Pico Pi Ducky

Thank you again for your purchase of the Pico Pi Ducky! Follow us on Instagram @unlimited.coverage or check out our shopify https://unlimitedcoverage.myshopify.com/ for more products and updates.

Feel free to email us at @unlimitedcoveragestore@gmail.com with any addition questions or concerns.

There are a few things to note about the Pico Pi ducky:

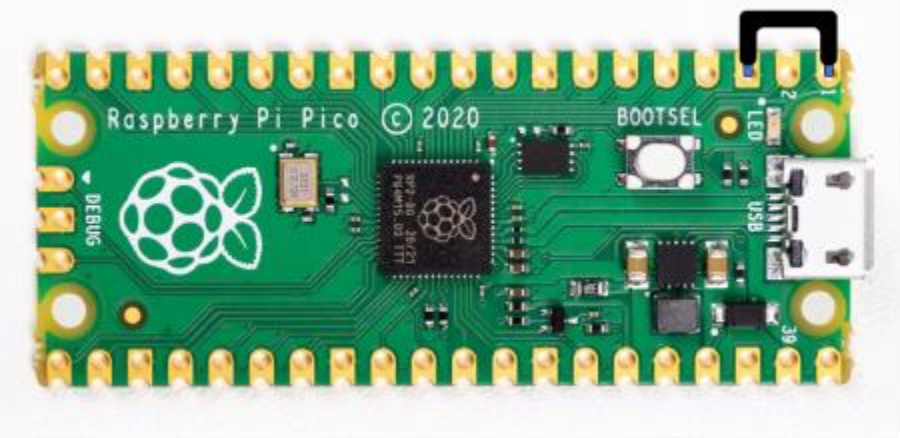
1. when it is plugged in, it will still be identified as a USB, but it will be able to execute script commands as if it were a keyboard.

2. When downloading a payload script, MAKE SURE to either have the Pico Pi ducky in setup mode (shown below), or to unplug the Pico the second the payload file has been downloaded to the Pico, as it will run the second it is downloaded if not in setup mode.

3. The Pico does not come with scripts pre-installed both for your safety, and so that you can chose the scripts you want, it is an incredibly simple process to load the scripts onto the Pico after we configure it. The setup instructions are below.

Setup mode

To edit the payload, enter setup mode by connecting pin 1 (GP0) to pin 3 (GND), this will stop the pico-ducky from injecting the payload in your own machine. The easiest way to do so is by using a jumper wire between those pins as seen below.



Pico Ducky Setup:

1. Plug your Pico Ducky into your computer using the cord provided. There are adapters for Apple computers and other models without a USB port. You should see it pop up as CIRCUITPY.

2. Remove this starter guide file from the USB and follow the rest of the instructions. The Pico may not work if this file stays on it.

3. Navigate to https://github.com/hak5/usbrubberducky-payloads/tree/master/payloads/library

4. Select the payload file you want from any of the folders in the library. Currently Pico-Ducky only supports DuckyScript 1.0 not 3.0. NOTE: We provided summaries of some of the more popular attacks in the library, but some folders have over 25+ attack files so just select the one that fits your needs best.

5. Once you have selected and navigated to the payload file you want, copy the script and paste it into a notepad or text editor. Make sure that you are making and saving this file OUTSIDE of the Pico Pi device, on your desktop or elsewhere.

6. Save the text file and set the name as payload.dd. Make sure that it saves as a .dd file and has no caps.

7. Once you are ready, copy the payload.dd file into the CIRCUITPY ROOT folder (the main folder that holds all the files and folders on the Pico, just click on the CIRCUITPY device and you will be in the root folder). After this, your Pico Pi is ready to be plugged in and used!

