Adaptive Model for Protection of Electronic Resources against Information Security Threats

Komil Kerimov
Department of System and Applied Programming
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi
Tashkent, Uzbekistan
kamil@kerimov.uz

Zarina Azizova
Department of Information Security
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi
Tashkent, Uzbekistan
z.i.azizova18@gmail.com

***Abstract*. The rapid development of digitalization and the creation of electronic resources, in areas such as e-commerce, government portals and others leads to the actualization of data protection issues. The protection of electronic resources is becoming more and more relevant every day. This article presents the concept of adaptive protection of electronic resources from information security threats. In the course of this research, an adaptive model of protection of electronic resources from threats to information security based on behavioral analysis was developed.**

Keywords: adaptability, threat, behavioural analysis, information security, electronic resource, cross-site scripting (XSS), SQL-injection

I. INTRODUCTION.

Many companies do not pay attention to the fact that their employees periodically make changes to the electronic resource itself. Consequently, there can be new types of in-formation security (IS) threats, which related to the network, or to the operating system. In addition, new software appear with great speed, and different information technologies change. This can lead to a reduction in the level of protection of electronic re-sources over time.

Administrators usually take certain actions only on the security threats they know, but the security threats may be much more. It is necessary to provide a clear control analysis of the protection of electronic resources, and use a comprehensive protection of electronic resources from IS threats. An adaptive mechanism will allow to identify and take decisions on IS threats, with well-established and managed means. Adaptive security of an electronic resource includes the following components:

- IS threat classification algorithms;
- Adaptive models of electronic resources protection against IS threats;
- Adaptive methods of electronic resources protection against popular IS threats.

Adaptive protection monitors popular IS threats and provides timely protection and alerts the administrator, and it allows to apply specific protection based on the type of IS threat. In other words, adapt to the IS threat and apply the right protection.

The Web Application Firewall (WAF) works at the application layer of the TCP/IP protocol stack. This allows it to use it to protect against attacks at the application layer, unlike a conventional firewall. The classic WAF model based on the principle of mapping existing patterns to attack signatures. This approach is possible to implement it in two ways: either through blacklists or whitelists. Much research in this area has focused on improving the detection accuracy of malicious packets packet detection using ma-chine-learning techniques.

The works of researchers such as [2] have developed separate rules for checking HTTP data streams and finding HTTP transactions. The authors developed a hybrid SQL Injection Prevention System (HIPS) that uses a machine learning classifier together with pattern-based security rule checking. This optimized detection efficiency through a prediction module that separates legitimate requests from attacks. In turn, the authors of paper [3] also note the high efficiency of the Naive Bayes method used in conjunction with pattern matching. The accuracy of functioning and attack detection of the hybrid system was almost 98%. In practical perspective, this score might become better by using Convolutional Neural Networks (CNNs) due to regularization of over fitting suppression. As noted in [4] malware detection, using convolutional neural networks achieves accuracy of about 94%.

Regarding cross-site scripting (XSS) attacks, work [5] classified cross-site scripting vulnerabilities into three

Print ISSN 1691-5402
Online ISSN 2256-070X
https://doi.org/10.17770/etr2024vol4.8217

levels: local, reflected and persistent. The attack action occurs when a user initiates access to a web page, and since the attacker embeds the malware into a persistent web page, the unpatched malware is potentially dangerous. A group of researchers in their paper [6] considered the role of blockchain to enhance security by preventing XSS attacks. Their proposed system uses WAF with deep learning approach and pattern blocking after detecting and preventing SQL injection and successful user login.

As noted by the authors of the article [7], web applications are directly dependent on a database that provides legitimate data. Their proposed method uses input data categorization and input data verifier. These two steps increase the effectiveness of automatic detection and prevention of web attacks. The research paper reports on the researchers' assessment of the effectiveness of the "Modsecurity" web application firewall in preventing SQL injection attacks. According to them, regular updates of "Modsecurity" rules are essential to achieve effective protection to block new threats. The main words in the title start with capital letter, articles and conjunctions with lowercase letters.

The adaptive model developed to protect electronic resources against information security threats based on behavioral analysis of the system and the user. The use of the adaptive model provides effective protection of electronic resources. The model adapts depending on the presence of an IS threat, either signature-based or behavioral-based protection is applied.

