**Group Name: Innoventive**

**Module: Application Development Security (APDS7311)**

**Lecturer: Issac Leshaba**

**04 September 2024**

**Part 1**

**Blessing Adefila ST10041144**

**Phuti Motimele ST10191686**

**Muhammad Bhula ST10073908**

**Lesego Ramosa ST10209476**

**Rebotile Molala ST10185742**

Table of Contents

[1a. SECURING USER INPUT 3](#_Toc176290492)

[Hashing 3](#_Toc176290493)

[Salting 3](#_Toc176290494)

[How the data will be secured 4](#_Toc176290495)

[1b. SECURING DATA IN TRANSIT 4](#_Toc176290496)

[1c.HARDENING PORTAL 6](#_Toc176290497)

[i.Session Jacking 6](#_Toc176290498)

[ii. Clickjacking 9](#_Toc176290499)

[iii. SQL Injection Attacks 11](#_Toc176290500)

[iv.How to prevent Cross Site Scripting Attacks 12](#_Toc176290501)

[v. “Man in the Middle “ attacks (MitM) 13](#_Toc176290502)

[vi. DDos Attacks 15](#_Toc176290503)

[2a. MobSF 17](#_Toc176290504)

[Report regarding the use of MobSF 17](#_Toc176290505)

[Overview 18](#_Toc176290506)

[Support for using MobSF 18](#_Toc176290507)

[2b. ScoutSuite 20](#_Toc176290508)

[Report regarding the use of ScoutSuite 20](#_Toc176290509)

[Overview of ScoutSuite 20](#_Toc176290510)

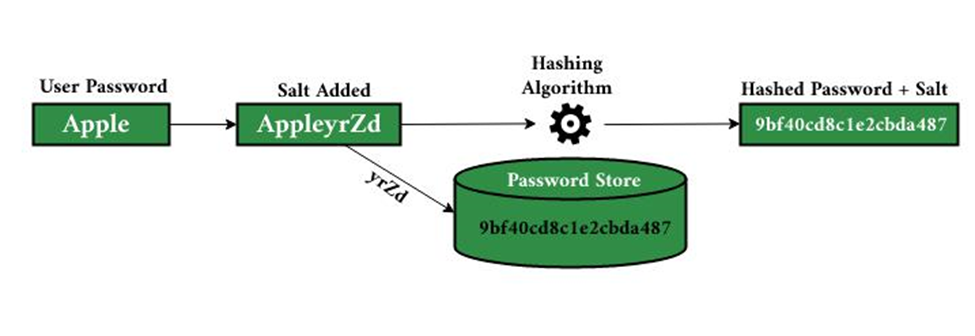
[The key components include: 21](#_Toc176290511)

[Support for using ScoutSuite 21](#_Toc176290512)

[Implementation 21](#_Toc176290513)

[REFERENCES 24](#_Toc176290514)

# 1a. SECURING USER INPUT



(GeeksforGeeks, 2022)

To secure sensitive customer information, particularly during the registration and login processes, we'll use a combination of hashing and salting techniques.

## Hashing

Hashing is the process of converting input data (such as a password) into a set number of characters, usually a hash code. A crucial aspect of a hash function is its one-way nature: once data is hashed, it's difficult to reverse or decode to uncover the original input. This is important for security as even if a hacker obtains the hashed information, they are unable to recover the initial passwords. Hash algorithms like SHA-256 are created to generate distinct results for slightly varying inputs, guaranteeing that even minor alterations in the information create completely different hash values (Nihe, 2018 ).

## Salting

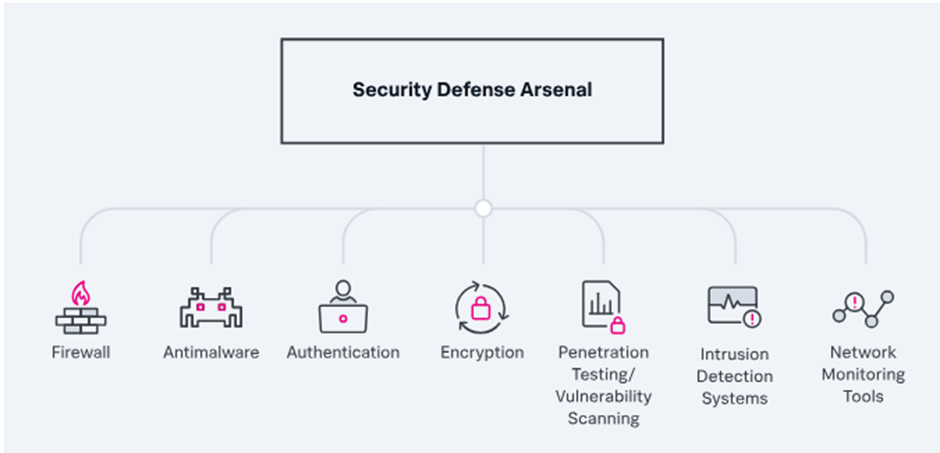
Adding salt to the hashing process provides an extra level of security. A salt, which is a random sequence of characters, is included in the input data (e.g., a password) before it is hashed. Every password is assigned a distinct salt, which is saved in the database together with the hashed password. Salting is done to make sure that if two users have the same password, their hashes will be different because of the individual salts. This stops hackers from using pre-calculated hash tables (rainbow tables) to decrypt passwords, because the salt makes the attacker create hashes for every possible salt and password mix. (Yasar, 2024)

## How the data will be secured

After a customer signs up on the platform, their password is mixed with a one-of-a-kind salt before being hashed. Both the hash and the salt are saved in the database. While logging in, the system gets the saved salt and mixes it with the password provided by the user. After hashing the combined value once more, the resulting hash is then checked against the one saved in the database. When the hashes correspond, the login is considered successful.

This technique guarantees that, in the event of a breach where an intruder obtains the hashed passwords and salts stored in the database, they will find it difficult to uncover the actual passwords. Having a different salt for each password ensures that even if passwords are the same, their hashes will be different, effectively guarding against different types of attacks like brute force and rainbow table attacks (Nihe, 2018 ).

# 1b. SECURING DATA IN TRANSIT



Secure Data in Transit (Raza, 2023).

1. **TLS Encryption**

* TLS encryption should be employed on all data sent between the client's browser and the bank's servers. It would work in ensuring encrypted information, including login credentials, account numbers, and payment details, is not intercepted or read by other parties.

(Checkout, 2024)

1. **Virtual Private Network (VPN):**

* Use VPNs for internal communication, especially between internal systems of the bank and the SWIFT network, by adding an extra layer of encryption and hence security so that such communications could hardly be intercepted or tampered with by any attacker.

(Checkout, 2024)

1. **Strong Authentication Mechanisms:**

* Strong methods of authenticating customers and employees should be used, such as Multi-Factor Authentication. It adds a layer of security since several verifications have to be made before access is allowed into the system or approvals of transactions are made (Checkout, 2024).

1. **End to End Encryption:**

* Consider using end-to-end encryption for very sensitive transactions that ensure the information is encrypted right from sender to receiver; this reduces interception or even data tampering.  
   (checkout 2024)

1. **Checks on Data Integrity:**

* Employ hash algorithms like SHA - 256, which create checksums over data in transit. The thing is, no alteration should happen during transit, and it also allows verification of the integrity of the data following the receipt.  
    (Checkout, 2024)

1. **Secure APIs:**

* Development of the system in a way to serve data between different components with the use of TLS-secured APIs is crucial. API gateways can be used to enforce authentication as well as rate limiting, log traffic for monitoring, and protection.  
   (Checkout, 2024 )

1. **Whitelisting and Blacklisting:**

* Whitelisting IP addresses means allowing only a few numbers of IP addresses with access to view the sensitive systems. The blacklisting of IP blocks malicious known IP addresses from gaining access to the system.  
   (Checkout, 2024)

1. **Have Regular Security Updates and Patch Management:**

* Keep all systems, software, and protocols updated with the latest security patches and updates to protect against known vulnerabilities and exploits.

1. **Monitoring and Intrusion Detection Systems:**

* Deploy monitoring and intrusion detection systems that monitor and detect unusual or suspicious activity in real time. Thus, it helps find the potential threats as quickly as possible.  
   (Checkout, 2024)

# 1c.HARDENING PORTAL

## i.Session Jacking

Session Jacking takes place when a cyber attacker hijacks a user’s session by stealing the token of the session that was created for the user. (Trevino, 2024)

Figure:1 


Illustration of session hijacking using (invicti, 2019)

The ways in which this can be mitigated are listed and explained below:

**1. HTTPS Everywhere:**

A key way to secure a user’s session from being hijacked is by making each page of the banking website HTTPS. This helps to secure the session tokens through encryption when transmission takes place, making it more difficult for attackers to intercept the user’s session. (Trevino, 2024)

**2. Secure Session Cookies:**

Session cookies are another way in which attackers can gain access to a user’s session. The way to prevent this is by securing the flag on session cookies, which ensures that they are always transmitting over HTTPS. (Conrad, 2023)

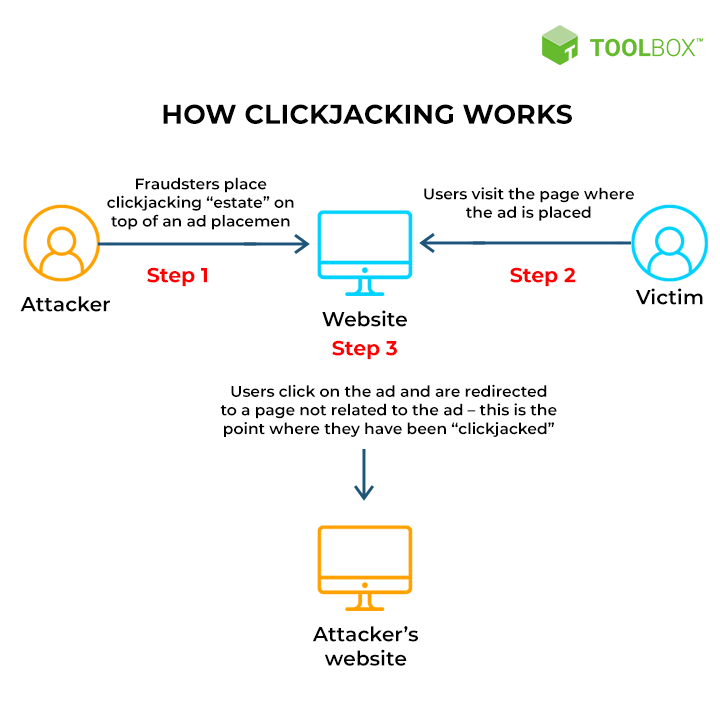
**3. Session Timeout:**

Timeouts are a simple yet effective way to prevent session jacking. This is key for a website such as a banking site where if a user is not active for a period of time on the site the system will automatically log them out and require them to login again. This safety measure limits the amount of time a session is running for a hijacker to intercept it. (Conrad, 2023)

**4. Session Regeneration:**

Regenerating the session ID after the user logs in as well as while they are using the portal is another way to help prevent the session from being hijacked. The attacker will need the ID of the session to be able to hijack it. However, if the session ID periodically changes, by the time the attacker gains the session ID it will not be valid anymore. (Trevino, 2024).

