A SURVEY OF E-COMMERCE SECURITY THREATS AND SOLUTIONS

Stanislav Dakov[1], **Anna Malinova**[2]

Abstract: E-commerce security is part of the Web security problems that arise in all business information systems that operate over the Internet. However, in e-commerce security, the dimensions of web security – secrecy, integrity, and availability-are focused on protecting the consumer's and e-store site's assets from unauthorized access, use, alteration, or destruction. The paper presents an overview of the recent security issues in e-commerce applications and the usual points the attacker can target, such as the client (data, session, identity); the client computer; the network connection between the client and the webserver; the web server; third party software vendors. Discussed are effective approaches and tools used to address different e-commerce security threats. Special attention is paid to Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), phishing attacks, SQL injection, Man-in-the-middle, bots, denial-of-service, encryption, firewalls, SSL digital signatures, security certificates, PCI compliance. The research outlines and suggests many security solutions and best practices.

UDC Classification: 004.42, **DOI:** https://doi.org/10.12955/pns.v2.135

Keywords: e-Commerce, security, user experience.

Introduction

Nowadays, a significant amount of internet traffic is used for surfing e-commerce websites. The coronavirus pandemic situation has led to unprecedented growth of e-commerce during the lockdown of 2020. According to Statista (Statista, 2021), online retail websites have made strong traffic gains due to the global coronavirus pandemic. For instance, Amazon.com had a monthly traffic average of almost 3.68 billion visitors in 2020, followed by eBay.com with 1.01 billion visits on average each month. E-commerce sales are expected to reach $6.5 trillion by 2023 (Bhatti, 2020). This steady rise in the e-commerce retail market also means more exposure to e-commerce security violations.

Security is one of the most important aspects of an e-commerce business, and customers' trust is a top priority. Trust is essential to the users in their decision to risk time, money, and personal data on a website. E-commerce is expected to provide safe web browsing and secure transactions. To provide customers with the safest possible online shopping experience, there are some main security threats that e-commerce websites should deal with.

Internet security of web applications is generally considered to include three main elements: secrecy, integrity, and availability. Secrecy refers to protection against unauthorized data disclosure and ensuring the authenticity of the data source. Integrity is about prevention against unauthorized data modification. Availability refers to preventing data delays or denials (removal).

In this paper, we consider the main security threats and possible solutions focusing on how they apply to e-commerce. A common approach to investigate e-commerce security is to follow the transaction-processing flow, beginning with the consumer and ending with the webserver (or servers) at the e-commerce site, including any other computers connected to the web servers. This means exploring client threats, communication channel threats, and server threats. In our approach, we concentrate on the assets that must be protected to ensure security in e-commerce. The rest of the paper explores the main security threats and aims to identify the factors that enhance or damage the credibility of e-commerce sites.

Common security threats in e-commerce

E-commerce systems have several usual points that the attacker can target, such as:

- The client (data, session, identity).
- The client computer.
- The network connection between the client and the web server.
- The web server.
- Third-party software vendors.

[1] Faculty of Mathematics and Informatics, University of Plovdiv „Paisii Hilendarski", Bulgaria, stanislav.dakov@uni-plovdiv.bg

[2] Faculty of Mathematics and Informatics, University of Plovdiv „Paisii Hilendarski", Bulgaria, malinova@uni-plovdiv.bg

The client as target

The attack against the client is extremely common. It means taking control of the user's session and identity. This technique is also called "Session Hijacking". With this method, the attacker accesses the user's session and can do everything the authorized user can do on the e-commerce website.

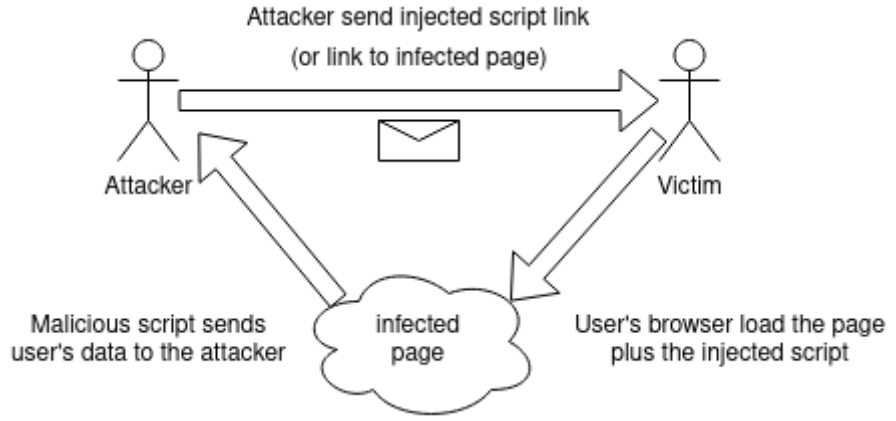
There are different ways to make "Session Hijacking" attack (OWASP, 2021):