8. **To edit files and Payloads:** This is a slightly inconvenient step as it is hard to edit the files on the Pico Ducky without it running the script once it has been placed on the drive, unless it is in setup mode. Try to ensure that you have the correct script files written and downloaded onto Pico initially. If you must edit the files of delete a payload, follow this guide:

Follow these instructions if your Pico ends up in an odd state or if you need to edit scripts:

Download the reset firmware from flash_nuke.uf2

While holding the BOOTSEL button on the Pico, plug in the USB cable to your computer.

When the RPI-RP2 drive shows up on your computer, copy the flash_nuke.uf2 file to the Pico

After the device reboots, follow the Install instructions here

Attacks:

Credentials Folder:

BitLockerKeyDump:

Purpose: This script attempts to extract the BitLocker recovery keys from a target Windows system.

Mechanism: It likely uses Windows commands to access the stored recovery keys, leveraging the 'manage-bde' command or similar methods to dump the keys into a file, which can then be exfiltrated.

Effect: If successful, it can provide the attacker with the recovery keys to access BitLocker-encrypted drives.

Browser-Passwords-Dropbox-Exfiltration:

Purpose: This script is designed to extract saved passwords from web browsers and upload them to Dropbox.

Mechanism: It uses tools or scripts to access the password storage of browsers (like Chrome or Firefox), collects the data, and then uses Dropbox's API or a pre-configured Dropbox folder to upload the stolen credentials.

Effect: Compromises the victim's online accounts by stealing stored browser credentials and securely transferring them to the attacker's Dropbox.

DevilsCupid:

Purpose: The exact function is unclear from the name alone, but it likely involves some form of credential harvesting or keylogging.

Mechanism: Typically, such scripts might involve setting up a keylogger or phishing attempt to steal user credentials.

Effect: Depending on its exact functionality, it could lead to credential theft or unauthorized access to sensitive information.

DuckyLogger:

Purpose: This is a keylogger script meant to capture all keystrokes on the target machine.

Mechanism: It installs a keylogging program or script which runs in the background, recording keystrokes and periodically sending the logs to the attacker.

Effect: Steals sensitive information such as passwords, personal messages, and other typed data.

DuckyLogger2:

Purpose: A likely updated or alternative version of the DuckyLogger script.

Mechanism: Similar to DuckyLogger, it probably includes improvements or different methods for keylogging.

Effect: Similar to DuckyLogger, it captures keystrokes for credential theft and data exfiltration.

ExfiltrateWiFiPasswords_Linux:

Purpose: To extract and exfiltrate saved WiFi passwords from a Linux system.

Mechanism: It uses Linux commands to access and retrieve saved WiFi passwords from network manager or wpa_supplicant files, then sends this data to the attacker.

Effect: Provides the attacker with access to WiFi networks that the victim's device has connected to.

Funni_Stick_V3:

Purpose: This script's name suggests a potentially less serious or more experimental payload, possibly involving credential theft or system manipulation.

Mechanism: Depending on its exact script, it might include credential harvesting or deploying a humorous payload with some data exfiltration.

Effect: Effects can vary, potentially causing disruption or stealing credentials.

SamDumpDucky:

Purpose: Designed to dump SAM (Security Account Manager) database hashes from a Windows system.

Mechanism: Uses tools like mimikatz or built-in Windows commands to extract password hashes stored in the SAM database.

Effect: Allows an attacker to obtain hashed passwords of user accounts, which can then be cracked offline to retrieve plaintext passwords.

Simple_User_Password_Grabber:

Purpose: To grab and possibly display the currently logged-in user's password.

Mechanism: This might involve using a phishing prompt or another method to trick the user into entering their password, or exploiting system weaknesses to retrieve the password.

Effect: Directly compromises the user's account by capturing their password.

WLAN-Windows-Passwords:

Purpose: To retrieve saved WiFi passwords from a Windows system.

Mechanism: Uses Windows command line tools like netsh wlan show profile to list and extract WiFi credentials.

Effect: Provides the attacker with passwords to WiFi networks the victim's machine has connected to.

WindowsLicenseKeyExfiltration:

Purpose: To extract and exfiltrate the Windows license key.