## ii**. Clickjacking**



Clickjacking is a process where the cyber attacker tricks the user into clicking on a “fake” page disguised as a legitimate site page often through the use of a transparent frame overlay. (Katz, 2023).

**The following ways listed and explained below will help with clickjacking:**

**1. Content Security Policy (CSP)**

CSP will help in minimizing the data sources which are allowed by the website. CSP will review any unknown requests that are transmitted by attackers. CSP provides a flexible way to protect the banking portal from being framed. (Katz, 2023)

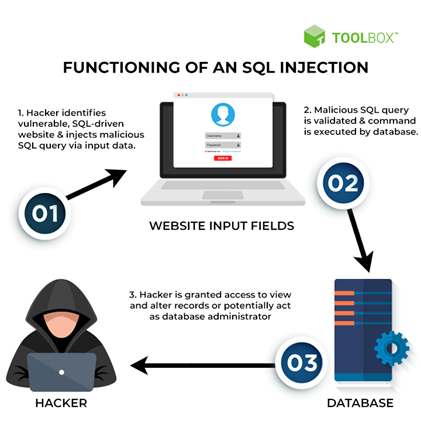
**2. Frame Busting Scripts:**

This technique is Implemented in the JavaScript code. It will help detect if the webpage is being loaded inside an iframe and force it to break out, keeping the web pages secure from iframe overlays. (Katz, 2023)

**3. User Interface (UI) Design:**

The UI needs to be designed in a way that minimizes the impact of potential clickjacking attacks. This can be done for the web pages which will require the user to input sensitive information. The payment gateway is an example of this, the UI should include pop-up confirmation dialogs before the user proceeds. (Katz, 2023)

## iii. **SQL Injection Attacks**



What is an SQL Injection (BasuMallick, 2022)

An SQL Injection (SQLi) attack is when a web security vulnerability which allows an attacker to interfere with the queries that an application makes to the database. This allows an attacker to be able to view data that they aren’t normally able to retrieve. If an attacker has access to this data, they’re able to modify or delete data which causes persistent changes to the application’s behaviour or content. (Portswigger).

**How to harden against SQL Injection Attacks:**

**Training and Maintaining Awareness:**

To keep your application safe, everyone in the organization should be aware of the risks with SQL Injections. There should be suitable security training provided to all the developers. (Acunetix).

**Use of ORM Frameworks:**

This framework is a software layer that sits between your application and the database, which provides a way to be able to manipulate the data without writing SQL queries. ORMs can assist with preventing SQL Injection attacks by sanitizing input data automatically and generate secure SQL queries. (Martinelli, et al., 2024).

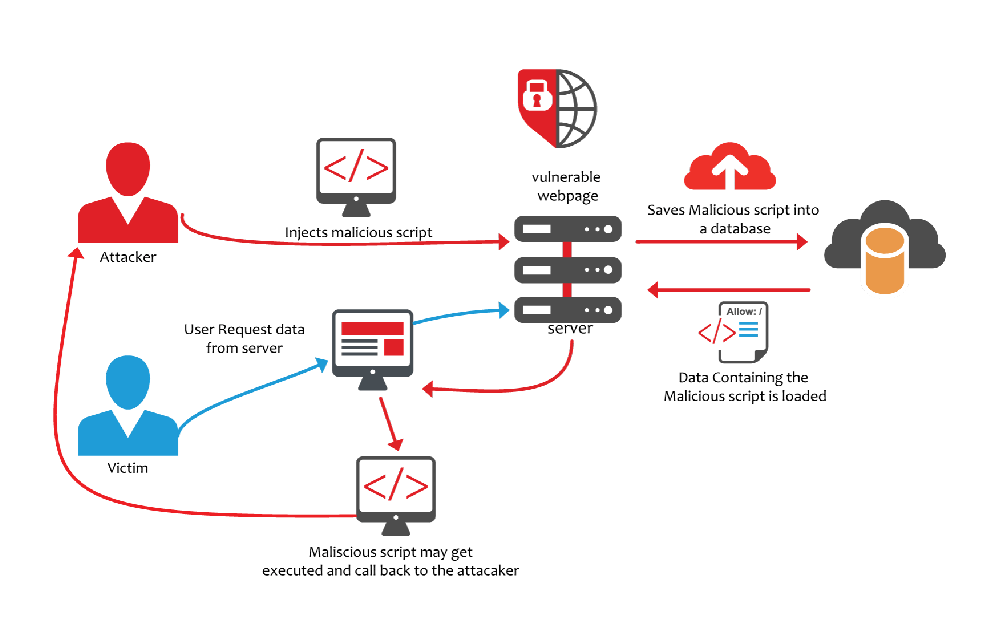
**Input Validation and Parameterized Queries:**

This is when you structure the application code so that it should never use the input directly. All inputs should be sanitized, and developers must remove any potential malicious code elements like single quotes. (Acunetix).

