
**E-ISSN:** 2706-8927
**P-ISSN:** 2706-8919
www.allstudyjournal.com
IJAAS 2024; 6(1): 79-85
Received: 04-11-2023
Accepted: 11-12-2023

**Shubham**
Research Scholar, Om Sterling Global University, Hisar, Haryana, India

**Dr. Rajinder Singh Sodhi**
Professor (CSE), Om Sterling Global University, Hisar, Haryana, India

**Dr. Preet Kaur**
Assistant Professor, JC Bose University of Science & Technology, YMCA, Faridabad, Haryana, India

**Corresponding Author:**
**Shubham**
Research Scholar, Om Sterling Global University, Hisar, Haryana, India

# Measures and security to avoid ethical hacking

**Shubham, Dr. Rajinder Singh Sodhi and Dr. Preet Kaur**

**DOI:** https://doi.org/10.33545/27068919.2024.v6.i1a.1174

**Abstract**
With the increasing reliance on digital technologies in every aspect of life, the threat of malicious hacking has become ever more pertinent. Ethical hacking, also known as penetration testing, involves authorized attempts to exploit system vulnerabilities to identify weaknesses and improve security. However, preventing unauthorized individuals from exploiting these vulnerabilities is crucial to safeguarding sensitive data and maintaining trust in digital systems. This paper explores various measures and security practices aimed at preventing ethical hacking, including robust cybersecurity frameworks, secure coding practices, employee training, and proactive threat intelligence. By implementing these measures, organizations can fortify their defenses against potential breaches and minimize the risk of unauthorized access to critical systems and data.

**Keywords:** Digital, ethical, hacking, cybersecurity, coding, security

## Introduction
In an era defined by digital interconnectedness, the integrity and security of digital systems are paramount. While ethical hacking serves as a proactive means to identify and address vulnerabilities, unauthorized exploitation of system weaknesses can have devastating consequences. Thus, understanding and implementing effective measures to prevent ethical hacking are essential for protecting sensitive data, preserving privacy, and maintaining the integrity of digital infrastructure.

Ethical hacking, or penetration testing, involves authorized attempts to breach a system's defenses to assess its security posture. However, when individuals with malicious intent exploit these same vulnerabilities, the consequences can be severe, ranging from financial losses and reputational damage to legal ramifications and compromised national security.

To mitigate the risk of unauthorized access and prevent ethical hacking, organizations must adopt a multifaceted approach to cybersecurity. This entails implementing robust security frameworks, adopting secure coding practices, and fostering a culture of cybersecurity awareness among employees. Additionally, proactive threat intelligence gathering and continuous monitoring are crucial for staying ahead of emerging threats and vulnerabilities.

This paper will delve into the various measures and security practices aimed at preventing ethical hacking. It will explore the role of cybersecurity frameworks such as the NIST Cybersecurity Framework and ISO/IEC 27001 in establishing comprehensive security controls and risk management processes. Furthermore, it will examine the importance of secure coding practices, including input validation, parameterized queries, and secure authentication mechanisms, in reducing the attack surface and fortifying software defenses.

Employee training and awareness initiatives will also be highlighted as essential components of a robust cybersecurity posture. By educating employees about common security threats, phishing scams, and social engineering tactics, organizations can empower their workforce to recognize and mitigate potential risks effectively.

Moreover, proactive threat intelligence gathering and continuous monitoring will be emphasized as critical elements of a proactive security strategy. By leveraging threat intelligence feeds, security teams can identify emerging threats, vulnerabilities, and attack patterns, enabling timely remediation and threat mitigation.

In summary, this paper aims to provide a comprehensive overview of the measures and security practices essential for preventing ethical hacking. By implementing these strategies, organizations can strengthen their defenses against potential breaches, minimize the risk of unauthorized access, and safeguard their digital assets against malicious exploitation.

**Method of threat assessment**

We used the strategy outlined in the IoT Penetration Testing Cookbook after carefully considering all of the available options and classifying the risks.

1. Identify IoT assets.
2. Decompose the IoT device.
3. Identify threats – by using the STRIDE threat categorization.
4. Score the sensitivity of functions based on attack vector's cost, complexity, reputational impact, repeatability, and damage impact.

