A vulnerability scanner in Python can be created using a variety of libraries and modules. Some popular libraries and modules that can be used to create a vulnerability scanner in Python include:

1. Scapy: A powerful packet manipulation library that can be used to perform network scans and identify open ports on a target system.
2. nmap: A popular open-source port scanner that can be used to perform network scans and identify open ports on a target system.
3. sqlmap: An open-source SQL injection tool that can be used to test web applications for SQL injection vulnerabilities.
4. Metasploit: A powerful framework for developing, testing, and executing exploits that can be used to target vulnerabilities in a wide range of systems.
5. OpenVAS: An open-source vulnerability management system that can be used to perform vulnerability scans and identify vulnerabilities in a target system.
6. OWASP ZAP: is a free and open-source web application security scanner which can be integrated with python as a library.
7. Python Requests library: To interact with web application and check for vulnerability

To create a vulnerability scanner in Python, you would typically use a combination of these libraries and modules to perform network scans, identify open ports, test for vulnerabilities, and exploit any vulnerabilities that are found. Additionally, you can use Python scripts to automate the process of scanning and reporting vulnerabilities, as well as integrating it with other tools.

import nmap

# Initialize the nmap object

nm = nmap.PortScanner()

# Perform a basic scan on the target host

nm.scan('192.168.1.1', '1-1024')

# Print out the results of the scan

for host in nm.all\_hosts():

print('Host : %s (%s)' % (host, nm[host].hostname()))

print('State : %s' % nm[host].state())

for proto in nm[host].all\_protocols():

print('Protocol : %s' % proto)

lport = nm[host][proto].keys()

lport.sort()

for port in lport:

print ('port : %s\tstate : %s' % (port, nm[host][proto][port]['state']))

import nmap

import sqlmap

# Initialize the nmap object

nm = nmap.PortScanner()

# Perform a basic scan on the target host

nm.scan('192.168.1.1', '1-1024')

# Print out the results of the scan

for host in nm.all\_hosts():

print('Host : %s (%s)' % (host, nm[host].hostname()))

print('State : %s' % nm[host].state())

for proto in nm[host].all\_protocols():

print('Protocol : %s' % proto)

lport = nm[host][proto].keys()

lport.sort()

for port in lport:

print ('port : %s\tstate : %s' % (port, nm[host][proto][port]['state']))

if nm[host][proto][port]['state'] == 'open':

# Check for SQL injection vulnerabilities on open ports running web applications

sqlmap.scan(host, port, options='-u "http://192.168.1.1:8080/test.php?id=1" --level=5 --risk=3')

This script uses the nmap library to perform a basic port scan on the target host (192.168.1.1) and print out the results of the scan. The scan is performed on ports 1-1024. The script loops through all the hosts that are found and prints out information about each host, including the state of the host and the protocols and ports that are open on the host. If the port is open and running a web application, the script uses sqlmap library to check for SQL injection vulnerabilities on the web application running on that port.

You should be aware that using sqlmap without the proper understanding of it could cause harm to the target system or the network, it's important to be aware of the options and arguments that you pass to the scan function.

Additionally, it's a good practice to handle any exception that might occur and also to save the results in a file for further analysis.

Please be aware that before running a vulnerability scanner on any system, you must have explicit permission from the system owner and also be aware of laws and regulations that might apply to your actions.