HardnBot

INTELLIGENT SERVER HARDENING SOFTWARE

Project ID: **19_20-J01**

Preliminary Progress Review Report

R.M.B.B Rathnayake

B.Sc. (Hons) Degree in Information Technology

Sri Lankan Institute of Information Technology Sri
Lanka

13th May 2019

HardnBot

INTELLIGENT SERVER HARDENING SOFTWARE

Project ID: **19_20-J01**

Preliminary Progress Review

Supervisor:

Mr. Amila Senarathne

B.Sc. (Hons) Degree in Information Technology

Sri Lankan Institute of Information Technology Sri
Lanka

13th May 2019

DECLARATION

I declare that this is my own work and this Preliminary Progress Review (PPR) report does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

………………………..

R.M.B.B Rathnayake

IT16054400

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Supervisor

……………………….

Mr. Amila Senarathne

Table of Contents

List of Figures

1. Introduction

1.1. Purpose

Purpose of this Preliminary Progress Review is to provide a clear understanding of our research project to the interested parties. Through this document we will explain the purpose, functional and non-functional requirements, software/hardware components to be used and the end product in implementing following components:

- Implement scans to detect compliance issues.
- Automatically harden the system to classified compliance issues.
- Bring industry best practices into system.

In here, we consider the interested parties to be research team, supervisors, research parties who are implementing the same sort of tool and the businesses who can benefit from our product. This document will regularly review above mentioned phases of our research project and identify that the research team can fulfil requirements mentioned above. This will compare the parameters used in the project and their outcome. This PPR will allow to identify and correct any deviation in outcomes and changes in requirements.

This document will be written in a way that everyone can understand about the research and this document can be used to communicate with industry members regarding our research. This PPR report will be a good reference to other who are interested in implementing a same kind of tool.

1.2. Definitions, Acronyms, and Abbreviations

CIS	Center for Internet Security
GUI	Graphical User Interface

1.3. Scope

The scope of this PPR is a detailed review of functions mentioned below.

- Implement scans to detect compliance issues.
- Automatically harden the system to classified compliance issues.
- Bring industry best practices into system.

Each of these functions are important to the development of final product and each of these functions have different phases and technical requirements. These phases and requirements will be described and validated through this report.

1.4. Overview

Server Hardening is the one of the most important tasks to be handled on servers. Server hardening, which is also referred to as operating system hardening, is the process of making the server stronger and more resistant to security issues. Server hardening is an inexpensive and simple task to improve the overall operating system for maximum performance and to reduce expensive failures. Hardening is a Process requires many steps, all of which are critical to the success of the hardening system. The more steps a user follows, the safer and more resilient the system will be.

Using strong passwords, establish a password lockout policy, establish data encryption mechanism are sample tasks that are just the tip of the iceberg as far as server hardening is concerned. Many more tasks must be completed, and each is more complex. For proper execution and to maximum protection, professional assistance from an expert is needed.

Currently there is no fully automated server Hardening tool in information system. Although there are many vulnerability scanners and other auditing tools available for Application and Network audits, there are no proper tools for OS hardening.

This research is on building an automated tool which is capable of conducting a complete information systems compliance scan and classify compliance issues with a trained machine learning model. Which predicts the risk score efficiently and perform system hardening automatically to classified compliance issues and bring industry best practices to systems while utilizing minimum resources.

The main goal here is to implement scans to detect compliance issues in a system. These scans will retrieve existing system configurations and those configurations will be compared and detect compliance issues.

System generates hardening scripts to correct the classified compliance issues and automatically execute in particular server. System owner doesn’t have to access Unix environment anymore. The hardening configurations will be displayed to users with appropriate parameters.

For a particular parameter in the system configuration, there is an industry recognized value. For example, a password should expire at most in 90 days. A company may not adhere to these standard values; their security policies may not describe them. In such occasions, system administrators may have assigned them with default values. Through our software, we plan to introduce industry accepted values and configurations into the information systems.

