HardnBot

Intelligent Server Hardening Software

Project ID: 19_20-J01

Preliminary Progress Review Report

H.G.R Aruna shan - IT16099746

Bachelor of Science (Hons) in Information Technology

Specialization in Cyber Security Sri Lanka Institute of Information Technology

16th September 2019

HardnBot

Intelligent Server Hardening Software

Project ID: 19_20-J01

Preliminary Progress Review Report

H.G.R Aruna shan – IT16099746

Supervisor: Mr. Amila Senarathne

Bachelor of Science (Hons) in Information Technology Specialization in Cyber Security

Sri Lanka Institute of Information Technology

Sri Lanka

16th September 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.

……..

H.G.R Aruna shan(IT16099746)

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

Preliminary progress Review (PPR) report is created to give a clear idea about progress and the process of development of the research Project "**HardnBot".** and also by using this document we will explain feature vice details of the project requirements (functional, non-functional vice),how this software works, what are the novelty features of this project, what are the software requirements for best performance and for availability , what are the components we hope to cover in this project, how are we going to cover and fulfil the requirements, so on.

By using this Preliminary Progress Review anyone can review the process of work flow in the different phases of the project and check whether the project team meets requirement specifications which identified before. This document allows to identify deviation of project outcomes and also accept change in requirements.

This document is created for the research team and the supervisor to get clear idea about functional, non-functional requirements. And also, this document gives clear description about the technologies used to solve research problem which is assigned to our group. Also, this HardnBot PPR report will be guide for researchers interested in implementing this type of tool.

1.2. Scope

HardnBot is a software that has the capability to identify failed operating system compliances of a unix based servers and classify those failed compliances and use those data to apply industry recommended best practices or organizational required fixes to the unix based operating system of a server. Throughout this preliminary progress review document, we will explain how failed compliances classification process is going to achieve its goal and steps that needed to be taken.

Under this preliminary progress review document, following components are described.

a) Predict the overall risk score of the server using the classified compliance issues.

b) Display the asset value to be threaten.

HardnBot consist with four main novel components,

- Issue classification
- Risk score prediction
- Intelligent hardening
- Backup and smart rollback

Within those main components there are subcomponents/functionalities and throughout this document, flow of the predicting risk score and identify threatened asset value, these subcomponents will be discussed.

1.3. Overview

Hardenbot is a software package that has capability to detect poor/non-compliant configurations in a system (OS/DB/Application) and applying industry recommended fixes for them

This Hardening process is currently done manually with system administrators executing commands on a terminal where they have located. Automating these manual tasks, such as copy and pasting data between systems or reconciling and cross-referencing data can help expand its risk coverage and help address the ongoing compliance burden by doing more with less.
Hardenbot will be implemented to ease the entire information system hardening process. Hardenbot will include many tools to detect and correct non-compliant system configuration as well as checklist and follow up sheets to guide the hardening process of physical configuration and documentation. Fixing the non-compliant configuration will be one-click away and the user will be given a detailed report on what has detected, it's implication and the recommended fix.

Hardenbot use advance scan patterns to detect issues in system configurations. It will use formatted configuration libraries for these scans. These libraries include all configurations areas of OS and database versions. This software automatically recognizes what is OS versions and database versions. This software automatically recognizes what is OS version and database version currently installed in user environment. These libraries contain correct hardening configuration as parameter and relevant value vice, that should be in particular operating system or a database. When user run the scan, these libraries will be loaded to the system and compare the current system values with the correct configuration values. If any alternation detected, that's identify as misconfiguration.

To fix those misconfigurations we design solution libraries. Solution libraries code which need to change misconfiguration value to correct configuration value. We will design configuration libraries and solution libraries as classes. We will recorder items of classes in a manner that will enhance the scan performance.

2. Statement of Work

2.1. Background

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.

2.2. Identification and significance of the problem

When performing a hardening process, system administrators, network administrators, server custodians or outsourced expertise need to ensure security of the operating system that runs on a server (in our case its unix based servers) database and application because a single mistake can affect the whole production line which the server is in. Even though there are many scanning tools that has the capability of scan a server and identify compliance failures along with the solutions, the solutions are going to apply with a human interaction. So, the probability of mistake occurrence is higher. And manual hardening process consume much more resources such as time, human, cost likewise. As a solution for the lack of resources, organizations are tending to consider about hiring external professionals and assets to perform the hardening task. In a scenario like this, internal critical classified information might have a possible chance to expose via outsourced professionals intentionally or unintentionally and leave the server in a critical position of been hacked or data leaked. And there is a compulsory requirement of the root (administrator) access to the server terminal in order to scan and perform operating system hardening. Also, the server needs to be temporarily out of live production, because operating system hardening cannot be performed while the server is in live production environment. So, when a critical day-to-day serving server is downed for maybe more than six hours to perform operating system hardening, it will be a critical impact on the organization’s day-to-day business activities.

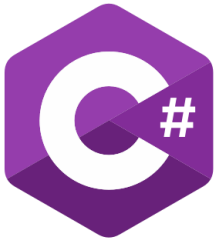
As an example, for all these above described scenarios, we can consider a operating system hardening process of a card server of a commercial bank that provide customers with all kind of credit and debit card services. Since the hardening is done by outsources professionals and a down time is required, all server's information is available to outsourced personals that include customer card details as well as the server details such as the root password and also because of the downtime required, legitimate customer services will be down. So, in a case like this it will be a critical situation to the organization. Likewise, there are more drawbacks to a manual operating system hardening process.

To solve these types of difficulties and prevent intentional and unintentional human errors, we are going to implement a software platform (HardnBot) which automate the server operating system hardening process and has the capability of detect failed compliances, classify them based on their criticality/severity levels and apply industry recommended best fixes for them via CIS benchmarks or via organizational requirements. HardnBot is also consist with an automated hardening function that harden a server according to classified (categorized) compliance issues and a rollback function that rollback to a point so that it will maintain and improve the productivity and integrity of the server.

In order to identify failed compliances, we will design a shell-based script that has the capability of executing and collecting all compliance failures in an unix system. after that there will be a classification segment where the identified failed compliances will be classified (categorized) according to a severity level which measured by the impact criticality. Later these data will used to identify the level of hardening and as a risk factor parameter for the risk score prediction algorithm.

2.3. Technical objectives

Main Programming Language: C#



Almost all operating systems support C# language. C# is a general-purpose, multi-paradigm programming language including strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented, and component-oriented programming disciplines. And, C# contains with lots of supporting third party libraries. Therefore, C# will be used as the main programming language to implement this software.

IDE: Visual Studio

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. It is mainly support for C# development and contains lots of features that improve programming experience. And because of the inbuild functions and configurations, we are going to use Microsoft visual studio as our development IDE.

Modern UI Development: Bunifu DLL

Bunifu framework is recognized for innovative interface development in C# and C++ environments. Bunifu includes a range of colors and animated controls. It also includes scripts that allows to apply animated effects into the interfaces. Therefore, we will use this framework in designing our interfaces.

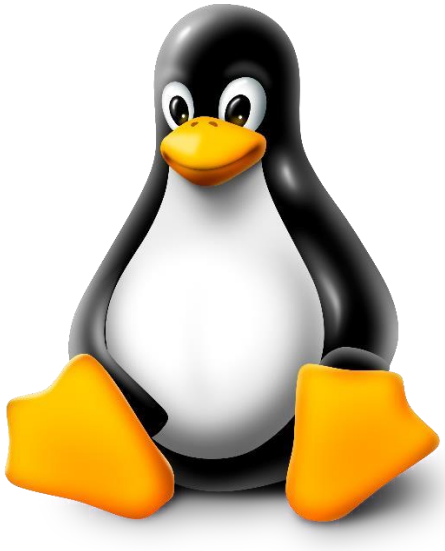
Virtual Environment: VMware Workstation

VMware Workstation is the industry standard for running multiple operating systems as virtual machines (VMs) on a single Linux or Windows PC. As the need we have to run several server-based operating systems to test our product, we will use VMware workstation for run multiple operating systems and observe the product outcomes.

Virtual Environment: Oracle VM VirtualBox

Oracle VM VirtualBox is a free and open-source hosted hypervisor for x86 visualization. In some cases where we have any difficulties to work with VMware, we will consider using Oracle VM VirtualBox as an alternative.

Operating System: Linux



Since our project is targeting linux based systems, we will implement hardening for a series of linux based operating systems listed below.

{RedHat Enterprise Linux 7 RedHat Enterprise Linux 8, CentOS 7}

Operating System: Windows

Microsoft Windows operating systems are the most popular and wide used operating systems in the world. We will use Microsoft windows operating systems to run our IDEs and implement our software. Below listed versions will be used.

{Windows 8.1, Windows 7, Windows 10 version 1903}

Programming Language: Python

Python is an interpreted, high-level, general-purpose programming language. In our research, python will be mainly used for data training and machine learning purposes. Python includes a lot of mathematical libraries and data manipulation libraries which will be useful for our data training processes. We use following inbuild and third-party libraries. Also, we will use latest stable python version at the time of developing (currently 3.7.4).

{Numpy, Scipy, Scikit-learn, Theano, TensorFlow, Keras, PyTorch, Pandas, Matplotlib, NLPs}

Automation Tool: Ansible

Ansible is an open-source software provisioning, configuration management, and applicationdeployment 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.

Live Coding platform: Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows to create and share documents that contain live code, equations, visualizations and narrative text. Using Jupyter Notebook we can perform data cleaning and transformation, numerical simulation, statistical modeling, data visualization and all machine learning tasks. To use Jupyter notebook, we will use following techniques.

