RESOURCES USED

- ➢GITHUB Repositories: used examples to help create and execute the python script
- ➢Used physical Hacki_Pi device to run the script
- ➢Powershell tool to also execute script-command in the background
- ➢Visual Studio code: executed python script
- ➢Thonny
- ➢Outlook email
- ➢ Google resources– search more on python and hacki_pi
- ➢Lecture notes
- ➢Chat Gpt
- ➢https://www.kickstarter.com/projects/diytech/hackypi-your-diy-tool-for-learning : Link used to understand Hacki-Pi
- ➢https://www.python.org/ : Link used to understand Python
- ➢GitHub Repositories:
    - ○ https://github.com/python/pythondotorg :

PRESENTATION GUIDE

Slide 1: Introduction

VAL = tot time 3 min

- **Welcome to our presentation on "THE EMUS Network Security- Python-Base Script Exploitation with HackyPi."**
- **Today, we delve into the world of cybersecurity by exploring the network and how hackers think.**
- **We'll demonstrate how a system can be exploited within seconds and the critical importance of efficient mitigation strategies.**

Slide 2: WHY WE CHOSE TO PWN YOU?!

VAL

- **We'll show how hackers think and act fast to exploit weaknesses.**
- **Stress the importance of quickly fixing these issues to stop cyber threats and avoid harm.**

Slide 3: BECOMING A "HACKER" IS ACCESSIBLE TO EVERYONE! Access to Hacker Resources

VAL

- **Anyone can explore the world of hacking.**
- **The internet is vast, offering resources on platforms like YouTube and GitHub Repositories.**
- **Now, Let's dive into the fundamentals of scripting and its role in retrieving information and automating actions with DIY devices.**

Slide 4: HackyPi- Python-based USB Drive Manipulation

VAL

- **Now let's introduce HackyPi, a Python-based tool for USB drive manipulation. HackyPi enables us to interact with USB devices using Python scripts. We'll discuss crafting Python scripts that utilize HackyPi to send commands, collect data, or emulate device behaviors, showcasing real-time interactions with USB drives.**

3-seconds. + Next, we'll have ANDRE take us through the upcoming slides

Slide 5: Choose Target Windows Devices

ANDRE = tot time 3 min

- **In this step, we focus on selecting Windows devices as our target platform. We delve into understanding HackyPI**

- **'s capabilities for interacting with USB devices and performing data extraction tasks. This foundational research prepares us for developing scripts that can interact with USB-connected devices on Windows systems.**

Slide 6: Developing Python Script (yes)

ANDRE

- **With a clear understanding of HackyPi's capabilities, we proceed to develop a Python script using HackyPy. This script is designed to access Wi-Fi information on Windows devices. Additionally, we leveraged Python to automate Outlook tasks, such as configuring server settings, sending emails, and attaching files. The goal is to create a comprehensive script that streamlines Wi-Fi access and Outlook automation.**

Slide 7: Research Wi-Fi Access Methods

ANDRE

- **In this step, we conduct detailed research into methods to access Wi-Fi information on Windows. We explored techniques and tools that allowed us to extract Wi-Fi profile names and passwords from the system. By utilizing HackyPi alongside Python, we aim to develop efficient and reliable methods for retrieving Wi-Fi details from targeted Windows devices.**

3-seconds. +Now, we'll get insights from Martina with a Live demostration.

Slide 8: Live demo Martina note:

Let's explore a Python script crafted to exploit Windows system WiFi profile names and passwords.
Let's begin by breaking down the script to understand how it works.

IMPORT Subprocess enables running system commands directly from Python, providing control over the operating system.
IMPORT CSV manages data by reading and writing CSV files within the script.

Email modules like 'email.mime.multipart', Text, and Base create emails with text and attachments.

Encoders prepare attachments for email transmission.

Os The module helps interact with the system and manage files.

get_wifi_profiles: Uses 'subprocess' to gather Windows WiFi data, saving it to a CSV.

send_email: Create emails with sender, recipient, content, and attachments, utilizing the SMTP library for delivery.

It uses encoders to ensure that the attachments are properly formatted for transmission.

After sending the email, the script cleans up by removing the attachment file.

This script showcases Python's ability to gather system data, handle CSV files, and send emails with attachments via SMTP, focusing on the Windows operating system.

Now let's jump right into this live demonstration..We'll plug in the Hacky-Pi to our USB port, which will automatically transfer data from our Windows system to email. The process happens in the background and in seconds ,

and soon we'll find a new email with a CSV attachment in our inbox.

3-seconds. +Alright! Let's now return to our slides for some insights " **= tot time 3 min**

Slide 9: Legal vs. Illegal Uses

ANDRE

- **Script-based exploits have ethical and unethical applications. Ethical uses include penetration testing, security assessments, and education. Conversely, illegal uses involve data theft, malware distribution, and identity fraud.**

3-seconds. + Alright, folks! Now that we've sorted the good from the bad with script-based exploits, let's dive into how this exploit actually works. Time to pass the mic to Trevor for the techy details!"

Slide 10: Understanding Script-Based Exploits

TREVOR = tot time 3 min

- **Scripting involves crafting sequences of commands for computer systems, automating tasks, and potentially bypassing security controls. With scripting, we can execute commands swiftly, boosting efficiency and allowing for faster exploitation while minimizing detection.**

Slide 11: Importance of System Updates

TREVOR

- **Keeping systems updated is crucial to patching vulnerabilities that could be exploited by scripts. Regular updates ensure that our devices are fortified against potential intrusions.**

Slide 12- Mitigation MORE WAYS TO OUTSMART THE SNEAKY STUFF ?

TREVOR

- **Proactive measures such as restricting physical access to devices, updating system software, and improving firewall policies are vital to mitigating script-based exploits.**

Slide 13: Conclusion

TREVOR

- **In conclusion, with this demonstration, we observed how a straightforward script was executed via simulated keystrokes to carry out actions on the target system. This also serves as a reminder of the importance of cybersecurity awareness and the responsible use of scripting tools to protect against potential threats. Just remember to stay vigilant and keep your cybersecurity practices in check. Never underestimate peoples' abilities or intentions.**

- **Thank you for joining our presentation on cybersecurity and script exploitation using HackyPy. Let's work together to strengthen our defenses and safeguard against evolving threats in the digital landscape.**

Slide 14: ANY QUESTIONS?

RED- DO NOT READ THE TITLE BLUE- READ THE TITLE
TAKE YOUR TIME DO NOT RUSH!