- Predictable session token.
- Session Sniffing.
- Cross-Site Scripting (XSS).
- Man-in-the-middle attack.
- Man-in-the-browser attack.
- Cross-Site Request Forgery (CSRF).

Cross-Site Scripting (XSS)

XSS is a code injection technique (Figure 1). All sites that visualize information entered by a site user are endangered (Google Application Security, 2021), (Mozilla Web Security, 2021). Such sites are guest books, forums, blogs, e-commerce with feedback sections, sites that have comment sections etc. According to (OWASP, 2021), XSS attacks can be categorized into three types - reflected, stored and DOM based (Fox, 2012), (Manna, 2016). The goal of each of these attacks is to steal the user's information.

Figure 1: Cross-Site Scripting attack



Source: Author

Reflected XSS is the most common Cross-Site Scripting vulnerability. The attacker injects malicious JavaScript script directly into the client browser (Fox, 2012). This can happen in websites that output the user input data (Manna, 2016), e.g., search results, error messages, etc. An example, given below:

```
http://www.example.com/search.php?q=<script>document.location='https://a
ttacker.com/?cookie='+encodeURIComponent(document.cookie)</script>
```

When the user clicks on this link, it will open a website with a search engine that will output the injected script from the URL. The browser will run that script, and it will send all cookie data to the attacker.

Stored XSS is another type of XSS in which the attacker injects a script directly on websites (Rodriguez, 2019). When the attacker injects the script, it stays there permanently. The most common attacked section is the comment or the feedback section. When the victim opens that section, it will automatically run the malicious script in his browser.

DOM-based XSS attack usually happens when the website uses JavaScript to load data from untrusted sources and then writes it back to the DOM (Rodriguez, 2019).

To prevent this attack, Mohammadi (Mohammadi, 2019) suggests using unit tests to detect and repair Cross-Site Scripting vulnerabilities caused by incorrect encoder usage.

Cross-Site Request Forgery (CSRF)

CSRF is another quite common attack against the end-user. It is also known as a one-click attack or session riding. CSRF takes advantage of the trust between the client web browser and the web application. The attacker makes a hidden clone of a real form based on POST request, then it sets some

default values, and when the victim opens a link to that form, a JavaScript script automatically submits the form to the real website. In most cases, these attacks are executed on the functionality of websites that use form-based submissions like POST requests and cookie-based authentication (Bache, 2014).

Nowadays, almost all modern frameworks, like Laravel, Spring, etc., already provide protection against this kind of attack. The most common protections are:

- CSRF token - a randomly generated string by the web application (Calzavara, 2020), (Semastin, 2018), (Laravel, 2021), (Spring, 2021). Whenever the user makes any PUT/POST/DELETE request, it provides the token via an HTTP header X-CSRF-Token to the web application, or it can be used as a hidden field in the HTML form. Using this token, the web application can be sure that the request was made by the user itself. Each time the session is regenerated, this token is changed, so if malicious applications access it, they would not be able to use it.
- Cookies Attributes (SameSite) - according to (OWASP, 2021) we always have to use SameSite cookie attribute for session cookies. (Bulgarian government requirements, 2019) suggests the cookies should have a security flag, which instructs the browser that the cookie can only be accessed through secure SSL channels. There are other attributes that can be used like cookie prefixes (Google Chrome, 2021).

Attacks against the client's computer

There is a lot of malicious software that can be installed on the client's computer (Iliev, 2019). The attacker usually does it without the client-finding out. In many cases, a phishing attack is used to trick the user.