Instead of using the DREAD rating system, which is discussed in the book, this technique will score threats according to NCSC. This thesis will show that, like the DREAD grading system, scores will be based on three levels: low, medium, and high.

**Penetration testing methodology**

The books Ethical Hacking and Penetration Testing Guide and IoT Penetration Testing Cookbook repeat attacks and security measures. Continuation or exclusion of steps may occur based on relevance and boundaries. On top of that, the thesis will not cover some more advanced methodologies.

It is most appropriate to combine two established approaches in penetration testing in order to accomplish the following: Methods developed by the International Council of Electronic Commerce Consultants and the National Institute of Standards and Technology for conducting penetration tests, both of which consist of five stages. Combining the two approaches provides a powerful motivation for penetration testing.

The purpose of planning and reconnaissance is to get a complete picture of the intended system or equipment. Such materials may include how-to guides, open-source code, or even details of past assaults. Determine valuable assets to evaluate.

The goal of scanning and discovery is to actively collect more detailed information about the subject by using tools and approaches. Find possible weak spots, measure them, then rank them in order of importance.

Attempting to exploit vulnerabilities and get high privileged access is known as attacking or gaining access. If everything works well, divide the discovery process into more manageable chunks and go from there.

To report an exploit, one must contact the company in question.

Evaluate potential dangers and ways to avoid them.

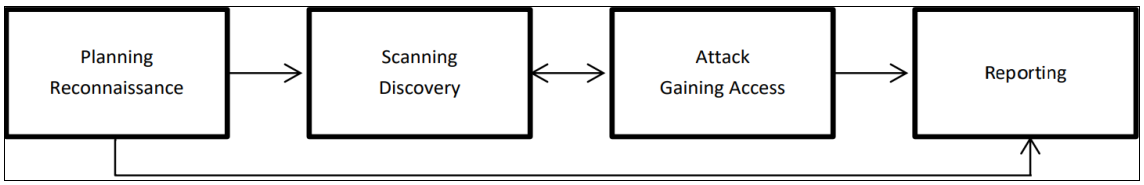


**Fig 1:** The four stages of the combined penetration testing methods

The first stage in acquiring knowledge about the system or gadget is the planning and reconnaissance phase. Separate from active information collecting are passive methods of acquiring data. Gathering useful information about the device in a passive manner is possible even while the targeted system is unaware or disconnected. Consult resources such as online databases, user guides, and lists of known information security flaws and exposures. Sniffing the network is a hands-off way to collect data and useful tools like Wireshark.

**Environment and Utility**

The operating system, Kali Linux, is presented in the book: IoT Penetration Testing Cookbook, which served as the basis for the environment configuration.

Digital forensics and penetration testing are the specialties of Kali Linux, an operating system and distribution based on Debian. Port scanners and packet analyzers are only two of the many preloaded programmes on the platform. Information collecting and testing will take place in this setting.

**Information Gathering Utilities**

The information is gathered using Nmap, a popular tool for identifying target systems. By using scripts that validate unpatched exploits, it may also be used for vulnerability analysis. The goals of data collection on a device include:

1. Finding open and accessible ports.
2. What services the ports are running.
3. If unable to identify service, try identifying the fingerprint.

The last step is to use the information gathered using Nmap to document the findings.

The usage of scripts allows for the discovery of security flaws. Another name for this is a vulnerability assessment. Not the attack itself, but the presence or absence of protection is what is evaluated against the system using well-known vulnerabilities. To determine if a client-server secure connection is legitimate, SSLyze15 is a good tool. It scans the connection for security flaws in the cypher suite. Kali Linux comes with the SSL yze tool already installed.

**Tools for man-in-the-middle**

It is possible to listen in on and study data packets as they travel over a network. To sniff packets, the IoT Penetration Testing Cookbook recommends using TCPDump16. The Wireshark network protocol analyzer may be connected to the TCPDump utility over a pipe. By inspecting the contents of packets-whether they be bits, hexadecimal, or cleartext-it employs a more comprehensive perspective over the data that is sent. Details like the protocol and port numbers, as well as the packet's intent and origin.

Whether you choose to actively or passively analyses protocols (even encrypted ones) exchanged between victims, Ettercap18 has you covered. Methods like DNS spoofing and ARP poisoning are shown and made easier to understand using the tool. Because of this, packets may be manipulated and tainted. In the IoT Penetration Testing Cookbook, the tool is introduced and seen in action.

