USB Attack Workshop

(Using the USBNugget)

What we’re doing today

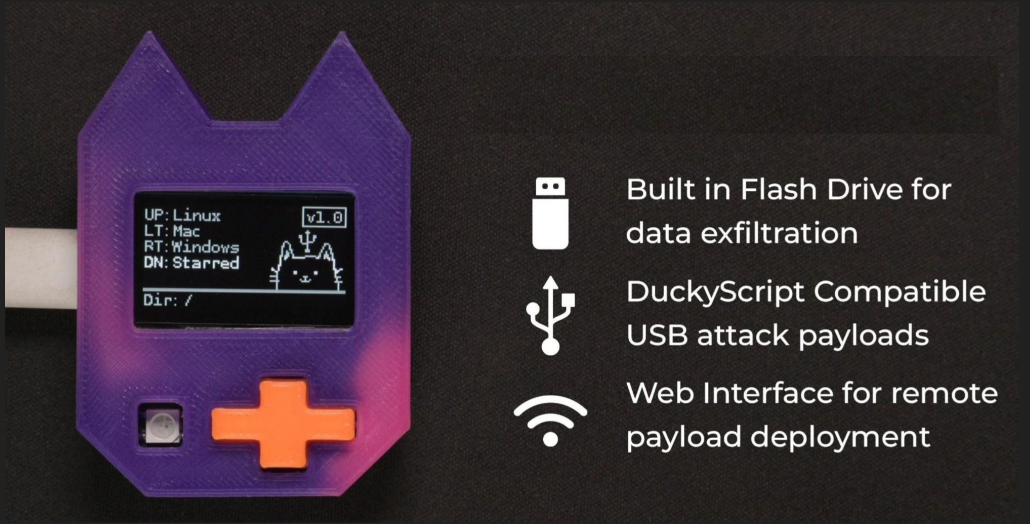
- Learning about HID / USB attacks
- Learning to flash a firmware to your USBNugget
- Writing your own CatScratch keystroke injection payloads
- Learning how to use the USBNugget for security testing
- USB attack CTF with a Raspberry Pi to win a prize

What is the USB Nugget?

The USB Nugget is a hardware tool that makes it easy for beginners to learn **hacking techniques** and **hardware development**!

Features:

- Built-in screen
- RGB LED
- 4 D-Pad style buttons
- WiFi Microcontroller
- Plug-and-play expansion pins



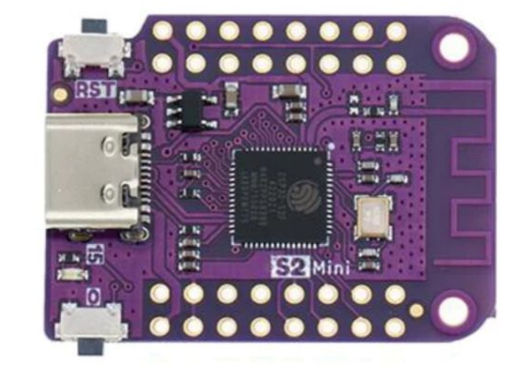
Built in Flash Drive for data exfiltration

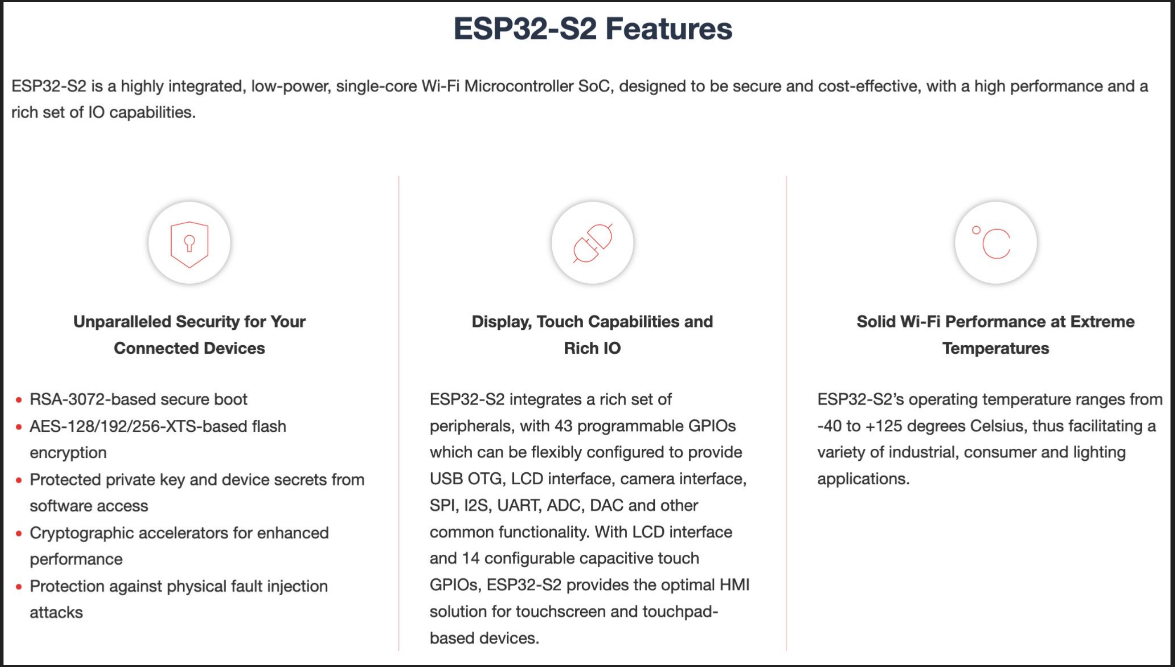
DuckyScript Compatible USB attack payloads

Web Interface for remote payload deployment

What’s under the hood?

The USB Nugget is powered by the ESP32s2, a Wi-Fi enabled microcontroller





ESP32-S2 Features

ESP32-S2 is a highly integrated, low-power, single-core Wi-Fi Microcontroller SoC, designed to be secure and cost-effective, with a high performance and a rich set of IO capabilities.

Unparalleled Security for Your Connected Devices

- RSA-3072-based secure boot
- AES-128/192/256-XTS-based flash encryption
- Protected private key and device secrets from software access
- Cryptographic accelerators for enhanced performance
- Protection against physical fault injection attacks

