

Automating the Creation, Maintenance, and Use of Nessus Scanners Using Ansible and Jenkins



Joseph Dickinson, University of Illinois at Urbana Champaign, M.Eng. Electrical and Computer Engineering

Project Mentor: Jayson Grace, Org. 9317

■ Problem Statement:

In order to evaluate the health of a network, vulnerability handlers utilize products such as Nessus and SecurityCenter that automate the process of scanning servers for known security vulnerabilities. While these products address the issue of having to manually test an entire network for vulnerabilities, maintaining the Nessus/SecurityCenter ecosystem requires constant monitoring of Nessus Scanners' health, manual updates when newer versions are released, and evaluating scan results to determine the presence of critical vulnerabilities on the network.

■ Objectives:

- Create a system to automatically pull down, rebuild, and register Nessus Scanners with SecurityCenter if they break or a newer version of Nessus is available
- Alert vulnerability handlers when Nessus finds critical vulnerabilities so that they can be handled immediately

■ Approach:

- Create Ansible playbooks that use pre-existing bootstrapping methods in an automated fashion
- Utilize Ansible Vaults so that credentials and other sensitive information are stored encrypted on the system and decrypted only during runtime
- Schedule a job after each Nessus scan using Jenkins that evaluates the severity of discovered vulnerabilities and alerts handlers of critical results in Mattermost

■ Impact and Benefits:

- Nessus Scanners are automatically updated on a weekly basis to the latest version of Nessus. This allows for vulnerability handlers to have the most up-to-date information about vulnerabilities on Sandia's network
- If a Nessus Scanner stops communicating with SecurityCenter, it is automatically torn down and rebuilt. If the rebuild process isn't successful, a team member is notified.