**Use of Web Applications Firewall(WAF):**

This is a security tool which monitors and controls traffic between your application and the internet. It works by applying rules which are known as policies, to the incoming and outgoing requests and responses, which can detect and prevent various types of attacks (LinkedIn, 2024).

## **iv.How to prevent Cross Site Scripting Attacks**



Research Gate XSS attacks (Mohaty, 2021)

* Avoid HTML code in inputs in order to prevent users from posting malicious code. This simply means preventing HTML injection requires a combination of enhanced secure coding practices and the implementation of appropriate security controls. An example of a technique that could be used is an input sanitization which is simply making sure all the user’s inputs is sanitized, basically remembering HTML tags and special characters before rendering them in web pages. (Timonera, 2023)
* Confirm all the user’s inputs to ensure that they meet the specific criteria. Using whitelisting in order to verify the user’s inputs are only explicitly allowed characters, tags or formats, as this is more secure than blacklisting specific characters or patterns. (Timonera, 2023)
* The use of output encoding is essential to protect against XSS attacks. It is important that before displaying user-generated content on a web page, ensure that it is encoded accordingly. Same applies to the URL encoding, ensuring that the user inputs included in URLs are encoded accordingly to prevent injection attacks. (PortSwigger, 2023)
* Implementing a web application firewall helps with preventing XSS attacks. The WAFs can filter bots as well as other malicious activity that may indicate an attack and they can immediately block attacks before any script is executed and before it compromises the network. (PortSwigger, 2023)
* Regular security audits and testing could help detect potential vulnerabilities, using automated tools to identify weak spots. (Timonera, 2023)
* Implement a secure error handling mechanism that does not display a detailed error message to the users that may reveal sensitive information which may cause XSS attacks. (Timonera, 2023)

## **v. “Man in the Middle “ attacks (MitM)**

A man-in-the-middle (MITM) attack happens when a malicious actor covertly intercepts communication between two parties. This occurs commonly when hackers gain unauthorized access to a Wi-Fi network or router, or when they pretend to be someone else by imitating a website in order to spy, alter, or pilfer data.



Man in the Middle Attack (Magnusson, A. 2024)

1. **Use HTTPS (TLS/SSL):**

Make sure that the communication between the client and server is encrypted with HTTPS. TLS/SSL certificates are utilized on the server to accomplish this. HTTPS encrypts the information exchanged between the browser and server, making it challenging for attackers to intercept or modify the data. (Malik, 2024)

**2. Strong Authentication Mechanisms Implementation:**

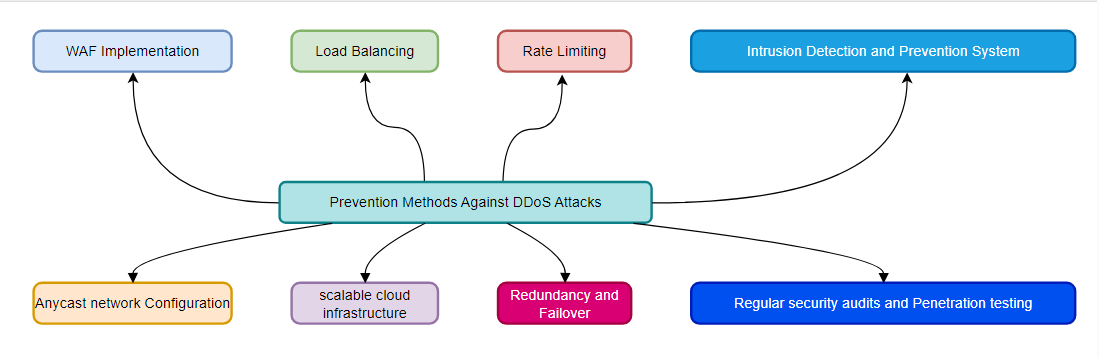
Utilize multi-factor authentication (MFA) for an additional level of security in addition to passwords. If a hacker somehow obtains the password, they will still require the second factor (such as a code sent to a phone) to access the account. Enforce strong password regulations, including the use of intricate passwords and regularly prompting users to update them (Malik, 2024).

**3. Use Secure Cookies**

Establish Secure attribute for cookies to exclusively transmit them over HTTPS and utilize HttpOnly attribute to block access to cookies through JavaScript, reducing the risk of XSS attacks and potential MitM attacks. Furthermore, enable the SameSite attribute on cookies to avoid them being included in cross-site requests, decreasing the likelihood of cross-site request forgery (CSRF) (Malik, 2024).

**4.Certificate Pinning**

Certificate pinning is the act of connecting a server with a particular certificate or public key. Upon connecting to the server, the client verifies the certificate with a pre-approved value. This prevents potential Man-in-the-Middle attacks, where a hacker could exploit a compromised or unauthorized certificate (Malik, 2024)

vi. DDos Attacks 

Preventing DDOS Attacks (Radware, 2024)

**1. WAF Implementation:**

* WAF shall filter and monitor the HTTP traffic between the internet and the payment system. Therefore, it can detect malicious traffic patterns that may show a DDoS attack, such as abnormally high requests originating from the same IP address.
* It also helps in blocking the common DDoS attack methods against the application, which may include SQL injection and cross-site scripting. (SmartCom, 2024)

**2. Load Balancing and Rate Limiting:**

* Load Balancers: Utilize load balancers that distribute the incoming traffic across multiple servers so that no single server gets overwhelmed.
* Rate Limiting: Establish a rate limit to restrict the number of requests from a single IP address in a predetermined amount of time. This can help in dealing with those DDoS attacks that rely on the overloading of a server by many requests (Haproxy, 2024).

