

Broken Authentication and Prevention Using Multi-Factor Authentication

Chinmay Nrusingh Choudhury 18BIT0097
Eeshan Pandey 18BIT0242
Vellore Institute Of Technology, Vellore

Abstract—There can be several types of weaknesses in a website that can allow an attacker to either capture or bypass authentication methods that are used by web applications. Hackers employ a wide variety of strategies to take advantage of these weaknesses, ranging from huge credential-stuffing attacks to highly targeted schemes aimed at gaining access to a specific person's credentials. This is known as broken authentication. A broken authentication means that an unauthorized user has been able to bypass authentication policies, compromise passwords, keys or session tokens which give access to user identities and privileges.

Due to these vulnerabilities Broken Authentication also finds a place in OWASP's (Open Web Application Security Project) top 10 application security flaws. As of 2020, it's in the 2nd spot!

Multi-factor authentication can be effective in preventive credential surfing, brute-force and other attacks that can compromise authenticity. In this project we aim to implement multi-factor authentication along with effective session management, to minimize the threat posed by broken authentication.

Keywords—*Session management, session fixation, session hijacking, multi-factor authentication, OWASP, credential surfing, session rotation, Burp Suite, broken authentication.*

I. INTRODUCTION

A. What is Broken Authentication?

Broken Authentication refers to compromise to passwords, session or cookies tokens, account details, keys and any other information related to an user identity. This is a result of weak session and cookie management, weak encryption of data sent over the network and bad communication practises.

Failures in **session management** leads to broken authentication. Authentication not only refers to 'username' and passwords but it also involves the concept of session duration and data (such as cookies).

A **session** describes the interaction of users with an application for a certain duration of time. A new session ID is allotted for a particular user every time a user logs into a system. This session ID is used to identify users as they move through the website.

Without proper management of sessions, the web application becomes vulnerable to **session hijacking**. In this case a hacker steals session IDs to masquerade as another user. This can happen in very simple cases where a user forgets to logout of the web application and an attacker continues the session. Another attack known as **session fixation** is possible if session IDs aren't rotated after successful login.

If an attacker has access to a database containing a list of default usernames and passwords, they can brute-force various accounts by using that list to enter legitimate accounts. This type of attack is called **Credential Surfing**. Botnets are often used to test stolen credentials on different sets of accounts. And because people frequently use the same password over and over, hence it works a lot of time.

In 2018 according to a report by Akamai, cybercriminals made 30 billion login attempts, highlighting the credential-stuffing attacks. The attacks were automated and performed through malicious leverage bots.

Apart from session mismanagement, it is important to cross check user identity before performing vital tasks such as checking out items from an e-commerce website. There are several methods for cross-authentication and some of the most important ones are discussed below.

B. BurpSuite as a Pen-testing tool

Burp Suite is one of the most popular penetration testing tools used for finding vulnerabilities in web applications. It is a proxy based tool used for security based evaluation and hands on testing. It is developed by Portswigger.

We have used it to evaluate the issues with broken authentication on websites that are purposefully made vulnerable by us. Session Stealing, Cookie manipulation and Brute force attacks are the most common threats to a website. Burp Suite makes it possible to practice these attacks by intercepting packets between client and server. By performing these attacks we aim to get a better idea on how to make a website more secure and safe from such attacks.

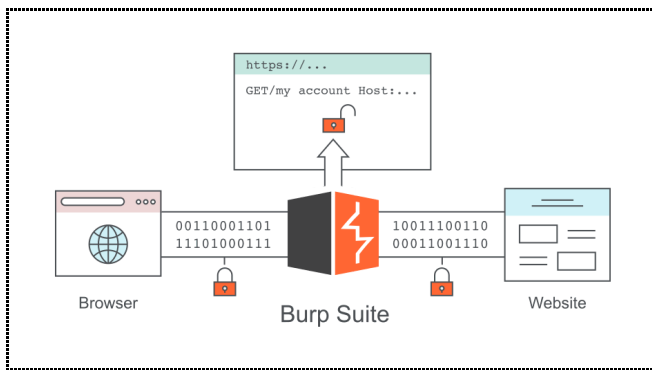


Fig 1.1 An illustration of working of Burp Suite

II. LITERATURE SURVEY

1. Multi-Factor Authentication System : Aldwairi, Monther & Aldhanhani, Saoud. (2017).[1]

In this paper, four authentication methods/stages have been implemented. The first stage consists of writing user credentials and identifying a pattern which was created by the user during signup. Second stage asks for a numerical PIN. The third stage involves token generation in smartphones and finally the last stage asks two security questions. Four stages can be time consuming and tedious for users, but for security critical applications it is very beneficial.

2. Ometov, Aleksandr & Bezzateev, Sergey & Mäkitalo, Niko & Andreev, Sergey & Mikkonen, Tommi & Koucheryavy, Yevgeni. (2018). Multi-Factor Authentication: A Survey. [2]

Elaborated about various widely-deployed MFA sources such as: Password Protection, Token Presence, Voice Biometrics, Facial Recognition, Hand Geometry, Vein Recognition, Fingerprint Scanner, Geographic Location, Behaviour detection, Thermal image recognition, DNA recognition, etc.

3. Implementing Resiliency of Adaptive Multi-Factor Authentication Systems by Kim Phan [3]

This paper presents Adaptive Multi-Factor Authentication (AMFA), which is an enhanced version of MFA that provides a method to allow legitimate users to access a system using different factors that are changing based on different considerations. In other words, authentication factors include passwords, biometrics among others are adaptively selected by the authentication system based on criteria (e.g., whether the user is trying to log in from within the system boundary, or whether the user is trying to access during organization operating hours).

4. Security Of Multifactor Authentication Model To Improve Authentication Systems : Tamara, Mrs & Mohamed, Tamara. (2019). [4]

Multi Factor authentication (MFA) is a security system in which more than one form of authentication is implemented to verify the legitimacy of a transaction. The goal of MFA is to create a layered defense and make it more difficult for an unauthorized person to access a computer system or network.[8]

Multifactor authentication is achieved by combining two or three independent credentials: what the user knows (knowledge-based authentication), what the user has (security token or smart card) and what the user is (biometric verification). Single-factor authentication (SFA), in contrast, only requires knowledge the user possesses. Although password-based authentication is well-suited for website or application access, it is not secure enough for online financial transactions.

5. Root cause analysis of session management and broken authentication vulnerabilities. [5]

The paper proposes strategic approaches to analyse and understand the problem. The paper implements Root Cause Analysis method in session management and vulnerabilities in broken authentication. By employing Root Cause Analysis 11 causes in session management and 9 causes of broken authentication vulnerabilities. In addition to that, the paper also suggests effective solutions to minimise the attacks on web application.

6. Broken Authentication and Session Management Vulnerability : A Case study of Web Application [6]

The paper has analyzed Broken Authentication and Session management vulnerabilities by investigating 257 websites of public and private sectors. The analysis method included manual penetration testing followed by blind testing strategy. And shows results in terms of impact and percentage of vulnerability attacks. In the end the paper proposes prevention techniques to mitigate the vulnerabilities.

7. Automatic recognition, processing and attacking of single sign-on protocols with burp suite. [7]

The paper talks about the working of various single sign on (SSO) methods employed by various industries such as SAML, Mozilla BrowserID, OpenID, OpenID Connect, Facebook, Microsoft Account and OAuth. The paper gives an overview of protocols used by these SSO methods and uses Burp Suite to identify and analyse the browser HTTP traffic to manipulate SSO flows.

8. Burp Suite: Automating Web Vulnerability Scanning [8]

The research aimed at demonstration of a method to automate pentesting of web applications using Burp Suite

vulnerability scanner. The research was conducted to assist organizations to begin implementation of automated web application penetration testing. It also compares the advantages and disadvantages of automated and manual testing. The paper concludes that automated testing reduces organization's risk and financial loss, prevents downtime and data loss, increases productivity and produces a more secure web application.

9. Web Application Exploitation with Broken Authentication and Path Traversal [9]

The paper covers a detailed study on a combination of the most important network vulnerabilities that affect almost every single web app; broken authentication mechanisms, poor session management and the ability of hackers to compromise web apps using path traversal. The most commonly used tool for authentication attack is Burp Intruder. The paper also discusses cookie attacks which aren't and its ineffectiveness in predictive (cracking) cookies.

10. Preventive Measures for Securing Web Applications Using Broken Authentication and Session Management Attacks [10]

In this paper various broken authentication and session management attacks such as brute force attack, session spotting, replay attack, session fixation attack, insufficient session expiration and session hijacking is discussed. Preventive measures such as using secure socket layer, expiry of session after inactivity, secure logout and hidden session identifiers are also suggested in the study.

