SYSTEMATIC REVIEW OF SESSION HIJACKING ATTACKS ON 5G NETWORK

Abdulkarim Muhammad Kabir,
Cyber Security Department,
Federal University of Technology, Minna, Niger State.

Olawale Surajudeen Adebayo,
Cyber Security Science Department,
Federal University of Technology, Minna, Niger State.

Shafii Muhammad Abdulhamid
Cyber Security Department,
Federal University of Technology, Minna, Niger State

Sulaiman Adebayo Bashir
Computer Science Department,
Federal University of Technology, Minna, Niger State..

Abstract

Cyberattacks, such as session hijacking, are possible on computer networks, with attackers abusing the network's inherent vulnerability. This attack occurs when hacker takes over a user's session id and gains complete control of the system while the session is continuing. An adversary serving as a proxy can get unauthorized access to data and applications in systems/services/networks after obtaining the compromised session key or token. This research carried out a thorough examination of the activities of session hijacking attacks on computer networks which were published between 2017 and 2021 and focused more on 5G network. Different techniques adopted by hackers were examined and x-rayed and mitigation strategies are reported. In addition, the paper proposes a model that can be used to check the effect of session hijacking attacks on 5G network.

Keywords: *Cyberattacks,* S*ession Hijacking,* S*ession Hijacking Attacks, 5G network, Attack Mitigation Strategies*

1. INTRODUCTION

Session hijacking attack is a network attack where an attacker gains unauthorized access to the legitimate user's network and takes control of it in order to steal vita information and commit other crimes. After gaining unauthorized access to the system, the adversary can either steal or guess a valid session token (Pothumarti et al., 2021). The Internet of today serves as more than just a global

platform for communication and information sharing. Instead, it has developed into a productive platform for business and commerce, banking, and interpersonal interaction. The majority of these Internet-based transactions are carried out utilizing HTTP and HTTPS, two application layer protocols. The Internet of today serves as more than just a global platform for communication and information sharing. Instead, it has developed into a productive platform for business and commerce, banking, and interpersonal interaction. The majority of these Internet-based transactions are carried out utilizing HTTP and HTTPS, two application layer protocols. Since HTTP does not maintain the status of a client connection to the server, this has been its primary drawback. Cookies and sessions were established to deal with the issue of storing client state data (Barth, 2011). Figure 1 depicts a typical client-server HTTP conversation using cookies and session-IDs. However, the design of client-side browser which stores cookies and Session-IDs is not capable of preventing several sophisticated attacks like cross-site request forgery (CSRF), and session hijacking using cross-site scripting (XSS).

Session begins when a user accesses or logs in to a certain online page or program on their computer, and ends when the user shuts down or logs out the computer, or closes the web page or program. A session of a network keeps the information about the user's activities, while users are connected, A session begins when a user accesses or logs in to a certain online page or program on their computer, and ends when the user shuts down or logs out the computer, or closes the web page or program. In (Thakkar & Vaghela, 2018), an unauthorized user taking over an already established trustworthy and lawful session between two systems in order to steal and compromise the user's sensitive data, also known as a man-in-the-middle attack. When a user logs into a web application, the three-way handshaking protocol establishes a secure connection between the client and the web server. Three-way handshaking is a method of establishing a safe and trustworthy

connection between a client system and a web server. Once a secure and trusted connection has been established, only the client and server can communicate with each other and transmit and receive data.

In a session hijacking attack, the attacker takes control of a valid trusted connection and sends packets to a server as a real client, receives packets from the server, and sends packets back to the client as a genuine server. The main benefit of a session hijacking assault is that it does not require breaking any defensive or security firewalls; all it requires is to keep listening to the network and hijack any legal session.

Session hijacking attack is one of the 5G network's security challenges. Fifth-generation (5G) systems are those that meet the ITU's IMT-2020 criteria, as stated in 2015 IMT-2020 anticipated support for a variety of scenarios, including improved mobile broadband (eMBB), ultra-reliable and low latency communications (URLLC), and massive machine-type communications (mMTC): eMBB is the inevitable progression of 4G network-based broadband services according to Ahmad et al., (2018). There are three type of session hijacking attacks which does the same work in different way namely: active session hijacking, passive session hijacking, and hybrid session hijacking.

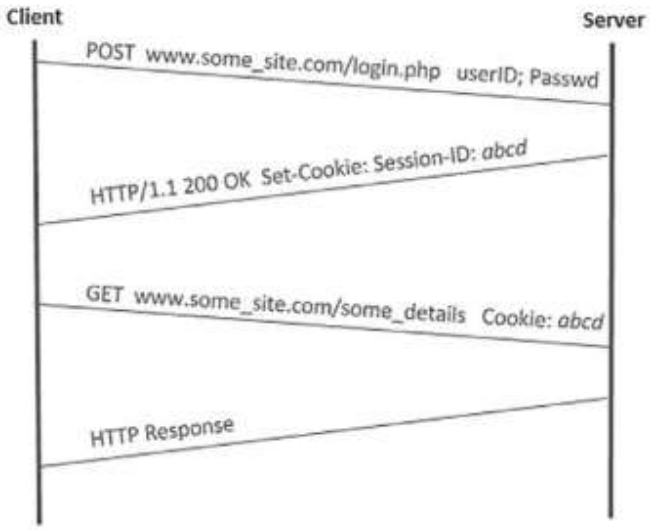

Figure 1. Typical client-server communication over HTTP
Sinha, A.K., & Tripathy, S. (2019).

2. RELATED WORKS

Given the detrimental consequences of session hijacking, researchers have been working to develop a long-term solution (Ogundele et al., 2020), (Chen et al., 2020) and (Meenakshi, 2020). As a result, the recommended solutions are expected to significantly minimize the attack's negative impact, prevent the attack, or, ideally, eradicate it entirely. Analysis of session hijacking activities, including attack kinds, detection, and protection measures, flooded the literature as a result of identifying a solution to session hijacking.

(Mao et al., 2018) elucidated that Ron Rivest, Adi Shamirh, and Len Adleman developed RSA in 1977 as a popular public key encryption scheme that may be used for both encryption and digital signatures. It generates a publicly distributed key and a secret private key for encryption and decryption. Its security is based on the idea that large number factorization is difficult, and it has shown to be quite secure, as it has so far resisted all known password attacks. Since 1992, it has been the ISO-recommended public-key data encryption standard.

(Koblitz et al., 2000) revealed how Elliptic Curve Cryptography (ECC) was invented in 1985 by Victor Miller and Neal Koblitz. ECC has recently attracted a lot of interest from industry and academia as a viable alternative to well-known public-key systems such as Digital Signature algorithm (DSA) and RSA. The primary reason for ECC's attraction is that no sub-exponential solution exists to overcome the discrete logarithm issue on a properly built elliptic curve. This means that, while maintaining the same level of security, ECC can use far fewer parameters than

rival systems like RSA and DSA. Lower key sizes offer benefits such as faster computations, as well as savings in computing power, storage space, and bandwidth.

3. SESSION HIJACKING ATTACKS CATEGORIES

The three types of session hijacking attacks are:

a. Active session hijacking b. Passive session hijacking. c. Hybrid session hijacking.

A. Active Session Hijacking

An attacker attacks an already active session between the user and the server using active session hijacking. Using a denial-of-service attack, the attacker disrupts an active session and takes the place of a valid user (DOS). Using a packet-capturing tools such as Wireshark, the attacker sniffs the connection and collects all data packets between the user and the server before launching the DOS assault. Denial of service attack occurs when an attacker floods the target with traffic by sending a large number of requests or information to the target network, rendering the server unavailable. As a result, the target system is unable to use the services sent by the server, and the target machine may shut down or crash in order to handle the traffic flood. The server waits a little time before sending another connectivity request to the user computer, at which point the attacker disguises himself as a valid user and sends an acknowledgment to the server, allowing the attacker to connect to the server in the place of a valid user. Figure 1 shows a diagram of an active session hijacking attack.

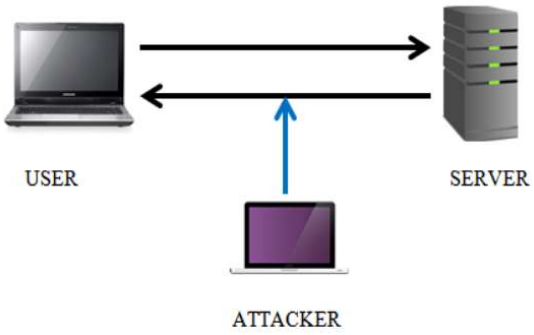

Fig. 1: Active Session Hijacking

Source: (Baitha & Vinod, 2018)

B. Passive Session Hijacking

In Passive Session Hijacking, the attacker places himself between the legitimate user and the server. The attacker inserts himself between the legitimate user and the server, sending valid packets to the user while pretending as a server and receiving packets from the user while appearing as a real user in passive session hijacking. The attacker can transit all data through his system and even make changes to data packets using this Passive Session Hijacking approach, with neither the user nor the server being able to detect the changes. As a result, the attacker will have all of the information he or she needs to carry out their wicked plans. The attacker, on the other hand, has a flaw. If a user sign s up for a session, the data between the user and the server is only visible while the session is active.

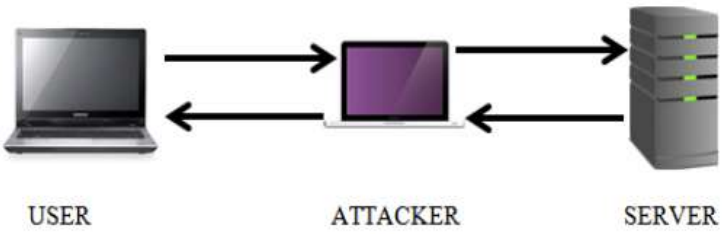

Fig. 2: Passive Session Hijacking

Source: (Baitha & Vinod, 2018)

C. Hybrid Session hijacking

Active Session Hijacking and Passive Session Hijacking are combined in Hybrid Session Hijacking. The attackers in this case use both active and passive session hijacking techniques to achieve their goal. Hybrid Session Hijacking is classified into two categories, as follows:

A. Non-Blind Spoofing Attack

B. Blind Spoofing Attack

A. Blind Spoofing

A type of attack in which the attacker attacks the target system without making any changes to the server-to-victim machine connection. An attacker merely collects the entire packet from the network between the user and the server in order to get the TCP sequence number and gain complete control over the session. However, there is a significant flaw in this attack: it is extremely difficult to determine or guess the TCP sequence number from captured packets because TCP sequence numbers are generated at random intervals, making it extremely difficult to determine the correct sequence number. Finding the correct sequence number takes a long time, so the attacker must continue to capture packets in order to analyze the TCP sequence. It may take a long time to locate the correct sequence number, or the attacker may have to wait patiently to succeed in these types of attacks (Thakkar & Vaghela, 2018).

B. Non-Blind spoofing attack

Non-Blind spoofing attack in which the attacker is on the same network as the victim and is on the same subnet as the victim, allowing the attacker to monitor the communication between the victim and the server. Because the attacker can see the packets moving over the same network, it is simple for the attacker to monitor traffic from the same network. Attackers keep an eye on the connection and try to estimate the TCP sequence number of the next packets in order to use the TCP sequence

number to get authentication through the connection. Attackers determine the right sequence number and re-establish the connection with the server using that number. But the fundamental difficulty with this approach is that today's routers don't allow packets to be broadcast over the network; instead, they keep it switched off to protect packets. To circumvent this obstacle, the attacker resets the connection, putting him between routers and allowing him to grab the first broadcast packet (Thakkar & Vaghela, 2018)

4. RESEARCH METHODOLOGY

This study conducted a search on the electronic digital databases. Different criteria were used to select the most relevant articles to the study. Only articles that are five years old were considered relevant. Reviewed papers, which address session hijacking attacks mitigation, were considered in the study. The initial search was conducted in February 2022 in popular computing electronic databases. Electronic databases used include Science direct, IEEE Xplore and Google Scholar, Scopus, Springer. For each of the hit search results, the maximum of the first 200 hits were screened. That was because the search beyond 200 was unlikely going to yield relevant results. Table1 shows the list of electronic databases that were used to perform the literature search. Different keywords were used in order to extend the search and to ensure that we cover as much related manuscripts as possible.

Table.1 Sample of Electronic Databases used during literature search

Name of Database	Access Method	Website
Scopus	Online Search	https://www.elsevier.com/solutions/scopus

Science Direct	Online Search	https://www.sciencedirect.com/
IEEE Xplorer	Online Search	https://ieeexplore.ieee.org
Springer	Online Search	https://www.springer.com
Google Scholar	Online Search	https://scholar.google.com/

4.1 Search keywords

The literature search approach of (Kitchenham et al., 2009) was used in this study. To determine the most appropriate search phrases, the core search terms were carefully chosen. The following terms were used to search the relevant material in some renowned academic archives using the review specified goals: "session hijacking prevention in 5g network", "session hijacking mitigation in 5g network"

4.2 Explicit inclusion and exclusion criteria

We employed a number of principles in the selection of articles in order to maintain a direct focus on the subject matter and avoid any biases in the review of papers, which are described in Table 4 below.

Table 4.3. Inclusion/Exclusion criteria

S/N	Inclusion criteria	Exclusion criteria
1	The research focuses on the attack and countermeasures against session hijacking.	The study did not focus on session hijacking detection

2	The topic was peer-reviewed and published in scholarly publications or conference papers.	The topic has not been peer reviewed or published in any academic journals or conference papers.
3	The research paper is written in English.	The study is not written in English language
4	The articles are either surveys or research papers that have been published.	The pieces are not surveys or research studies, but rather news flashes or magazine articles.

5. RESULTS

This study's peer-reviewed academic publications between 2017 and 2021 on session hijacking attacks. This was because the goal was to figure out what mitigating measures researchers have tried in the previous five years. After screening, the relevant review papers published in 2017 totaled 2 and accounted for 7.4 percent of our study; those published in 2018 totaled 5 and accounted for 18.5 percent; and those published in 2019 totaled 1 and accounted for 3.7 percent. In 2020, only 5 studies were published, accounting for 18.5 percent of the total. Finally, in 2021, there were 0 and contributed 0.0 percent.

Publication Year	Number	Percentage
2017	2	7.4
2018	5	18.5
2019	1	3.7
2020	5	18.5
2021	0	0.0

The reviewed articles selected above are summarized in the Table 3:

S/N	Title	REFERENCE	EXISTING SOLUTION/TECHNIQUE/CONRIBUTION	PROBLEM ADDRESSED	METHODOLOGY	RESEARCH GAP/LIMITATION	RESULTS/FINDINGS/ ADVANTAGE
1	Detection and Prevention of Session Hijacking in Web Application Management	(Ogundele et al., 2020)	Developed a web application for e-Commerce session hijacking detection and prevention. Furthermore, a session key is generated using a technique that generates lengthy random alphanumeric characters.	Session hijacking	The application was created with the help of HTML, PHP, CSS, and JAVASCRIPT programming languages, as well as MySQL as the database.		
2	Encryption Algorithm for TCP Session Hijacking	(Chen et al., 2020)	Combines RSA-based cryptography, RSA-based signatures, the DH key exchange algorithm, and HAMC-SHA1 integrity verification technology with the TCP protocol to present a security method that effectively defends against TCP session hijacking.	TCP Session Hijacking	Cryptography	Each data packet's HAMC-SHA1 signature and integrity verification consume a significant amount of system resources. Future work will include a huge number of cyberattack tests and	During the TCP three-way handshake phase, session hijacking can occur. The proposed approach can successfully alleviate this problem.

						mathematical proofs, as well as finding a balance between security and resource consumption by using suitable window settings.	
3	Prevention of session hijacking using token and session id reset approach	(Meenakshi, 2020)	To prevent session hijacking by cookie cloning, a Token and Session id Reset Approach has been proposed and implemented. To authenticate the user on the web server, the proposed technique uses session id, token, IP, and bowser fingerprint . This approach saves the token in local storage on the client side, rather than in cookies. The token is a 32-character long string encrypted with Triple DES.		Cryptography		
5	An Effective Method for Preventing SQL Injection Attack and	(D'silva et al., 2017)	Hashing technique is used for implementing the prevention.	Proposes a practical method for avoiding SQL injection attacks and session hijacking.	Hashing Technique. The hashing algorithm used isö SHA1	The proposed solution is used to avoid SQL injection and session hijacking, however it does	Without significant overhead on the application, the suggested method efficiently eliminates

	Session Hijacking					not address other Web application flaws.	SQLIAs and session hijacking attacks.
6	Preventing Session Hijacking using Encrypted One-Time-Cookies	(Prapty & Azmin, 2020)	A cookie protection mechanism that is both secure and effective was proposed. To prevent an attacker from injecting cookies, we've employed one-time cookies. We've encrypted critical information in the cookie to ensure cookie integrity and confidentiality.	Session hijacking	Use of Enctypted One-Time-Cookies	Only 1 session/user and reliant on the reverse proxy server This could cause problems with online services when it comes to establishing connections and creating sessions for web applications.	By leveraging OTC instead of expensive HTTPS connections, the suggested method effectively prevents session hijacking and cookie poisoning attacks.

7	Prevent Session Hijacking Using Cookie Authenticatio n	(Kumar, 2018)	During the initial HTTP response phase, I proposed using two cookies instead of one for authentication purposes. Discuss some of the ideas of session hijacking in the context of a Man-In-The-Middle attack, as well as				

			processes for preventing session hijacking using the forms authentication mechanism..				
8	Secure Model for Session Hijacking using Hashing Algorithm	(Thakkar & Vaghela, 2018)	The suggested system produces the session id using the MD5 method using the MAC address and client's id. The attacker will be unable to access the session if the session key is successfully stolen, as the MAC is also required to access the session.	To prevent session hijacking by generating a hash string from the combined string of the user machine's MAC address and the user ID.	The MD5 technique is used to generate the session id using the MAC address and the client's id.		The proposed solution precisely mitigates a web application session hijacking attack.
9	Session Hijacking Attacks in Wireless Networks: A Review of Existing Mitigation Techniques	(Letsoalo & Ojo, 2017)	This article looked at literature review papers that were published between 2012 and 2016. Various ways for reducing SHAs have been presented in the literature, most of which are based on media access control addresses, with varying degrees of influence on network performance and resource consumption. It has also been discovered that full scale authentication can avoid session hijacking attacks, which is not	In terms of minimizing SHA, current SHA techniques do not effectively address efficiency and accuracy.	Mitigation approaches are classified in this research based on their strengths and limitations, as well as gaps and places for improvement.	Future research will focus on developing a method that detects both managemen t and control frame spoofing with little infrastructur e and computing overhead, as well as	The existing literature study will serve as a foundation for future thinking about SHA mitigation approaches that will take use of the synergy of two or more SHA mitigation techniques.

			achievable in wireless networks. Both authentication and encryption of management and control frames are not available in 802.11 wireless networks. The plain text management and control frames render a network vulnerable to session hijacking attempts.			fewer false positives and false negatives. Various efficient and reliable strategies will be integrated with machine learning techniques to build a hybrid solution for detecting and preventing session hijacking attempts with low false positives and false negatives while not compromising network performance.	

10	Session Hijacking and Prevention Technique	(Baitha & Vinod, 2018)	The goal of this paper is to provide detailed information on session hijacking and how to protect yourself from such attacks.	Session Hijacking has been highlighted, with a particular focus on one of the most common attacks, the SSL Strip attack, which plays a critical role in this type of attack.	On the Networking Layer: SSL and SSH, as well as HTTPS, were recommended. Application Layer: -A Session ID that is both complex and strong -Session ID created by server -Random Session ID -Automatic Logout - Encrypted Session ID	The offered countermeasures do not completely prevent a session hijacking attempt, but they do make it more difficult for the attacker to succeed.	
11	One-time cookies protocol based on one-time password	(He et al. 2019)	This study presented an OTC protocol based on OTP. The OTP algorithm was developed to generate a one-time password in response to the flaws in Dacosta et alOTC. .'s vital to ensuring cookie security while also avoiding the time synchronization issue	Dacosta et al. suggested the one-time cookies (OTC) protocol to thwart assaults. Unfortunately, one of its fundamental flaws is that it relies on temporal synchronization between two workstations for	Hashing To eliminate time synchronization issues and generate a dynamic key for OTC security, the protocol uses the OTP algorithm based on a hash chain. We have improved the	Improvements to the OTP algorithm are being considered as future work to ensure that the hash chains on both sides of the communication are	The results suggest that the proposed OTC methodology has the potential to provide excellent security with minimum performance impact..

				availability, while another is that it generates OTC using a fixed session key during the session period, making it vulnerable to cracking by potential adversaries. In response to these flaws, the study proposes a unique OTC protocol based on a one-time password (OTP)..	OTP algorithm for efficiency.	synchronized. Furthermore, the novel design of two-factor authentication applications is expected to be examined using the OTC protocol.	
12	A proposed System for Preventing Session Hijacking with Modified One-Time Cookies	(Niranjan et al., 2017)	This study proposes an OTC-based approach for preventing an attacker from gaining access to a cookie and backend server. To prevent an opponent from collecting session credentials, a reverse proxy server with OTC, IP, session ID, and browser fingerprinting is utilized.	Session HIjacking	Use of One Time Cookie		

13	Model to Mitigate Session Hijacking Attacks in Wireless Networks	(Letsoalo & Ojo, 2018)	Introduces the hybrid method, which combines three techniques for triangulation: received signal strength (RSS), round trip time (RTT), and de-authentication frames analysis (DFA)	Session Hijacking Attacks in Wireless Networks	A hybrid technique was used, which combines RSS, RTT, and DFA.		The experiment's findings showed that our proposed SHA mitigation hybrid model has the potential to improve accuracy and efficiency.
14	Past Event Recall Test for Mitigating Session Hijacking and Cross-Site Request Forgery	(Salami et al., 2021)	PERT has been offered as a method of preventing session hijacking and Cross-Site Request Forgery attacks.	session hijacking and Cross-Site Request Forgery attacks	It employs PERT, which assures that a node communicates only with known systems with which it has previously transacted successfully.		It took 35 percent longer on average to execute than the faster benchmark, but 20.06 percent less time on average than the slower benchmark. It blocked 97 percent of requests from an identity thief and 95 percent of requests from a spoofing attacker, respectively. The prevention efficiency of the benchmark solutions was lower.

16	CookiesWall: Preventing Session Hijacking Attacks Using Client Side Proxy	(Tripathy Somanath & Kumar Praveen, 2017)	We developed a Proxy-based approach at the client side called CookiesWall to drop the reply/packet containing Cookies in this paper.	Introduces the cookiewall approach. CookiesWall is written in Python as a client-side proxy.	Use of Python	The mechanism's false positive and false negative states were not described.	CookiesWall has been discovered to be more efficient and speedier.
17	A Preliminary Review on Web Session Hijacking	(Naematul et al., 2018)	Highlights prior scholars' definitions of session hijacking, taxonomy, and attacks that may occur, as well as a variety of suggested methods to detect or avoid the attack in various situations		.		

6. CONCLUSION

This research presents a systematic review on session hijacking attack on 5G network. Only articles published between 2017 to 2021 were considered. Different techniques adopted by hackers were examined and mitigation techniques reported. Finally, the paper proposed to develop a secure model for enhancing security of 5g network against session hijacking using Rivest Shamir Adleman (RSA) and elliptical curve cryptography (ECC).

7. FUTURE WORK

This study gives a systematic review of session hijacking attacks. The publications that were studied discuss what session hijacking attack is and how it works. Majority of the studies suggested mitigation strategies such as token and session id reset, utilizing safe and efficient cookie protection system, using the hashing algorithm methodology, encrypted One-Time-Cookies, and employing Cookie authentication. The proposed research intends to develop a secure model for enhancing security of 5g network against session hijacking using Rivest Shamir Adleman (RSA) and elliptical curve cryptography (ECC).

REFERENCES

Ahmad, I., Kumar, T., Liyanage, M., Okwuibe, J., Ylianttila, M., & Gurtov, A. (2018). *Overview of 5 G Security Challenges and Solutions*. *March*, 36–43.

Baitha, A. K., & Vinod, P. S. (2018). *Session Hijacking and Prevention Technique*. *7*, 193–198.

Chen, M., Dai, F., Yan, B., Cheng, J., & Wang, L. (2020). Encryption Algorithm for TCP Session Hijacking. *In International Conference on Artificial Intelligence and Security*, *12240*, pp 191-202.

D'silva, K., Vanajakshi, J., Manjunath, K. N., & Prabhu, S. (2017). An effective method for preventing SQL injection attack and session hijacking. *RTEICT 2017 - 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, Proceedings*, *2018-Janua*, 697–701. https://doi.org/10.1109/RTEICT.2017.8256687

He, J., & Han, D. (2019). On one-time cookies protocol based on one-time password. *Soft Computing*, *5*. https://doi.org/10.1007/s00500-019-04138-5

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering - A systematic literature review. In *Information and Software Technology* (Vol. 51, Issue 1, pp. 7–15). https://doi.org/10.1016/j.infsof.2008.09.009

Koblitz, N., Menezes, A., & Vanstone, S. (2000). The State of Elliptic Curve Cryptography. *Designs, Codes, and Cryptography*, *19*(2–3), 173–193. https://doi.org/10.1023/a:1008354106356

Kumar, G. S. (2018). *Prevent Session Hijacking Using Cookie Authentication*. *5*(6), 383–387.

Letsoalo, E., & Ojo, P. S. (2017). *Session Hijacking Attacks in Wireless Networks : A Review of Existing Mitigation Techniques*. 1–9.

Letsoalo, E., & Ojo, S. (2018). *A Model to Mitigate Session Hijacking Attacks in Wireless Networks*. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8417199&isnumber=8417

Mao, J., Zhu, H., Liu, Y., Liu, Y., Qian, W., Zhang, J., & Huang, X. (2018). RSA-Based Handshake Protocol in Internet of Things. *Proceedings - 9th International Conference on Information Technology in Medicine and Education, ITME 2018*, 989–993. https://doi.org/10.1109/ITME.2018.00220

Meenakshi, T. S. (2020). Prevention of session hijacking using token and session id reset approach. *International Journal of Information Technology*. https://doi.org/10.1007/s41870-020-00486-w

Minghan Chen, Fangyan Dai, Bingjie Yan, and J. C. (2020). *Encryption Algorithm for TCP Session Hijacking*.

Naematul, N., Ismail, S., Warip, M. N. M., Elias, S. J., & Ahmad, R. B. (2018). *A Preliminary Review on Web Session Hijacking*. *7*, 124–129.

Niranjan, S. K., Chirala Engineering College, IEEE Computational Intelligence Society, & Institute of Electrical and Electronics Engineers. (2017). *A Proposed System for Preventing Session Hijackingwith Modified One-Time cookies*.

Ogundele, I. O., Akinade, A. O., & Alakiri, H. O. (2020). *Detection and Prevention of Session Hijacking in Web Application Management*. *September*. https://doi.org/10.17148/IJARCCE.2020.9601

Prapty, R. T., & Azmin, S. (2020). *Preventing Session Hijacking using Encrypted One-Time-Cookies*. 1–6.

Salami, O. W., Bashir, A. M., Adedokun, E. A., & Basira, Y. (2021). Past Event Recall Test for Mitigating Session Hijacking and Cross-Site Request Forgery. *2021 International Conference on Information and Communication Technology for Development for Africa, ICT4DA 2021*, 190–195. https://doi.org/10.1109/ICT4DA53266.2021.9672244

Sinha, A.K., & Tripathy, S. (2019). CookieAmor: Safeguarding against crosssite request forgery and session hijacking. ''Security and Privacy 2(2), p.e0

Thakkar, N., & Vaghela, R. (2018). *Secure Model for Session Hijacking using Hashing Algorithm*. *3*, 70–75.

Tripathy Somanath, & Kumar Praveen. (2017). *Network and System Security* (Z. Yan, R. Molva, W. Mazurczyk, & R. Kantola, Eds.; Vol. 10394). Springer International Publishing. https://doi.org/10.1007/978-3-319-64701-2