TWELVESEC

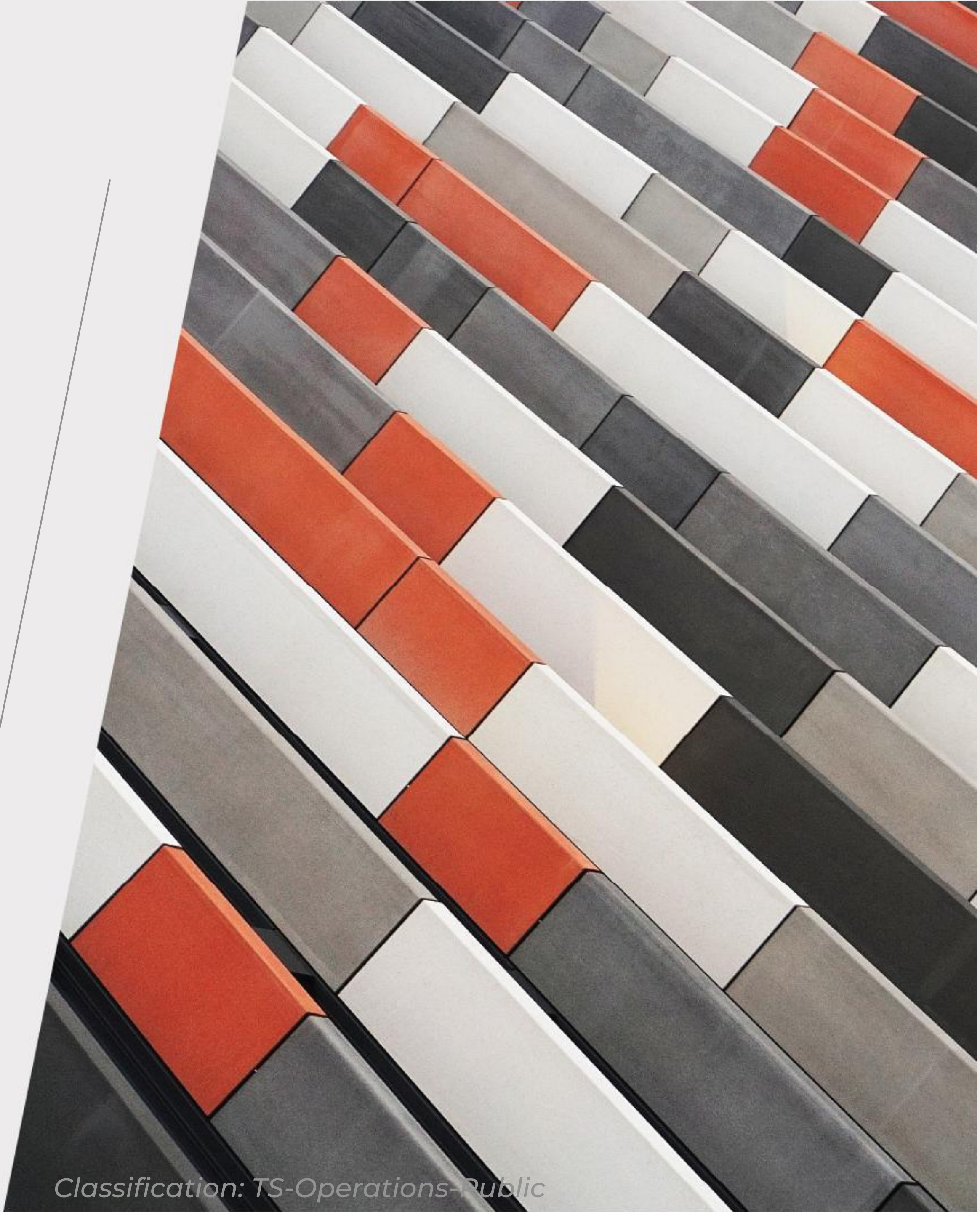
Your Partner in Cyber Security

Soldering & Firmware Extraction Workshop

Security BSides Athens 2025

www.twelvesec.com

hello@twelvesec.com



Soldering Workshop

What We Need ?

- Soldering Iron
- Soldering Wire
- Flux
- Protective Glasses

PATIENCE

How To Solder Though Hole Technology (THT) Components

⚠️ **Always Solder When The Device Is Powered Off**

Firmware Extraction Workshop

Some Basics

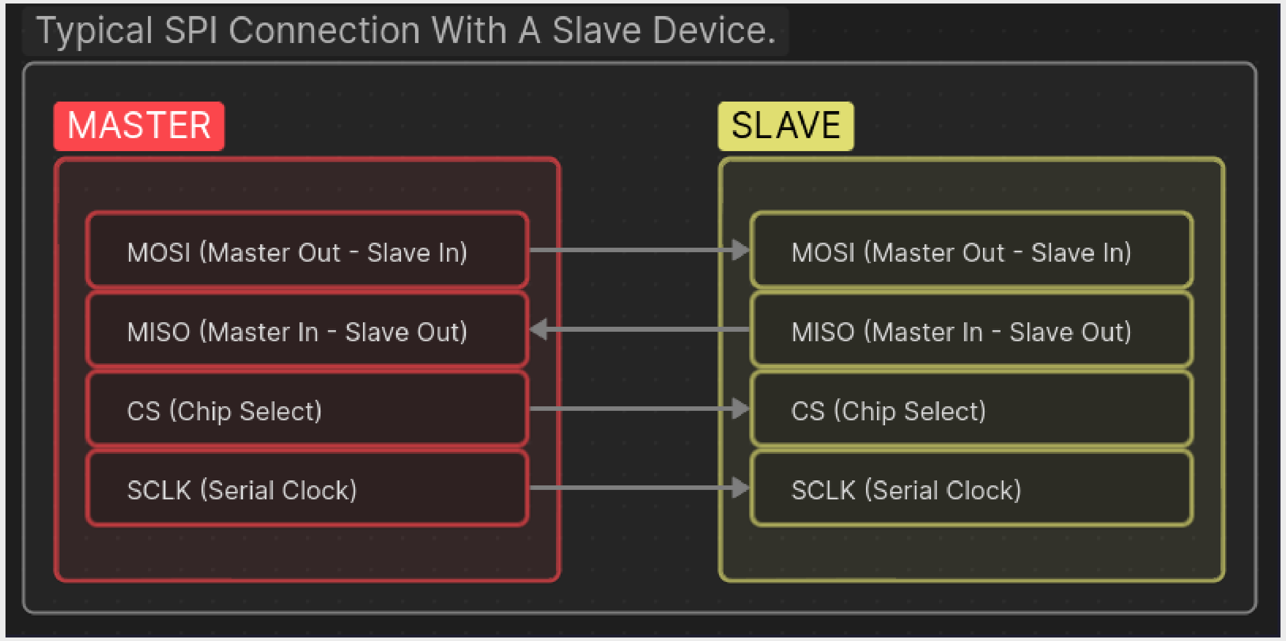
- What Is a Microcontroller
- Why To Extract The Firmware