Phishing attacks are not like standard attacks that an attacker seeks in a web application vulnerability. Instead, they are aimed at the end-user. In most cases, these attacks are carried out due to the user's inattention. Attackers take advantage of that by trying to lure them into external malicious applications or simply installing malware without the user suspecting anything (Alotaibi, 2021). These kinds of attacks can be categorized into two main types (Itgovernance, 2021):

- Malicious attachments – emails with attached content installed as soon as they are opened (Bhavsar, 2018). Most such emails have enticing titles.
- Links to malicious websites are malicious links pointing to websites that are often clones of legitimate websites. Through them, users download and install malware. In many cases, these sites contain login pages through which attackers prepare scripts to collect credentials (Apandi, 2020).

There are different techniques of phishing attacks over the Internet. The most common are:

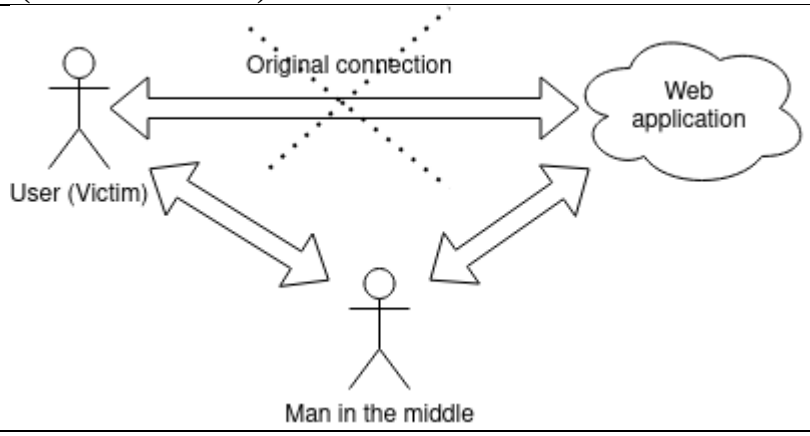
- Pharming/DNS cache poisoning
- Typosquatting/URL hijacking - the hacker makes a clone of a website and sends the URL with a typo in the name to the user (Bhavsar, 2018). For example, https://amazonn.com. The user will think that this website is Amazon, and it will try to log in. Instead of logging in successfully, the credentials will be sent to the attacker.
- Clickjacking/UI (user interface) redressing/iframe overlay - the hacker uses an extra layer on the website. The user may think that he clicks on a button on a trusted and secure site e.g., to make a purchase, but instead, malware is downloaded. Another example - the user enters their password or credit card number and inadvertently delivers them directly to the attacker.
- Tabnabbing and reverse tabnabbing

When malicious software of the same type is installed on a large number of client machines, they form a BotNET - a set of installed bots connected to each other. They are mostly used for Denial of Service (DDoS) attacks and are often used for spamming (Niranjanamurthy 2013).

Attack against the network connection between the client and the webserver

The most common attack against the connection between the client and the website is Man-in-the-middle (MITM) (Figure 2). The attacker masks himself as both endpoints that are sending details to each other. It sniffs the network traffic to catch the ongoing communication. If the network communication is not secure, the attacker has full access to all communication transferred data in each request and response. Attackers can send, intercept, and receive data without the awareness of the sender and receiver. This is a type of eavesdropping and exposes real-time conversations or data transfer.

Figure 2: MITM (Man-in-the-Middle) attack



Source: Author

Suggestions to avoid this attack:

- Surf in websites using HTTPS.
- Do not use public Wi-Fi hotspots.
- Use VPN.
- Check the System Security - malware and spyware get installed in a computer when your system is not adequately protected using an antivirus program.

The web server as a victim

- SQL injection

The website's databases can contain emails, passwords, card information, and more private data. There are different ways to get access to the database, but the most common is to use SQL injection. With this attack, the hacker can take the user's credentials and has access to the whole database, which means all products, user details, stored credit cards, and more confidential information. This technique exploits a security vulnerability occurring in the database layer of an application (Halfond). Hackers use injections to obtain unauthorized access to the underlying data structure, and DBMS. SQL injection is the most famous type of injection attack which also counts LDAP or XML injections (Towson University, 2021). The idea behind SQL injection is to modify an application's SQL (database language) query (Alenezi, 2020) in order to access or modify unauthorized data or run malicious programs. For example, the SQL below authenticates users. This is common in many (not properly secured) web applications:

```
myQuery=" SELECT * FROM user WHERE username = 'username_value' and
password ='password';"
```

Suppose we replace 'username_value' with 'OR 1=1'). In that case, the attacker will have access to the database without knowing the real username and password because the assertion "1=1" is always true and the rest of the query is being ignored by the comment character.

