Task 1: SAST Tool Analysis Report

This document presents the results of a Static Application Security Testing (SAST) analysis using the Semgrep tool on a sample Python script. The objective is to identify vulnerable code units and provide recommendations for remediation.

Vulnerabilities Identified

1. Use of subprocess with shell=True

File: task1_code.py
Line: 7
Issue: The 'subprocess.call' function is used with 'shell=True'. This can be dangerous because it allows shell injection vulnerabilities, making it easier for a malicious actor to execute arbitrary commands.
Recommendation: Use 'shell=False' or use safer alternatives like 'subprocess.run' with a list of arguments.

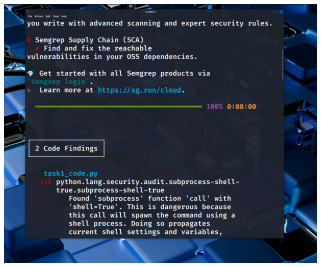
2. Use of MD5 for password hashing

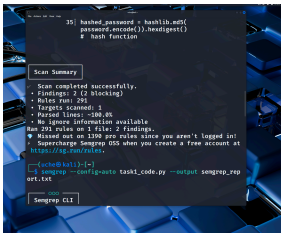
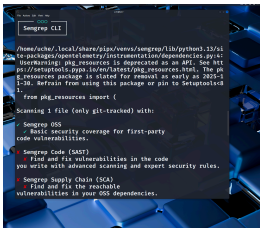
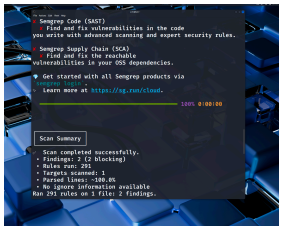
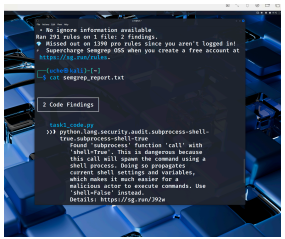
File: task1_code.py
Line: 35
Issue: The code uses the MD5 hashing algorithm for password hashing. MD5 is not considered secure as it is susceptible to collision and brute-force attacks.
Recommendation: Use a modern and secure password hashing algorithm such as 'scrypt', 'bcrypt', or 'argon2'.

Conclusion

Two critical issues were identified in the sample code using Semgrep: one issue related to unsafe subprocess usage and the other to insecure password hashing. Remediation steps should be applied to improve the security of the code.

Screenshots:(1-8) Semgrep scan command, Findings on subprocess.call and MD5 hashing warnings.


Task No 2: Crack the hash using "Hashcat" or "John the Ripper" and provide a screenshot of the program output in the answer.

Here are the hashes to crack:

- 5d41402abc4b2a76b9719d911017c592
- 098f6bcd4621d373cade4e832627b4f6
- ed8578edf8458ce06fbc5bb76a58c5ca4
- 25d55ad283aa400af464c76d713c07ad
- 99a18c428cb38d5f260853678922e03

Cracking MD5 Hashes Using Hashcat with the command:

(hashcat -m 0 -a 0 hash.txt /usr/share/wordlists/rockyou.txt)

MD5 Hash	Cracked Password
5d41402abc4b2a76b9719d911017c592 -	hello
098f6bcd4621d373cade4e832627b4f6 -	test
e99a18c428cb38d5f260853678922e03 -	abc123
d8578edf8458ce06fbc5bb76a58c5ca4 -	qwerty
25d55ad283aa400af464c76d713c07ad -	12345678

Screenshot of Cracked Hashes Output: