

1)Installation steps for Ghidra:

Official Ghidra Website: Visit ghidra-sre.org and navigate to the "Download" section

Move the ghidra zip file to the /opt/ directory

```
sudo mv ghidra_11.4_PUBLIC_20250620.zip /opt/
```

Here you can unzip the file and then remove the zip version

```
cd /opt/
```

```
sudo unzip ghidra_11.4_PUBLIC_20250620.zip
```

```
sudo rm -r ghidra_11.4_PUBLIC_20250620.zip
```

Then rename the the unzipped directory to ghidra

```
sudo mv ghidra_11.4_PUBLIC/ ghidra
```

Now cd into the ghidra directory



```
cd ghidra/
```



Try running ghidra






```
sudo ./ghidraRun
```

What This Script Detects

Check	What It Means	Output
Debug Symbols	If names like <code>main</code> , <code>printf</code> are present	<code>stripped</code> or <code>not stripped</code>
High Entropy	Encrypted/compressed data	Lists suspicious memory addresses
Control Flow	Obfuscated dispatchers or switch cases	Reports complex blocks

Feature	Description
 Suspicious Function Detection	Lists functions with generic names like <code>FUN_1234</code> or <code>sub_4567</code>
 XOR Operation Scan	Finds functions that contain XOR instructions

 Ghidra Bookmarks	Adds bookmarks to high-entropy areas, unnamed functions, and XOR-heavy functions
 Output File	Saves report to the same folder as the script file

Feature	Description
 AES Constant Detection	Scans for AES S-box
 XOR Loop Scan	Flags functions using <code>xor</code> (often obfuscators/decryptors)
 JSON Report	Machine-readable structured output
 CSV Report	Easy to open in Excel/sheets
 Ghidra Bookmarks	Add visual markers inside the disassembly

To run the Python (Jython) script in Ghidra, follow these simple steps:

✓ Prerequisites

Make sure you've already opened a project and imported a firmware binary in Ghidra.

Ghidra uses Jython, so the script is written in Python 2-style syntax and runs in Ghidra's Script Manager.

Steps to Run the Script

1. Open Ghidra and load your firmware project
Start Ghidra.

Open your project.

Import the firmware binary and analyze it (you can accept default options for most cases).

2. Open Script Manager
Go to Window → Script Manager
(or press Shift + F3)

3. Create a New Python Script
In the Script Manager, click File → New...

Choose Python as the language.

Name it something like `firmware_analysis.py`.

Click OK — this will open the script in Ghidra's built-in editor.

4. Paste the Script Code

Delete any default code.

Paste the full script I gave you (or your version with enhancements).

Save the file (Ctrl+S or click the disk icon).

5. Run the Script

In the Script Manager, find your script in the list.

Click it and press the green play/run button (triangle).

The script will:

Analyze symbols

Search for XOR and crypto patterns

Identify high-entropy regions

Save a JSON and CSV report

Add bookmarks to Ghidra views

6. View the Output

Console output: Bottom panel in Ghidra will show logs.

Bookmarks: Use Window → Bookmarks to jump to flagged areas.

Output files: JSON and CSV reports are saved in the same folder as your script (check the path in the console log).

Optional: Verify Python Support is Installed

If you **don't see "Python" at all** in the language list:

1. Go to [Help](#) → [About Ghidra](#) → [Installed Extensions](#)

2. Ensure **"Python" or "Jython Scripting"** is installed

3. If not installed:

- Go to **File → Install Extensions**
- Check **"Python" or "Jython support"**
- Click **Next → Finish**
- Restart Ghidra

Step 1: Install the correct **venv package**

You need the versioned venv package that matches the Python version Ghidra uses—likely Python 3.10 on Ubuntu 24.04 or Python 3.8/3.9 on earlier versions. For instance, on Ubuntu 24.04:

```
sudo apt update
sudo apt install python3.10-venv python3-pip
```

On Ubuntu 22.04 or 20.04, replace **3.10** with your system's default:

```
sudo apt install python3.8-venv python3-pip
```

This adds the missing **ensurepip** module to the standard library.

[reddit.com+12askubuntu.com+12askubuntu.com+12](#)

Step 1: Bootstrap **pip into the venv**

Run the following to install **pip** inside the Ghidra virtual environment:

```
/home/anandemb/.config/ghidra/ghidra_11.4_PUBLIC/venv/bin/python3 -m
ensurepip --upgrade
```

This uses Python's built-in **ensurepip** to add pip into the venv [Reddit+15Stack Overflow+15GitHub+15Reddit](#).

✅ Step 2: Install PyGhidra using the venv pip

Once pip is available:

```
/home/anandemb/.config/ghidra/ghidra_11.4_PUBLIC/venv/bin/python3 -m  
pip install --no-index \  
-f /opt/ghidra/Ghidra/Features/PyGhidra/pypkg/dist pyghidra
```

This matches the typical offline installation method from the Ghidra docs .

↺ Step 3: Relaunch PyGhidra

After installation completes without errors, run:

```
/opt/ghidra/support/pyghidraRun
```

or the equivalent `pyghidraRun` to start Ghidra with Python 3 support. You should now see the **Python 3** option in Script Manager [Google Cloud+2Reddit+2GitHub+2](#).