EasyChair Preprint

№ 3717

Security Aspects of e-Payment System and Improper Access Control in Microtransactions

Md. Asaduzzaman

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

July 3, 2020

Security Aspects of e-Payment System and Improper Access Control in Microtransactions

Md. Asaduzzaman
Dept. of Computer Science & Engineering
Military Institute of Science and Technology
Dhaka-1216, Bangladesh
asadbd45@gmail.com

***Abstract*—E-payment system has paved the way for many problems of physical money transfers. Nowadays financial services are one of the most attractive targets for cyber attackers. Some involved components (sub-systems) with the e-payment system are- customer, merchant infrastructure, payment service provider and banking server. In this paper, a study of security aspects of these involved components is conducted. It is found that attack on customer can be carried out by lower-skilled attackers and a specific system will face a limited loss. On the other hand, other components can be compromised with less effort by high skilled attackers which can have a devastating effect on the financial infrastructure. A closer look is given at the improper access control in e-payment system, which will give a proper idea about the attackers' entry points from an attacker's point of view. It also shows how an attacker escalates such an ignorant flaw to gain financial benefit.**

***Index Terms*—Online Payment, Access Control, Ecommerce Security, Online Transaction**

I. INTRODUCTION

Online payment systems have brought a revolutionary change in social life. It has also broadened the area of e-business. About 70% of the bill payments, 80% of the P2P transfers and 80% of the cash-out and cash-in was performed globally according to a report of September 2018 [1]. A significant amount of other payments including government tax and disbursements are being paid with e-money. Each of the component of an online payment system is a separate sub-system that works independently and communicates with other sub-systems using application programming interfaces (APIs). We denote each different entity (i.e. Customer, Merchant, Payment Gateway, Bank Server etc.) of the system as a component. Holloway et al. depicted a payment system configuration where a payment utilizes internet to enable e-banking and thus a customer can buy products from a merchant [2]. The system involves customer, merchant, payment gateway and the bank server which can be said to be the components of the system. Security is one of the greatest concerns in case of online banking and online payment systems. Nowadays most of the cyberattacks are occurring on the financial infrastructures. Cyberattack also has a greater impact on the user adaption of online payment systems. Lai et al. conducted an empirical study which shows that security is one of the significant factors that has impact on consumers' intention to adapt a payment system [3]. A survey by Alshehri et al. showed that security and payment has influence on the e-commerce usage [4]. Social engineering attacks are found to be effective to deploy mass targeted attacks. To carry out social engineering attack, attackers are to invest a lot of time and effort on the targets. On the other hand, during mitigating the social engineering attacks, other application security risks remain to be ignored. Although preventive measures against some popular attacks are taken, other unpopular attacks remain to be unnoticed. Most of the automated security analysis tools are unable to detect many vulnerabilities that resides in the code and manual testing is required to mitigate these. Broken Access Control is such an attack that is listed in number 5 of OWASP top 10 vulnerabilities [5]. In case of online payment systems, the attack involves three or more components (e.g. merchant system, payment gateway, intermediate systems, bank) which enhances the scope of the attack. For an attacker it becomes easier to find new bugs in any of the systems and launch attack. Moreover, fixing the flaws and visualizing the procedure of the whole system is beyond the control of a single component. As a result, proper implementation of the APIs cannot be carried out which results in broken access control vulnerabilities.
In this research, a study on the security aspects of the components of e-payment system is conducted. Technical aspects of improper access control among the components of online payment systems are also described, which will give a proper idea of the entry point of an attacker from an attacker's point of view. It shows how an attacker uses such an ignorant flaw to gain financial benefit and why it is a matter of concern.

This paper is organized as follows: Sect. II reviews related work in brief. We describe the technical aspects of online payment systems in Sect. III. Section IV shows the attack scenario on the involved components. Conclusion and future work of this paper is presented in Sect. V.