2. Statement of work

2.1. Background

There are many vulnerability scanners available to detect vulnerabilities in an operating system and a network. But these vulnerability scans are traditional (conduct vulnerability scans on common vulnerable areas in a system) and their capabilities are limited. These vulnerability scanners cannot be used to enforce a system policy.

Automated Audit Platform is a complete software toolkit that has the capability of detecting, fixing and monitoring poor system configurations. It has advanced scan patterns that are capable of moving through system policies and configurations to detect issues with them.

Unlike other vulnerability scanners, in order to identify compliance issues HardnBot will scan configurations of,

1. Install Updates, Patches and Additional Security Software
2. OS Services
3. Special Purpose Services
4. Network Configuration and Firewalls
5. Logging and Auditing
6. System Access, Authentication and Authorization
7. User Accounts and Environment
8. Warning Banners
9. System Maintenance

A research conducted by Prowse D. in 2010 on “OS Hardening and Virtualization” describes how to perform OS hardening using OS security audit method. In order to perform OS hardening, as a first step they perform vulnerability assessment over windows, then perform the security audit and fully analysis system logs. Further they explained how important it is to perform periodic security audit over an OS in order to track vulnerabilities and take relevant countermeasures. As benefit of doing OS hardening, they printed out how it helps to reduce the risk, improve the performance, eliminates vulnerable entry points and mitigate security risks. As this paper status OS hardening can be done using techniques such as program clean-up, service packs, patch management, group policies, see templates and configuration baselines. Further for more user friendliness, operating systems like windows provides facilities to prioritize vulnerabilities as high, medium and low. To strengthen the security of OS, they discussed manual technology as well as semi-automated terms under manual techniques. Preparing checklist for security parameters, reviewing security configuration aspects, manually set security configuration and explaining OS as per configuration parameter included. In semi-automated way they are using scripts for audit such as .bat, .ps, set security configuration using script, exploiting OS scripted pay load. In discussion they showed how important it is to perform periodic audits to identify security issues, prioritize those and treat in order to mitigate risk over operating systems [1].

A research conducted by Sanjay Garg on “Network Scanning & Vulnerability Assessment with Report Generation” ha reviewed two of the well-known open source scanners NMAP (Network Mapper) & OpenVAS (Open Vulnerability Assessment System).

They show us how to incorporate these two scanners into a decently outlined GUI and give reliable information. Effectiveness of network scanning and vulnerability testing depends on scanners and processes to scan the network and its devices. Sometimes, use of these tools can lead to device or information being compromised or destroyed by exploits. Different implementations & tools of network scanning have different kinds of outputs. But these outputs are typically heterogeneous.

The network scanner created in this thesis carries out the scanning through the network identifying the active hosts and guessing the operating system of the remote hosts and the programs installed in the remote hosts.

In addition to identifying active hosts, you could find open ports and list the services that run on the host. The exploration of additional vulnerabilities is done by comparing the information obtained from a network scan with a database of vulnerability signatures to generate a list of vulnerabilities that are likely to be present in the network. In this dissertation, the characteristics of the new tool are explored. In other words, network mapping, vulnerabilities, and configuration failures in the network are displayed in various formats. In addition, an easy approach is defined to reduce the duration of vulnerability exploration [2].

AAP is not just a vulnerability scanner. It is a complete system auditing tool that can perform variety of security tasks on a system.

2.2. Identification and significance of the problem

Massive amounts of data are created daily across the planet. By 2021, the annual global Internet Protocol (IP) traffic is predicted to reach 3.3 zettabytes. To match this huge data environment, the data center industry is anticipating unprecedented growth. Data is the most precious asset in data centers. Data centers require abilities to ensure data service works properly; many technologies are used in data centers to achieve this goal. Data centers are supported to run 24/7/365 without interruption. Planned or unplanned downtime can cause business users serious damage.

Most data centers include Linux servers; Ubuntu Server, Red Hat Enterprise Linux and CentOS. Datacenter includes about 200 live servers it is very difficult to do the operating system hardening manually.

There is no fully automated hardening platform implemented yet. Even the network administrators plan to do the Hardening processes manually it might take more than 6 hours for the complete harden processes for only for one and may contain lots of paper work. By the way there can be mistakes and faults in the hardening process that can effect the live Servers.