11. Authentication and Session Management based on AJAX [11]

The Ajax interaction model changes the posture of web applications to become stateful over HTTP. Ajax applications are long-lived in the browser. XMLHttpRequest (XHR) is used to facilitate data exchange. Using HTTPS over this interaction is not viable because of the frequency of data exchange. Moreover, switching of protocols from HTTP to HTTPS for sensitive information is prohibited because of server-of-origin policy. The longevity, constraint, and asynchronous features of Ajax applications need to have a different authentication and session handling mechanism that invoke re-authentication. This paper presents an authentication and session management scheme using Ajax. The scheme is designed to invoke periodic and event based re-authentication in the background using digest authentication with auto-generated password similar to OTP (One Time Password). The authentication and session management are wrapped into a framework called AWASec (Ajax Web Application Security) for coupling to avoid broken authentication and session management.

12. Developing a Secure Web Application Using OWASP Guidelines. [12]

Developing a secure Web application is a very difficult task. Therefore developers need a guideline to help them to develop a secure Web application. Guidelines can be used as a checklist for developers to achieve a minimum standard of secure Web application. This study evaluates how good the OWASP guideline is in helping developers to build secure Web applications. The developed system is then tested using code auditing and penetration testing to identify the achievement of the system security for the application. After applying the testing techniques from Open Source Security Testing Methodology (OSSTMM) on the Top Ten Critical vulnerabilities as defined by OWASP, a standard measure score are calculated. The score is used to decide on the level of security of the developed web application. A high percentage score would indicate that the guideline helps in building a secured web application. Hence, the result proved that OWASP guidelines are effective in ensuring the trustworthiness of the system and can be used as referral by other web developers, especially in developing applications for a university.

13. Evaluation of tools for assessing Web applications. [13]

In this paper, we provide a selected list of tools that can be used for assessing Web application security. Intention is to present available tools that can help in search of vulnerabilities in Web applications and thus make them more secure. The paper is written for Web application developers and testers, and of course, students who have interest in web development. So, it is concentrated on free and/or open source tools, and thereby, commercial products are not mentioned. First part of the paper provides an overview of vulnerabilities that can be detected by using mentioned tools. In the second part, the tools are described and it is shown for which actions these tools can be used.

14. Secure session management and authentication for web sites [14]

The present invention comprises a system and method for secure session management and authentication between web sites and web clients. The method includes both secure and non-secure communication protocols, means for switching between secure and non-secure communication protocols, a session cookie and an authcode cookie. The session cookie is used for session management and the authcode cookie is used for authentication. The session cookie is transmitted using a non-secure communication protocol when the web client accesses a non-secure web page, whereas, the authcode cookie is transmitted using a secure communication protocol when the web client accesses a secure web page. Session management architecture and usage of two distinct cookies along with both secure and non-secure communication protocols prevents unauthorized users from accessing sensitive web client or web site information.

15. A Comparative Usability Study of Two-Factor Authentication[15]

This paper presents an exploratory comparative study of multi-factor authentication techniques. An Online survey of 219 mechanical Turk users was done which aimed at measuring the usability of popular two factor technologies such as: one time security tokens, one time PINs received via SMS and dedicated third party smartphone apps. An analysis was also presented to document a series of attributes/parameters to evaluate the usability of 2FA and also show how ease of use, trustworthiness and required cognitive effort are the three key aspects defining MFA.

III. DEMONSTRATION OF BROKEN AUTHENTICATION

A. Scenario 1: A student management portal restricted to admin users only. (Unauthorised privilege)

This scenario demonstrates improper use of cookies and session management, bad server side validation. In the demonstration, a non admin user can also gain access to the portal by analysing the cookies and manipulating them.

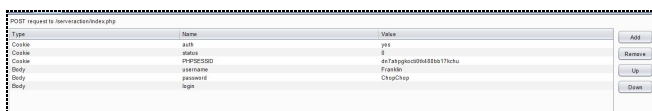


Fig 3.1 Trying to login as a non admin user.



Fig 3.2 An error message saying that the access is restricted to admins only.

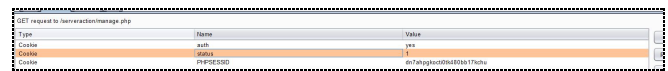
To elevate the user's privileges we will intercept and see how the cookie is defined.



Type	Name	Value
Cookie	auth	yes
Cookie	status	1
Cookie	PHPSESSID	de7ahqpsu08d83ba17tchu
Body	username	Franklin
Body	password	ChopChop
Body	login	login

Fig 3.3 A screenshot of intercepted web parameters in Burp Suite.

Cookie has 3 parameters viz. auth, status and PHPSESSID. PHPSESSID is the session id given by the server. "auth" denotes if the user is being authenticated or has been authenticated. Next is status which must signify user privileges. Upon changing status to 1 from 0 the hacker is able to login as with elevated privileges as an admin.



