The ISC Int'l Journal of Information Security

ISeCure

November 2021, Volume 13, Number 3 (pp. 31–38)

http://www.isecure-journal.org

Selected Paper at the ICCMIT'21 in Athens, Greece

Secure Coding Guidelines — Python **

Mohammad Ali A. Hammoudeh [1,*], Renad Ibrahim [1], Lama Alshraryan [1], Manar Alnomise [1], and Ragad Alhumidan [1]

[1] *Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia.*

ARTICLE INFO.

Keywords:
Cybersecurity, Secure Coding Practices, Coding Guidelines, Vulnerabilities, OWSAP, Python Challenges, SQL Injection

Type: Research Article

doi: 10.22042/ISECURE.2021.13.3.0

dor: 20.1001.1.20082045.2021.13.3.4.1

ABSTRACT

Recently, the interest in cybercrime and cybersecurity has increased dramatically both in terms of critical security issues and national economic information infrastructure and sensitive dealing policies, such as protection and data privacy. Moreover, the growing threat of cybersecurity has prompted the kingdom to pay more attention to its national cybersecurity strategy as the state embarks on a Vision 2030 plan, which aims to diversify the economy and create new jobs. Therefore, Any Computer system is always having security threats which are considered a big problem, and this includes application Codes as increasing demand. The paper aims to give detailed information about secure coding with Python and present security guidelines and considerations in different disciplines. It focuses on giving an overview of the authentication methods used in the application (Code) and showing program security mistakes to introduce vulnerabilities (Ex. SQL Injection). We review the new user authentication techniques, making it easier for the manager to choose the appropriate techniques for his organization by understanding the way it works, its advantages, and disadvantages. The administrator can integrate these mechanisms in a manner that is appropriate for his security plan. This will be useful for programmers and users to keep their codes and applications more secure and viable for usage in sensitive environments.

1 Introduction

In recent years, the utilization of applications (like mobile and web applications) has expanded in numerous organizations, such as public and private, government, healthcare, basic frameworks, etc. These applications have to be continuously developed within the most limited time conceivable to confront the competitors. In this manner increased cybersecurity threats for a wide range of users and as targets for cyber-attack than its security. Shockingly, most of these applications are not tested for security and Programmers do not follow instructions to write secure code. These applications can be prone to Vulnerability. Some of them are due to the programmer's omission; the other part is due to the vulnerabilities, which exist in the programming languages and their

* Corresponding author.
**The ICCMIT'21 program committee effort is highly acknowledged for reviewing this paper.
Email address: maah37@qu.edu.sa
ISSN: 2008-2045

libraries [1]. This study aims to find and analyze the best solutions and practices to keep our information protected, make the economy save, and create jobs.

One of the best ways of coding practice is secure coding practice, which can prevent applications from exploits and vulnerabilities. This assists the developers to include Software Security Principles into their day-to-day development lifecycle and operations. The Challenge is with increasing demand for application code. In addition, as software developers when authoring the code that Creates an application. They might make mistakes. These mistakes can lead to unintentional vulnerabilities that potentially compromise that software or the data it processes, they have to embrace and practice various secure coding techniques [2]. Most developers did not learn about secure coding or crypto in school. Developers make coding security vulnerabilities or utilize third-party modules or components that are powerless [3]. These cases frequently make them ignore a basic component inside the Secure Software lifecycle Development (Programmer's Omission). The major objective of the article is to reduce vulnerability over an application code using python programming languages (PYPL) as in Figure 1.

Therefore, we will report the common threats and vulnerabilities that can threaten any python code and suggest protection methods. Therefore, our target of this research will be to satisfy the following objectives:

- Presenting recommendations and guidelines to the programmers to follow during creating their projects.
- Decreasing vulnerabilities in the python codes.
- Reducing the bad effects of attacks on the economic and data privacy.

Cybersecurity is a global issue that poses a complex threat to everywhere technology is being used. Governments, businesses, and individuals are all affected by cyber-attacks. Furthermore, new applications and web services serve our daily needs more and more and they arrive at a great pace. The government and private sectors of Saudi Arabia have jointly built and implemented approximately 19 public health applications and platforms that provide health services. Saudi Arabia has been subject to approximately 160,000 cyberattacks daily according to King Abdul-Aziz City National Cyber Security for Science and Technology [5]. On the other hand, 82 percent of vulnerabilities were located in application code [6]. Hence the importance of focusing on writing a secure code to reduce the vulnerability that the attacker might exploit.

Internet-connected devices, including hardware, software, and data, provide cybersecurity protection against cyber-attacks. Cyber security and physical security are also used by companies to secure data centers and other computerized systems from unauthorized access. Data protection is a subset of cyber security that is designed to ensure the availability, integrity, and confidentiality of data [7]. Secure coding is also an important part of Cyber Security, so software developers must be aware of secure data by using secure coding practices, which must be based on a set of guidelines to prevent security risks by writing secure code in the programming language. It has been shown that there is no single programming language that is substantially more vulnerable than the others. Vulnerabilities can be found in all programming languages [8]. However, Python seems to have fewer vulnerabilities than some of the other languages, also it has extensive use in the Information Security Industry.

Cybersecurity: is described by the National Institute of Standards and Technology as "the method of protecting information by avoiding, detecting, and responding to attacks".

Software Security: refers to the methods, frameworks, techniques, and procedures used to develop the security of software and the environment in which it runs. Software security aims to improve software's integrity by testing and fortifying it at multiple stages and settings across the software development lifecycle (SDLC) and after it has been released.

Application Security: involve security measures taken at the application level to deter the theft or hijacking of data or code within the app. It includes security considerations made during application development and design, as well as systems and methods for protecting apps after they have been deployed. Web application security: is the process of defending websites and online services from various security threats that take advantage of vulnerabilities in the code of the application. Secure Coding: this is a collection of secure coding guidelines that must be followed while coding. Secure coding is entrusted to software developers' awareness of secure coding practices. Aware and train software developers on how to develop, escape security vulnerabilities and write safe code in the programming language that will be used to develop a product.

Python: Guido van Rossum designed Python in the early 1990s Python is an object-oriented, interpreted, high-level programming language with dynamic semantics. Integrated high-level data structures, along with dynamic writing and dynamic connectivity, make them very attractive for developing fast applications, as well as for using them as a text language or paste language to connect existing components. Python

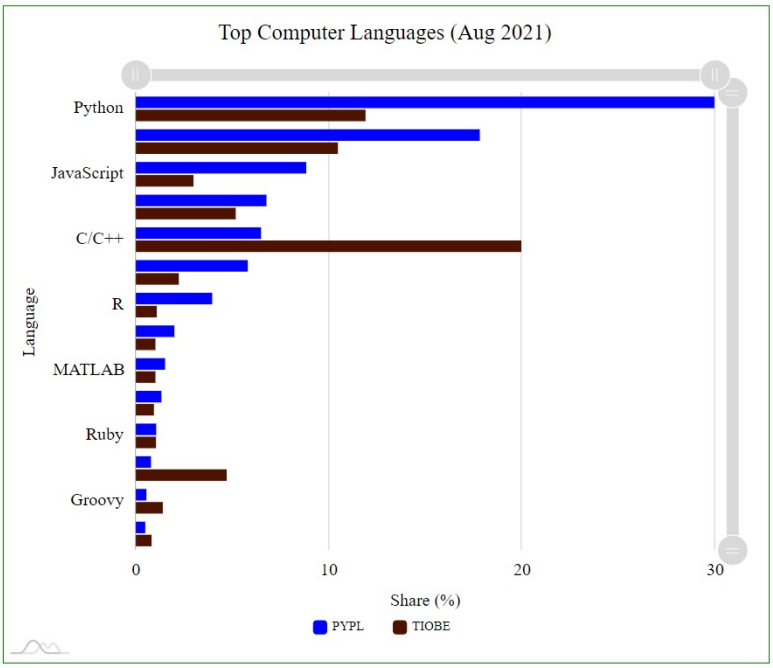

Figure 1. The top computer languages (Aug 2021) [4]

is simple and easy to learn, emphasizes readability, and thus reduces the cost of program maintenance. Python supports units and packages, encouraging code reuse and program modularity. Python translator and comprehensive standard library are available in source or binary form free of charge for all major platforms and can be distributed free of charge [9].

The paper is organized as follows: in the Section 1, explains an introduction to Secure Coding and its applications, Section 2, displays the recent technologies and techniques used in security, Section 3 rates some statistics and studies, analyses some python framework recommends the best solution. The last section presents the conclusion and future work.

2 The Recent Technologies and Techniques

This segment showcases works that are relevant to the paper. There are several organizations and establishments answerable for developing standards and best practices for writing secure coding.

In [10], to follow acceptable coding standards, web application developers must be aware of various web application assaults. This paper outlined various research methods for identifying bugs during the coding process. The primary goal of these methodologies is used to identify flaws in source code before they are exploited in a real-world situation.

The results of [11] indicate that students' adherence to secure coding practices can be positively impacted through a formal educational intervention. However, it is important that such an intervention address both the knowledge and behavior of students since having the requisite knowledge does not ensure compliance. It is for this reason that a behavioral compliance-monitoring instrument formed part of the study. This is a step toward educating students in secure application development, which is essential in addressing the many security vulnerabilities existing in Web applications today. Limitations of this study do exist.

Firstly, this study addressed only the identified secure coding practices, which were determined from OWSAP. Secondly, the identified secure coding practices only focused on the data access layer of Web applications developed in the .NET environment. Future research could investigate similar interventions within various other application development contexts. We must mention Bandits are a tool designed to find common security issues in Python code. The thieves' machine processes each file, builds an AST from it and runs the right plug-ins for an AST contract. Once Bandit has finished checking all the files, he creates a report.

This section reviews various researches related to all aspects of secure coding and python besides some related works, define SQL Injection, and explain the main issues about it. In this part, we spotlight some studies that have given some solutions over five years, starting with Petar who suggested in his study to design, create, and implement a database management system that collects information about university doc-

toral students [12]. It is possible to use this system in other educational institutions. Also, optimizing data-related operations and providing their reliable storage. They use Python, a high-level language, and the MySQLdB library on a MySQL database.

In aspects of security, they apply HTTP certification and electronic signatures provide greater reliability to users. And they focus on such security vulnerabilities of web applications: Cross-site scripting (XSS) protection Cross-site request forgery (CSRF) protection SQL injection protection.

In [13] they use a predefined Python tool to detect and exploit vulnerabilities. The paper is to help find XSS vulnerabilities. The study is work by using this tool to search entire websites for affected endpoints. The system implementation requires several processes that require users to enter input as a URL to analyze and locate the vulnerable endpoints.

Finally, It will show whether it is vulnerable to XSS attack or not. Arafa Anis *et al.* start to propose policies that map to some type of attacks. The proposed policies are as follows: for SQL injection, they suggested input sanitization, output validation, and the principle of least privilege. And for XSS attacks they suggested input sanitization, output validation, the principle of least privilege, and Content Security. And for resource Alteration attacks they suggested the Principle of Least Privilege and Subresource Integrity. They suggested using the integration verification model IVM to ensure the client-side code it's not modified. They do experiments that simulate attacks like cross-site scripting, SQL injection, and code tampering.

The purpose of the experiment is to demonstrate the attack prevention rate of their approach using vulnerability scanners and attack tools. They experiment with 22 web applications that have been alerted by their approach. As a consequence, to experiment, the preventive rate is rising as follows: SQL injection has a 24 percent, cross-site scripting has a 31 percent, and resource alteration has a 43 percent prevention rate. From only incorporating the security policies, the average percentage increase is about 33% [14]. In addition to other related papers have compared in Table 1 [15], [16], [17], [18], [19], [20].

Secure coding guidelines are not universally accepted in all programming languages. As a result, some organizations and institutions Improve secure standards and practices. The Open Web Application Project (OWSAP), as well as CERT, are examples of these. Division is an organization that pioneers in cybersecurity. To harden the resilience and security of computer systems and networks, they collaborate with government, industry, law enforcement, and academia. They research issues with broad implications for cybersecurity and face large-scale, sophisticated cyber threats by developing innovative methods and tools [21].

OWASP Top Ten is considered an awareness standard sheet for experts (Developers) and secure web applications. It is universally recognized by developers as the first step toward safer encryption. Companies should adopt this document and start the process of making sure that their web applications reduce these risks. Using OWASP is perhaps the quantum leap in the culture of software development within your organization to one that produces a safer code.

OWASP provides cheat sheets for each of the different web risks. Details vary in each cheat sheet but some elements may be included in each of them an introduction describing risks, risk defenses, prevention measures that do not work, General rules, and recommendations. The (OWSAP) is another organization that launched the OWSAP Proactive Controls 2018 is a Top Ten list of security techniques that any software development project should implement it. They are listed from most important to least important, with control number one being the most important. This list was created by developers to help those developers who are new to secure development [3], as summarized in Figure 2.