II. LITERATURE REVIEW

Financial infrastructure is one of the major targets of the attackers these days. There are a number of vulnerabilities and attacks that affect m-payment systems. Shivani Agarwal et al. depicted a taxonomy of vulnerabilities that affect m-payment system [6]. The vulnerabilities and attacks include man-in-the-middle attack, replay attack, repudiation, impersonation

and unauthorized access. The research also described the reasons for these attacks. The taxonomy is nicely written in the paper but there is a lack of technical aspects on design flaws in m-payment system. Social engineering attack and man-in-the-middle attacks (including wi-fi intrusion) are found to be having hazardous effect in case of mass targeted attacks. Researchers devised detection and in some cases, prevention strategies for the social engineering [7] [8] [9] [10] and man-in-the-middle attack [11] [12]. In most of the cases almost all components of a payment system interact with each other using web-based APIs in http protocol. SSL and SET are two protocols that provide security in http for online transaction [13]. SET is a protocol for secure transaction where customer, seller, gateway, issuer and certificate authority are involved but it is not recommended for micro payments for its time-consuming nature. SSL also makes transaction secure from man-in-the middle attack. As a part of the system, security of web applications plays an important role to make the whole system secure. security analysis tools and WAFs provide the scope to find popular and known vulnerabilities of web applications like injection [14] and plugin flaws [15]. Aspen Olmsted investigated problem of e-commerce security constraints in distributed cloud system and constraints were expressed as hardening against unwanted meaning, scripted or deleted activities, fake Users, agent spoofing etc [16]. However, in this paper we limit our research to the security of traditional e-payment systems that use payment service providers. Researchers also devised many effective methodologies to detect access control vulnerabilities in web application. But the implementation is much more complex for cross server systems [17] [18] like e-payment system. PCI-DSS devised clear instructions of best practices for securing e-commerce [19]. Potential security flaws can be appeared due to improper implementation in the proposed practices by PCI-DSS, which will be discussed later. Mimi Wang et al. proposed an effective approach to find vulnerable points of e-commerce transaction system using Petri net and used a dynamic slicing method to locate the vulnerable points [20]. On another research, Mimi Wang et al. conducted vulnerability analysis of e-commerce systems by a dynamic data slice approach considering both transaction and data state consistency [21] and proposed methods to detect vulnerabilities. But both of the works yet to be implemented in real world systems. Faisal Nabi et al. nicely described technical and logical vulnerabilities attack taxonomy in component based e-commerce system that uses CSB based web application [22]. The research showed that design flaws are hard to be fixed or detect by any vulnerability tools or web application code scanning tools. Improper access control is such a design flaw. So, the only way to defend a system from the threats of improper access control is to design a secure system by giving special considerations to some specific locations. So, this paper will focus identifying the locations of potential vulnerable points of e-payment system.

III. SECURITY ASPECTS OF ONLINE PAYMENT SYSTEMS

In a e-payment system several sub-systems interact with each other. The sub-systems are customers' browser, e-commerce website, payment solution provider (also known as gateway) and online banking server. The whole system is depicted in figure 1. Security aspects and some important security parameters of the components of e-payment system are given in table I. The result is obtained from the literature study and practical analysis.

A. Customers' Browser

Customer orders a product and initiates the transaction process. Customer's security can be compromised in several ways. Social engineering, man-in-the-middle which includes phishing, pharming, sniffing etc. are effective attack methodologies to compromise a customer's browser. In these attacks, attacker tricks the customers to give their credentials or access to his payment account. In almost all approaches user interaction is required to carry out such attack [23]. Potential financial loss due to cognitive hacking is low for a specific system but it has a greater impact on global perspective [24]. Anthony Luvanda et al. proposed solutions to identify the techniques associated with man in the middle attacks on user devices but the solutions rely heavily on human intervention [25]. Also a survey showed that most of the respondents have fundamental knowledge of security risk in terms of disclosing their online bank transactions details and other components (PSP, bank) should improve their security [26]. The attacker has to depend on the response of a customer and invest effort and time that leads to a successful attack. Customer's browser can also be infected with malware that gives the attacker privilege to perform unauthorized transactions. All of these attacks directly affect a customer and the loss is to be carried solely by the customer. User awareness is the first and foremost thing that can help dealing with these attacks.