Sometimes a company may have to hire an external party to perform the hardening or they may have to outsource their systems to external organizations to assess their compliance. This will cost them in advance. Cost of maintaining compliance and governance may higher than the risks associated with these systems. And also there could be risks in outsourcing critical information systems.

The aim of this research is to find an effective way to secure the datacenter RHEL 7,6 and CentOS 7,6 servers and enhance productivity for the customers and employees with a low cost and few human interactions.

Hence there is no any fully automated hardening platform implemented yet, in this system implementation a novel automated open source software is proposed. This research focuses to find an automated way to perform all Level 1 configuration profiles of CIS benchmark with Ansible automation tool.

Here fully automated free open source hardening platform is developed with the capability to detect poor or non-compliant configurations in a system (OS/DB/Application) and applying industry recommended fixes/configurations and secure systems by reducing its surface of vulnerabilities.

This research composes a great business value as it can cover 90% of an Information Systems Audit and hardening process. For internal audit, this software presents both opportunity and responsibility. By helping the organization understand and control risks and identifying opportunities/industry best practices to embrace.

2.3. Technical objectives

In this section it specifies software requirements and technologies.

Main Programming Language: C#

Most of the operating systems support for C#. And C# can be used to implement complex desktop applications as it contains many pre-defined functions, exception handling and lots of supporting third party libraries. Therefore, C# will be used as the main programming language to implement this software.

IDE: Visual Studio

Visual Studio IDE is mainly support for C# development and it contains lots of features that facilitates programming. There are lots of in-built functions and configurations libraries in Visual Studio. This IDE is very powerful and easy to use. Therefore, we will use Visual Studio as the IDE for our project development.

Modern UI Development: Bunifu DLL

Bunifu framework is known for modern interface development in C# and C++ environments. Bunifu contains a variety of colors and animated controls. It also contains scripts that allow to include animated effects into the interfaces. Therefore, we will use this framework in designing our interfaces.

Virtual Environment: VMware Workstation

We need a virtual environment to install and test all the OS and DB components that we are about to audit. For that purpose, we can use VMWare workstation virtual environments. By using VMWare, we can also conduct several tests and gather data under different hardware conditions.

Operating System: Linux

Although there are many auditing tools available for MS Windows environment there are very few tools available for Linux. We are planning on implementing the hardening for below Linux products.

RHEL 7,6 and CentOS 7,6

Automation Tool: Ansible

Ansible is an open-source software provisioning, configuration management, and application deployment tool. It runs on many Unix-like systems and can configure both Unix-like systems as well as Microsoft Windows. It includes its own declarative language to describe system configuration. We will use Ansible to achieve our expected automation hardening process.

Script Language: Shell

Shell script A shell script is a list of commands (a program) designed to be run by the Unix shell. We will use shell script for scanning purposes and other types of command executions in Linux servers.

3. Research Methodology

1. Implement scans to detect compliance issues

HardnBot perform scans to identify misconfigurations in operating systems. The software will run scripts to retrieve configurations for relevant locations and record them. These scans will retrieve existing system configurations and those configurations will be compared and detect compliance issues. Then these compliance issues will be presented to the user in a well formatted way.

In order to identify compliance issues HardnBot will scan configurations of followings,

1. Install Updates, Patches and Additional Security Software
2. OS Services
3. Special Purpose Services
4. Network Configuration and Firewalls
5. Logging and Auditing
6. System Access, Authentication and Authorization
7. User Accounts and Environment
8. Warning Banners
9. System Maintenance

Scanning for compliance issues is carried out by a well prepared shell scripts.

2. Automatically harden the system to classified compliance issues

After identifying compliance issues, to remediate the issues a script is automatically generated. This script is executed to systems by Ansible automation tool.

The default hardening and remediation to identified compliance issues is carried by configurations in CIS bench marks. System owner doesn't have to access Unix environment anymore. The hardening configurations will be displayed to users with appropriate parameters.