Type	Name	Value
Cookie	auth	yes
Cookie	status	1
Cookie	PHPSESSID	de7ahqpsu08d83ba17tchu

Fig 3.4 Changing the status parameter in cookie.



Student ID	Name	Action
1	Student 1	Remove
2	Student 2	Remove
3	Student 3	Remove
4	Student 4	Remove

Fig 3.5 The portal is now accessible even by logging in with a non-admin user.

B. Scenario 2: Bypassing login by reusing cookies. (Session Fixation)

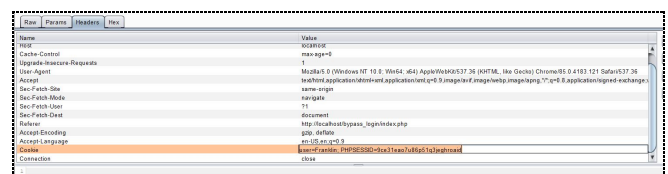
The scenario consists of a banking portal which goes to a welcome page after logging in. In this demonstration we bypass authorisation by reusing an authorised cookie to fixate an ongoing session in another window/computer/browser.



Fig 3.6 When a user tries to go to the welcome page without logging in first they are redirected to the login page with the error message as shown.

Cookie of a user who has been logged in could be accessible if the user forgets to logout of the system and there are no cookie and/or session expiration.

Assuming that a user hasn't logged out of the system and the session is still going on, to get the cookie first we refresh the current page and get the cookie from the intercepted request via Burp Suite.



Name	Value
Host	localhost
Cache-Control	max-age=0
Upgrade-Insecure-Requests	1
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4481.121 Safari/537.36
Accept	text/html,application/javascript,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site	navigate
Sec-Fetch-Mode	document
Sec-Fetch-User	no
Sec-Fetch-Dest	http://localhost/pass_login.php
Referrer	gdp_dafoke
Accept-Encoding	gzip, deflate
Accept-Language	en-US,en;q=0.8
Cookie	auth=de7ahqpsu08d83ba17tchu;status=1;PHPSESSID=de7ahqpsu08d83ba17tchu
Connection	close

Fig 3.7 intercepting the HTTP request and copying the whole cookie.

Now to fixate the session in any other browser/computer we will try to access the welcome page. But it will give the previously mentioned error and ask for authentication. Therefore we will intercept the request while going to the welcome page and then paste the intercepted cookie in this session.

process will remain secure even if one of the authentication factors is compromised.

More checks means more security, but customers can find them tedious, so it's wise to choose the number of extra checks based on both the minimum security level suitable for that service or account, and the frequency and proximity to other verification checks.

When choosing which method to use, it is vital to take into account things like the level of security needed, technologies used by the website to interact with users, cost, etc.

Multi Factor authentication can also be used to improve security in smart cars, smart homes and airports. Smart cars can be protected by combining a facial or fingerprint identification with a gesture recognition system. It does not only protect the vehicle from theft, but allows for easy authentication of multiple drivers.

B. Session Timeouts

The current session must timeout after a specific period of inactivity. For example vtop logs out automatically after 15 minutes of inactivity. This makes sure that if an user has forgotten to logout the system and has left, another authorised user cannot continue the same session. Session must be unique to each user. Therefore if the user in session is not authorised, it becomes a case of broken authentication.

The inactivity period should not be too long that it gives window to an authorised access and at the same time it shouldn't be too small that a user is logged out repeatedly before he can perform another action.

C. Session lock upon failure in authentication

A user will be given a specific number of chances to authenticate themselves. Say 3 attempts are provided to the user to authenticate via password and username or a security question or any other form like secret codes, TOTP etc, upon exceeding the number of allowed failed attempts the user will be locked out for a defined period of time.

The time to lockout has been suggested to be at least 10 minutes. Users must not be locked out for a very small time because that will make the system vulnerable to brute force attacks, and the lock out period must not be too long as a malicious user might cause DoS (Denial of Service) attack.

D. Implementation of End-to-end encryption by SSL

By implementing SSL we can ensure that all the data that goes through the network is encrypted and undecipherable by any hacker. This makes sure the integrity and confidentiality of cookies and other parameters passed through the network.

A hacker might use a network sniffing tool to capture the requests and HTTP parameters but if they are encrypted by SSL it is close to impossible to manipulate the session and

servers as demonstrated in the broken authentication scenarios.

V. SYSTEM ARCHITECTURE

The components that form the two factor authentication are:

1. Security Question
2. Image based Authentication (click based graphical password)

Overall the whole connection will be secured end to end by an SSL certificate which enables the https protocol.

