# COMMON CRITERIA BASED OS COMPLIANCE FRAMEWORK FOR WINDOWS 10


By

Rabiya Farooq

A thesis submitted to the faculty of Information Security Department, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of MS in Information Security

JUNE 2021

# THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by **Ms. Rabiya Farooq**, Registration No. **00000206743**, of **Military College of Signals** has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations/MS Policy, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members and local evaluators of the scholar have also been incorporated in the said thesis.

Signature:

Name of Supervisor **Assoc Prof Dr. Haider Abbas**

Date: _____

Signature (HOD): _____

Date: _____

Signature (Dean/Principal) _____

Date: _____

# Declaration

I hereby declare that no portion of the work “Common Criteria Based OS Compliance Framework for Windows 10” exhibited in this thesis has been submitted in provision of any other award or educational qualification, either at this institution or anywhere else.

# Dedication

“In the name of Allah, the most Benevolent, the most Merciful"

I dedicate this thesis to,
my parents, who have been a source of unwavering support, prayers and love throughout my work,
my supervisor and faculty, who have always been a beacon of professional guidance and insight, whenever I needed any.

# Abstract

This era of global inter-connectivity has made the working community utterly reliant on computer systems for their operations. Such dependence has led to an increased number of cyberattacks that adversely impact the business objectives of organizations and single users alike. Internationally recognized standards such as Common Criteria (CC), NIST SP 800-53 and ISO 27001-2 provide guidelines for the security of IT products. These standards can also be applied to assess security functionality of Operating Systems (OS) that act as the last defensive layer in case of cyberattacks. Considering this, computer system users must adopt a reliable strategy for analyzing their OS's security potency. The already existing methods to achieve this purpose are either not reliable or are complex and expensive for application by every organization or single user. Hence, we have used an integrated and systematic approach to propose two flexible and cost-effective Security Compliance Evaluation (SCE) frameworks that perform tests to evaluate Windows 10 and Linux Ubuntu 20.04 OSs in the light of internationally recognized security guidelines. The frameworks so formulated can be easily adopted by any user and incorporates the use of scoring system for each aspect of cybersecurity in order to compute percentage compliance of the evaluated PC. Validation has been done on a personal computer at home for both the frameworks and on a system in a security research lab for only Windows 10 framework to demonstrate the efficacy of correct security policy implementation on the extent of compliance of the OS. Lastly, an operating system security policy has been proposed which can be adopted by organizations or single users to ensure their compliance with NIST SP 800-53, ISO 27001-2 and Common Criteria along with extended packages for VPN, WLAN and SSH for broader aspect of security.

# Acknowledgments

All praise belongs to Allah for His blessing in successful completion of this work, and only the faults have been mine.

I would like to convey my deepest gratitude to my supervisor, Assoc Prof. Haider Abbas, PhD, for his mentorship and constant professional guidance. From his irreplaceable help, constructive criticism and propositions to imbuing a sense of confidence in my approach and work, his support has been instrumental in making my thesis in the shape it is today. Also, I would like to acknowledge my committee members; Asst Prof. Mian Muhammad Waseem Iqbal and Maj. Sohaib Khan Niazi for sharing their valuable opinion and knowledge on certain facets of my work.

Lastly, I would like to thank my parents and husband for their constant love and support. Their resolute belief in my vision and eventual success carried me through the toughest times during my endeavor.

# Table of Contents

# LIST OF FIGURES

# LIST OF TABLES

# ACRONYMS

| | |
|---|---|
| OS | Operating System |
| CC | Common Criteria |
| NIST | National Institute of Standards and Technology |
| ISO | International Organization for Standardization |
| IEC | International Electrotechnical Commission |
| PP | Protection Profile |
| GPOSPP | General Purpose Operating System Protection Profile |
| EP | Extended Package |
| WLAN | Wireless Local Area Network |
| VPN | Virtual Private Network |
| SSH | Secure Shell Protocol |
| SFRs | Security Functional Requirements |
| UEFI | Unified Extensible Firmware Interface |
| BIOS | Basic Input/output System |
| IT | Information Technology |
| SARs | Security Assurance Requirements |
| TOE | Target of Evaluation |
| ST | Security Target |

*Chapter 1*

# 1 INTRODUCTION

## 1.1 Overview

In this age of rapidly transforming technology, our lives have become more dependent on computer systems than ever before. Their application touches all aspects of modern life from routine communication to the productivity of the everyday workplace. Our dependence and reliance on such systems has resulted in us trusting these with our sensitive personal information which demands privacy. The use of these systems is diverse: from laptops and smartphones to embedded devices in automobiles and automated manufacturing systems. Resultantly, this makes computer systems an extremely attractive target for attackers with malicious intent [1].

These days everything is technology driven to the extent that companies excessively rely on computer systems to carry out their business objectives, which makes their security management a challenging task [2]. The Operating System (OS) is a core program that performs a computer's basic functions connecting the hardware to the software and provides support to the installed applications. Therefore, any vulnerability in running applications will be handled by underlying OS as a last defensive layer and any vulnerability in the OS will give rise to more weaknesses in the computer system. Considering this critical role of OS in any computing environment, it is safe to say that its security is of paramount importance and thus needs to be ensured. In any given systems-based work environment, the OS has to deal with complex applications routinely. Therefore, it must cope up with an increasing number of software bugs, malicious attacks and hardware failures for smooth functioning of businesses. A trusted OS must provide reliability and efficiently support and address the security issues of the computer system [3] [4].

The evolution in communication standards coupled with the resulting revolution in transmission speeds has given rise to a new yet lethal sector of criminal activity in the form of cybercrimes. NETSCOUT Threat Intelligence Report 2019 states that about 8.4 million DDoS attacks were observed on computer systems in the entire year [5]. According to the Annual Cybercrime Report 2019, every 14 seconds a business falls victim to ransomware attacks and this is likely to increase to every 11 seconds by the end of 2021 [6]. Not only has the rate of cybercrime gone up but it has increased in lethality as well: in 2020 the

average ransomware demands spiked up to $1.3 million from $800,000 the previous year [7]. Cyberattacks seem to affect everyone and all organizations alike regardless of their type or size. The global number of cybersecurity incidents in 2019 reached a staggering 32,002 in numbers, among which professional services and the public sector were the most targeted with 7,463 and 6,843 reported incidents respectively [8]. Today many businesses rely on the built-in security of OSs for various operation critical applications and sensitive data handling. In 2021 alone, 58% of the companies were reported to have over 1,000 inactive user accounts which could be utilized by the attackers to gain unauthorized access to networks [9]. Strikingly, about 60% of the breaches in cyber secure environments occur due to negligence in installing security patches on time [10]. However, all of these misconfigurations can be identified and overcome by carrying out a detailed evaluation of the security components of any OS. Also, COVID-19 has greatly affected, and to an extent disturbed, the dynamics of the modern workplace. Many companies have shifted from office based to home based working models which have significantly increased the attack surface. This upsurge is attributed to increased use of comparatively less cyber secure environments such as the rampant use of VPNs, insecure internet connections and remote desktop services. Since the start of the pandemic, cybercrime has seen an increase of 600% [11] and this is likely to increase even further if proper security of OSs is not ensured. Therefore, the security evaluation of an OS has become absolutely essential for the organization's safety and success. One can argue that the importance of cyber and digital security is at par, if not more than the physical security of data. Any lapse in the standard of security provided by the OS could lead to critical vulnerabilities resulting in a myriad form of cyberattacks.

Although the more recent OSs provide robust security there is a need to ensure whether these security features are properly utilized or not. Cybercriminals are now at an advantage to exploit those who lack basic security and are, therefore, more susceptible to attacks. Given the monetary, moral and social effects cybercrimes have, it is imperative to study and evaluate the security facets and features of OSs. Therefore, we can understand that there is a dire need to develop a framework which can not only help in security evaluation of OSs but also be easily adoptable by organizations as well as single users without the need of exorbitant expenditure.

## 1.2 General Trend of Windows Operating Systems in Global Market

Over the years Windows OS has emerged as the market leader for commercial and domestic use of computer OSs. According to StatCounter in February 2021, Windows OS accounted for 32.34% of the total market share for OSs worldwide [12] whereas it accounted for 75.89% of the global market share for desktop OSs [13] as shown in Figure 1.1. In Pakistan alone 81.65% of desktops have Windows OS as the primary operating system installed [14] and Windows 10 in specific has a market share of 77.65% among all other versions of Windows's OS [15]. In addition to its popularity for home computers, Windows OS also has a wide spectrum of commercial applications due to its enhanced built-in security suite [16]. Considering this, we have designed our framework for the security evaluation of Windows 10 OS used both domestically and commercially.

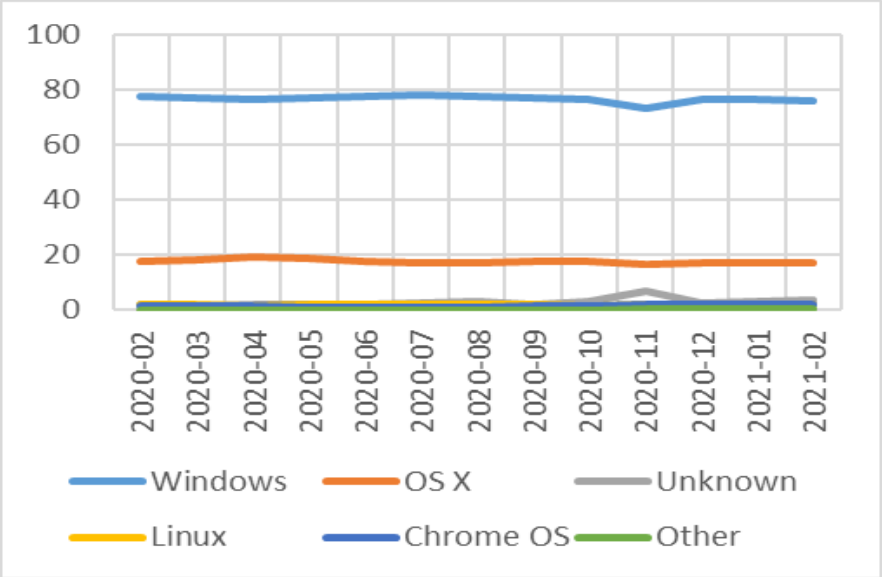


**Figure 1. 1 Desktop OS Market Share Worldwide (Feb 2020 - Feb 2021)**

## 1.3 Security of Windows 10 Operating Systems

Windows 10 OS offers many new security features to cope with the ever-expanding threat landscape. These features make it a top choice of many enterprises and home users to protect themselves against cyber-attacks. Some of the prominent security features provided by Windows 10 OS are discussed in Table 1.1. They mainly help provide protection against threat exposure, rootkits, zero-day attacks, ransomware, unauthorized installation of applications, unauthorized disclosure of information, accidental configuration changes, phishing and other malware attacks [17].

**Table 1. 1 Prominent Security Features of Windows 10**

| Security Feature | Description |
|---|---|
| Windows Update | It offers automatic security updates and fix past bugs by introducing new functions. |
| Windows Defender Antivirus (WDA) | It comes with built-in firewall and automatically scans newly downloaded files for viruses as a part of real-time protection defense posture. |
| Microsoft Smart Screen | It scans and block the execution of known malicious programs and notify the users regarding suspicious mails and websites. |
| Windows Defender Application Guard | It provides protection against advanced threats by designating whitelists (containing trusted websites) or opening untrusted websites in an isolated container with no connections to the corporate network or other sensitive resources. |
| Windows Sandbox | It provides flexibility to administrators for considerable freedom concerning application permissions as it opens new apps in an isolated virtual environment to prevent threat exposure. |
| Windows Defender Device Guard | It provides protection to kernel processes and drivers from zero-day attacks by enabling the mode where OS will only trust administrator authorized apps and lock the device if code integrity is violated. |
| Windows Credential Guard | It isolates secrets and allow only privileged system software to access them by using virtualization-based security. |
| Windows Defender Exploit Guard | It is designed to perform network protection, controlled folder access, block low integrity images, block untrusted fonts and address filtering etc. |
| Secure Boot | It safeguards UEFI/BIOS by making sure any code that runs immediately after the start of the OS is signed by Microsoft or the hardware maker. It helps prevent malware installations that are hardware based. |
| Microsoft Defender Advanced Threat Protection | It detects any kind of suspicious behavior by monitoring endpoints via behavioral sensors and cloud-based analytics. |

| | |
|---|---|
| User Account Control | It provides protection against unauthorized changes by asking for an administrator level permission in case of any important change like removing or installing an application |
| Windows Hello | It is a platform providing multifactor authentication (like facial recognition or fingerprints) that can pair biometric data with companion devices (i.e., smartphones, smart watches etc.) for ensuring only authorized access to the computer. |
| Find My Device | It can locate the stolen device with the help of internet and even lock it down. |
| BitLocker | It can encrypt the entire drive with standard XTS-AES encryption scheme without affecting the system performance. |

However, most of these security features are often not fully utilized by users, which results in vulnerabilities on the OS. The AV-Test Security Report 2020 suggests that approximately 83.45% of all malware attacks targeted the Windows OS [18] [19]. And that 91% of Windows computers were targeted by ransomware in 2020 according to Statista [20] [21] as shown in Figure 1.2. Therefore, we can understand that due to their popularity and dominance among computer systems, Windows OSs are susceptible to a host of different attacks and vulnerabilities being developed every day. And these attacks can only be avoided by performing thorough security evaluation of OSs against internationally accepted best security practices.

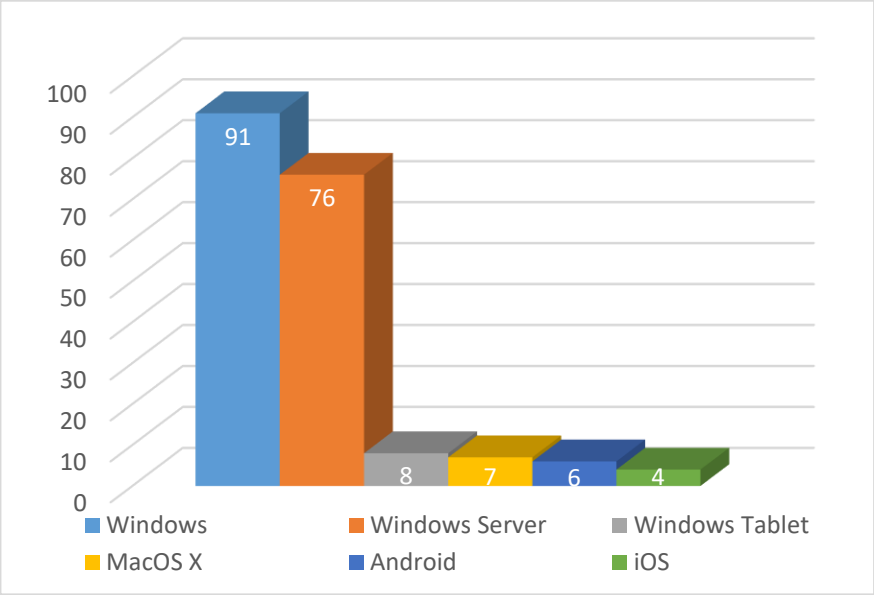


**Figure 1. 2 Major Operating Systems Targeted by Ransomware**

## 1.4 Problem Statement

In today's world computer systems are used by everyone regardless of being a single user or part of multi-million industry. This dependence on technology has increased the use of various software applications for carrying out routine tasks. These applications might have different vulnerabilities that can be malicious for the system, therefore, to ensure the overall security and stability of computer systems, the underlying OS must be secure enough to tackle with any malicious application. Moreover, complex network topologies and excessive use of internet also poses many threats which can be contained to some extent by ensuring the security of OS installed. Although many recent OSs provide robust security features to deal with the ever-growing cyber-threat landscape but many a times these security features are not properly utilized or configured by the users thus making the OS vulnerable to cyberattacks. This demands a need to evaluate OS against some standard guidelines that are universally accepted for the security of OSs.

Many international standards exist, that provide extensive guidelines for the security assessment of IT products. Among them, requirements of Common Criteria (CC), ISO 27001-2 and NIST SP 800-53 are best suited for evaluation of operating systems. There are many accredited labs that work independently to evaluate security features of OS against CC provided guidelines. These labs follow a certain procedure that requires the source code and involvement of developer to rigorously test the security of OS, however this process is quite expensive and time consuming, therefore cannot be adopted by organizations that lack significant budget to spend on security. Moreover, Microsoft has already achieved CC certification for Windows 10 OS against following Protection Profiles (PPs):

- General Purpose Operating System Protection Profile (GPOSPP).
- General Purpose Operating Systems Protection Profile Extended Package for Wireless Local Area Network Clients (WLAN). And,
- General Purpose Operating Systems Protection Profile Extended Module for Virtual Private Network Clients (VPN).

Therefore, such an extensive method for evaluation is not required to be performed again to assess the security configuration of Windows 10 OS in its operational environment. Considering this we saw the need to develop a methodology, which will evaluate the security of OS against internationally accepted guidelines of CC, NIST SP 800-53 and ISO

27001-2. And also, is flexible, easy and cost effective to be adopted by any type of organization for evaluation of its indigenous OS without having to spend any money.

## 1.5 Research Objective

There are two main objectives of our research which we deem necessary to achieve for solving the above-mentioned security problem.

- To perform critical analysis of existing evaluation techniques for OS.
- To propose and develop a framework for evaluation of Windows 10 OS based on internationally recognized Common Criteria (CC) standard.

## 1.6 Additional Contribution

Initially the scope of our thesis was limited to designing an evaluation framework for Windows 10 OS against CC security functional requirements (SFRs). However, during subsequent phases of our research, we came to realize that only CC SFRs for GPOSPP is not good enough to rely on for OS security, especially in case of organizations who deal with critical data routinely and use remote access and other network services more frequently than the normal user. Therefore, in order to facilitate their security needs we consider it necessary to also include SFRs of the following three CC Extended Packages for GPOSPP in the security criteria defined by our framework:

- Extended Package for Wireless Local Area Network (EP-WLAN) Clients
- Extended Package for Secure Shell Protocol (EP-SSH)
- Extended Module for Virtual Private Network (EP-VPN) Clients

NIST SP 800-53 is another international standard that provides controls and guidelines to ensure confidentiality, integrity and availability of critical data stored on computer systems. Moreover ISO 27001 provides best code practices to plan, develop, operate and maintain an Information Security Management System (ISMS) which also includes security of OSs used in an organization. Hence, in addition to CC SFRs we have also incorporated some of the NIST SP 800-53 and ISO 27001 controls related to security of OSs to diversify the security criteria used by our framework for evaluation.

Linux is the third most popular desktop OS and has small yet stable share of 1.97% of the total global market [13] and a share of 1.23% of Pakistan’s market [14]. Therefore, we have also designed a second framework for evaluation of Linux Ubuntu OS against the security requirements of only NIST SP 800-53.

In the end we have proposed an OS Security Policy based on the best practices adopted by CC, NIST and ISO standards for the organizations to adopt for secure operation of their OSs.

## 1.7 Scope of Research

The research applies to Windows 10 and Linux Ubuntu OS used in both domestic as well as commercial environment for security compliance evaluation. Proposed frameworks can be used by any user to assess their security with respect to any security class/domain like networks, cryptography, user accounts, auditing, identification and authentication etc. and as a result allow them to improve their security infrastructure according to internationally accepted criteria.

## 1.8 Significance of Research

The research has opened new doors for the security compliance evaluation of Windows 10 OS by incorporating security criteria derived from three universally adopted standards along with the security of additional components provided by OSs i.e., WLAN, VPN and SSH, which allows for more thorough security evaluation. This framework is easy to adopt and will help Windows 10 users to:

- Strengthen their OS security according to internationally accepted criteria without spending any money.
- Evaluate and maintain the security of their indigenous OS to safeguard it against various attacks.
- Prepare themselves before applying for assessment by Common Criteria Testing Labs (CCTL) for OS security.

The proposed OS security policy for organizations if implemented correctly will strengthen the security of OS to avoid cyberattacks and will help them to attain compliance with NIST SP 800-53, ISO 27001 and CC along with its three extended packages for WLAN, VPN and SSH.

## 1.9 Research Methodology

The research explores different techniques for security evaluation of an OS and proposes a framework that uses an integrated and systematic approach to evaluate Windows 10 OS against SFRs provided by NIST SP 800-53, ISO 27001-2, CC GPOSPP and extended

packages like EP-VPN, EP-WLAN and EP-SSH. The framework further classifies these SFRs into different security domains according to their area of concern and use test commands to check compliance of each SFR. Considering the security needs and operational environment of different users or organizations, our framework assigns security weights and levels to each individual SR according to their critical role in attaining operational objectives of organizations and further incorporates them in the calculation of compliance score. This is because some requirements will be more critical than others for certain prospective users. In this way the framework not only calculates the percentage compliance of a given OS but also considers the security needs of that particular user. And thus, a wholesome picture of the essential SRs is given which clearly enunciates the efficacy of a security suite in light of existing recognized standards.

## 1.10 Thesis Outline

The thesis document is organized into eight chapters. Chapter 1 is brief introduction of proposed security compliance evaluation framework. Chapter 2 covers the background of evaluation standards and literature review of existing evaluation techniques for OSs. Chapter 3 gives description of proposed security compliance evaluation framework for Windows 10. Chapter 4 validates the proposed evaluation framework for Windows 10. Chapter 5 presents the security complaince evaluation framework for Linux Ubuntu against NIST SP 800-53 security guidelines. Chapter 6 validates the Linux evaluation framework. Chapter 7 presents an operating system security policy for organizations and Chapter 8 concludes the research with future recommendations.

# 2 BACKGROUND AND LITERATURE REVIEW

Many international standards exist that guide organizations in ensuring safety and security of their critical infrastructure including information systems. These standards include ISO/IEC 27001, ISO/IEC 27002, ISO/IEC 15408 (Common Criteria), ISO/IEC 27033, NIST SP 800-53 and NIST SP 800-123 etc. for information security management systems, network security, general server security and security of IT products and components of information systems. However, the following two standards are considered as best practices for evaluation of operating systems.

## 2.1 NIST SP 800-53

It is a framework that provides controls and procedures which can be adapted by organizations to strengthen the security, by maintaining confidentiality, integrity and availability of their critical information systems. Any component of computer systems that stores, processes and transmits critical information can apply these set of guidelines. By law, all U.S. federal government agencies are required to comply with this standard [22]. NIST SP 800-53 contains 18 control families and 3 categories of baseline security controls i.e., low, medium & high [23] [24]. These baselines controls highlight operational and functional needs to cater for most common threats faced by information systems. The control families are described in Table 2.1.

**Table 2. 1 NIST SP 800-53 Control Families**

| SR | Control Families | No. of controls | Description |
|---|---|---|---|
| 1 | AC: Access Control | 25 | Deals with user login and account configurations. |
| 2 | IA: Identification and Authentication | 11 | Deals with authentication of user and accounts, passwords and security identifiers. |
| 3 | CP: Contingency Planning | 13 | Deals with training, planning and testing of contingency plans and backups. |
| 4 | CA: Security Assessment and Authorization | 9 | Deals with authorization of connections and security certificates. |

| | | | |
|---|---|---|---|
| 5 | SI: System and Information Integrity | 17 | Deals with flaw remediation, security patches, spam protection and integrity validation of software. |
| 6 | SC: System and Communication Protection | 44 | Deals with isolation, segmentation, DNS, firewalls and cryptography related controls. |
| 7 | CM: Configuration Management | 11 | Deals with controls regarding configuration settings, access restrictions and software usage restrictions. |
| 8 | AU: Audit and Accountability | 16 | Deals with controls about event logging and protection mechanisms put around them. |
| 9 | AT: Awareness and Training | 5 | Deals with security training and cybersecurity awareness related controls. |
| 10 | IR: Incident Response | 10 | Deals with controls related to incident monitoring, reporting and handling along with response training and testing. |
| 11 | MA: Maintenance | 6 | Deals with maintenance of security components, tools and personnel. |
| 12 | MP: Media Protection | 8 | Deals with media access, use, storage and transportation controls. |
| 13 | PS: Personnel Security | 8 | Deals with personnel security, training, screening, termination and transfer controls. |
| 14 | PE: Physical and Environmental Protection | 20 | Deals with physical access controls and authorizations along with temperature, fire and power outage protection. |
| 15 | PL: Planning | 9 | Deals with system and security plans of an organization. |
| 16 | PM: Program Management | 16 | Deals with system inventory and risk management strategy. |
| 17 | RA: Risk Assessment | 6 | Deals with vulnerability scanning and risk assessment methodology. |
| 18 | SA: System and Services Acquisition | 22 | Deals with installed software, allocation of resources, tampering detection and acquisition process controls. |

## 2.2 Common Criteria

It consists of internationally accepted security guidelines for evaluation of Information Technology (IT) products. It allows users to specify Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) in the form of Protection Profiles (PPs) and provides a way for vendors to describe security features of their products in the form of Security Targets (ST). The designed IT products are then evaluated by an

accredited third-party lab that tests all the acclaimed security attributes of the product in detail and assigns certification, if all the requirements are satisfied by Target of Evaluation (TOE). Common Criteria consists of many PPs for evaluation of different IT products like there is a separate PP for evaluation of general-purpose operating systems (GPOSPP) security [25]. In addition to PPs, there also exist a number of extended packages that describes additional security requirements of components present in a general protection profile. For the evaluation of OS in our framework, we have considered the following CC extended packages of GPOSPP:

- **Extended Package for Secure Shell (EP-SSH):** It describes extended security requirements of SSH protocol that is used to ensure security of remote login and other network services over an untrusted network [26].
- **Extended Package for WLAN (EP-WLAN) Clients:** It describes the additional security requirements for built-in Wireless Local Area Network (WLAN) clients and ensures protection of data on a wireless network [27].
- **Extended Module for VPN (EP-VPN) Clients:** It provides the additional security criteria for built-in VPN clients using IPsec and IKE secure protocols [28].

Considering the significant increase in cyber threat landscape and dependence on the security of computer systems, the guidelines provided by NIST SP 800-53, and CC are considered best practices to provide reliability and security of the operating systems. The inclusion of extended packages in the security criteria will help provide a broader picture for ensuring security to tackle with ever-growing cyberattacks.

### 2.3 ISO/IEC 27001

It is an international standard that has been developed in collaboration with International Electrotechnical Commission (IEC) and provides controls regarding planning, development, operation, maintenance and auditing of Information Security Management Systems (ISMS) in organizations. It helps organizations in risk assessment of their assets and help them maintain security in an organized manner [29] [30].

### 2.4 Related Work

In addition to security standards, we came across a few techniques to evaluate operating systems either through some standard or self-assessment of the performance and security features provided by them. The analysis of these papers is shown in Table 2.2. Daniel et al. [31] proposed a Security Requirements Engineering Process (SREP) based on

guidelines of Common Criteria that incorporated security requirements of CC at the early development stage of software and used security resources repository to support reuse of these security requirements, assets, threats and countermeasures. This is an iterative process that adds different CC components throughout the lifecycle of software according to each activity phase. The end product will be a CC compliant software containing all the necessary security requirements for compliance. However, this approach is feasible only during the development of an operating system and, therefore, cannot be used for the evaluation of already developed OSs. Moreover, it does not include security requirements of extended packages for more detailed and thorough security compliance of OSs to cope with changing threat environments.

Another research evaluated Ubuntu 20.04 LTS, Linux Mint 19.3 Tricia and Pop! _OS 20.04 OSs on the basis of their performance [32]. They used benchmark tools namely Hardinfo, Geekbench and Phoronix Test Suite for assessing quality of CPU, RAM, SSD and GPU respectively. All testing was done in idle state i.e., all other applications were uninstalled, and all user files were deleted from OS, which is not a practical approach as the real performance and security will get affected by quality of installed applications and amount of stored data. In the end three main metrics namely speed, time and points that are achieved during a certain task are used for overall performance evaluation of considered OSs. In [33] operating system security was evaluated by enlisting all the capabilities and mechanisms that are mostly implemented to increase security in OS. They classified these security mechanisms according to the security parameter structure of CC, which is independent of the nature of any operating system. Three security levels i.e., low, medium and high are assigned to each mechanism. The OS is then compared against these categorized features and is allotted a corresponding security level. The security features used in this approach for evaluation are not chosen according to any international standard, rather they are the collection of already implemented features in most OSs. Consequently, the extent of security provided by this method cannot be completely trusted.

Vulnerability analysis is one of the widely adopted methods for security evaluation of computer OSs. In a research Movahedi et al. [34] did vulnerability assessment of commonly used OSs by using a clustering technique. In this method all vulnerabilities pertaining to a particular OS are grouped into multiple clusters based on their description present in Common Vulnerabilities and Exposures (CVE) database. Afterwards, Software Reliability Models (SRMs) were applied on each cluster to predict new vulnerabilities for

each OS that are probable to be discovered in the future. Alenezi et al. [35] did security analysis of Windows, Linux and Mac OS by foretelling their vulnerabilities using machine learning approach. The method used CVSS, CVE database and NVDR to extract data pertaining to vulnerabilities and introduced additional variables calculated from this data for more effective analysis. Besides, OS severity levels were predicted by using different machine learning algorithms like Logistic Regression, K nearest neighbors, Gaussian Naive Bayes, Random Forest, and Adaptive Boosting ML algorithms. In another paper Nawa Raj et al [36] used time series approach to develop an analytical model for prediction of new vulnerabilities based on linear and nonlinear techniques. This approach used reported vulnerabilities in NVD to predict future vulnerabilities of Windows 7, Mac OS X and Linux kernel. Even though these methods are reliable for predicting future vulnerability tendencies of different OSs however they are not a reliable means for security evaluation of OSs in their operational environment. For that, the user's take on security and the seriousness of its employees must be kept in mind. Moreover, for effective evaluation one must follow some internationally accepted guidelines.

An application for security evaluation of Android OS was developed by Khokhlov et al. [37] that analyzed different system parameters and then assigned an overall security level to OS based on this analysis. The application evaluated parameters like android OS version, screen lock, permission to install applications from unknown sources, potentially harmful applications, developer option menu, basic integrity test and android capability test. In the end output from these parameters was merely added to calculate security score for tested Android OS. The result could vary from 0 to 7 with "7" being the highest level of security and "0" being the lowest. Another paper, [38] also did evaluation of data security for Android OS and calculated corresponding security score by measuring several security parameters including root access, unlocked bootloader, device lock, device model, android OS version, installed security patch, unknown sources, installed applications, developer menu, device rating, installed application rating and system vulnerabilities. Andrea et al. [39] shared their experience of certifying FIN.X RTOS Linux operating system against CC (EAL 4+) evaluation. They used LTP (Linux Test Project) test suite and some derived test cases from Red Hat 5 EAL 4+ certification test suite to verify acclaimed SFRs. Considering this approach, we have developed a user-friendly test suite to evaluate Windows 10 OS against security criteria of NIST SP 800-53 and CC GPOSPP along with extended packages for SSH, WLAN and VPN. Moreover, this test suite can be

applied by any user regardless of his expertise and calculates a comprehensive compliance score.

### 2.5 Common Methodology for IT Security Evaluation (CEM)

It is a document that supports CC standard and contains the methodology and actions that guide the evaluators to conduct CC evaluation of IT products. Each Security Functional Requirement (SFR) present in CC is evaluated using CEM work units and then corresponding verdict (pass or fail) is issued [40]. CEM comprises of four evaluation tasks described as follows:

- **Input Task:** It deals with obtaining supporting documents like ST, guidance document, document containing tests performed by developer, detailed design document of IT product containing TOE background and the source code. And evaluating them as a part of detailed security evaluation of TOE.
- **Evaluation Sub-activities:** In this task different tests are performed in the form of CEM work units to evaluate correct working of security functions as claimed by a given ST or PP.
- **Output Task:** It deals with written correspondence in the form of Observation Reports (OR) and Evaluation Technical Report (ETR). The ORs are written in case of any ambiguity faced during evaluation of IT product to ask for clarification and more supporting documents to ease the evaluation process whereas ETR is the final report written to justify the verdict for security evaluation and achievement of CC certification.
- **Technical Competence to Evaluation Authority Task:** This is the last task in CEM evaluation methodology where the ETR is reviewed for justification of verdict and finally the authority issues CC certificate to the evaluated IT product.

CEM is widely followed by CC accredited labs for the certification of OSs. However, this method is very extensive and demands a certain level of expertise to perform evaluation tasks. This inspires us to develop a framework which provides users with an evaluation scheme that can be easily adopted to evaluate the security of indigenous OS in order to avoid cyberattacks.

**Table 2. 2 Analysis of Existing Literature**

| Paper | Operating Systems | Security Standards | Contribution |
|---|---|---|---|
| [31] | All | Common Criteria | Developed a CC centered Security Requirements Engineering Process (SREP) that incorporates CC security requirements at the early stages of the software development. |
| [32] | Ubuntu, Linux Mint and Pop!_OS | None | Used benchmark tools for performance evaluation of operating systems by assessing capability of CPU, RAM, SSD and GPU. |
| [33] | All | None | Evaluated OS against a list of security mechanisms mostly used to enhance operating system security. |
| [34] | Windows, MAC, IOS and Linux | None | Used clustering technique for the vulnerability assessment of all operating systems. |
| [35] | Windows, Linus and MAC | None | Used machine learning algorithms like Logistic Regression, K nearest neighbors, Gaussian Naive Bayes, Random Forest, and Adaptive Boosting ML for predicting vulnerability security level of OSs. |
| [36] | Windows 7, Mac X and Linux Kernel | None | Used time series analysis to design a model based on linear and nonlinear approaches to predict future vulnerabilities. |
| [37] | Android | None | Developed an application that evaluates Android OS security by analyzing important system parameters. |
| [38] | Android | None | Evaluated data security of Android OS by performing tests to measure defined security metrics. |
| [39] | Linux | Common Criteria | Used LTP and Red Hat 5 certification test suite for conformance of FIN.X RTOS Linux operating system against Common Criteria to assurance level of EAL 4+. |
| [40] | All | Common Criteria | A detailed document containing the methodology to carry out evaluation process of IT security products according to CC requirements. |

## 2.6 Conclusion

The increase in cyber threat landscape and applications of computer systems have made security evaluation of OSs a necessity to avoid cyberattacks. These evaluations cannot rely on vulnerability analysis alone, rather must be carried out according to universally accepted standards to provide significant assurance of security and protection against breaches and malicious attacks. The existing techniques on evaluation of OS security are either too weak or complex to be implemented easily by any type of user. Few of the techniques also rely on the use of benchmark tools, hence the security evaluation through this method will be jeopardized if any of the tool is compromised or lose their authenticity. Thus, we need a more reliable method for security evaluation of operating systems that does not involve any third-party application. Rather use a strong and reliable security baseline for its assessment.

*Chapter 3*

# 3 PROPOSED WINDOWS 10 SCE FRAMEWORK

## 3.1 Introduction

To cater for the shortcomings, we found during literature review and to provide a more systematic and reliable approach for security evaluation of operating systems, we have developed a methodology that is easy and flexible to adopt by every user and does not require any advance human skillset or external tool. Moreover, this method defines baseline security criteria derived from internationally accepted standards that provides assurance in the quality of thorough security evaluation carried out during the process. Using this approach, we have proposed two evaluation frameworks for checking security compliance of operating systems as listed below:

- Security compliance evaluation framework for Windows 10, and
- Security compliance evaluation framework for Linux Ubuntu

Microsoft has already certified Windows 10 OS against the CC protection profile for general purpose operating systems (GPOS PP), including the EP for WLAN and VPN clients [41]. However, the OS security still needs to be verified during its use at home or in organizations, due to the fact that mostly the built-in security features of OSs are not properly utilized by the user, or he is simply unaware of the security misconfigurations present on their computer system. Therefore, for protection of Windows 10 users from unauthorized access and disclosure of their critical information we have designed a security compliance evaluation (SCE) framework that caters for all the security concerns of users and not only assess the Windows 10 OS security according to internationally accepted criteria but can also be applied by any user irrespective of size or nature of work. Important steps of proposed Windows 10 SCE framework are exhibited in Figure 3.1 and are explained in detail in rest of the chapter. The general flow of the evaluation steps is shown in Figure 3.2.

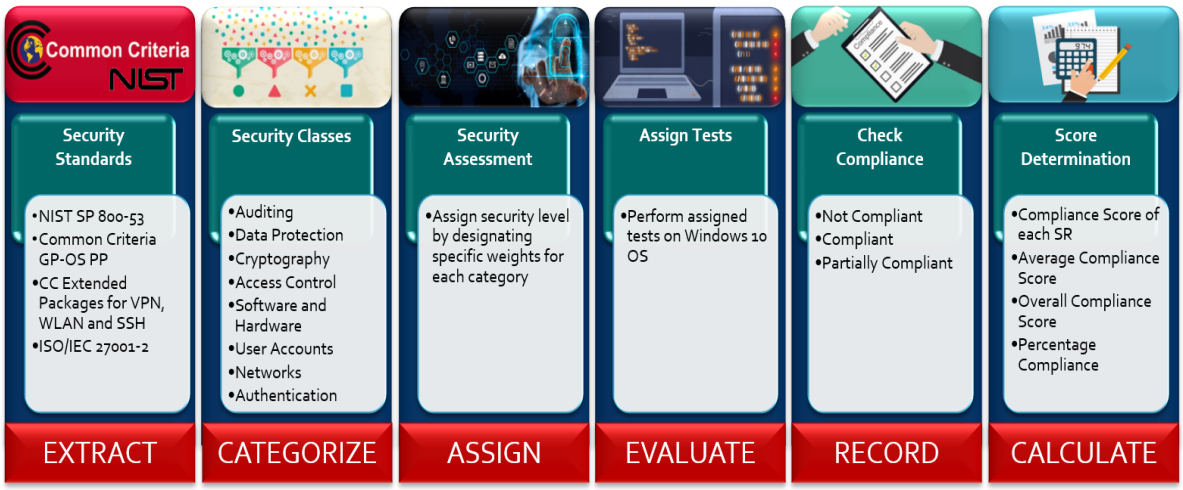


**Figure 3. 2 Proposed SCE Framework for Windows 10 OS**

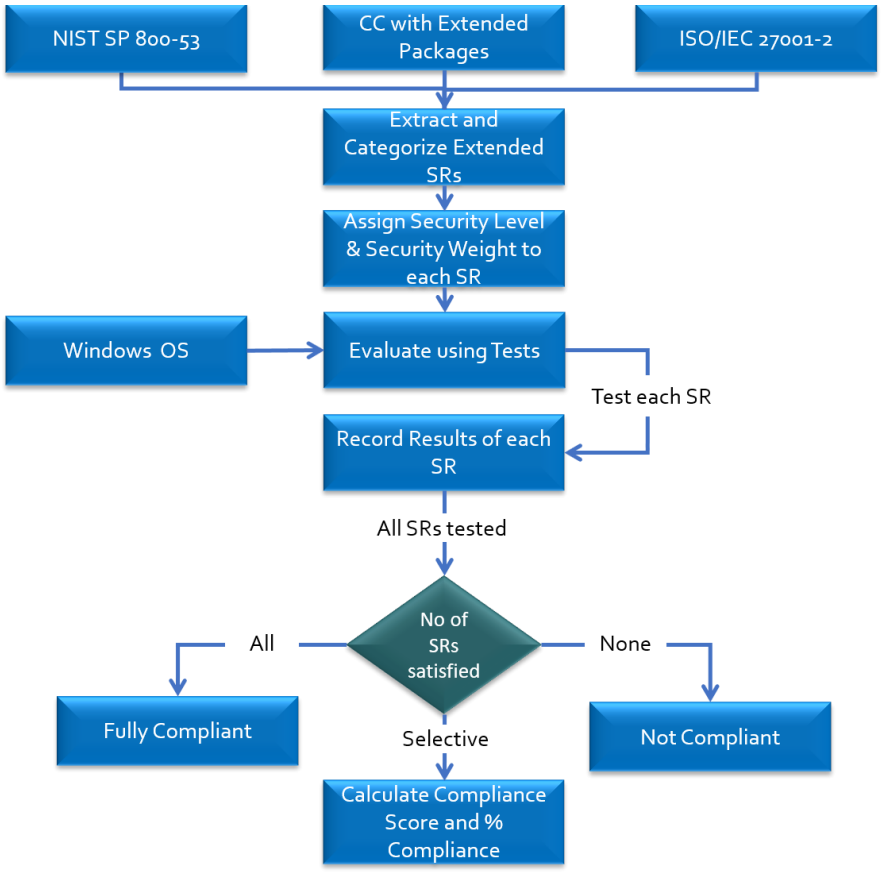


**Figure 3. 1 Workflow of Proposed SCE Framework**

## 3.2 Steps of Proposed SCE Framework for Windows 10

There are 6 steps that needs to be performed in order to assess security of Windows 10 OS. Each step is explained in detail as follows:

### 3.2.1 Extraction of Security Requirements

This step builds the security criteria against which given Windows 10 OS will be evaluated. We have used NIST SP 800-53, ISO 27001-2 and CC PP for GPOS along with EP for WLAN clients, SSH protocol and VPN clients to extract internationally accepted security functional requirements for thorough security assessment of Windows 10 OS. NIST SP 800-53 is not specifically designed for OS security, therefore we have used following control families as shown in Table 3.1 to select 55 sub-controls that were most suitable for the OS security evaluation. And from ISO 27001-2 we have used controls that are related to security of OSs. As a result, our evaluation criteria cover wide range of security domains that are most subjected to cyber threats as shown in Table 3.2. Information obtained in this phase will be used for evaluation in the next phases of the framework.

**Table 3. 1 Selected NIST SP 800-53 Controls**

| Control Families | Total no. of Controls | No. of Selected Controls | Selected Controls |
|---|---|---|---|
| AC: Access Control | 25 | 6 | AC-2, AC-6, AC-7, AC-9, AC-17, AC-18 |
| IA: Identification and Authentication | 12 | 2 | IA-3, IA-5 |
| CA: Assessment, Authorization and Monitoring | 9 | 1 | CA-3 |
| CP: Contingency Planning | 13 | 1 | CP-9 |
| SI: System and Information Integrity | 23 | 2 | SI-2, SI-7 |
| SC: System and Communications Protection | 51 | 3 | SC-28, SC-41, SC-18 |
| CM: Configuration Management | 14 | 6 | CM-2, CM-3, CM-7, CM-8, CM-10, CM-11 |
| AU: Audit and Accountability | 16 | 6 | AU-2, AU-3, AU-4, AU-8, AU-9, AU-14 |

### 3.2.2 Categorization of Security Requirements

This phase categorizes all the extracted security requirements in the previous step according to the targeted security domains. There are 14 different security categories that are considered vital for OS security evaluation. These categories, along with the description of selected security criteria are shown in Table 3.2. There are a total of 137 security requirements which are distributed as follows:

- 55 basic security requirements from NIST SP 800-53.
- 51 basic security requirements from CC PP for GPOS.
- 60 basic security requirements from ISO 27001.
- 26 additional security requirements from EP WLAN.
- 14 additional security requirements from EP VPN.
- 10 additional security requirements from EP SSH.

12 of the security requirements from CC overlap with NIST SP 800-53 security controls. This categorization of security requirements into different domains help us in understanding the security areas concerning OS evaluation for a better and more detailed evaluation approach.

**Table 3. 2 Security Categories and Subsequent Controls**

| Security Categories | Security Requirements | Description |
|---|---|---|
| Logging/Auditing | 18 | It contains SRs related to generation and security of event logs. |
| Cryptography | 34 | It contains SRs related to cryptographic algorithms adopted for data handling and operations. |
| Data Protection | 4 | It contains SRs related to integrity of stored executables. |
| Access Control | 16 | It contains SRs related to access control of objects and other resources. |

| | | |
|---|---|---|
| Management | 3 | It contains SRs related to security management of auditing, WLAN and VPN. |
| System Information | 2 | It contains SRs related to hardware and software information. |
| User Accounts | 12 | It contains SRs related to security and management of user accounts. |
| Networks | 24 | It contains SRs related to secure network protocols that covers WLAN, SSH and VPN. |
| Hardware | 2 | It contains SRs related to external connected media. |
| Software | 12 | It contains SRs related to security of installed software and its execution. |
| Notifications and Triggered Events | 1 | It contains SRs related to security notifications in case of configuration change. |
| Execution Privileges | 1 | It contains SRs related to execution security of connected media. |
| Identification and Authentication | 5 | It contains SRs related to authentication mechanisms. |
| Custom Configurations | 3 | It contains SRs related to security of customized accounts and registry. |

### 3.2.3 Assignment of Security Weight and Security Level

With increase in security threat landscape cyber criminals have become smarter and more focused on making money and intellectual theft by gaining access to critical information which affect businesses. Every business has different security needs according to their security objectives and operational environment. If critical data vital to operation of the business is compromised, the organization gets exposed to risks that could potentially lead to financial loses, legal issues, bankruptcy or even closure. Although confidentiality, integrity and availability of critical information are key factors for every business, there are some cases in which compromise of one factor will not entirely affect the business goals. For example, in the case of hospitals, banks, life insurance companies etc., the confidentiality, integrity and availability of critical information is very important, and their compromise can lead to a much greater loss. Whereas, for small scale businesses such as restaurants, the confidentiality is comparatively less important and if compromised, will

not hamper the overall business. Therefore, there is a need to assign security weights to each requirement based on its criticality for the maintenance of business goals and the security threats faced. This approach makes our framework flexible enough to be used by any size or type of user according to specific requirements. We have used three security levels and assigned them corresponding values of security weights. "L" is for Low, "M" is for Medium, and "H" is for High. Let's assume $L_i = 0$; $L_s = 25$, $M_i = 25$; $M_s = 50$ and $H_i = 50$; $H_s = 100$. Then the corresponding security weights are calculated as follows:

$$L_i/100 \leq W_L \leq L_s/100$$

$$0/100 \leq W_L \leq 25/100$$

$$0 \leq W_L \leq 0.25 \quad (1)$$

$$M_i/100 \leq W_M \leq M_s/100$$

$$25/100 \leq W_M \leq 50/100$$

$$0.25 \leq W_M \leq 0.5 \quad (2)$$

$$H_i/100 \leq W_H \leq H_s/100$$

$$50/100 \leq W_H \leq 100/100$$

$$0.5 \leq W_H \leq 1 \quad (3)$$

The subscripts 'i' and 's' represents the initial and final values selected for calculating the range of security weights for each level respectively. For simplicity, we have selected the highest value of each calculated range of weights from Equation (1), (2) and (3) i.e. $W_L = 0.25$, $W_M = 0.5$ and $W_H = 1$. Lowest values cannot be selected as $W_L = 0$ would mean the corresponding requirement has no weightage in the security of evaluated OS, which of course cannot be true in any scenario. The description of security levels and weights are given in Table 3.3.

**Table 3. 3 Assignment of Security Weights and Levels**

| Security Level | Security Weight | Description |
|---|---|---|
| L | 0.25 | Absence of these security requirements can be tolerated and, therefore, do not pose any critical cyber threat to the operation of business. |
| M | 0.5 | These requirements contribute to the security of business operations, but their compromise will not entirely affect the working of organization. |
| H | 1 | These requirements are crucial for the security of the user and their absence will expose the business to a high risk of cyberattacks greatly affecting its operation. |

### 3.2.4 Defining Tests and Evaluating Windows 10 OS

This phase is at the heart of our proposed framework that deals with defining tests to check status of each SR in Windows 10 OS. As stated earlier, the proposed framework should be easily adoptable by users to self-assess security of their OSs. So, this step focuses on designing tests that are easy to perform by non-technical individuals and does not require any complex tool or system setup. To achieve the same and avoid using any third-party application we have mostly used Command Prompt, PowerShell and User Interface of Windows 10 OS to check the status of each security requirement defined in previous phases of our framework. In this way, we have defined commands to test all 137 SRs. However, for the purpose of brevity, all tests cannot be shown here. So, to give a general idea the commands to check a few of the SRs related to “Cryptography” class are shown in Table 3.4. These tests have to be performed on the same computer system containing the Windows 10 OS under evaluation.

### 3.2.5 Recording and Classification of Results

After each test has been performed, this phase compiles and classifies results obtained from the previous step. Each SR is allotted one of the three values i.e., ‘0’, ‘0.5’ or ‘1’ depending upon the results obtained from testing. These values and their corresponding descriptions are shown in Table 3.5. Once these values have been assigned, the evaluated OS is classified into three categories i.e., “fully compliant”, “partially complaint” and “not

complaint". This classification is shown in Table 3.6. Operating systems that lie under "fully compliant" and "not complaint" classes are not dealt with any further. However, the subsequent steps are performed for "partially compliant" OSs to deduce their exact compliance score.

**Table 3. 4 Commands to Test "Cryptography" Related SRs**

| Security Requirement (SR) | ISO 27001 | NIST | CC PP GP-OS | EP WLAN | EP VPN | EP SSH | Commands |
|---|---|---|---|---|---|---|---|
| Information about windows encryption scheme used. | ✓ | ✓ | ✓ | | | | **Command Prompt:** >manage-bde –status **PowerShell:** >get-bitlockervolume |
| Only administrator shall have access right of event logs | | ✓ | | | | | **Command Prompt:** >wevtutil gl application >wevtutil gl security >wevtutil gl system >wevtutil gl setup |
| All certificates shall have expiry date | ✓ | ✓ | ✓ | | | | **Command Prompt:** >certmgr /all |
| Use RSA and Diffie Hellman for distribution of cryptographic keys | | | ✓ | | | | **Command Prompt:** >regedit HKLM\SYSTEM\CurrentControlSet\Control\Lsa\FipsAlgorithmPolicy >Disable-TlsCipherSuite –Name "the cipher suite we want to disable" >Enable-TlsCipherSuite –Name "the cipher suite we want to enable" **PowerShell:** > Get-TlsCipherSuite >CertUtil.exe –DisplayEccCurve |
| Generate symmetric, RSA, DSA cryptographic keys as specified by NIST FIPs | | | ✓ | | | | |
| Implement TLS 1.0 and TLS 1.1 or TLS 1.2 for EAP-TLS protocol, | ✓ | | | ✓ | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| supporting the mandatory cipher suite TLS_RSA_WITH_AES_128_CBC_SHA | | | | | | | The cipher suites for IPsec protocol can be seen by going to windows defender firewall with advanced security > properties > IPsec settings > IPsec defaults > customize, here "key exchange" and "data protection" cipher suits can be seen. We can see that key exchange algorithms DH 1, DH 2, DH 14, DH 24, ECDH P-256 and ECDH P-384 are supported by the operating system (windows 10).<br>Go to group policy editor > computer configuration > administrative templates > network > SSL configuration settings<br>Configure both 'SSL cipher suit order' and 'ECC curve order' |
| Administrator shall configure the list of algorithm suites for EAP-TLS exchanges | | | | ✓ | | | |
| Perform cryptographic signature services (generation and verification) in accordance with NIST, FIPs | | | ✓ | ✓ | | | |
| Perform cryptographic hashing service in accordance with NIST, FIPs (SHA-1 and SHA-2 family) | | | ✓ | ✓ | | | |
| Perform encryption/decryption services for data in accordance with NIST, FIPs | | | ✓ | ✓ | | | |
| Perform keyed hash message authentication as defined in NIST FIPS | | | ✓ | ✓ | | | |
| Authorized administrator | | | | ✓ | | | **PowerShell:** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| shall configure the list of CAs that are allowed to sign authentication server certificates | | | | | | | >Get-Childitem cert:\LocalMachine\root \|format-list<br>>Get-Childitem cert:\LocalMachine\root \| Where {$_.NotAfter -lt (Get-Date).AddDays(30)} |
| SSH protocol shall perform encryption/decryption services for data using AES-CTR mode | | | | | | ✓ | **Command Prompt:**<br>>ssh –Q cipher |
| SSH protocol shall use aes128-ctr, aes256-ctr, aes128-cbc/aes256-cbc, AEAD_AES_128_GCM/ AEAD_AES_256_GCM algorithms for encryption | | | | | | ✓ | **Command Prompt:**<br>>ssh –Q cipher<br>>ssh –Q cipher-auth |
| VPN client shall use RSA or ECDSA schemes for IKE peer authentication | | | | | ✓ | | **User Interface:**<br>windows defender firewall with advanced security > properties > IPsec settings > IPsec defaults > customize > authentication method > advanced > customize > add > computer certificate > signing algorithm |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SSH protocol shall use ssh-rsa/ecdsa-sha2-nistp256 and ecdsa-sha2-nistp384/x509v3-ecdsa-sha2-nistp256/ x509v3-ecdsa-sha2-nistp384 as public key algorithms | | | | | | ✓ | **Command Prompt:**<br>>ssh –Q key<br>>ssh –Q key-cert (certificate key type)<br>>ssh –Q key-plain (non-certificate key type) |

**Table 3. 5 Justification of Values Assigned to Tested SRs**

| **Status of SR 'S'** | **Description** |
|---|---|
| 0 | This value is given when a security requirement is not met. |
| 0.5 | This value is given when a security requirement is partially met i.e., a portion of SR is fulfilled. |
| 1 | This value is given when a security requirement is completely met. |

**Table 3. 6 Classification of Evaluated Windows 10 OS Based on Test Results**

| **Class** | **Description** |
|---|---|
| Not Compliant | When none of the tested security requirements are fulfilled i.e., the status of each SR is 0. |
| Partially Compliant | When some of the security requirements are either partially met or not fulfilled at all i.e., the status of some of the SRs is either 0 or 0.5. |
| Fully Compliant | When all the security requirements are completely fulfilled i.e., the status of all SRs is 1. |

### 3.2.6 Calculation of Percentage Compliance

After testing and classification of results, another important phase is calculation of security compliance score and percentage compliance of the evaluated OS. This phase will only consider OSs that fall under "partially compliant" class as other two classes will naturally have 0% and 100% compliance. To calculate compliance score of a single SR, this framework uses Equation 4, which incorporates security weights for flexibility of different organizations to determine the overall compliance with the provided security criteria.

$$\theta_c = W_i \text{ x } S_i \tag{4}$$

Where,

- '$\theta_c$' is the calculated compliance score for considered SR.
- '$W_i$' is the value of security weight assigned to considered SR.
- '$S_i$' is the status of considered SR.
- 'i' is the number of SR being considered.

The values of '$W_i$' and '$S_i$' will be from [0.25, 0.5, 1] and [0, 0.5, 1] set respectively and '$\theta_c$' will always lie between 0 and 1. In our framework, security requirements are categorized into security classes for more focused security assessment, therefore, the average compliance score for each class can be calculated separately to depict the security of OS with respect to that particular security domain. It is calculated by using Equation 5 as follows:

$$\theta c(\text{avg}) = \frac{\sum_{i=1}^{i=r} Wi \; x \; Si}{\text{r}} \tag{5}$$

Where,

- '$\theta_{c\,(avg)}$' is the average compliance score of a particular class.
- 'r' is the total number of SRs in that class.
- 'i' is the number of SR being considered in that class.

The value of average compliance score will also lie within range of 0 to 1. This calculation will help organizations to figure out weak areas of their OS security and would allow them to focus more on strengthening security of those areas which mostly affect their business

goals. The overall compliance score for the operating system is calculated by Equation 6 as follows:

$$\theta co = \sum_{i=1}^{n} Wi\ x\ Si \tag{6}$$

Where 'n' is the total number of SRs in all the classes combined. In the case of our proposed framework, the value of 'n' is 137. Value of '$\theta_{co}$' will give us a general idea regarding the security of the evaluated Windows 10 OS. A greater '$\theta_{co}$' value corresponds to a higher degree of security as it shows that SRs with greater security weightages have been fulfilled by the evaluated OS. Subsequently, the percentage compliance can be calculated as follows:

$$\theta_{\%} = \frac{\theta co}{\theta cmax} \times 100 \tag{7}$$

Where,

- '$\theta co$' is the overall compliance score of OS, and
- 'θcmax' is the maximum attainable compliance score.

θcmax is calculated by considering the OS to be "fully compliant" i.e., value of all '$S_i$' is considered '1'. The outputs from Equation 6 and 7 will present a complete picture of the security evaluation of Windows 10 OS keeping in view the internationally recognized security criteria defined in the framework. Compliance score of greater than or equal to 90% is considered to be reasonable secure. Whereas a score less than 90% would indicate that the users need to re-evaluate their security policy and procedures regarding OS security.

### 3.3 Comparative Matrix for the Proposed Framework

Due to limited work available on the subject we have carried out comparative analysis of our proposed SCE framework for windows with two other evaluation schemes given in [38] and [40]. Unlike others, this framework provides an integrated and flexible approach that facilitates even a non-technical user to assess OS security according to specific operational environment. Moreover, this approach is easy to adopt due to lack of involvement of any external assessment tool or technical personnel and provides reasonable accuracy for security assessment of OS. Table 3.7 shows the comparative matrix of our framework.

**Table 3. 7 Comparative Matrix**

| Contribution | DSE for Android | CEM | SCE |
|---|---|---|---|
| Integrated approach | ✗ | ✗ | ✓ |
| Based on well recognized security standard | ✗ | ✓ | ✓ |
| Easy to use | ✓ | ✗ | ✓ |
| Does not require technical expertise | ✓ | ✗ | ✓ |
| Cost effective | ✓ | ✗ | ✓ |
| Time efficient | ✓ | ✗ | ✓ |
| Does not require source code for evaluation | ✓ | ✗ | ✓ |
| Good accuracy | ✓ | ✓ | ✓ |

## 3.4 Conclusion

The cyber security threat affects all the users of operating systems alike, therefore we have designed a framework that works for everyone according to their specific needs of security and guides them in checking compliance of their operating systems against internationally adopted security standards. Our framework can also be used by individuals in assessing their OS security with respect to a particular security domain and this assessment would allow them to critically analyze their security policies for operating systems.

*Chapter 4*

# 4 VALIDATION OF PROPOSED WINDOWS 10 SCE FRAMEWORK

For the purpose of validation and demonstration, we have applied the proposed SCE framework on Windows 10 OS at both a home-based PC as well as a security research laboratory to assess OS security with erstwhile mentioned criteria. Application of all the steps and their results are explained individually for both validations throughout this chapter.

## 4.1 Validation for Single or Home User

Single users refer to individuals making regular use of OS and they form the most common category of computer users. In this category, personal computers are used for storage of private data and for internet access applications such as online shopping, internet banking, gaming, downloading software and other forms of media consumption. Hence, the security assessment of their OS is equally important like in any other organization or corporate setup utilizing computer systems. As an example, we applied this framework on our local PC running a 64-bit Windows 10 Pro OS. All security patches were installed till date of application. The first two phases namely extraction and categorization of SRs will remain same for every device. Therefore, they are not discussed here. Whereas the rest of the phases are explained in subsequent sections of this chapter.

### 4.1.1 Assignment of Security Weight and Security Level

This phase varies for each type of user depending upon security objectives and threats faced. Single or home users install a number of applications to perform various tasks and store important personal files on their systems. Therefore, their security objectives are mostly concerned with the smooth running of installed applications, secure system boot, closure of unnecessary ports to avoid network attacks, significant use of data encryption, making secure network connections, implementing strong password policies to avoid unauthorized access and avoiding privilege escalation in case of multiple users etc. Considering these security needs of home users, we have assigned security weights and corresponding security levels to each SR. In our validation we have assumed that there are no multiple accounts on the system i.e., the user is the administrator and sole owner of the computer. Once more, owing to the limitation of space, a few of the security requirements

belonging to the ‘Cryptography’ class are chosen to assign security weights and levels as shown in Table 4.1. These requirements will be used throughout both the validations for reference of our calculations.

### 4.1.2 Applying Tests for Evaluation

In this phase we performed tests to evaluate all 137 security requirements. Figure 4.1, 4.2, 4.3 and 4.4 shows evaluation results of SRs mentioned in Table 4.1. According to the results obtained, important data stored on personal computers is not encrypted. The administrator account has full access rights to all event logs and the SDDL string (D;;0xf0007;;;AN) is missing that denies all anonymous users any kind of access to event logs. RSA and DSA are used for generation of asymmetric keys as required by the security standards. SHA-1 and SHA-2 family are used for secure hashing algorithms. From registry, we verified that the ‘FIPs algorithm policy’ is enabled which ensures all cryptographic services are performed using algorithms supported by NIST FIPs 140-2. TLS 1.2 protocol is supported, and any TLS cipher suite can be enabled and disabled only by the administrator. The Windows OS built-in VPN client uses RSA and ECDSA schemes for peer authentication. The built-in SSH client/server of the PC under evaluation had some additional algorithms configured for use like rijndael-cbc and cahcha20-poly1305 against the provided security criteria. These results show that some security loopholes exist in our Windows 10 configuration which could be improved to make our working environment more secure.

```
PS C:\WINDOWS\system32> CertUtil.exe -DisplayEccCurve
Microsoft SSL Protocol Provider:
------------------------------
Curve Name          Curve OID                   Public Key Length   CurveType
-----------------------------------------------------------------------------
curve25519                                      255                 29
nistP256            1.2.840.10045.3.1.7         256                 23
nistP384            1.3.132.0.34                384                 24
brainpoolP256r1     1.3.36.3.3.2.8.1.1.7        256                 26
brainpoolP384r1     1.3.36.3.3.2.8.1.1.11       384                 27
brainpoolP512r1     1.3.36.3.3.2.8.1.1.13       512                 28
nistP192            1.2.840.10045.3.1.1         192                 19
nistP224            1.3.132.0.33                224                 21
nistP521            1.3.132.0.35                521                 25
secP160k1           1.3.132.0.9                 160                 15
secP160r1           1.3.132.0.8                 160                 16
secP160r2           1.3.132.0.30                160                 17
secP192k1           1.3.132.0.31                192                 18
secP192r1           1.2.840.10045.3.1.1         192                 19
secP224k1           1.3.132.0.32                224                 20
secP224r1           1.3.132.0.33                224                 21
secP256k1           1.3.132.0.10                256                 22
secP256r1           1.2.840.10045.3.1.7         256                 23
secP384r1           1.3.132.0.34                384                 24
secP521r1           1.3.132.0.35                521                 25
```

**Figure 4. 1 Elliptical Curve Algorithms Configured to be used by Tested Home PC**

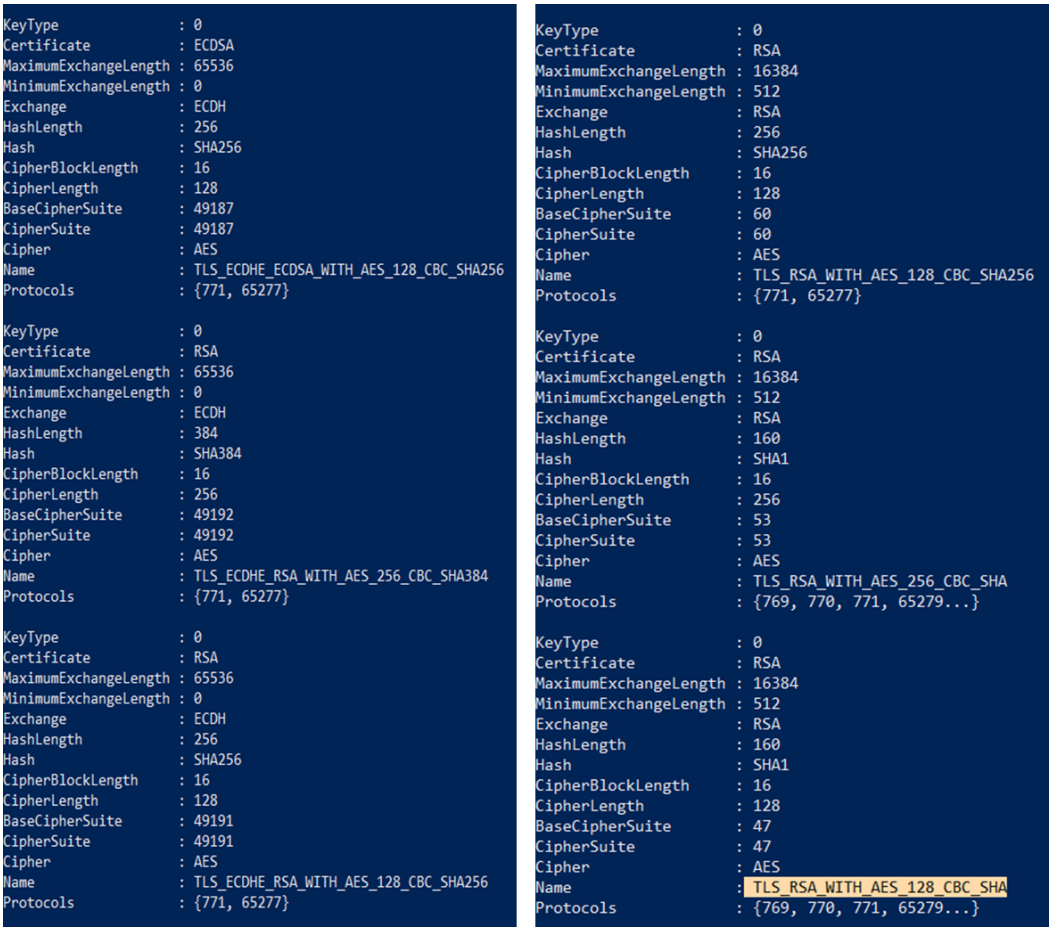


**Figure 4. 2 TLS Cipher suites and Hashing Algorithms used by Tested Home PC**

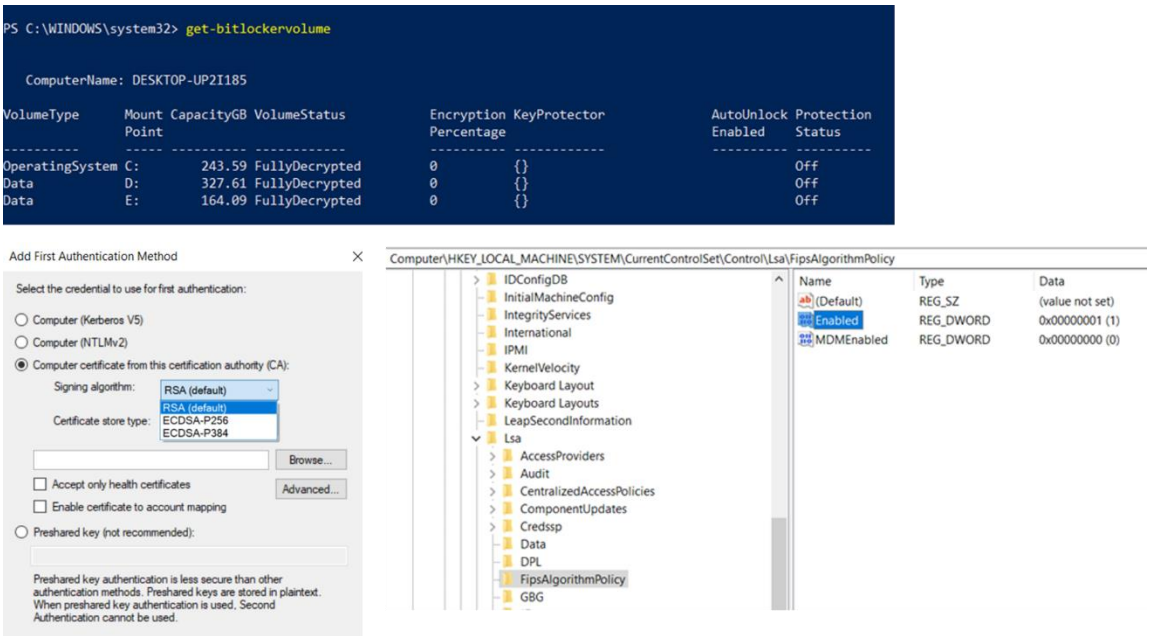


**Figure 4. 3 Data Encryption, VPN Authentication Algorithms and FIPs Algorithm Policy used by Tested Home PC**

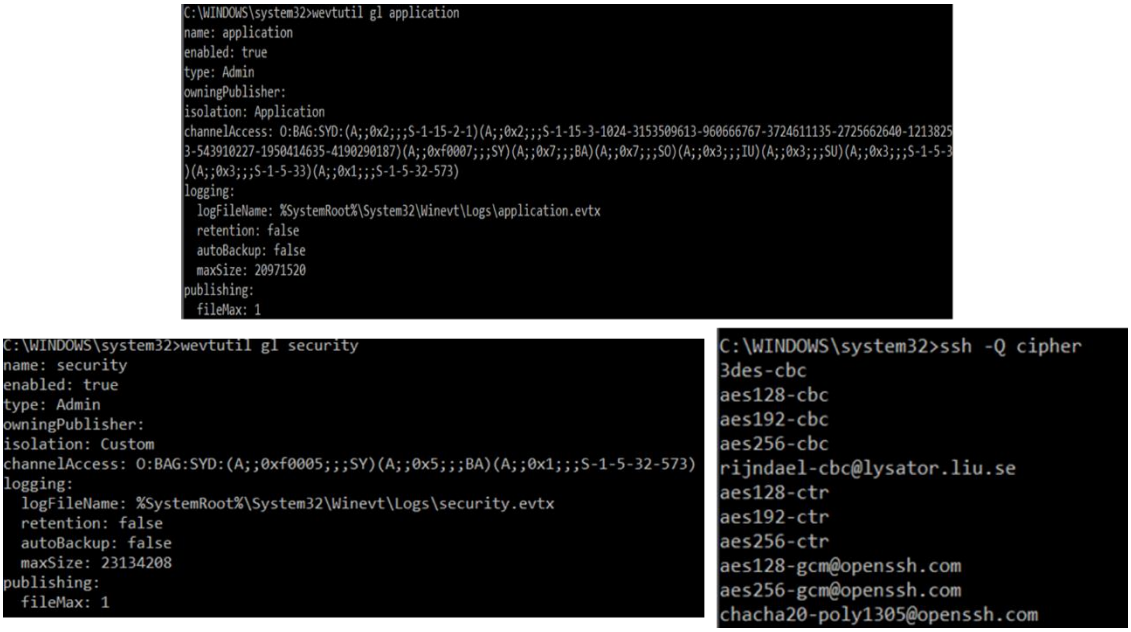


**Figure 4. 4 Result of Event Log Access Permissions and SSH Encryption Ciphers for Tested Home PC**

### 4.1.3 Recording Results of Evaluation

After applying tests and carefully analyzing the results, we recorded values of '$S_i$' for each SR as can be seen in Table 4.1. These values show that our PC under test is "partially compliant" with the security criteria recommended for Windows framework, as there are some requirements which are either "not fulfilled" or "partially fulfilled".

**Table 4. 1 Compliance Score of Each SR for Home User**

| **Security Requirement 'SR'** | **Security Level 'L'** | **Security Weight '$W_i$'** | **Security Status '$S_i$'** | **Compliance Score '$\theta_c$'** |
|---|---|---|---|---|
| Use built-in encryption scheme to protect data | H | 1 | 0 | 0 |
| Only administrator shall have access right of event logs | M | 0.5 | 0.5 | 0.25 |
| All certificates shall have expiry date | H | 1 | 1 | 1 |
| Use RSA and Diffie Hellman for distribution of cryptographic keys | H | 1 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| Generate symmetric, RSA, DSA cryptographic keys as specified by NIST FIPs | H | 1 | 1 | 1 |
| Implement TLS 1.0 and TLS 1.1 or TLS 1.2 for EAP-TLS protocol, supporting the mandatory cipher suite TLS_RSA_WITH_AES_128_CBC_SHA | H | 1 | 1 | 1 |
| Administrator shall configure the list of algorithm suites for EAP-TLS exchanges | M | 0.5 | 1 | 0.5 |
| Perform cryptographic signature services (generation and verification) in accordance with NIST, FIPs | H | 1 | 1 | 1 |
| Perform cryptographic hashing service in accordance with NIST, FIPs (SHA-1 and SHA-2 family) | H | 1 | 1 | 1 |
| Perform encryption/decryption services for data in accordance with NIST, FIPs | H | 1 | 1 | 1 |
| Perform keyed hash message authentication as defined in NIST FIPS | H | 1 | 1 | 1 |
| Authorized administrator shall configure the list of CAs that are allowed to sign authentication server certificates | M | 0.5 | 1 | 0.5 |
| SSH protocol shall perform encryption/decryption services for data using AES-CTR mode | H | 1 | 0.5 | 0.5 |
| SSH protocol shall use aes128-ctr, aes256-ctr, aes128-cbc/aes256-cbc, AEAD_AES_128_GCM/ AEAD_AES_256_GCM algorithms for encryption | H | 1 | 0.5 | 0.5 |
| VPN client shall use RSA or ECDSA schemes for IKE peer authentication | H | 1 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| SSH protocol shall use ssh-rsa/ecdsa-sha2-nistp256 and ecdsa-sha2-nistp384/x509v3-ecdsa-sha2-nistp256/ x509v3-ecdsa-sha2-nistp384 as public key algorithms | H | 1 | 0.5 | 0.5 |

### 4.1.4 Calculation of Percentage Compliance

Further calculations are carried out by using the values of security weight and security status. Table 4.1 contains the compliance score for each SR, which is calculated separately by using Equation 4. The average compliance score for 'Cryptography' class is calculated using Equation 5 as below:

$$\theta c(avg) = \frac{26.25}{34} = 0.772 \quad (8)$$

Table 4.2 contains the values of average compliance score for all security classes. For added precision, the values have been considered up to 3 decimal places. These calculations deduce that the PC under test is most secure in terms of data protection and least secure in terms of execution privileges. The overall compliance score for the Windows 10 OS of the test PC is calculated by using Equation 6 as under:

$$\theta co = \sum_{i=1}^{137} Wi\ x\ Si = 101.75 \quad (9)$$

We have used Equation 7 for calculating percentage compliance of our PC with recommended security criteria, according to its operational environment which is set by values of security weights assigned to each SR. $'\theta cmax'$ is calculated as follows by considering the ideal situation where all the SRs are completely fulfilled i.e., the value of '$S_i$' for all SRs is considered 1:

$$\theta\ \% = \frac{101.75}{120.5} \times 100 = 84.44\ \% \quad (10)$$

The security evaluation of our OS has shown that this PC is not satisfactorily secure in its operation. Especially since the automatic execution security is neglected which could lead to 29 different types of USB-based attacks. Which can manifest by injecting the system with embedded malicious payload resulting in comprise of critical data [42].

**Table 4. 2 Average Compliance Score of Security Classes for Home User**

| Security Classes | Number of SRs 'r' | Sum of Compliance Score '$\sum_{i=1}^{i=r} Wi\ x\ Si$' | Average Compliance Score '$\theta c(avg)'$ |
|---|---|---|---|
| Logging/Auditing | 18 | 12 | 0.667 |
| Cryptography | 34 | 26.25 | 0.772 |
| Data Protection | 4 | 4 | 1 |
| Access Control | 16 | 12.25 | 0.766 |
| Management | 3 | 1.5 | 0.5 |
| System Information | 2 | 1.5 | 0.75 |
| User Accounts | 12 | 8.5 | 0.708 |
| Networks | 24 | 18.75 | 0.781 |
| Hardware | 2 | 0.5 | 0.25 |
| Notifications and Triggered Events | 1 | 0.5 | 0.5 |
| Software | 12 | 11 | 0.917 |
| Execution Privileges | 1 | 0 | 0 |
| Identification and Authentication | 5 | 3 | 0.6 |
| Custom Configurations | 3 | 2 | 0.667 |

## 4.2 Validation for Security Research Lab

It is a security lab with the aim of researching and developing products/tools that help in mitigating the effects of ever-increasing threats in the cyber security domain. The success and reputation of these labs greatly rely on their security, privacy and authenticity. And any security incident can potentially tarnish the trust established by the public on the quality of projects produced by these labs. The operational objectives of such a lab are mostly concerned with the following requirements:

- Providing flexibility of remote access to workers.
- Having secure internet access.
- Logging all security incidents related to crucial tasks.
- Safely storing critical data about developed products/ software to avoid compromising their confidentially and integrity.
- Smart utilization of network resources.

- Implementing strict access control policies and password policies to avoid unauthorized access to system or data.
- Using authentic software to avoid any data breaches or malicious activities.
- Having backup of data and servers to recover quickly from unavailability of important information and implementing strong encryption algorithms to ensure confidentiality and integrity of data in transition or at rest.

All of these requirements must be incorporated in the underlying OS for smooth and error-free working of the research lab. Therefore, we applied our framework for security assessment of Windows 10 OS of a certain lab. The phases of evaluation are given as follows:

### 4.2.1 Assignment of Security Weight and Security Level

Standard Operating Procedures (SOPs) followed by the research lab, which contains all the policies crucial for their successful operation and security, were obtained for the purpose of this phase. Based on these policies and the operational environment described above, we assigned security weights to each SR that compliments the security needs of the research lab. Table 4.3 contains the values of security levels and corresponding security weights assigned to a few of the Cryptographic SRs.

### 4.2.2 Applying Tests for Evaluation

Due to security hazards and strict access control policies of the lab, we could not perform evaluation tests directly on one of their PCs running Windows 10 OS. Therefore, we have used three case studies for demonstrating application of our framework.

#### *4.2.2.1 Case 1*

In this scenario the owner of the PC strictly adheres to the complete security policy of the lab. No negligence is observed on part of implementing security according to the defined Standard Operating Procedures (SOPs).

#### *4.2.2.2 Case 2*

In this scenario the employee has not completely followed the security policy provided by the research lab. Some of the requirements, which are only presented as suggestions or recommendations in the policy, are ignored/not implemented.

#### *4.2.2.3 Case 3*

In this scenario only those requirements are implemented that are absolutely essential for achieving compliance with the standards followed by the lab i.e., ISO/IEC 27001 and

27002. All other security recommendations/suggestions and guidelines have been ignored. Thus, such a PC is bound to be comparatively less secure than the other two cases described.

### 4.2.3 Recording Results of Evaluation

For all three case studies, we have assigned values of security status '$S_i$' to each SR by consulting with the SOPs obtained from the research lab. These values depict that the evaluated PC is "partially compliant" in all cases with the security criteria recommended in this framework. Consequently, there are some requirements which are either "not fulfilled" at all or "partially fulfilled".

**Table 4. 3 Compliance Score of each SR for Research Lab**

| **'SR'** | **'L'** | **'$W_i$'** | **Case 1 '$S_i$'** | **Case 2 '$S_i$'** | **Case 3 '$S_i$'** | **Case 1 ' $\theta_c$'** | **Case 2 ' $\theta_c$'** | **Case 3 ' $\theta_c$'** |
|---|---|---|---|---|---|---|---|---|
| Use inbuilt encryption scheme to protect data. | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Only administrator shall have access right of event logs | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| All certificates shall have expiry date | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Use RSA and Diffie-Hellman for distribution of cryptographic keys | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Generate symmetric, RSA, DSA cryptographic keys as specified by NIST FIPs | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Implement TLS 1.0 and TLS 1.1 or TLS 1.2 for EAP-TLS protocol, supporting the mandatory cipher suite TLS_RSA_WITH_AES_128_CBC_SHA | H | 1 | 1 | 0.5 | 0 | 1 | 0.5 | 0 |
| Administrator shall configure the list of algorithm suites for EAP-TLS exchanges | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Perform cryptographic signature services (generation and verification) in accordance with NIST, FIPs | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Perform cryptographic hashing service in accordance with NIST, FIPs (SHA-1 and SHA-2 family) | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Perform encryption/decryption services for data in accordance with NIST, FIPs | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Perform keyed hash message authentication as defined in NIST FIPS | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Authorized administrator shall configure the list of CAs that are allowed to sign authentication server certificates | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| SSH protocol shall perform encryption/decryption services for data using AES-CTR mode | H | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| SSH protocol shall use aes128-ctr, aes256-ctr, aes128-cbc/aes256-cbc, AEAD_AES_128_GCM/ AEAD_AES_256_GCM algorithms for encryption | H | 1 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| VPN client shall use RSA or ECDSA schemes for IKE peer authentication | H | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| SSH protocol shall use ssh-rsa/ecdsa-sha2-nistp256 and ecdsa-sha2-nistp384/x509v3-ecdsa-sha2-nistp256/ x509v3-ecdsa-sha2-nistp384 as public key algorithms | H | 1 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |

### 4.2.4 Calculation of Percentage Compliance

In this phase we have performed calculations by using the values of security weights and security status given in Table 4.3 and calculated the compliance score for each SR separately for all the three cases, which is also given in Table 4.3.

#### *4.2.4.1 Case 1*

The average compliance score for 'Cryptography' class is calculated using Equation 5 as below:

$$\theta c(avg) = \frac{32}{34} = 0.941 \quad (11)$$

Table 4.4 contains the values of average compliance score for all the defined security classes. For precision the values have been considered up to 3 decimal places. These results deduce that the evaluated PC of the research lab is more secure in terms of data protection, access control, system information, user accounts, hardware information, event notification and execution privileges. Whereas the same PC is least secure in terms of management controls when the complete security policy of the lab is followed strictly. The overall compliance score is calculated using Equation 6 as follows:

$$\theta co = \sum_{i=1}^{137} Wi\ x\ Si = 124 \tag{12}$$

Subsequently, we used Equation 7 for calculating percentage compliance. For that first we need to calculate $'\Theta cmax'$ by considering the ideal situation where all the SRs are completely fulfilled i.e., the value of '$S_i$' for all SRs is considered 1.

$$\theta_{\%} = \frac{124}{135.25} \times 100 = 91.68\ \% \tag{13}$$

The evaluated PC is 91.68% compliant with NIST, CC and ISO standards. Considering the security interests of the lab and the kind of tasks performed, we cannot say that all Windows 10 security features are being used in their full capacity to avoid cyberattacks.

**Table 4. 4 Average Compliance Score of Security Classes for Research Lab**

| **Security Classes** | **'r'** | **Case 1** $\sum_{i=1}^{i=r} Wi\ x\ Si$ | **Case 1** $\theta c(avg)$ | **Case 2** $\sum_{i=1}^{i=r} Wi\ x\ Si$ | **Case 2** $\theta c(avg)$ | **Case 3** $\sum_{i=1}^{i=r} Wi\ x\ Si$ | **Case 3** $\theta c(avg)$ |
|---|---|---|---|---|---|---|---|
| Logging/Auditing | 18 | 16.5 | 0.917 | 16.5 | 0.917 | 13.5 | 0.75 |
| Cryptography | 34 | 32 | 0.941 | 28.5 | 0.838 | 22.5 | 0.662 |
| Data Protection | 4 | 4 | 1 | 4 | 1 | 2 | 0.5 |
| Access Control | 16 | 16 | 1 | 16 | 1 | 15 | 0.937 |
| Management | 3 | 1.5 | 0.5 | 1.5 | 0.5 | 1.5 | 0.5 |
| System Information | 2 | 2 | 1 | 2 | 1 | 2 | 1 |
| User Accounts | 12 | 12 | 1 | 12 | 1 | 11 | 0.917 |
| Networks | 24 | 18.5 | 0.770 | 13 | 0.542 | 10.5 | 0.437 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Hardware | 2 | 2 | 1 | 2 | 1 | 1 | 0.5 |
| Notifications and Triggered Events | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Software | 12 | 11 | 0.917 | 11 | 0.917 | 11 | 0.917 |
| Execution Privileges | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Identification and Authentication | 5 | 4.5 | 0.9 | 4.5 | 0.9 | 4.5 | 0.9 |
| Custom Configurations | 3 | 2 | 0.667 | 2 | 0.667 | 2 | 0.667 |

#### *4.2.4.2 Case 2*

The average compliance score for 'Cryptography' class is calculated using Equation 5 as below:

$$\theta c(avg) = \frac{28.5}{34} = 0.838 \quad (14)$$

Table 4.4 contains the values of average compliance score for all the defined security classes. These results deduce that evaluated PC of the research lab is more secure in terms of data protection, access control, system information, user accounts, hardware information, event notification and execution privileges, whereas it is least secure in terms of management functions. This is valid in the case where some of the suggestions/guidelines provided in the lab policy are ignored. The overall compliance score and percentage compliance are calculated using Equation 6 and 7 respectively as follows:

$$\theta co = \sum_{i=1}^{137} Wi\ x\ Si = 115 \quad (15)$$

$$\theta_{\%} = \frac{115}{135.25} \times 100 = 85.03\ \% \quad (16)$$

In this case the evaluated PC is 85.03% compliant with NIST and CC. This proves that a slight negligence in implementation of security policies could lead to a considerably less secure Windows 10 OS, thus making the research lab more susceptible to cyber-attacks. Some of the loopholes and misconfigurations might result in network attacks that attempt to bypass security mechanisms in place and disrupt legitimate network operations

including malfunctioning of network devices, denying services to legitimate users and reducing network throughput [43].

#### *4.2.4.3 Case 3*

The average compliance score for 'Cryptography' class is calculated using Equation 5 as below:

$$\theta c(avg) = \frac{22.5}{34} = 0.662 \quad (17)$$

Table 4.4 contains the values of average compliance scores for all the defined security classes. These results conclusively deduce that Windows 10 PC of the research lab is more secure in terms of system information, event notification and execution privileges. Whereas it is least secure in terms of network controls. This result is valid in the case when only those requirements are implemented that are absolutely essential for achieving compliance with standards followed by the research lab. The overall compliance score and percentage compliance are calculated as follows:

$$\theta co = \sum_{i=1}^{137} Wi\ x\ Si = 98.5 \quad (18)$$

$$\theta\ \% = \frac{98.5}{135.25} \times 100 = 72.83\ \% \quad (19)$$

According to Equation 19, the evaluated PC is 72.83% compliant with given security criteria. This case is the weakest in terms of Windows 10 OS security as no Random Address Space Layout and Stack Based Buffer Overflow protections are enabled by the employee, which lead to software security vulnerabilities resulting in buffer overflow attacks (BOF) [44]. Several BOF can easily manipulate program control flow by avoiding specific instructions related to security, and thus, the attack code is successfully executed on the system [45]. However, they can be detected by pattern analysis, information flow analysis and constrain solving techniques [46]. Negligence in correct management of security updates, backup of system and file data, secure firewall configurations to block known malicious IPs, efficient incident response plans, restricting unsecure Wi-Fi networks and avoiding unreliable websites result in ransomware and other malware attacks [47] [48] [49]. The detection of Ransomware is difficult as its system calls are a subset of all the system calls that are logged during normal operation of the PC [50] and its payload, like any other malware, contains techniques which makes its analysis more difficult [51].

## 4.3 Resilience to Cyber Threats

Our SCE framework provides resilience to most of the OS related cyber threats due to presence of diverse and extensive security requirements in security criteria proposed for OS evaluation. These threats are shown as follows in Table 4.5.

**Table 4. 5 Resilience to Cyber Threats**

| Cyber Threats | Requirements to Avoid Cyber Threats |
|---|---|
| USB based attacks | Display list of all connected external devices, prevent/block automatic execution of CD, DVD and USB. |
| Privilege escalation | Strong password policy, user groups with minimum necessary permissions, install updates and patches, close unnecessary ports, remove unused user accounts, change default credentials and passwords regularly. |
| Buffer overflow | Enable data execution protection (DEP), address space randomization (ASLR) and exception handler overwrite protection (SEHOP). |
| Denial of service | Limit ingress and egress traffic, firewall configurations, audit failed logon attempts and install all security patches. |
| Unauthorized access | Strong password policy, close unnecessary ports, session timeout, threshold for failed authentication attempts, install security patches, access control policy and audit logon attempts. |
| Eavesdropping | Use encryption algorithms, VPN for network traffic and HTTPS for web-based communication. |
| Malware | Strong password policy, multi factor authentication, backup critical data, install security updates, close unused ports/protocols, remove inactive user accounts, configure firewall rules, block automatic execution of USB, block use of unauthorized software. |

## 4.4 Conclusion

Validation of our home PC has revealed that all the security features provided by Windows 10 that ensures compliance with CC, ISO and NIST are not properly utilized. This security misconfiguration mostly occurs in the domain of hardware management and execution privileges as ‘automatic execution’ of external media is not restricted. This weakens our Windows 10 OS against all types of USB based attacks and other malwares injected through external media. Validation of security research lab through our framework reveals that the SOPs adopted by lab does not ensure its absolute compliance with NIST SP 800-53 and CC standard. Even if the policy is carefully implemented by the employee, it is

only 91.68% compliant with the given criteria in its operational environment and lacks a bit of security in management controls. Therefore, it is recommended to separately adopt an operating system security policy that is designed according to the requirements of NIST SP 800-53, ISO 27001-2 and CC (with extended packages) to strengthen the OS security in all the operational domains of the research lab.

# 5 PROPOSED LINUX UBUNTU SCE FRAMEWORK

## 5.1 Introduction

In addition to SCE framework for Windows 10 we have proposed another framework for the security evaluation of Linux operating systems. We chose Ubuntu 20.04 to perform all the evaluation tests and check its compliance against the security criteria provided by NIST SP 800-53. The proposed framework is easy to adopt by any user and calculates the percentage compliance of the tested PC with provided security criteria. The use of security weights and security levels for each SR makes it flexible to be used by any user according to their security needs or operational environment. Important phases of this framework are presented in Figure 5.1 and the general flow of the evaluation steps are displayed in Figure 5.2.


**Figure 5. 1 Proposed SCE Framework for Linux Ubuntu**

## 5.2 Steps of Proposed SCE Framework for Linux Ubuntu

There are 6 steps that needs to be performed in order to assess security of Linux Ubuntu. Each step is explained in detail below:

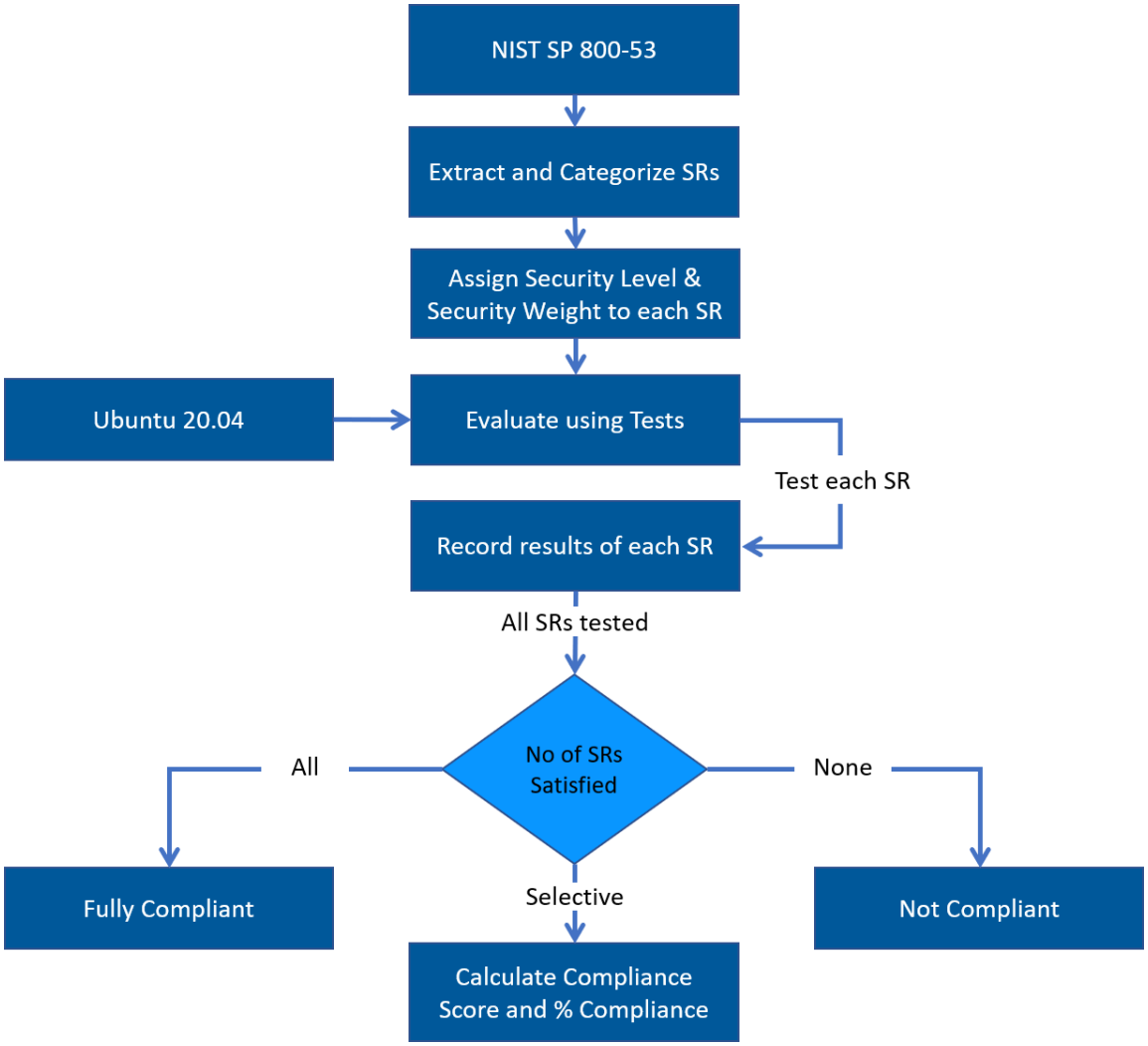


**Figure 5. 2 Workflow of Proposed SCE Framework**

### 5.2.1 Extraction of Security Requirements

This step focuses on building the criteria against which given Linux OS will be evaluated. We have used NIST SP 800-53 to extract internationally accepted security functional requirements for general security assessment of Linux Ubuntu OS. NIST SP 800-53 is not specifically designed for OS security, therefore we have used following control families as shown in Table 5.1 and selected 49 sub-controls that deemed most suitable for the OS evaluation. Our evaluation criteria cover most of the security domains as can be seen from Table 5.2. Information obtained in this phase will be helpful for evaluation in subsequent phases.