What Is ISP

- In-System Programmers (ISPs)
- Specialized tools for programming or reprogramming internal memory (firmware) without removing the chip
- Enable developers to modify, update, or repair software on devices
- Firmware extraction:
    - Reads device memory and saves to a file
    - Used for analysis, backups, and modifications
- Arduino Nano as an ISP:
    - Uses ArduinoISP sketch
    - Communicates with ATMEGA168 chip via SPI interface

What Is SPI

- **Serial Peripheral Interface (SPI)**
- Informal standard for synchronous serial communication in embedded systems
- Operates on a master–slave architecture:
  - Master controls communication with one or more slave devices
  - Manages clock and chip select signals
- Introduced by Motorola in the early 1980s
- Utilizes a four-wire serial bus for full-duplex communication:
  - Distinct from three-wire half-duplex and two-wire systems (I^2C, 1-Wire)
- Common applications:
  - Connecting microcontrollers to SD cards, LCDs, ADC/DAC converters, flash memory, and communication modules


Typical SPI Connection With A Slave Device.
MASTER
SLAVE
MOSI (Master Out - Slave In)
MISO (Master In - Slave Out)
CS (Chip Select)
SCLK (Serial Clock)
MOSI (Master Out - Slave In)
MISO (Master In - Slave Out)
CS (Chip Select)
SCLK (Serial Clock)

How To Create Our Own ISP Programmer

- Start Arduino IDE:

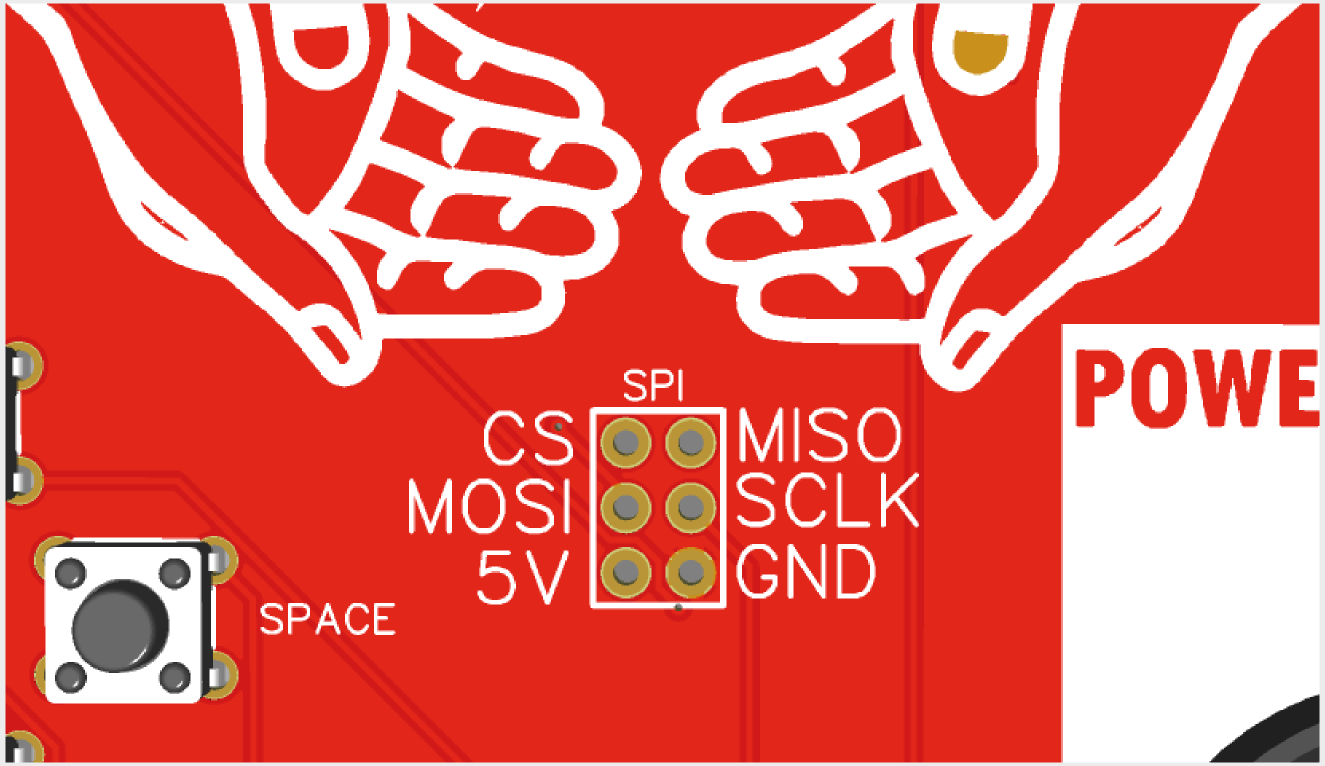
```
./arduino-ide_2.3.4_Linux_64bit.AppImage --no-sandbox
```

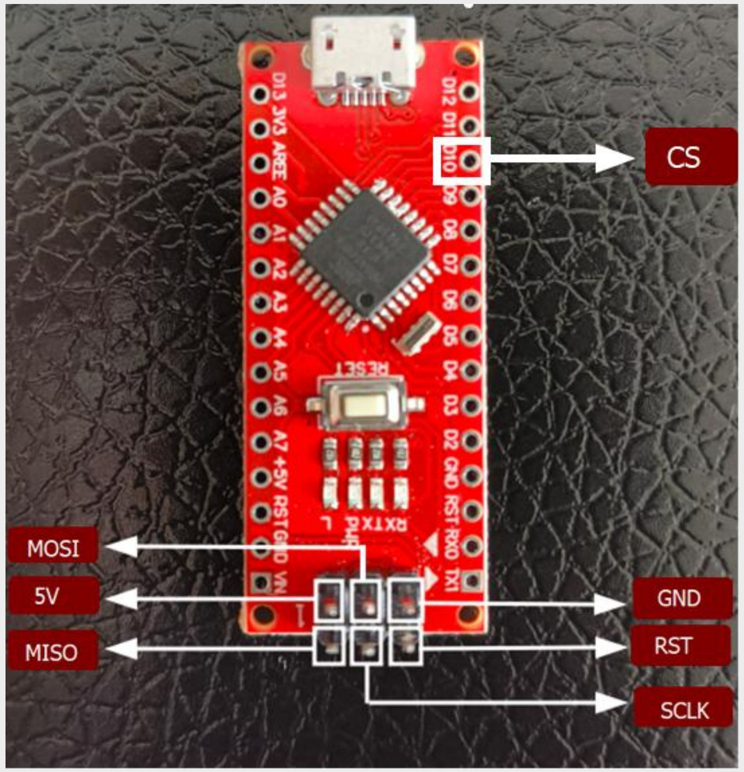
- Connect Arduino Nano to your laptop
- Open ArduinoISP sketch
- Select the correct Board and PORT:
- Set Processor to Atmega168:
- Upload firmware

How To Extract The Firmware From The Internal Flash Memory

<u>DO NOT CONNECTION JUMPER WIRES WHEN POWER IS ON</u>

Objective: Connect ArduinoISP with the SPI Interface of the Badge.





CONNECT THE ISP's CS (D10) PIN TO THE RESET PIN OFF THE TARGET

How To Extract The Firmware From The Internal Flash Memory

Objective: Use AVRDUDE to dump the internal flash memory of the ATmega168 microcontroller.

1. Dump Flash Memory in Binary Format:

```
avrdude "-C/etc/avrdude.conf" -v -V -patmega168p -carduino -P/dev/ttyUSB1 -b19200 -D "-Uflash:r:firmware.bin:r"
```

2. Dump Flash Memory in HEX Format:

```
avrdude "-C/etc/avrdude.conf" -v -V -patmega168p -carduino -P/dev/ttyUSB1 -b19200 -D "-Uflash:r:firmware.hex:i"
```

3. Convert HEX to Binary (if required):

```
objcopy --input-target=ihex --output-target=binary firmware.hex firmware.bin
```

4. Verify the extracted firmware using strings:

```
strings firmware.bin > strings_output.txt
```

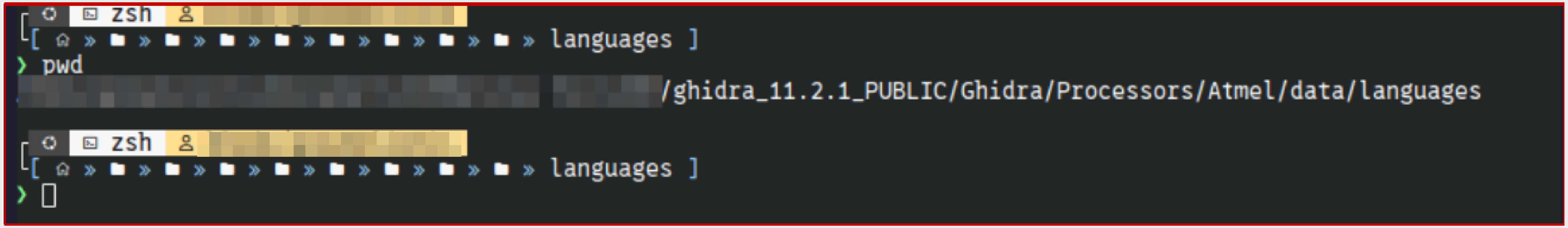
Basic Firmware Analysis With Ghidra

Objective: Analyze the extracted firmware using Ghidra to locate hardcoded secrets.

Prepare Ghidra for AVR Reverse Engineering

Download the helper file:

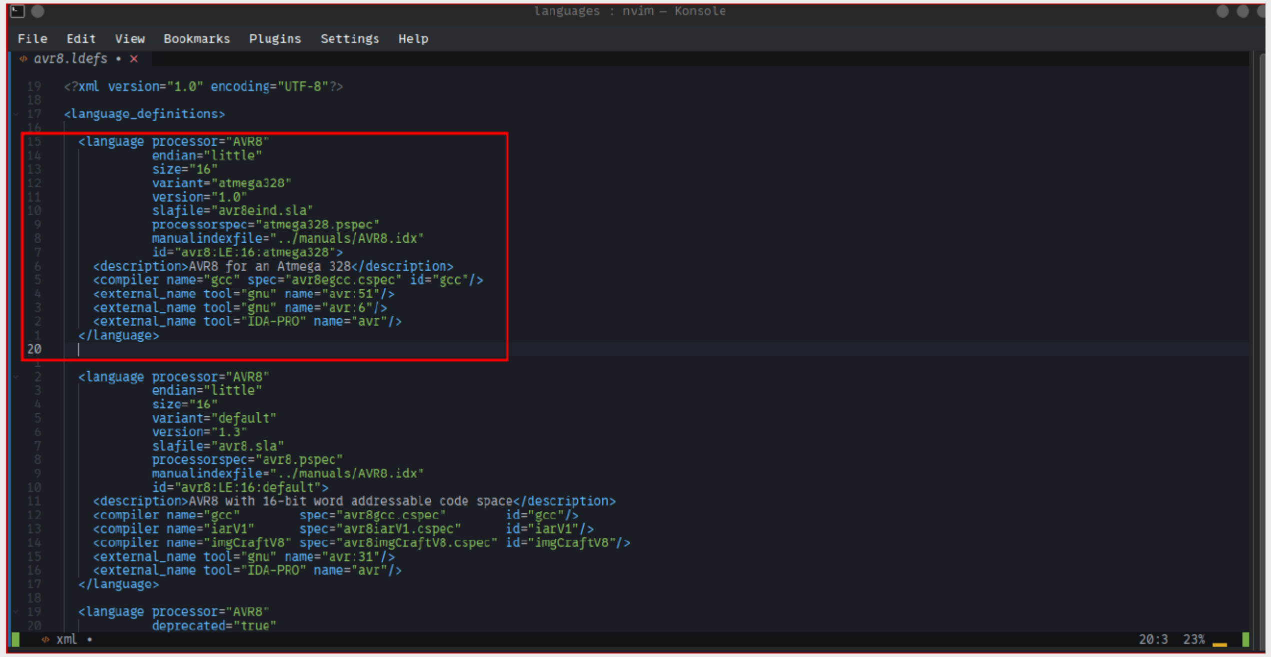
```
wget https://raw.githubusercontent.com/ahroach/avr_ghidra_helpers/a98c18b2ec627a6a2e16df360f98ce59a00eb187/atmega328.pspec
```



```
[ zsh [redacted]
[ » » » » » » » » » languages ]
❯ pwd
[redacted]/ghidra_11.2.1_PUBLIC/Ghidra/Processors/Atmel/data/languages

[ zsh [redacted]
[ » » » » » » » » » languages ]
❯ 
```

Add the following <language> tag to the file Ghidra/Processors/Atmel/data/languages/avr8.ldefs

```
<language processor="AVR8"
  endian="little"
  size="16"
  variant="atmega328"
  version="1.0"
  slafile="avr8eind.sla"
  processorspec="atmega328.pspec"
  manualindexfile="../manuals/AVR8.idx"
  id="avr8:LE:16:atmega328">
 <description>AVR8 for an Atmega 328</description>
 <compiler name="gcc" spec="avr8egcc.cspec" id="gcc"/>
 <external_name tool="gnu" name="avr:51"/>
 <external_name tool="gnu" name="avr:6"/>
 <external_name tool="IDA-PRO" name="avr"/>
</language>
```


languages : nvim — Konsole
File Edit View Bookmarks Plugins Settings Help
avr8.ldefs
xml
20:3 23%