Display, Touch Capabilities and Rich IO

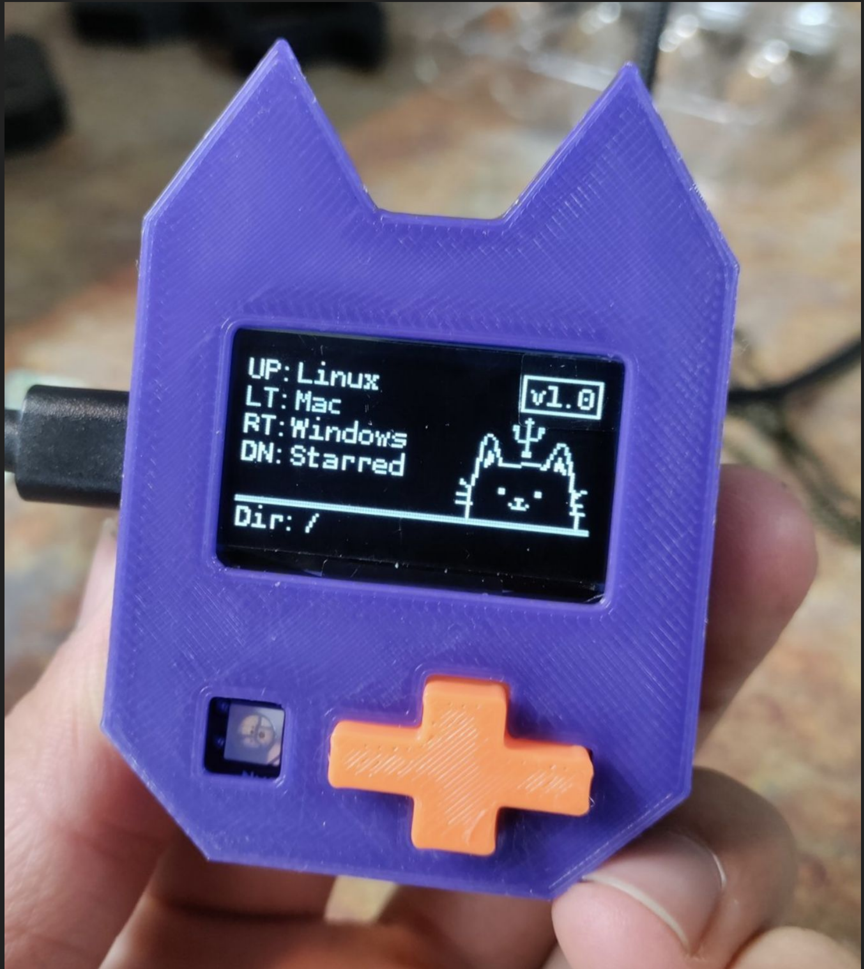
ESP32-S2 integrates a rich set of peripherals, with 43 programmable GPIOs which can be flexibly configured to provide USB OTG, LCD interface, camera interface, SPI, I2S, UART, ADC, DAC and other common functionality. With LCD interface and 14 configurable capacitive touch GPIOs, ESP32-S2 provides the optimal HMI solution for touchscreen and touchpad-based devices.

Solid Wi-Fi Performance at Extreme Temperatures

ESP32-S2’s operating temperature ranges from -40 to +125 degrees Celsius, thus facilitating a variety of industrial, consumer and lighting applications.

What can the USB Nugget do?

- Run USB Attacks
- Teach programming
  - CircuitPython
  - Arduino
- Control Hardware / Sensors
- Run Community Projects
- Display cute animations!



What is the USB Nugget OS software?

The USB Nugget OS is a program that lets you run USB Attacks in seconds using the USB Nugget!

Current Features:

- CatScratch Compatible
- Built-in flash drive storage
- Store and run Payloads
- WiFi Control

USB Attack Class

1 Hour

What are USB & HID Attacks?

- **USB attacks** pretend to be a trusted USB device like a keyboard or an ethernet adapter.
- **HID (Human Interface Device) attacks** pretend to be a keyboard or mouse since these are plug and play.
- Anything you can do behind a computer with a keyboard or mouse can be automated.

What is Keystroke Injection?

- We can make our nugget pretend to be a keyboard
- Computers will trust it and let it type
- We can take advantage of this to do things very fast

Common HID Attacks & Use Cases

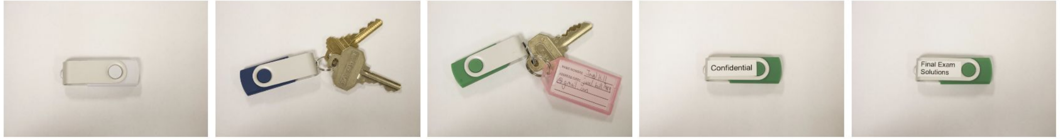
Common HID attacks involve taking advantage of physical access to a device.

This could look like:

- Dropping malicious USB drives in a parking lot
- Running a payload on an unattended laptop by plugging in a USB Nugget
- Preventing a screen from locking by plugging in a device that jiggles the mouse

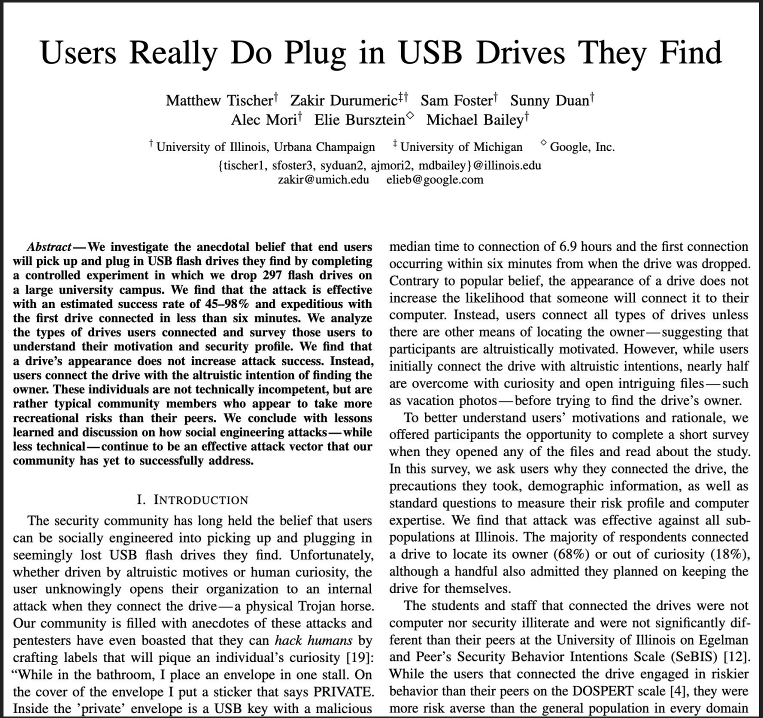
Do HID Attacks work?

- Yes! A study showed that 45% of USB drives left on a university campus were plugged in



(a) Unlabeled drive (b) Drive with keys (c) Drive with return label (d) Confidential drive (e) Exam solutions drive

Fig. 1: **Drive Appearances**—We dropped five different types of drives. We chose two appearances (keys and return label) to motivate altruism and two appearances (confidential and exam solutions) to motivate self-interest, as well as an unlabeled control.



Users Really Do Plug in USB Drives They Find

Matthew Tischer[†] Zakir Durumeric[‡†] Sam Foster[†] Sunny Duan[†]
Alec Mori[†] Elie Bursztein[◇] Michael Bailey[†]

[†] University of Illinois, Urbana Champaign [‡] University of Michigan [◇] Google, Inc.
{tischer1, sfoster3, syduan2, ajmori2, mdbailey}@illinois.edu
zakir@umich.edu elieb@google.com

***Abstract*—We investigate the anecdotal belief that end users will pick up and plug in USB flash drives they find by completing a controlled experiment in which we drop 297 flash drives on a large university campus. We find that the attack is effective with an estimated success rate of 45–98% and expeditious with the first drive connected in less than six minutes. We analyze the types of drives users connected and survey those users to understand their motivation and security profile. We find that a drive's appearance does not increase attack success. Instead, users connect the drive with the altruistic intention of finding the owner. These individuals are not technically incompetent, but are rather typical community members who appear to take more recreational risks than their peers. We conclude with lessons learned and discussion on how social engineering attacks—while less technical—continue to be an effective attack vector that our community has yet to successfully address.**