3 Discussion

3.1 Statistics and Studies

In this part, we look for some statistics and studies to select the riskiest vulnerability, besides understanding its effect. These data are shown in the following:

The average web application was attacked 20,000 times between January and February 2020. The majority of the attackers went after typical vulnerabilities including path traversal, SQL injection, and XSS. Almost all of the attacks 99% failed to exploit a specific vulnerability [22]. In 2020, the NVD database will have 18,362 vulnerabilities. This number is greater than in previous years (17,382 in 2019 and 17,252 in 2018). In Figure 3 shows vulnerabilities from 2019 divided into the OWSAP top 10 2017 categories [23].

3.2 Security Framework

- Python Web Frameworks: Given the difficulty of brevis problems and the size of code, based on the average software project, having a tool that can assist in the discovery of security vulnerabilities would be advantageous. We decided to look into tools that could help us develop secure Python web applications because we are

Figure 2. The top computer languages (Aug 2021) [3]

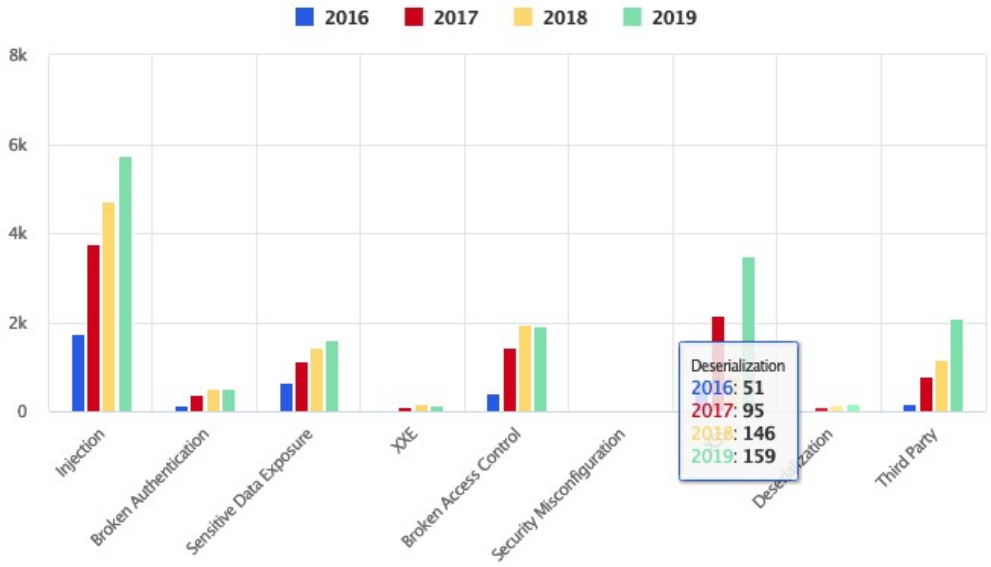

Figure 3. Vulnerabilities into OWASP categories [23]

Table 1. A previous study and solutions in secure coding [15–20]

Paper	Brief Description	Threat	How to solve?	Result	PL
[15]	The proposed tool detects and reveals vulnerabilities with predefined python tool.	Cross site scripting XSS	The users will enter input as URL to analyze and find the affected endpoints to exploit the vulnerability.	A generated payload will reveal the vulnerabilities in affected endpoints then generate a report.	Python
[16]	The proposed model is a novel online detection method against SQL injection attack.	SQL injection	The model was validated using tools and web server protection datasets.	The proposed method is quite simple to implement yet highly effective overall kinds of SQL injection	ASP.NET with MS-SQL
[17]	A secure coding solution is proposed that web developers to test the possibility of such attacks.	SQL injection	Steps will be taken during the coding process.	Satisfactory	JAVA
[18]	Teaching principles and practices of Java secure coding.	Buffer over flows	Automatically tool for detecting API misuse in Java	Learn about the requirements for improved software support for secure coding practice.	Java
[19]	Revealing the lack of secure coding assistance and documentation, as well as a significant gap between security theory and coding practices.	-Cross-Site Request Forgery (CSRF)	Improving secure coding practices.	Five parts tutorials for developers.	Python and Java
[20]	Blindspots in APIs are a significant the problem across languages and knowledge or experience alone is insufficient to solve the problem.	SQL injections and buffer over flows.	Tools are required to assist developers in identifying API blindspots, as they write code that uses those APIs.	Spot API blindspots when writing code that uses certain APIs, alerting t hem and reducing the chance of vulnerabilities being introduced.	Python and Java

both interested in Python development. Django and Flask are two Python web frameworks that we encountered. The Django web platform is a full-stack framework and was selected first it's one of the most common web frameworks, The Flask micro web platform is a common non-full-stack framework that came in second [12].

