Judging a Book By Its Plugins:

A Static-Analysis Security Evaluation of Calibre

CollinHatcher
Kevin Rodriguez

Project Overview

Scope: Provide a comprehensive security assessment of Calibre’s plugin architecture.

Goals: Identify potential plugin vulnerabilities using formal threat modeling and static analysis.

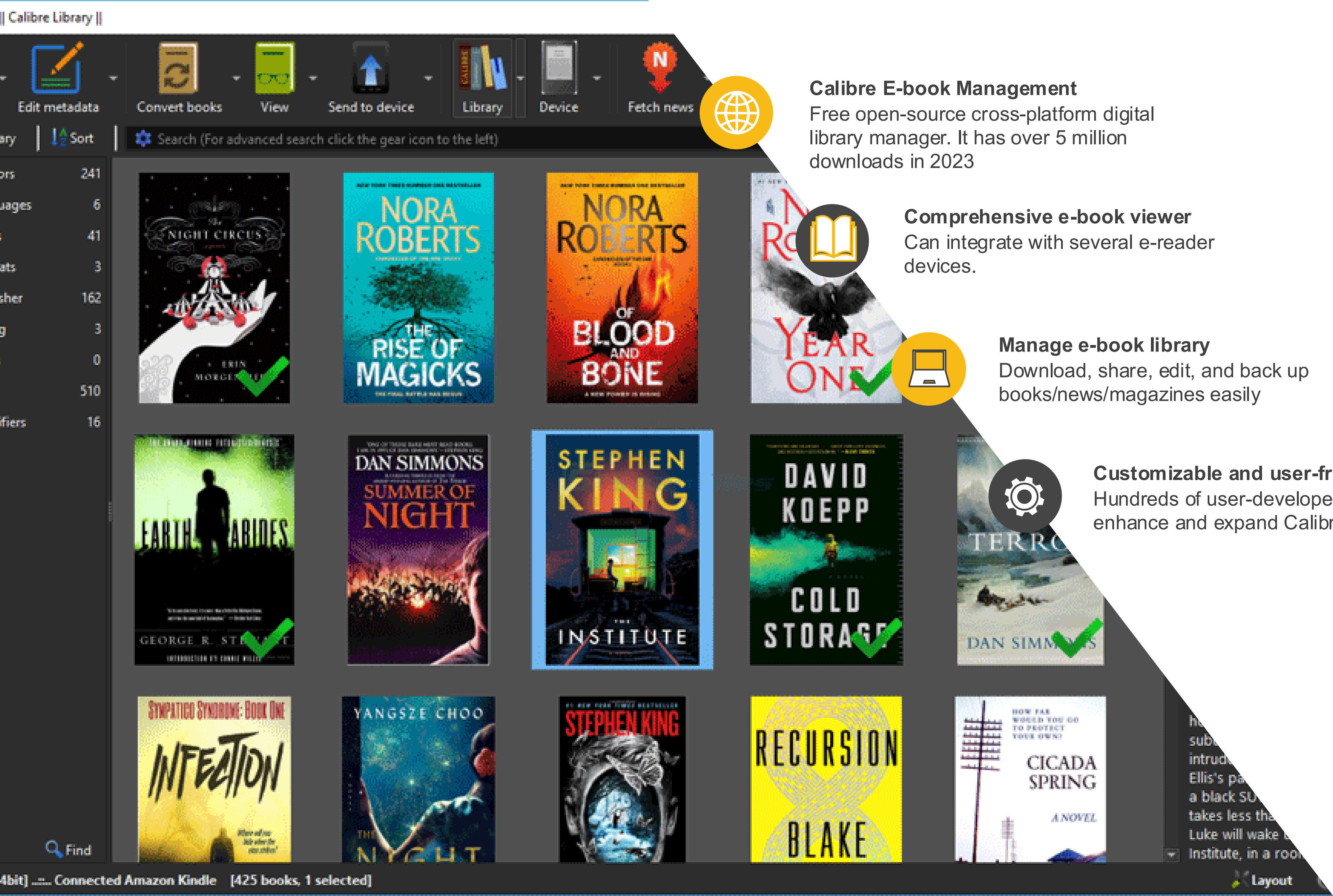
Outcome: Provide an evidence-based report and security recommendations to improve Calibre’s security and protect user data.