3. Bring industry best practices into system.

For a particular parameter in the system configuration, there is an industry recognized value. For example, a password should expire at most in 90 days. A company may not adhere to these standard values; their security policies may not describe them. In such occasions, system administrators may have assigned them with default values. Through our software, we plan to introduce industry accepted values and configurations into the information systems.

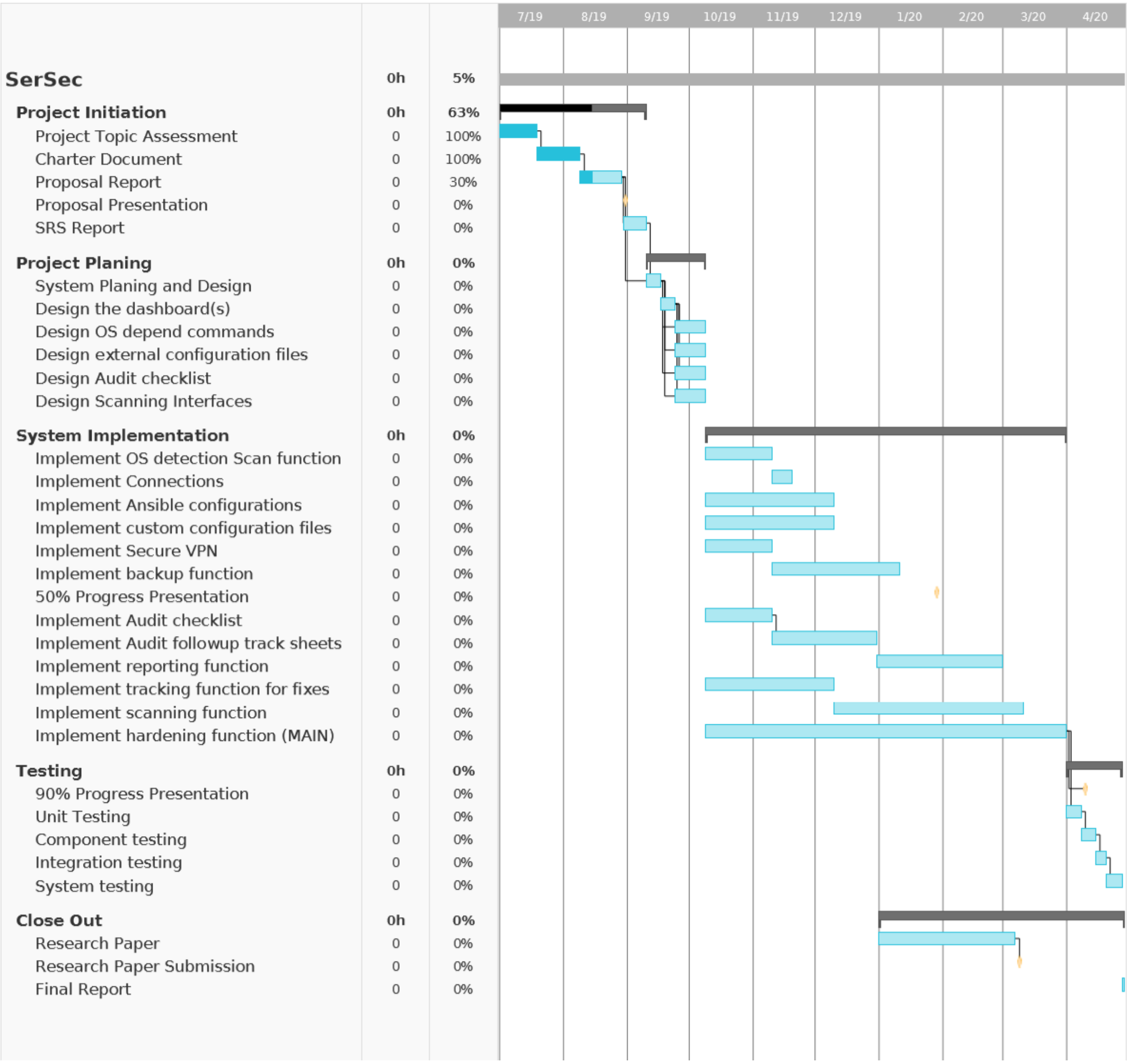
The hardening configurations will be displayed to users with default CIS benchmarks parameters. Editable GUIs are designed to users to customize these configurations according to industry requirements. Then these parameterized configurations should be passed to Ansible play books through HardnBot.

