CTF HOW TO

[**HARDWARE 1**](#_b0wnxm9iel0z)

[● ESP32 chip board 1](#_gd1n052ft6bo)

[● Memory 2](#_tpdp0snpvb5p)

[**WEB 2**](#_683quh40y1g)

[● Curl 2](#_80d0cz5pommp)

[● Brup Suite 2](#_ker4ay7frrba)

[● ffluf (url) 2](#_lsleass10k2w)

[**Forensics 2**](#_yvlapl2xpvpv)

[**PWN 3**](#_vundzhk6o3i4)

[**STEGANOGRAPHY 3**](#_l1vky5rsmp0p)

[● IMAGES 3](#_iscf3wciq8z)

[**CAR 3**](#_mkora5fddjde)

[**Helpful notes 3**](#_5lqafp5zz7su)

[**Helpful links 3**](#_miy1fz9ol6dd)

[**CTF SPECIFIC 3**](#_w7wfns2eip6g)

[● HTH 3](#_wi6p9ufkccp6)

[● SHMOOCON steganography 4](#_j84ac9lrpsjh)

[○ Layered QR code / QR code reconstruction 4](#_x4ttwurd9cyf)

[**○ STL file 4**](#_93syornr1e2v)

[○ AWS 5](#_b3n45oxfdw06)

[○ PCAP 5](#_bspqpn28vl18)

[○ DOCKER 6](#_ih97usxs0bkm)

[● HACK-A-SAT 6](#_f7v1o9ac9vg7)

[● THOTCON 6](#_bfriogaahm40)

[○ crypto 6](#_m2637fcqz4g9)

[○ RF 7](#_b3zoo9b07rtd)

[○ Badge 7](#_prc2fue358yx)

[● BATTELLE CYBER CHALLENGES https://solvers.battelle.org/cyber-challenge/cyber-challenge 7](#_az7nsbhqbllk)

[○ REVERSE ENGINEERING 7](#_2rn7jm7wozao)

[○ BINARY EXPLOITATION 8](#_i4m5dkxdw16z)

[○ FORENSICS / CRYPTO 8](#_31opvjgswrga)

[○ MULTI-ARCHITECTURE SHELLCODE 9](#_tlil2u5vh5a8)

# HARDWARE

## ESP32 chip board

* + How to connect to a esp32 chip board connect through UART.
    - serial connection through the usb port (micro usb / type c) if it has a FTDI chip (usb to serial). FTDI chip is usually connected to the esp32
    - Pin out esp32 chip, connect to Rx, Tx, and ground or io, gpio, and ground. Use uart to usb cable
  + After plugged in, in Terminal run either
    - Picocom
      * example: picocom /dev/ttyUSB0 -b 115200
      * example: picocom /dev/ttyUSB0 -b 9600
        + picocom (port in use) (baud rate ####)
    - Esptool
    - Putty
    - Screen
    - Arduino IDE
    - minicom
  + Helpful notes
    - Common baud rates: 9600, 1200, 2400, 4800, 19200, 38400, 57600, and 115200 bpds
    - To find ports in use run in terminal lsusb or ls /dev/tty\*

## Memory

* + command to dump 4MB of flash to a file: python -m esptool --port COM5 -b 115200 read\_flash 0 0x400000 flashdump.bin
* python3 -m esptool --port /dev/ttyUSB0 -b 9600 read\_flash 0 0x400000 flashdump.bin
  + - Esptool (port used) (bod rate) (command read flash from memory location ### to memory location ###) (name of file to create and put data in)
  + binwalk
  + To view strings use in terminal command \*strings\* with your .bin file strings flashdump.bin
  + Ghidra
  + GHex

# WEB

## Curl

* + Web request

## Brup Suite

## ffluf (url)

# RF

## GQRX

## Universal Radio

## Apps to use rom ZeroChaos for fox hunt and ect

* + nRF Connect
  + Flipper
  + AirGuard
  + Aruba Utilities
  + Wifi Analyzer
  + WiFiman
  + Meshtastic
  + goTenna
  + Zello