Mechanism: Uses registry queries or Windows Management Instrumentation (WMI) to retrieve the license key.

Effect: Compromises the victim's Windows license key, which could potentially be used for illegal activation of Windows on other machines.

datacopier:

Purpose: To copy files or data from the victim's machine to the attacker's storage.

Mechanism: Likely uses simple file copy commands to transfer specified files or directories.

Effect: Exfiltrates potentially sensitive files from the victim's machine.

sudoSnatch:

Purpose: To capture sudo passwords from Linux systems.

Mechanism: This script might use a phishing method to prompt the user for their sudo password or exploit terminal vulnerabilities to capture it.

Effect: Provides the attacker with elevated privileges on the target Linux system by capturing sudo passwords.

Exfiltration folder:

Bash-History

Mechanism: This script retrieves the history of bash commands executed on a Linux system. It locates the .bash_history file and copies its contents.

Effect: Exfiltrating bash history can reveal sensitive command-line activities, such as executed commands that include passwords or other confidential information.

ClipBoard-Creep

Mechanism: This payload monitors the clipboard and captures its contents. It then sends the captured data to a specified endpoint.

Effect: Clipboard data can include copied passwords, credit card numbers, and other sensitive information, making it a valuable target for attackers.

Copy-And-Waste

Mechanism: This script copies all files from a targeted directory and then deletes them, sending the copied files to a remote server.

Effect: This attack results in data exfiltration and potential data loss, making it a destructive method for stealing information.

Create_And_Exfiltrate_A_Webhook_Of_Discord

Mechanism: This script creates a Discord webhook and uses it to send data from the victim’s machine to a Discord channel.

Effect: Leveraging Discord for data exfiltration can bypass traditional security measures and alerting systems, as it uses a legitimate service for malicious purposes.

DUCKY-WIFI_GRABER

Mechanism: This payload extracts saved Wi-Fi passwords from a Windows machine using command-line tools and exfiltrates them to an external server.

Effect: Compromising Wi-Fi credentials allows attackers to gain unauthorized access to wireless networks, potentially leading to further breaches within the network.

Discord_Windows_Wifi_IP-Info

Mechanism: This script gathers Wi-Fi SSIDs, passwords, and the device's IP configuration, then sends this information to a Discord webhook.

Effect: By obtaining network and IP information along with Wi-Fi credentials, attackers can map the network and prepare for further attacks.

Dropbox-Bandit

Mechanism: This payload utilizes Dropbox to upload and exfiltrate files from the victim's machine. It automates the process of logging into Dropbox and uploading selected files.

Effect: Using Dropbox for exfiltration can evade detection by security systems since it involves legitimate cloud storage services.

Dump_Windows_Memory_Through_ProcDump

Mechanism: This script uses Sysinternals ProcDump to dump the memory of a Windows machine and then exfiltrates the dump file.

Effect: Memory dumps can contain sensitive data such as passwords, encryption keys, and other in-memory secrets, providing a rich source of information for attackers.

Execution Folder:

$MFT-Duck-Crasher:

Mechanism: This script targets the Master File Table (MFT) of NTFS file systems, creating a scenario where the system cannot access files.

Effect: By creating a malformed file name and saving it to the MFT, it can cause Windows systems to become unresponsive or crash due to the corruption of the file system. This is a denial-of-service (DoS) attack that makes the system unstable.

Add_Local_Admin:

Mechanism: This script adds a new user to the local administrators group on a Windows machine using PowerShell commands.

Effect: This provides the attacker with administrative privileges on the compromised machine, allowing full control over the system, installation of software, and modification of security settings.

Admin_Who_Never_Sleeps:

Mechanism: This script creates a hidden administrator account on a Windows system.

Effect: The hidden account remains unknown to the legitimate users and administrators of the system, allowing the attacker to maintain persistent access with elevated privileges.