4. Anticipated benefits

Our research project contains a great commercial value with the functionalities that it will perform. We are targeting to automate entire information systems hardening process through our software. With the capabilities that we are about to include into our software, user can gain following benefits.

- **Fully automated operations**
  Needs less user interactions
- **Automatically detect compliance issues**
- **Apply fixes upon user confirmation**
- **Brings industry best practices into your system configurations**
- **Ease the internal audit**
  This software package can be handled by a single user. Therefore, the entire system hardening process can be conducted by a single user. This requires minimal amount of organization's resources. As for the internal audit, this software will cover up to 90% of their work.
- **Reduce manual work**
  Through this software, we target to reduce the manual effort needed for a complete system hardening. Almost all the parts of the audit and hardening could be performed through this software. Compliance issues are automatically classified using a trained machine learning model and risk score is predicted. Reports can be generated through this software at the end of the hardening process.
- **Rollback the backup when abnormal behavior is detected.**
  Through this software System will automatically detect any abnormal behavior in the system after the hardening is carried out. If any anomaly detected rollback function will take place and proceed to established earlier status (backup) of the server.

5. Project plan or schedule



SerSec	**0h**	**5%**
Project Initiation	**0h**	**63%**
Project Topic Assessment	0	100%
Charter Document	0	100%
Proposal Report	0	30%
Proposal Presentation	0	0%
SRS Report	0	0%
Project Planing	**0h**	**0%**
System Planing and Design	0	0%
Design the dashboard(s)	0	0%
Design OS depend commands	0	0%
Design external configuration files	0	0%
Design Audit checklist	0	0%
Design Scanning Interfaces	0	0%
System Implementation	**0h**	**0%**
Implement OS detection Scan function	0	0%
Implement Connections	0	0%
Implement Ansible configurations	0	0%
Implement custom configuration files	0	0%
Implement Secure VPN	0	0%
Implement backup function	0	0%
50% Progress Presentation	0	0%
Implement Audit checklist	0	0%
Implement Audit followup track sheets	0	0%
Implement reporting function	0	0%
Implement tracking function for fixes	0	0%
Implement scanning function	0	0%
Implement hardening function (MAIN)	0	0%
Testing	**0h**	**0%**
90% Progress Presentation	0	0%
Unit Testing	0	0%
Component testing	0	0%
Integration testing	0	0%
System testing	0	0%
Close Out	**0h**	**0%**
Research Paper	0	0%
Research Paper Submission	0	0%
Final Report	0	0%

Figure 1: Project Schedule

6. Research constraints

Unable of detect physical configuration

A system may include physical components that are needed to be audited. For example, a computer that handles sensitive information system must not be placed in a public area. These kind of configurations still needs to be audited manually. We can only provide checklists to audit those configurations.

Unable to audit MAC operating systems

MAC is a widely used operating system in the world. Even though auditing should be conducted on these operating systems, our research team does not have the required knowledge in configuring MAC operating systems. Therefore, this software cannot be used by organizations that use MAC devices.

7. Specified deliverables

At the end of this phase, following products can be obtained:

- Formatted set of OS commands and Queries
- Software capable of Detecting misconfiguration in a system
- A software that will require very less human interaction

8. References

[1] Prowse, D. (2010). *CompTIA Security+ Cert Guide: OS Hardening and Virtualization*. [ebook] PEARSON. Available at: http://www.pearsonitcertification.com/articles/article.aspx?p=1667482 [Accessed 12 Mar. 2019].

[2] Sanjay Garg (2014). *Network Scanning & Vulnerability Assessment with Report Generation*. [online] NIRMA University. Available at: https://www.researchgate.net/publication/263779662 [Accessed 12 Mar. 2019].