About Calibre

What is it: Popular cross-platform e-book manager.

Plugin System: Supports user-installed plugins to extend functionality.

Why Calibre?: Widely used tool with a large user base. The plugin architecture is a prime candidate for uncovering potential security vulnerabilities.



4bit] ..::.. Connected Amazon Kindle [425 books, 1 selected]

Calibre E-book Management
Free open-source cross-platform digital library manager. It has over 5 million downloads in 2023

Comprehensive e-book viewer
Can integrate with several e-reader devices.

Manage e-book library
Download, share, edit, and back up books/news/magazines easily

Customizable and user-friendly
Hundreds of user-developed plugins to enhance and expand Calibre's features

Threat Model & Attacker Roles

Modelling Approach

- ✓ Applied STRIDE to identify threat categories.
- ✓ Applied DREAD to score and prioritize each threat.

Spoofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of Service, **E**levation of Privilege

S
T
R
I
D
E
D
R
E
A
D

Damage, **R**eproducibility, **E**xplotability, **A**ffected Users, **D**iscoverability

Malicious Plugin Authors

This attacker creates and distributes malicious plugin code containing vulnerabilities with the goal to gain unauthorized access to data or disrupt services.

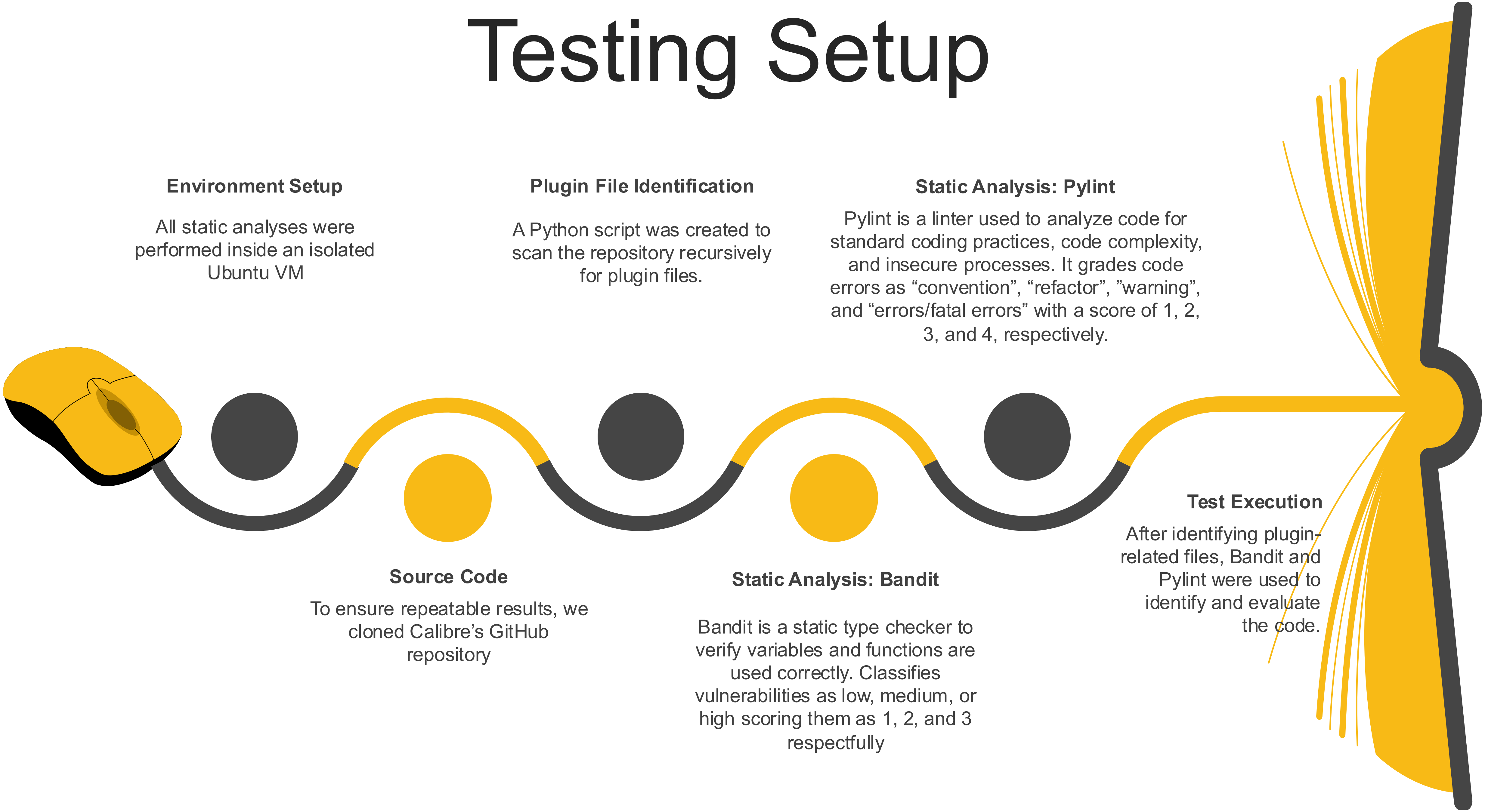
Remote Attacker

This attacker aims to target the network capability of Calibre’s plugin systems. Their goal is to identify vulnerable network endpoints for further malicious actions.

Local Attacker

This attacker is defined as someone who can access a local user's machine with the goal of tampering with plugins.

Testing Setup



Environment Setup

All static analyses were performed inside an isolated Ubuntu VM

Source Code

To ensure repeatable results, we cloned Calibre’s GitHub repository

Plugin File Identification

A Python script was created to scan the repository recursively for plugin files.

Static Analysis: Bandit

Bandit is a static type checker to verify variables and functions are used correctly. Classifies vulnerabilities as low, medium, or high scoring them as 1, 2, and 3 respectfully

Static Analysis: Pylint

Pylint is a linter used to analyze code for standard coding practices, code complexity, and insecure processes. It grades code errors as “convention”, “refactor”, ”warning”, and “errors/fatal errors” with a score of 1, 2, 3, and 4, respectively.

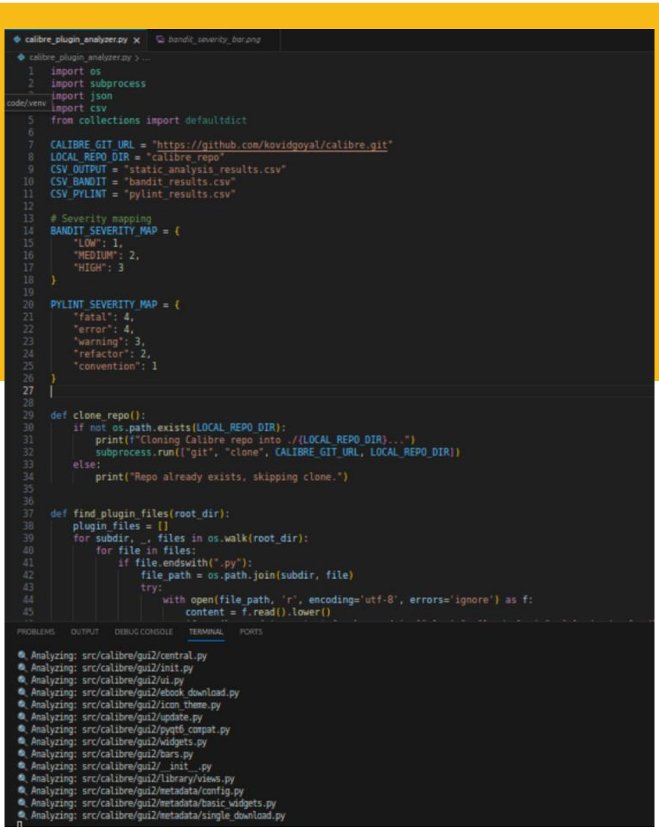
Test Execution

After identifying plugin-related files, Bandit and Pylint were used to identify and evaluate the code.

