Linux Hardening Audit Tool

@SRCybersecurity

Introduction

The **Linux Hardening Audit Tool** to audit a Linux system's security configuration.

Features:

- Check firewall rules, unused services, SSH settings
- Verify permissions on key files
- Check for rootkit indicators
- Generate a score/report based on CIS benchmarks
- Recommend hardening actions

Abstract

The **Linux Hardening Audit Tool** is a security-focused utility designed to assess and improve the security posture of Linux-based systems. The tool performs a comprehensive audit by evaluating critical system components and configurations, aligning its checks with industry-recognized CIS (Center for Internet Security) benchmarks. Key features include analysis of firewall rules, detection of unused or vulnerable services, inspection of SSH configurations, verification of file permissions for sensitive system files, and scanning for rootkit indicators. Upon completion, the tool generates a detailed report with a security score, highlighting compliance gaps and offering actionable recommendations for system hardening. This project aims to simplify the security auditing process, helping system administrators proactively identify and remediate vulnerabilities in Linux environments.

Tools Used:

- Bash
- Python,
- OS,
- subprocess

Steps Involved in Building the Project:

Dissected into four main steps:

1. Check firewall active service functions are used to interact with the system's firewall using the Linux iptables command.
2. Check network services, ssh configuration, file permission integrity, basic rootkit/malware detection functions provide a foundational security audit covering SSH hardening, critical file permissions, rootkit indicators, and risky service detection.

3. Generate report function summarize the audit in both terminal and browser. Format the results into a structured, human-readable report (HTML).
4. The main() function is the core driver of a Linux security audit tool. It systematically performs multiple security checks, evaluates the system against best practices (like those from CIS), and generates clear, actionable output.

Conclusion

The **Linux Hardening Audit Tool** successfully meets its primary objective: to audit a Linux system's security configuration and provide clear, actionable insights based on CIS (Center for Internet Security) benchmarks. Developed in Python using built-in modules like os, subprocess, and stat, the tool automates essential checks that are often performed manually during system hardening.

By examining firewall rules, detecting insecure or unnecessary services, analysing SSH configurations, verifying file permissions, and scanning for potential rootkit indicators, the tool provides a comprehensive snapshot of the system's security posture. Each audit section contributes to a cumulative score, giving users a thorough understanding of how secure their system is.

Additionally, the tool generates a well-structured HTML report and provides tailored hardening recommendations to help administrators take corrective actions efficiently. Its modular design also makes it easy to expand with additional security checks in the future.

Overall, this mini project demonstrates the power of scripting in automating routine security tasks and reinforces the importance of regular audits in maintaining a secure Linux environment.

THANK YOU

END