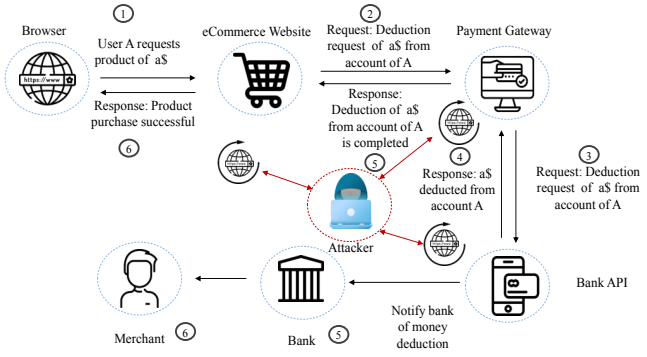

Fig. 1. E-payment example in e-commerce system.

B. Merchant Website

E-commerce websites often uses frameworks or content management systems that provides the website security by design. Many severe attacks including sql injection, os injection,

TABLE I
SECURITY ASPECTS AND PARAMETERS OF THE COMPONENTS OF E-PAYMENT SYSTEM.

Parameters	Attack on Customer	Attack on Merchant Infrastructure	Attack on PSP	Attack on Banking Server
User Interaction needed	Yes	Very less/ No	No	No
Potential Financial Impact	Low	Medium	High	High
Risk of MITM	High	Medium	Low	Low
Attack Effort Needed	High	High	Medium	Medium
Attacker's required expertise level	Low	High	High	High
Attack Success	Medium	Low	High	Medium
Patch release/ fixing effort	High	Medium	Medium	Low
Vulnerability requirement	Low	High	High	High
Example of attacks	Phishing, Malware, MITM, Pharming, Fraud	Injections, Forgeries, Broken access control, XXE, RCE, SDE	Injections, Forgeries, Broken access control, XXE, RCE, SDE	Injections, Forgeries, Broken access control, XXE, RCE, SDE, Fraud
Prevention Methods	Awareness, SSL, SET, Antivirus, Security Patch Update	Regular VAPT, Patch update, plugin and framework update	Regular VAPT, Security by Design, Fixing flaws, Encryption, patch update	Awareness, Regular VAPT Security by Design, Fixing flaws, Encryption, patch update

XSS, CSRF etc. are nicely handled by the latest frameworks and content management systems like magento [27],shopify [28], wordpress etc. The frameworks provide multi-purpose security protection. But one of the major problems with the framework and content management systems is zero-day vulnerability can be found at any module at any time. This may expose all the servers installed with that module to a risk of security. To deal with these vulnerabilities regular assessments are to be conducted and plugins should be updated regularly. There are a lots of paid security assessment tools in market like nessus [29], secubat [30] etc. There also free and open source tools like Nmap [31] that can be used for some specific content management systems. M. Asaduzzaman et al. also devised a solution to conduct more extensive scans for all content management systems [15]. Frameworks made tasks easier for e-business platforms by defending most common security vulnerabilities. Regular security assessment should be conducted to ensure the security of the e-commerce sites and to protect the system from various attacks.

There are other digital payment platforms (often web or mobile application) that allow users to add money to their account(including banks, mobile operators' bill, other utility bill) in the same manner of e-commerce payment transaction. Special concern should be paid on these platforms because whole process of transaction takes place digitally. No interaction is required between the merchant and the attacker unlike e-commerce sites. Furthermore, there is no standardized framework or content management system for these platforms and each security issue is to be handled carefully. For this purpose, dedicated third party security analysis team can conduct assessment to find hidden bugs in the system.

C. Payment solution provider

Payment solution provider or PSP(also known as payment gateway) acts an important role in online transaction process for micro transactions. PSP stands between the banking server and the merchant website and process the transaction. When an order is placed and payment procedure is started, the merchant site is redirected to the PSP that assigns a unique transaction id and an amount against the payment order. The id and amount are stored in the database of PSP and it is then redirected to the bank server along with the amount and the id. As the API is served as web service, any of the attacks that is applicable for web application can be conducted on the PSP. Developers and testers have to analyze the codes carefully. Each endpoint of the request must be tested to ensure the security from common vulnerabilities like injection, xss, broken access control etc. There is no effective automated solution that can find vulnerability in multiple servers at a time with such kind of dependency. Small mistake in program can lead the whole system to a severe vulnerability. PSP is the most common and popular point of entry for the attackers [32]. Special consideration should be given on the PSP in order to ensure security of the system.