**3. Intrusion Detection and Prevention Systems (IDPS):**

* Deploy IDPS to monitor network traffic for unusual patterns that could signal a DDoS attack. This system will automatically trigger responses from blocking malicious IP addresses to triggering alerts for security teams for further investigation (Rapid7, 2024).

**4. Anycast Network Configuration:**

* This could utilize anycast networking for both DNS and content delivery, enabling various locations around the world to advertise the same IP address. This aids in the distribution of the load and minimizes the effect a DDoS attack has on any location (CloudFare, 2024)

**5. Scalable Cloud Infrastructure:**

* Leverage infrastructure in the cloud that can scale up or down resources depending on demand. Additional servers can be mobilized to absorb the additional traffic during an attack without any impact on the availability of the service (Nops, 2024).

**6. Redundancy and Failover:**

* Make sure critical parts of the system, like authentication and payment processing servers, are duplicated and provide failover mechanisms. This is to say, when one server gets overloaded or falls down, the traffic can get rerouted to another server without disruption of service (Weaver, 2023).

**7. Regular Security Audit and Penetration Testing**

* Issues such as regular security audits and penetration testing are necessary to identify system vulnerabilities. This proactive approach helps fix problems before vulnerabilities can be used for a DDoS attack (Henretta, 2024).

**8. Monitoring and Alerting Systems:**

* Real-time monitoring needs to be done to track patterns of traffic that could depict anomalies. Automated alerts will trigger potential DDoS attacks to be taken notice of by the IT personnel to respond on time.
* These tools include SIEM systems that can correlate different varieties in order to find and respond to threats.(Bramhe, 2024)

# 2a. MobSF

## Report regarding the use of MobSF

Mobile Security Framework is an open-source security tool that provides progressive security analysis for Ios, Android as well as windows mobile applications. The tool is designed to identify potential vulnerabilities in mobile applications by analysing source code (APK, IPA) as well as other mobile application components. This security framework allows dynamic analysis as well as malware analysis which makes it a versatile tool for ensuring the security of the mobile apps throughout the development lifecycle.

## Overview

MobSF is a powerful, open-source framework designed to perform security assessments on mobile apps. MobSF offers a dynamic analysis which provides comprehensive insights into potential vulnerabilities, privacy issues, and the code quality. It also offers a static analysis.

When using the static analysis results MobSF identified insecure storage of sensitive data, for instance user credentials in shared preferences meaning they were not encrypted. As a result, this could expose sensitive information if the device is compromised. There could be unsecured API endpoints, which may expose user data to potential interception and manipulation. The apps code was not obfuscated which results in the increase of risk of reserve engineering, making it easier for attackers to analyse the apps logic and exploit any weaknesses.

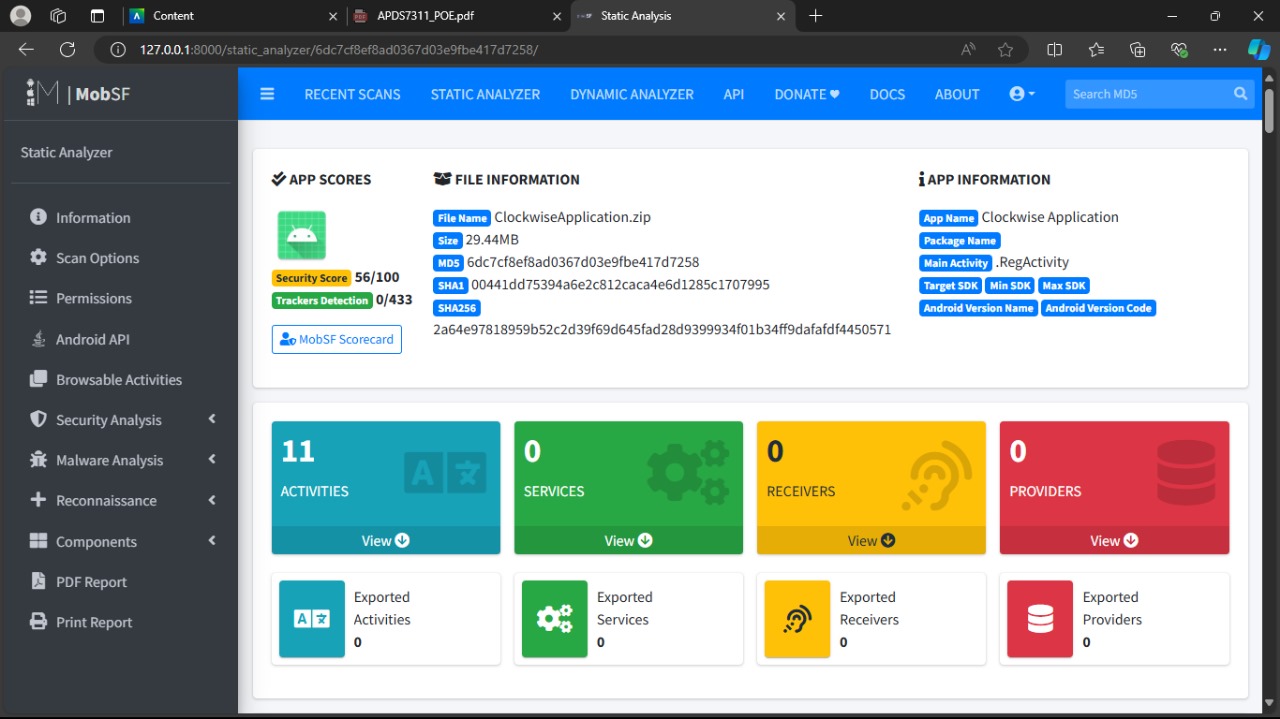
When using the dynamic analysis, it is found that the app was found to log sensitive information during runtime, which could be accessed by unauthorized users if logs are not securely managed. (Muthii, 2023)