There are many techniques to prevent SQL injection (Boydand, 2004), (Chen, 2021), but the most popular is preparing the SQL query before execution.

Some of the data can be encrypted to protect the user (Shmueli, 2010). Database encryption can be divided into two basic types:

- Transparent/External Encryption - represents the encryption of the entire database. This is done by native encryption functions within the database engine. This is called 'transparent' database encryption because it is invisible to the applications. It is used to prevent exposure of information due to loss of the physical media or compromise of the database files in storage.
- User/Data Encryption - encryption of specific columns, tables, or even data elements within the database. The goal is to provide protection against inadvertent disclosure. The concept is to encrypt only the highly sensitive data we are worried about, reducing the overall performance impact and minimizing code and database changes.

Encryption techniques can be used to enhance databases by focusing on their respective targets for encryption. There are several levels of database encryption: Cell-Level; Roll-Level; Column-Level;

Tablespace-Level; File-Level. User`s data must be well stored in the database, so the passwords must always be irreversibly encrypted. There are different encryption algorithms for that purpose (IBM, 2021), (OWASP, 2021):

- Salted SHA-1
- SHA-1
- MD5
- bcrypt
- SHA-2
- Salted SHA-2
- Argon2 - it is a relatively new algorithm and has three variants: Argon2d, Argon2i and Argon2id. Argon2i is optimized for password hashing. Argon2 has 6 input parameters. In July 2015 the Argon2 was the winner of the Password Hashing Competition (Wetzels, 2016).

Sometimes we can use more than one algorithm together like bcrypt(sha256($password)), but generally, this is not recommended.

Most of the hash algorithms like LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults are already cracked and can be reverted quite easily.

If someone has access to the user's email and password, it will be able to take advantage of the current account and check the network for other existing user accounts. In many cases, the passwords among different applications match, which is a big hole in the security of the user. When an account is stolen, the hacker can access all of its other registrations, including bank accounts. In some cases, user's emails can also be stolen for malicious purposes. Sometimes they are sold to SPAM advertising companies. Apple (Apple, 2020) has found a way to prevent that with virtual emails. When the client registers on a site using "Sign with Apple ID" service, this service generates a virtual email. Thus, even if the entire database is stolen, hackers will not be able to take advantage of the user's real email and later to check for other user accounts associated with that email.

- Bots

Bots are also a common threat against a website. There are also useful bots over the web, such as Googlebot, but most are malicious ones that look for vulnerabilities in the web application. Once they find a vulnerability, they execute simple attack patterns.

There are different types of bots (Imperva, 2021):

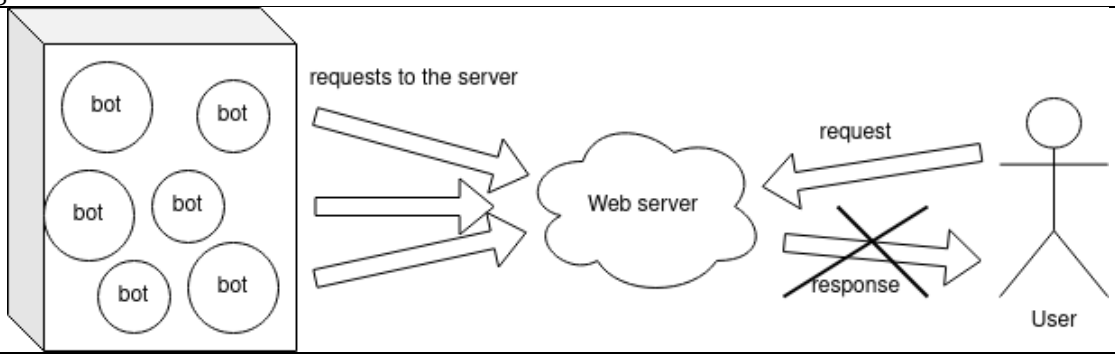
- Spider Bots
- Scraper Bots
- Spam Bots
- Social Media Bots
- Download Bots
- Ticketing Bots

Bots often go around shopping sites and check for security breaches, such as comments. If they see that a comment can be posted without a registration requirement, or at least a CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart), they start generating many comments with links to malware or advertising pages. They are also often used for XSS attacks. According to (Rahman, 2021), the price scraping bots are a big threat. A large amount of the e-commerce traffic is generated by content scraping bots (Rahman, 2021). To protect from this kind of attack (Rahman, 2021) suggests using CAPTCHA, machine learning or genetic algorithms. Other bots check for open ports on the host server itself. Once they find an open port, e.g., SSH, FTP, etc., they start generating large number of authentication attempts.