# Forensics

# PWN

# STEGANOGRAPHY

## IMAGES



# CAR

* <https://github.com/LittleBlondeDevil/TruckDevil>
  + > git clone <https://github.com/LittleBlondeDevil/TruckDevil.git>
* <https://github.com/iDoka/awesome-canbus>

# Helpful notes and tools

* Audacity
* Wireshark
* Ardunio IDE
* Ghidra
* Burp Suite
* Putty
* GHex

# Helpful links

* <https://medium.com/quiknapp/fuzz-faster-with-ffuf-c18c031fc480>
* <https://cantreally.cyou/>
* <https://www.dcode.fr/>
* <https://gchq.github.io/CyberChef/>
* <https://python-can.readthedocs.io/en/stable/>
* <https://github.com/iDoka/awesome-canbus>

# Setup

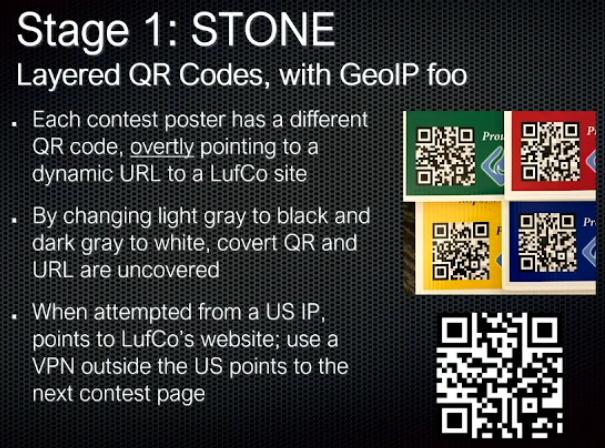
* sudo apt install python3
* Pwntools
  + sudo apt-get install python3 python3-pip python3-dev git libssl-dev libffi-dev build-essential
  + python3 -m pip install --upgrade pip
  + python3 -m pip install --upgrade pwntools
* CANBus
  + sudo apt update
  + sudo apt -y install can-utils
  + pip install python-can
* Visual Studio Code
  + sudo apt-get install wget gpg
  + wget -qO- https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > packages.microsoft.gpg
  + sudo install -D -o root -g root -m 644 packages.microsoft.gpg /etc/apt/keyrings/packages.microsoft.gpg
  + sudo sh -c 'echo "deb [arch=amd64,arm64,armhf signed-by=/etc/apt/keyrings/packages.microsoft.gpg] https://packages.microsoft.com/repos/code stable main" > /etc/apt/sources.list.d/vscode.list'
  + rm -f packages.microsoft.gpg
  + sudo apt install apt-transport-https
  + sudo apt update
  + sudo apt install code # or code-insiders
* Hardwear
  + Picocom
    - sudo apt update
    - sudo apt -y install picocom
  + Esptool
    - sudo pip install esptool
    - sudo apt -y install python3-pip
  + Putty
    - sudo add-apt-repository universe
    - sudo apt update
    - sudo apt install -y putty
  + Screen
    - sudo apt-get update -y
    - sudo apt-get install screen -y
  + Minicom
    - sudo apt-get install minicom -y
* Memory
  + Binwalk
    - sudo apt update
    - sudo apt -y install binwalk
  + Ghidra
    - sudo apt-get install openjdk-17-jdk
    - download Ghirda from its [official repository](https://github.com/NationalSecurityAgency/ghidra/releases) page and extract it into a directory.
    - After extracting the files, please go to the directory through the cd command.
    - chmod +x ghirdRun
    - ./ghirdaRun
  + Create a desktop entry file
    - “[Desktop Entry]
    - Version=10.0
    - Type=Application
    - Terminal=false
    - Icon=/home/artemix/ghidra/support
    - Exec=sh /home/artemix/ghidra/ghidraRun.sh
    - Name=Ghidra”
    - Replace the data in the **Icon** and **Exec** fields with the location of the Ghidra icon and the launch script in your machine.
    - Save the file as "Ghidra.desktop" in the Desktop directory.
    - Right-click on the file and set it to **Allow Launching** or fire up a terminal and use the chmod command to make it executable for all users.
    - chmod a+x Ghidra.desktop
  + Ghidra using snap
    - sudo apt update
    - sudo apt install snapd
    - sudo snap install ghidra
  + Ghex
    - sudo apt-get update
    - sudo apt-get install ghex -y
* Web
  + Burp
    - **Download:** [Burp Suite](http://portswigger.net/burp/releases/professional-community-2023-1-3?requestededition=community&requestedplatform=)
* Wireless
  + Wireshark
* JDK 17
  + Open JDK
    - sudo apt update
    - sudo apt install -y openjdk-17-jdk
    - sudo apt install -y openjdk-17-jre
  + Oracle JDK 17
    - sudo apt update
    - sudo apt install -y libc6-x32 libc6-i386
    - wget <https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.deb>
    - sudo dpkg -i jdk-17\_linux-x64\_bin.deb
    - sudo update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk-17/bin/java 1
  + java -version
* Wine