II. PROPOSED METHODOLOGY.

The most fundamental solution for removing a web application vulnerability needs to have been resolved by making patches to the system or to the software itself. This can also be done using the white box testing method, which involves code analysis, system or web application vulnerability scanning tools, a penetration test, that requires higher level methods in order to find out about the problems of your own application. It is also possible to install a firewall for the web application in the front-end of the application node to ensure that the system protected.

In addition to the security features in the system, we also deal with key characters for suppressive attack schemes such as SQL injection and cross-site scripts. Coding, conversion, deletion and other handling on the key symbols are required to avoid this attack behaviour on the server or browser side. We also maintain a blacklist of keywords that need to prevent it in network traffic to improve system security. As a test environment, we create an e-commerce web application and install vulnerability-scanning software of the application; we also test the servers by applying the most appropriate protection setting. The result compared with a server-scanning test without such setting to confirm the efficiency of the protection, which effectively prevents SQL injection and cross-site scripting attacks.

A. Adaptive concept of electronic resources protection against information security threats

For a more detailed consideration of the concept of protecting electronic resources from information security threats, consider the four levels that make up an electronic resource:

- The web application layer, i.e. this layer processes communications with users. For example, the electronic resource of an appliance portal, various organization web-sites;
- Layer of working with databases, i.e. this layer processes system data and carries out storage of information. For example MySQL, PostgreSQL;
- Layer of working with the OS, i.e. this layer is responsible for all system components such as web server, interpreter, system kernel itself;
- Layer working with the network, this layer is responsible for the network interaction between the users and the electronic resource.

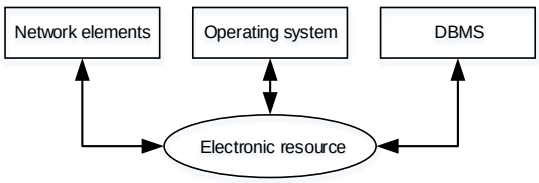

Fig. 1. Diagram of interaction of an electronic resource with all elements of the infrastructure.

Now, many protection systems base their protection on outdated data protection mechanisms. This does not take into account modern types of IS threats. The Figure 2 illustrates mechanism how the attacker realize an attack.

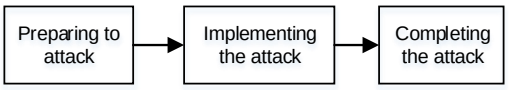

Fig. 2. Attacker's sequence of actions.

Preparing an attack means that an attacker searches for IS threats in an electronic resource. The search is carried out with security scanners in automatic mode, or popular types of IS threats are checked manually. Implementing an attack means that the attacker conducts an actual attack using the vulnerability found in an electronic resource. Completing an attack means that the attacker attempts to cover his actions and his tracks after the attack is complete.

Our research of existing electronic security mechanisms reveals that these mechanisms work only during the second stage, i.e. performing of the attack. It is better to prevent an attack as early as the first stage, i.e. the preparation stage. For example, when an electronic resource scanned, the IP address from which the requests come is blocked.

B. Developing the adaptive model of protecting electronic resources from information security threats

In order to develop an adaptive model for protecting electronic resources from information security threats based on behavioral analysis, consider how user requests for electronic resources occur (Fig.3).

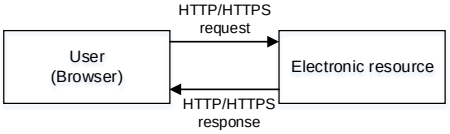

Fig. 3. Scheme of HTTP/HTTPS requests.

The user, through his browser, sends HTTP/HTTPS requests to an electronic resource and receives a response in the same form of an HTTP response.

The developed adaptive model based on the analysis of HTTP/HTTPS requests and their comparison with the benchmark. If the request differs from the benchmark, it is evidence of an IS threat. The adaptive model also implemented on the side of an electronic resource. This allows to analyses both simple HTTP requests and encrypted HTTPS re-quests. By examining HTTP/HTTPS requests we found that the following requests indicate the presence of an IS threat:

- Requests that contain malicious characters in the URL parameter;
- Requests that request pages of an electronic resource which do not exist;
- Requests in which the User-Agent parameter is missing or distorted;
- Requests in which the Referer parameter is distorted or contains malicious code;
- Requests in which the Cookie parameter is distorted or contains malicious code;
- Requests in which the length of the parameters exceeds the specified limits.

The adaptive model scheme given below. If the request differs from the benchmark, it is evidence of an IS threat. The adaptive model implemented on the side of an electronic resource. This allows analysing both simple HTTP requests and encrypted HTTPS requests.

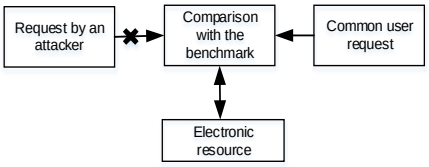

Fig. 4. Functional scheme of the adaptive model.

If the request differs from the benchmark, it is evidence of an IS threat. The adaptive model also implemented on the side of an electronic resource. This allows to analyses both simple HTTP requests and encrypted HTTPS requests.

1) Requests that contains malicious characters in the URL parameter.

The URL parameter sent to the web server via user-side data, which modified by an attacker. Here are the developed criteria of the benchmark URL:

- K_1 – URL length, which is customizable to a specific electronic resource;
- K_2 – Absence of certain special symbols, indicating the presence of an IS threat;
- K_3 – Absence of certain specific words indicating an IS threat.

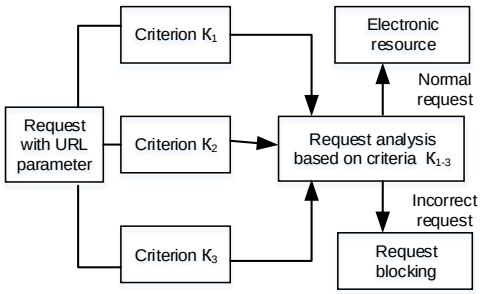

Fig. 5. Detection and protection of electronic resource based on analysis and identification of URL parameter.

These parameters are adjustable and can be adapted to each electronic resource. Let us look at an example of how it works. Reference query with URL parameter:

```
GET
http://site.uz/1.php?name=kamil&surname=keri
mov HTTP/1.1
Host: site.uz
Request modified by an attacker:
GET
http://site.uz/1.php?name=kamil<script>alert
</script>&surname=kerimov'union pass=''
HTTP/1.1
Host: site.uz
```

The analyzer checks query lengths and special characters accordingly. As can be seen from the modified query, the length of the query increased compared to the benchmark, and there are special characters and keywords indicating the threats of XSS and SQL injection. As a result, such kind of requests blocked.

2) Requests of non-existent pages of electronic resource.

The Location parameter transmitted to the web server via user-side data, or this parameter modified by an intruder or by electronic resource scanning software. Here are the criteria developed of the benchmark parameter Location:

- K_4 – Requests that query existing pages i.e. produce a response from server 200;
- K_5 – No requests giving a response 404 from the server;
- K_6 – No requests giving a response 403 from the server.

These parameters are configurable and can be adapted to each electronic resource. Consider a scheme for detecting and protecting an electronic resource, based on the analysis and identification of the Location parameter.

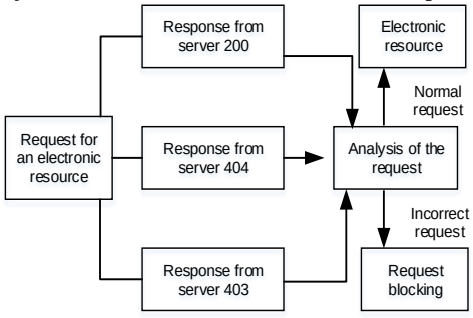

Fig. 6. Detection and protection of electronic resource based on analysis and identification of Location parameter.

Benchmark request with Location parameter:

```
GET http://site.uz/1.php HTTP/1.1
Host: site.uz
Response from the server:
HTTP/1.1 200 OK
Request modified by an attacker
GET http://site.uz/config.php HTTP/1.1
Host: site.uz
Response from the server
HTTP/1.1 404
GET http://site.uz/admin HTTP/1.1
Host: site.uz
Response from the server
HTTP/1.1 403
```

The analyzer checks the response from the web server accordingly. As can be seen from the modified request, the response from the server differs from the benchmark. As a result, the request is blocked.

3) Requests with missed or distorted User-Agent parameter.