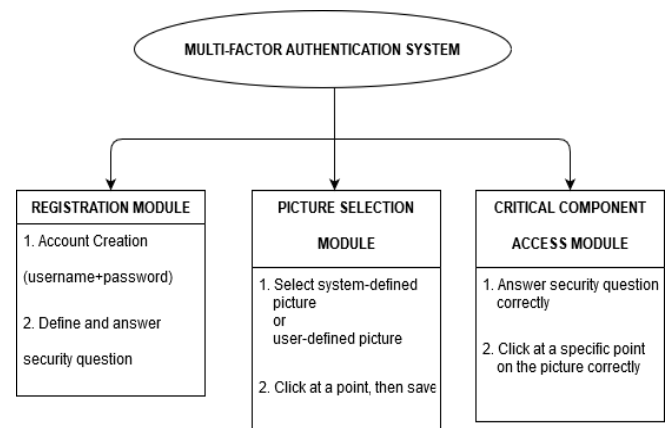


Fig 5.1 System architecture chart.

VI. PROCESS DIAGRAM

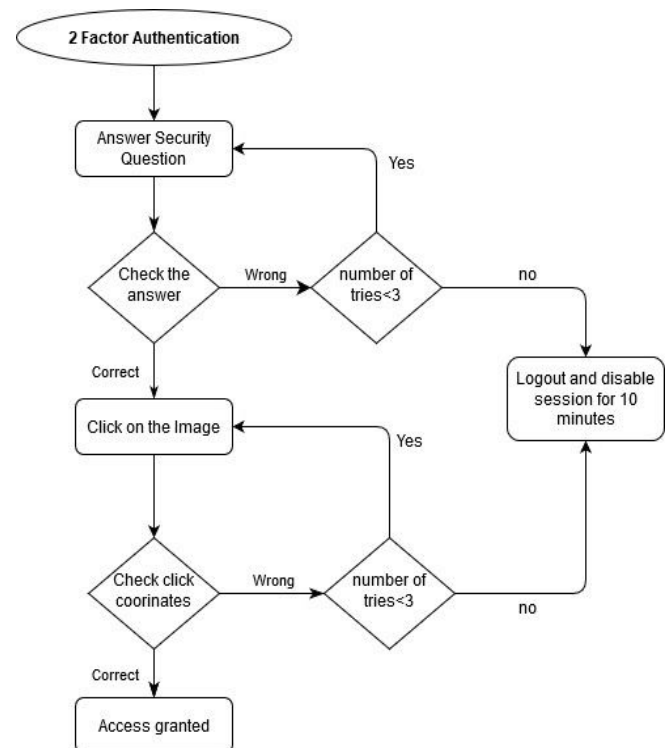
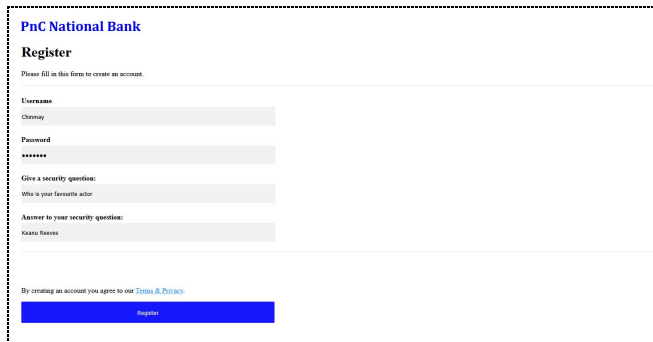


Fig 6.1 *Process flowchart.*

VII. IMPLEMENTATION

First we register for the website.

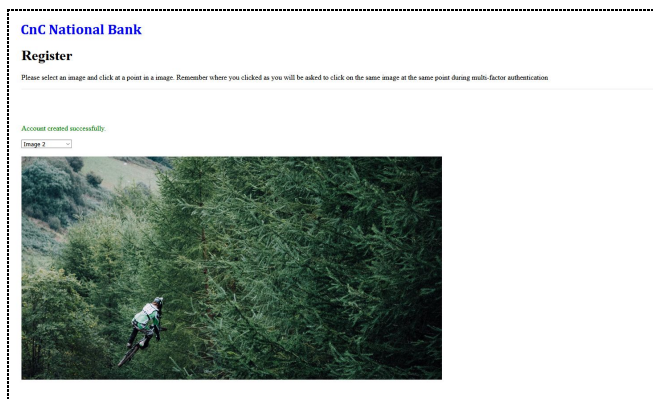
When we register we need to give a security question and a corresponding answer which will be used as a second factor of authentication.