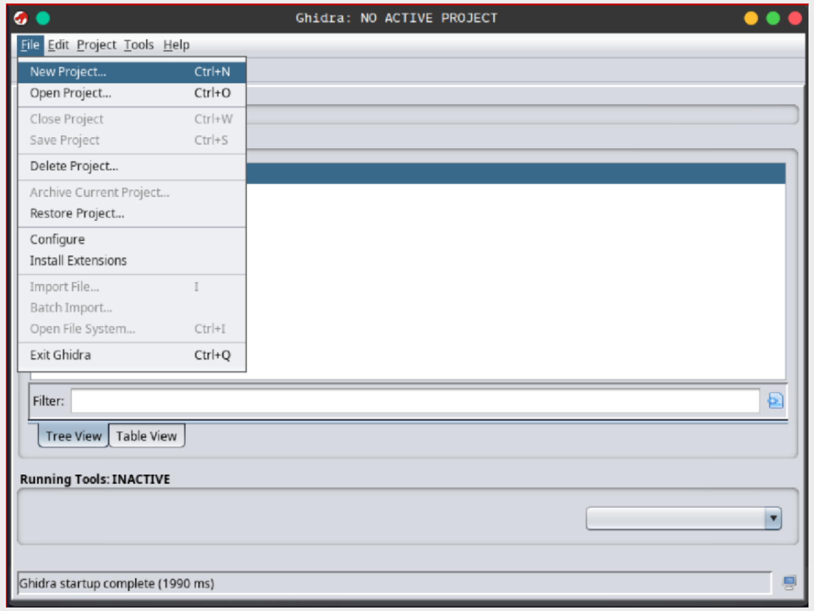
```
<?xml version="1.0" encoding="UTF-8"?>

<language_definitions>

  <language processor="AVR8"
            endian="little"
            size="16"
            variant="atmega328"
            version="1.0"
            slafile="avr8eind.sla"
            processorspec="atmega328.pspec"
            manualindexfile="../manuals/AVR8.idx"
            id="avr8:LE:16:atmega328">
    <description>AVR8 for an Atmega 328</description>
    <compiler name="gcc" spec="avr8egcc.cspec" id="gcc"/>
    <external_name tool="gnu" name="avr:51"/>
    <external_name tool="gnu" name="avr:6"/>
    <external_name tool="IDA-PRO" name="avr"/>
  </language>


  <language processor="AVR8"
            endian="little"
            size="16"
            variant="default"
            version="1.3"
            slafile="avr8.sla"
            processorspec="avr8.pspec"
            manualindexfile="../manuals/AVR8.idx"
            id="avr8:LE:16:default">
    <description>AVR8 with 16-bit word addressable code space</description>
    <compiler name="gcc"        spec="avr8gcc.cspec"        id="gcc"/>
    <compiler name="iarV1"      spec="avr8iarV1.cspec"      id="iarV1"/>
    <compiler name="imgCraftV8" spec="avr8imgCraftV8.cspec" id="imgCraftV8"/>
    <external_name tool="gnu" name="avr:31"/>
    <external_name tool="IDA-PRO" name="avr"/>
  </language>

  <language processor="AVR8"
            deprecated="true"
```

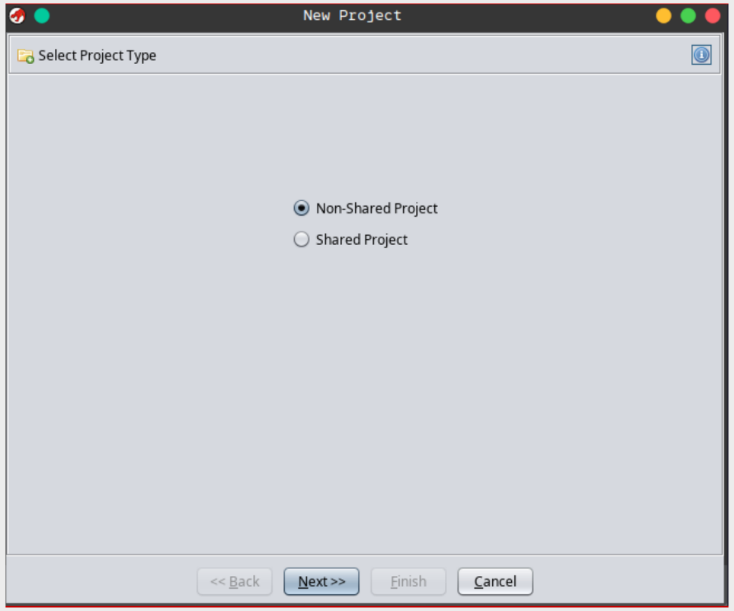
TWELVESEC Your Partner in Cyber Security

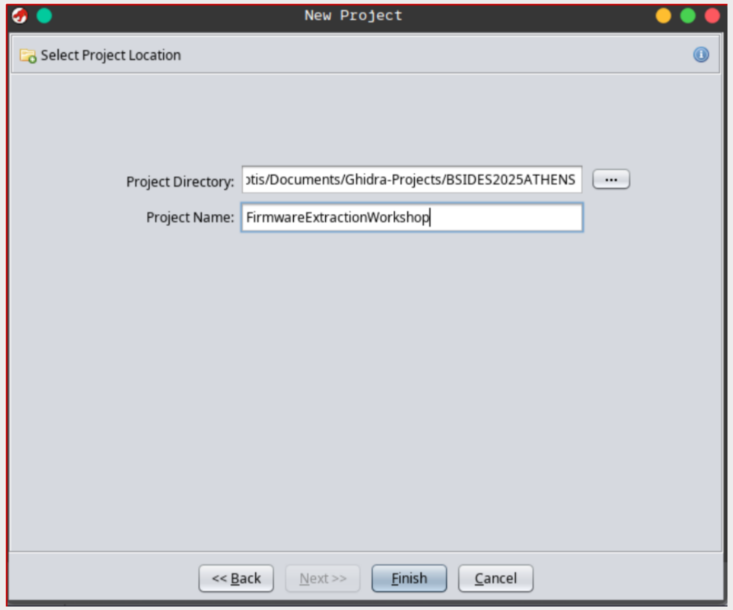
Analyze Firmware in Ghidra

1. Import the binary (firmware.bin) into Ghidra.
2. Select atmega328 as the processor.
3. Search for hardcoded strings using Ghidra's search capabilities.

Creating a new project in Ghidra.