BeEF_Injection:

Mechanism: This script injects a hook from the Browser Exploitation Framework (BeEF) into a target's browser. This is typically done by opening a malicious URL that includes the BeEF hook.

Effect: Once the hook is injected, the attacker can control the browser, steal credentials, and execute arbitrary JavaScript in the context of the target's web session.

Change_Windows_User_Name:

Mechanism: This script changes the username of a Windows account using command line instructions.

Effect: Changing the username can confuse legitimate users and administrators, potentially leading to account lockouts or loss of access to personalized settings and files associated with the original username.

Disable_Windows_Defender22H2:

Mechanism: This script disables Windows Defender using PowerShell commands.

Effect: Disabling Windows Defender lowers the security defenses of the system, making it vulnerable to malware and other malicious activities. This creates an opportunity for further exploitation without being detected.

Persistent_Reverse_Shell-Telegram_Based:

Mechanism: This script establishes a persistent reverse shell that communicates with the attacker via Telegram. It uses PowerShell to create a backdoor and sends connection details to a specified Telegram bot.

Effect: The attacker gains continuous remote access to the victim's system through Telegram, allowing for data exfiltration, system manipulation, and further attack deployment.

Starting_a_PowerShell_with_administrator_permissions_in_Windows:

Mechanism: This script opens a PowerShell session with elevated (administrator) permissions.

Effect: With administrative PowerShell access, an attacker can perform a wide range of malicious activities, such as modifying system configurations, installing software, disabling security features, and exfiltrating sensitive data.

General Folder:

-RD-PineApple:

This script sets up a WiFi Pineapple device, which is used for network auditing and penetration testing. The payload likely configures the WiFi Pineapple to create a rogue WiFi access point, allowing the attacker to capture network traffic and potentially gain unauthorized access to sensitive data on the network.

Ascii:

The Ascii payload uses ALT codes to produce ASCII art or messages on the target machine. This is more of a harmless prank or demonstration script, which shows how Rubber Ducky can automate keystrokes to produce text or patterns.

Defeat_Defender:

This payload aims to disable Windows Defender on the target machine. By executing commands to turn off real-time protection and other security features, it compromises the security of the system, making it vulnerable to further attacks or malware.

DuckyCave-Game:

This script installs and launches a game called "DuckyCave" on the target computer. It demonstrates the capability of using Rubber Ducky to deploy and execute software automatically, which can be used for both benign and malicious purposes.

EngagementDucky:

This payload is used to engage or distract the user. It may open various applications or websites, generate pop-ups, or perform other actions to keep the user occupied while another payload performs more covert operations in the background.

Hotfix_Warning:

This script simulates a warning message about a required hotfix. It can be used as a social engineering tactic to trick users into performing certain actions, such as disabling security settings or installing malicious software disguised as a legitimate update.

Multi_HID_The-Penny-Drops:

This payload demonstrates the use of multiple HID (Human Interface Device) functions. It can perform a variety of tasks like typing, moving the mouse, or executing scripts. The versatility of this payload showcases how Rubber Ducky can mimic various input devices to control the target system.

Open4Gmail:

This script opens the Gmail login page in the target's browser. It might be used to phish for user credentials by redirecting the user to a fake login page or capturing their keystrokes as they enter their credentials.

Incident Response Folder:

RD-ET-Phone-Home

Mechanism: This payload establishes a connection back to a remote server, essentially creating a reverse shell. It uses PowerShell or another scripting tool to open a command-and-control channel.

Effect: This allows the attacker to remotely control the compromised machine, execute commands, and potentially exfiltrate data. This is useful in scenarios where the attacker needs persistent access to the system.

Auto-Check_Cisco_IOS_XE_Backdoor_based_on_CVE-2023-20198_and_CVE-2023-20273

Mechanism: This script automatically checks for backdoors in Cisco IOS XE devices based on the specified CVEs. It runs commands to identify if the vulnerabilities are present and can be exploited.