I. Introduction

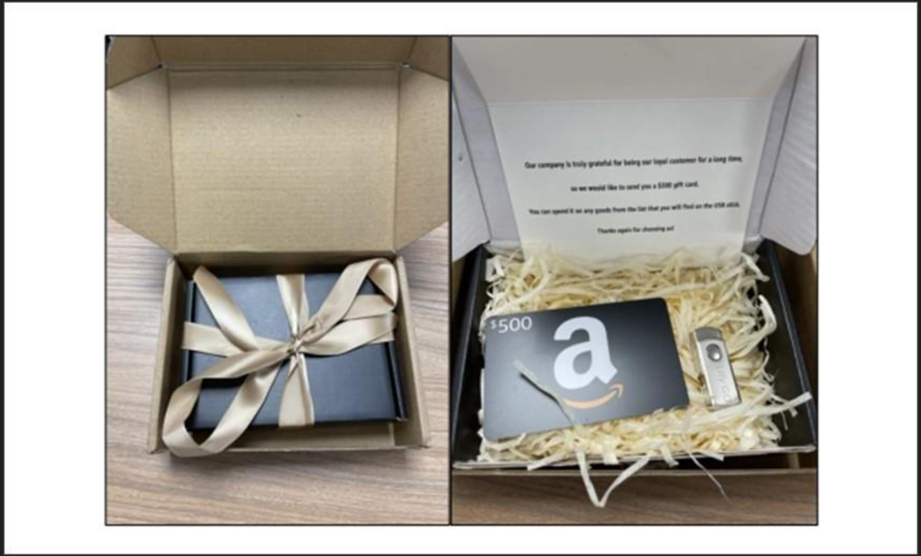
The security community has long held the belief that users can be socially engineered into picking up and plugging in seemingly lost USB flash drives they find. Unfortunately, whether driven by altruistic motives or human curiosity, the user unknowingly opens their organization to an internal attack when they connect the drive—a physical Trojan horse. Our community is filled with anecdotes of these attacks and pentesters have even boasted that they can *hack humans* by crafting labels that will pique an individual's curiosity [19]: "While in the bathroom, I place an envelope in one stall. On the cover of the envelope I put a sticker that says PRIVATE. Inside the 'private' envelope is a USB key with a malicious

median time to connection of 6.9 hours and the first connection occurring within six minutes from when the drive was dropped. Contrary to popular belief, the appearance of a drive does not increase the likelihood that someone will connect it to their computer. Instead, users connect all types of drives unless there are other means of locating the owner—suggesting that participants are altruistically motivated. However, while users initially connect the drive with altruistic intentions, nearly half are overcome with curiosity and open intriguing files—such as vacation photos—before trying to find the drive's owner.

To better understand users' motivations and rationale, we offered participants the opportunity to complete a short survey when they opened any of the files and read about the study. In this survey, we ask users why they connected the drive, the precautions they took, demographic information, as well as standard questions to measure their risk profile and computer expertise. We find that attack was effective against all sub-populations at Illinois. The majority of respondents connected a drive to locate its owner (68%) or out of curiosity (18%), although a handful also admitted they planned on keeping the drive for themselves.

The students and staff that connected the drives were not computer nor security illiterate and were not significantly different than their peers at the University of Illinois on Egelman and Peer's Security Behavior Intentions Scale (SeBIS) [12]. While the users that connected the drive engaged in riskier behavior than their peers on the DOSPERT scale [4], they were more risk averse than the general population in every domain

Real Life Scenario: Fin7 USB Mailing Attack



Our company is truly grateful for being our loyal customer for a long time,

so we would like to send you a $500 gift card.

You can spend it on any goods from the list that you will find on the USB stick.

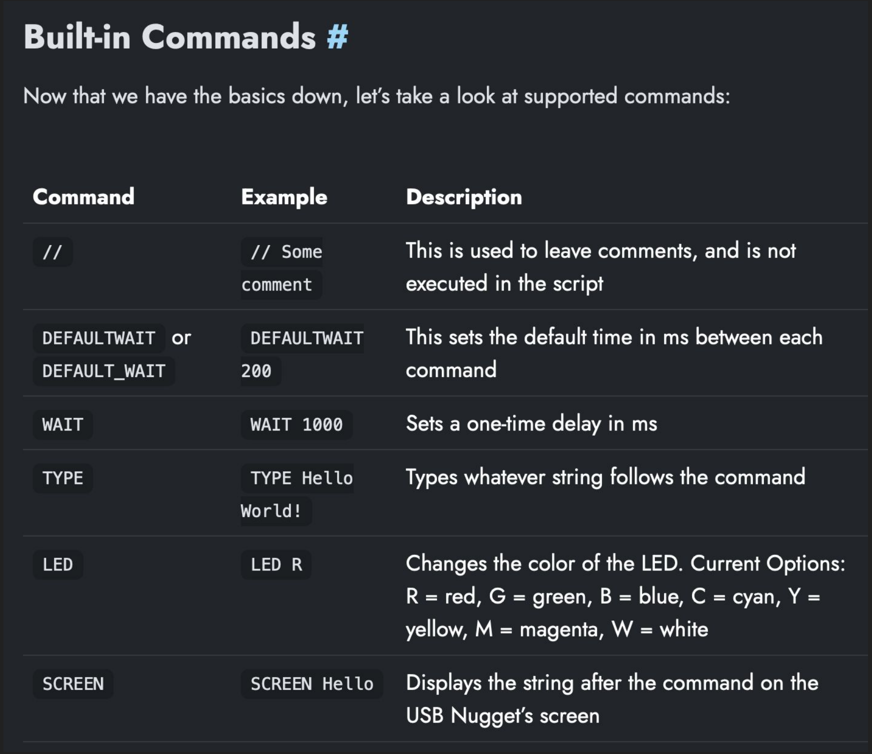
Thanks again for choosing us!

Fin7 Cybercrime group suspected of mailing malicious USB drives to install ransomware onto targets’ computers

What is CatScratch?

CatScratch is a simple language for scripting keyboard-based HID attacks

- Each CatScratch command is on a new line
- Commands are written in ALL CAPS
- Most commands invoke keystrokes, key-combos or strings of text
- Others commands create delays or pauses



Built-in Commands

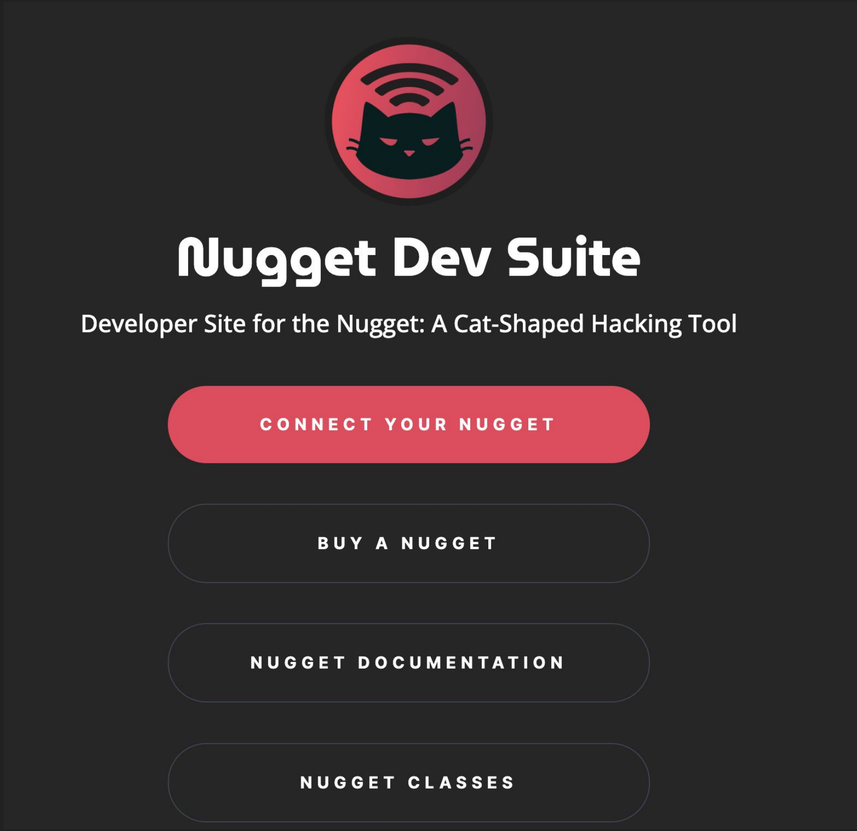
Now that we have the basics down, let's take a look at supported commands:

Command	Example	Description
`//`	`// Some comment`	This is used to leave comments, and is not executed in the script
`DEFAULTWAIT` or `DEFAULT_WAIT`	`DEFAULTWAIT 200`	This sets the default time in ms between each command
`WAIT`	`WAIT 1000`	Sets a one-time delay in ms
`TYPE`	`TYPE Hello World!`	Types whatever string follows the command
`LED`	`LED R`	Changes the color of the LED. Current Options: R = red, G = green, B = blue, C = cyan, Y = yellow, M = magenta, W = white
`SCREEN`	`SCREEN Hello`	Displays the string after the command on the USB Nugget's screen

Getting Started w/ the USB Nugget

How to Flash your USB Nugget

- Open Google Chrome and navigate to: www.nugget.dev

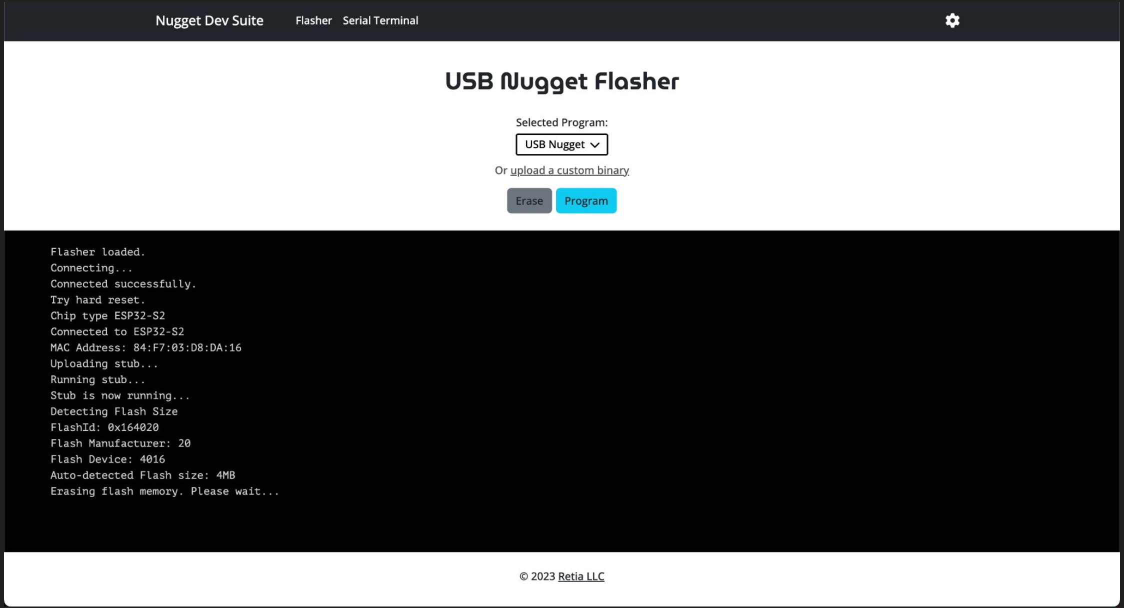


Nugget Dev Suite

Developer Site for the Nugget: A Cat-Shaped Hacking Tool

How to Flash your USB Nugget

- Click connect, select your Nugget, then click “Erase”



USB Nugget Flasher

Selected Program: USB Nugget

Or upload a custom binary

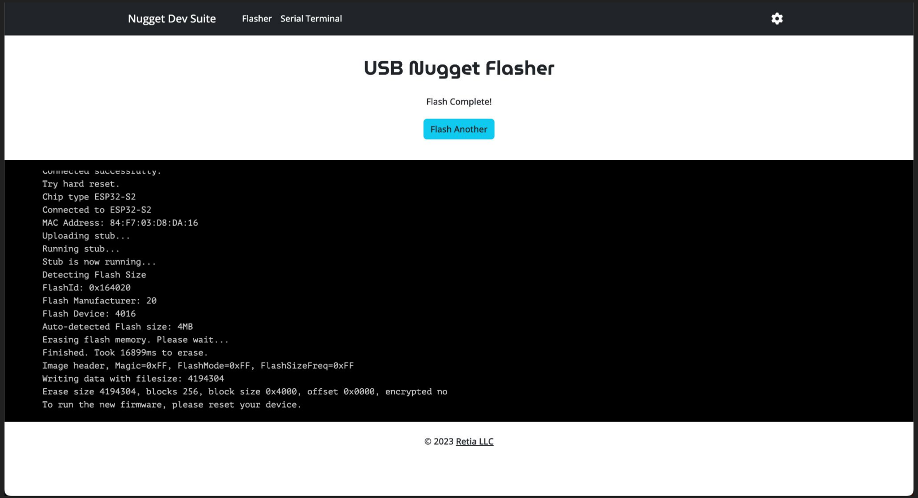
```
Flasher loaded.
Connecting...
Connected successfully.
Try hard reset.
Chip type ESP32-S2
Connected to ESP32-S2
MAC Address: 84:F7:03:D8:DA:16
Uploading stub...
Running stub...
Stub is now running...
Detecting Flash Size
FlashId: 0x164020
Flash Manufacturer: 20
Flash Device: 4016
Auto-detected Flash size: 4MB
Erasing flash memory. Please wait...
```

© 2023 Retia LLC

How to Flash your USB Nugget

- Once erasing is finished, select the .BIN file you downloaded and click “program” to flash the Nugget.