# CTF SPECIFIC

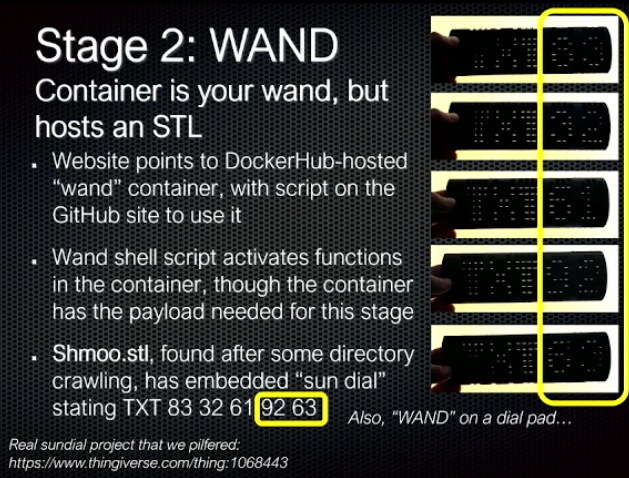
## HTH

## SHMOOCON steganography

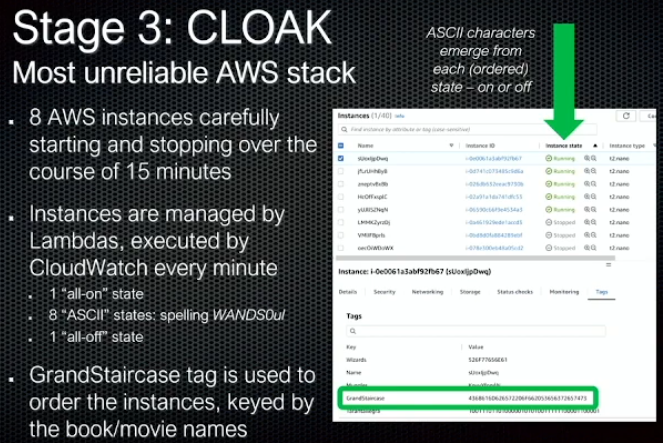
### Layered QR code / QR code reconstruction