D. Banking server

Payment requests are redirected to banking API (e.g. VISA, Master, Mobile Bank etc.) from PSP. Banking server assigns another id to a transaction request and stores it in the database upon a successful payment. The it redirects a success status to the PSP. In general, banks are aware of the security aspects. Although these are secure from critical vulnerabilities, in many cases small bugs remain in the APIs. Manual analysis is needed to deal with these issues. Vulnerability of a banking API can lead all the services that are taken from the bank to a security threat. Consistencies in input validation logic between apps and their respective web API services are needed to ensure security [33]. Security tester and developers have to pay attention to each of the API endpoints to mitigate the vulnerable points.

IV. ATTACK SCENARIO AND PAYMENT FLOW

A. Attack scenario on the involved components

The main involved components in a e-payment system are merchant site, PSP and banking server. Handling the consistency and security of the system is not an easy task

because flaw in any of the component can lead the whole system to a security risk. Improper access control can be very tricky bug. Attackers can take bigger advantage from a simple and small bug. The bug must be taken care of by the administrators of each of the components. Potential point of flaw of each of the component is described as follows-

a) Attack scenario in merchant site: Merchant site is responsible for its own bug. When a customer acts as an attacker, the attacker has more power to play with the system. Attacker can modify any data anywhere and send it to the PSP (payment gateway). The attack procedure is as follows-

1) An attacker creates an order of product of $b first and notes the order ID that goes to the PSP, let ID is xxx-xxxx-xxx (or cookie data).
2) The attacker creates another order of $a where $b<$a and order ID for $b is zzz-zzzz-zzz.
3) Before sending the request to the PSP, he changes the order ID to xxx-xxxx-xxx.
4) The further procedure is carried out in normal way.
5) Before the final request from PSP to the merchant site, he again changes the ID from xxx-xxxx-xxx to zzz-zzzz-zzz.
6) If improper access control exists in the merchant website, the order of $a will placed but money deduction amount will be $b. Where, $b<$a.

The whole procedure is depicted in figure 2. In many cases,

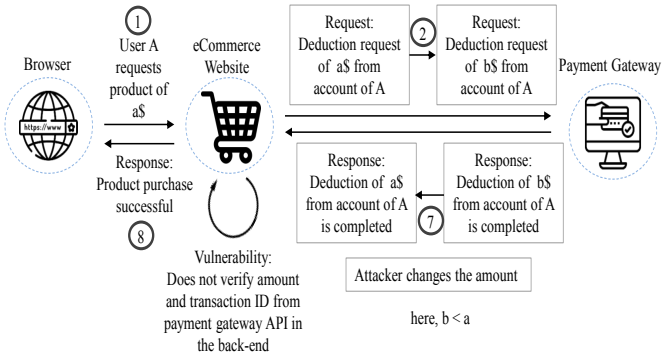

Fig. 2. Improper access control in e-commerce site.

attacker has to change the cookie instead of plain http payload which can be tricky in fixing the flaw. So, the developer has to focus and verify on each and every data that is coming from the customer's browser.

b) Attack scenario in payment solution provider: Payment solution providers (PSPs) are more likely to become the target of the attackers. As it is an interface between two components, flaw can be hidden in both of the sides.

1) At first attacker creates a payment transaction ID in the PSP using a comparatively lower amount $b and notes the ID.
2) The attacker creates another payment of amount $a where $b<$a.
3) Before sending the request to the bank API, attacker replaces the transaction ID of amount $a with the ID of amount $b.
4) Bank server deducts the amount $b and sends a success request to the PSP with the transaction ID of $b.
5) Attacker replaces the transaction ID with the previously noted transaction ID of $a.
6) If improper access control vulnerability exists, PSP will consider it as a valid transaction and return success status of the order of $a.

