Next Steps in Rule Writing and Testing with Semgrep

As mentioned in the webinar, writing rules is the best way to learn about Semgrep! We've come up with four real-world scenarios for writing Semgrep rules. Your objective is to use the Semgrep Playground to fill out skeleton rules and make their corresponding tests pass. Check out the Semgrep docs if you need a refresher on creating rules and testing rules in the Playground.

Click on each of the Playground links below, read through the tests in the code panel on the right, and fill out the Semgrep rule to make all the tests pass. There is a hint included with each exercise. Highlight the text to uncover the hint. text

1. Disabling TLS verification may allow an attacker to intercept and modify encrypted network traffic. Fix this Semgrep rule to detect insecure TLS connections using the Python requests library: https://semgrep.dev/playground/s/6JlE1.
   Hint:

2. In Python, the __init__ function is a class constructor. It constructs the internal state of a class upon initialization. Using return statements inside a class's __init__ function is a semantically invalid operation. Fix this Semgrep rule to detect the use of return inside a class's __init__ function: https://semgrep.dev/playground/s/pKLBZ.
   Hint:

3. Web services listening on all network interfaces may allow an attacker to access the local service on public WiFi networks. They may also allow unintended access on internal production networks. Fix this Semgrep rule to detect Docker Compose services that are listening on all interfaces: https://semgrep.dev/playground/s/zdLzd.
   Hint:

4. Superuser, or "well-known", ports are Linux ports below 1024. These typically require root access to listen on, and running services as root violates the principle of least privilege. Fix this Semgrep rule to detect JavaScript Express web services that are listening on ports below 1024: https://semgrep.dev/playground/s/X5RXB.
   Hint:

Have Questions?

#testing-handbook

Join our Tool Testing Handbook Slack to give feedback, ask questions, and chat with the community.