Initial Scans

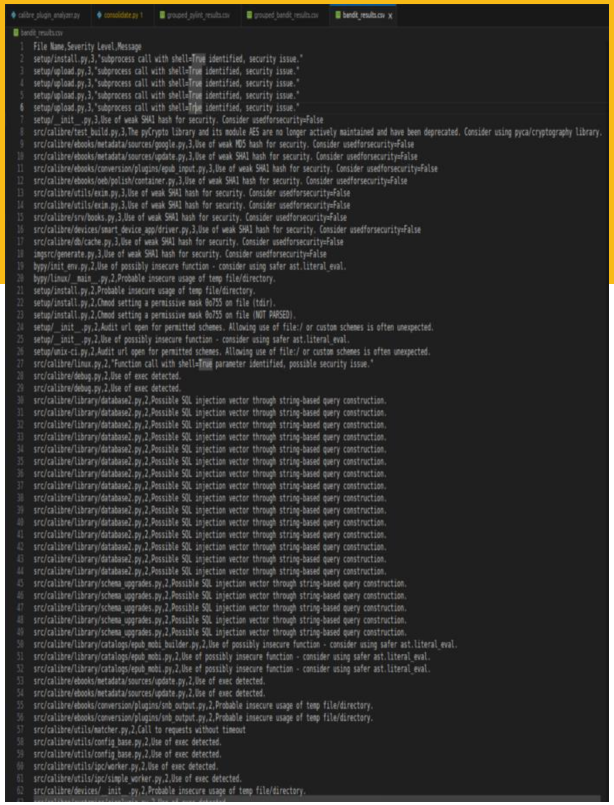
Locate
Scan
Analyze

- Our Initial run found 319 plugin-related files and scanned them with Pylint and Bandit
- Resulted in over 23,000 hits combined
- We wanted to find the most critical areas within Calibre's plugin system so we consolidated our data ...



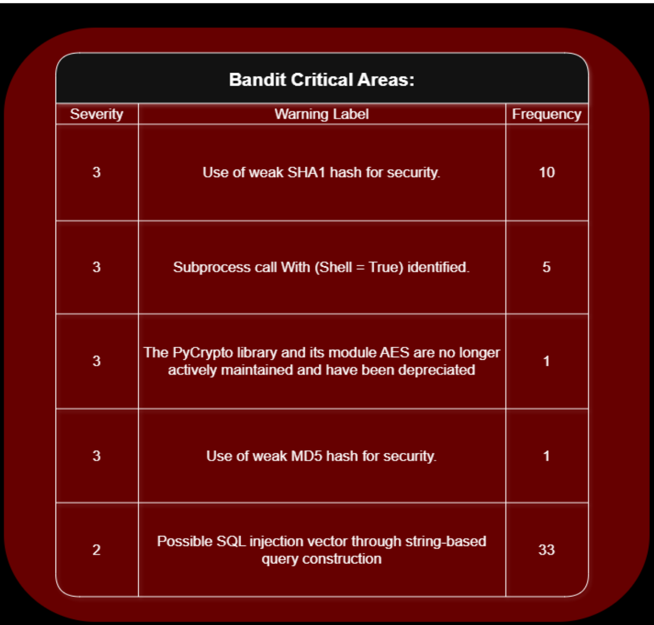
```python
import os
import subprocess
import json
import csv
from collections import defaultdict

CALIBRE_GIT_URL = "https://github.com/kovidgoyal/calibre.git"
LOCAL_REPO_DIR = "calibre_repo"
CSV_OUTPUT = "static_analysis_results.csv"
CSV_BANDIT = "bandit_results.csv"
CSV_PYLINT = "pylint_results.csv"

# Severity mapping
BANDIT_SEVERITY_MAP = {
    "LOW": 1,
    "MEDIUM": 2,
    "HIGH": 3
}

PYLINT_SEVERITY_MAP = {
    "fatal": 4,
    "error": 4,
    "warning": 3,
    "refactor": 2,
    "convention": 1
}


def clone_repo():
    if not os.path.exists(LOCAL_REPO_DIR):
        print(f"Cloning Calibre repo into ./{LOCAL_REPO_DIR}...")
        subprocess.run(["git", "clone", CALIBRE_GIT_URL, LOCAL_REPO_DIR])
    else:
        print("Repo already exists, skipping clone.")


def find_plugin_files(root_dir):
    plugin_files = []
    for subdir, _, files in os.walk(root_dir):
        for file in files:
            if file.endswith(".py"):
                file_path = os.path.join(subdir, file)
                try:
                    with open(file_path, 'r', encoding='utf-8', errors='ignore') as f:
                        content = f.read().lower()
```

```
🔍 Analyzing: src/calibre/gui2/central.py
🔍 Analyzing: src/calibre/gui2/init.py
🔍 Analyzing: src/calibre/gui2/ui.py
🔍 Analyzing: src/calibre/gui2/ebook_download.py
🔍 Analyzing: src/calibre/gui2/icon_theme.py
🔍 Analyzing: src/calibre/gui2/update.py
🔍 Analyzing: src/calibre/gui2/pyqt6_compat.py
🔍 Analyzing: src/calibre/gui2/widgets.py
🔍 Analyzing: src/calibre/gui2/bars.py
🔍 Analyzing: src/calibre/gui2/__init__.py
🔍 Analyzing: src/calibre/gui2/library/views.py
🔍 Analyzing: src/calibre/gui2/metadata/config.py
🔍 Analyzing: src/calibre/gui2/metadata/basic_widgets.py
🔍 Analyzing: src/calibre/gui2/metadata/single_download.py
```