- Denial-of-service (DOS)

Sometimes the cluster of bots makes a botnet attack which is also known as Denial-of-service (DOS) attack (Figure 3) (Rovetta, 2020). Usually, the attacker sends a lot of requests to the server, which must respond to them, and during that time, it might be unavailable for any other requests. Also, the attack can be done by depleting resources or by taking advantage of a bug in the victim's software.

Figure 3: Denial-of-service



Source: Author

The purpose of this attack is to disrupt the server's normal operation or stop it completely.

Common types of DoS attacks:

- Smurf Attack
- SYN Flood Attack
- UDP Flood Attack
- Plashing
- IP Fragmentation Attack

To avoid this attack, some good solutions may be implemented:

- Attack detection
- IP Whitelisting
- Blacklisting
- Rate Limiting

Third-party software vendors

Payment plays the biggest role in the user experience. The user must be sure and reassured that the transaction will take place and will not be compromised. There are different payment methods for e-commerce stores. It is important to choose the correct one or to have the option to let the customer choose how to pay. The end-user has to be sure that the transaction is safe.

The most common payment methods:

- Credit / Debit / Prepaid card payments are the most common and most insecure payments. There are many ways to compromise this type of payment and steal card information.
- Bank transfers - some stores offer consumers to make a direct bank transfer by providing an IBAN.
- E-Wallets - this service obliges the user to log in to an external payment service such as PayPal, Alipay, ePay, Amazon Pay. They provide a high level of security.
- Cash - the most secure payment. After ordering, the user pays upon delivery of the product. The absence of such a payment method might make many consumers refuse to buy a product.
- Cryptocurrencies - nowadays, more and more stores offer the option to pay with cryptocurrency. It is of growing interest among young consumers.
- Direct carrier payments - some stores offer payment through a mobile operator. A rare method but quite secure. This method of payment is most common in mobile application stores.

Many frameworks offer ready-made payment solutions. They strictly comply with the requirements of The Payment Card Industry Data Security Standard (PCI DSS) (Shopify, 2021). PCI compliance is a standard which major players created in the credit card industry in 2006. It ensures that all online businesses that process, store, and transfer credit card information implement some requirements such as (Magento, 2021):

- Use a firewall configuration to protect cardholder data.
- Do not use vendor-supplied defaults for system passwords and other security parameters.
- Protect stored cardholder data.
- Encrypt transmission of cardholder data across open, public networks.
- Restrict access to cardholder data by business need-to-know.

- Restrict physical access to cardholder data.
- Track and monitor all access to network resources and cardholder data.

There are many actions that can be taken against most threats. Some of them were already discussed in previous sections. Below we consider the following proven practices: firewall, digital signature, secure socket layer (SSL).