User-Agent parameter sent to the web server via data from the browser side of the user, or an attacker to hide their real data can change this parameter. The criteria developed for the User-Agent benchmark parameter:

- K_7 – Requests in which the User-Agent is present, as well as its constituent parts, such as: Engine version, OS version, browser version;
- K_8 – Absence of queries without User-Agent parameter;
- K_9 – Absence of queries containing User-Agent in garbled form or without any of its component parts.

These parameters are configurable and can be adapted for each electronic resource. Figure 7 illustrates scheme for detecting and protecting an electronic resource, based on the analysis and identification of the User-Agent parameter.

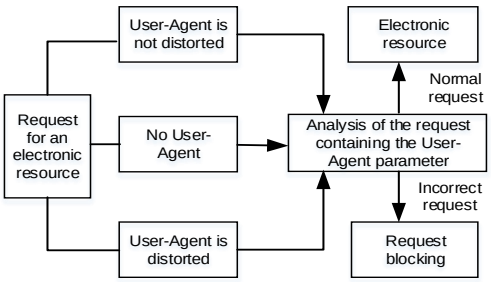

Fig. 7. Detection and protection of electronic resource based on analysis and identification of User-Agent parameter.

Benchmark request with User-Agent parameter given below:

```
GET http://site.uz/1.php HTTP/1.1
Host: site.uz
User-Agent: Mozilla/5.0 (Linux; Android 5.0;
SM-G900P Build/LRX21T) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/80.0.3987.149
Mobile Safari/537.36
Request modified by an attacker
GET http://site.uz/config.php HTTP/1.1
Host: site.uz
User-Agent:
GET http://site.uz/admin HTTP/1.1
Host: site.uz
User-Agent: Mozilla (Chrome/80.0.3987.149)
```

The analyzer checks the User-Agent parameter accordingly. As you can see from the modified request, the User-Agent may not exist or been changed and therefore will be different from the reference and will be blocked.

4) Requests with contained malicious code or distorted Referer parameter.

Referer parameter sent to the web server via data from the browser side of the user, an attacker to insert malicious code could modify this parameter. Criteria developed for the Referer benchmark parameter:

- K_{10} – Length of Referer, which is configured for a specific electronic resource;
- K_{11} – Absence of certain special characters in Referer, indicating an IS threat;
- K_{12} – Absence of certain special words in Referer, indicating an IS threat.

These parameters configurable and can be adapted for each electronic resource. Electronic resource detection and protection scheme based on the analysis and identification of Referer parameter illustrated in Figure 8.

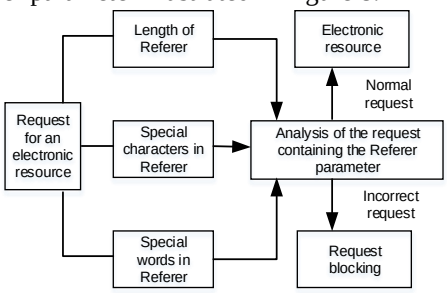

Fig. 8. Detection and protection of electronic resource based on analysis and identification of Referer parameter.

Reference

```
GET http://site.
Host: site.uz
User-Agent: Mozi
SM-G900P Build/L
(KHTML, like Gec
Mobile Safari/53
Referer: http://
Request modifiec
GET http://site.
Host: site.uz Us
(Linux; Android
AppleWeb-Kit/537
Chrome/80.0.3987
Referer:http://s
ipt>alert</scrip
pass=''
```

```
GET http://site.uz/1.php HTTP/1.1
Host: site.uz
User-Agent: Mozilla/5.0 (Linux; Android 5.0;
SM-G900P Build/LRX21T) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/80.0.3987.149
Mobile Safari/537.36
Cookie:name=kamil; surname=kerimov
Request modified by attacker
GET http://site.uz/1.php HTTP/1.1
Host: site.uz
User-Agent: Mozilla/5.0 (Linux; Android 5.0;
SM-G900P Build/LRX21T) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/80.0.3987.149
Mobile Safari/537.36
Cookie:name=kamil;surname=kerimov<script>doc
ument.write('test')</script>
```

query with Referer parameter:

The analyzer checks the length of Referer parameter, special characters and words. As can be seen from the modified request, the length of the request increased and compared to the benchmark, and there are special characters and keywords that indicate a security threat such as XSS and SQL injection. Therefore, such request blocked.