calibre_plugin_analyzer.py
consolidate.py 1
grouped_pylint_results.csv
grouped_bandit_results.csv
bandit_results.csv
bandit_results.csv

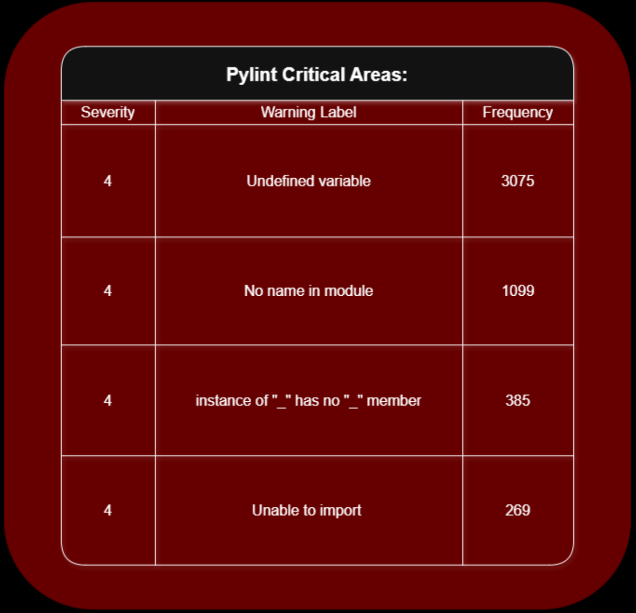
```
File Name,Severity Level,Message
setup/install.py,3,"subprocess call with shell=True identified, security issue."
setup/upload.py,3,"subprocess call with shell=True identified, security issue."
setup/upload.py,3,"subprocess call with shell=True identified, security issue."
setup/upload.py,3,"subprocess call with shell=True identified, security issue."
setup/upload.py,3,"subprocess call with shell=True identified, security issue."
setup/__init__.py,3,Use of weak SHA1 hash for security. Consider usedforsecurity=False
src/calibre/test_build.py,3,The pyCrypto library and its module AES are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.
src/calibre/ebooks/metadata/sources/google.py,3,Use of weak MD5 hash for security. Consider usedforsecurity=False
src/calibre/ebooks/metadata/sources/update.py,3,Use of weak SHA1 hash for security. Consider usedforsecurity=False
src/calibre/ebooks/conversion/plugins/epub_input.py,3,Use of weak SHA1 hash for security. Consider usedforsecurity=False
src/calibre/ebooks/oeb/polish/container.py,3,Use of weak SHA1 hash for security. Consider usedforsecurity=False
src/calibre/utils/exim.py,3,Use of weak SHA1 hash for security. Consider usedforsecurity=False
src/calibre/utils/exim.py,3,Use of weak SHA1 hash for security. Consider usedforsecurity=False
src/calibre/srv/books.py,3,Use of weak SHA1 hash for security. Consider usedforsecurity=False
src/calibre/devices/smart_device_app/driver.py,3,Use of weak SHA1 hash for security. Consider usedforsecurity=False
src/calibre/db/cache.py,3,Use of weak SHA1 hash for security. Consider usedforsecurity=False
imgsrc/generate.py,3,Use of weak SHA1 hash for security. Consider usedforsecurity=False
bypy/init_env.py,2,Use of possibly insecure function - consider using safer ast.literal_eval.
bypy/linux/__main__.py,2,Probable insecure usage of temp file/directory.
setup/install.py,2,Probable insecure usage of temp file/directory.
setup/install.py,2,Chmod setting a permissive mask 0o755 on file (tdir).
setup/install.py,2,Chmod setting a permissive mask 0o755 on file (NOT PARSED).
setup/__init__.py,2,Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected.
setup/__init__.py,2,Use of possibly insecure function - consider using safer ast.literal_eval.
setup/unix-ci.py,2,Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected.
src/calibre/linux.py,2,"Function call with shell=True parameter identified, possible security issue."
src/calibre/debug.py,2,Use of exec detected.
src/calibre/debug.py,2,Use of exec detected.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/database2.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/schema_upgrades.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/schema_upgrades.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/schema_upgrades.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/schema_upgrades.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/schema_upgrades.py,2,Possible SQL injection vector through string-based query construction.
src/calibre/library/catalogs/epub_mobi_builder.py,2,Use of possibly insecure function - consider using safer ast.literal_eval.
src/calibre/library/catalogs/epub_mobi.py,2,Use of possibly insecure function - consider using safer ast.literal_eval.
src/calibre/library/catalogs/epub_mobi.py,2,Use of possibly insecure function - consider using safer ast.literal_eval.
src/calibre/ebooks/metadata/sources/update.py,2,Use of exec detected.
src/calibre/ebooks/metadata/sources/update.py,2,Use of exec detected.
src/calibre/ebooks/conversion/plugins/snb_output.py,2,Probable insecure usage of temp file/directory.
src/calibre/ebooks/conversion/plugins/snb_output.py,2,Probable insecure usage of temp file/directory.
src/calibre/utils/matcher.py,2,Call to requests without timeout
src/calibre/utils/config_base.py,2,Use of exec detected.
src/calibre/utils/config_base.py,2,Use of exec detected.
src/calibre/utils/ipc/worker.py,2,Use of exec detected.
src/calibre/utils/ipc/simple_worker.py,2,Use of exec detected.
src/calibre/devices/__init__.py,2,Probable insecure usage of temp file/directory.
```