## Support for using MobSF

The reasons that support the use of MobSF are that MobSF has the ability to perform both static and dynamic analysis which ensures that all aspects of the applications security are covered from code vulnerabilities to the runtime behaviour. The MobSF tool automates the detection of security issues which helps with reducing the time as well as the effort required for manual code reviews and also provides developers with immediate feedback. The tool generates actionable reports, which offer a clear guidance on how to address identified vulnerabilities. Lastly the tool consists of an open-source nature which means it is continuously updated by a community of developers, making sure that it remains relevant in identifying the latest security threats.

Using MobSF in analysing the time tracking app has demonstrated its effectiveness, especially in identifying critical vulnerabilities, supporting its adoption as a standard tool for mobile application security assessments. The MobSF tool has the ability to provide a comprehensive and automated analysis which aligns well with the organizations needs for efficiency as well as reliability for security practices. I would recommend the tool to organizations for them to implement it as part of mobile apps security toolkit, as ti will significantly improve the ability to detect and mitigate security risks, ensuring the protection of both users and the infrastructure. (Singh, 2018)

In Conclusion MobSF is considered a powerful mobile app security testing tool which offers significant benefits for ensuring the security of the organization’s mobile apps. It enhances the overall security posture of the mobile applications.

****

This is an image of a summary of how an MobSF was used to test an OPSC application in order to see how secured is that mobile application.

A screenshot of a computer

Description automatically generated

This is the code analysis of the application. This is just a summary of the analysis.

# 2b. ScoutSuite

## Report regarding the use of ScoutSuite

Scout Suite is regarded as an open-source multi-cloud security auditing tool which is used to provide security posture assessment across AWS, Azure, Google Cloud and other cloud providers. The tool works by automatically aggregating configuration data for an environment and applying rules to audit the environment.

## Overview of ScoutSuite

The basic tasks of a ScoutSuite functions by accessing cloud environments with read-only credentials, collecting configuration data and analysing it against known best practices and security standards. Main functionality of the tool is that it generates a detailed report highlighting issues such as insecure configurations, excessive, lack of encryption and other security concerns. The presentation of the report in HTML format is user-friendly which enables security teams to promptly discern and prioritize remediation tasks.

## The key components include:

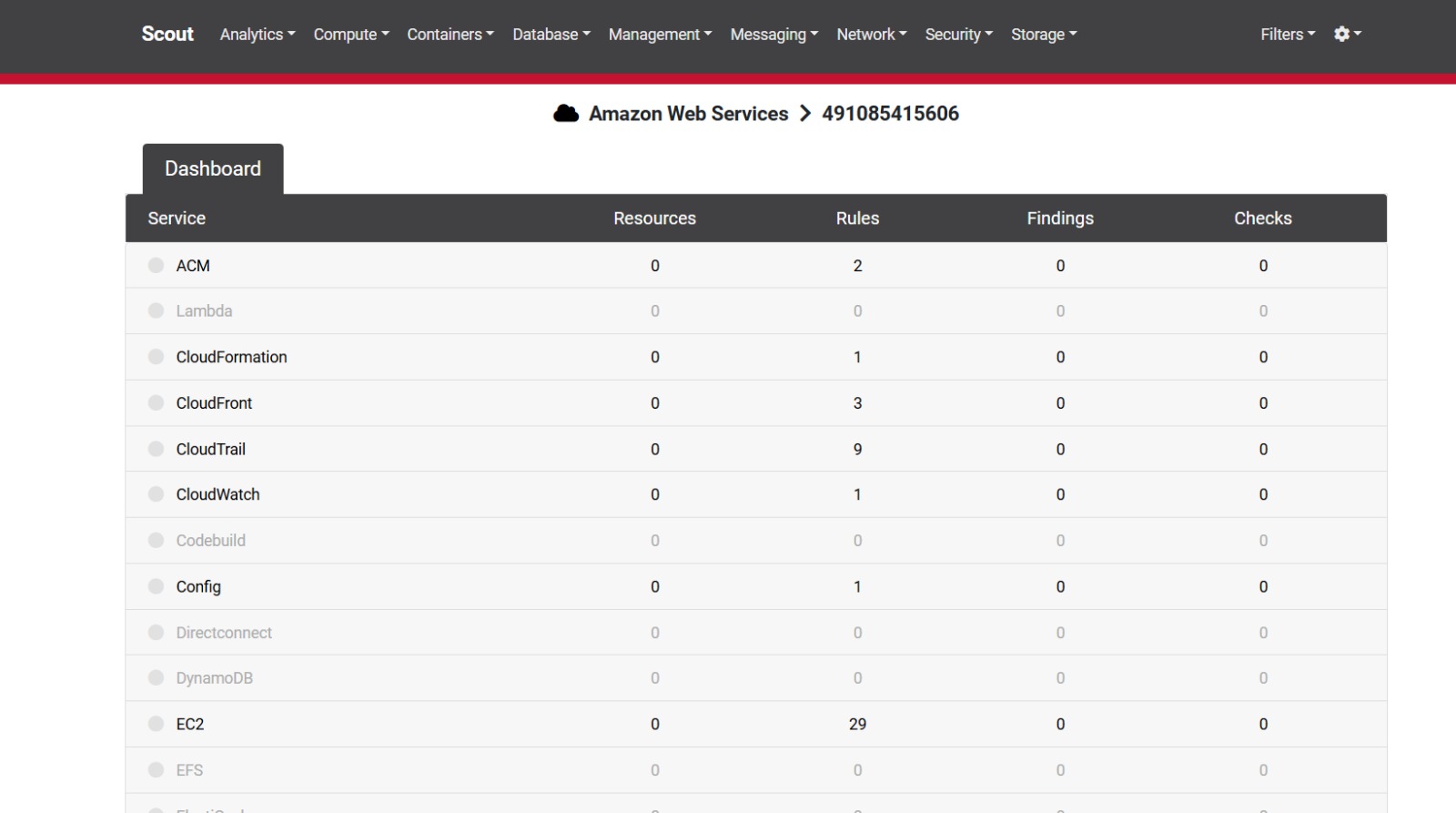
* Multi-cloud support – This supports AWS, Azure, Google cloud as well as major cloud platforms.
* User-friendly report – This basically generates HTML reports that are easy to understand and navigate.
* Customizable Checks – This basically allows customisation of security checks in order to align with organisational policies and requirements.

