The importance of Ethical Hacking tools and techniques in Software Development Life Cycle

Avito Da Silva

Lusófona University, Porto. Rua Augusto Rosa n24 4000-098 Porto-Portugal
alexandreavito@gmail.com

Abstract. When developing software, it is important to develop secure software before deploying and most of the time developers inject vulnerable code into the software without realizing it. It is very hard to have perfectly secure software, but it is possible to minimize most of the risks and vulnerabilities. For instance, a hacking attack on a company's information system for example can generate a lot of expenses and problems like leaking or losing meaningful data, reports, and sensitive client information.

Over the years, technology has been increasing its presence all around the world and now we use the software in almost every place like hospitals, local shops, supermarkets, in the industry, and many more. So ethical hackers use a variety of tools and techniques to simulate a hacking attack to try to find risks and vulnerabilities and minimize them if they find any to protect the software so that a company or user can use it safely.

In this paper, we are going to discuss what is ethical hacking, who is an ethical hacker, the tools, and the techniques ethical hackers use, and how they can interact with the software development life cycle.

Keywords: Ethical Hacking, Software Development, SDLC, Software Security, Software Vulnerabilities, Mobile Applications, Web Application, Embedded System

1 Introduction

Developing software is not the easiest task and takes a lot of resources like developers, managers, designers, analysts, hardware, money, and time. The most difficult resources to manage to develop a secure software are time and money because the longer it takes to develop the software the more expensive it gets and customers most of the time try pay the minimum possible which sometimes takes the time of the software testing because developers are more focused on implementing all the requirements that when it comes to testing phase there isn't not too much time to make all the tests. Einstein once said "A clever person solves problems. A wise person avoids it" and when it comes to software security, we should avoid all possible vulnerabilities during the development because after the deployment it will be more expensive and could also lead to some serious problems, for instance hacking an embedded sys-

tem such as aircraft can be used as an act of terrorism by taking it down and consequently arm people.

The remainder of this paper is organized as follows: Section 1 "Software development" is talked about what are processes of software development and is life cycle, the risks associated with not developing a secure software now a days and costs of data breaching.

In section 3 "Ethical Hacking" is addressed what is ethical hacking, who is an ethical hacker, tools and techniques used in the ethical hacking phases and importance of ethical hacking in mobile applications, web applications and embedded software.

2 Software development

Software Development refers to a set of computer science activities dedicated to the process of creating, designing, deploying, and supporting software. Development involves the tools, methodologies, and processes necessary to create software, it also concerns the code and algorithms that physicists, device fabricators, service scientists, chemists, and hardware makers need to write in the course of doing their work. Software development also involves the activities of skilled individuals who develop project-specific software code even though they themselves are not primarily software developers [6].

All software projects go through the phases of requirements gathering, business analysis, system design, implementation, and quality assurance testing.

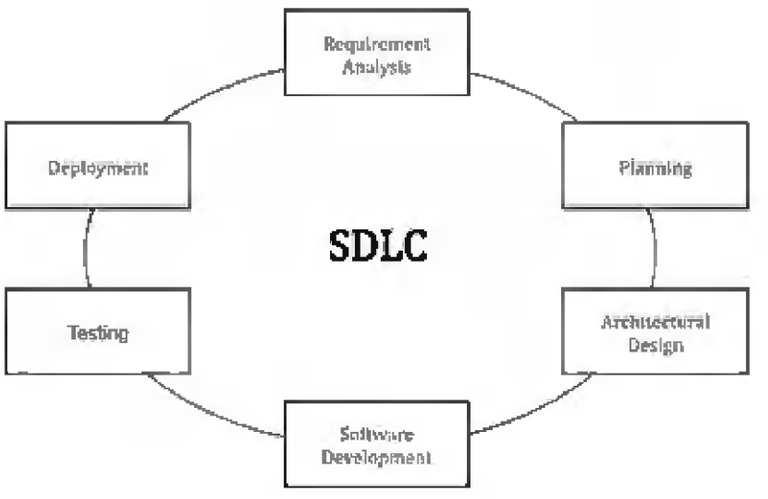

Fig. 1. Software development life cycle [2]

2.1 The risks of developing a non-secure software now a days

We live in a digital era where we use software almost all day from the simplest applications like online shopping, social media, listening to an audiobook or music through most complex ones like hospital systems, organization portals, business process management tools or software to control devices or even industries machine. These software or applications capture and generate a lot of data and most people need this critical data available and secure for their day-to-day activities or jobs.

The total amount of data created, captured, copied, and consumed globally increased rapidly over the years reaching 64.2 zettabytes in 2020. Over the next five years up to 2025, global data creation is projected to grow to more than 180 zettabytes [4].

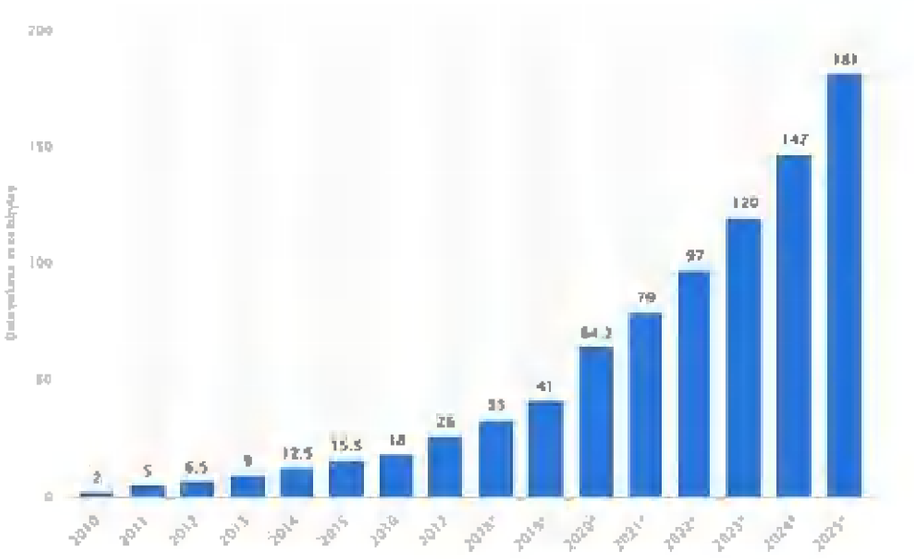

Fig. 2. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025

As we can see in the figure 2, with the amount of data increasing every year the necessity of securing and maintaining the integrity of this data is increasing and below in figure 3 is a report of the data breach cost in euros during the years of 2014 – 2019.

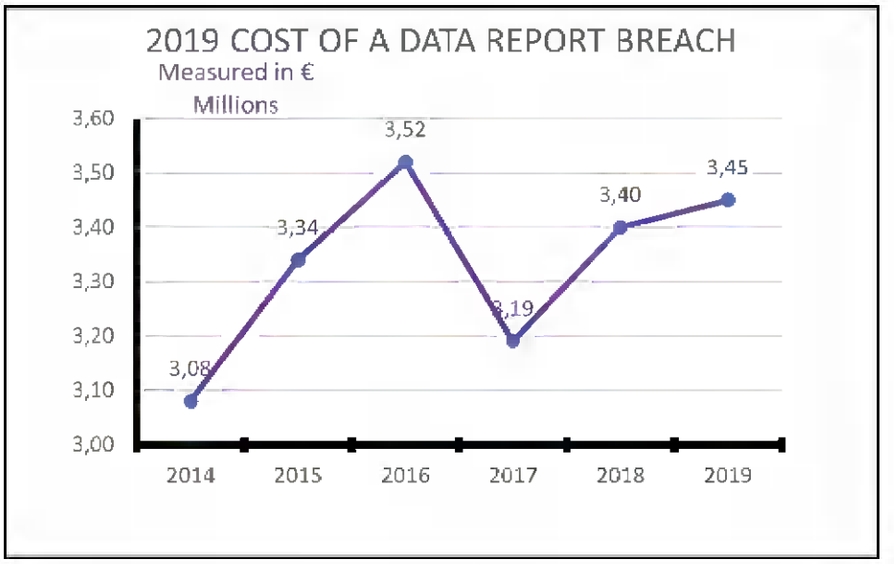

Fig. 3. Cost of a data breach report [1]

A malware, or malicious software, is any piece of software that was written with the intent of doing harm to data, devices, or to people. Types of malwares include computer viruses, trojans, spyware, ransomware, adware, worms, file-less malware, or hybrid attacks. Recent malware attacks have become more sophisticated with the advent of machine learning and targeted spear-phishing emails [1].

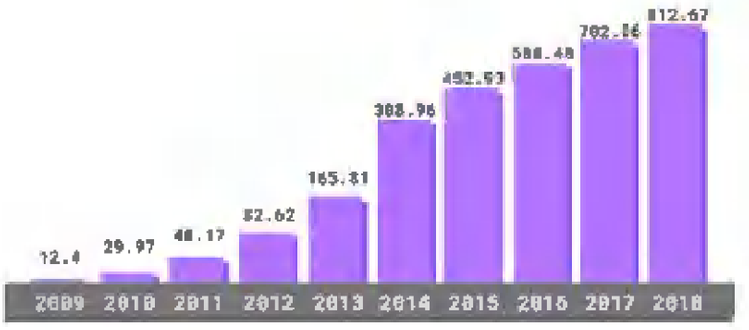

Fig. 4. Total malware infection growth rate in millions

Some vulnerabilities can come from a bad database design, weak implementation (both frontend and backend) and weak testing.

The testing phase is very important and should always be scheduled so he goes as planned that is why is very important to develop a testing report where all the tests

that are going to be applied to the software should be documented. Is also a good idea to have a maintenance plan report, in this report it should be documented how the software will be maintained and things to do if something happened for instance if an hacker gained access to system it should be documented what the organization should do to minimize the impact or solve this particular issue or any other possible issues.

3 Ethical hacking

Is an authorized attempt to do penetration testing on a company's system before the attempt is written a contract where the company specifies what the ethical hacker can attempt to hack, and the ethical hacker should not try to hack any component outside the ones specified in the contract.

Who is an ethical hacker?

The EC-Council, the leading cyber security professional certification organization, defines an ethical hacker as "an individual who is usually employed with an organization and who can be trusted to undertake an attempt to penetrate networks and/or computer systems using the same methods and techniques as a malicious hacker." Sometimes ethical hackers come from the "dark side" after repaying their debt to society, but you can also learn ethical hacking skills in a classroom setting and become certified [8].

3.1 Ethical hacking phases and tools

Are the phases and tools an ethical hacker can follow to perform the hacking attack. They are like the hacking phases the only difference is that an ethical hacker after performing the attack it should analyze the test result and produce to the organization an improvement plan to minimize the risks if they exist. It is not required to sequentially follow these phases.

In this phase it will be shown the tools and techniques that are used in each phase of the hacking attack and how to avoid some of these attacks.

There are two most known operating system hackers or ethical hackers tend to use:

Kali Linux is a Linux distribution for ethical hackers, hackers and computer forensics that comes with a pre-installed with an arsenal of hacking tools and has some fantastic features like full customization of kali ISOs which means it can be customizable to a person needs, usb live boot, kali undercover which changes the look of the environment to a Windows10 desktop environment, and many more [9].

Parrot OS – is another famous operating system when it comes to cybersecurity and computer forensics, it is secure, lightweight, free and open source and comes preinstalled with some IDE's and Compilers which helps IT teams of all sizes develop software and perform security-related tasks such as computer forensics, penetration testing, cryptography, hacking or reverse engineering [10].

Reconnaissance. According to the book "Ethical Hacking and Countermeasures" it's the phase where the hacker gathers as much information as possible about the target

or system and carefully plans the attack. Part of this reconnaissance may involve social engineering. A social engineer is a person who convinces people to reveal information such as unlisted phone numbers, passwords, and other sensitive information. Another technique is Dumpster Diving. Dumpster diving is, simply enough, looking through an organization's trash for any discarded sensitive information.

When an attacker is using passive reconnaissance techniques, he or she does not interact with the system directly. Instead, the attacker relies on publicly available information, social engineering, and even dumpster diving as a means of gathering information.

Active reconnaissance techniques, on the other hand, involve direct interactions with the target system by using tools to detect open ports, accessible hosts, router locations, network mapping, details of operating systems, and applications. Active reconnaissance is usually employed when the attacker discerns that there is a low probability that these reconnaissance activities will be detected [3].

Social engineering can be avoided by implementing good security policies describing how to act correctly with the organizations resources or data and regularly perform information systems audit.

Some tools used during this phase:

- Network Mapper (Nmap): free and open-source tool used for discovering available network hosts, what applications and the operating system they are running.
- Whois lookup: Contains personal information of the domain owners, the database is maintained by Regional Internet Registries. Can be used to get domain name details, contact details of the domain owner and domain name servers.

Scanning. Here the hacker/attacker uses gathered information in the reconnaissance phase to identify specific vulnerabilities, this phase can be considered a logical extension of an active reconnaissance. Often the reconnaissance and scanning phases overlap, and it is not always possible to separate the two [3].

The tools listed below can search for thousands of vulnerabilities and most of them verify web apps, databases, some of them verify networks and servers. They should be used during the development of the software and after the deployment so the vulnerabilities can be minimized if they exit.

Table 1. Scanning tools

Tool	**Short description**
Metasploit	One of the most famous and powerful vulnerabilities tools it is open source which means it can be customized. Metasploit contains a lot of tools that can help the ethical hacker execute attacks and evade detection
Netsparker	Paid web application capable of delivering auto verification of vulnerabilities in web applications or integrating security testing into the entire SDLC of the web app and creating a scan report summary

W3AF	Free and open source also known as Web Application Attack Framework. It´s a user-friendly app that can secure web apps by finding and exploiting vulnerabilities.
Nikto2	An open-source web application that focuses on web applications security and is capable of scanning web servers
Acunetix	Paid web application comes with many functionalities, capable of finding web applications and network vulnerabilities
OpenVas	Powerful tool, suitable for the organization. Can find vulnerabilities in databases, operating systems, networks, and virtual machines.

Gaining Access. In this phase the hacker will use the information he knows about the system to attempt to gain access. If the system doesn't have a robust implementation hackers can cause some damage to system just by trying to penetrate it. Knowing the system, you're penetrating makes it easier to for hackers to hack, that's why is important have a secure implementation in the software.

For instance, external denial-of-service attacks can either exhaust resources or stop services from running on the target system. Service can be stopped by ending processes, using a logic bomb or time bomb, or even reconfiguring and crashing the system. Resources can be exhausted locally by filling up outgoing communication links.

Attackers use a technique called spoofing to exploit the system by pretending to be a legitimate user or different systems. They can use this technique to send a data packet containing a bug to the target system to exploit a vulnerability. Packet flooding may be used to remotely stop the availability of essential services. Smurf attacks attempt to cause users on a network to flood each other with data, making it appear as if everyone is attacking each other, and leaving the hacker anonymous [3].

Hackers sometimes get into an organization system by the organization employee, they can send a phishing email or crack the password and get into the system, to crack the password they can use join brute force attack with social engineering, trojan, spyware keyloggers and maybe man in the middle.

Maintaining Access. After gaining access the ethical hacker should work to try to keep the access, he can keep a low profile and keep exploring the current system or attack other systems.

Attackers, who choose to remain undetected, remove evidence of their entry and install a backdoor or a Trojan to gain repeat access. They can also install rootkits at the kernel level to gain full administrator access to the target computer. Rootkits gain access at the operating system level, while a Trojan horse gains access at the application level [3].

Trojans can be avoided by automatically updating the operating system and the antivirus, the applications should also be updated to avoid security flaws, avoid suspicious sites and emails, and use complex passwords.

Rootkits can be avoided by scanning the system, updating, and avoiding suspicious links.

Covering or clearing tracks. For obvious reasons, such as avoiding legal trouble and maintaining access, attackers will usually attempt to erase all evidence of their actions. Trojans such as ps or netcat are often used to erase the attacker's activities from the system's log files. Once the Trojans are in place, the attacker has likely gained total control of the system. By executing a script in a Trojan or rootkit, a variety of critical files are replaced with new versions, hiding the attacker in seconds. Types of Hacker Attacks 1-9 Other techniques include steganography and tunneling. Steganography is the process of hiding data in other data, for instance image and sound files. Tunneling takes advantage of the transmission protocol by carrying one protocol over another [3].

3.2 The importance of ethical hacking in web applications

When it comes to application software web applications are the ones who suffer most hacking abuse since they can be accessed by everyone at any time and place so is important to conduct security testing during the testing phase of the SDLC and after the application deployment. To minimize the probability of successful attacks, software engineering teams must apply the effort necessary to introduce adequate security precautions. Achieving this goal is only possible by using various techniques and tools to ensure security in all phases of the software product's development life cycle [12]. The automated testing such as OWASP ZAP and Nikto tool is used to detect weaknesses in network infrastructure and web application [11].

The two most common risks in the Web environment, injection—namely SQL injection, which lets attackers alter SQL queries sent to a database—and cross-site scripting (XSS), are also two of the most dangerous (www.owasp.org/ index/Category:OWASP_Top_Ten_Project)[12].

SQL Injection is a database attack and a manifestation of the existence of flaws in WEB application programs. The mechanism of attack is to insert the user data into the actual database manipulation language by using the peripheral interface of some database. In this way, the goal of invading the database and even the operating system can be realized [5].

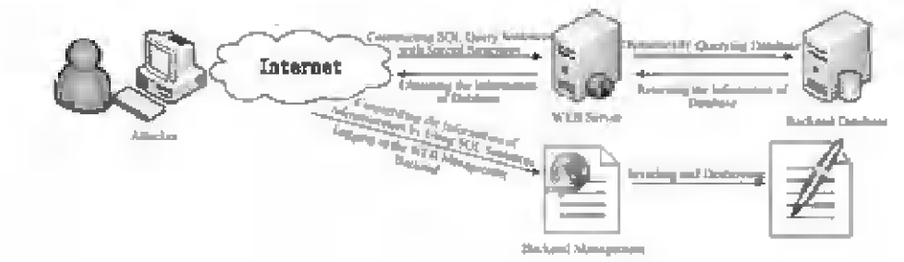

Fig. 5. Procedure of SQL Injection

Cross-Site Scripting also known by XSS attack is the topmost vulnerability found in the today's web applications which to be a plague for the modern web applications. XSS attacks permit an attacker to execute the malicious scripts on the victim's web browser resulting in various side-effects such as data compromise, stealing of cookies, passwords, credit card numbers etc. We have also discussed a high level of taxonomy of XSS attacks and detailed incidences of these attacks on web applications [7].

3.3 The importance of ethical hacking in mobile applications

Over the years mobile devices have become more indispensable and is presence has been increasing over these past years. Now a days we use and store a lot of sensible information in these devices so the necessity of having a secure application that stores these information or data has also increased.

The most important threats to enter phones start with malware or Trojans, these malicious programs hide inside good programs, stealing information and running automatically to other devices [14] [15].

Table 2. Security Threats in mobile applications [16]

Security threats	Explanation
Malware	Threats installed on the terminal for malicious behavior
Spam	Threats used to distribute advertisements and malware that can be sent to an unspecified number of people
Application vulnerability	Threats that perform malicious actions such as elevation of privileges by using the vulnerability of the developed application
Personal information extrusion	Threat of personal information leakage due to user carelessness when developing installed applications
Authentication bypass	Threats that randomly bypass or steal authentication for applications that require authentication
DoS	Threats that make the service provided by the application unusable

3.4 The use of ethical hacking in the development of embedded software

Is a developed software that grounds on devices like blood pressure monitors, gamming microcontrollers, aircrafts, cooking machines and many other devices that have custom hardware to perform specific functions.

Some incidents:

- Computer security researcher Chris Roberts was arrested on suspicion of having hacked into a United Boeing 737 during an April 2015 flight from Denver, Colorado, to Syracuse, New York [18] [19].
- In July 2015, two researchers demonstrated how to take over a Jeep Cherokee using the car's telematics system, shutting off the engine and disabling the brakes while a journalist drove the car [18] [20].
- In September 2015, Volkswagen admitted to installing software that defeated the emissions control system during testing on as many as 11 million diesel cars going back to 2009 [18] [21].
- In 2018, ethical hackers found Meltdown and Spectre hardware vulnerabilities that affect all Intel x86 and some AMD processors. Both vulnerabilities mess up isolation between user applications, giving applications access to sensitive data and expanding the attack surface. Both Linux and Windows developers have issued patches for their operating systems that partially protect devices from Meltdown and Spectre. However, lots of devices (especially old ones) running on vulnerable processors are still unprotected [22].

Vulnerability in embedded software can give hacker the opportunity of gaining sensible data, cause physical damage to the device hacked or even arm humans. Since they are expensive and valuable machines is important to ensure their security. Yet implementing security measures in embedded systems is connected with numerous challenges like power limitation, development expertise, network connectivity and poor access control, physical exposure [22].

Counter measures. Some counter measures could be considered to make sure all the vulnerabilities associated with the embedded system software is minimized.

The follow counter measures could help achieving a secure software:

- Conducting end to end threat analysis: The security of an embedded device can be improved by starting with identifying the potential threats. These threats must be evaluated in the context of the device manufacturer, operators (if the device is provisioned in such a way, and end users, including their usage). The attacks can be done in terms of wired Ethernet connection with the device used for communication, and common services such as web (HTTP). A complete product life cycle analysis needs to be performed [23].
- Select an Appropriate Run-Time Platform: - Restricting use of common platform govt should ensure that organizations should select an appropriate commercial runtime platform for an embedded system and make it mandatory for use. Implementing a system with components that have COTS security can increase the security and reduce the cost of development of the overall platform [23].

- Software design and implementations: It is important to write secure code so that the risks are of being attacked are minimized during the SDLC, is also a good idea to have a robust software architecture because he can make the software harder to hack and even minimize the hacking impact.
- Secure the Applications: - Same as the products should be tested first the application should also be tested first. Standards should be made and tested and then only permit the apps to get launch [23]
- Design and test for security: Security vulnerabilities are a class of software requirement deficiencies in design or implementation and earlier they are caught in the product development life cycle, the less costly it is to fix them and harden a system against attack. Security testing must involve defining the boundaries of a system and determining methods of exploiting weak defenses along these boundaries [23].

4 Conclusion

Is important to ensure the software is secure before deploying to the customers so it is crucial to conduct ethical hacking tests during the testing phase and before the deployment or production phase of SDLC. When it comes to developing a secure software is important to plan and schedule the software testing phase because if they manage to find vulnerabilities in this phase it will be less expensive correcting them now then correcting them during or after the production phase and consequently being vulnerable of a hacking attack like data breach and possibly generated even more expenses. It is a goal not only for developers but also for the customers to be confident that the software is less vulnerable to hacking and, in this paper, it is not only discussed the hacking phases the tools and techniques hackers used in these phases, but also particular security problems related to web application, mobile application and embedded system software security, the most common vulnerabilities associated and what are the risks of consequences if someone exploit them and what are measures avoid them.

5 References

1. "2021 Cyber Security Statistics: The Ultimate List Of Stats, Data & Trends." *PurpleSec*, https://purplesec.us/resources/cyber-security-statistics/. Accessed 22 November 2021.
2. Altvater, Alexandra. "What Is SDLC? Understand the Software Development Life Cycle." *Stackify*, 8 April 2020, https://stackify.com/what-is-sdlc/. Accessed 26 November 2021.
3. EC-Council. *Ethical Hacking and Countermeasures: Attack Phases*. vol. 1, EC-COUNCIL | PRESS. 5 vols.
4. Holst, Arne. *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025*. 07 June 2021. *Statista*, Statista, https://www.statista.com/statistics/871513/worldwide-data-created/. Accessed 03 October 2021.

5. Hu, Haibin. "Research on the technology of detecting the SQL injection attack and non-intrusive prevention in WEB system." *AIP Publishing*, 8 May 2017, https://doi.org/10.1063/1.4982570. Accessed 23 November 2021.
6. IBM. "Software development." *IBM Research*, https://researcher.watson.ibm.com/researcher/view_group.php?id=5227. Accessed 17 October 2021.
7. Gupta, Shashank & Gupta, B B. (2015). Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art. International Journal of Systems Assurance Engineering and Management. 8. 10.1007/s13198-015-0376-0.
8. simplilearn. "What Ethical Hacking Skills Do Professionals Need?" *Simplilearn*, 8 October 2021, https://www.simplilearn.com/roles-of-ethical-hacker-article. Accessed 29 November 2021.
9. "Kali Linux Features." Kali Linux, https://www.kali.org/features/.
10. "Parrot OS - 2022: recensioni, prezzi e demo." Software Advice, https://www.softwareadvice.pt/software/238015/parrot-os. Accessed 12 January 2022.
11. Devi, R. & Kumar, Mohankumar. (2020). Testing for Security Weakness of Web Applications using Ethical Hacking. 354-361. 10.1109/ICOEI48184.2020.9143018.
12. Antelo, Elisardo. "Defending against Web Application Vulnerabilities." free statistics, https://eden.dei.uc.pt/~mvieira/2012_Computer_DefendWeb.pdf. Accessed 13 January 2022.
13. Zed Attack Proxy (ZAP), 20 January 2020, https://owasp.org/www-chapter-dorset/assets/presentations/2020-01/20200120-OWASPDorset-ZAP-DanielW.pdf. Accessed 13 January 2022.
14. Hernández, Miguel, Luis Baquero, and Celio Gil. "Ethical Hacking on Mobile Devices: Considerations and practical uses." International Journal of Applied Engineering Research 13.23 (2018): 16637-16647.
15. Craig Heath, Symbian OS Platform Security: Software Development Using the Symbian OS Security Architecture. 2006
16. "A Study on the Mobile Application Security Threats and Vulnerability Analysis Cases." Korea Science, https://www.koreascience.or.kr/article/JAKO202034465346164.pdf. Accessed 13 January 2022.
17. Mutchler, Patrick & Doupé, Adam & Mitchell, John & Kruegel, Chris & Vigna, Giovanni. (2015). A Large-Scale Study of Mobile Web App Security.
18. M. Wolf, "Embedded Software in Crisis," in Computer, vol. 49, no. 1, pp. 88-90, Jan. 2016, doi: 10.1109/MC.2016.18.
19. http://securityaffairs.co/wordpress/36872/cyber-crime/researcher-hacked-flight.html
20. A. Greenberg, "Hackers Remotely Kill a Jeep on the Highway—with Me in It", Wired, July 2015, [online] Available: www.wired.com/2015/07/hackers-remotely-kill-jeep-highway.
21. M. Thompson and I. Kottasova, "Volkswagen Scandal Widens", CNNMoney, Sept. 2015, [online] Available: http://money.cnn.com/2015/09/22/news/vw-recall-diesel/index.html
22. https://www.apriorit.com/dev-blog/690-embedded-systems-attacks
23. EMBEDDED SYSTEMS SECURITY, https://tec.gov.in/pdf/Studypaper/Embedded%20sytem%20security.pdf. Accessed 14 January 2022.