- Unplug your Nugget once the flashing is done. Your Nugget is ready to hack!



```
Connected successfully.
Try hard reset.
Chip type ESP32-S2
Connected to ESP32-S2
MAC Address: 84:F7:03:D8:DA:16
Uploading stub...
Running stub...
Stub is now running...
Detecting Flash Size
FlashId: 0x164020
Flash Manufacturer: 20
Flash Device: 4016
Auto-detected Flash size: 4MB
Erasing flash memory. Please wait...
Finished. Took 16899ms to erase.
Image header, Magic=0xFF, FlashMode=0xFF, FlashSizeFreq=0xFF
Writing data with filesize: 4194304
Erase size 4194304, blocks 256, block size 0x4000, offset 0x0000, encrypted no
To run the new firmware, please reset your device.
```

© 2023 Retia LLC

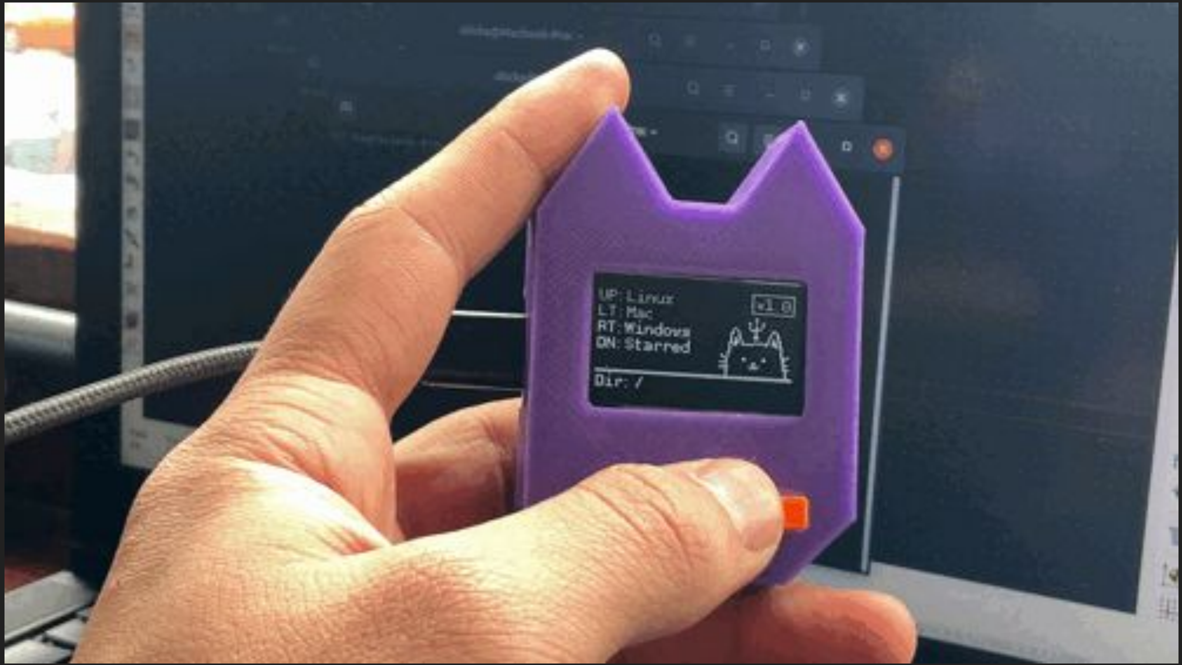
Break - Let’s Flash!

- Once erasing is finished, select the .BIN file you downloaded and click “program” to flash the Nugget.

- Unplug your Nugget once the flashing is done. Your Nugget is ready to hack!

How Run Payloads On the USBNugget

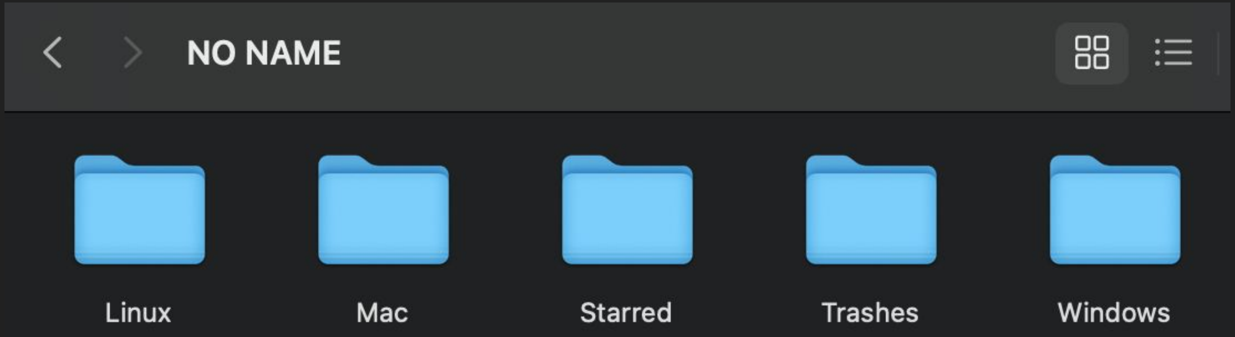
To run a payload, use the buttons to select the target OS, payload type, and the payload to run



How to Add Payloads to the USB Nugget

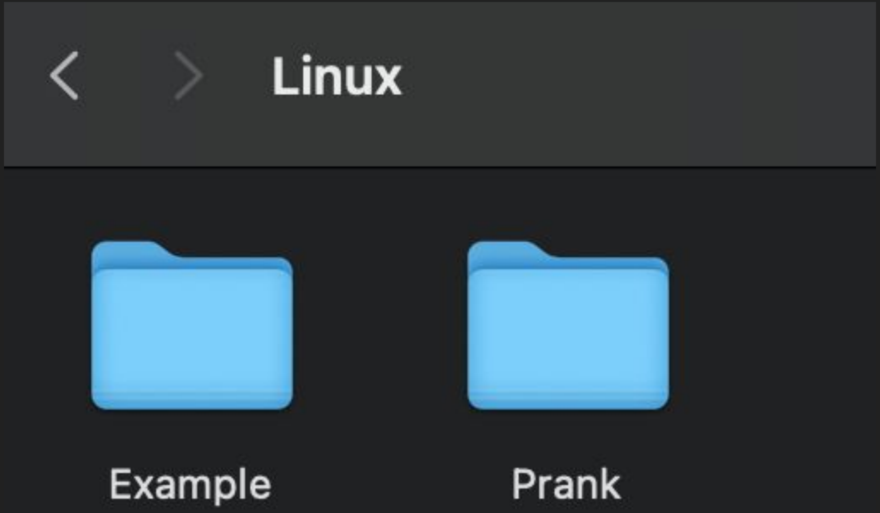
Plug in your Nugget and look for a USB drive to appear.

You’ll see folders to keep payloads for different operating systems.