The registration page for PnC National Bank. It includes fields for Username (Chinmay), Password (*****), a security question (Who is your favourite actor), and an answer to the security question (Keanu Reeves). There is a checkbox for 'I'm not a robot' and a 'Register' button. A message at the top says 'Account created successfully, login to continue'. At the bottom, there is a link to 'New user? Register here'.

Fig 7.1 *Registration page.*

After we click register, the user is asked to set their third factor authentication by clicking on a part of the image. We give the user option between using 3 images, whichever they find easier to remember. The place where they click has an absolute coordinate in reference to the image and that coordinates are saved in the database.



The image selection screen for 3 factor authentication. It shows a large image of a person climbing a tree. A small red dot indicates the selected point. A message at the top says 'Account created successfully'. Below the image, there is a dropdown menu for 'Image 2' and a 'Submit' button.

Fig 7.2 *The user is asked to register the image and a point on the image for 3 factor authentication.*

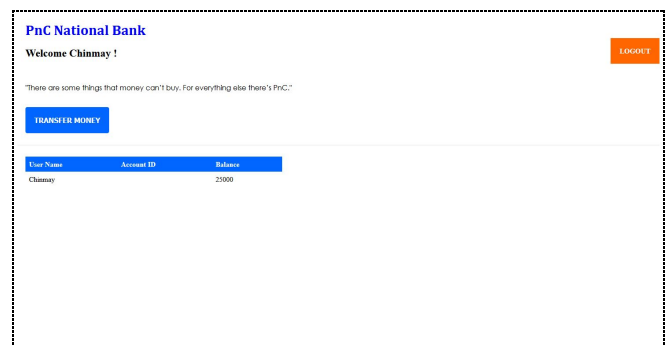
Once we click on the image at the desired point and the coordinates are secretly stored in the database and then we are redirected to the login page. We put our credentials and then verify reCaptcha that makes the website immune to brute force attacks.



The login page for PnC National Bank. It includes fields for Username (Chinmay) and Password (*****). There is a checkbox for 'I'm not a robot' and a 'Login' button. A message at the top says 'Account created successfully, login to continue'. At the bottom, there is a link to 'New user? Register here'.

Fig 7.3 *Login page.*

After login we land on the welcome page where we can see our user details and an option to transfer money.

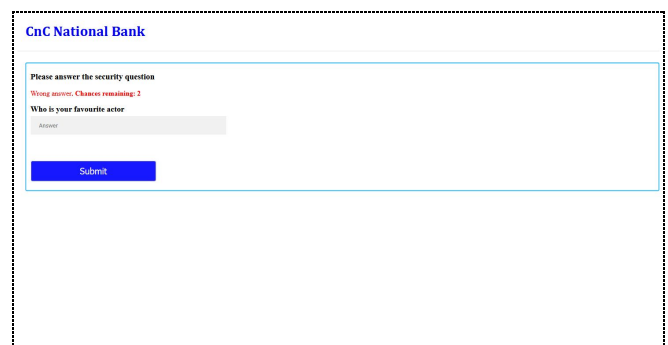


The welcome page for PnC National Bank. It shows the user's name (Chinmay) and account ID (25000). There is a 'TRANSFER MONEY' button. A message at the top says 'Welcome Chinmay !'. At the bottom, there is a link to 'Logout'.

Fig 7.4 *Welcome page.*

When we click on transfer money we are asked to authenticate ourselves again because it is a critical action.

The user is given maximum 3 attempts to authenticate. When they put a wrong answer they get an error message as follows



The second factor of authentication page. It shows the security question (Who is your favourite actor) and the answer (Keanu). There is a 'Submit' button. A message at the top says 'Please answer the security question'. Below the question, there is a message 'Wrong answer. Chances remaining: 2'.

Fig 7.5 *Second factor of authentication.*

Upon exhausting all the attempts the **user is locked for 10 minutes.**

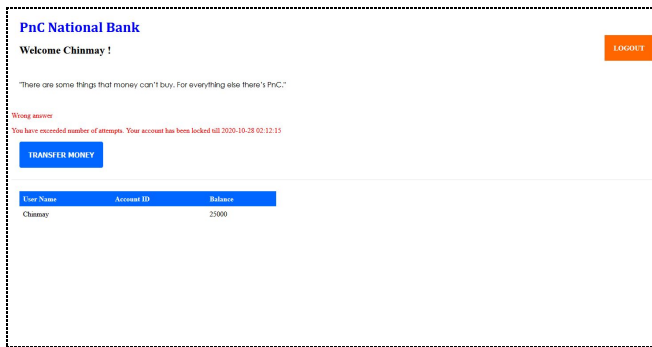


Fig 7.6 An error message showing the user has exhausted all attempts and is not allowed to try the same till a period of time.