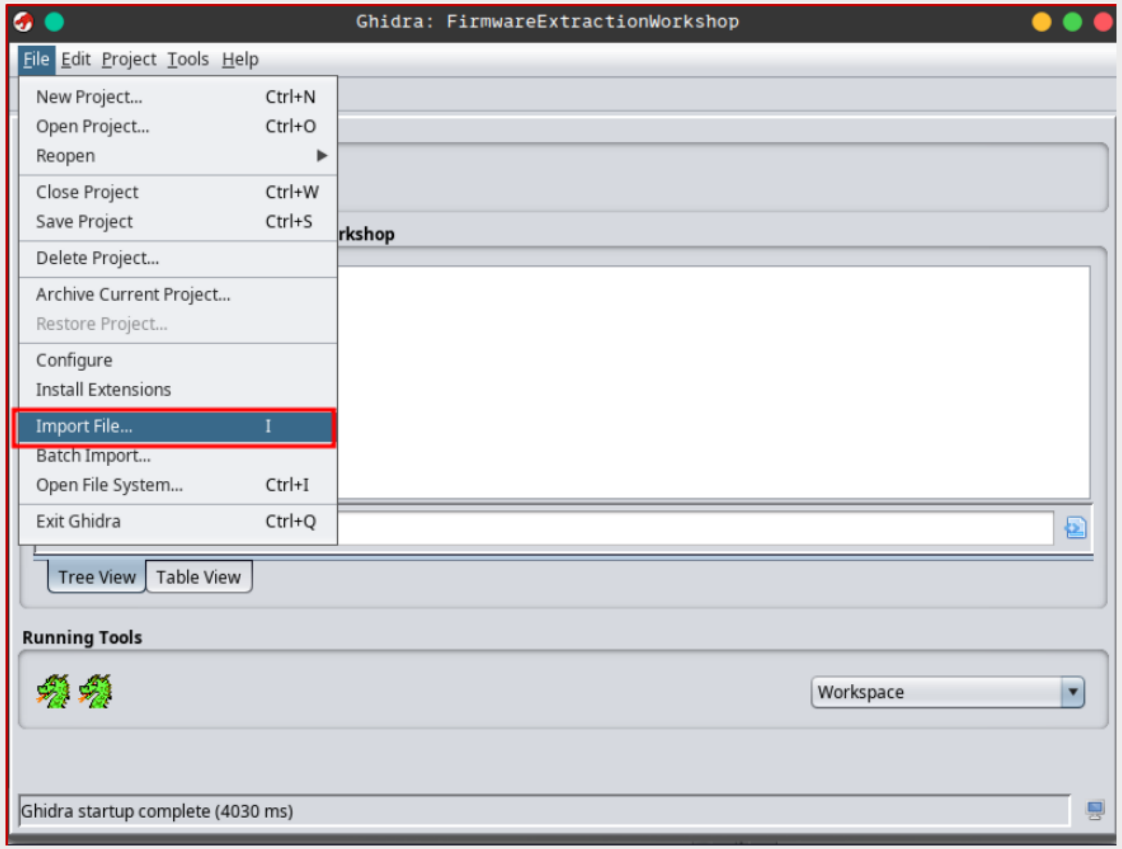




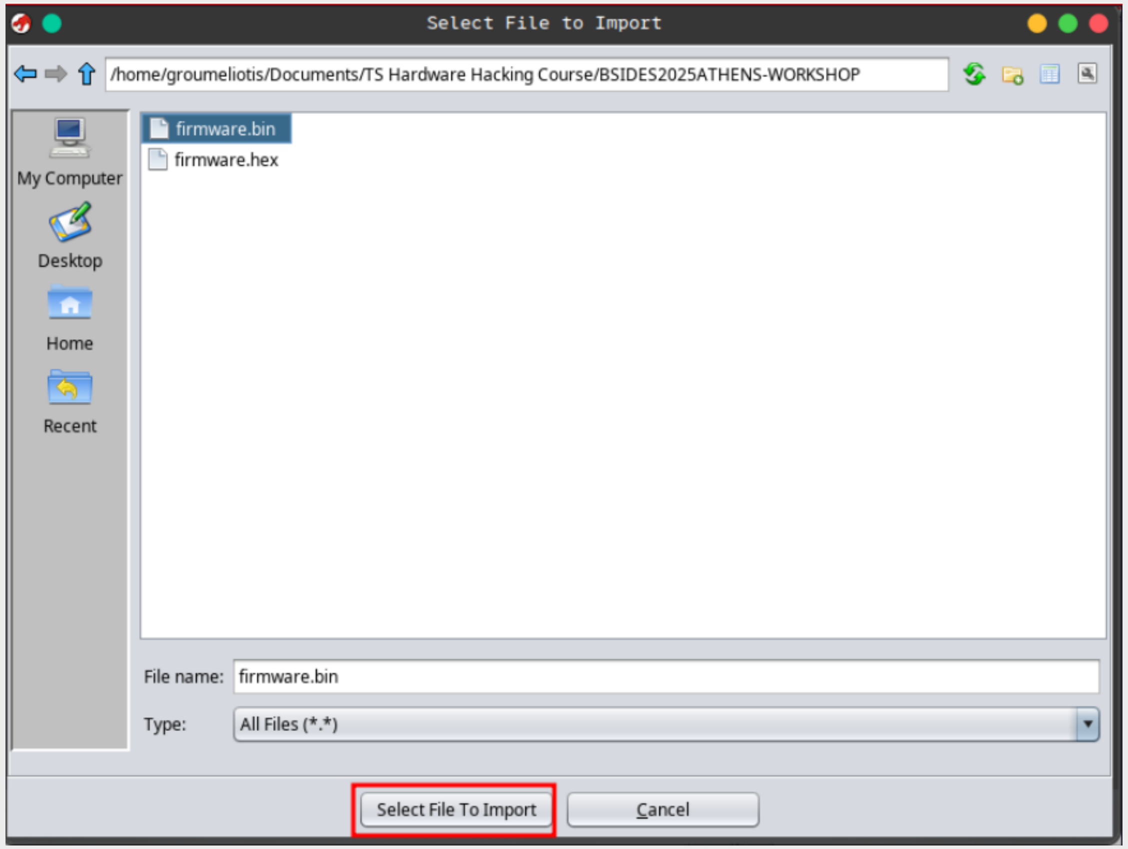
Project Directory: ɔtis/Documents/Ghidra-Projects/BSIDES2025ATHENS
Project Name: FirmwareExtractionWorkshop

TWELVESEC | Your Partner in Cyber Security

Importing The Firmware



Ghidra startup complete (4030 ms)

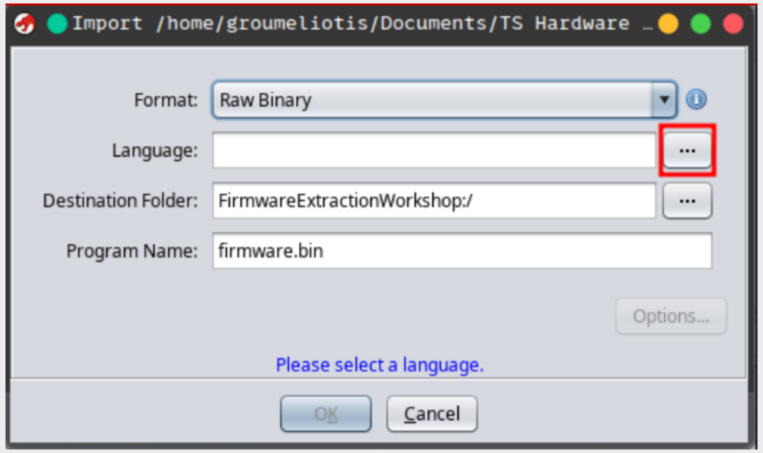


/home/groumeliotis/Documents/TS Hardware Hacking Course/BSIDES2025ATHENS-WORKSHOP

File name: firmware.bin

Type: All Files (*.*)

Analyzing the Firmware.


Import /home/groumeliotis/Documents/TS Hardware ...

