Automating Personal File Integrity Monitoring with Python

Lately, I've been getting back into programming and learning various ways to "automate the boring stuff" with Python. As a Security Engineer, File Integrity Monitoring (FIM) is an important security control we often deal with, ensuring that checksums match to verify file integrity.

In enterprise architectures, we have tools like Trend Micro that offer FIM capabilities, along with other specialized FIM solutions. But I got to thinking - how can we implement something similar for our personal use cases at home?

That's the question I wanted to answer, and in the process, I created a tool to automate this task. Let me walk you through my thought process and the solution I developed.

**The Problem: Personal File Integrity Monitoring**

In our professional lives, we have robust tools to monitor file integrity across large networks. But what about our personal computers? We download files, install software, and transfer data regularly. How can we ensure these files haven't been tampered with during transit or storage?

**The Solution: A Python-based File Checksum Tracker**

To address this, I created a Python script that essentially brings enterprise-level FIM capabilities to personal use. Here's what it does:

1. Monitors the Downloads folder for new or modified files.

2. Automatically calculates SHA256 checksums for these files.

3. Stores the checksums in both Excel and JSON formats for easy reference.

4. Continuously updates these records as files change.

**You can find the full project on GitHub:**

(https://github.com/GurkhaShieldForce/File-Checksum-Tracker)

**Why This Matters**

While this might seem like a simple tool, its implications for personal cybersecurity are significant:

1. Malware Detection: By tracking checksums, you can quickly identify if a downloaded file has been altered, potentially indicating malware.

2. Software Integrity: Ensure that the software you've downloaded matches the checksum provided by the vendor.

3. Personal Data Protection: Monitor the integrity of your important personal files over time.

**Looking Forward: Potential Enhancements**

This project is just a starting point. There are numerous ways it could be expanded:

1. Implement automatic cross-checking of checksums against vendor-provided values.

2. Integrate with threat intelligence feeds to flag known malicious checksums.

3. Extend monitoring to other crucial directories beyond Downloads.

4. Develop a user interface for easier interaction and monitoring.

**Call to Action**

If you're interested in personal cybersecurity or looking for a Python project to contribute to, I encourage you to check out our GitHub. Whether you want to use it, enhance it, or just explore how it works, your involvement is welcome.

Remember, good cybersecurity practices start at home. By bringing enterprise-level thinking to our personal computing environments, we can significantly enhance our digital safety.

Have you worked on similar personal security projects? I'd love to hear about your experiences and ideas in the comments!