Consolidated Results



Bandit Critical Areas:

Severity	Warning Label	Frequency
3	Use of weak SHA1 hash for security.	10
3	Subprocess call With (Shell = True) identified.	5
3	The PyCrypto library and its module AES are no longer actively maintained and have been depreciated	1
3	Use of weak MD5 hash for security.	1
2	Possible SQL injection vector through string-based query construction	33



Pylint Critical Areas:

Severity	Warning Label	Frequency
4	Undefined variable	3075
4	No name in module	1099
4	instance of "_" has no "_" member	385
4	Unable to import	269

Findings:

vulnerabilities & misconfigurations

Security Threats

- Cryptographic weaknesses were found due to the use of outdated hashing algorithms like SHA1 and MD5, which are susceptible to collision attacks
- The use of `shell=True` in subprocess calls introduces a risk of shell injection, especially if tied to user input
- Reliance on the deprecated PyCrypto library exposes the system to known security flaws
- Raw string formatting in SQL queries opens the door to SQL injection attacks

Misconfigurations

- invalid or deprecated imports
- Attempts to access nonexistent object attributes
- Undefined Variables
- Unresolved module configurations

=========================

These issues can cause runtime crashes, broken plugin functionality, or failure to load key components, all of which reduce system reliability and increase the attack surface.

Proposed Mitigations

Secure Configuration Tips

Disabling the “Allow installing from untrusted sources” option prevents unverified plugins downloads and updates.

Users should routinely verify that they are current on all software and security updates

Code/Design Changes

Introducing a permission-based system with strong authentication to monitor and restrict plugin actions.

Adding an official plugin repository for monitoring and vetting.

Plugin Security Suggestion

Adding a ”required reading” for all plugin authors. Can include guidelines of secure coding practices or a plugin's rulebook.

The goal is to enhance security without compromising customizability.

Conclusion

Assessment Summary

Our analysis revealed over 22,000 Pylint messages and 462 Bandit security warnings, highlighting both minor issues and critical security flaws.

Bandit exposed vulnerabilities such as weak cryptographic algorithms, areas where unwanted shell command execution may be present, import errors that indicate weak runtime errors, and areas that may be open to SQL injections, to name a few.

Takeaway

Calibre's flexible plugin system is core to its appeal but represents one of its most significant security challenges. Our analysis shows that the plugin system can become a large surface area for attacks and system instability without careful management.

THANK YOU