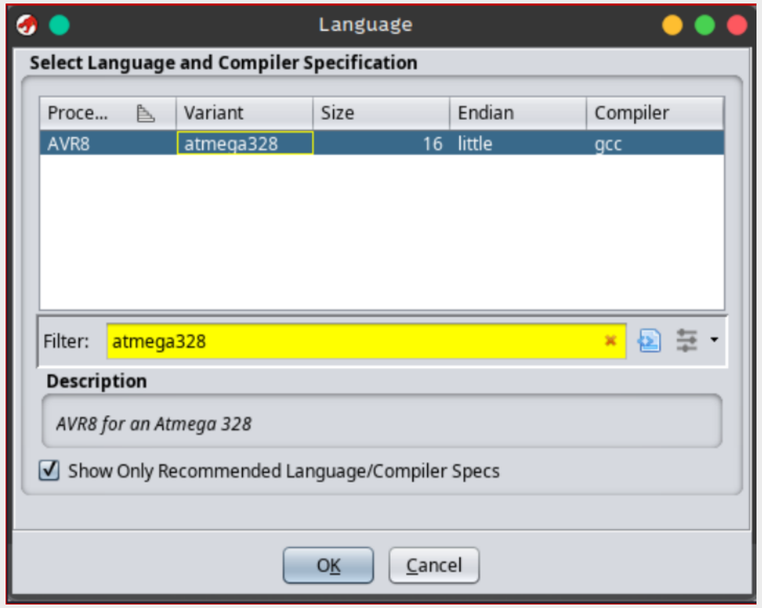
Format: Raw Binary

Language:

Destination Folder: FirmwareExtractionWorkshop:/

Program Name: firmware.bin

Please select a language.


Language

Select Language and Compiler Specification

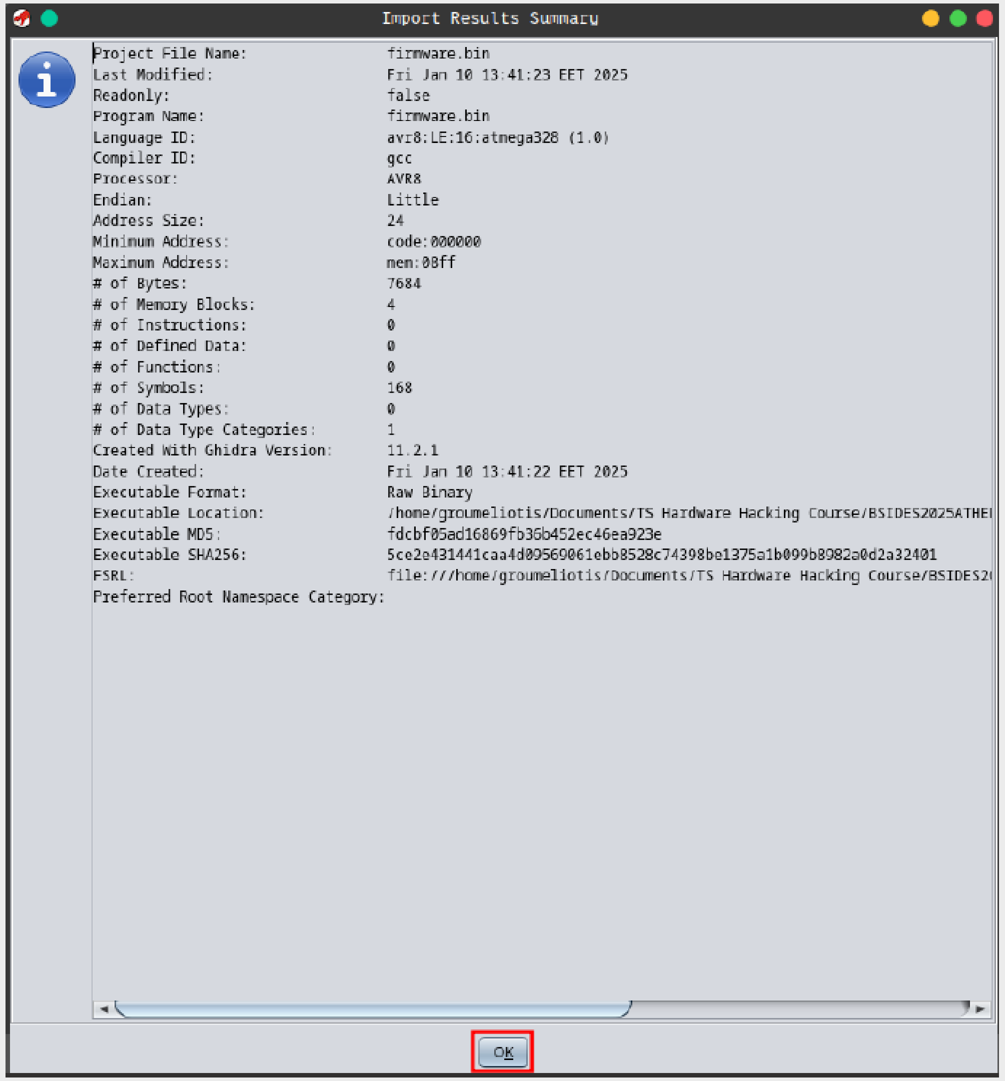
Proce...	Variant	Size	Endian	Compiler
AVR8	atmega328	16	little	gcc