## Support for using ScoutSuite

* Easy to use – The tool has a user-friendly interface and HTML reporting format which ensures that security personnel can quickly analyse findings and take necessary actions, without the need for extensive training or specialised knowledge.
* Cost-effectiveness – The tool is regarded as an open-source tool which provides a cost-effective solution for cloud security auditing, which eliminates the need for expensive proprietary software without compromising on functionality.
* Customizability – ScoutSuite allows the ability to customize checks and adapt the tool to specific organizational security policies which makes it a versatile tool that can be tailored to meet the unique needs for an organization.
* Support and Continuous improvement – ScoutSuite benefits from continuous contributions ensuring that it stays up to date with the latest cloud security challenges as well as best practices. (Meller, 2023)

Implementation  
In terms of implementing Scoutsuite into the organisation’s security the workflow can be achieved through a progressive approach which will ensure a smooth transition and effective utilisation. It can be deployed in a read-only mode across the organisations cloud environments in order to perform a baseline security assessment. Basically, this would involve configuring access with the appropriate credentials to ensure the tool can gather necessary data without having an impact on operational services. In this instance the security team can then analyse the initial reports in order to identify critical vulnerabilities. In order to make sure that it is secure the security team can establish regular training and update sessions to ensure that all relevant personnel are proficient in using ScoutSuite enhancing a proactive security management across the organisation.

In Conclusion ScoutSuite is regarded as a powerful tool that offers significant benefits for improving the security posture of cloud environments. It has benefits such as comprehensive capabilities, easy to use, cost efficient which makes it ideal for organisations security needs. (Gavali, 2023)



A screenshot of a computer

Description automatically generated

# REFERENCES

Acunetix, n.d. Acunetix. [Online]. Available at:<https://www.acunetix.com/websitesecurity/sql-injection/#:~:text=How%20to%20Prevent%20an%20SQL,inputs%20such%20as%20login%20forms>. [Accessed 21 August 2024].

BasuMallick,C.2022. Spiceworks. [Online]. Available at:<https://www.spiceworks.com/it-security/application-security/articles/what-is-sql-injection/>. [Accessed 21 August 2024].

Bramhe,R.2024. OnPage. [Online]. Available at:<https://www.onpage.com/monitoring-and-alerting-101-monitoring-best-practices/>. [Accessed 29 August 2024].

Checkout.com, 2024. Checkout.com. [Online]. Available at:<https://www.checkout.com/blog/best-practices-for-secure-online-payment-processing>. [Accessed 13 August 2024].

CloudFare, 2024. CloudFare. [Online]. Available at:<https://www.cloudflare.com/learning/ddos/how-to-prevent-ddos-attacks/#:~:text=DDoS%20prevention%20methods&text=Several%20methods%20for%20reducing%20this,ports%2C%20protocols%2C%20and%20applications>. [Accessed 13 August 2024].

CloudFare, 2024. CloudFare. [Online]. Available at:<https://www.cloudflare.com/learning/cdn/glossary/anycast-network/>. [Accessed 29 August 2024].

Conrad, S., 2023. What is session hijacking and how do you prevent it?. [Online]. Available at:<https://www.emsisoft.com/en/blog/44071/what-is-session-hijacking-and-how-do-you-prevent-it/>. [Accessed 25 August 2024].

GeeksforGeeks, 2022. GeeksforGeeks. [Online]. Available at:<https://www.geeksforgeeks.org/what-is-salted-password-hashing/>. [Accessed 19 August 2024].

Haproxy, 2024. Haproxy. [Online]. Available at:<https://www.haproxy.com/glossary/what-is-rate-limiting>. [Accessed 29 August 2024].

Henretta, J., 2024. SkyWardit. [Online]. Available at:<https://skywardit.com/blog/security-audits-and-penetration-testing/#:~:text=Regular%20security%20audits%20and%20penetration%20testing%20can%20help%20prevent%20data,and%20protect%20sensitive%20customer%20data>. [Accessed 29 August 2024].