Effect: If the backdoor is found, it can allow unauthorized access to the network devices, leading to potential data exfiltration or network disruption. This is crucial for network administrators to identify and patch these vulnerabilities quickly.

Defend_yourself_against_CVE-2023-36884_Office_and_Windows_HTML_R

Mechanism: This payload implements defenses against the CVE-2023-36884 vulnerability affecting Microsoft Office and Windows. It likely alters system configurations or patches the vulnerability.

Effect: Protects the system from exploitation by this specific CVE, preventing remote code execution attacks that could be initiated through malicious Office documents or Windows HTML files.

Exploit_Citrix_NetScaler_ADC_and_Gateway_through_CVE-2023-4966

Mechanism: This script exploits the specified CVE to gain unauthorized access to Citrix NetScaler ADC and Gateway devices. It runs a series of commands to leverage the vulnerability.

Effect: Successful exploitation can lead to full control over the affected devices, allowing for data exfiltration, service disruption, or further network infiltration.

GoodUSB

Mechanism: This payload likely involves a methodology or script designed to test USB ports and devices for security compliance. It might check for unauthorized devices or enforce security policies.

Effect: Ensures that USB ports and devices are secure and compliant with security policies, reducing the risk of USB-based attacks and unauthorized data transfers.

Mobile Folder:

IOS – Attacks on IOS devices using adapter

Android – Attacks on Android devices using adapter

Prank Folder:

RD-ADV-RickRoll

Mechanism: This payload is designed to play the "Rick Astley - Never Gonna Give You Up" video at maximum volume on the target machine. It likely opens a web browser and navigates to a YouTube link for the song.

Effect: The user is "Rickrolled," a popular internet prank where unsuspecting users are tricked into watching the music video. This can be disruptive and amusing but generally harmless.

RD-JumpScare-2.0

Mechanism: This payload executes a jump scare by displaying a frightening image or video on the target machine, accompanied by a loud, startling sound. It uses scripting to open media files.

Effect: The target user experiences a sudden scare, causing a moment of shock or fright. This is meant to be a playful prank but can be unsettling for the target.

RD-Rage-PopUps

Mechanism: This payload creates a series of continuous pop-up windows on the target's screen. It uses scripting to open multiple message boxes in rapid succession.

Effect: The user becomes annoyed or frustrated due to the overwhelming number of pop-ups. This can disrupt productivity and requires the user to close each pop-up manually.

RD-We-Found-You

Mechanism: This payload displays a message indicating that the target user has been "found" or "located." It uses scripting to open a message box with a custom message.

Effect: The target user may feel uneasy or paranoid, thinking they have been tracked or identified. This prank leverages psychological impact rather than technical disruption.

AUTOinCORRECT

Mechanism: This payload modifies the autocorrect settings on the target machine to replace common words with humorous or incorrect alternatives. It uses scripting to access and change system settings.

Effect: The target user experiences frustration and confusion when typing, as their input is unexpectedly altered. This prank can be amusing for observers and irritating for the user.

Always-Minimize

Mechanism: This payload continuously minimizes all open windows on the target machine. It uses scripting to send minimize commands to the system.

Effect: The user is unable to keep any window open, leading to confusion and annoyance. This disrupts normal workflow and forces the user to repeatedly reopen minimized windows.

The_Mouse_Moves_By_Itself

Mechanism: This payload randomly moves the mouse cursor on the target machine. It uses scripting to simulate mouse movements at random intervals.

Effect: The target user experiences difficulty controlling their mouse, leading to confusion and frustration. This prank can be subtle but highly disruptive over time.

RickRoll_ASCII

Mechanism: This payload displays an ASCII art version of the "Rick Astley - Never Gonna Give You Up" music video lyrics in a command prompt or terminal window. It uses scripting to print the ASCII art.

Effect: The user is "Rickrolled" in a more creative and technical way. This prank is generally harmless and serves as an amusing surprise for the target.