Firewall

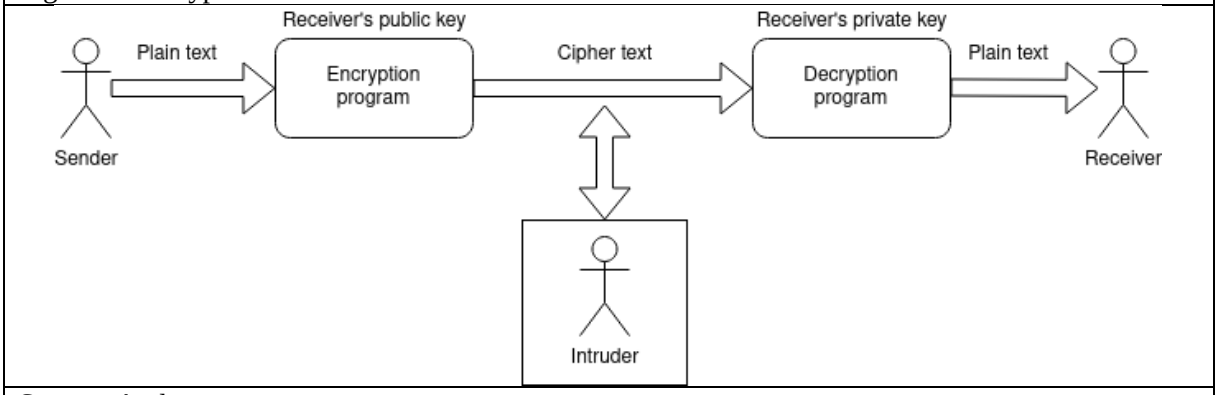
Firewalls carefully analyze incoming traffic based on pre-established rules and filter traffic coming from unsecured or suspicious sources to prevent attacks (Nife, 2020). Firewalls guard traffic at a computer's entry points (ports). To protect web servers from attacks, firewalls block access using ports different from 80 and 443 (Nycz, 2017). Firewalls can either be software or hardware. A software firewall is a program installed on each computer that regulates traffic through port numbers and applications, while a physical firewall is a piece of equipment installed between your network and gateway. It's best to have both. A web application firewall (WAF) helps protect a company's web applications by inspecting and filtering traffic between each web application and the internet. A WAF can help defend web applications from attacks such as Cross-Site Request Forgery (CSRF), Cross-Site Scripting (XSS), file inclusion, and SQL injection.

Digital signature

The digital signature is based on asymmetric cryptography, in which each user has encryption (public) and decryption (private) key (Aldeem 2018). Everyone can access the public key. Anyone who wants to send a message can use it, but only the user who has the private key can read it (Figure 4). The most common use of Digital Signature is for software distributions, financial transactions, and in other cases - to detect forgery and tampering (Lasheen, 2018). The digital key has three main purposes (Dzhangarov, 2020):

- Authentication - gives the recipient reason to believe that the sender sent the message
- Non-repudiation - through the digital backfill, you can certainly prove who owns the message.
- Integrity - the digital signature protects the integrity of the message by preventing it from being altered in its transfer.

Figure 4: Encrypted communication



Source: Author

But there is one weakness in digital signature - lack of authentication. Digital signature cannot verify the identity of the real sender and his public key. The solution for that is digital certificates. Digital certificates can verify the identity of the sender and that the public key belongs to them. In this way, attacks like Man in the middle can be prevented. Digital certificates are used in Secure Socket Layer (SSL).

Secure Sockets Layer (SSL)

SSL provides a security "handshake" in which the client and server computers exchange messages. It is the standard technology for keeping an Internet connection secure and safeguarding any sensitive data that is being sent between two systems, thus preventing criminals from reading and modifying any information being transferred (Arai, 2015), (Google Application Security, 2021), (Mozilla Web Security, 2021). The two systems can be a server and a client (e.g., a shopping website and browser) or server to server (e.g., an application with personally identifiable information or with payroll information)

(Dastres 2020). If a website doesn't use SSL certificate, then most modern browsers will mark it as "Not Secure". SSL protects information by encrypting the data transfer between the visitor's browser and the website. When a user visits an SSL/HTTPs website, the browser first verifies if the website's SSL certificate is valid. If everything checks out, then the browser uses the website's public key to encrypt the data. This data is then sent back to the intended server (website), where it is decrypted using the public key and a secret private key. The SSL prevents most of the phishing and Man in the middle attacks (Devi, 2020). SSL can prevent session hijacking as well, which is also commonly known as cookie hijacking. SSL encrypts the data on a website login page, which prevents hackers from finding out the password. This method is especially effective for banks and e-commerce sites (Nycz, 2017).

Conclusion

E-commerce is vulnerable to a wide range of security threats and, with the advance of AI and machine learning, new threats emerge every day. Effective actions need to be taken to address them. However, some of the threats are aimed directly at consumers. Nothing can be done there; it all depends on the watchfulness and awareness of the users. The human factor cannot be eliminated. Customers must follow instructions that can protect them against threats as much as possible. Consumers' awareness includes:

- Always shop on sites with HTTPS.
- To make sure that the site they shop is secure, there is a story with positive comments.
- To watch out for external links that redirect them to other sites.
- Be careful whenever they receive an email redirecting to a website.