Filter: atmega328

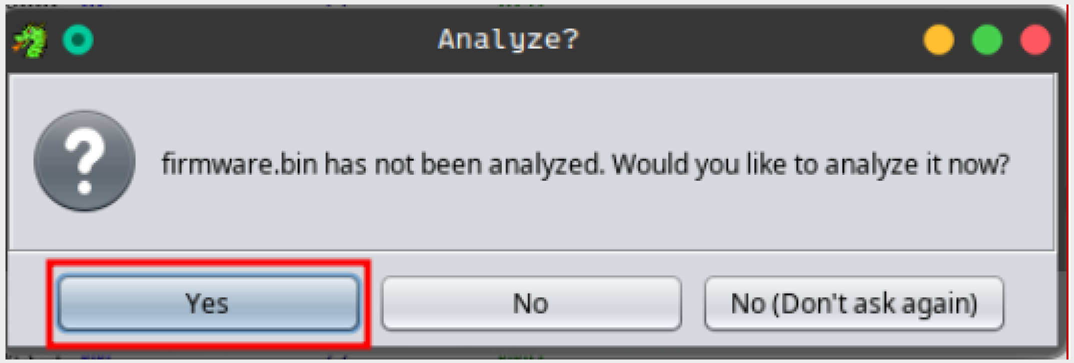
Description

AVR8 for an Atmega 328

☑ Show Only Recommended Language/Compiler Specs

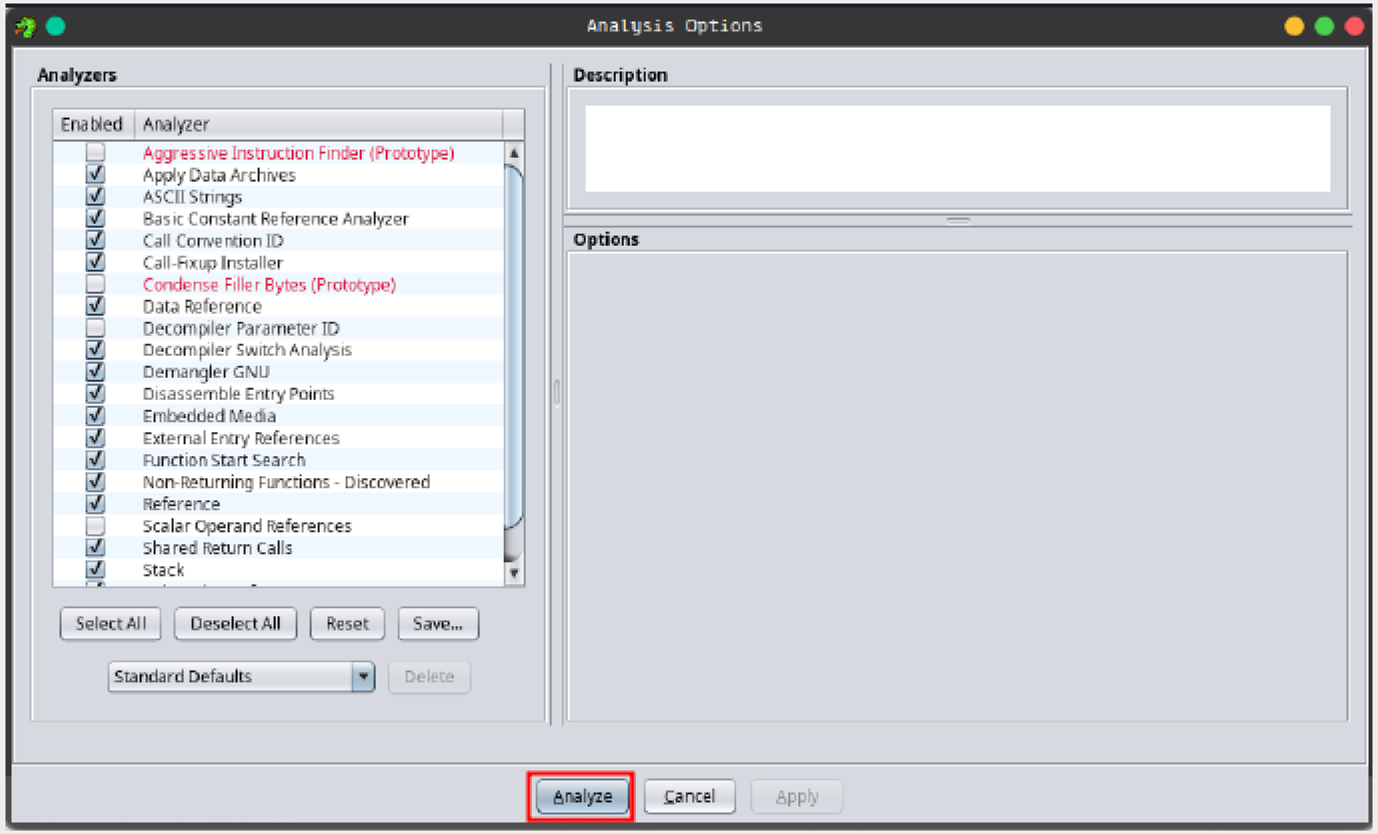

Import Results Summary

```
Project File Name:              firmware.bin
Last Modified:                  Fri Jan 10 13:41:23 EET 2025
Readonly:                       false
Program Name:                   firmware.bin
Language ID:                    avr8:LE:16:atmega328 (1.0)
Compiler ID:                    gcc
Processor:                      AVR8
Endian:                         Little
Address Size:                   24
Minimum Address:                code:000000
Maximum Address:                mem:08ff
# of Bytes:                     7684
# of Memory Blocks:             4
# of Instructions:              0
# of Defined Data:              0
# of Functions:                 0
# of Symbols:                   168
# of Data Types:                0
# of Data Type Categories:      1
Created With Ghidra Version:    11.2.1
Date Created:                   Fri Jan 10 13:41:22 EET 2025
Executable Format:              Raw Binary
Executable Location:            /home/groumeliotis/Documents/TS Hardware Hacking Course/BSIDES2025ATHE
Executable MD5:                 fdcbf05ad16869fb36b452ec46ea923e
Executable SHA256:              5ce2e431441caa4d09569061ebb8528c74398be1375a1b099b8982a0d2a32401
FSRL:                           file:///home/groumeliotis/Documents/TS Hardware Hacking Course/BSIDES2
Preferred Root Namespace Category:
```



Analyze?

firmware.bin has not been analyzed. Would you like to analyze it now?



Analysis Options

Analyzers

Enabled	Analyzer
☐	Aggressive Instruction Finder (Prototype)
✓	Apply Data Archives
✓	ASCII Strings
✓	Basic Constant Reference Analyzer
✓	Call Convention ID
✓	Call-Fixup Installer
☐	Condense Filler Bytes (Prototype)
✓	Data Reference
☐	Decompiler Parameter ID
✓	Decompiler Switch Analysis
✓	Demangler GNU
✓	Disassemble Entry Points
✓	Embedded Media
✓	External Entry References
✓	Function Start Search
✓	Non-Returning Functions - Discovered
✓	Reference
☐	Scalar Operand References
✓	Shared Return Calls
✓	Stack