How to Add Payloads to the USB Nugget

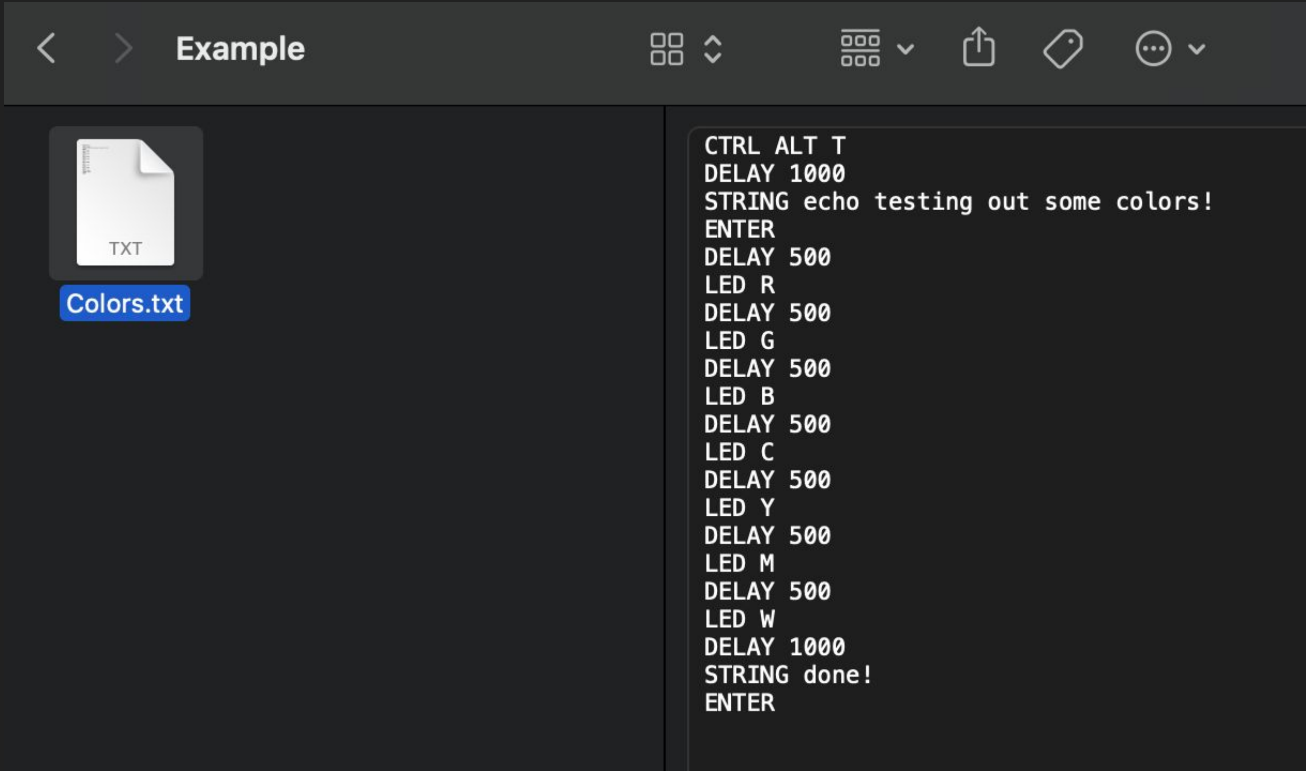
Inside each OS folder you’ll see a folder for types of payloads. The default is examples and pranks, but you can add more.



How to Add Payloads to the USB Nugget

Finally, you'll see a payload .TXT file.

You can open and edit it in your favorite text editor to change a payload!



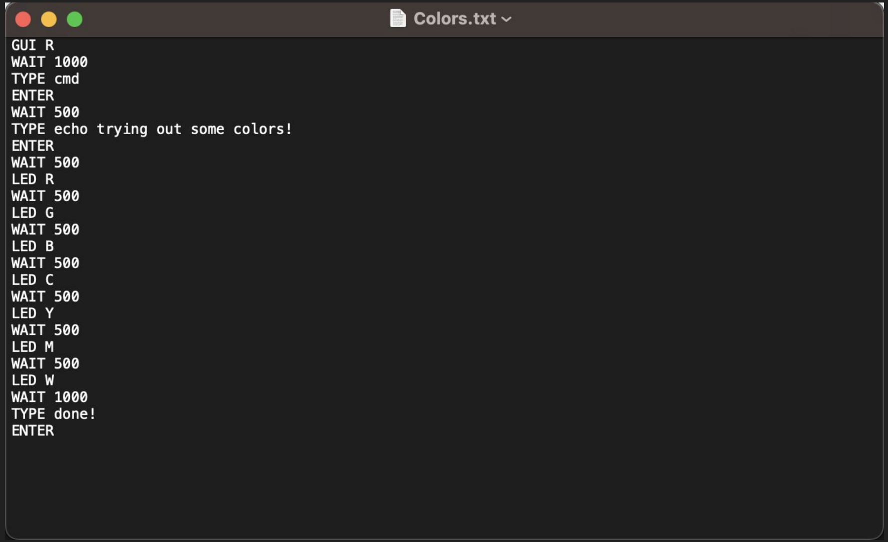
```
CTRL ALT T
DELAY 1000
STRING echo testing out some colors!
ENTER
DELAY 500
LED R
DELAY 500
LED G
DELAY 500
LED B
DELAY 500
LED C
DELAY 500
LED Y
DELAY 500
LED M
DELAY 500
LED W
DELAY 1000
STRING done!
ENTER
```

Coding your first CatScratch payload

There are example scripts preloaded on your USB Nugget!

Open any in a text editor, and make a change.

Save the script, and when your Nugget re-loads, press the button for the script you edited to see it run!



```
GUI R
WAIT 1000
TYPE cmd
ENTER
WAIT 500
TYPE echo trying out some colors!
ENTER
WAIT 500
LED R
WAIT 500
LED G
WAIT 500
LED B
WAIT 500
LED C
WAIT 500
LED Y
WAIT 500
LED M
WAIT 500
LED W
WAIT 1000
TYPE done!
ENTER
```

CatScratch Payloads

Get Started Creating Payloads!

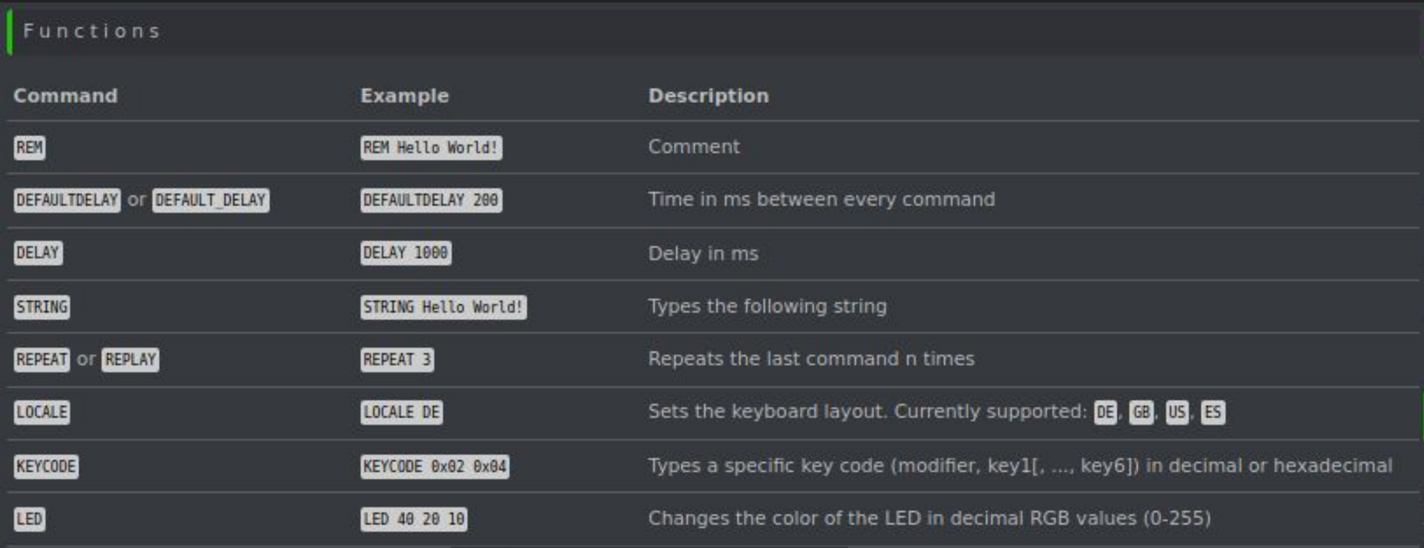
Methodology: Working Backwards

To write code for the USB Nugget OS, we need to work backwards from what we want to do. We’ll be creating some basic scripts based on how you do simple actions on your computer.

To design your first script, think about something you do all the time on your computer that you could accomplish with only a keyboard.

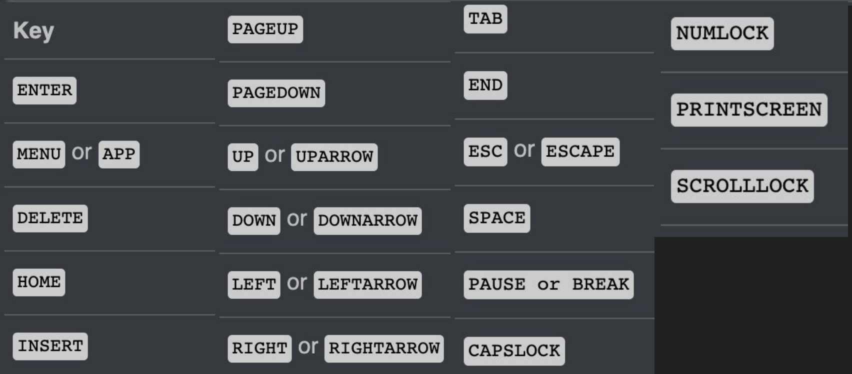
Break down the steps into a list of things you need to do to accomplish the task. In general, getting to the command line is the fastest way to take advantage of the USB Nugget OS’s speed.

Basic CatScratch Commands



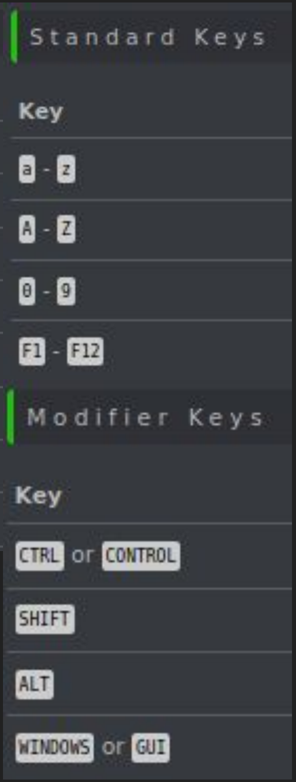
Functions

Command	Example	Description
REM	REM Hello World!	Comment
DEFAULTDELAY or DEFAULT_DELAY	DEFAULTDELAY 200	Time in ms between every command
DELAY	DELAY 1000	Delay in ms
STRING	STRING Hello World!	Types the following string
REPEAT or REPLAY	REPEAT 3	Repeats the last command n times
LOCALE	LOCALE DE	Sets the keyboard layout. Currently supported: DE, GB, US, ES
KEYCODE	KEYCODE 0x02 0x04	Types a specific key code (modifier, key1[, ..., key6]) in decimal or hexadecimal
LED	LED 40 20 10	Changes the color of the LED in decimal RGB values (0-255)



Key	PAGEUP	TAB	NUMLOCK
ENTER	PAGEDOWN	END	PRINTSCREEN
MENU or APP	UP or UPARROW	ESC or ESCAPE	SCROLLLOCK
DELETE	DOWN or DOWNARROW	SPACE	
HOME	LEFT or LEFTARROW	PAUSE or BREAK	
INSERT	RIGHT or RIGHTARROW	CAPSLOCK	

Graphic design is my passion



Standard Keys

Key
a - z
A - Z
0 - 9
F1 - F12

Modifier Keys

Key
CTRL or CONTROL
SHIFT
ALT
WINDOWS or GUI

Delays and timing

Delays make one-way scripts possible.

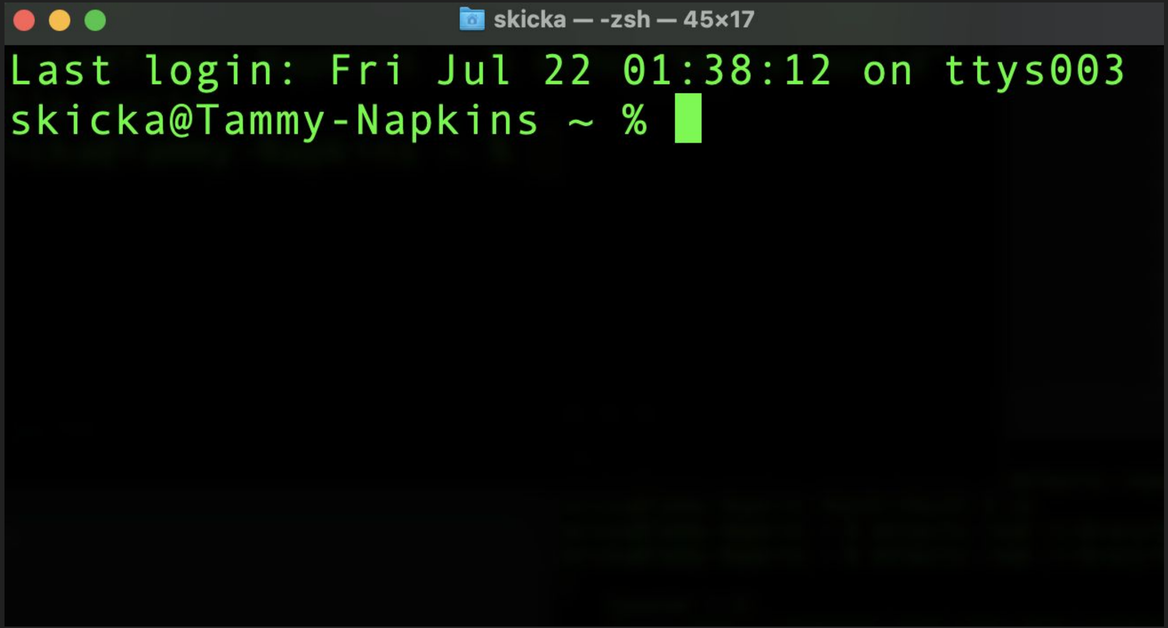
Because microcontrollers work so quickly, many of the commands would not work without adding time for commands to finish.

In testing, we should start out with generous delays and move into a more optimized design that works quickly without breaking anything.

Strategy: Race to the Terminal

The fastest way to do bad things on a computer is opening a terminal or powershell window.

What’s the fastest way to open a terminal window?



```
Last login: Fri Jul 22 01:38:12 on ttys003
skicka@Tammy-Napkins ~ %
```

Terminal Shortcuts

Quickest way to open a terminal on different operating systems.

Linux: CTRL ALT T

Mac: Cmd SPACE

Windows:

- GUI R - opens run dialog
- cmd - types a program
- ENTER - opens command prompt

Using Keyboard Shortcuts

Windows 10 Keyboard Shortcuts: https://www.windowscentral.com/best-windows-10-keyboard-shortcuts

Linux Keyboard Shortcuts (Debian): www.computerhope.com/ushort.htm

MacOS Keyboard Shortcuts: https://support.apple.com/en-us/HT201236

Raspberry Pi OS Shortcuts:

https://defkey.com/raspbian-raspberry-pi-shortcuts https://defkey.com/raspbian-raspberry-pi-shortcuts

Example Payload 1: RickRoll

Payload Breakdown:

- Open a terminal window
- Launch browser of choice (Chrome or Firefox)
- Open a custom url (like a Youtube Video)
- Play video and open fullscreen

STRING

DELAY

ENTER

F11 - fullscreen

Example Payload 1: PseudoCode

- **Press a key combo** to open a terminal window
- **Wait** for Terminal to open
- **Type** in a command to launch chrome / firefox
- **Wait** for browser to open
- **Type** in the url
- **Press enter**
- **Wait** for url to load
- **Press a function key** for full screen

Hint:

“start firefox” or “firefox” can be used to launch firefox from a terminal. You can also launch a url with this command.