- Django: is a software platform that comes with everything you will need to create a complete-featured web application. The Model View Controller (MVC) architecture pattern is used by Django. It is designed in a way that the developer is forced to incorporate features in a particular way. This means that the project's architecture does not allow for much customization. Django is based on the idea of software, which is an abstract concept. The following modules make up an app:
  - Main module - where the app is starting to execute code.
  - Tests module - testing of the app.
  - Views module - visualization of the app.
  - URLs module - maps URLs to views.
  - Models module - models for instance from a database.
  - Apps module - nested apps.

  A Django web application is made up of many applications. Django's Influence is the ease at which an app can be reused since they can be connected together using URLs [12–23].
- Flask: is a platform for building micro websites. Flask is highly customizable, as you can choose from a variety of options. Your web application's design is special to you. It is also possi-

ble, for example, to make your form validation or use one of the several form validation packages available [12]. The following features are included with the flask:

- Development server.
- Unit test support.
- REST support.

Flask's strength is that it allows the developer to customize everything.

3.3 Recommendation and Solution

As mentioned before, we looked at software security, specifically software application security, and how the integrity of software applications is critical, as well as the requirements for ensuring this integrity. We explored the Web Applications vulnerabilities and some of the practices that would mitigate web application vulnerabilities. To find and exploit secure coding over SQL injection vulnerabilities and improve the security level by doing the best practice and programming solution to be extracted as a report. The report will give a guideline to the developer to enhance the code by getting rid of SQL injection as shown in Figure 4. Figure 4 shows all the components of the system as a high-level view. the client sends requests to the database via the web interface. All requests and responses must go through Proxy by the web application. The client's web app requests SQL statement information that uses input parameters for the web model to create the correct SQL statement. This SQL statement is then sent to the proxy. When it receives the SQL statement, the proxy filters it first. Only clean SQL phrases are then sent to the database. The database processes the request and sends its response through the proxy. The proxy, in turn, sends the response to the web application for processing to produce the correct client view by the following phases:

(1) Coding Phase: the implementation part, which has two stages that create an infected code, applying a detection to catch the vulnerability.
(2) Testing Phase: to ensure that previous stages are executed as required and the code (case study) is protected and clean against SQLite. This is also where we'll probably figure out what's working and what's not in our plan.
(3) Delivery and Maintenance Phase: the last phase, where the report is generated, the instruction and guidelines are sent to the developers that help to create a secure code against SQL Injection vulnerability.

4 Conclusion and Future Work

This paper presented an overview of recent authentication techniques during the design system for use in the Cybersecurity field that conducts the best solutions and practices to keep our information protected, making the economy save and create jobs, we aim to review the new user authentication techniques, making it easier for the manager to choose the appropriate techniques for his organization by understanding the way it works, advantages, and disadvantages, the used techniques were analyzed and evaluated, in addition to mentioning any of the attacks are impervious to it.

References

[1] Amir A Khwaja, Muniba Murtaza, and Hafiz F Ahmed. A security feature framework for programming languages to minimize application layer vulnerabilities. *Security and Privacy*, 3(1):e95, 2020.

[2] Eva Hariyanti, Arif Djunaidy, and Daniel Siahaan. Information security vulnerability prediction based on business process model using machine learning approach. *Computers & Security*, 110:102422, 2021.

[3] Gleidson Sobreira Leite and Adriano Bessa Albuquerque. An approach for reduce vulnerabilities in web information systems. In *Proceedings of the Computational Methods in Systems and Software*, pages 86–99. Springer, 2018.

[4] Top Computer Languages. Statisticstimes.com, 2021. Accessed 18 February 2022.

[5] F Al-sharif. Cybersecurity awareness: A challenge for saudi arabia. *Arab News*, 2018.

[6] Positive Technologies. Web applications vulnerabilities and threats: statistics for 2019. ptsecurity, 2020. Accessed 18 February 2022.

[7] PS Seemma, S Nandhini, and M Sowmiya. Overview of cyber security. *International Journal of Advanced Research in Computer and Communication Engineering*, 7(11):125–128, 2018.

[8] Tiago Espinha Gasiba and Ulrike Lechner. Raising secure coding awareness for software developers in the industry. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 141–143. IEEE, 2019.

[9] Executive Summary. What is python?, 2021. Accessed 18 February 2022.