**Table 5. 1 Selected Controls of NIST SP 800-53 for Linux Ubuntu Evaluation**

| Control Families | Total no. of Controls | No. of Selected Controls | Selected Controls |
|---|---|---|---|
| AC: Access Control | 25 | 6 | AC-2, AC-6, AC-7, AC-9, AC-17, AC-18 |
| IA: Identification and Authentication | 12 | 2 | IA-3, IA-5 |
| CA: Assessment, Authorization and Monitoring | 9 | 1 | CA-3 |
| CP: Contingency Planning | 13 | 1 | CP-9 |
| SI: System and Information Integrity | 23 | 2 | SI-2, SI-7 |
| SC: System and Communications Protection | 51 | 3 | SC-28, SC-41, SC-18 |
| CM: Configuration Management | 14 | 6 | CM-2, CM-3, CM-7, CM-8, CM-10, CM-11 |
| AU: Audit and Accountability | 16 | 6 | AU-2, AU-3, AU-4, AU-8, AU-9, AU-14 |

### 5.2.2 Categorization of Security Requirements

This phase categorizes all the extracted security requirements in the previous step according to the targeted security domain. There are 11 different security categories that are considered important for OS security evaluation against NIST SP 800-53. These categories, along with the description of selected security criteria for Linux framework are shown in Table 5.2. This categorization is same as done for Windows 10 framework which helps in understanding the security areas of OS in more detail and, therefore, will result in a more focused evaluation approach.

**Table 5. 2 Security Categories and Subsequent Controls**

| Security Categories | Security Requirements | Description |
|---|---|---|
| Logging/Auditing | 5 | It contains SRs related to generation of event logs |
| Cryptographic Solutions | 3 | It contains SRs related to encryption of audit logs and critical data |
| Access Control | 3 | It contains SRs related to access control of audit logs |
| System Information | 2 | It contains SRs related to hardware and software information |
| User Accounts | 7 | It contains SRs related to security and management of user accounts |
| Networks | 12 | It contains SRs related to secure network protocols and operations |
| Hardware | 2 | It contains SRs related to external connected media |
| Software | 9 | It contains SRs related to security of installed software and its execution |
| Notifications and Triggered Events | 1 | It contains SRs related to security notifications in case of configuration change |
| Execution Privileges | 1 | It contains SRs related to execution security of connected media |
| Identification and Authentication | 4 | It contains SRs related to authentication strength and mechanisms |

### 5.2.3 Assignment of Security Weight and Security Level

Every user has different security needs according to their operational environment therefore the OS evaluation methodology must be flexible enough to be adjusted according to needs of every Linux user. For this purpose, we have assigned security weights and security levels to each requirement based on its criticality for the maintenance of user security goals and the security threats faced. We have used three security levels and assigned them corresponding values of security weights in the same way as we did for

Windows 10 SCE framework. The description of security levels and weights can be seen from Table 3.3 in chapter 3.

### 5.2.4 Defining Tests for Evaluation of Linux Ubuntu

This is the core step of our proposed Linux framework that deals with defining tests to check status of each SR in Linux Ubuntu. As stated earlier, the proposed framework is made to be easily adopted by users to self-assess security of their OSs. So, this step focused on designing tests that are easy to perform by even non-technical individuals and does not require any complex tool or system setup. To achieve the same, we have mostly used Linux Terminal commands to check the status of each requirement defined in previous phases of this framework. For the purpose of brevity, all tests cannot be shown here. So, to give a general idea the commands to check a few of the SRs related to "Network" class are shown in Table 5.3. These tests have to be performed on the same computer system containing the Linux Ubuntu OS under evaluation.

**Table 5. 3 Commands to Test Network Related SRs**

| **Security Requirements (SR)** | **Commands** |
|---|---|
| The OS shall be able to provide information regarding firewall configurations. Firewall shall have all the rules defined properly. | **Terminal:**<br>$ sudo ufw status<br>$ sudo ufw status verbose<br>$ sudo ufw status numbered |
| The OS shall be able to provide information about the traffic route. It should be routed through trusted proxy server. | **Terminal:**<br>$ sudo route –n<br>$ netstat –rn<br>These commands display routing table with IP addresses.<br>$ traceroute –n <any website><br>Output contains address of the proxy server. Private IP will indicate the presence of a trusted proxy server. |
| In case of static IP's and DNS implementation the organizations shall preferably have backup server /second server to maintain/cater faults in the system. | **Terminal:**<br>$ sudo cat /etc/network/interfaces<br>For static IP address look for the line "iface eth0 inet static" in the output, it means static IP address is present. This file also contains IP addresses and information regarding DNS servers used. "8.8.8.8" Usually is the primary DNS and any other DNS mentioned will be a backup DNS server. |

| | |
|---|---|
| The OS shall be able to provide information about the services, protocols and ports that are authorized and unauthorized. Unauthorized ones should not be used. | **Terminal:**<br>$ sudo watch netstat –anlp<br>It will display all associated protocols and services and tells the status of the ports whether they are listening on wait or the connection is already established. Check if any unauthorized service or protocol like TOR or HTTP, FTP etc. is running and on which port.<br>$ sudo systemctl list-unit-files --type service –all<br>It will list all the services along with their status. Unauthorized services are mostly masked that won't run until this property is taken away from them.<br>$ sudo netstat –lntu<br>It displays all open ports. Unnecessary and unused ports should be kept closed. |
| The OS shall be able to provide information about the open ports. There should be no unnecessary open ports. | **Terminal:**<br>$ sudo nmap –sT -O <local host ip><br>Display all open TCP ports.<br>$ sudo nmap –sU -O <local host ip><br>Display all open UDP ports.<br>$ sudo netstat –ntlp \| grep LISTEN<br>Will display all the ports that are listening. |
| Extract information about the authentication and encryption of wireless network. It should have static IP and must be password protected. | **Terminal:**<br>$ sudo wpa_cli status<br>This command gives information regarding SSID, BSSID, encryption scheme, IP address and MAC address of the wireless network.<br>$ sudo iwlist wlan0 scan<br>It shows the security of all the networks in range. |
| Time of the local machine should be synchronized with the main server of the organization. | **Terminal:**<br>$ timedatectl status<br>If system clock synchronization and ntp synchronized is ON, then it means system clock is synchronized with main internet servers. If they are off, we can find the difference by using chrony utility and synchronize the clock.<br>$ sudo chronyd –q |
| External insecure connections must be channeled through protected gateway. | **Terminal:**<br>$ ifconfig<br>If the gateway given has a private IP it means, it's a protected gateway otherwise not. |
| Unauthorized connections should be logged. | **Terminal:**<br>$ sudo cat /var/log/auth.log<br>Any connections made or denied will be reported in this log file. |

| | |
|---|---|
| | $ sudo cat /var/log/faillog<br>This file contains all failed connection attempts. |
| The OS shall be able to check whether RDP is enabled or not. Should be preferably disabled. | **Terminal:**<br>$ sudo systemctl status xrdp<br>If the status returned is active it means xrdp is enabled and it will be listening on port 3389. |
| The OS shall be able to give information about the insecure protocols which are configured for use. They should be avoided. | **Terminal:**<br>$ sudo netstat –lntu<br>If any port operating on ftp, http and telnet type services is listening then it means insecure protocols are enabled.<br>$ openssl s_client –connect host:port -ssl3<br>If this command returns the actual certificate, the checked server is vulnerable for poodle attack. Similarly, this can be checked for ssl2 as well.<br>$ openssl s_client –connect host:port tls1<br>This should return the public certificate of connection. |

### 5.2.5 Recording and Classification of Results

After each test has been performed, this phase compiles and classifies results obtained from the previous step. Each SR is allotted one of the three values i.e., '0', '0.5' or '1' depending upon the results obtained from testing. These values and their corresponding descriptions are same as for Windows 10 evaluation framework. Hence their narrative can be seen from Table 3.5. Once these values have been assigned, the evaluated OS is classified into three categories i.e., "fully compliant", "partially complaint" and "not complaint". This classification is also the same as in Windows 10 evaluation framework and can be seen from Table 3.6. After this classification the Linux OS that lie under "fully compliant" and "not complaint" classes are not dealt with any further. However, the subsequent steps are performed for "partially compliant" OSs to deduce their exact compliance score.

### 5.2.6 Calculation of Percentage Compliance

After testing and classification of results, another important phase is calculation of security compliance score and percentage compliance of the evaluated OS. This phase will only consider OSs that fall under "partially compliant" class as other two classes will naturally have 0% and 100% compliance. To calculate compliance score of a single SR, this framework will also use Equation 4 as before which incorporates security weights for flexibility of different users to determine the overall compliance with the provided security

criteria. The values of '$W_i$' and '$S_i$' will be from [0.25, 0.5, 1] and [0, 0.5, 1] set respectively and '$\theta_c$' will always lie between 0 and 1.

In Linux framework, the average compliance score for each class will also be calculated separately to depict the security of OS with respect to that particular security domain. This value will lie within range of 0 to 1. It will be calculated by using Equation 5 as before. This calculation will help users to figure out weak areas of their OS security and would allow them to focus more on strengthening security of those areas which mostly affect their security goals.

The overall compliance score for the Linux OS is also calculated in the same way by using Equation 6. However, in the case of this proposed framework for Linux Ubuntu OS, the value of 'n' is considered to be 49. This calculation will give us a general idea regarding the security of the evaluated Linux Ubuntu OS. A greater value of compliance score corresponds to a higher degree of security as it shows that SRs with greater security weightages have been fulfilled by the evaluated OS. Subsequently, the percentage compliance can also be calculated in the same way as Windows 10 framework. $'\theta cmax'$ will be calculated by considering the OS to be "fully compliant" i.e., value of all '$S_i$' is considered '1'. Compliance score of greater than or equal to 95% is considered to be reasonable secure. Whereas a score less than 95% would indicate that the users need to re-evaluate their OS security.

### 5.3 Conclusion

Currently the proposed Linux framework include tests that are suitable for evaluation of Linux Ubuntu OS, however it can be extended to cover other Linux distributions as well. This framework checks the compliance of Linux OS against the security criteria provided in NIST SP 800-53, but it can be extended in the future to include security requirements of Common Criteria along with its suitable extended packages for more detailed security analysis. The framework can be tailored by any user to meet their security needs with the help of security weights for effective evaluation of operating systems.

# 6 VALIDATION OF PROPOSED LINUX UBUNTU SCE FRAMEWORK

For the purpose of validation and demonstration, we have applied proposed Linux Ubuntu SCE framework on a home-based personal computer to assess its security with erstwhile mentioned criteria. The test PC runs on Linux Ubuntu 20.04 OS and all the security patches are installed till date of evaluation. The first two phases namely extraction and categorization of SRs are same for every device, hence they are not performed again. Application of remaining steps and their results are explained in this chapter.

## 6.1 Assignment of Security Weight and Security Level

This phase varies for each type of user depending upon security objectives and threats faced. The security environment and assumptions for a home-based user will remain same as was for Windows 10 framework, therefore security levels and corresponding security weights to each SR are assigned based on the same ground. Due to limited space only, SRs related to "Network" security class have been shown along with values of assigned security levels and weights in Table 6.1. These requirements will be used throughout the validation for reference of our calculations.

**Table 6. 1 Compliance Score of each SR for Home User**

| **Security Requirement 'SR'** | **Security Level 'L'** | **Security Weight 'W'** | **Security Status '$S_i$'** | **Compliance Score '$\theta_c$'** |
|---|---|---|---|---|
| The OS shall be able to give information about the insecure protocols which are configured for use. They should be avoided. | H | 1 | 1 | 1 |
| The OS shall be able to check whether RDP is enabled or not. Should be preferably disabled. | H | 1 | 1 | 1 |
| The OS shall be able to provide information about the unauthorized connections. There should be no unauthorized connections. | H | 1 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| The OS shall be able to provide information about the wireless connection, it should be secure and authorized. | H | 1 | 1 | 1 |
| Time of the local machine should be synchronized with the main server of the organization. | L | 0.25 | 1 | 1 |
| External insecure connections must be channeled through protected gateway. | H | 1 | 1 | 1 |
| The OS shall be able to provide information about the open ports. There should be no unnecessary open ports. | H | 1 | 1 | 1 |
| Extract information about the authentication and encryption of wireless network. It should have static IP and must be password protected. | H | 1 | 1 | 1 |
| The OS shall be able to provide information about the services, protocols and ports that are authorized and unauthorized. Unauthorized ones should not be used. | H | 1 | 1 | 1 |
| In case of static IP's and DNS implementation the organizations shall preferably have backup server /second server to maintain/cater faults in the system. | M | 0.5 | 1 | 0.5 |
| The OS shall be able to provide information about the traffic route. It should be routed through trusted proxy server. | H | 1 | 0.5 | 0.5 |
| The OS shall be able to provide information regarding firewall configurations. Firewall shall have all the rules defined properly. | H | 1 | 0 | 0 |

## 6.2 Applying Tests for Evaluation

In this phase we performed tests to evaluate all 49 security requirements. Figure 6.1, 6.2, 6.3 and 6.4 show evaluation results of SRs related to network class as mentioned in Table 6.1. According to the results obtained the firewall of test PC is inactive which goes against the security requirement of NIST SP 800-53. Moreover, ports for unsecure protocols like ftp: 21, http: 80 and telnet: 23 were not open whereas ports 53 and 631 used by DNS server and internet printing protocol respectively were seen to be in the listening state. Two DNS servers were configured i.e., 8.8.8.8 as primary and 1.1.1.1 as secondary. The connected wireless network was password protected and secured with WPA and WPA 2 encryption. All network connections were channeled through protected gateway i.e., the gateway had private IP address.

Time of local PC was synchronized with the main server and NTP service was active. The remote desktop service (RDP) was disabled and therefore port 3389 was closed. No unauthorized processes were seen running in the background of test PC. Few of the services were disabled and masked in order to prohibit their use. These results show that some loopholes exist in our test PC's security configuration which could be improved to make our working environment more secure.

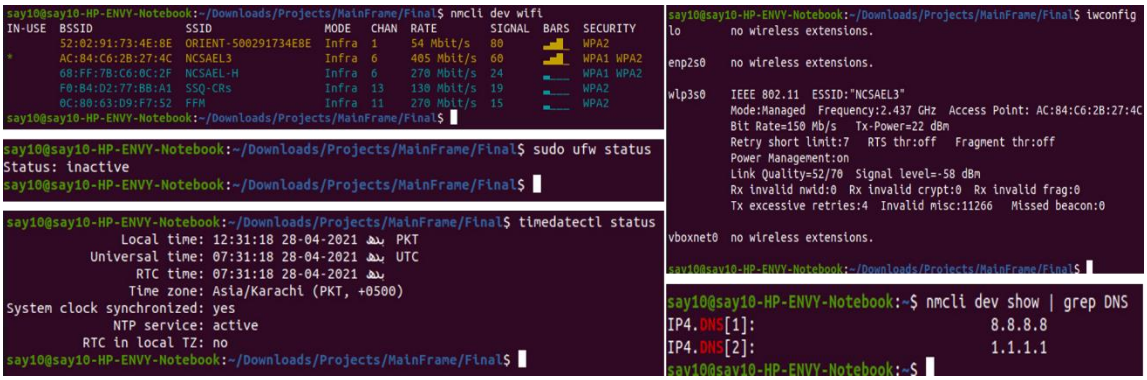


**Figure 6. 1 Results Pertaining to Wireless Network Security, Firewall Configuration, Clock Synchronization and DNS Implementation**