* + - 

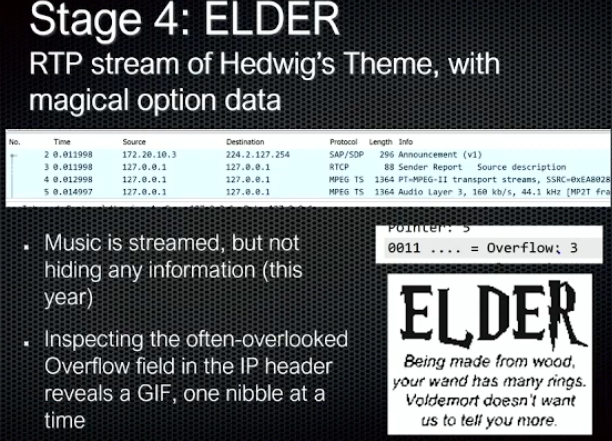
### STL file

* + - 
      * Load in blender
        + Rener light hitting it

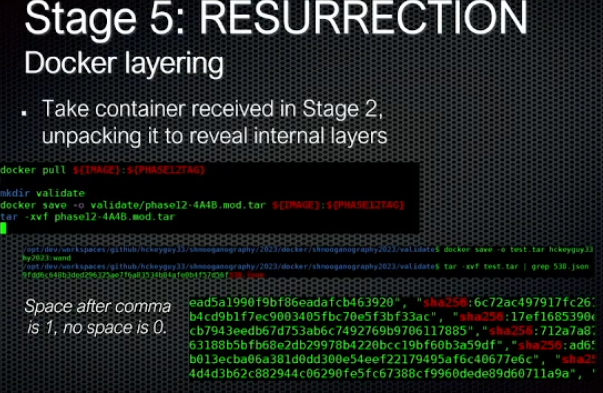
### AWS

* + - 

### PCAP

* + - 
      * <https://cantreally.cyou/posts/shmoocon-shmooganography/>

### DOCKER

* + - 

## HACK-A-SAT

## THOTCON

### crypto

* + - Find the flag in the text string: f33\_qn\_E0G
      * Hint: Rot & Decay
      * Apply ROT13
        + flag{s33\_da\_R0t}
    - Find the flag hidden in this text. Think Bacon: 01100010010\_01001310011101100\_00001400010001100\_0101000101010111
      * Its binary but in a weird format
        + flag{n0t\_k3v1n\_b4c0n\_l0lz}
    - In the hands of an allied spy, you've found a small piece of paper with a strange sequence of characters: 'RmxhZ3tiYXNlNjRfaXNfZWFzeX0='. What could it mean?
      * Base 64 encoding (give away is the “=”)
        + base64\_is\_easy
    - Ykj pynih fgl iak weru tlmt utut hor dp oyu ytd? Smisu nn ia eeco ayhc sr heme. ftuo{so1o\_t0\_so\_n0k}
    - Ftue ue FTAFOAZ 0jO agd fiqxrft kqmd. Iq tabq kag tmhq mz mymluzs fuyq mzp qzvak xqmdzuzs eayqftuzs zqi. Rxms{d0f12\_ue\_ea\_qmek}
      * ROT [A-Z]+12
        + This is THOTCON 0xC our twelfth year. We hope you have an amazing time and enjoy learning something new. Flag{r0t12\_is\_so\_easy}
    - Ace of Base TGVzcyBjcnlwdG8sIG1vcmUgZW5jb2RpbmcuIGZsYWd7MV9zYXdfdGgzX3MxZ259==
      * Base 64 encoding (give away is the “=”)
        + Less crypto, more encoding. Flag{1\_saw\_th3\_s1gn}
    - Cracking the Code A cryptic message, "nzdpmphz", has found its way to you. The circumstances surrounding the receipt of this message suggest that it holds an important secret, but it appears to be encrypted. Your task is to decrypt this mystery text, which will serve as the flag.
      * Caesar cipher with a shift of 1 to the right
        + mcology

### RF

* + - Find the Signal
      * Search for radio peak that is transmitting. This one was transmitting on 99.1
      * Morse code can be herd. Translate the morse code
        + m0rs3c0d3

