**Tools for Enforcing Secure Code**

**Student**: Patrick Walsh

**Date**: 4/23/2021

**School**: University of Maryland Global Campus

**Course**: SDEV 360 7380

**Professor**: Dr. Alla Webb

**Overview of secure coding tools**

**GuardRails**

This secure coding tool provides “continuous security feedback” (GuardRails, 2021) so that developers can get status reports of their code in real-time. GuardRails alerts development teams of security alerts as they happen when code changes occur and integrates with Slack to send security notifications (GuardRails, 2021). It is highly flexible, without the need for special configuration and with support in a wide array of programming languages, including JavaScript, Python, Ruby, PHP, C, C++, and Java (GuardRails, 2021). GuardRails also provides a helpful dashboard for developers to view security reports (GuardRails, 2021).

*Strengths:*

* *Supports multiple languages*
* *Provides continuous secure code feedback*
* *Does not require configuration*
* *Provides dashboards to view security reports*

*Weaknesses*

* *Does not currently support C# or .NET, though these languages are being added (GuardRails, 2021)*
* *Must pay professional plan ($550 per month) for full language support and other features (GuardRails, 2020)*

**Safety**

This secure coding tool uses a vulnerability database that is updated monthly to check for security weaknesses in Python code (Bar-Gil, 2020). While Safety is listed as free and open-source (Bar-Gil, 2020), using the tool for commercial projects requires purchasing a license with flexible billing plans that range from $99 per year to $333 each month (PyUp, 2021). Safety can work in tandem with an automated security bot called PyUp Bot which keeps Python code dependencies up to date, but the two tools can be used separately (PyUp, n.d.). Safety also offers a command line tool for configuration and other tasks (PyUp, n.d.).

*Strengths:*

* *Uses vulnerability database that is updated monthly*
* *Free and open-source for personal projects*
* *Offers command line tool option*

*Weaknesses*

* *Designed exclusively for Python language*
* *Requires paid plan for commercial use*

**SonarQube**

Similar to GuardRails, this tool provides secure code analysis to discover vulnerabilities in a wide range of programming languages (Segal, 2019). In addition to alerting developers to code vulnerabilities, SonarQube also ranks vulnerabilities according to severity level (Segal, 2019). It provides an interactive Graphical User Interface (GUI) that is ideal for beginners, but also offers advanced features like analyzing pull requests, code branch tracking, project timeline visualization, and continuous integration with tools like Jenkins (Segal, 2019). SonarQube offers multi-language support in 27 different languages, including Java, C#, Python, JavaScript, PHP, Scala, HTML/CSS, and more (SonarQube, 2021a). Like other secure coding tools, SonarQube offers free and paid versions. Unlike other tools, however, SonarQube offers a free open-source Community Edition that supports 15 different programming languages with Continuous Integration/Continuous Delivery (CI/CD) and access to over 50 community code plugins (SonarQube, 2021b). The pricing varies widely from Developer Edition to Enterprise and Data Center Editions that can support global code deployment (SonarQube, 2021b), with the paid service supporting analysis of up to 100,000 lines of code for $150 per year (SonarQube, 2021c).

*Strengths:*

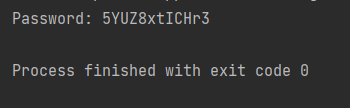
* *Wide language support in 27 different programming languages*
* *Scalable to support anyone from beginner-level developers all the up to global companies*
* *Code vulnerability ranking feature*
* *Offers advanced features for secure software development projects*

*Weaknesses*

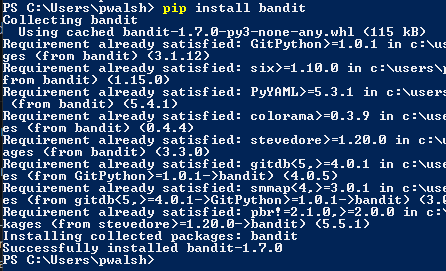
* *Pricing can be steep for larger code analysis tasks*

**Code vulnerability analysis use-case**

The code vulnerability use-case will focus on a completely free and open-source tool called Bandit. Bandit is designed to find common security weaknesses exclusive to Python (Bandit, 2020). To analyze Python code and generate vulnerability reports, Bandit processes individual Python files, builds an Abstract Syntax Tree (AST), and runs appropriate plugins against AST nodes (Bandit, 2020). For the use-case, I created a Python program which generates a random password made up of lowercase letters, uppercase letters, numbers, and special characters. Here is an example output using a password length of 12 characters:



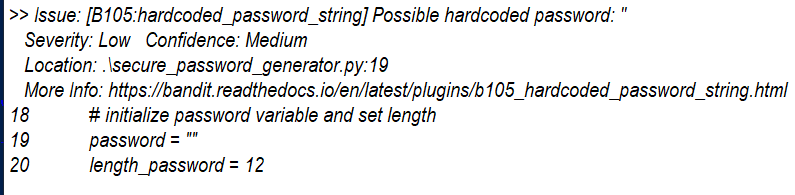
The program uses the Python ‘random’ module to generate a password. However, the ‘random’ module is considered a pseudo-random generator which is not considered secure (Python Software Foundation, 2021a). To get started, first install Bandit through pip on the command line:



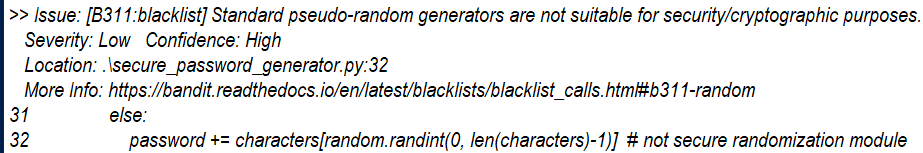
To run a security scan of the Python file and generate a report, navigate to the folder in the command line tool where the Python file is located and run the following command:



The report is created inside the same folder and reveals two security issues. The first is a minor warning about a possible hardcoded password found in the password variable. The password variable is initialized and set to an empty string at first. This issue is not a major concern:



The second security issue is more critical. Bandit recognized that use of the ‘random’ module is not suitable for secure cryptographic purposes. The secure password generator should not use a pseudo-random generator to generate passwords due to known security vulnerabilities:



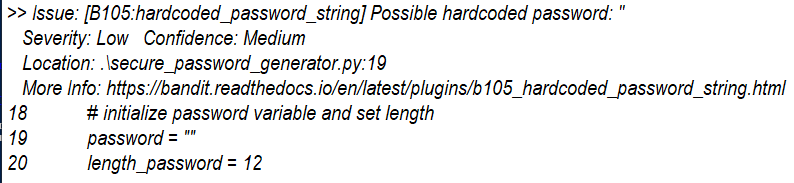
To mitigate this problem, the Python file was modified to use the ‘secrets’ module rather than ‘random’. The ‘secrets’ module is suitable for secure cryptographic purposes like secure password generators (Python Software Foundation, 2021b). The program is updated on line 33 to use the ‘secrets’ module:



Now Bandit is run again, and a second report is generated:



The first security issue referencing the possible hardcoded password is still present, but the second, more severe issue has been resolved:



**References**

Bandit. (2020, December 12). *PyCQA/bandit*. Bandit GitHub Page. https://github.com/PyCQA/bandit

Bar-Gil, G. (2020, September 14). *Best Python Open Source Security Tools*. SecureCoding. https://www.securecoding.com/blog/best-python-open-source-security-tools/

GuardRails. (2020, November 2). *GuardRails Pricing*. https://www.guardrails.io/pricing/

GuardRails. (2021). *What is GuardRails?* GuardRails GitHub Page. https://github.com/apps/guardrails

Python Software Foundation. (2021a). *random — Generate pseudo-random numbers — Python 3.9.4 documentation*. Python Standard Library. https://docs.python.org/3/library/random.html

Python Software Foundation. (2021b). *secrets — Generate secure random numbers for managing secrets — Python 3.9.4 documentation*. Python Standard Library. https://docs.python.org/3/library/secrets.html#module-secrets

PyUp. (2021). *Safety Pricing*. PyUp.Io. https://pyup.io/pricing/

PyUp. (n.d.). *PyUp Documentation*. PyUp.Io. https://pyup.io/docs/

Segal, E. (2019, November 15). *7 Open-Source Tools for Securing Your Code*. DEV Community. https://dev.to/eddiesegal/8-open-source-tools-for-securing-your-code-4gm5

SonarQube. (2021a). *Static Code Inspection & Code Analysis Tools | SonarQube*. SonarQube Website. https://www.sonarqube.org/features/multi-languages/

SonarQube. (2021b, March). *Download | SonarQube*. SonarQube Website. https://www.sonarqube.org/downloads/

SonarQube. (2021c). *Request Free Trial | SonarQube Developer Edition | SonarQube*. SonarQube Website. https://www.sonarqube.org/trial-request/developer-edition/