[10] Ossama B Al-Khurafi and Mohammad A Al-Ahmad. Survey of web application vulnerability attacks. In *2015 4th International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, pages 154–158. IEEE, 2015.

[11] Vuyolwethu Mdunyelwa, Lynn Futcher, and Johan van Niekerk. An educational intervention for teaching secure coding practices. In *IFIP World Conference on Information Security Education*, pages 3–15. Springer, 2019.

[12] Stefan Micheelsen and Bruno Thalmann. A static

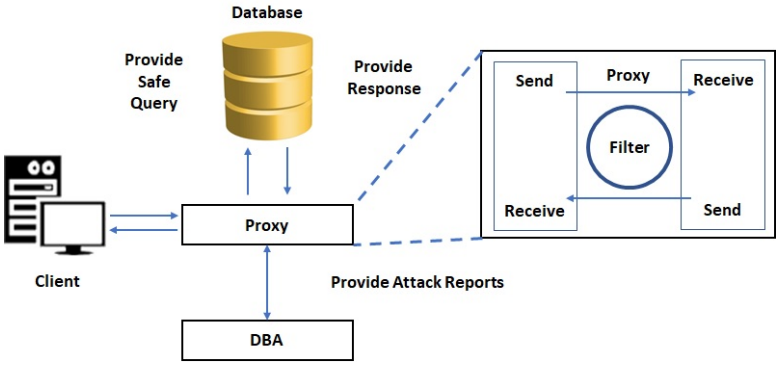

Figure 4. The solution tool structure

analysis tool for detecting security vulnerabilities in python web applications, 2016.

[13] R Abirami, DC Joy Winnie Wise, R Jeeva, and S Sanjay. Detecting security vulnerabilities in website using python. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 844–846. IEEE, 2020.

[14] Arafa Anis, Mohammad Zulkernine, Shahrear Iqbal, Clifford Liem, and Catherine Chambers. Securing web applications with secure coding practices and integrity verification. In *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing*, pages 618–625. IEEE, 2018.

[15] R Abirami, DC Joy Winnie Wise, R Jeeva, and S Sanjay. Detecting security vulnerabilities in website using python. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 844–846. IEEE, 2020.

[16] Asish Kumar Dalai and Sanjay Kumar Jena. Neutralizing sql injection attack using server side code modification in web applications. *Security and Communication Networks*, 2017, 2017.

[17] Bhawana Gautam, Jyotiraditya Tripathi, and Satwinder Singh. A secure coding approach for prevention of sql injection attacks. *International Journal of Applied Engineering Research*, 13(11):9874–9880, 2018.

[18] Na Meng, Stefan Nagy, Danfeng Yao, Wenjie Zhuang, and Gustavo Arango Argoty. Secure coding practices in java: Challenges and vulnerabilities. In *Proceedings of the 40th International Conference on Software Engineering*, pages 372–383, 2018.

[19] Sazzadur Rahaman, Na Meng, and Danfeng Yao. Tutorial: Principles and practices of secure crypto coding in java. In *2018 IEEE Cybersecurity Development (SecDev)*, pages 122–123. IEEE, 2018.

[20] Yuriy Brun, Tian Lin, Jessie Elise Somerville, Elisha Myers, and Natalie C Ebner. Blindspots in python and java apis result in vulnerable code. *arXiv preprint arXiv:2103.06091*, 2021.

[21] contrast security. contrast labs application security intelligence bimonthly report, 2021. Accessed 18 February 2022.

[22] D. S. Bekerman, Yerushalmi. The state of vulnerabilities in 2019. medium, 2020. Accessed 18 February 2022.

[23] Django Djangoproject. [23] the web framework for perfectionists with deadlines, 2021. Accessed 18 February 2022.

Mohammad Ali A. Hammoudeh is Assistant Professor in the Department of Information Technology (Collage of Computer) at Qassim University – Saudi Arabia. He received his PhD in Engineering Science from the National Technical University – Kyiv (Ukraine), in 2007. His research interests are Computing and Networks, E-Leaning Systems, IoT, 5G, Security and Cloud Technology

Renad Mohammad Ibrahim is a lecturer in the Department of Information Technology (Collage of Computer) at Qassim University – Saudi Arabia. She has received her Master degree in Computer Science from Al-Balqa' Applied University– Amman (Jourdan), in 2009. Her research interests are Network, Education, Web, Security and Software Engineering.

Lama Alshraryan, Manar Alnomise, and Ragad Alhumidan are graduate students from the Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia.