Vol. 9(1), pp. 1-11, July-December, 2023
DOI: 10.5897/IJCER2023.0304
Article Number: 0B513B771288
ISSN: 2141-6494
Copyright ©2023
Author(s) retain the copyright of this article
http://www.academicjournals.org/IJCER

ACADEMIC JOURNALS
expand your knowledge

International Journal of Computer Engineering Research

Review

Security threats affecting user-data on transit in mobile banking applications: A review

Daniel Okari Orucho[1]*, Fredrick Mzee Awuor[1], Cyprian Ratemo[1] and Collins Oduor[2]

[1]School of Information Science and Technology, Kisii University, Kenya.
[2]School of Science and Technology, United States International University, Africa, Kenya.

Received 28 January, 2023; Accepted 29 August, 2023

Mobile banking technology is beneficial to customers because it is convenient for conducting bank services remotely. Newer channels used for mobile banking such as mobile banking applications have been developed as an innovation that is utilized to offer traditional banking services remotely such as money withdrawals, deposits, cheque book requests among others. This saves time and costs of going to physical banking halls and having to wait in long queues. However, utilization of mobile banking applications to access remote banking services is not immune from malicious attackers who steal users' data such as login credentials and use them to access the users' accounts and steal money. Securing user data on transit is important to mobile banking application users. It is also important to protect user data from online profile stealing which often leads to losing money from customers due to cyber thieves. This paper used descriptive research approach and desktop research and reviewed emerging security threats in mobile banking applications. It also identified and analyzed the cutting-edge mechanisms available to mitigate security threats, and lists some of the pertinent open research issues.

Key words: Mobile banking applications, security threats.

INTRODUCTION

Using portable devices like smart phones to execute financial exchanges related to a client's bank account is known as mobile banking. These services include, among others, checking your balance, paying bills, sending money, and requesting cheques (Raharja and Tresna, 2019). Bookings, loan repayments, and airtime top-up are some additional services that mobile banking provides (Bagudu et al., 2017).

Mobile banking programs, often known as applications, are created by banks to make it simpler for customers to access their accounts and call the bank server to access services remotely. Consequently, the usage of mobile banking applications aids banks in improving after-sales services, resource management that is effective, improved business aggression and gift packages through the sale of goods and services like client credit and platinum cards, among others (Sang, 2021). Consequently, emergence of mobile banking applications has energized mobile banking market with better banking service delivery with the help of wireless networks (Tran and Corner, 2016). A bank customer must register for mobile banking with the relevant bank in order to access

*Corresponding author. E-mail: danielokari@yahoo.com. Tel: +254717541878.

mobile banking services remotely using mobile banking applications.

Customers download and install mobile banking applications on their smart phones after purchasing them from application stores. A consumer must sign up for mobile banking using a smart phone after installing the client application. A customer can now use the client installed on the smart phone to access mobile banking services following successful registration. A secure connection to a bank's server is made possible by procedures like username and password, as well as biometrics such as iris, thumb, and face recognition among others at login, and requests for financial services. The applications frequently demand a One-Time Password (OTP) as an additional authentication factor for important actions like money transfers (Abuhamad et al., 2020). Users can utilize mobile banking applications to access bank services under the condition that the bank server can validate the login information they supply. It is crucial to protect and ensure users' safety and security because these applications handle sensitive personal information about them.

Using mobile banking applications, customers are able to manage their bank accounts remotely in the easiest way possible. In order for customers to continue using this technology, some of the fundamental consideration that is taken into account by banks when designing mobile banking applications is security. This is because financial institutions must protect customer’s data and money from unauthorized access (Khomich, 2022). Nevertheless, other applications such as those used in social network communications are not designed with security as a priority which compromises users privacy and security while online (Ford et al., 2022). Moreover, mobile banking applications’ security depends on a secure software development lifecycle as compared to mobile applications that involve deep link handling and thus have a larger number of vulnerabilities (Positive Technologies, 2020).

Nevertheless, mobile banking applications are susceptible to security threats such as mobile malware, unauthorized access, packet sniffing attacks (Bhattacharya and Reddy, 2022), Man-In-The-Middle (MITM) attacks (Javeed et al., 2020), Domain Name System (DNS) poisoning (Brandt et al., 2018), Secure Sockets Layer (SSL) strip session hijacking (Hossain et al., 2014), eavesdropping attack (Talom and Tageh, 2019), Denial of Service (DoS) attacks (Kaka et al., 2017), and social engineering attacks. This form of attacks can be launched on mobile banking applications with the intention to seize confidential data of customers and use the information to traverse and gain access into the bank of the customer to siphon money.

Due to a lack of robust security implementation, 85% of Android and 70% of iOS mobile banking applications are susceptible to different threats and vulnerabilities that could allow unauthorized management of user accounts as well as theft of user funds and financial information. Even if mobile banking applications give users the convenience of accessing financial services from anywhere, round-the-clock, cybercriminals access private financial data by using a variety of attack vectors to take advantage of weaknesses present in mobile banking applications. Drive-by downloads, malware, and viruses (such as Trojan horses, worms, viruses, botnets, ransom ware and spyware) are examples of malicious software that can be utilized by adversaries to gain access into a user’s account. MITM, eavesdropping, and social engineering are just a few of the techniques hackers utilize to gain access to user’s bank accounts to commit cyber-crimes (Nerwal et al., 2019).

Banks have suffered significant financial losses as a result of the inherent risks and weaknesses in mobile banking. Cybercriminals steal nearly USD 114 through mobile banking each year all across the world. Similarly, to combat cyber attacks, the banking sector invests $274 billion globally (Acharya and Joshi, 2020). A considerable loss of confidentiality, integrity, and availability is also brought on by attacks on user data on transit in mobile banking applications. However, some of the mitigation strategies employed to annihilate the current security threats in mobile banking applications include information hiding, secret writing, use of hybrid algorithms, authentication, utilization of mobile antivirus programs, and user education about social engineering attacks.

This paper is organized as follows: section one deals with introduction, section two explains about problem definition, section three discusses about methodology, section four is an overview of security of mobile banking applications, section five explores threats affecting user-data on transit in mobile banking applications, section six discusses other authors related work and findings and section seven is discussion and future research.

Problem definition

Mobile banking through utilization of mobile banking applications is a technological strategic resource for bank as well as customers which has increased customer satisfaction and decreased production costs. Majority of banking services are now provided to customers via their mobile devices rather than the traditional banking hall. Apparently, mobile banking applications utilize client administration attributes such as username and passwords, personal identification numbers, and user authentication to be validated to access mobile banking services from the server. Nevertheless, cybercriminals take advantage of vulnerabilities in mobile banking applications and change their behaviours and extract client administration details, manipulate this information to traverse customers’ bank accounts remotely.

The goal of cybercriminals is to access customers’ bank accounts and steal money by breaking into the mobile banking system using client administration details. The major root of the problem is that wireless network channels used by mobile banking applications to transfer critical and private client data are not immune from cyber attacks. For example in 2018, the financial conduct authority penalized Britain's Tesco bank $18.4 million for the hacking incident that utilized mobile banking applications reverse engineering to steal clients' login credentials and communication protocols that led to stealing of $2.26 million from over 9,000 accounts over the course of one night (Digital Transformation Cyber Security News, 2016).

Although there are several techniques available to prevent clients' confidential information from being collected, they lack the strength to combat emerging cybercrime methods used to unlawfully access consumers' bank accounts using wireless networks. For example, 90% of mobile banking applications tested are found to have weak cryptography (EMBASB, 2022). Consequently, it is paramount to fortify current methods employed to secure access to mobile banking services. Notably, the dependability and tamper-proof connection between mobile banking applications placed on customers' mobile devices and the bank systems is crucial to the security of these applications since it prevents fraudsters from accessing customers' bank accounts. The user's mobile client application and the bank server must utilize secure connection in order for clients to successfully access banking services and remotely access their bank accounts without being concerned about hackers. To this end, this study sought to answer the following questions;

i) What are some of the security threats that affect user-data in on transit mobile banking applications?
ii) Which techniques are used to secure user-data on transit in mobile banking applications?
iii) What are some of the open research issues in securing user-data on transit in mobile banking applications?

METHODOLOGY

This study took advantage of descriptive research design. Descriptive research is a technique used to accurately characterize the phenomena that already exists. Using this method, the researcher was able to assess security threats that affect user-data on transit in mobile banking applications and present open research issues in securing mobile banking. The study collected secondary data from multiple search engines and databases using desktop research. Data analysis was done using content analysis and gap analysis. Content analysis was used to assess security threats in mobile banking applications, techniques used to secure user-data on transit in mobile banking applications, and open research issues in mobile banking. Gap analysis was used to review existing data on mobile banking and identify areas of focus for future research. Thus data were analyzed critically and presented in thematic areas.

Security of mobile banking applications

Without going through a different browser or third-party software, mobile banking applications offer a direct connection from the mobile device application to the bank server. The security and connectivity of consumer contacts are presumably under much better control for banks in light of this. Because the mobile banking applications were created especially for a certain bank and its clients, the bank is able to offer a secure connection that complies with their specific requirements using Secure Socket Layer (SSL), encryption, and two-factor authentication. Therefore, data must be transmitted securely, access to programs should be limited, data honesty should be protected, and effects of device loss must kept minimum. These are the main security issues that mobile banking addresses.

Some of the cyber-attacks that could harm mobile banking technology include Personal Identification Number (PIN) recovery attempts, unauthorized usage, hacking, eavesdropping; and malware. To prevent unauthorized individuals from accessing mobile devices, passwords and authentication methods are utilized. Before accessing the mobile device, the user must enter the right password.

Physical devices, OTPs, transaction profile scripts, and other biometric authentication methods are additional authentication methods utilized in mobile banking.

THREATS AFFECTING USER-DATA ON TRANSIT IN MOBILE BANKING APPLICATIONS

The epoch and sporadic technological advancement in mobile banking has presented both advantages and disadvantages for banks as well as bank clients. However, security of mobile banking still remains a key concern for banking institutions as well as clients. This is because mobile banking systems are susceptible to attacks mostly due to the large number of access channels they provide. The mobile banking threat landscape is proportionately ever-evolving with the development of mobile banking technology. Threats in mobile banking are divided into three groups: threats of a mobile device, on the network, and at the data center (Yildirim and Varol, 2019). The following parts of the study discuss threats and surface attacks on mobile banking applications.

Mobile malware

Malware attacks utilize security flaws which can enter computer systems without authorization. The malware crucial information, packets can be transferred from one attacks are motivated significantly for financial or political gains, which encourages attackers to get into as many network devices as they can in order to accomplish their harmful goals (Stevens, 2020). Based on its damaging aims and activities, mobile malware can theoretically be divided into a variety of types. A separate benchmark and extra criteria is the distribution strategy. The two main distribution techniques frequently utilized are social engineering and self-propagation. The first strategy automatically installs malware onto mobile devices via a variety of methods, such as worms, while the second technique uses users' curiosities and unsafe understanding to lure them to actively install applications (such as adware).

There are several non-mainstream malware varieties in addition to these mainstream groups. The most prevalent mobile malware kinds are Trojans, backdoors, spyware, and adware. Banking Trojans, a special form of Trojan type focused at mobile banking activities and services are in first place in the Trojan family. It can collect private user information and delicate user passwords when a network connection is available, store them in a secret location, and upload them to a command and control server (Imagine, 2023). The transition of banking from personal computers to mobile devices coincided with the rise of banking Trojan horses.

Some techniques that can be used to annihilate mobile malware in mobile banking applications include updating application regularly, use of mobile security software, use of firewall, use of mobile phone screen lock protection, and downloading applications from official stores. Updating mobile banking applications and operating system regularly entails running the most recent version of applications on the smartphone. This will guarantee most advanced security patches and updates are installed. Secondly utilization of mobile security software including antivirus for smartphones protects mobile banking applications against viruses and malware. Additionally, downloading mobile banking applications from official stores such as Apple store or Google store assures that they have undergone safety checks. Therefore, it is always better to download verified applications from approved and official stores. (Imagine IT, 2023).

Packet sniffing attacks

The practice of packet sniffing involves keeping track of packet information as it moves through a network. It is a tool that keeps track of all network activity. Additionally, it has the capacity to eavesdrop on and record any network traffic both inbound and outbound. The method through which data is transmitted over a network is known as packets. By dividing a packet into smaller pieces, each of which has a destination and source address and other computer to another on a network. However, if a packet sniffer is installed at any of the network's nodes, it is feasible to assess a network's performance or locate a bottleneck (either the source or the destination). Network administrators are primarily responsible for using sniffer packets (Bhattacharya and Reddy, 2022).

Sniffing can be categorized into various types such as Media Access Control (MAC) flooding, Domain Name System (DNS) poisoning, Address Resolution Protocol (ARP) poisoning, Dynamic Host Configuration Protocol (DHCP) attacks and password sniffing among others (Anu and Vimala, 2017). Sniffer packets provide a human readable representation of binary network data, such as a clear username and password for a voice over internet protocol connection, as well as assistance in debugging, monitoring, and spotting any network vulnerabilities. These could be illegal usage if you do not have permission to use a packet sniffer on a particular network within your company. Packet sniffers can also be used interchangeably with protocol and network analyzers. Active and passive packet sniffers are also available. Passive packet sniffers only collect data and cannot be observed; they do not send or receive any responses. Passive sniffers, for instance, are useful in telecommunications, radar systems, and medical equipment. Wireshark and Colasoft Capsa are examples of passive packet sniffers. Data can be sent by active packet sniffers (Bhattacharya and Reddy, 2022).

In order to detect sniffers, systems should be checked for promiscuous mode. Promiscuous mode is a mode of the Network Interface Card (NIC) that allows all the packets without validating its destination address. Stand alone sniffers are difficult to detect as they do not send packets. The reverse DNS lookup can be used to detect non-stand-alone sniffers. An Intrusion Detection System (IDS) is a security mechanism that helps to detect sniffing activities. Using Ping method: a Ping request has to be sent to the suspect machine with incorrect MAC address. Only the sniffer will respond to this request. Using ARP method: ARP should be added to all the nodes. The node which runs in promiscuous mode will cache the MAC address. The Ping message should be sent with different MAC. Only the sniffer will be able to respond. Using promiscuous detection tool such as Promqryui helps to detect the node which is in promiscuous mode (Anu and Vimala, 2017).

Defense against sniffing can utilize the following countermeasures: use of encryption to protect confidential information, use static Internet Protocol (IP) addresses and static ARP tables to prevent attackers from adding spoofed ARP entries, restrict the network to authorized users only, use Internet Protocol version 6 (IPv6) instead of IPv4, use encrypted session such as Secure Shell (SSH) instead of telnet, use Hypertext Transfer Protocol Secure (HTTPS) instead of Hypertext Transfer Protocol (HTTP) to protect usernames and passwords, use switches is better than hubs as switches deliver data only to the recipient, password authenticate the shared folders and services, and encrypt the communication between mobile device access point to prevent address spoofing (Anu and Vimala, 2017).

Man-in-the-middle attacks

A Man-in-the-Middle (MITM) assault occurs when a hacker stands between the information receiver and the sender of information (Javeed et al., 2020). A MITM attack can be passive, where a hacker merely identifies the client-server traffic and monitors it with the potential to make use of the knowledge later, or active, in which a hacker can impede or alter the information being transmitted (Kieseberg et al., 2015).

The integrity and confidentiality linking the mobile banking application and the bank's main server in terms of communication are both violated by active MITM attacks (Nerwal et al., 2019). There are various techniques to conduct active MITM attacks, such as secure socket layer hijacking, domain name system spoofing, and poisoning of the Address Resolution Protocol (ARP) cache. ARP cache poisoning involves inserting a fake IP or media access control address pair into a host's ARP cache. Erroneous MAC address may be returned. ARP requests are sent by the host against a legitimate IP address, such as gateways IP address. The requesting host receives a response in return following a gateway ARP request, which is frequently advertised across the Local Area Network (LAN). Therefore, by turning on its adapter in promiscuous mode, an attacker connected to the LAN can hear this inquiry and respond. Finally, the attacker can force traffic to pass through him or her instead of a legitimate gateway by using ARP cache update policy to set their own MAC address against gateway's IP address (Kaka et al., 2017).

By deploying contemporary network switches with security features like dynamic ARP inspection and port security, it is possible to mitigate many network threats. The amount of MAC addresses which may display an outlet port is limited by port security. Thus, a port is automatically stopped if the number of MAC addresses observed rises above a set threshold, preventing MAC spoofing. By interrogating the valid IP-MAC bindings database in the DHCP snooping protocol, dynamic ARP inspection verifies an ARP message's authenticity to avoid ARP spoofing (Tripathi and Hubballi, 2021)

However, if the malicious client's fake ARP replies does not get through the switch, dynamic ARP is unable to prevent ARP spoofing and the resulting DHCP starvation (Hubballi and Tripathi, 2017).

Defense measures against Domain Name System (DNS) attacks can be roughly categorized into three categories: mapping DNS queries and answers, adding DNS, and altering DNS cache by default. Matching the DNS queries that a server receives to the responses that it sends is the simplest approach to identify a DNS flood attack in the first scenario. The server is thought to be under a flood attack if the quantity of DNS requests without an equivalent answer goes beyond a predetermined acceptable level, and firewall regulations can then be changed to halt the harmful traffic (Hubballi and Tripathi, 2017).

DNS flood attacks can be prevented by using a firewall that recognizes identity spoofing. This technique specifically calls for a DNS server for delivery of a unique cookie to each client, who then equates every appeal they send to the backend with the earlier sent tracker. This tracker can be used to check whether the IP listed in the packet actually sent a DNS. T-DNS is an additional DNS attack defense method that protects DNS communication using the Transport Layer Security (TLS) protocol (Zhu et al., 2015).

The final defense tactic is to increase the DNS infrastructure entries' time to live values, for navigating the DNS echelons. Because of this, the resolver's cache holds the framework documentation for a longer amount of time and can be used to answer queries every time the DNS server is under assault (Zhu et al., 2015). Some assaults are made possible as a result of MITM attacks by taking advantage of the communication exchanged between clients and servers.

Eavesdropping attacks

Attacks by eavesdrop take place through intercepting network traffic. Without the consent of the trusted parties, the attacker listens in on information being transmitted across the network between the bank server and the mobile banking application. Insecure network communications are used by the attacker to access sensitive data. This frequently occurs when plaintext data is transferred via a communication channel (Talom and Tengeh, 2019).

In order to engage in active eavesdropping, a hacker must grab information by pretending to be a helpful unit and by asking questions to transmitters, as opposed to passive eavesdropping, in which a hacker passively discovers the information by listening to the message transmission in the network. This is referred to as tampering, scanning, or probing. Attackers leverage the communication network's frail Secure Shell (SSH) layer administration and frail encryption keys that are being used to their advantage. The communication's secrecy is violated through eavesdropping.

Data in transit is intercepted by attackers using network sniffer software, formerly known as Ethereal (Mtaho, 2015). Ethereal, a simple network traffic sniffer displays all wired and wireless network traffic. This protocol analyzer supports multiple platforms and multiple protocols. In addition to 802.11 and Bluetooth support, it has decryption support for a number of well-known wireless security protocols (Ndatinya and Xiao, 2015). Acquired information is presented by Wireshark program in an easy-to-read and logical format. Additionally, Wireshirk program contains a lot of built-in filters and lets users create their own filters. You can use these filters to exclusively collect particular information, such as Internet protocol addresses and port numbers. Information transmitted in clear text works well for sniffer programs. An encrypted key cracker is required for encrypted data. Encryption standards such as 802.11 and Wireless Application Protocol version 2 (WAP2) are the most recent wireless encryption protocols that have not yet been cracked. WPA and Wired Equivalent Privacy (WEP) are two older encryption standards with many tools such as AirSnort for WEP and AirCrack that can crack their encryption keys (Jain and Anubha, 2021).

Due to the lack of network disturbances or modifications, passive eavesdropping is challenging to identify and prevent. Nevertheless, data is frequently intercepted once the network changes are observed, but active attacks are easier to spot. Utilizing security technologies such as network segmentation, network monitoring, authentication, and understanding recommended security methods like firewalls, virtual private networks, and antimalware software, active eavesdropping attacks can be avoided (Salim et al., 2020; Okpara and Bekaroo, 2017).

Denial of service attacks

By routing traffic through a personal access point held by the attacker from a customer's mobile banking application to the server, Denial of Service (DoS) attacks are launched. Thus, the adversary drops each and every packet component of a flow between the bank's server and the system application. As a result, the mobile banking application encounters connections that have timed out, which eventually exhaust system assets (Kaka et al., 2017).

DoS assaults affecting application layer are most frequent kind of flaws in network protocols that stop network devices from operating properly. Both instances restrict or bar intended users from using network resources or services. The fact that Internet activity is a mixture of typical or legal traffic and abnormal activity makes it extremely difficult to identify these attacks. Additionally, attack traffic typically appears to be regular traffic (Kaka et al., 2017). Protocol-specific DoS attacks and generic DoS attacks are two primary application layer DoS threats. Network Time Protocol, timeshifting DoS attacks, Slow hypertext transfer protocol DoS attacks, and DHCP hunger assaults are a few examples of protocol-specific assaults (Tripathi and Hubballi, 2021).

Social engineering attacks

The largest dangers to banking organizations' cyber security framework are social engineering assaults.

Social engineering is preventable but may not be identified easily. Despite their differences, social engineering attacks all have a similar pattern. Four steps make up the typical pattern: researching the target, getting to know the target, applying what you've learned to carry out the attack, and leaving no evidence (Mouton et al., 2016). In order to circumvent a bank's security restrictions, enemies therefore do study on human behavior rather than using technological system assault methods (Nerwal et al., 2019).

Social engineering assaults frequently use phishing, spear phishing, scareware, pretexting, and baiting (Nerwal et al., 2019). Phishing attacks' main objective is to deceitfully obtain the targets' private banking details over the phone or through emails. To get private and sensitive information, attackers trick bank clients by utilizing websites, emails, adverts, scareware, anti-virus software, PayPal websites, prizes, and freebies to deceive bank clients.

Snapping on e-mail links or receiving calls from fictional bank department asking for personal information, for instance is a social engineering behavior attack. These efforts are geared towards mining important confidential bank details that could be used to access private accounts by utilizing mobile banking channel (Salahdine and Kaabouch, 2019).

Road apples are phishing attacks that trick users into clicking a link in order to claim a free gift. Like Trojan horses, they use unsecured computer resources like storage media or Universal Serial Bus (USB) drives that are stashed away at coffee shops and contain malware to launch assaults. The USB drive attacks the victims' computers when they enter it into them, acting like a Trojan horse in real life. The evil deeds that are being done in the background are not known to the victims of this attack (Costantino et al., 2018).

In order to get a victim's personal information, pretexting assaults include fabricating false yet believable circumstances. They are based on justifications that make the victim and the attacker both believe on each other. Attackers carry out their attacks using physical media, emails, or phone calls. Attackers provide information to carry out their attack on public web pages, at conferences where experts in the same field congregate, or in phone directories. A pretext could be anything from a job or service offer to a request for personal information to helping a friend get access to something to winning the lottery (Ghafir, 2016).

These attacks present significant risks and as such businesses organizations should strategize the best methods to control the assaults. Banking organizations should institute a philosophy that teaches bank customers as well as staff on the different forms of propagation and how they can be controlled. These can be inform of planned retreats for staff or by way of advertising to customers using videos as well as emails sent to customers.

Cross-site scripting attacks

Cross-Site Scripting (XSS) is a programming error which occurs when user inputs un-sanitized data into a system (Kirsten, 2016). The attacker takes advantage of flaws to inject unfiltered scripting code into the online application, leading to account takeover, session stealing, and rerouting traffic to the attacker's domain (Agrawal and Wang, 2018; Jiang et al., 2020). Any website that is vulnerable and is developed in any programming language is susceptible to XSS attack. The XSS vulnerability runs scripts in user browsers (Chen et al., 2021). A dynamically generated script becomes dangerous if a user tampers with it or changes it. The various types of XSS attacks include mutation-based XSS attacks, document object model-based assaults among others.

Adversaries run JavaScript code in the victim's browser in this type of attack in order to steal sensitive information from the victim. It is a vulnerability that is frequently found in modern web pages. Malware attacks frequently aim to compromise the online privacy of computers, mobile devices, and Internet of Things (IoT) nodes. Additionally, because the victims think the communication connection is safe, they are ignorant of the intruder. To find malware that targets Windows operating systems, a number of scanning techniques based on certain signatures are available. Static and dynamic methodologies are used to separate the malware identification analysis (Hasham et al., 2019).

The Dynamic Approach uses code execution in a real-time virtual environment to find malware trends. Malicious behavior can be discovered through function calls, function parameter research, data flow, instruction traces, and visual code analysis. Online automated tools are available for use in analyzing the dynamic behavior of hazardous code.

Specifically, TT analyzer, Anubis, and CW Sandbox are utilized. Since every dynamic behavior of the source code is monitored, this method requires more time. Approaches to static malware analysis do not require real-time source code execution (Camillo, 2017).

Control flow graph, opcode frequency, n-gram, and string signature are some of the static malware identification methods. The executable is first discovered using the disassembly programs IDA Pro binary code analysis and OllyDbg before static-based techniques are applied. This disassembles take binary executables and extracts hidden patterns from them. The encoded text is then extracted from the executable using these patterns. A static-based analytical technique called the byte sequence methodology takes out n-byte sequences from these patterns. A static technique for removing code structure analysis is a functional call graph. The majority of the common fixes that can be used to drastically lessen the effects of XSS attacks are highlighted in this section (Sahoo and Gupta, 2019; Kaur et al., 2018). It places a strong emphasis on outlining the XSS mitigation guidelines that programmers can use to stop XSS attacks. Filtering, Hypertext Markup Language (HTML) entity encoding, sanitization, content security policy, data validation, and other sorts of escaping, such as attribute value escaping, javascript escaping, and Universal Resource Locator (URL) escaping, are some of the several types of protection measures for XSS assaults (Gupta and Gupta, 2016; Gupta and Gupta, 2018; Gupta et al., 2017; Xu et al., 2020; Sahin et al., 2022).

Sql injection attacks

Structured Query Language (SQL) is a programming language used to manage relational databases and is widely used in web applications. It is made up of declarative elements such as queries, expressions, clauses, statements among others (Nagpal et al., 2017). Structured Query Language Injection Attack (SQLIA) is a type of code injection attack that is carried out by adding SQL code into the user’s input to gain access to unauthorized resources. It may occur when the query is built by concatenating the user’s input, such as data entered into an interactive web form, with unintended data, including URL data, data obtained from cookies among others without proper validation.

A successful SQLIA can have severe repercussions such as data loss and system compromise. The attacker has the ability to manipulate database data as well as access private information like login passwords and financial information. In some circumstances, the attacker might even succeed in seizing total authority over compromised system. Given the severe repercussions of a successful SQLIA, it is crucial for banks to put in place efficient prevention and detection techniques to safeguard their web applications (Crespo-Martínez et al., 2023).

Banks can suffer serious harm from SQLIA and therefore it is critical to have methods in place to detect and prevent them. The most popular detection methods include log analysis, intrusion detection systems and honeypots. Log analysis involves reviewing log files from the web server and database server to find potential security threats and identify SQLIA. Both manual and automated methods are available for performing log analysis. Manual log analysis entails going line by line through log files to look for signs of SQLIA. On the other hand, automated tools can analyze log files in real-time and send alerts when they find proof of an attack. Log analysis has the benefit of providing a thorough record of all requests made to the web application, which makes it simpler to identify and address security threats (Hadabi et al., 2022). However, log analysis takes time and challenging to complete particularly in big log files.

Another method for detecting SQLIA is by using software programs such as Intrusion Detection Systems (IDS). These software analyzes network activity in real-time and can be installed on the database server or web server. They employ a range of methods, such as signature detection, behavior-based detection, and anomaly-based detection, to find security risks. Network traffic is compared to a database of known security risks in signature-based detection. Monitoring web applications and the database's behavior to search for signs of an attack is known as behavior-based detection while anomaly-based detection entails tracking the web applications and the database's behavior and contrasting it with a typical baseline to look for signs of an attack. IDS have the benefit of being very good at spotting security threats in real-time, enabling banks and other organizations to react rapidly to potential threats (Manhas, 2022; Alajanbi et al., 2021). However, IDS have the drawback of producing a lot of false positives, making it challenging to distinguish between genuine and bogus alerts.

The third method that can be utilized to detect SQLIA is a honeypot which is a type of decoy system used to attract and trap attackers. Honeypots can be used to detect SQLIA by providing a simulated web application that is vulnerable to attack. It will recognize an attempt to inject malicious code into the simulated web service and produce proof of the attempt. Honeypots are good in spotting SQLIA because they offer a controlled environment where the attacker is free to try and insert malicious code into the SQL query. The benefit of honeypots is that they are less likely to produce false positives than other detection methods and can be very effective at identifying actual threats (Chou and Jiang, 2021). The drawback of honeypots is that they are costly to setup and maintain.

There are several techniques that guarantee the correct validation of user-supplied data and secure processing of SQL statements and prevent SQLIA. The most popular prevention strategies are input validation, parameterized queries, and stored procedures. A critical stage in preventing SQLIA is input validation. Before being handled by the application, user-supplied data must be validated to make sure it complies with certain requirements. Input validations' primary goal is to stop harmful code from running as part of SQL query to the database. Input validation should be done in the client-side using JavaScript or other client-side scripting languages as well as at the server-side using a server-side programming language like Hypertext Preprocessor (PHP) or Active Server Pages (ASP.NET). Without proper input validation, bank's web applications are at a high risk of SQLIA (Kumar, 2023).

Parameterized queries are a secure way to execute SQL statements. With this technique, the chance of SQLIA is completely eliminated because malicious code cannot be injected into the query by the attacker. Rather than being a component of the SQL statement, the parameters are passed to the query as distinct values, making it impossible for an attacker to insert malicious code into the query. Thus utilization of parameterized queries is the safest way to safeguard SQLIA (Baklizi et al., 2023).

SQLIA can also be prevented using stored procedures which are pre-compiled SQL statements that are saved in a database. These procedures offer an additional layer of protection against SQLIA and can be used to carry out complex database tasks. Injecting harmful code into the SQL query is more challenging because stored procedures run on the database server rather than the web server. When processing large amount of data or performing complicated database operations like aggregating data, stored procedures are especially helpful (Yadav and Shekokar, 2023). This is because stored procedures are efficient than dynamic SQL statements as they are pre-compiled and optimized for performance.

OTHER AUTHORS RELATED WORK AND FINDINGS

In order to steal confidential information from communicating devices over wireless networks, different attacks can be launched using the following methods: MAC flooding, DHCP attacks, and ARP spoofing, DNS poisoning. These form of attacks can be prevented using a variety of techniques such as use of encryption to protect confidential information, use static Internet Protocol (IP) addresses and static ARP tables to prevent attackers from adding spoofed ARP entries, restrict the network to authorized users only, use of IPv6 instead of IPv4, use encrypted session such as SSH instead of telnet, use of HTTPS instead of HTTP to protect usernames and passwords, use switches is better than hubs as switches deliver data only to the recipient, password authenticate the shared folders and services, and encrypt the communication between mobile device access point to prevent address spoofing (Anu and Vimala, 2017).

A study by Positive technologies (2020) reveals that 100% of mobile banking client-side applications contain vulnerabilities in their code. For example code is not obfuscated, protection against code injection and repackaging is absent, and code contains names of classes and methods. This demonstrates that insufficient code protection leaves banks vulnerable to source code analysis. Lack of obfuscation allows attackers to analyze the code and find important data such as testing related usernames and passwords, encryption keys and parameters from which keys can be derived and salts for hashing and encryption.

Furthermore, mobile banking applications should incorporate and leverage biometric solutions like fingerprint scanning designed for mobile devices. Banks should advise customers not to click on phishing links and not to tell anyone, even bank employees, about their one-time passwords. The study also found that users should avoid using mobile banking apps on jailbroken iOS devices and protected wireless networks, as well as be aware of social engineering assaults (Yildirim and Varol, 2019). A study on security threats on threat mode for vulnerability assessment and penetration testing for Android and iOS mobile banking applications established that these applications are not verified and tested exhaustively for security vulnerabilities. Moreover, banks developing mobile banking applications are not conducting vulnerability assessment regularly. Therefore, mobile banking applications are susceptible to security threats such as insecure data storage, malware in the applications, weak cryptography, MITM attacks among others (Bojjagani and Sastry, 2017).

A study by Usman et al. (2019) on securing data transmission from MITM attacks proposed the use of Deffie Hell-Man key exchange encryption. The proposed system could be used to send messages to user's email addresses, and can be used by organizations in sending secure files to their clients and employees. Messages sent through the system are encrypted to prevent its content from MITM attacks. While the encrypted message is on transit, the secret key does not go along with the message, as it is sent through a different channel to the legitimate user. The implementation of the proposed system would enable the system to encrypt data and automatically generate a secret key. This mechanism improves the integrity and enhances the confidentiality of the encrypted message to be sent. If an attacker succeeds to intercept the transmitted data, they would not be able to decode since they do not have the key to decrypt it.

A study by Mastkar et al. (2018) reveals that there are various attacks in the mobile banking channel such as active and passive attacks. Thus to secure user data in mobile banking, a combination of steganography and cryptography can be implemented in order to provide an extra level of security. Steganography and cryptography are related to each other. Steganography hides the presence of a message in a video or image format and cryptography converts the original text into cipher text.

A study by Abdullayev and Chauhan (2023) on SQL injection attack reveals that SQLIA are a serious security risk that can harm businesses and their private data. Organizations therefore need to use a synergy of detection and prevention techniques in order to successfully detect and stop attacks. The different available detection techniques include log analysis, intrusion detection systems and honeypots. On the other hand, prevention techniques include utilization of input validation, parameterized queries, and use of stored procedures. Thus to offer the highest degree of protection against SQLIA, it is crucial for organizations to adopt a comprehensive security strategy that combines a variety of these techniques. Additionally, since the threat landscape is constantly changing, organizations should stay abreast with newest security technologies and best practices.

DISCUSSION AND FUTURE RESEARCH

This study sought to answer three research questions. The first question sought to assess security threats that affect user-data in mobile banking applications. Findings from this study indicate that there are several security threats that affect user-data in mobile banking applications such as mobile malware, packet sniffing attacks, MITM attacks, eavesdropping attacks, DoS attacks, social engineering attacks, Cross-Site scripting attacks, and SQL injection attacks. These threats are dangerous to both users of mobile banking technology as well as banks. Due to weak implementation of security architecture in mobile banking applications, cyber-criminals can launch different types of attacks on mobile banking applications and steal confidential information and money from users of this technology. Because of this reason, confidentiality, integrity, and availability of user-data is lost and therefore current users of this technology as well as prospective users will not embrace it. On the other hand there is a negative image on banks since banks have immersed a lot of money in developing mobile banking systems with the aim to improve customer service as well as reduce operational costs.

The second question sought to assess techniques used to secure user-data in mobile banking applications. Findings for this question reveal that there are several techniques that are used to protect user-data from the identified security threats. Mobile malware was identified as a security threat for mobile banking applications. This form of threat can be mitigated by using the following methods: mobile banking applications should be updated regularly in order to fix security patches, mobile security software should be used to annihilate malware, applications should only be downloaded from official stores, and finally users should employ mobile phone screen lock protection. Attacks that arise from packet sniffing can be prevented through the use of strong encryption mechanisms, utilization of static IP and static ARP tables, restrictions of network to authorized users only, utilization of IPv6 instead of IPv4, use of HTTPS for secure connection instead of HTTP, and use of network switches instead of hubs. Similarly, attacks that arise from MITM can be prevented through deployment of network switches, use of firewalls, and increasing DNS infrastructure entries Time-to-Live (TTL) values for navigating the DNS echelons.

Techniques that prevent eavesdropping attacks are: network segmentation, network monitoring, authentication, use of firewalls, use of virtual private networks, and antimalware software. In regard to social engineering attacks, user education such as planned retreats for bank staff or advertisements to customers using videos as well as emails sent to customers can be used. For cross-site scripting attacks, various techniques can be used such as use of filters, HTTP entity encoding, content security policy, data validation, and the different forms of escaping. Finally, SQL injection attacks can be prevented by use of input validation, use of parameterized queries, and utilization of stored procedures.

Finally the third question sought to assess some of the open research issues in securing mobile banking applications. Findings for this question reveal that as mobile banking applications technology continue to advance the state-of-the-art mechanisms used to mitigate the security threats in mobile banking also advance in equal measure. Some of the current techniques securing user data include authentication, password restrictions, cryptography, hash functions, and steganography. Passwords such as OTPs can enhance security of mobile banking. Nevertheless, when they are captured through social engineering, they may be used for unauthorized transactions. In the case of encryption, only the intended recipient of sent information can decrypt it. However, encryption does not protect data from being intercepted and therefore can be exploited using vulnerabilities inherent in encryption schemes. Similarly, application of hush functions is useful for data security, which offers message authentication, data integrity, and digital signatures. Unfortunately, hash functions suffer from brute force attacks and cryptanalysis attacks. Steganography is used to hide data so that no one should discover the presence of the hidden data except the sender and the recipient. However, steganography suffers from steganalysis attacks which can be visual, statistical or structural attacks.

One of the widely used and effective security mechanisms in mobile banking is two-factor authentication. This involves use of biometric authentication using facial or fingerprint authentication. This study suggests the use of three-factor authentication mechanism in which face scan is done which then allows one to speak (voice input) before being authenticated to access the banking services remotely, which finally is followed by an encrypted OTP to carry out any transaction.

Conclusion

The security threat environment in mobile banking keeps changing with advancement in technology. As a result, several attacks can be conducted on the mobile banking channel such as eavesdropping attacks, man-in-the-middle attacks, mobile malware, packet sniffing attacks, denial of service attacks, social engineering attacks, cross-site scripting attacks, and SQL injection attacks. Fortunately, these attacks can be mitigated. For example, solutions that can be applied to eavesdropping and man-in-the-middle attacks include deployment of network switches, application of cryptography to enhance mobile banking security and use of firewalls. One key solution to social engineering attacks is training users in order to create awareness so that their susceptibility to fall victim to cyber criminal attacks can be reduced. The user education can take the form of showing warnings and online training through games and television videos. SQL injection attacks can be mitigated through the use of input validation, parameterized queries, and stored procedures. Mobile malware can be mitigated by updating applications regularly, use of antivirus software, and downloading applications from official stores. Cross-site scripting attacks can be prevented through use of content security policies, data validation, network filtering, and various forms of escaping. There are several security solutions to mobile banking such as authentication through use of passwords, biometric authentication, encryption, hash functions, and steganography.

CONFLICT OF INTERESTS

The authors have not declared any conflict of interests.

REFERENCES

Abdullayev V, Chauhan AS (2023). SQL Injection Attack: Quick View. Mesopotamian Journal of Cybersecurity 2023:30-34

Abuhamad M, Abusnaina A, Nyang DH, Mohaisen D (2020). Sensor Based Continuous Authentication of Smartphones' Users Using Behavioral Biometrics: A Contemporary Survey. IEEE Internet of Things Journal 8(1):65-84.

Acharya S, Joshi S (2020). Impact of Cyber-attacks on Banking Institutions in India.A Study of Safety Mechanisms and Prevention Measures. Palarch's Journal of Archeology of Egypt 17(6):4656-4670.

Agrawal DP, Wang H (2018). Computer and Cyber Security. Auerbach Publications, New York. https://doi.org/10.1201/97804 29424878 on 11/1/2023

Alajanbi M, Ismail MA, Hasan RA, Sulaiman J (2021). Intrusion Detection: A Review, Mesopotamian Journal of Cyber Security 2021:1-4.

Anu P, Vimala S (2017). A Survey of Sniffing Attacks on Computer Networks. International Conference on Intelligent Computing and (I2C2) pp. 1-5.

Bagudu H, Khan S., Roslan A (2017). The Effect of Mobile Banking on the Performance of Commercial Banks in Nigeria. International Research Journal of Management, IT & Social Sciences 4(2):71-76.

Baklizi M, Atoum I, Hasan MAS, Abdullah N, Al-Wesabi OA, Otoom AA (2023). Prevention of Website SQL Injection Using a New Query Comparison and Encryption Algorithm. International Journal of Intelligent Systems and Applications in Engineering 11(1):228-238.

Bhattacharya I, Reddy PS (2022). Packet Sniffer. Journal of Engineering Sciences 13(6):204-211.

Bojjagani S, Sastry VN (2017). A Threat Model for Vulnerability Assessment and Penetration Testing for Android and Ios Mobile Banking Applications. 3rd International Conference on Collaborative and Internet Computing pp. 77-86. IEEE.

Brandt M, Dai T, Klein A, Shulman H, Waidner M (2018). Domain Validation for Man in the Middle Resilient Public Key Infrastructure. In Proceedings of the 2018 Conference on Computer and Communications Security pp. 2060-2076

Camillo M (2017). Cybersecurity. Risks and Management of Risks for Global Banks and Financial Institutions. Journal of Risk Management in Financial Institutions 10(2):196-200.

Chen HC, Nshimiyimana A, Damarjati C, Chang PH (2021). Detection and Prevention of Cross-Site Scripting attack with Combined Approaches. International Conference on Electronics, Information, and Communication pp. 1-4. IEEE

Chou D, Jiang M (2021). A survey on data-driven network intrusion detection, ACM Computing Surveys 54(9):1-36.

Costantino G, La Marra FA, Martinelli F, Matteucci I (2018). Candy. A Social Engineering Attack to Leak Information from Infotainment System. 87th Vehicular Technology Conference pp. 1-5.

Crespo-Martínez IS, Campazas-Vega A, Guerrero-Higueras AM, Riego-DelCastillo V, Álvarez-Aparicio C, Fernández-Llamas C (2023). SQL injection attack detection in network flow data. Computers and Security 127:103093

Digital Transformation Cyber Security News (2016). Tesco Bank Fined £16.4m after Hackers siphoned £2.26m from customers in 2016**.** https://www.thedrum.com/news/2018/10/01/tesco-bank-fined-164m-after-hackers-siphoned-226m-customers-2016

EMBASB (2022). European Mobile Banking Apps White Paper. Retrieved from https://www.kartensicherheit.de/media/banking_mobile_apps_-_white_paper_-_eshard.pdf

Ghafir I (2016). Social Engineering Attack Atrategies and Defense Approaches. Proceedings of the IEEE International Conference on Future Internet of Things and Cloud pp. 145-149.

Gupta S, Gupta BB (2016). XSS-SAFE: A Server-Side Approach to Detect and Mitigate Cross-Site Scripting (XSS) Attacks in JavaScript Code. Arabian Journal for Science and Engineering 41:897-920. https://doi.org/10.1007/s13369-015-1891-7

Gupta BB, Gupta S, Chaudhary P (2017). Enhancing the Browser-Side Context-Aware Sanitization of Suspicious HTML5 Code for Halting the DOM-Based XSS Vulnerabilities in Cloud. International Journal of Cloud Applications and Computing 7(1):1-31.

Gupta S, Gupta BB (2018). XSS-Secure as a Service for the Platforms of Online Social Network-Based Multimedia Web Applications in Cloud. Multimedia Tools and Applications 77:4829-4861.

Hadabi A, Elsamani E, Abdallah A, Elhabob R (2022). An Efficient Model to Detect and Prevent SQL Injection Attack, Journal of Karary University for Engineering and Science pp. 1-6.

Hasham S, Joshi S, Mikkelsen D (2019). Financial crime and fraud in the age of cybersecurity. McKinsey & Company pp. 1-11.

Hossain MA, Ahmed F (2014). Evaluating the Impact of Mobile Banking Deployment for Microfinance. University of Dhaka Journal of Marketing 2012(15):144-157.

Hubballi N, Tripathi N (2017). A Closer Look into DHCP Starvation Attack in Wireless Networks. Computers & Security 65:387-404.

Imagine IT (2023). The Rise of Mobile Malware. https://imagineiti.com/the-rise-of-mobile-malware/

Javeed D, Badamasi UM, Ndubuisi CO, Soomro F, Asif M (2020). Man in the Middle Attacks: Analysis, Motivation and Prevention. International Journal of Computer Networks and Computing Security 8(7):52-57.

Jiang F, Fu Y, Gupta BB, Liang Y, Rho S, Lou F (2020). Deep Learning Based Multi-Channel Intelligent Attack Detection for Data Security. IEEE Transactions on Sustainable Computing 5(2):204-212.

Kaka S, Sastry VN, Maiti RR (2017). On the MitM Vulnerability in Mobile Banking Applications for Android Devices. International Conference on Advanced Networks and Telecommunications Systems pp. 1-6.

Kaur G, Pande B, Bhardwaj A, Bhagat G, Gupta S (2018). Efficient yet Robust Elimination of XSS Attack Vectors from HTML5 Web Applications Hosted on OSN-Based Cloud Platforms. Procedia Computer Science 125:669-675.

Khomich A (2022). Designing Mobile Banking Apps. ttps://www.uxmatters.com/mt/archives/2022/07/designing-mobile-

Kieseberg P, Fruhwirt P, Schrittwieser S, Weippl E (2015). Security Tests for Mobile Applications why using tls/ssl is not enough. 8th International Conference on Software Testing, Verification and Validation pp. 1-2.

Kirsten S (2016). Cross Site Scripting (XSS) Software Attack. https://owasp.org/www-community/attacks/xss

Kumar M (2023). SQL Injection Attack on Database System. Wireless Communication Security pp. 183-198.

Manhas S (2022). An Interpretive Saga of SQL Injection Attacks," in Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 1:3-12. Singapore: Springer Nature Singapore.

Mastkar N, Isankar M, Sheikh FR, Singh D, Ramteka S (2018). Survey Paper on Securing Online Transaction using Cryptography and Steganography. International Journal of Scientific Research in Science, Engineering and Technology 4(4):463-465.

Mtaho AB (2015). Improving Mobile Money Security with Two-Factor Authentication. International Journal of Computer Application 109(7).

Ndatinya V, Xiao Z (2015). Network Forensic Analysis using Wireshark, International Journal of Sensor Networks 10(2):97-106

Nerwal B, Mohapatra AK, Usmani KA (2019). Towards a Taxonomy of Cyber Threats against Target Applications. Journal of Statistics and Management Systems 22(2):301-325.

Okpara OS, Bekaroo G (2017). Fingerprint-Based Authentication in M-wallets using Embedded Cameras. IEEE International Conference on Environment and Electrical Engineering and Industrial and Commercial Power Systems Europe 1-5.

Positive Technologies (2020). Vulnerabilities and Threats in Mobile Banking. https://www.ptsecurity.com/ww-en/analytics/vulnerabilities-mobile-banks-2020/

Raharja PSJ, Tresna R (2019). Adoption of Information and Communication Technology on Enhancing Business Performance. Study on Creative Industry SMEs in Bandung City, Indonesia. Review of Integrative Business and Economics Research 8(3):20-30.

Sahoo SR, Gupta BB (2019). Classification of Various Attacks and Their Defense Mechanism in Online Social Networks. A Survey. Enterprise Information Systems 13(6):832-864. https://doi.org/10.1080/17517575.2019.1605542

Salahdine F, Kaabouch N (2019). Social Engineering Attacks: A Survey. Future Internet 11(4):89.

Sahin M, Ünlü T, Hébert C, Shepherd LA, Coull N, Mc Lean C (2022). Measuring developers' web security awareness from attack and defense perspectives. In 2022 IEEE Security and Privacy Workshops (SPW) pp. 31-43.

Salim A, Sagheer AM, Yaseen L (2020). Design and Implementation of a Secure Mobile Banking System based on Elliptic Curve Integrated Encryption Schema Springer Nature Switzerland 1174:424-438 https://doi.org/10.1007/978-030-78752-5_33

Sang NM (2021). Critical Factors affecting Consumer Intention of using Mobile Banking Applications During CCOVID-19 Pandemic: An Empirical Study from Vietnam Journal of Asian Finance, Economics and Business 8(11):157-167.

Shachi M, Shourav NS, Ahmed ASS, Brishty AA, Sakib NA (2021). Survey on Detection and Prevention of SQL and NoSQL Injection Attack on Server-side Applications. International Journal of Computer Applications 183(10):1-7.

Stevens C (2020). Assembling Cybersecurity: The politics and materiality of Technical Malware Reports and the Case of Stuxnet. Contemporary Security Policy 41(1):129-152.

Tran HTT, Corner J (2016). The impact of communication channels on mobile banking adoption. International Journal of Bank Marketing 34(1):78-109.

Tripathi N, Hubballi N (2021). Application Layer Denial-of-Service (DoS) Attacks and Defense Mechanisms: A Survey. Association for Computing Machinery 1(1):1-33.

Usman K, Obilikwu P, Patrick K, Karim R (2019). Securing Data on Transmission From Man-In-The-Middle Attacks using Diffie Hell-man Key Exchange Encryption Mechanism. International Journal of Engineering and Science 8(8):88-94.

Xu G, Xie X, Huang S, Zhang J, Pan L, Lou W (2020). A Novel Policy-Based XSS Defense Mechanism for Browsers. IEEE Transactions on Dependable and Secure Computing 19(2):826-878.

Yadav N, Shekokar NM (2023). SQL Injection Attacks on Indian Websites: A Case Study, in Cyber Security Threats and Challenges Facing Human Life: Chapman and Hall/CRC pp. 153-170.

Yildirim N, Varol A (2019). A research on Security Vulnerabilities in Online and Mobile Banking Systems. 7th International Symposium on Digital Forensics and Security pp. 1-5.

Zhu L, Hu Z, Heidemann J, Wessels D, Mankin A, Somaiya N (2015). Connection-Oriented DNS to Improve Privacy and Security. IEEE Symposium on Security and Privacy pp. 171-186.