If we try to transfer money when we are locked out we get the following message saying that the account has been blocked till what time

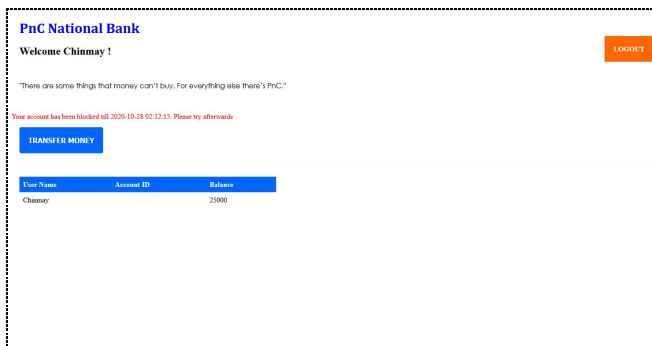


Fig 7.7 Trying to access transfer money action while the user is locked out.

If we give the right answer to the security question the transfer process proceeds and the user is asked for the account details of the recipient.

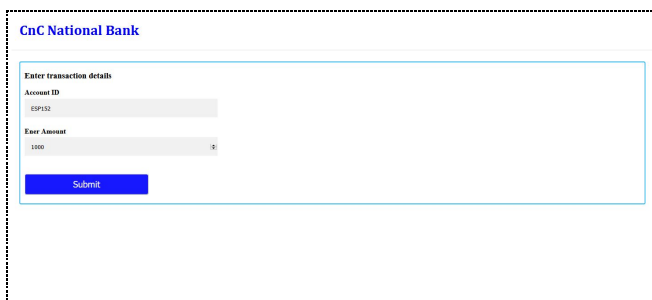


Fig 7.8 Transfer process proceeding.

Upon submitting this detail, the third factor of authentication is implemented. The user is asked to verify authentication by clicking at the secret point on the image to verify again. Similarly on this page also the user is allowed only 3 attempts and if fails the user is locked out and a notification email goes to the user.

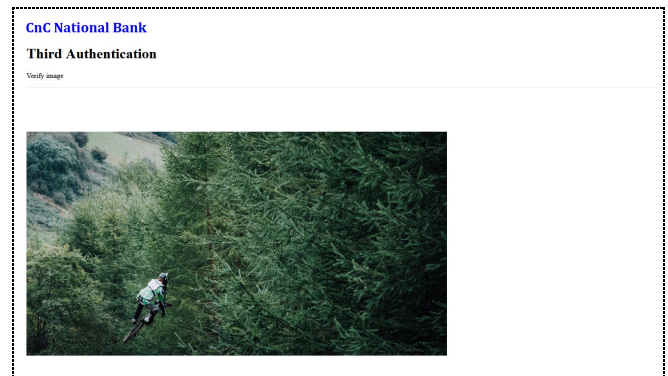


Fig 7.9 User asked for the third factor of authentication.

Since the user cannot remember the exact point, to verify we check if the difference between the click of the user and the coordinates we saved should be less than 10. So this allows the user to click anywhere inside a small box of 10x10 coordinates which is not big enough so that anyone can guess and also not small enough that a user has a difficult time.

Once we click anywhere inside that 10X10 box, we see a successful message, otherwise the similar warning message as before.



Fig 7.10 A success message upon clicking at the right position.

VIII. CONCLUSION

After demonstrating the vulnerabilities that give rise to broken authentication we went ahead and implemented three factor authentication (username-password, security questions and image based authentication), which is a very prominent remedial method for broken authentication. A banking web application was selected to demonstrate the effectiveness of multi factor authentication. The second step involving image based authentication was a critical step as it is a unique and highly dependable method for authentication. And on top of that, an SSL certificate provided end to end data encryption over the network.

A. Advantages

Graphical passwords have been designed to make passwords more memorable and easier for people to use. Psychology

studies have also revealed that the human brain is better at recognizing and recalling images than text. Pictures are generally easier to be remembered or recognized than text, especially photos, which are even easier to be remembered than random pictures.

Image based authentication systems allow us to create passwords that are resistant to guessing, dictionary attack, keyloggers, and social engineering.

Hence even if the security question phase is compromised, the image authentication step can be trusted. The only attack it is vulnerable towards is shoulder surfing. We intend to tackle this issue by using a cascade of clickable images instead of a single one. (explained in future work).

Moreover the implementation includes a session timeout feature that ends the session and logs out automatically if the user has been inactive for more than 15 minutes (the time after which the user should be logged out could be changed as per need). This prevents attacks similar to the one demonstrated in scenario B.

Use of SSL ensures that all the requests sent to the server are properly encrypted and that it is not possible to decipher/understand/manipulate the traffic if accessed with the help of any network traffic sniffer.

B. Future work

Regenerating the session key after initial authentication can also be implemented. This will cause the session key to change immediately after authentication. So even if the attacker knows the initial session ID, it becomes useless before it can be used.

A sequence of images and a click pattern can also be configured during registration. To add a layer of security, we ask users to assign a sequence number for each image used during the registration phase. The user has to reproduce the same sequence during his login phase.

During authentication, the next image displayed is always based on the location of the previously entered click-point, creating a path through an image set. Thus a wrong click leads to an incorrect path, with an explicit indication of authentication failure only after the final click.

REFERENCES

- [1] Aldwairi, Monther & Aldhanhani, Saoud. (2017). Multi-Factor Authentication System.
- [2] Ometov, Aleksandr & Bezzateev, Sergey & Mäkitalo, Niko & Andreev, Sergey & Mikkonen, Tommi & Koucheryav, Yevgeni. (2018). Multi-Factor Authentication: A Survey. *Cryptography*. 2. 10.3390/cryptography2010001.
- [3] Phan, Kim, "Implementing Resiliency of Adaptive Multi-Factor Authentication Systems" (2018). *Culminating Projects in Information Assurance*. 65.
- [4] Tamara, Mrs & Mohamed, Tamara. (2019). SECURITY OF MULTIFACTOR AUTHENTICATION MODEL TO IMPROVE AUTHENTICATION SYSTEMS.
- [5] D. Huluka and O. Popov, "Root cause analysis of session management and broken authentication vulnerabilities," World Congress on Internet Security (WorldCIS-2012), Guelph, ON, 2012, pp. 82-86.
- [6] Hassan, Md Maruf, et al. "Broken authentication and session management vulnerability: a case study of web application." *International Journal of Simulation Systems, Science & Technology* 19.2 (2018): 6-1.
- [7] Mainka, C., Mladenov, V., Guenther, T. & Schwenk, J., (2015). Automatic recognition, processing and attacking of single sign-on protocols with burp suite. In: Hühnlein, D., Roßnagel, H., Kuhlisch, R. & Ziesing, J. (Hrsg.), Open Identity Summit 2015. Bonn: Gesellschaft für Informatik e.V.. (S. 117-131).
- [8] Kim, Joseph. *Burp Suite: Automating Web Vulnerability Scanning*. Diss. Utica College, 2020.
- [9] Pauli, Josh. (2013). Web Application Exploitation with Broken Authentication and Path Traversal. 10.1016/B978-0-12-416600-4.00005-8.
- [10] Nagpal, Bharti & Professor, Asstt & Chauhan, Naresh & Singh, Nanhay & Sharma, Pratima. (2019). Preventive Measures for Securing Web Applications Using Broken Authentication and Session Management Attacks: A Study.
- [11] Nam, Sang-On, et al. "Authentication and Session Management based on Ajax." *Journal of Internet Computing and Services* 7.6 (2006): 157-174.
- [12] Sedek, Khairul Anwar, et al. "Developing a Secure Web Application Using OWASP Guidelines." *Computer and Information Science* 2.4 (2009): 137-143.
- [13] Suhina, Vanja, Mario Kozina, and Stjepan Groš. "Evaluation of tools for assessing Web applications." *30th Jubilee International Convention MIPRO 2007*. 2007.
- [14] Kou, Wei Dong, Lev Mirlas, and Yan Chun Zhao. "Secure session management and authentication for web sites." U.S. Patent No. 7,216,236. 8 May 2007.
- [15] De Cristofaro, Emiliano & Du, Honglu & Freudiger, Julien & Norcie, Greg. (2013). Two-Factor or not Two-Factor? A Comparative Usability Study of Two-Factor Authentication. USEC. 10.14722/usec.2014.23025.
- [16] <https://www.sitelock.com/blog/owasp-top-10-broken-authentication-session-management/>
- [17] <https://hdivsecurity.com/owasp-broken-authentication#:~:text=Where%20possible%2C%20implement%20multi%2Dfactor,credentials%2C%20particularly%20for%20admin%20users.>
- [18] <https://www.geeksforgeeks.org/broken-authentication-vulnerability/>
- [19] <https://techbeacon.com/security/8-reasons-you-should-turn-multi-factor-authentication>
- [20] <https://www.welivesecurity.com/2019/04/10/credential-stuffing-g-attacks-login/>