### Badge

* + - esptool can be installed with pip install esptool
    - Dump memory
      * command to dump all 4MB of flash to a file: python -m esptool --port COM5 -b 115200 read\_flash 0 0x400000 flashdump.bin
    - found badge flags as strings: flag{welcome\_to\_the\_8bit\_world} and flag{lets\_fight!}
    - To view strings use in terminal command \*strings\* with your .bin file strings flashdump.bin

## BATTELLE CYBER CHALLENGES <https://solvers.battelle.org/cyber-challenge/cyber-challenge>

### REVERSE ENGINEERING

* + - Keepin' It Real:
      * Our engineers have managed to recover an old control system, but they can’t figure out how to get the thing to work! Device documentation says that it shipped with some kind of client software that is no longer available. Luckily, they were able to recover the system firmware image…
    - What the... Frob?:
      * What the... Frob?
    - Humpty Dumpy's Fall:
      * Humpty Dumpty sat on a wall.... you know the rest. Can you figure out how the kings men and horses botched the repair on poor humpty?
    - The Legend of the Headless Horseman:
      * A mysterious figure has been terrorizing the village of Sleepy Hollow. He rides a massive horse, swings a might scythe and has been collecting heads from any who draw near. A group of locals, Ichabod Crane, Katrina Van Tassel and Abraham "Brom Bones" Van Brunt have been working to discover the secret behind this mysterious menace, but just as they were on the verge of putting the pieces together, the Headless Horseman struck!
    - Feed the Magical Goat:
      * Once upon a time, there was a little reverse engineer who found a special bell. When the bell was struck, they say a magical billy goat appeared looking for food. Everyone knows billy goats will eat anything, but this is all the little reverse engineer had lying around.

### BINARY EXPLOITATION

* + - Holy Grail of ROP:
      * The ROP God has tasked King Arthur with finding the function called "holy\_grail". You must aid him on his quest! But be warned, the way is guarded by a text-based sorcerer whom loves old British comedy movies, and just to make things harder, you're going to have to find "holy\_grail" 3 times! ...Or was it 5 times? Use your pwning knowledge to answer the sorcerer's questions and ROP your way to the holy grail and bring this holy relic home for the glory of England!
    - Ghosted:
      * You're trying to save some money on your flight home so you've decided to fly Ghost Airlines, unfortunately their chatbot AI is being very unhelpful and is totally ignoring you! Can you figure out a way around the ghosting and hack your way home?