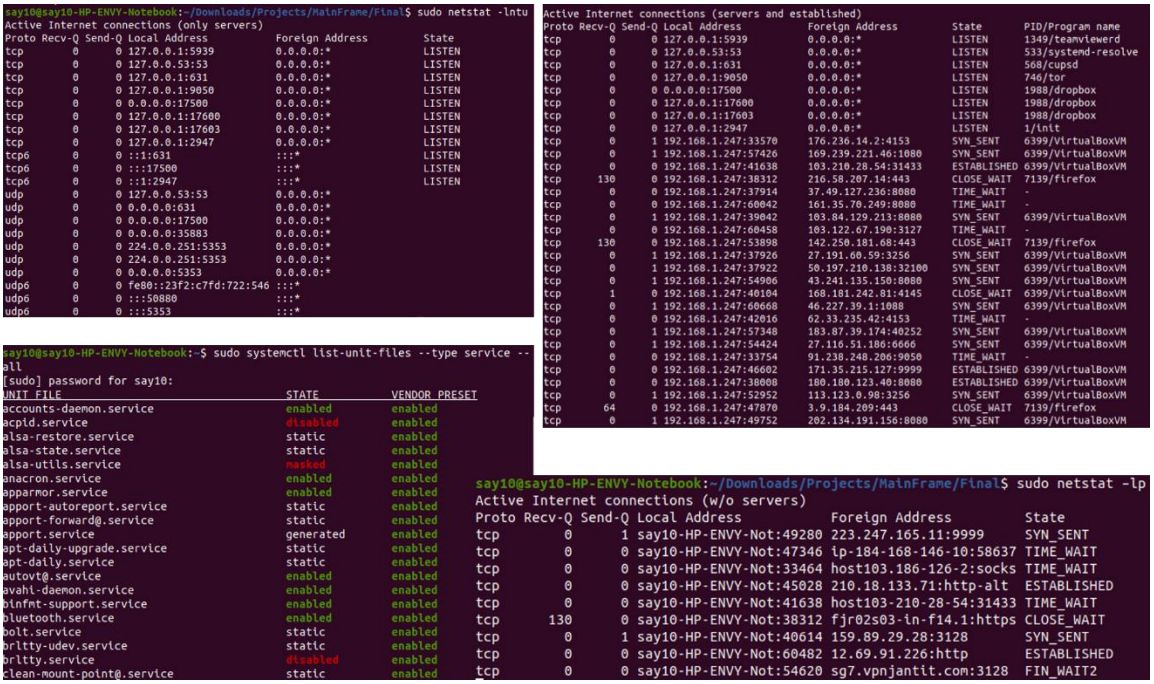


**Figure 6. 2 Results Pertaining to Use of Services, Protocols and Open Ports**

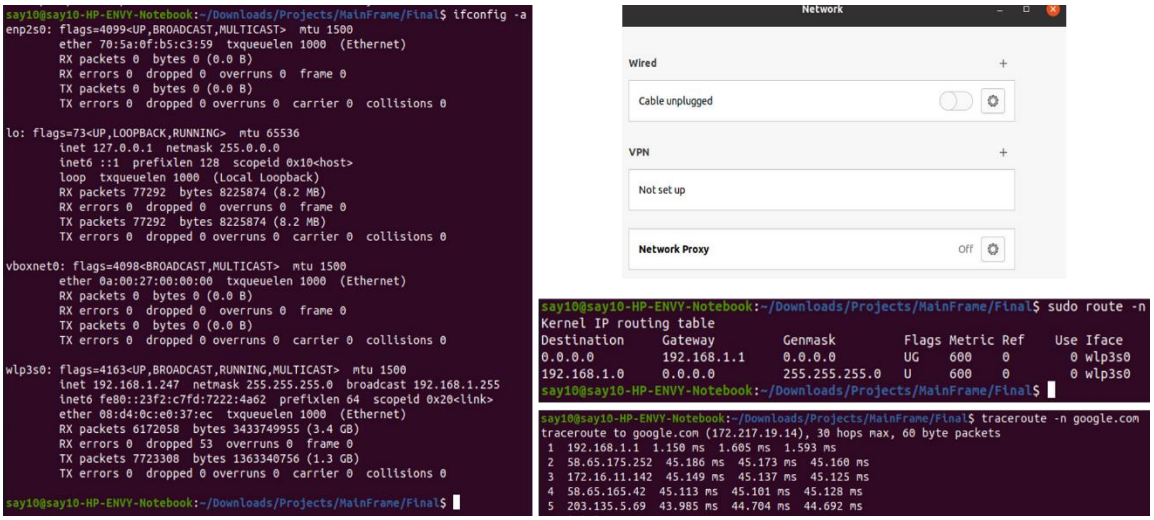


**Figure 6. 3 Results Pertaining to use of Proxy Servers, Protected Gateway and Network Route**

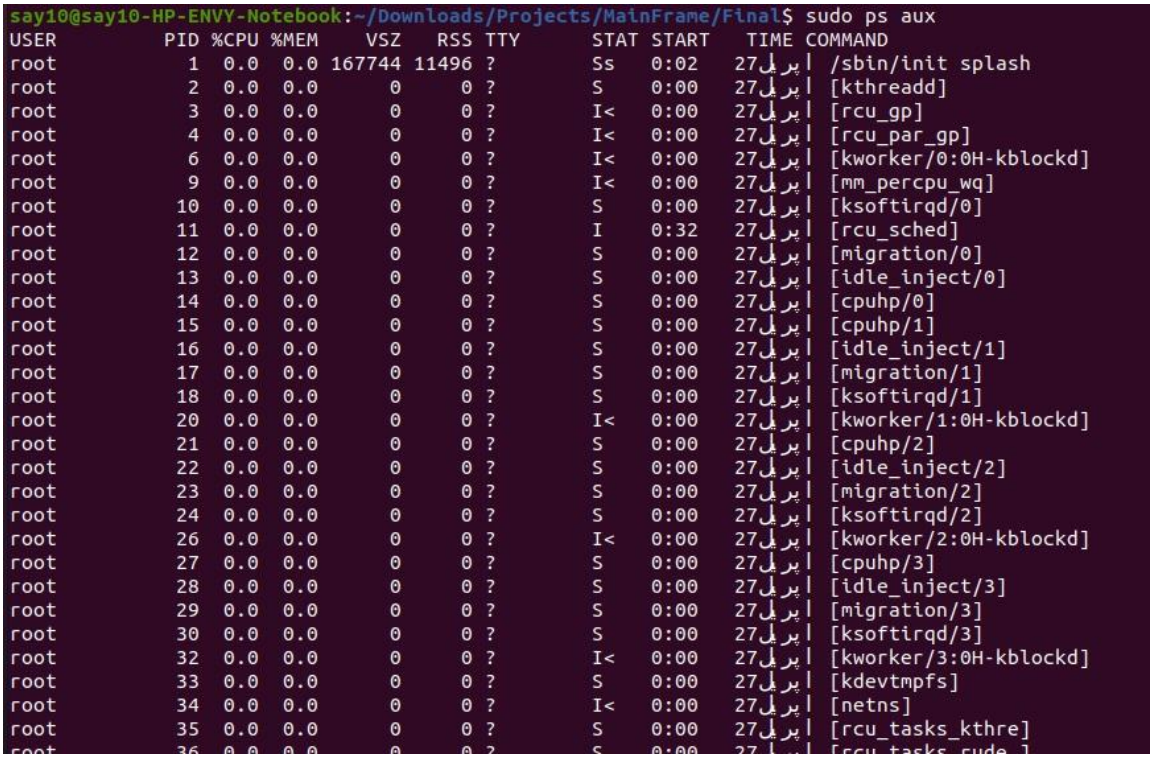


**Figure 6. 4 No Unauthorized Process Running in the Background**

## 6.3 Recording Results of Evaluation

After applying tests and carefully analyzing the results, we have recorded values of '$S_i$' for each SR as can be seen in Table 6.1. These values show that our Ubuntu PC under test is "partially compliant" with the security criteria recommended for Linux framework, as there are some requirements which are either "not fulfilled" or "partially fulfilled".

## 6.4 Calculation of Percentage Compliance

Further calculations are carried out by using the values of security weight and security status. Table 6.1 contains the compliance score for each SR, which is calculated separately by using Equation 4 as before. The average compliance score for 'Network' class is calculated using Equation 5 as below:

$$\theta c(avg) = \frac{10}{12} = 0.833 \quad (20)$$

Table 6.2 contains the values of average compliance score for all security classes. For added precision, the values have been considered up to 3 decimal places. These calculations deduce that the PC under test is more secure in terms of execution privileges and audit logging whereas less secure in terms of access control and authentication

mechanisms. The overall compliance score for the Ubuntu 20.04 OS of the test PC is calculated by using Equation 6 as under:

$$\theta co = \sum_{i=1}^{49} Wi\ x\ Si = 32.5 \quad (21)$$

We have used same Equation 7 for calculating percentage compliance of our PC with recommended security criteria, according to its operational environment which is set by values of security weights assigned to each SR. $'\theta cmax'$ is calculated as follows by considering the ideal situation where all the SRs are completely fulfilled i.e., the value of '$S_i$' for all SRs is considered 1:

$$\theta_{\%} = \frac{32.5}{41} \times 100 = 79.27\ \% \quad (22)$$

The security evaluation of our Ubuntu OS has shown that this PC is not satisfactorily secure in its operation. Especially weak authentication and access control mechanisms render our PC vulnerable against different authentication and authorization attacks that aim at gaining access to confidential data and resources without proper permissions and credentials. This could also lead to privilege escalation. Implementation of weak passwords subject the test PC to various types of dictionary and brute force attacks.

**Table 6. 2 Average Compliance Score of Security Classes for Home Users**

| **Security Classes** | **Number of SRs 'r'** | **Sum of Compliance Score '$\sum_{i=1}^{i=r} Wi\ x\ Si$'** | **Average Compliance Score '$\theta c(avg)'$** |
|---|---|---|---|
| Logging/Auditing | 5 | 3 | 0.6 |
| Cryptographic Solutions | 3 | 1 | 0.333 |
| Access Control | 3 | 0.75 | 0.25 |
| System Information | 2 | 1.5 | 0.75 |
| User Accounts | 7 | 4.75 | 0.678 |
| Networks | 12 | 10 | 0.833 |
| Hardware | 2 | 0.5 | 0.25 |
| Software | 9 | 8 | 0.889 |
| Notifications and Triggered Events | 1 | 1 | 1 |

| | | | |
|---|---|---|---|
| Execution Privileges | 1 | 1 | 1 |
| Identification and Authentication | 4 | 1 | 0.25 |

## 6.5 Conclusion

Validation of our home test PC has revealed that it is only 79.27% compliant with the given criteria in its operational environment, which reflects that all the security features provided by Ubuntu 20.04 that ensures compliance with NIST SP 800-53 are not properly utilized. This security misconfiguration mostly occurs in the domain of cryptographic solutions, external hardware security, access control and identification & authentication mechanisms. Thus, results in privilege escalation to gain unauthorized access to otherwise restricted resources. Moreover, weak password implementation is one of the main reasons for ransomware attacks. In 2019 about 30% of the ransomware infections were the results of using weak passwords [50]. Therefore, it is imperative to ensure at least strong access control and authentication mechanisms for secure operation of operating systems.

*Chapter 7*

# 7 PROPOSED OPERATING SYSTEM SECURITY POLICY

## 7.1 Overview

Security of operating systems is crucial for seamless and secure operation of a wide number of applications. With the increase in complexity of applications and their software, a dire need of users is to keep their operating system security effective against all threats.

## 7.2 Purpose

The purpose of this policy is to provide guidance on the secure usage of operating systems. Adhering to these guidelines will enable a reduced probability of cyberattacks and security breaches by defining best practices that can be adopted by users to keep their data secure from unauthorized access. This will, in turn, fulfill core objectives of cybersecurity i.e., Confidentiality, Integrity, Availability, Non-repudiation and Identity. These guidelines are made for users to comply with universally accepted security standards such as NIST's SP 800-53, ISO 27001 and Common Criteria along with its Extended Packages for VPN, WLAN and SSH.

## 7.3 Scope

This policy covers all security aspects of operating systems like user accounts, system information, networks, cryptography, event logging, data protection, hardware, software, access control, execution privileges, management, identification and authentication. It is applicable to all employees and affiliates of an organization or user and the IT security administrator is responsible for the correct execution of this policy.

## 7.4 Policy

### 7.4.1 Audit Logging

7.4.1.1 Audit logs of all activities performed by the user and system must be maintained. These activities must at least include remote access, system startup and shut down time, authorized/ unauthorized wireless connection attempts, user authentication attempts, modification of access rights to objects, establishment/termination of a wireless session, execution of self-tests and attempts to establish a trusted channel.

7.4.1.2 These logs must contain associated user ID, object ID and detailed description of the event that has taken place along with reliable time stamps.

7.4.1.3 Backup of all event logs, especially security logs, should be maintained and a copy must be kept on separate servers/ database/ hard drives so that if any security incident occurs, the event logs are kept safe and can be accessed later on for forensic analysis.

7.4.1.4 System must generate alerts in case of event log failure. This could occur due to low storage space, and, in such cases, the system must automatically delete oldest logs thereby creating space for newer ones.

7.4.1.5 All event logs must be encrypted and protected from unauthorized modification and deletion. Only an IT administrator must have all access rights to event logs according to access control policy.

7.4.1.6 All event logs must be in human readable format.

### 7.4.2 Cryptography

7.4.2.1 Critical data stored on the internal hard disk/ partitions of the work PC must be encrypted.

7.4.2.2 Operating system must be configured to use cryptographic key sizes for symmetric and asymmetric keys according to NIST FIPs 140-2.

7.4.2.3 Operating system must be configured to use one of the following cryptographic protocols for key exchange: Diffie-Hellman, RSA encrypted exchange of pre-master key, IKE, or Elliptic curve Diffie-Hellman (ECDH).

7.4.2.4 Cryptographic signature services (both generation and verification) must be performed by specific cryptographic algorithms defined in NIST, FIPs 140-2.

7.4.2.5 FIPs policy must be enabled from registry to ensure use of secure cryptographic protocols.

7.4.2.6 Cryptographic hashing service must use SHA-1 or SHA-2 family with message digest size as specified in NIST, FIPs 140-2.

7.4.2.7 Encryption and decryption of critical data must be performed according to algorithms specified in NIST, FIPs 140-2.

7.4.2.8 Keyed hash message authentication must be performed according to algorithms defined in NIST, FIPs 140-2.

7.4.2.9 All authentication credentials and cryptographic keys must be secured with the help of encryption and stored in cryptographic key store provided by operating system.

7.4.2.10 All cryptographic keys and keying material must be destroyed as soon as possible after use when no longer needed.

7.4.2.11 TLS 1.2 protocol must be enabled supporting all mandatory and, one or more, optional cipher suites as defined in RFC.

7.4.2.12 VPN connections must use asymmetric cryptographic keys for IKE peer authentication in accordance with RSA schemes or ECDSA schemes.

7.4.2.13 X.509 v3 certificates must be used to support authentication for IPsec exchanges, digital signatures, integrity checks and EAP-TLS protocol.

*7.4.2.14* Symmetric cryptographic keys for wireless connections must be used that are generated using a random bit generator meeting the requirements of IEEE 802.11-2012 and IEEE 802.11ac-2014 standard.

7.4.2.15 Wireless connection must be configured to use AES key wrap in an EAPOL key frame (that meets the requirements of RFC 3394) for key distribution.

7.4.2.16 Wireless connection must be configured to use TLS 1.2 in support of EAP-TLS protocol and mandatory cipher suite TLS_RSA_WITH_AES_128_CBC_SHA must be supported.

7.4.2.17 For wireless communication, allow only trusted CAs (certification authority) to sign authentication certificates that are accepted by the operating system. IT Administrator is to configure and maintain the list of authorized/trusted certification authorities (CAs).

7.4.2.18 For wireless connections, only IT administrator must configure the list of algorithm suites that may be proposed and accepted during EAP-TLS exchanges.

7.4.2.19 In SSH protocol the encryption/decryption services for data must be in accordance with cryptographic algorithm AES-CTR mode with cryptographic key sizes of 128 or 256 bits.

7.4.2.20 SSH protocol implementation must be configured to support both public key based and password-based authentication methods.

7.4.2.21 SSH transport connection must drop packets greater than 1560 bytes.

7.4.2.22 SSH protocol implementation must support following cryptographic algorithms: aes128-ctr, aes256- ctr, aes128-cbc, aes256-cbc AEAD_AES_128_GCM, AEAD_AES_256_GCM for both SSH client and server.

7.4.2.23 SSH protocol implementation must support following public key generation algorithms: ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384 for both SSH client and server.

7.4.2.24 SSH protocol implementation must support following data integrity algorithms: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM for both SSH client and server.

7.4.2.25 SSH protocol implementation must support following key exchange algorithms: diffiehellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521 for both SSH server and client.

7.4.2.26 SSH transport connection must be rekeyed in case of both client and server under following conditions:

- After no more than $2^{28}$ packets have been transmitted.
- After no more than 1 Gigabyte of data has been transmitted.
- After no more than 1 hour of using the same key.

### 7.4.3 Data Protection

7.4.3.1 The operating system must be configured to run a suite of self-tests during start-up in order to demonstrate the correct operation of all security functions.

7.4.3.2 Configure the operating system to verify the integrity of stored executable code prior to its execution in order to avoid malware invasion and other security breaches.

7.4.3.3 The operating system must have stack-based buffer overflow protection enabled to avoid security incidents.

7.4.3.4 The operating system must always randomize process address space memory locations to prevent attackers from executing their malicious code.

7.4.3.5 Refer to section 4.2 for protection of critical data through cryptographic services.

### 7.4.4 Access Control

7.4.4.1 Persistent and transient storage object access control policy must be defined so as to cover all the list of operations that can be performed on persistent (files, directories and documents etc.) and transient (shared memory and message queues etc.) objects. These policies must be enforced by the operating system in order to give access of objects/resources to subjects/users.

7.4.4.2 Access to objects must be granted on the basis of security attributes of subjects/users as defined in the persistent and transient storage object access control policy.

7.4.4.3 Network information flow control policy must be defined that covers all the operations that cause the information to flow between subjects of the network.

7.4.4.4 Security attributes must be clearly defined for the identification of network data (logical/physical network interfaces, source/destination IP addresses, TCP port number, UDP port number and network protocol).

7.4.4.5 Only the owner of an object must have permission to modify the security attributes of that object covered by the object access control policy.

7.4.4.6 Only IT administrator and authorized users must have the ability to query, create and modify the set of audited events according to object access control policy.

7.4.4.7 Only IT administrator must have the ability to clear or delete the audited events according to object access control policy.

7.4.4.8 Only IT administrator must have the ability to modify the threshold of the audit trail according to object access control policy.

7.4.4.9 Only IT administrator must have ability to modify the actions to be taken in case of audit storage failure.

7.4.4.10 Only authorized users must have the ability to query, modify and delete the security attributes and the allowed set of actions performed on the network data.

7.4.4.11 Only IT administrator must have the ability to modify the threshold for unsuccessful authentication attempts.

7.4.4.12 Only IT administrator must re-enable the authentication to the account subjected to authentication failure.

7.4.4.13 IT administrator must be able to initialize, modify and delete the user security attributes according to object access control policy.

7.4.4.14 Only IT administrator can revoke object security attributes defined by the policy and associated with the corresponding object.

7.4.4.15 IT administrator must configure a list of acceptable wireless networks. Attempts should be made for connection to only those networks which are present on that list.

7.4.4.16 All type of access to registry for all users must be locked except IT administrator.

7.4.4.17 Access to all social media websites must be blocked from organization's PCs.

### 7.4.5 Management

7.4.5.1 IT administrator must have the authority to configure and manage auditing functions, cryptographic network protocols, security attributes, object access control and information flow control policy.

7.4.5.2 IT administrator must manage security functions for both VPN and wireless network connections.

7.4.5.3 Authorized VPN gateways and VPN clients must be defined by IT administrator.

7.4.5.4 MAC addresses of IPsec capable network devices must be specified for use in VPN connections.

7.4.5.5 Security policy for wireless networks must be configured, specifying the certification authorities (CAs) from which operating system will accept WLAN authentication server certificates, security type of WLAN connections, authentication protocol and client credentials used for authentication.

7.4.5.6 Only WPA and WPA2 security based wireless connections must be accepted.

7.4.5.7 Certificate revocation list must be checked during authentication via certificates.

7.4.5.8 Wireless networks (using SSID and security type) must be specified that are allowed for connection.

7.4.5.9 IEEE 802.1X pre-authentication must be supported for wireless network connections.

7.4.5.10 PMK caching may be configured and enabled for wireless connections.

7.4.5.11 Roaming capability must be disabled for wireless connections.

7.4.5.12 The wireless network bridging (hotspot) capability must be kept disabled.

7.4.5.13 Ad hoc wireless client to client connection capability must be disabled.

### 7.4.6 System Information (hardware and software)

7.4.6.1 The user must have detailed knowledge of hardware and software information of the system they work on.

7.4.6.2 User must be aware of the connected external devices (e.g., USB, CD ROM and other media devices) on their systems.

7.4.6.3 Connection of external devices like USB, CD ROM, hard disks, scanners and printers may be disabled from the operating system.

7.4.6.4 Disable automatic execution of CD, DVD, USB or other removable media.

7.4.6.5 User must be aware of the BYOD security policy and USB security policy and should refrain from connecting their personal devices to work PCs.

7.4.6.6 Only digitally signed software must be installed on organization's PCs.

7.4.6.7 Digital lists (blacklist and whitelist) must be maintained to identify authorized and unauthorized software.

7.4.6.8 Allow execution of only digitally signed software that are present on the whitelist and block all unauthorized software.

7.4.6.9 Only authorized personnel must be allowed to install new software on the work PCs.

7.4.6.10 System must verify digital signatures and certificates of installed software before usage.

7.4.6.11 Authorized personnel must keep a record of removed/uninstalled software from work PCs.

7.4.6.12 User must update all installed software and hardware drivers on time.

7.4.6.13 Software and security updates must be automatically installed.

7.4.6.14 User must verify the integrity of updates to software and applications using digital signatures prior to installation.