This attack procedure is depicted in figure 3. This problem

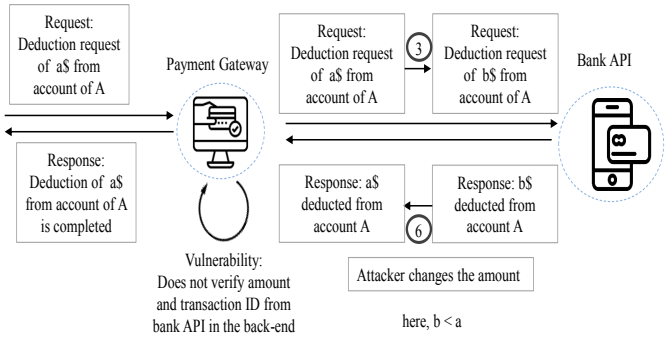

Fig. 3. Improper access control in payment solution provider.

occurs when the PSP does not verify both the received transaction ID and payment amount from the banking server in its back-end.

c) Attack scenario in banking server: Banking server deals with multiple financial tasks. The servers generally maintain a highly secure environment. As improper access control is a subtle bug, it can exist in the system. Normally, user ID, OTP, pin, http traffic are some important points to ensure security. But smaller bugs can also have greater impact on the system. Improper access control vulnerability is misused by an attacker using the below methods-

1) Attacker replaces the amount with a smaller amount.
2) Attacker creates a fake failed transaction or a valid transaction with smaller amount.
3) Attacker forces a request to redirect to the PSP (using previously saved requests) with the failed transaction ID.

The procedure is depicted in figure 4. If improper access control exists in any point of the banking system, PSP will consider any of the above points as valid request. The whole banking service for micro-payment will be affected by this type of attack.

B. Attack in PCI-DSS model

Best Practices for securing E-commerce of PCI-DSS security standard proposes a method to ensure the security in the e-payment system for e-commerce platforms. However, the most potential points of improper access control vulnerability in PCI-DSS framework is depicted in figure 5. The customer

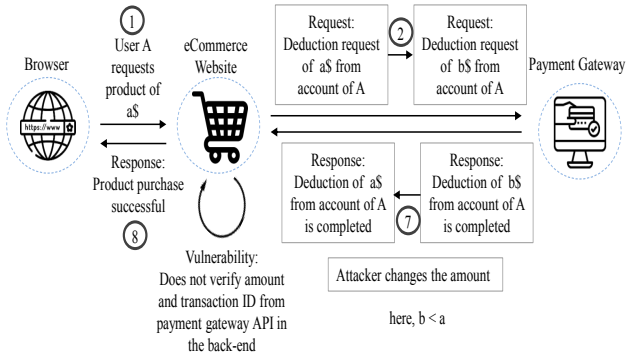

Fig. 4. Improper access control in bank API.

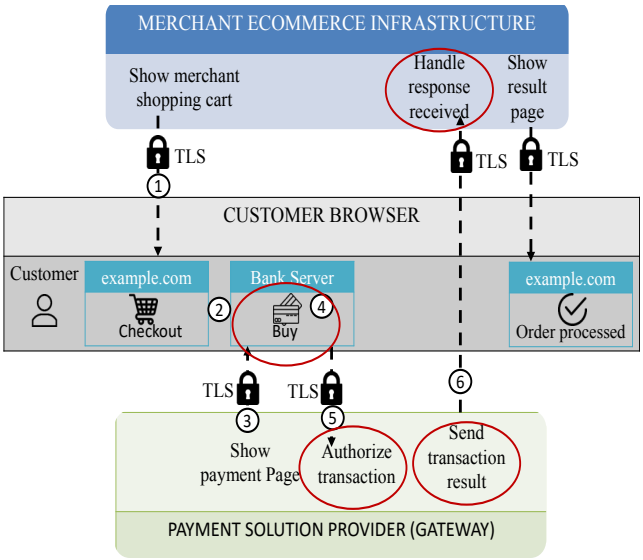

Fig. 5. Improper access control in PCI-DSS payment flow.

browser interacts with merchant infrastructure and the PSP. Red circled points in the figure are the target of the attackers to exploit improper access control vulnerability. The potential vulnerable locations are bank server, transaction authorization and sending transaction result in PSP and handling the received response in the merchant e-commerce infrastructure. The vulnerability can be raised in one or more locations.