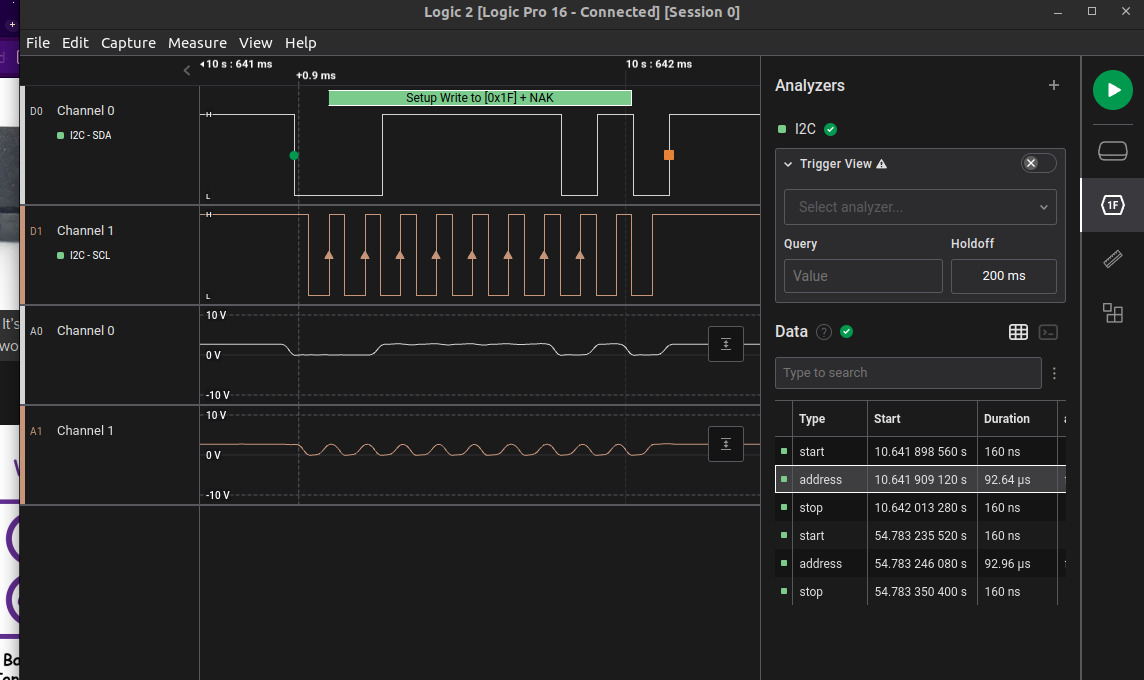
### FORENSICS / CRYPTO

* + - The Thanksgiving Bandits:
      * Last year, the notorious Thanksgiving Bandits struck for the third time. They stole over 1,000 potatoes that were meant to be mashed for the annual Thanksgiving Day feast. Two years ago, they burgled 400 pounds of cranberries, and they took 104 pumpkin pies the year before that. No one knows how they pull off such masterful heists or what they do with their score, but everyone agrees that they must be stopped at all costs.
    - Dragons and Dwarves:
      * A wise dragon decided that dwarves were too easily stealing his treasure while he slept. To thwart these villains he has placed his prized possessions inside a magic portal that transmutes the valuables into worthless junk unless one knew the magic pass phrase.

### MULTI-ARCHITECTURE SHELLCODE

* + - Unicorns Undercover:
      * You arrive at the meeting location. It's dark The handoff is scheduled to take place at 10 p.m. The unicorns arrive, but you don't see the flag. They approach you and the first one says, "push eax, call joke." The second says, "jalr blankstare." They stare at each other. They stare at you. They aren't speaking the same language.

Logic analyzer



### name type start\_time duration ack address read data

### I2C start 5.78338656 1.60000001e-07

### I2C address 5.78339712 0.00012096 true 0x1F false

### I2C data 5.78352432 0.00010976 true 0x42

### I2C data 5.78364016 0.00010992 true 0x44

### I2C data 5.78375616 0.00010976 true 0x47

### I2C data 5.783872 0.00010976 true 0x49

### I2C data 5.783988 0.00010976 true 0x49

### I2C data 5.78410384 0.00011088 true 0x4D

### I2C data 5.7842208 0.00010976 true 0x4F

### I2C data 5.7843368 0.00010976 true 0x3F

### I2C stop 5.78445792 1.6e-07

### I2C start 5.79451088 1.6e-07

### I2C address 5.79452144 0.00012128 true 0x1F true

### I2C data 5.79471984 9.568e-05 true 0x4B

### I2C data 5.79488128 9.456e-05 true 0x4B

### I2C data 5.79504176 9.232e-05 true 0x55

### I2C data 5.79519984 9.248e-05 true 0x78

### I2C data 5.79535824 9.44e-05 true 0x36

### I2C data 5.79551888 9.2e-05 true 0x76

### I2C data 5.79567728 9.392e-05 true 0x73

### I2C data 5.7958384 9.152e-05 false 0x30

### I2C stop 5.79600112 1.6e-07

### I2C start 15.2575363 1.60000001e-07

### I2C stop 15.2575906 1.6e-07

### I2C start 15.2589378 1.59999998e-07

### I2C stop 15.2707021 1.60000001e-07

### I2C start 15.2720619 1.6e-07

### I2C stop 15.2733035 1.59999998e-07

### I2C start 15.2734474 1.6e-07

### I2C stop 15.2751781 1.60000001e-07

### 

### 