Recon Folder:

Drop_Zip_Execute

Mechanism: This payload drops a ZIP file onto the target machine and then executes a script or executable contained within the ZIP. It uses scripting to download the ZIP file from a remote server, extract its contents, and run the included executable.

Effect: This allows the attacker to deploy and run arbitrary code on the target machine. The ZIP file could contain any type of payload, ranging from benign scripts to malicious executables. This is useful for deploying complex payloads or multi-stage attacks.

Tree_of_Knowledge

Mechanism: This payload generates a directory tree of the target system's file structure and saves it to a file or sends it to a remote server. It uses commands to recursively list directories and files, capturing detailed information about the file system.

Effect: Provides the attacker with a comprehensive overview of the target's file system, revealing the structure and contents of directories. This can help in identifying valuable files, potential entry points, or hidden data. It is a useful reconnaissance tool to map out the target environment.

x-frame-options_scanner

Mechanism: This payload scans the target system or network for the presence of "X-Frame-Options" headers in web applications. It uses HTTP requests to check for this security header, which is used to prevent clickjacking attacks.

Effect: Identifies web applications that are potentially vulnerable to clickjacking due to the absence or misconfiguration of the "X-Frame-Options" header. This information is valuable for assessing the security posture of web applications and identifying targets for further exploitation.

Remote Access Folder:

EnableSSH-Android

Mechanism: This payload enables SSH on an Android device. It typically requires root access on the device to modify the SSH configuration and start the SSH service.

Effect: By enabling SSH, an attacker can remotely access and control the Android device over a secure shell. This allows for a wide range of remote administrative tasks, from file transfer to command execution.

Hidden_access

Mechanism: This payload creates a hidden user account on a Windows system. It uses command line instructions to add a new user and hides it from the login screen and user account lists.

Effect: The hidden user account can be used by the attacker to access the system without detection. It provides persistent and stealthy access, making it harder for users and administrators to notice unauthorized access.

NSHELL

Mechanism: This payload initiates a reverse shell connection from the target machine to the attacker's machine. It uses netcat or a similar tool to establish a backdoor connection.

Effect: The reverse shell provides the attacker with a command-line interface on the target machine, allowing for full remote control. This can be used for further exploitation, data exfiltration, or maintaining access.

Netcat-Reverseshell-On-Log-In

Mechanism: This payload sets up a reverse shell that triggers when a user logs into the system. It modifies startup scripts or scheduled tasks to initiate the connection upon login.

Effect: Ensures that the attacker gains remote access whenever a user logs in. This persistence mechanism makes it difficult to remove the backdoor without thorough system inspection.

PingZhellDucky

Mechanism: This payload sets up a reverse shell using the ICMP (ping) protocol to send commands and receive responses. It leverages the fact that ICMP traffic is often overlooked by security systems.

Effect: Provides a covert communication channel for the attacker. Since ICMP is not typically monitored as closely as other protocols, this method can bypass firewalls and intrusion detection systems.

RegDoor

Mechanism: This payload creates a backdoor by modifying the Windows registry to execute a reverse shell on startup or at specific intervals.

Effect: Establishes a persistent backdoor that survives reboots and provides the attacker with consistent remote access to the target machine. Registry modifications can be challenging to detect without thorough inspection.

ReverseDucky

Mechanism: This payload sets up a basic reverse shell on the target machine. It uses a script or command sequence to connect back to the attacker's machine.

Effect: Provides the attacker with remote command-line access to the target system. This simple yet effective method can be used for various malicious activities.

VillainShellviaNGROKTunnel

Mechanism: This payload sets up a reverse shell using an NGROK tunnel. NGROK is a service that creates secure tunnels to localhost, allowing remote access over the internet.

Effect: The NGROK tunnel makes it easy to bypass NAT and firewall restrictions, providing the attacker with remote access to the target system from anywhere on the internet. This is particularly useful for accessing systems behind restrictive network configurations.