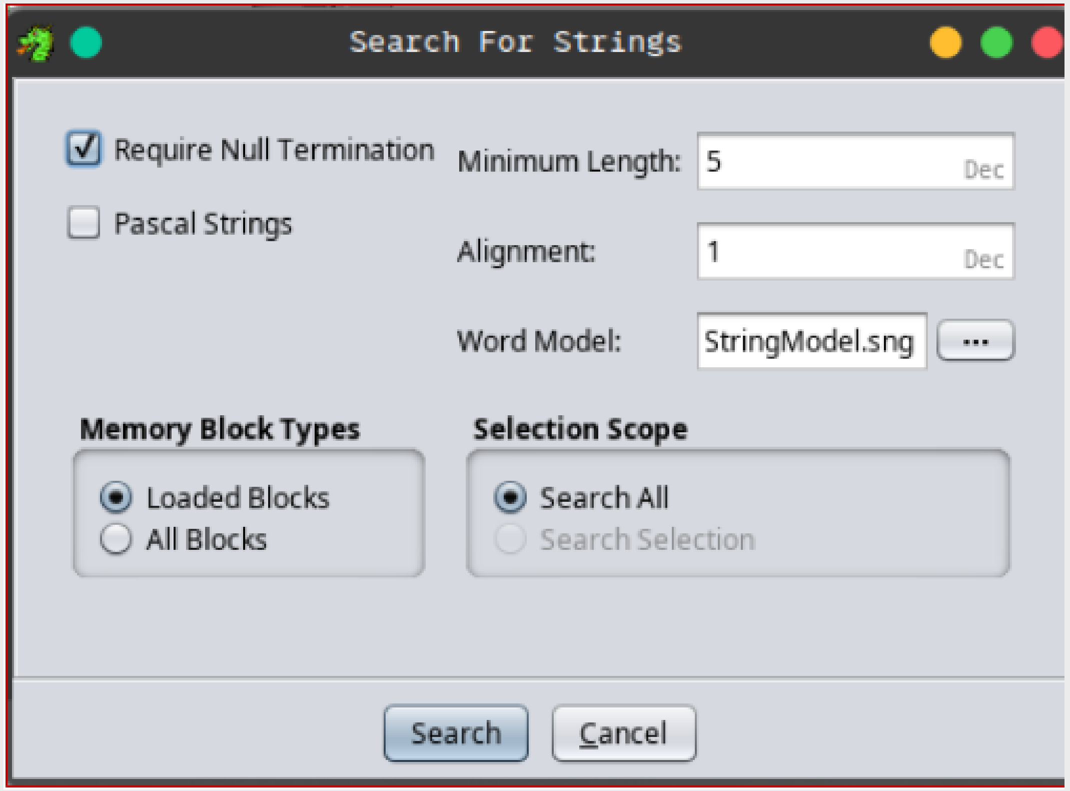
Description

Options

Standard Defaults

Search for Hardcoded strings using Ghidra's search capabilities.

Navigate to “Search > For Strings …” and Click Search.



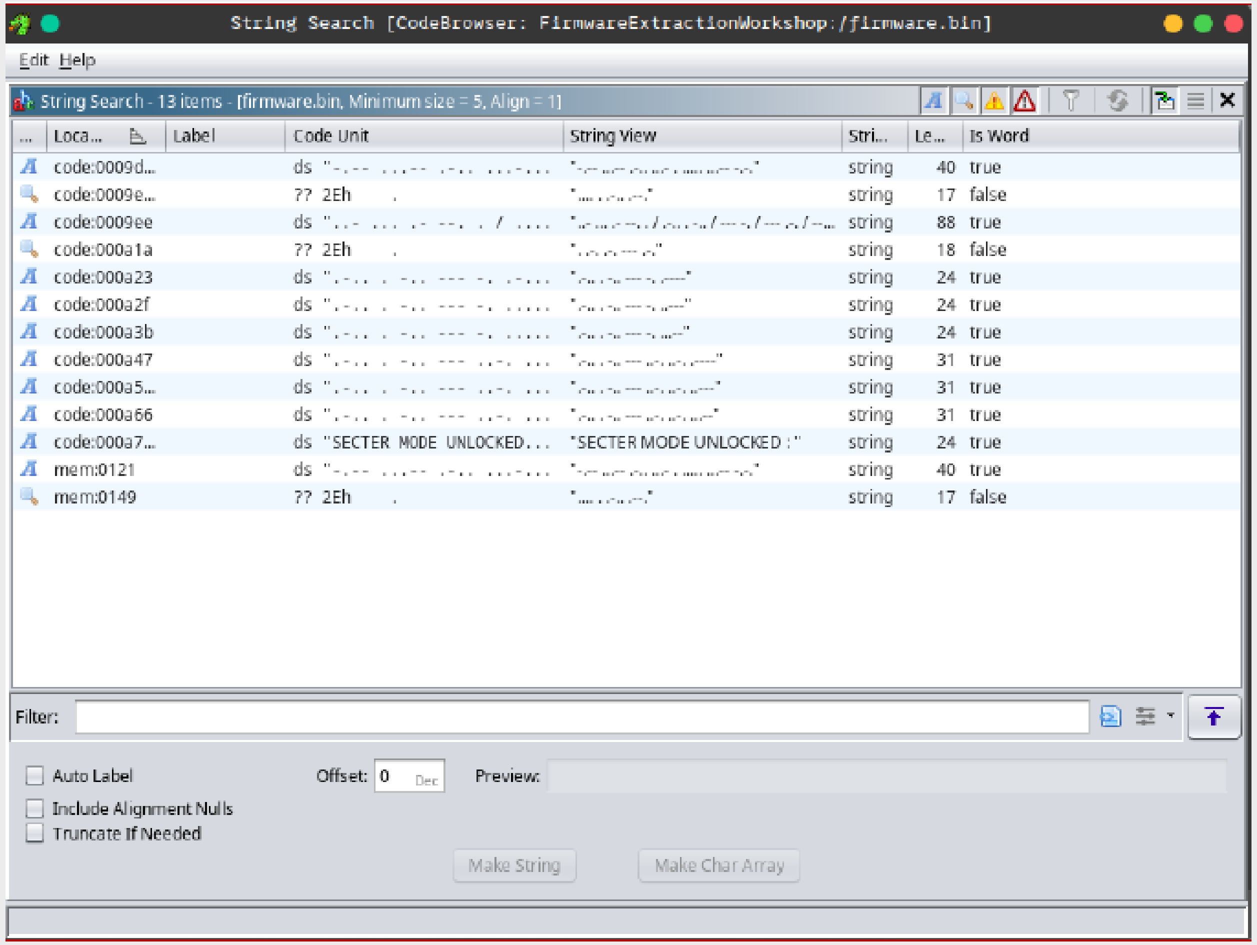
Search For Strings

[✓] Require Null Termination
[ ] Pascal Strings

Minimum Length: 5 Dec
Alignment: 1 Dec
Word Model: StringModel.sng

Memory Block Types
(●) Loaded Blocks
( ) All Blocks

Selection Scope
(●) Search All
( ) Search Selection



String Search [CodeBrowser: FirmwareExtractionWorkshop:/firmware.bin]

String Search - 13 items - [firmware.bin, Minimum size = 5, Align = 1]

...	Loca...	Label	Code Unit	String View	Stri...	Le...	Is Word
	code:0009d...		ds "-.-- ...-- .-.. ...-..."	[illegible]	string	40	true
	code:0009e...		?? 2Eh .	[illegible]	string	17	false
	code:0009ee		ds "..- ... .- --. . / ...."	[illegible]	string	88	true
	code:000a1a		?? 2Eh .	[illegible]	string	18	false
	code:000a23		ds ".-.. . -.. --- -. .-..."	[illegible]	string	24	true
	code:000a2f		ds ".-.. . -.. --- -. ....."	[illegible]	string	24	true
	code:000a3b		ds ".-.. . -.. --- -. ....."	[illegible]	string	24	true
	code:000a47		ds ".-.. . -.. --- ..-. ..."	[illegible]	string	31	true
	code:000a5...		ds ".-.. . -.. --- ..-. ..."	[illegible]	string	31	true
	code:000a66		ds ".-.. . -.. --- ..-. ..."	[illegible]	string	31	true
	code:000a7...		ds "SECTER MODE UNLOCKED..."	"SECTER MODE UNLOCKED :"	string	24	true
	mem:0121		ds "-.-- ...-- .-.. ...-..."	[illegible]	string	40	true
	mem:0149		?? 2Eh .	[illegible]	string	17	false

Filter:

[ ] Auto Label
[ ] Include Alignment Nulls
[ ] Truncate If Needed

Offset: 0 Dec

Preview:

What's Next

PwnPad

A Hardware Hacking Learning Platform

https://github.com/twelvesec/PwnPad

Questions !?

www.twelvesec.com
hello@twelvesec.com