Example Payload 2: Ransom Message

- Open a terminal window
- Use volume keys or a command to turn up the volume
- Use “say” or “espeak” to demand a dogecoin ransom to be paid
- Open a full screen browser window to a fake ransomware window:
    - https://www.cryptoprank.com/#/crypto

Hint: function keys can be used to raise the volume.

Example Payload 3: Advanced Ransom

Add-on the following objectives to the previous payload:

- Create a text file on the user’s desktop that contains the dogecoin ransom text
- After letting the website run for 10 seconds, lock the user’s computer

Hint: Use a keyboard shortcut to lock the computer. Creating a text file via command line is the fastest method, but you can also manually open notepad or another text editor.

Example Payload 4: Data Exfiltration

Finally, lets try out a data exfiltration payload!

Setup

- Generate a **web url bug** Canary Token at: canarytokens.org
- Copy the token link

Payload

- Open a terminal and get the MAC address (hardware address)
- Using the Curl tool, create a web request to the Canary Token that exfiltrates the MAC address in the user-agent field.

Payload Repository

For more payloads, check out the Hak5 Payload Repository:

https://hak5.org/blogs/payloads/

Links to USB Rubber Ducky Payloads

- Payload - Non-Malicious Auto Defacer
- Payload - Lock Your Computer Message
- Payload - Ducky Downloader
- Payload - Ducky Phisher
- Payload - FTP Download / Upload
- Payload - Restart Prank
- Payload - Silly Mouse, Windows is for Kids
- Payload - Windows Screen rotation hack
- Payload - Powershell Wget + Execute
- Payload - mimikatz payload
- Payload - MobileTabs
- Payload - Ugly Rolled Prank
- Payload - XMAS
- Payload - Pineapple Assocation (VERY FAST)
- Payload - Remotely Possible
- Payload - Batch Wiper/Drive Eraser
- Payload - Generic Batch
- Payload - Paint Hack
- Payload - Local DNS Poisoning
- Payload - Deny Net Access
- Payload - RunEXE from SD
- Payload - Run Java from SD
- Payload - Download mimikatz, grab passwords and email them via gmail
- Payload - Hotdog Wallpaper
- Payload - Android 5.x Lockscreen
- Payload - Chrome Password Stealer
- Payload - Website Lock
- Payload - Windows 10 : Download & Change Wallpaper
- Payload - Windows 10 : Download & Change Wallpaper another version
- Payload - Windows 10 : Download and execute file with Powershell
- Payload - Windows 10 : Disable windows defender
- Payload - Windows 10 : Disable Windows Defender through powershell
- Payload - Windows 10 : Wifi, Chrome Dump & email results
- Payload - Windows 7 : Logoff Prank
- Payload - Netcat Reverse Shell
- Payload - Fake Update screen
- Payload - Rickroll
- Payload - Fast Meterpreter
- Payload - Data-Exfiltration / Backdoor
- Payload - Fake Update screen
- Payload - OSX Sudo Passwords Grabber
- Payload - OSX Root Backdoor
- Payload - OSX User Backdoor
- Payload - OSX Local DNS Poisoning
- Payload - OSX Youtube Blaster
- Payload - OSX Photo Booth Prank
- Payload - OSX Internet Protocol Slurp
- Payload - OSX Ascii Prank
- Payload - OSX iMessage Capture
- Payload - OS X Wget and Execute
- Payload - OSX Passwordless SSH access (ssh keys)
- Payload - OSX Bella RAT Installation
- Payload - OSX Sudo for all users without password
- Payload - MrGray's Rubber Hacks
- Payload - Copy File to Desktop
- Payload - Youtube Roll
- Payload - Disable AVG 2012
- Payload - Disable AVG 2013
- Payload - EICAR AV test

CTF: Attack a Raspberry Pi

30 Min.

Example Actions

- Steal a file
- Delete a file
- Write a file with a message in it
- Steal a hash
- Corrupt a hash
- Kill the computer
- Plant a keylogger
- Rickroll
- Join rogue Wi-Fi network
- Team ASCII banner
- Grabify link tracker
- Cron task
- Netcat backdoor
- Change background
- Auto-restart computer
- Auto-quit programs

Pseudocode: Inject Payload Into Raspberry Pi

What are the steps we need to write code for?

```
Delay for the keyboard to be recognized
Open the run menu by pressing ALT and F2 at the same time
Wait for it to open
Type “lxterminal” to search for the Terminal application
A brief delay to finish typing
Press enter
Wait about 5 seconds for the window to open
Write whatever string we want
Wait to finish typing
Press enter
A short delay before the final line
Pressing Control and D at the same time closes the Terminal window
```

Final Break - Try some scripts!

Take a break, after this, we’ll be trying out our new skills in a team CTF!

CTF: Design the Highest Scoring Payload

In our last section, we’ll be working together to write payloads to win a prize!

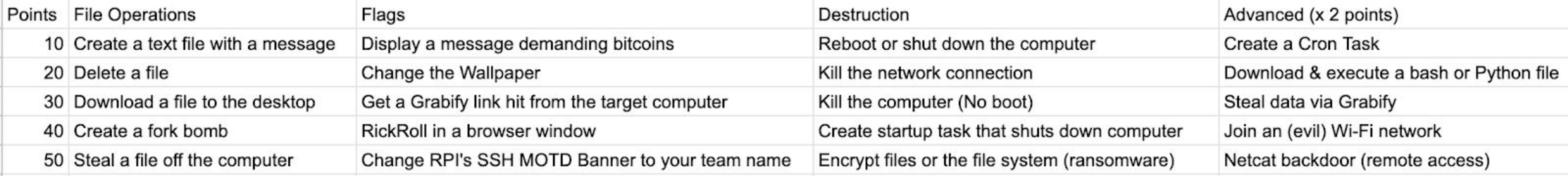
Our target is a Raspberry Pi computer running Raspbian. Your goal is to work as a team to make a payload that does the most number of bad things.

CTF Challenge: Attack The Raspberry Pi

For our final challenge, we’ll be dividing into teams and working on HID attack scripts to achieve a number of specific goals.

Each team will get time to write their script, and then 90 seconds to plug in and run their script.

The team to earn the most number of points wins a prize! Points are awarded when a team achieves the actions below:



Points	File Operations	Flags	Destruction	Advanced (x 2 points)
10	Create a text file with a message	Display a message demanding bitcoins	Reboot or shut down the computer	Create a Cron Task
20	Delete a file	Change the Wallpaper	Kill the network connection	Download & execute a bash or Python file
30	Download a file to the desktop	Get a Grabify link hit from the target computer	Kill the computer (No boot)	Steal data via Grabify
40	Create a fork bomb	RickRoll in a browser window	Create startup task that shuts down computer	Join an (evil) Wi-Fi network
50	Steal a file off the computer	Change RPI's SSH MOTD Banner to your team name	Encrypt files or the file system (ransomware)	Netcat backdoor (remote access)