V. DISCUSSION AND CONCLUSIONS

In this paper, detailed information about the security aspects of e-payment system is described. It is found that, attack on customers can be conducted by lower-skilled attackers, higher effort and with less financial loss compared to the other components. On the other hand, attack on PSP and banking server is trickier and carried out by expert attackers, which has a devastating effect on financial infrastructure. Improper access control is an underrated bug which can lead the system to a huge loss. Most of the attackers aim at the weak points of the components to carry out the attacks. Developers and analysts must not undervalue improper access control. In future, a detection and mitigation methodology of improper access control will be proposed. Developers must fix all of the weak points of the components that are stated in this paper to mitigate a higher risk of financial loss.

REFERENCES

[1] "The World Bank Payment systems worldwide — a snapshot. Summary Outcomes of the Fourth Global Payment System Survey September 2018 ," Available on: http://pubdocs.worldbank.org/en/591241545960780368/GPSS-4-Report-Final.pdf, [Online; Last accessed: 18 June 2020].

[2] D. L. Holloway and A. Anderson, "Online payment system for merchants," Dec. 10 2009, uS Patent App. 11/922,346.

[3] P. Lai, "Design and security impact on consumers' intention to use single platform e-payment," *Interdisciplinary Information Sciences*, vol. 22, no. 1, pp. 111–122, 2016.

[4] H. Alshehri and F. Meziane, "The impact of trusted and secured transactions in an e-commerce environment on consumers' behaviour: The case of saudis in the uk," *Journal of Internet Technology and Secured Transactions*, vol. 6, no. 3, p. 596–604, Sep 2018. [Online]. Available: http://dx.doi.org/10.20533/jitst.2046.3723.2018.0073

[5] T. OWASP, "Top 10-2017 the ten most critical web application security risks," *URL: owasp. org/images/7/72/OWASP_Top_10-2017_% 28en*, vol. 29, 2017.

[6] S. Agarwal, M. Khapra, B. Menezes, and N. Uchat, "Security issues in mobile payment systems," in *Proceedings of ICEG 2007: The 5th International Conference on E-Governance*, 2007, pp. 142–152.

[7] A. Alturki, N. Alshwihi, and A. Algarni, "Factors influencing players' susceptibility to social engineering in social gaming networks," *IEEE Access*, 2020.

[8] D. Airehrour, N. Vasudevan Nair, and S. Madanian, "Social engineering attacks and countermeasures in the new zealand banking system: Advancing a user-reflective mitigation model," *Information*, vol. 9, no. 5, p. 110, 2018.

[9] Y. Sawa, R. Bhakta, I. G. Harris, and C. Hadnagy, "Detection of social engineering attacks through natural language processing of conversations," in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*. IEEE, 2016, pp. 262–265.

[10] R. Heartfield, G. Loukas, and D. Gan, "An eye for deception: A case study in utilizing the human-as-a-security-sensor paradigm to detect zero-day semantic social engineering attacks," in *2017 IEEE 15th international conference on software engineering research, management and applications (SERA)*. IEEE, 2017, pp. 371–378.

[11] B. Bhushan, G. Sahoo, and A. K. Rai, "Man-in-the-middle attack in wireless and computer networking — a review," in *2017 3rd International Conference on Advances in Computing,Communication Automation (ICACCA) (Fall)*, 2017, pp. 1–6.

[12] M. Asaduzzaman, M. Majib, and M. Rahman, "Wi-fi frame classification and feature selection analysis in detecting evil twin attack," in *IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 1704–1707.

[13] Y. Raivio and S. Luukkainen, "Digital rights management in the mobile environment." in *ICE-B*, 2006, pp. 182–185.

[14] D. Appelt, C. D. Nguyen, L. C. Briand, and N. Alshahwan, "Automated testing for sql injection vulnerabilities: an input mutation approach," in *Proceedings of the 2014 International Symposium on Software Testing and Analysis*, 2014, pp. 259–269.