As for the e-commerce sites, they must follow certain rules and hygiene to protect against attacks. They must maximally implement all imposed standards for data and user security. No site is immune to attack. The more functionality a site has, the more security attack opportunities it provides.

Acknowledgments

This paper is supported by the project FP21-FMI-002 of the Scientific Fund of the University of Plovdiv Paisii Hilendarski, Bulgaria.

References

Aldeen, F., Nasar, K.,Saeed, A.,Maheboob, S. (2018). Digital signature system, Journal of Resource Management and Technology, 2018, 09(02), 14-16.

Alenezi, M., Nadeem, M., Asif, R. (2021). SQL injection attacks countermeasures assessments, Indonesian Journal of Electrical Engineering and Computer Science, 2021, 21 (2), 1121-1131.

Alotaibi, A., Alsuwat , E. (2021). A study on social engineering attacks: phishing attack, International Journal of Recent Advances in Multidisciplinary Research, 2021, 7, 6374-6380.

Apandi, S., Sallim, J., Sidek, R (2020). Types of anti-phishing solutions for phishing attack, IOP Conference Series Materials Science and Engineering, 2020, 769, 012072.

Apple. (2020). Hide My Email for Sign in with Apple. Retrieved on March 5, 2021, from https://support.apple.com/en-us/HT210425

Arai, M. (2015). Development and Evaluation of Secure Socket Layer Visualization Tool with Packet Capturing Function, International Journal of Future Computer and Communication, 2015, 3(3), 06004.

Bache, B. (2014). Cross-Site Request Forgery on Android WebView, IJCSN International Journal of Computer Science and Network, 2014, 3, 119-124.

Bhatti, A., Akram, H., Basit, H.M. (2020). E-commerce trends during COVID-19 Pandemic, International Journal of Future Generation Communication and Networking, 2020, 13, 1449–1452.

Bhavsar, V., Kadlak, A., Sharma, S. (2018). Study on Phishing Attacks, International Journal of Computer Applications, 2018, 182(33), 27-29.

Boydand, S., Keromytis, A. (2004). SQLrand: Preventing SQL Injection Attacks, Applied Cryptography and Network Security, ACNS 2004, Lecture Notes in Computer Science, vol. 3089. Springer, Berlin, 292-302.

Bulgarian government requirements. (2019). Ordinance on the minimum requirements for network and information security. Retrieved on March 5, 2021, from https://www.mtitc.government.bg/sites/default/files/nar_minimalnite_iziskvaniq_mrejova_info_sigurnost-072019.pdf (in Bulgarian)

Calzavara, S., Conti, M., Focardi, R., Rabitti A., Tolomei, G. (2020). Machine Learning for Web Vulnerability Detection: The Case of Cross-Site Request Forgery, IEEE Security and Privacy Magazine, 2020, 18(3), 8-16.

Chen, D., Yan, Q., Wu C., Zhao, Z. (2021). SQL Injection Attack Detection and Prevention Techniques Using Deep Learning, Journal of Physics: Conference Series, 2021, 1757, 012055.

Dastres, R., Soori, M. (2020, October). Secure Socket Layer in the Network and Web Security, World Academy of Science, Engineering and Technology, International Journal of Computer and Information Engineering, 2020, 14(1), 330-333.

Devi, O., Vallabhaneni S., Hussain, M. ,Kumar, T. (2020). Security Analysis on Remote Authentication against Man-in-the-MiddleAttack on Secure Socket Layer, IOP Conference Series: Materials Science and Engineering, 2020, 981, 022015.

Dzhangarov, A., Suleymanova, M. (2020). Electronic digital signature, IOP Conference Series: Materials Science and Engineering, 2020, 862, 052054.

Fox, D. (2012). Cross-Site Scripting (XSS), Datenschutz und Datensicherheit – DuD, 2012, 36, 840.

Google Application Security. (2021). Cross-site scripting. Retrieved on April 10, 2021, from https://www.google.com/about/appsecurity/learning/xss/

Google Chrome. (2021). Cookie Prefixes Sample. Retrieved on March 3, 2021, from https://googlechrome.github.io/samples/cookie-prefixes/

Halfond, W. G., Viegas, J., Orso, A. (2006). A Classification of SQL-Injection Attacks and Countermeasures, IEEE, Computer science, 2006, 5969227.

IBM. (2021). Password encryption. Retrieved on March 4, 2021, from https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_74/rzahy/rzahypwdencrypt.htm

Iliev, A., Kyurkchiev, N., Rahnev, A., Terzieva, T. (2019). Some models in the theory of computer viruses propagation, LAP LAMBERT Academic Publishing, 2019, ISBN: 978-620-0-00826-8.

Imperva. (2021). Bots. Retrieved on March 5, 2021, from https://www.imperva.com/learn/application-security/what-are-bots/

Itgovernance. (2021). Phishing attacks and how to avoid them. Retrieved on March 3, 2021, from https://www.itgovernance.co.uk/phishing

Laravel. (2021). CSRF Protection. Retrieved on March 2, 2021, from https://laravel.com/docs/8.x/csrf#csrf-introduction

Lasheen, I. (2018). Digital signature in E-Commerce security, Middle East Journal for Scientific Publishing, 2018, 1(1), 26-34.

Magento. (2021). Magento's Approach to PCI Compliance. Retrieved on March 4, 2021, from https://magento.com/pci-compliance

Manna, M.,Hussein, R (2016). Preventing Cross-Site Scripting Attacks in Websites, Asian Journal of Information Technology, 2016, 15(16), 2797-2804.

Mohammadi, M., Chu, B., Lipford, H. (2019). Automated Repair of Cross-Site Scripting Vulnerabilities through Unit Testing, 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2019, 370-377.

Mozilla Web Security. (2021). Web Security. Retrieved on April 10, 2021, from https://developer.mozilla.org/en-US/docs/Web/Security

Nife, F., Kotulski, Z. (2020). Application‐Aware Firewall Mechanism for Software Defined Networks, Journal of Network and Systems Management (2020) 28, 605–626

Niranjanamurthy, M., Chahar, D. (2013). The study of E-Commerce Security Issues and Solutions, International Journal of Advanced Research in Computer and Communication Engineering, 2013, 2(7), 1:12.

Nycz, M., Hajder, M., Nienajadlo, S. (2017). Methods for increasing security of web servers, Annales UMCS, Informatica, 2016, 16(2), 39-42.

OWASP. (2021). Cheat Sheet Series. Retrieved on March 10, 2021, from https://cheatsheetseries.owasp.org/index.html

Rahman, R. (2021). Threats of price scraping on e-commerce websites: attack model and its detection using neural network, Journal of Computer Virology and Hacking Techniques, 2021, 17, 75–89.

Rodriguez, G., Torres, J., Flores, P., Benavides, E. (2019). Cross-Site Scripting (XSS) Attacks and Mitigation: A Survey, Computer Networks, 2019, 166, 106960.

Rovetta, S., Suchacka, G., Masulli, F. (2020). Bot recognition in a Web store: An approach based on unsupervised learning, Journal of Network and Computer Applications, 157, 102577.

Semastin, E., Azam, S., Shanmugam, B., Kannoorpatti, K., Jonokman, M., Samy, G., Perumal, S. (2018). Preventive Measures for Cross-Site Request Forgery Attacks on Web-based Applications, International Journal of Engineering & Technology, 2018, 7(4), 130-134.

Shmueli, E., Vaisenberg, R., Elovici, Y., Chanan Glezer, C. (2010). Database Encryption – An Overview of Contemporary Challenges and Design Considerations, ACM SIGMOD Record, 38(3), 29-34.

Shopify. (2021). PCI Compliance. Retrieved on March 4, 2021, from https://www.shopify.ie/security/pci-compliant

Spring. (2020). A Guide to CSRF Protection in Spring Security. Retrieved on March 2, 2021, from https://www.baeldung.com/spring-security-csrf

Statista. (2021). Top retail websites by global traffic 2020. Statista. Retrieved on March 3, 2021, from https://www.statista.com/statistics/274708/online-retail-and-auction-ranked-by-worldwide-audiences/

Towson University (2021), SQL Injections–Introduction. Retrieved on March 5, 2021, from http://cis1.towson.edu/~cssecinj/modules/other-modules/database/sql-injection-introduction/

Wetzels, J. (2016). Open Sesame: The Password Hashing Competition and Argon2, IACR Cryptol. ePrint Arch., 2016, 104.