5) Requests with contained malicious code or distorted Cookie parameter.

Cookie parameter sent to the web server via data from the browser side of the user. An attacker can modify this parameter to insert malicious code. Criteria developed for the Cookie benchmark parameter:

- K_{13} – Length of Cookie, which is configured for a specific electronic resource;

- K_{14} – Absence of specific characters in Cookie parameter, indicating an IS risk;
- K_{15} – Absence of certain special words in Cookie parameter, indicating an IS risk.

These parameters are adjustable and can be adapted to each electronic resource. Consider the scheme for detecting and securing an electronic resource, based on the analysis and identification of the Cookie parameter, which shown in Figure 9.

The analyzer checks Cookie length, special characters and words accordingly. As it can be seen from the modified request, the length of the request is longer than the benchmark, and there special characters indicating XSS threats. Therefore, such kind of requests blocks.

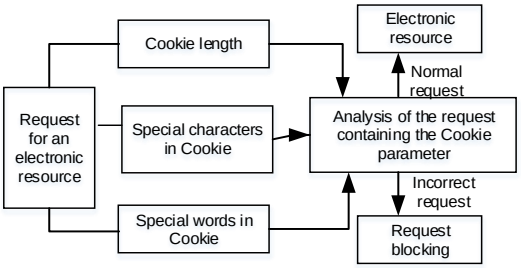

Fig. 9. Detection and protection of electronic resource based on analysis and identification of Cookie parameter.

Benchmark request with Cookie parameter:

6) Requests with exceeded specified Length Limits of parameters.

Parameters sent to the web server via data from the browser side of the user, an at-tacker could modify any HTTP/HTTPS request parameters. Criteria developed for the size of the HTTP/HTTPS request benchmark parameter shown below:

- K_{16} – The length of each HTTP/HTTPS request parameter is specified based on a specific electronic resource;
- K_{17} – The length of each HTTP/HTTPS request parameter value based on a specific electronic resource.

These parameters are configurable and can be adapted for each electronic resource. Scheme for detecting and protecting an electronic resource based on analysis and identification of parameter lengths and values shown in Figure 10.

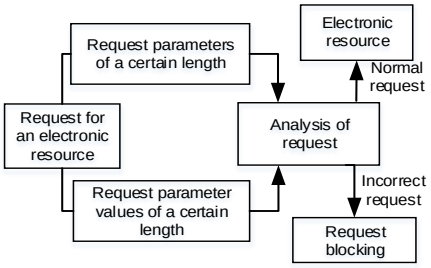

Fig. 10. Detection and protection of electronic resource based on analysis of parameter Lengths and Values.

Here is an example of how it works. Benchmark query with parameter lengths and values set:

```
GET http://site.uz/5.php HTTP/1.1
Host: site.uz
User-Agent: Mozilla/5.0 (Linux; Android 5.0;
SM-G900P Build/LRX21T) AppleWeb-Kit/537.36
(KHTML, like Gecko) Chrome/80.0.3987.149
Mobile Safari/537.36
Request modified by an attacker:
GET
http://site.uz/config.php?name=kamil<script>
alert</script>&surname=kerimov'union
pass=''HTTP/1.1
Host505: site.uz?test=passwd
User-Agent506: Mozilla/5.0
```

The analyzer checks the parameter length and HTTP/HTTPS request values accordingly. As you can see from the modified request, the length of the parameters and their values do not match the benchmark. As a result, the request blocked.

If a request to an electronic resource $\in K_n$, where n is from 1 to 17, the request will be allowed to the electronic resource,

otherwise it will be blocked.

```
select * from usertab where
uid=' ''or a=a --' …
```

III. Result and Discussion

This section details the research results of the proposed security mechanism against data entry on the serve. Such attacks as SQL Injection must use a logical approach and reasonable input val-ues, together with the disruption of special characters of the source program accompanied by a normal SQL-query, to provoke the return of a tautologically correct value. If this type of attack applied to the authentication login page, the identity authentication mechanism can easily avoided and the login can successfully logged in. It means that the hacker has a legal right to access the system resource.