[15] M. Asaduzzaman, P. P. Rawshan, N. N. Liya, M. N. Islam, and N. K. Dutta, "A vulnerability detection framework for cms using port scanning technique," in *2nd International Conference on Cyber Security and Computer Science (ICONCS 2020)*. Springer, 2020.

[16] A. Olmsted, "Securing e-loyalty currencie," *Journal of Internet Technology and Secured Transaction*, vol. 5, no. 1, Mar 2016. [Online]. Available: http://dx.doi.org/10.20533/jitst.2046.3723.2016.0057

[17] F. Sun, L. Xu, and Z. Su, "Static detection of access control vulnerabilities in web applications." in *USENIX Security Symposium*, vol. 64, 2011.

[18] T. Ryutov, C. Neuman, K. Dongho, and Z. Li, "Integrated access control and intrusion detection for web servers," *IEEE transactions on parallel and distributed systems*, vol. 14, no. 9, pp. 841–850, 2003.

[19] B. P. for Securing E-commerce Special Interest Group and P. S. S. Council, "PCI-DSS Information Supplement: Best Practices for Securing E-commerce ," Available on: https://www.pcisecuritystandards.org/pdfs/best_practices_securing_ecommerce.pdf, [Online; Last accessed: 21 June 2020].

[20] M. Wang, G. Liu, C. Yan, and C. Jiang, "Modeling and vulnerable points analysis for e-commerce transaction system with a known attack," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 2016, pp. 422–436.

[21] M. Wang, Z. Ding, P. Zhao, W. Yu, and C. Jiang, "A dynamic data slice approach to the vulnerability analysis of e-commerce systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[22] F. Nabi, J. Yong, and X. Tao, "Proposing a secure component-based-application logic and system's integration testing approach," *International Journal for Information Security Research*, vol. 9, no. 1, p. 839–847, Mar 2019. [Online]. Available: http://dx.doi.org/10.20533/ijisr.2042.4639.2019.0096

[23] K. Krombholz, H. Hobel, M. Huber, and E. Weippl, "Advanced social engineering attacks," *Journal of Information Security and applications*, vol. 22, pp. 113–122, 2015.

[24] G. Cybenko, A. Giani, and P. Thompson, "Cognitive hacking: A battle for the mind," *Computer*, vol. 35, no. 8, pp. 50–56, 2002.

[25] A. Luvanda, S. Kimani, and M. Kimwele, "Identifying threats associated with man-in-the middle attacks during communications between a mobile device and the back end server in mobile banking applications," *IOSR Journal of Computer Engineering (IOSR-JCI)*, vol. 12, no. 2, pp. 35–42, 2014.

[26] J. A. Ojeniyi and S. M. Abdulhamid, "Security risk analysis in online banking transactions: Using diamond bank as a case study," *International Journal of Education and Management Engineering*, vol. 9, no. 2, pp. 1–14, 2019.

[27] M.-K. Base, "Magento for developers: Part 7–advanced orm–entity attribute value–ecommerce software for growth," *Magentocommerce. com (2009-12-06). Retrieved on*, pp. 07–07, 2012.

[28] C. Newport, *Shopify: The Ultimate Shopify User Guide, Simplifying Shopify and Helping You to Make Money with Your Own Shopify Ecommerce Store!* North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2017.

[29] R. Rogers, *Nessus network auditing*. Elsevier, 2011.

[30] S. Kals, E. Kirda, C. Kruegel, and N. Jovanovic, "Secubat: a web vulnerability scanner," in *Proceedings of the 15th international conference on World Wide Web*, 2006, pp. 247–256.

[31] G. F. Lyon, *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure, 2009.

[32] "Financial fraud in the digital space, November 2018 ," Available on: https://www.enisa.europa.eu/publications/enisa-position-papers-and-opinions/financial-fraud-in-the-digital-space, [Online; Last accessed: 29 June 2020].

[33] A. Mendoza and G. Gu, "Mobile application web api reconnaissance: Web-to-mobile inconsistencies & vulnerabilities," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 756–769.