In the Kali Linux environment, the Macchanger19 tool is used. When attempting to impersonate an external server, this programme may help you fake your network card's

physical MAC address.

**Tools for target system exploitation**

The Nikto20 tool is a brute-force and vulnerability scanner for web servers. It checks whether the server has the standard file structures seen in webservers, checks for missing headers, and checks for Common Gateway Interface (CGI) directories. If the gadget has an accessible webserver, this utility will be useful. In the Kali Linux distribution, you'll find the Nikto utility.

A maven build is the means via which the cf-browser22 tool runs. Requests for CoAP may be sent and received with ease using this tool. A number of RESTful API methods, including GET, POST, PUT, and DELETE, are available. Changes to resources may be specified by attaching requests with payloads. The Hands-on with CoAP course introduces this instrument. A remote computer may establish a connection and exchange data with the host system by means of an Apache server. In testing, this is helpful since it lends credence to the concept of taking use of the target machine's web browser. The Browser Exploitation Framework (BeEF) is a publicly accessible framework that fully supports this idea. An additional usage for an Apache web server is as a proxy server, which controls the flow of requests from clients to servers by acting as a go-between.

**Utilities for Decompiling Mobile Applications**

Participants used the apktool25 and dex2jar26 tools to reverse engineer mobile apps for the 2019 study Security Analysis of Android Application by Using Reverse Engineering. You may use these tools to break down an Android application package (APK) into its component parts, which are classes and functions written in Java. In order to assess the application's security, this is helpful.

**Documentation and Journaling**

Using Keep Note, you may organise your strategies, findings, and assaults. Written descriptions and visual representations of progress are provided by this tool. Finally, when the threat assessment and penetration testing have been completed, a threat traceability matrix will be constructed to analyses the assaults.

**Scanning and Discovery**

This is the second step in doing penetration testing. Discussed before on this method: Use methods and tools to your advantage to learn more about the target. Find possible weak spots, measure them, then rank them in order of importance.

We couldn't go further without more information on the system. Since there was insufficient data from previous scans to make any judgements. Each open port was scanned using Nmap in order to recognised fresh data.

**Single port probing and analysis**

Further evolution of the scans was required in order to probe the ports. Treating each port independently. Initiating on TCP port 17654.

```
# nmap -v -A -p 17654 --version-intensity 9 TIZEN
...
PORT      STATE SERVICE VERSION
17654/tcp open  unknown
| fingerprint-strings:
|   FourOhFourRequest, GetRequest, HTTPOptions, RTSPRequest, SIPOptions:
|     HTTP/1.1 404 Not Found
|     Connection: close
|_    Content-Length: 0
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-
bin/submit.cgi?new-service :
SF-Port17654-TCP:V=7.80%I=7%D=5/4%Time=5EB00F52%P=x86_64-pc-linux-gnu%r(Ge
SF:tRequest,40,"HTTP/1\.1\x20404\x20Not\x20Found\r\nConnection:\x20close\r
SF:\nContent-Length:\x200\r\n\r\n")%r(HTTPOptions,40,"HTTP/1\.1\x20404\x20
SF:Not\x20Found\r\nConnection:\x20close\r\nContent-Length:\x200\r\n\r\n")%
SF:r(RTSPRequest,40,"HTTP/1\.1\x20404\x20Not\x20Found\r\nConnection:\x20cl
SF:ose\r\nContent-Length:\x200\r\n\r\n")%r(FourOhFourRequest,40,"HTTP/1\.1
SF:\x20404\x20Not\x20Found\r\nConnection:\x20close\r\nContent-Length:\x200
SF:\r\n\r\n")%r(SIPOptions,40,"HTTP/1\.1\x20404\x20Not\x20Found\r\nConnect
SF:ion:\x20close\r\nContent-Length:\x200\r\n\r\n");
MAC Address: XX:XX:XX:02:1C:0C (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=260 (Good luck!)
IP ID Sequence Generation: All zeros
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 176.59 seconds
Raw packets sent: 24 (1.850KB) | Rcvd: 16 (1.322KB)
```

**Fig 2:** Partial transcript of probing port 17654/tcp

Figure 2 shows that there was data to analyses, and in order to understand the possible uses of this port, it was necessary to parse the data.

It failed to identify the service that the port was running, according to the results. Nevertheless, upon accepting input, it did provide data along with a fingerprint. The operating system information that was also obtained pointed to the presence of a Linux kernel. For further in-depth analysis, Nmap used fingerprint strings to query the port and wait for a response. It became clear what protocol it was after getting an HTTP response. A webserver was waiting to accept HTTP requests on this port. The fingerprint confirmed that all GET requests returned the same "404 Not Found" error. Additionally, the fingerprint did not provide any indication that a file structure was revealed. It was necessary to go more into this webservice, nevertheless.

The 39164/tcp port was subsequently examined. Although both 51544/tcp and this port were inspected, only one is shown here, they were both temporary ports under the same protocol.

```
# nmap -v -sV --version-intensity 9 -p 39164 TIZEN
...
Discovered open port 39164/tcp on 192.168.1.134
rDNS record for 192.168.1.134: TIZEN.XXX.XXX.se

PORT      STATE SERVICE     VERSION
39164/tcp open  ssl/unknown
...
```

**Fig 3:** Output of scanning the port 39164/tcp

Using SSL/TLS encryption with a client-server connection via dynamic ports was shown in Figure 3. The programme SSLyze was used to get more detailed information.

```
# sslyze 192.168.1.134:39164 --regular

 SCAN RESULTS FOR 192.168.1.134:39164 - 192.168.1.134
 ---------------------------------------------------

 * Downgrade Attacks:
       TLS_FALLBACK_SCSV:                OK - Supported

 * TLSV1_1 Cipher Suites:
      Server rejected all cipher suites.

 * OpenSSL CCS Injection:
                                         OK - Not vulnerable to OpenSSL CCS injection

 * SSLV2 Cipher Suites:
      Server rejected all cipher suites.

 * SSLV3 Cipher Suites:
      Server rejected all cipher suites.

 * Session Renegotiation:
Unhandled exception while running --reneg:
SslHandshakeRejected - TLS / Alert: handshake failure

 * OpenSSL Heartbleed:
                                         OK - Not vulnerable to Heartbleed

 * TLSV1 Cipher Suites:
      Server rejected all cipher suites.

 * Deflate Compression:
                                         OK - Compression disabled
```

```
 * TLS 1.2 Session Resumption Support:
      With Session IDs:                  NOT SUPPORTED (0 successful, 5 failed, 0 errors, 5
total attempts).
      With TLS Tickets:                  NOT SUPPORTED - TLS ticket not assigned.

 * TLSV1_3 Cipher Suites:
      Server rejected all cipher suites.

 * Certificate Information:
     Content
       SHA1 Fingerprint:                 89a0c8a2236590efe9bfccde1a91d281247db59d
       Common Name:                      OCF Device: Appliance TZ (7aa75009-682e-4245-9720-
ec8cb31afe2a)
       Issuer:                           Samsung Electronics OCF HA Device SubCA v2
       Serial Number:                    412501082447657828913679542643043683923292993848
       Not Before:                       2019-06-04 04:40:08
       Not After:                        2069-12-31 14:59:59
       Signature Algorithm:              sha256
       Public Key Algorithm:             EllipticCurve
       Key Size:                         256
       Curve:                            secp256r1
       DNS Subject Alternative Names:    []

     Trust
       Hostname Validation:              FAILED - Certificate does NOT match 192.168.1.134
       Android CA Store (9.0.0_r9):      FAILED - Certificate is NOT Trusted
       Apple CA Store (iOS 13, iPadOS 13, macOS 10.15, watchOS 6, and tvOS 13):FAILED -
Certificate is NOT Trusted
       Mozilla CA Store (2019-11-28):    FAILED - Certificate is NOT Trusted
       Windows CA Store (2019-11-10):    FAILED - Certificate is NOT Trusted
       Java CA Store (jdk-13.0.2):       ERROR: SslHandshakeRejected - TLS / Alert:
handshake failure
       Symantec 2018 Deprecation:        OK - Not a Symantec-issued certificate
```

```
       Received Chain:                   OCF Device: Appliance TZ (7aa75009-682e-4245-9720-
ec8cb31afe2a) --> Samsung Electronics OCF HA Device SubCA v2
       Verified Chain:                   ERROR - Could not build verified chain
(certificate untrusted?)
       Received Chain Contains Anchor:   ERROR - Could not build verified chain
(certificate untrusted?)
       Received Chain Order:             OK - Order is valid
       Verified Chain contains SHA1:     ERROR - Could not build verified chain
(certificate untrusted?)


 * ROBOT Attack:
                                         OK - Not vulnerable, RSA cipher suites not
supported

 * TLSV1_2 Cipher Suites:
       Forward Secrecy                   OK - Supported
       RC4                               OK - Not Supported

     Preferred:
        TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256                    128 bits
     Accepted:
        TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384                    256 bits
        TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384                    256 bits
        TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256                    128 bits
        TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256                    128 bits
        ECDHE_ECDSA_WITH_AES_128_CCM_8                             128 bits
        ECDHE_ECDSA_WITH_AES_128_CCM                               128 bits


 SCAN COMPLETED IN 18.05 S
 -------------------------
```

**Fig 4:** Transcript of using SSLyze on the port 39164/tcp

Be aware that the SSLyze tool was validating server certificates, scanning for weak cyphers, and looking for vulnerabilities in the SSL protocol service.

A number of typical attacks were tried against the SSL/TLS protocol throughout the vulnerability investigation. These included: TLS downgrade, Heartbleed, CCS injection, CRIME, unsecured renegotiations, encryption liability, and compression leak. It was determined that the system and its SSL protocol did not have any of these vulnerabilities or exploits.

In conclusion, this part uncovered new ground that may be exploited to launch further assaults using scans and port probing.

**Attack and Gaining Access**

Exploitation and payload-based system access is the primary objective of the first penetration test. The theories, methodologies, and approaches used, as well as the assault surfaces and entrance points, are all detailed in this part. The section concludes with the results and discussion of findings -Parts of this episode show the several entry points that the gadget has.

**Access User and Application Data**

A locally hosted website will serve as the basis for testing, and the target system enters it. The goal of this webserver is to store malicious scripts and activities that the target machine's web browser (Samsung internet browser) might be susceptible to.

**Same-origin policy bypass**

The need to encapsulate scripts and documents built from a separate origin is a principle that is often seen in browsers. Since the protocols are different, http://www.example.com/ cannot connect with https://www.example.com/. Protocol, hostname, and port origin chain verification is an important part of this policy's proper implementation.

In order to conduct this penetration test, a hosted website must have JavaScript code. The script function has to be set to execute when an HTML or PHP button is clicked.

To begin, the script is executed in a typical manner on the malicious website, without accessing the legitimate website via a tab. This ensures that the JavaScript functionality is tested without attempting to circumvent the same-origin policy. Finally, try opening a new tab using the window. Open () method instead of the previous one; this time, you're hoping the malicious code is still running in the background.

Regardless of whether redirects were used or not, the execution of the JavaScript failed. It was confirmed that the Content Security Framework and same-origin policy were present since the function seizes to run when a new tab is opened. To ensure an accurate assessment of this policy's existence, certain script adjustments were implemented: linking the malicious website's document. Domain to the real website, in order to encourage the genuine origin to remain constant throughout. However, this had no discernible impact. The fact that the assault or bypass didn't succeed meant that the browser was current. As mentioned before, this assault relied on a publicly known CVE; nonetheless, it was necessary to verify and authenticate that the browser was really running version 1.0 due to the user-agent fingerprint. Since an attack might enable hostile websites to utilised scripts in other sites' Document Object Model (DOM), it is crucial to preserve the same-origin policy. The attacker might thus monitor and manipulate data coming from many sources, jeopardising any kind of secrecy.

**Conclusion**

Ethical hacking is a tool for data protection and prevention. Due to the proliferation of mobile devices, tablets and smartphones and the large number of applications, the phenomenon of computer insecurity has increased considerably and therefore these are highly vulnerable, because of the above, what is intended with this article is to be constantly ahead of those who try to attack us by doing their own tests and attacks with the help of computer experts.

A new device is not that it is so remotely vulnerable, if the user makes an adequate handling of the phone without connecting to insecure networks, much less entering passwords on sites that do not handle encryption security that make the device an attack target for the attacker can steal information, however the beginning of the attacks is due to the bad manipulation of the user, nor does it serve to have port blocking by default or the deletion of permissions to install unknown applications if the user gives permissions without reading or having knowledge of what is which is installing making the phone's security vulnerable.

For this reason an ethical hacker makes 'pentests' or penetration tests, these tests are composed of a set of methodologies and techniques. These methodologies and techniques reproduce access attempts from different points of entry of a computer environment, the primary objective is to find vulnerabilities in order to circumvent the security of the system by escalating privileges, finding errors and bad configurations, for which it uses both his knowledge in computer science as a wide range of tools, and in this way, pass a report so that measures are taken and thus reduce the risk in an organization.

**References**


1. Sharma N. A Study of the awareness level on Increasing Cyber Crime against Adolescents in Delhi Govt. DIETs. International Journal of Current Research and Trends. 2020 Sep;8(9):2320-2882. Available from: www.ijcrt.org
2. Anisha MS. Awareness and strategy to prevent Cybercrimes: An Indian perspective. Information Technology. 2017 Apr;7(4):2249-555X. IF: 4.894. IC Value: 79.96.
3. Malviya GK. Cyber Crime and Cyber Security. International Journal of Trend in Scientific Research and Development (IJTSRD). 2022 Mar-Apr;6(3):2456-6470. Available from: www.ijtsrd.com
4. Sharif, Haris M, Mohammed, Mehmood. A literature review of financial losses statistics for cyber security and future trend. World Journal of Advanced Research and Reviews. 2022;15:138-156. DOI: 10.30574/wjarr.2022.15.1.0573.
5. Phillips K, Davidson JC, Farr RR, Burkhardt C, Caneppele S, Aiken MP, *et al*. Conceptualizing Cybercrime: Definitions, Typologies and Taxonomies. Forensic Sci. 2022;2:379-398. DOI: 10.3390/forensicsci2020028
6. Perwej Y, Abbas SQ, Dixit JP, Akhtar N, Jaiswal AK. A Systematic Literature Review on the Cyber Security. International Journal of Scientific Research and Management. 2021 Dec;9(12):669-710. DOI: 10.18535/ijsrm/v9i12.ec04ff. hal-03509116
7. Mishra A, Alzoubi YI, Anwar MJ, Gill AQ. Attributes impacting cybersecurity policy development: An evidence from seven nations. Computers & Security. 2022;120:102820. DOI: 10.1016/j.cose.2022.102820
8. Sood A, Bansal R, Enbody RJ. Cybercrime: Dissecting the State of Underground Enterprise. IEEE Internet Computing. 2013;17:60-68.

DOI: 10.1109/MIC.2012.61
9. Mohammed M, Mohammed MA, Mohammed VA. The use of Blockchain in the Management of COVID-19 Vaccine Data. 2023;3:30-35.
10. Gupta K, Jiwani N. Cybersecurity Framework in Healthcare Sector and Techniques to Mitigate and Detect Attacks. Journal of Xidian University. 2022;16:516-521. DOI: 10.37896/jxu16.9/047
11. Mohammed M, Mohammed MA, Mohammed VA. Impact of Artificial Intelligence on the Automation of Digital Health System. International Journal of Software Engineering & Applications. 2022;13:23-29. DOI: 10.5121/ijsea.2022.13602
12. Thatikonda R. Effective Secure Data Agreement Approach-based cloud storage for a healthcare organization, 2023, 3.
13. Fazal S, Rehman U, Ansari MF. Predicting Customer Satisfaction of Online Shoppers Using AI -A Theoretic Framework. IJARCCE, 2023, 12. DOI: 10.17148/IJARCCE.2023.12112
14. Chagadama J, Luamba D, Mutamba I. Cyberattacks: A Huge Concern for Small Business Sustainability. 2022;1:60-75.
15. Alawida M, Omolara AE, Abiodun OI, Al-Rajab M. A deeper look into cybersecurity issues in the wake of Covid-19: A survey. Journal of King Saud University - Computer and Information Sciences. 2022;34(10A):8176-8206. DOI: 10.1016/j.jksuci.2022.08.003