{Anaconda distribution of Jupyter, Hard installed versions of Jupyter}

GPU Based Machine Learning libraries: CUDA Toolkit

The NVIDIA CUDA Toolkit provides a development environment for creating high performance GPU-accelerated applications. The toolkit includes GPU-accelerated libraries, debugging and optimization tools, a C/C++ compiler and a runtime library to deploy our application. CUDA will mainly be used for performing text recognition and for natural language processing tasks.

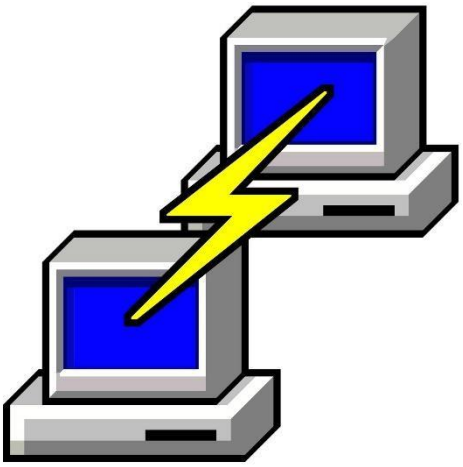
System monitoring tool: Nagios

Nagios, also known as Nagios Core, is a free and open source computer-software application that monitors systems, networks and infrastructure. Nagios offers monitoring and alerting services for servers, switches, applications and services. This tool will be used to monitor our servers for any misbehaviors after the hardening.

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.

Terminal emulator: Putty



PuTTY is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. We will use putty for testing purposes and to create connections between servers.

Hardware: External GPU installed PC

A system with an externally installed GPU will be used for machine learning purposes.

3. Research Methodology

1. **Predict the overall risk score of the server.**

   Using the classified compliance issues an equation/algorithm is developed to predict the overall risk score of the server. Using the likelihood of impact and the probability of occurrence, predict the overall risk score of the server. The scoring mechanism proceeds with a series of calculations to determine the score. For that Probability function and Bayesian model used to determine the likelihood of impact and probability of occurrence

2. **Display the asset value to be threaten**

   Consider about Information, host, servers, and telecommunication equipment, IT-services (confidentiality, integrity and availability) and identify asset and their values
   After that identify the asset value to be threatened and display these assets.

4. Test data & analysis

Gather the classified issues and using these classified issues implement the risk assessment. Through the risk assessment predict the overall risk score of the server

The risk assessment comprises the qualitative and quantitative measurement of individual risk. To calculate the risk score, we should consider about quantitative and qualitative measurements, hence the measurement of risk impact implement by using the potential risk value and probability of occurrence. The Potential risk is the product of total asset value. Potential risk can be predicted by using total asset value, vulnerability and threat. Probability of occurrence as an estimate of how often a hazardous event occurs. Asset valuation is a method of assessing the worth of the organization system assets based on its CIA security. Using these parameters calculate the risk score.

5. Anticipated benefits

HardnBot has a great commercial value because of the functionalities that it provides for users as well as organizations. Our primary goal is to automate an entire server hardening process and with the unique functionalities, HardnBot's user will gain following main benefits.

- **Fully in-depth server scanning specifically for OS compliances**
  With this functionality, HardnBot can scan thoroughly to find any compliance failures and any configuration errors.

- **Failed compliance classification and summarize After** collecting data from the scan, HardnBot's machine learning algorithms will provide a details list of compliances failures and misconfigurations with their criticality level. By using these, HardnBot will provides user with a detail percentage level of severity existence (For example: 10% Critical, 20% High, 30% Medium, 40% Low). With this information server custodians can get an overview idea about the severity level of the server.

- **Risk Score**
  By this functionality, an overall detailed risk score will be displayed to the custodian/user so that the organization can get a rough idea about server's possibility of compromise, likelihood, loss and probability of compromise.

- **Intelligent Hardening**
  HardnBot' s hardening function is a unique function because it will harden the system for an acceptance level as required by the organization. Obviously 0% risk acceptance

is not possible but in this function it's algorithms will use pre-classified compliance failure data and harden with respect to the severity levels.

- **Smart rollback functions**

  Using these functions, HardnBot will monitor applied fixes on the run as well as the server behavior parallelly and identify any misbehaviors and go back to that point where the misbehavior took place and rollback to the default or previously backed-up settings.

Besides those benefits, users will not require any down time to perform hardening because HardnBot's capability of performing stepwise hardening, stepwise error checking, stepwise backup and stepwise rollback. So, each fix will be monitored and having that, these functions will help to improve security operation center's (SOC) productivity and accuracy.

6. Project Schedule

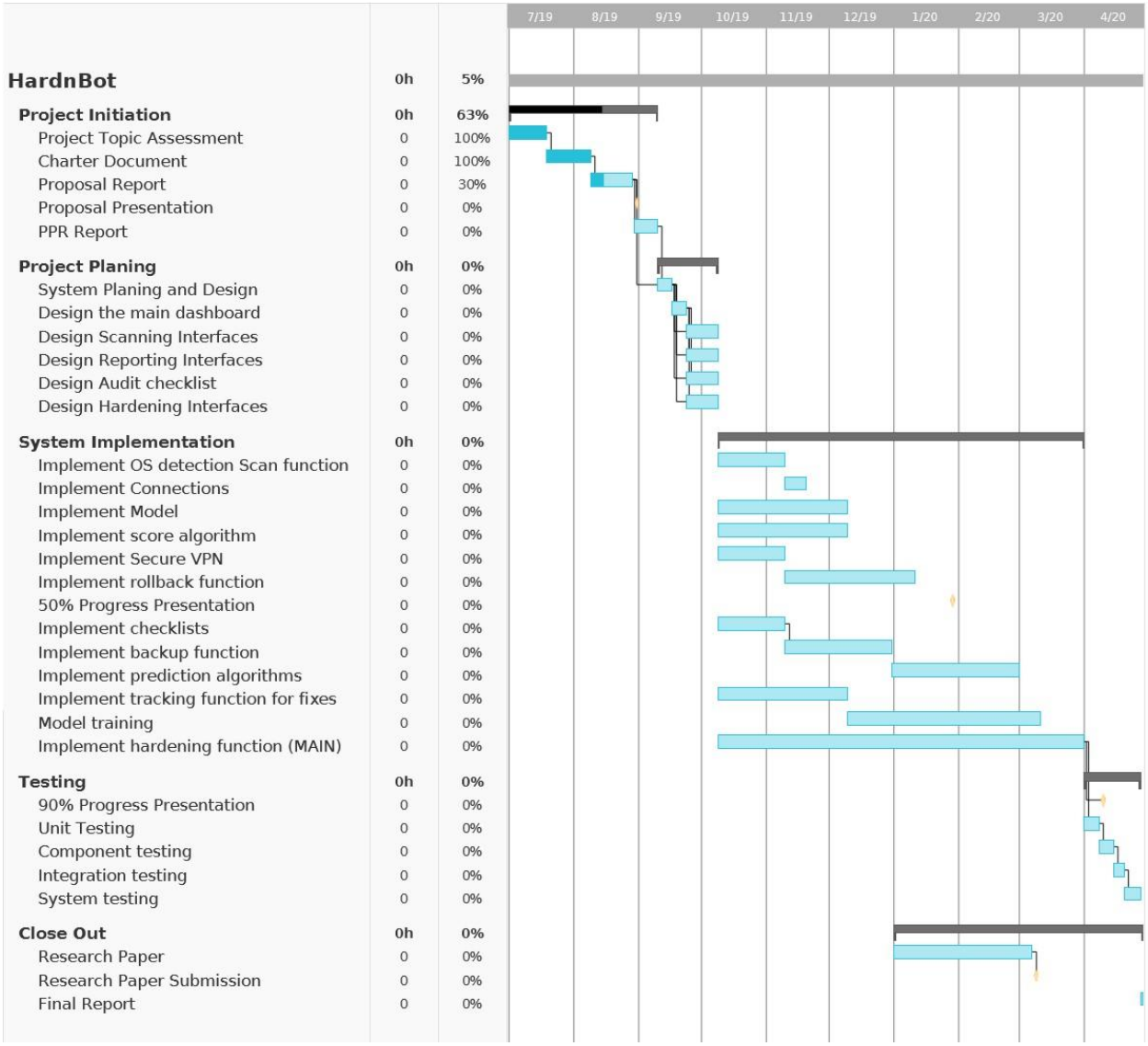


Figure 1: Project Schedule

7. Specified deliverables

- Predict the which machines are at the risk with highest accuracy achieved to date.
- Using parameters predict the overall risk score of the server.
- Display the asset value to be threaten.

8. References

[1] Sheung Yin Kevin Mo, Peter A. Beling, Kenneth G. Crowther, "Quantitative Assessment of Cyber Security Risk using Bayesian," in *IEEE*, 2009.

[2] Venkatesh Jaganathan, Priyesh Cherurveettil, Premapriya Muthu Sivashanmugam, "Using a Prediction Model to Manage Cyber Security Threats," in ResearchGate, 2015.

[3] Kai Liu, Yun Zhou, Qingyong Wang, Xianqiang Zhu,, "Vulnerability Severity Prediction With Deep Neural," in *IEEE*, 2019.

[4] Qian Yu, Yongjun Shen, "Research of Information Security Risk Prediction," in *IEEE*, 2016.

[5] I. V. Anikin, "Information Security Risk Assessment and," in *IEEE*, 2015.