7.4.6.15 Certificates must be validated along with their certification path, revocation status and certification authority (CA).

7.4.6.16 Secure boot option must be enabled to verify integrity of boot process during initial start-up (power on).

### 7.4.7 User Accounts

7.4.7.1 IT administrator must know all existing accounts on the PC.

7.4.7.2 Every account must belong to a specific user. Shared accounts must be avoided.

7.4.7.3 IT administrator must control all disabled accounts on the PC.

7.4.7.4 There should be no unused active accounts on the PC.

7.4.7.5 IT administrator must keep log of all deleted user accounts on the PC.

7.4.7.6 All unsuccessful login attempts must be recorded in security logs.

7.4.7.7 IT administrator must define installation privileges for every user account.

7.4.7.8 Users must have unique identification on the network in the form of IP address and MAC address.

7.4.7.9 Every user account must have associated security attributes like user identifier (UID), user password, group memberships and group permissions.

7.4.7.10 Multiple authentication mechanisms should be used for user authentication.

7.4.7.11 Only obscured feedback must be provided to user while authentication is in progress.

7.4.7.12 Users must only be authorized to modify their own password and access control permissions for the objects they own.

7.4.7.13 Users must lock their screens before leaving the workspace and they must successfully re-authenticate themselves in order to unlock their account.

### 7.4.8 Identification and Authentication

7.4.8.1 DHCP should be enabled.

7.4.8.2 Security logs must contain information regarding duration of connection to any wireless network.

7.4.8.3 A strong password policy must be configured for username/password-based authentication. Minimum and maximum life of password must be clearly defined, and user must change their account password before expiration.

7.4.8.4 Complex passwords must be implemented with combination of uppercase and lowercase alphabets, numbers and special characters.

7.4.8.5 The same password must never be repeated for any user account.

7.4.8.6 For additional security, biometric authentication must be enabled and include fingerprint recognition, face recognition or iris detection.

### 7.4.9 Networks

7.4.9.1 Insecure network protocols like ftp, http and telnet must not be configured for use.

7.4.9.2 Users may use RDP service for remote sessions.

7.4.9.3 Unauthorized network connections must be identified and blocked.

7.4.9.4 Time of local PCs must be synchronized with the main server of the organization.

7.4.9.5 All network traffic must be channeled through protected gateway (private IP address must be used for gateway).

7.4.9.6 All open ports must be scanned/monitored.

7.4.9.7 Unused ports especially those used by insecure protocols must be kept closed.

7.4.9.8 Only authorized wireless networks with strong authentication and WPA/WPA 2 encryption must be allowed for connection.

7.4.9.9 Use of unauthorized ports, services and protocols must be disabled.

7.4.9.10 Organizations must have a backup/second server in case of static IP and DNS implementation to cater for contingency planning.

7.4.9.11 In case of proxy implementation, the network traffic must be routed through trusted proxy servers.

7.4.9.12 Firewall rules must be configured properly, and all unauthorized network connections must be blocked.

7.4.9.13 By default, all inbound traffic must be blocked if not otherwise allowed in firewall rules.

7.4.9.14 By default, all outbound traffic must be allowed if not otherwise blocked-in firewall rules.

7.4.9.15 VPN should be used for secure communication between peer devices or networks.

7.4.9.16 VPN should be used in either transport or tunnel mode.

7.4.9.17 VPN connections may use IPsec or IKEv1/IKEv2 protocols to provide trusted communication channel between two end points that ensures protection of the data from disclosure and modification.

7.4.9.18 IPsec protocol implementation must use following cryptographic algorithms: AES-GCM-128, AESGCM-256 as specified in RFC 4106 and AES-CBC-128 or AES-CBC-256 (both specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC.

7.4.9.19 IKE protocols must use following cryptographic algorithms: AES-CBC- 31 128, AES-CBC-256 as specified in RFC 6379 and AES-GCM-128, AESGCM-256 as specified in RFC 5282.

7.4.9.20 IKE protocols must use following diffie-hellman groups for key exchange: DH groups 14 (2048-bit MODP), 19 (256-bit Random ECP), 20 (384-bit Random ECP), 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS) and 15 (3072-bit MODP).

7.4.9.21 IT administrator must configure SA (security association) lifetime based on number of packets, number of bytes or length of time for IKE protocol.

7.4.9.22 IKE protocols must perform peer authentication using RSA, ECDSA or Pre-shared keys.

7.4.9.23 X.509v3 certificates must be used to support authentication for EAP-TLS exchanges in wireless connections.

### 7.4.10 Notification and Triggered Events

7.4.10.1 Notification must be sent to system or IT administrator, in case of any change in the configuration of the PC.

7.4.10.2 Appropriate action must be taken by the IT administrator to revert harmful configuration changes in the PC.

## 7.5 Policy Compliance

### 7.5.1 Compliance Measurement

The IT security team of organization will verify compliance with this policy through various methods including, but not limited to, periodic walk through, interviews with employees, business tool reports, number of successful training sessions held on operating system security, internal and external audits, vulnerability assessment reports and feedback to IT security team.

### 7.5.2 Exceptions

Any exception to the policy must be approved by the head of IT security team in advance.

### 7.5.3 Non-Compliance

An employee found to have violated this policy may be subject to disciplinary action up to, and including, termination of employment.

## 7.6 Related Standards, Policies and Processes

**National Institute of Standards and Technology (NIST) publication FIPS 140-2**

**National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53**

**International Organization for Standardization & International Electrotechnical Commission (ISO/IEC) 27001 & 27002**

**National Institute of Standards and Technology (NIST) Special Publication (SP) 800-90A**

**Common Criteria General Purpose Operating System Protection Profile, Version 2.0 & Version 4.2.0**

**Common Criteria NIAP approved Extended Package for Secure Shell (SSH)**

**Common Criteria NIAP approved Extended Package for Wireless Local Area Network (WLAN) clients**

**Common Criteria NIAP approved PP-Module for Virtual Private Network (VPN) Clients**

*Chapter 8*

# 8 CONCLUSION AND FUTURE WORK

## 8.1 Conclusion

Cybersecurity has always been a major concern for any computer systems-based approach at work. With the increase in functionality of these systems and, in turn, on their reliance by users of all shades and scope, cybersecurity takes the center stage when it comes to security aspects of any business model. It is widely understood that complete system dependence cannot be achieved without security of the underlying OS. In the light of this fact, we have proposed two frameworks: one for Windows 10 OS and other for Linux Ubuntu. Both the frameworks provide an effective and budget friendly solution for OS security evaluation. Our research explored and compared some of the existing methods for security evaluation of OSs and proposed a robust technique that is flexible enough to be applied to any organization according to specific security infrastructure requirements. Windows 10 framework calculates security compliance score and percentage compliance of evaluated OS with security guidelines provided by NIST SP 800-53, ISO/IEC 27001-2 and CC along with three extended packages for VPN, WLAN and SSH. Whereas Linux Ubuntu framework calculates security compliance score and percentage compliance of evaluated OS against security guidelines of NIST SP 800-53. In the end an operating system security policy was proposed which would make the organizations compliant with CC, ISO/IEC 27001-2 and NIST SP 800-53 standards if implemented correctly.

The validation of Windows 10 framework is twofold: firstly, a personal home computer was evaluated with 84.44% compliance with the proposed extended security criteria. And secondly our framework evaluated a security research lab system using the data set derived from the lab's customized security policy. Three case studies were used for calculation of percentage compliance with the proposed criteria, and it was demonstrated that its value decreases with an increase in negligence of implementing security policies. This percentage decreased from 91.68% to 72.83% in case of the security research lab system. Whereas the validation of Linux Ubuntu framework was done only on a personal home computer and the evaluated PC was found to be 79.27% compliant with the security criteria defined by NIST SP 800-53.

## 8.2 Future Work

In the future, it is possible to extend scope of our research to include tests for evaluation of other OSs as well and automate the proposed methodology to develop a toolkit that would perform all the tests without any human intervention. This would eradicate the chances of human error and will indicate lapses in existing security protocols more efficiently.

## BIBLIOGRAPHY


[1] D. Gens, “OS-level Attacks and Defenses: from Software to Hardware-based Exploits,” Technische Universität, 2019.

[2] H. Afzali and H. Mokhtari, “A Quantitative Model of Operating System Security Evaluation,” in *Advances in Computer Science, Engineering & Applications*, Springer, 2012, pp. 345–353.

[3] D. L. Nazareth and J. Choi, “A system dynamics model for information security management,” *Inf. Manag.*, vol. 52, no. 1, pp. 123–134, 2015.

[4] S. G. M. Hasnain and F. A. Rafi, “Windows, Linux, Mac Operating System and Decision Making,” Int. J. Comput. Appl., vol. 975, p. 8887.

[5] C. Crane, "The Definitive Cyber Security Statistics Guide for 2020 - Security Boulevard", *Security Boulevard*, 2020. [Online]. Available: https://securityboulevard.com/2020/05/the-definitive-cyber-security-statistics-guide-for-2020/. [Accessed: 13- Apr- 2021].

[6] A. Graham, "The 5 most common cyber-attacks in 2020 - IT Governance UK Blog", *IT Governance UK Blog*, 2019. [Online]. Available: https://www.itgovernance.co.uk/blog/different-types-of-cyber-attacks. [Accessed: 13- Apr- 2021].

[7] B. Canner, "Trend Micro Releases 2020 Midyear Cybersecurity Report", *Top SIEM Vendors, News & Reviews for Security Information and Event Management*, 2020. [Online]. Available: https://solutionsreview.com/security-information-event-management/trend-micro-releases-2020-midyear-cybersecurity-report/. [Accessed: 13- Apr- 2021].

[8] J. Johnson, "Cyber crime incidents by victim industry and size 2019 | Statista", Statista, 2021. [Online]. Available: https://www.statista.com/statistics/194246/cyber-crime-incidents-victim-industry-size/. [Accessed: 13- Apr- 2021].

[9] D. Strickland, "40+ Terrifying Cybersecurity Statistics You Need to Know for 2021", business2community, 2021. [Online]. Available:

https://www.business2community.com/cybersecurity/40-terrifying-cybersecurity-statistics-you-need-to-know-for-2021-02384223. [Accessed: 13- Apr- 2021].

[10] J. Fruhlinger, "Top cybersecurity facts, figures and statistics", CSO Online, 2020. [Online]. Available: https://www.csoonline.com/article/3153707/top-cybersecurity-facts-figures-and-statistics.html. [Accessed: 13- Apr- 2021].

[11] J. Firch, "10 Cyber Security Trends You Can't Ignore In 2021", Purplesec.us, 2020. [Online]. Available: https://purplesec.us/cyber-security-trends-2021/. [Accessed: 13- Apr- 2021].

[12] "Operating System Market Share Worldwide | StatCounter Global Stats", StatCounter Global Stats, 2021. [Online]. Available: https://gs.statcounter.com/os-market-share. [Accessed: 11- Apr- 2021].

[13] "Desktop Operating System Market Share Worldwide | StatCounter Global Stats", StatCounter Global Stats, 2021. [Online]. Available: https://gs.statcounter.com/os-market-share/desktop/worldwide. [Accessed: 13- Apr- 2021].

[14] "Desktop Operating System Market Share Pakistan | StatCounter Global Stats", StatCounter Global Stats, 2021. [Online]. Available: https://gs.statcounter.com/os-market-share/desktop/pakistan. [Accessed: 13- Apr- 2021].

[15] "Desktop Windows Version Market Share Worldwide | StatCounter Global Stats", StatCounter Global Stats, 2021. [Online]. Available: https://gs.statcounter.com/os-version-market-share/windows/desktop/worldwide. [Accessed: 13- Apr- 2021].

[16] A. Sajid, M. A. Shah, M. Kamran, Q. Javaid, and S. Zhang, "An analysis on host vulnerability evaluation of modern operating systems," Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 4, pp. 245–254, 2016.

[17] D. Kostadinov, "Windows 10 Security Features - Infosec Resources", *Infosec Resources*, 2020. [Online]. Available: https://resources.infosecinstitute.com/topic/windows-10-security-features/. [Accessed: 17- Apr- 2021].

[18] "Distribution of malware by OS 2020 | Statista", Statista, 2021. [Online]. Available: https://www.statista.com/statistics/680943/malware-os-distribution/. [Accessed: 13- Apr- 2021].

[19] J. Cohen, "Windows Computers Were Targets of 83% of All Malware Attacks in Q1 2020", PCMAG, 2020. [Online]. Available: https://www.pcmag.com/news/windows-computers-account-for-83-of-all-malware-attacks-in-q1-2020. [Accessed: 13- Apr- 2021].

[20] J. Johnson, "Major operating systems targeted by ransomware 2020 | Statista", Statista, 2021. [Online]. Available: https://www.statista.com/statistics/701020/major-operating-systems-targeted-by-ransomware/. [Accessed: 13- Apr- 2021].

[21] M. Humayun, N. Z. Jhanjhi, A. Alsayat, and V. Ponnusamy, “Internet of things and ransomware: evolution, mitigation and prevention,” Egypt. Informatics J., 2020.

[22] R. Anthony, "NIST SP 800-53 Explained", Cybersaint.io. [Online]. Available: https://www.cybersaint.io/blog/what-is-nist-800-53. [Accessed: 13- Apr- 2021].

[23] "NIST Risk Management Framework | CSRC", Csrc.nist.gov. [Online]. Available: https://csrc.nist.gov/Projects/risk-management/sp800-53-controls/release-search#!/families?version=5.1. [Accessed: 13- Apr- 2021].

[24] "NIST SP 800-53 Full Control List", STIG Viewer | Unified Compliance Framework®. [Online]. Available: https://www.stigviewer.com/controls/800-53. [Accessed: 14- Apr- 2021].

[25] Common Criteria (NIAP), "Protection Profile for General Purpose Operating Systems", Version: 4.1, 2016.

[26] Common Criteria (NIAP), "Extended Package for Secure Shell (SSH)", Version: 1.0, 2016.

[27] Common Criteria (NIAP), "Extended Package for Wireless Local Area Network (WLAN) Clients", Version 1.0, 2016.

[28] Common Criteria (NIAP), "PP-Module for Virtual Private Network (VPN) Clients", Version: 2.1, 2017.

[29] ISO/IEC, "Information technology — Security techniques — Information security management systems — Requirements", ISO/IEC 27001:2005(E), 2005.

[30] ISO/IEC, "Information technology — Security techniques — Code of practice for information security controls", ISO/IEC 27002:2013(E), 2013.

[31] D. Mellado, E. Fernández-Medina, and M. Piattini, “A common criteria-based security requirements engineering process for the development of secure information systems,” Comput. Stand. interfaces, vol. 29, no. 2, pp. 244–253, 2007.

[32] M. Boras, J. Balen, and K. Vdovjak, “Performance Evaluation of Linux Operating Systems,” in *2020 International Conference on Smart Systems and Technologies (SST)*, 2020, pp. 115–120.

[33] H. Afzali and H. Mokhtari, “A Quantitative Model of Operating System Security Evaluation,” in *Advances in Computer Science, Engineering & Applications*, Springer, 2012, pp. 345–353.

[34] Y. Movahedi, M. Cukier, A. Andongabo, and I. Gashi, “Cluster-based vulnerability assessment applied to operating systems,” in 2017 13th European Dependable Computing Conference (EDCC), 2017, pp. 18–25.

[35] F. Alenezi and C. P. Tsokos, “Machine Learning Approach to Predict Computer Operating Systems Vulnerabilities,” in 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), 2020, pp. 1–6.

[36] N. R. Pokhrel, H. Rodrigo, and C. P. Tsokos, “Cybersecurity: time series predictive modeling of vulnerabilities of desktop operating system using linear and non-linear approach,” J. Inf. Secur., vol. 8, no. 04, p. 362, 2017.

[37] I. Khokhlov and L. Reznik, “Android system security evaluation,” in 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2018, pp. 1–2.

[38] I. Khokhlov and L. Reznik, “Data security evaluation for mobile android devices,” in 2017 20th Conference of Open Innovations Association (FRUCT), 2017, pp. 154–160.

[39] L. Recchia, G. Procopio, A. Onofrii, and F. Rogo, “Security Evaluation of a Linux System: Common Criteria EAL4+ Certification Experience,” in *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, 2014, pp. 77–81.

[40] Common Criteria. (2017). Common Methodology for Information Technology Security Evaluation. Version 3.1 Revision 5. Retrieved 19 April 2021.

[41] ENAC Certification, "CERTIFICATION REPORT", DEKRA Testing and Certification S.A.U., 2020.

[42] N. Nissim, R. Yahalom, and Y. Elovici, “USB-based attacks,” Comput. Secur., vol. 70, pp. 675–688, 2017.

[43] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita, “Network attacks: Taxonomy, tools and systems,” J. Netw. Comput. Appl., vol. 40, pp. 307–324, 2014.

[44] S. M. S. ALHusayn, “The Buffer Overflow Attack and How to Solve Buffer Overflow in Recent Research,” AJRSP J., vol. 2, no. 19, pp. 1–13, 2020.

[45] S. Nashimoto, N. Homma, Y. Hayashi, J. Takahashi, H. Fuji, and T. Aoki, “Buffer overflow attack with multiple fault injection and a proven countermeasure,” J. Cryptogr. Eng., vol. 7, no. 1, pp. 35–46, 2017.

[46] P. Luo, D. Zou, Y. Du, H. Jin, C. Liu, and J. Shen, “Static detection of real-world buffer overflow induced by loop,” Comput. Secur., vol. 89, p. 101616, 2020.

[47] S. Mohurle and M. Patil, “A brief study of wannacry threat: Ransomware attack 2017,” Int. J. Adv. Res. Comput. Sci., vol. 8, no. 5, pp. 1938–1940, 2017.

[48] J. P. Tailor and A. D. Patel, “A comprehensive survey: ransomware attacks prevention, monitoring and damage control,” Int. J. Res. Sci. Innov, vol. 4, no. 15, pp. 116–121, 2017.

[49] R. Brewer, “Ransomware attacks: detection, prevention and cure,” Netw. Secur., vol. 2016, no. 9, pp. 5–9, 2016.

[50] N. Hampton, Z. Baig, and S. Zeadally, “Ransomware behavioural analysis on windows platforms,” J. Inf. Secur. Appl., vol. 40, pp. 44–51, 2018.

[51] A. Kharraz, W. Robertson, and E. Kirda, “Protecting against ransomware: A new line of research or restating classic ideas?,” IEEE Secur. Priv., vol. 16, no. 3, pp. 103–107, 2018.

[52] *Securitymagazine.com*, 2020. [Online]. Available: https://www.securitymagazine.com/articles/91572-weak-passwords-caused-30-of-ransomware-infections-in-2019. [Accessed: 29- Apr- 2021].