Katz, E., 2023. 8 Steps To Prevent Clickjacking. [Online]. Available at: <https://www.memcyco.com/home/steps-to-prevent-clickjacking/>. [Accessed 25 August 2024].

LinkedIn, 2024. LinkedIn. [Online]. Available at:<https://www.linkedin.com/advice/0/what-most-effective-web-application-firewall-configurations#:~:text=A%20WAF%20is%20a%20security,DoS)%2C%20and%20SQL%20injection>. [Accessed 21 August 2024].

Magnusson, A., 2024. Man-in-the-Middle (MITM) Attack: Definition, Examples & More. [Online]. Available at:<https://www.strongdm.com/blog/man-in-the-middle-attack>. [Accessed 02 September 2024].

Malik, F., 2024. Strong DM. [Online]. Available at:<https://www.strongdm.com/blog/man-in-the-middle-attack-prevention>. [Accessed 19 August 2024].

Martinelli, S., Dar, S. & Eder, L., 2024. LinkedIn. [Online]. Available at:<https://www.linkedin.com/advice/3/what-advantages-disadvantages-using-orm#:~:text=ORM%20frameworks%20prevent%20SQL%20injection%20attacks%20by%20using%20parameterized%20queries,as%20data%2C%20not%20executable%20code>. [Accessed 21 August 2024].

Medium. Ashish Gvali. 2023. Medium. [Online]. Available at: [Audit AWS Cloud Security using ScoutSuite | by Ashish Gavali | Globant | Medium](https://medium.com/globant/audit-aws-cloud-security-using-scoutsuite-4bc9073d2fc4). [Accessed 03 September 2024].

Mohaty, S., 2021. Research Gate. [Online]. Available at:<https://www.researchgate.net/publication/343969555_Detection_of_XSS_Vulnerabilities_of_Web_Application_Using_Security_Testing_Approaches>. [Accessed 18 August 2024].

Muthii, K., 2023. Golinuxcloud. [Online]. Available at:<https://www.golinuxcloud.com/mobsf-app-pentesting/>. [Accessed 18 August 2024].

Nihe, P., 2018. Hashed Out. [Online]. Available at:<https://www.thesslstore.com/blog/difference-encryption-hashing-salting/#:~:text=Hashing%20is%20a%20one%2Dway,changes%20the%20hash%20value%20produced>. [Accessed 19 August 2024].

Nops, 2024. Nops. [Online]. Available at:<https://www.nops.io/blog/cloud-scalability/>[Accessed 29 August 2024].

PM, 2024. Diagrams.net. [Online]. Available at: [https://app.diagrams.net/?src=about#](https://app.diagrams.net/?src=about)[Accessed 29 August 2024].

Portswigger, n.d. Portswigger. [Online]. Available at:<https://portswigger.net/web-security/sql-injection>[Accessed 21 August 2024].

PortSwigger, 2023. PortSwigger. [Online]. Available at:<https://portswigger.net/web-security/cross-site-scripting>[Accessed 18 August 2024].

Radware, 2024. Radware. [Online]. Available at:<https://www.radware.com/cyberpedia/ddos-protection/how-to-prevent-ddos-attacks-best-practices-strategies/>. [Accessed 14 August 2024].

Rapid7, 2024. Rapid7. [Online]. Available at:<https://www.rapid7.com/fundamentals/intrusion-detection-and-prevention-systems-idps/#:~:text=Read%20Now-,What%20is%20an%20Intrusion%20Detection%20and%20Prevention%20System%3F,behavior%20once%20it%20is%20flagged>.[Accessed 29 August 2024].

SimpleThread, 2023. MobSF. [Online]. Available at: [Scanning Your AWS Environment for Vulnerabilities With ScoutSuite - Simple Thread](https://www.simplethread.com/scanning-your-aws-environment-for-vulnerabilities-with-scoutsuite/) . [Accessed 03 September 2024].

Singh, H., 2018. Infosecinstitute. [Online]. Available at:<https://www.infosecinstitute.com/resources/penetration-testing/android-penetration-tools-walkthrough-series-mobsf/>[Accessed 18 August 2024].

SmartCom, 2024. smartcom. [Online]. Available at:<https://www.smart-com.si/en/system-information-and-communication-solutions/cybersecurity/waf-and-ddos-protection/#:~:text=It%20allows%20for%20a%20proactive,Denial%20of%20Service%20(DDoS)>. [Accessed 29 August 2024].

Timonera, K., 2023. eSecurity Planet. [Online]. Available at:<https://www.esecurityplanet.com/endpoint/prevent-xss-attacks/>. [Accessed 18 August 2024].

Trevino, A., 2024. What Is Session Hijacking and How Do You Prevent It?. [Online]. Available at:<https://www.keepersecurity.com/blog/2024/04/03/what-is-the-best-way-to-prevent-session-hijacking/>. [Accessed 25 August 2024].

Weaver, J., 2023. cradlepoint. [Online]. Available at:<https://cradlepoint.com/resources/blog/what-is-network-redundancy-and-network-failover-and-when-do-you-need-it/>. [Accessed 29 August 2024].

Yasar, K., 2024. Tech Target. [Online]. Available at: <https://www.techtarget.com/searchsecurity/definition/phishing#:~:text=What%20is%20phishing%3F,private%20information%20such%20as%20passwords>. [Accessed 18 August 2024].