purpose of this server is to keep all the malware files that will be used by WindowsDump.dd, ChromePassword.dd and ReverseShell.dd script. These files work respectively with their scripts that will initially be downloaded and exploited in the machine that is connected to the device.



```
┌──(godfather23㉿godfather)-[~/Desktop]
└─$ python3 elpadrino.py
PROJECT EL PADRINO

Welcome to Project El Padrino
FOR PENETRATION TESTING & EDUCATIONAL PURPOSES ONLY
═══════════════════════════════════════════════════════
|                 AVAILABLE FEATURES:                 |
═══════════════════════════════════════════════════════
|   0   |Introduction, Information and Disclaimers     |
|   1   |Ducky Script Quick Reference                  |
|   2   |NTLM Hash Cracker                             |
|   3   |SSH Server                                    |
|   99  |Exit                                          |
═══════════════════════════════════════════════════════
Type your option: 3
Please enter the password
root@161.35.231.198's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-107-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Apr 21 18:39:53 UTC 2022

  System load:  0.04              Users logged in:       0
  Usage of /:   17.4% of 24.06GB  IPv4 address for eth0: 161.35.231.198
  Memory usage: 25%               IPv4 address for eth0: 10.48.0.6
  Swap usage:   0%                IPv4 address for eth1: 10.124.0.3
  Processes:    110

7 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


*** System restart required ***
Last login: Thu Apr 21 18:39:07 2022 from 180.75.254.60
root@godfather23:~# cd elpadrino
root@godfather23:~/elpadrino# ls
bb.jpg  chromepass.zip  ww.jpg
root@godfather23:~/elpadrino# █
```

Fig. 14. The Ubuntu-based server in the application

The second purpose of the server is to run MSF Console as a listener for the ReverseShell.dd script in the device. The Fig 15 shows that the listener has connected to the victim's machine after the ReverseShell.dd script is exploited in the machine. The advantage of this feature is that users can use this exploit anywhere and at any time instead of running at localhost. The server is online for 24 hours. Besides the main purposes, users can utilize this server to learn other penetration testing tools by accessing this server through this application at any machine that they own. The password for this server is donPablo007x.



```
       =[ metasploit v6.1.39-dev-                         ]
+ -- --=[ 2212 exploits - 1171 auxiliary - 396 post       ]
+ -- --=[ 615 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: After running db_nmap, be sure to
check out the result of hosts and services

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 161.35.231.198
lhost ⇒ 161.35.231.198
msf6 exploit(multi/handler) > set lport 7777
lport ⇒ 7777
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 161.35.231.198:7777
[*] Sending stage (175174 bytes) to 180.75.254.60
[*] Meterpreter session 1 opened (161.35.231.198:7777 → 180.75.254.60:43652 ) at 2022-04-21 18:53:22 +0000

meterpreter > █
```

Fig. 15. Meterpreter session connected to ReverseShell.dd malware

V. Conclusions

The research and investigation part of this project has improved the quality of the device and application. The aim and objectives set in the paper, Conceptual Model of Penetration Testing Device: Project El Padrino have been achieved in this paper after the development of Project El Padrino. Cybersecurity students can now have their own physical device that can perform Bad-USB attacks. Besides the benefits for cybersecurity students, penetration testers can also use this device for automation in daily task. Penetration testers always face with the same tools that are being used often during penetration testing. And every time they use the tools, they have to run it manually one by one which will take up their time. As a solution, penetration testers can use this device to run their own automated scripts to perform penetration testing. This will save their time and increase work efficiency.

References

A. Andreatos, E. Karystinos and C. Douligeris, "Spyduino: Arduino as a HID Exploiting the BadUSB Vulnerability" (2019), Ieeexplore.ieee.org. [Online]. Available: https://ieeexplore.ieee.org/document/8804730.

"Ducky Script Quick Reference - USB Rubber Ducky", Docs.hak5.org (2022). [Online]. Available: https://docs.hak5.org/usb-rubber-ducky-1/the-ducky-script-language/ducky-script-quick-reference.

K. Hardson-Hurley, "Thonny: The Beginner-Friendly Python Editor – Real Python", Realpython.com (2022). [Online]. Available: https://realpython.com/python-thonny/.

K. Rembor, "Welcome to CircuitPython!", Adafruit Learning System (2017). [Online]. Available: https://learn.adafruit.com/welcome-to-circuitpython/what-is-circuitpython.

"Raspberry Pi Documentation - Raspberry Pi Pico", Raspberrypi.com (2022). [Online]. Available: https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html.