The most fundamental solution to defend against such attacks is to strengthen the verification mechanism of the application program. All input requires detailed checks to determine the purity of the input data values before passed to the downstream program for subsequent execution. In this way, we can eliminate the possibility of malicious input attacks. Regarding database access, experience shows that passing an SQL-query by string concatenation not only leads to security problems, as mentioned above, but also inadvertently leaks information about the database structure and the logical way the program works. A better approach is to send parameters to gain access to the database, together with checking the input parameters. This is more effective in preventing security problems. Control of responses to error messages need to strengthened. The user does not need to see much information. They can gather error messages from the database by trial and error, and then refine the attack.

The check must be strengthen for input values of a parameter if they contains special characters. For example, further processing is required for single quotes ('), semicolons (;) and left slashes (\) so that the combination of SQL syntax with these special characters can be treated as a word or sequence rather than part of language syntax

```
select * from usertab where uid = '1989' …
```

or grammar. Taking Citrix as an example, in its special character handling mechanism, if single quotes (") is encountered, an extra single quote is added before the character. Thus, the original characters will become purely symbolic because of this inverted comma. Combining the SQL-query in the process will not lead to unexpected results because of the logical judgement. For example, the original SQL looks like this:

If the malicious user input uid - with single quotes to replace 1989, such as 'or a = a--, the original sentence will become:

```
select * from usertab where uid=’ ’or a=a --
```

The backend server will treat uid as an empty sequence and the grammar structure will return tautological TRUE; after contacting special characters acquire the following grammar:

Despite the fact that there is still a grammatical structure problem, at least it does not make the uid parameter values larger than a true logical constant. Similarly, whenever "\" (backslash) is encountered, we can simply insert an extra slash to the left before the character to take it out of subsequent characters; in case ";" (semicolon) is found, we can simply remove the characters to prevent the SQL syntax character from being erased, so it will not cut off the normal and unfinished statement following the semicolon

IV. Conclusion

This paper correlates the current prevailing methods according to their vulnerability to attacks and suggested protection attributes. We considered attacks based on string and command line operations and the corresponding protection mechanisms. If appropriate controls implemented correctly, it will effectively reduce the injection of SQL, cross-site scripting and other attacks. This results in a secure system environment. In conclusion, we can note the following results:

- The concept of adaptive protection of electronic resources from threats to information security, which allows to develop measures for the adaptive protection of resources using both signature and behavioural analysis, is proposed;
- An adaptive model of protection of electronic resources from threats to information security based on behavioral analysis was developed, this model allows to protect electronic resources from both those IS threats which already exist in the database, and new types of IS threats.

The developed adaptive mechanism will help to identify and make decisions on IS threats, with well-established and manageable means. Application of adaptive protection carries out control over popular IS threats and in time to provide protection and notification of the administrator, also such protection allows to apply a certain protection proceeding from type of IS threat. That is, adapt to the IS threat and apply the right protection.

References

[1] A. Makiou, Y. Begriche and A. Serhrouchni, "Improving Web Application Firewalls to detect advanced SQL injection attacks," presented at 10th International Conference on Information Assurance and Security, Japan, 2014.

[2] E. Raff, J. Barker, J. Sylvester and R. Brandon, "Malware Detection by Eating a Whole EXE," presented at the Workshops of the Thirty-Second AAAI Conference on Artificial Intelligence, Ithaca, NY, 2017.

[3] M. Ito and H. Iyatomi, "Web application firewall using character-level convolutional neural network," presented at the 14th International Colloquium on Signal Processing & Its Applications (CSPA), Penang, Malaysia, 2018.

[4] K. Pranathi, S. Kranthi, A. Srisaila and P. Madhavilatha, "Attacks on Web Application Caused by Cross-Site Scripting," presented at the 2nd International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2018.

[5] P. N. Joshi, N. Ravishankar, M. B. Raju and C. N. Ravi, "Encountering SQL Injection in Web Applications," presented at the 2nd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2018, pp. 257-261.

[6] A. Jana, P. Bordoloi and D. Maity, "Input-based Analysis Approach to Prevent SQL Injection Attacks," presented at the 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 2020.

[7] B. I. Mukhtar and M. A. Azer, "Evaluating the Modsecurity Web Application Firewall Against SQL Injection Attacks," presented at the 15th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 2020.