Oliver Gruhn-van Dorp

Smartphone Secure Development Addendum

Implementing The European Union Agency for Network and Information Security (ENISA) Smartphone Secure Development Guidelines

April 12, 2024

GROON VANDORP PUBLISHINGS

To my son Arthur

Disclaimer

This book contains AI-generated content. While reasonable effort was taken to ensure accuracy and quality, no responsibility is taken for any mistakes, errors or omissions of the information presented. Furthermore, the information included in this book has been reviewed by subject matter experts and does not represent a guarantee of any kind. The reader of this book should exercise judgment, and discretion, with respect to the information contained herein.

Contents

Part I
INTRODUCTION

Part I serves as an essential foundation for the book, introducing key guidelines and frameworks that are pivotal in the realm of secure mobile application development. It delves into the European Union Agency for Cybersecurity (ENISA)'s Smartphone Secure Development Guidelines and the OWASP Mobile Application Security Verification Standard (MASVS), offering a comprehensive overview of the principles and practices vital for developing secure mobile apps. Additionally, it explores the Apple Security Framework, providing insights into its structure, key topics, and additional resources that are instrumental for iOS app developers.

Chapter 1
Guidelines covered in this book

1.1 ENISA's Smartphone Secure Development Guidelines

The European Union Agency for Cybersecurity (ENISA) provides a comprehensive set of guidelines in the document "Smartphone Secure Development Guidelines" for secure mobile application development. It encompasses strategies for sensitive data protection, robust user authentication, secure data transmission, and backend security. The document also covers third-party code integration, user consent and privacy, protection of paid resources, and secure software distribution. Guidelines on runtime code interpretation, device and application integrity, client-side injection protection, and the use of biometric sensors and secure hardware are also included. These guidelines aim to address the unique risks and vulnerabilities in mobile computing, enhancing the security of mobile applications.

About ENISA

ENISA, established in 2004, is a center of expertise for cyber security in Europe. The agency is actively involved in supporting EU member states, EU institutions, and businesses in developing and implementing effective cybersecurity measures. It focuses on enhancing the resilience of Europe's critical information infrastructures and networks.

Detailed Summary of ENISA's Smartphone Secure Development Guidelines

The ENISA document "Smartphone Secure Development Guidelines" offers detailed controls for secure smartphone application development:

1. Identify and Protect Sensitive Data on the Mobile Device

The guidelines stress the importance of classifying data by sensitivity and applying appropriate controls. Sensitive data should be processed and stored on the server, and when stored on the device, use OS-provided file encryption APIs. Further, it advises on re-evaluating access

authorization based on contextual information and avoiding storing historical location data or other sensitive information beyond necessary.

2. Implement User Authentication, Authorization, and Session Management Correctly

Emphasizes on implementing both authentication and authorization controls on the server side, using asymmetric cryptography, enforcing strong password policies, and introducing brute force protection mechanisms for authentication controls.

3. Handle Authentication and Authorization Factors Securely on the Device

Discusses using longer-term authorization tokens, securely storing passwords and tokens, leveraging key-store mechanisms, and purging credentials or keys from memory after use.

4. Ensure Sensitive Data is Protected in Transit

Advises assuming the network layer is not secure and enforcing the use of an end-to-end secure channel (like TLS) for transmitting sensitive information. It also suggests verifying the identity of the remote endpoint and leveraging platform-specific support for additional security requirements.

5. Secure the Backend Services and the Platform Server and APIs

Highlights the need to check for sensitive data unintentionally transferred between the mobile device and web-server back-ends, regularly testing back-end services for vulnerabilities, and ensuring that the back-end platform is running with a hardened configuration.

6. Secure Data Integration with Third Party Code

Emphasizes vetting third-party code for security and authenticity, tracking third-party frameworks for security patches, and validating all data received from third parties before processing.

7. Consent and Privacy Protection

Focuses on creating a privacy policy, obtaining user consent for personal data usage, and auditing communication mechanisms for unintended leaks.

8. Protect Paid Resources

Discusses maintaining logs of access to paid resources, checking for anomalous usage patterns, and considering a white-list model for paid resources.

9. Secure Software Distribution

Stresses the importance of designing applications to allow security patches and distributing apps through official app stores to benefit from their security checks.

10. Handle Runtime Code Interpretation Correctly

Addresses the need to filter user data passed to interpreters and restrict access to third-party domains that do not comply with required security standards.

11. Check Device and Application Integrity

Advises checking the device/platform integrity and the application's integrity to ensure that they are not modified.

12. Protect the Application from Client Side Injections

Discusses restricting access to third-party domains, verifying dynamic code downloads, and mitigating common web vulnerabilities like SQL injections and JavaScript injections.

13. Ensure Correct Usage of Biometric Sensors and Secure Hardware

Emphasizes verifying the presence and availability of biometric sensors on the device and ensuring that biometric data has not been changed since the activation of the authentication control.

This comprehensive set of guidelines from ENISA addresses various aspects of secure mobile application development, ensuring a robust approach to protect sensitive data, maintain application integrity, and safeguard against common security vulnerabilities in the mobile environment.

1.2 OWASP Mobile Application Security Verification Standard (MASVS)

The OWASP Mobile Application Security Verification Standard (MASVS) is a security certification framework by which mobile apps can be created and tested to ensure they adhere to high standards for privacy and information security.

Objective

The MASVS aims to provide a security baseline for mobile applications and a framework for comprehensive security testing.

Security Verification Levels

MASVS defines three verification levels that balance the cost of defense with the risk of exploitation:

- Level 1: Standard security requirements for mobile apps.
- Level 2: Sensitive applications dealing with higher risks and requiring stringent security controls.
- Resiliency Against Reverse Engineering and Tampering: High-risk apps are expected to resist reverse engineering and tampering attacks, even if they are targeted.

Security Requirements

Security requirements are categorized by domains. A summary of key requirements includes, but is not limited to, the following:

- Architecture and Threat Modeling
  - Establishment of a secure architecture.
  - Integration of threat modeling into the architectural design.
  - Protection against known vulnerabilities.
- Data Storage and Privacy
  - Encryption of sensitive data.
  - Adoption of privacy principles consistent with regulatory requirements.
  - Prevention of data leakage through security misconfiguration.
- Cryptography
  - Application of industry-standard cryptographic protocols.
  - Secure key management and storage practices.
- Authentication and Session Management
  - Strong user authentication mechanisms.

  - Secure handling of sessions and tokens.
- Network Communication
  - Use of secure, encrypted channels for all communications.
  - Robust protection against man-in-the-middle (MITM) attacks.
- Environmental Interaction
  - Guidelines for secure interaction with the platform environment.
  - Ensuring appropriate permission use and management.
- Code Quality and Build Settings
  - Adherence to secure coding guidelines.
  - Compilation with appropriate flags to enhance security.
- Resilience Against Reverse Engineering
  - Techniques for tamper detection and response.
  - Code obfuscation practices to hinder reverse engineering.

Compliance and Testing

The MASVS references the OWASP Mobile Security Testing Guide (MSTG) for practical testing strategies to verify compliance with MASVS requirements.

Conclusion

The MASVS serves as a detailed guide and benchmark for mobile app security, providing a robust framework for developers, QA teams, and security professionals to design, develop, and test secure mobile applications. It encourages continuous improvement in mobile app security posture.

1.3 NIST SP 800-163 Rev. 1 - Vetting the Security of Mobile Apps

Introduction

The publication emphasizes the critical role mobile applications play in organizational and personal contexts, highlighting the need for stringent security measures to mitigate risks associated with vulnerabilities and defects. It defines a mobile application vetting process aimed at ensuring apps conform to an organization's security requirements.

App Security Requirements

Security requirements are delineated into general requirements, drawing from standards and practices by NIAP, OWASP, MITRE, and NIST, and organization-specific requirements, tailored to the entity's policies, regulations, and risk tolerance.

App Vetting Process

The vetting process is described as comprising four main sub-processes: app intake, app testing, app approval/rejection, and results submission. It details the steps involved in evaluating an app's conformity to security requirements and the decision-making process regarding its deployment within an organization.

App Testing and Vulnerability Classifiers

This section outlines the methodologies employed in app testing, including correctness testing, source and binary code testing, and static and dynamic testing. It also introduces vulnerability classifiers and quantifiers such as CWE, CVE, and CVSS, providing a framework for identifying and assessing security vulnerabilities.

App Vetting Considerations

Considerations include managing and unmanaging apps, the implications of app whitelisting and blacklisting, limitations of app vetting, the use of local and remote tools and services, automated approval/rejection processes, the concept of reciprocity, tool report analysis, and the distinction between compliance and certification. The section also underscores the importance of budgeting and staffing for effective app vetting operations.

App Vetting Systems

Lastly, the publication emphasizes the need for a systematic approach to app vetting, advocating for the integration of app vetting systems within organizational security frameworks to streamline the vetting process, enhance security postures, and facilitate the management of mobile applications in alignment with an organization's security and operational goals.

Part II
OWASP MASVS-MAPPING

In Part II, the book presents a detailed annotation of guidelines that are critical for the secure development of mobile applications. This section meticulously covers various aspects such as data classification, secure storage, secure handling of sensitive information, encryption practices, and strategies for preventing data leakage. Each guideline is thoroughly explained, providing practical insights into implementing robust security measures in mobile apps. This part is instrumental for developers and security analysts seeking to enhance the security posture of their mobile applications.

Chapter 2
Identify and protect sensitive data on the mobile device

Identify and protect sensitive data on the mobile device Mobile devices due to their portable nature have a higher risk of loss or theft. Mobile applications need to take this into account and need to add the possibility of device loss into their security model. Mobile applications need to protect the data associated with the application. Specifically, sensitive data such as user personal data and business critical data has to be protected.

2.1 Implementation Guidance (ENISA 1.1):

ENISA Secure Smartphone Development Guidance (1.1): In the design phase, classify data storage according to sensitivity and apply controls accordingly (e.g. passwords, personal data, location, error logs, etc.). Process, store and use data according to its classification. Validate the security of API calls applied to sensitive data.

2.1.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the mentioned ENISA guideline is evident in the focus on security protocols and practices associated with user authentication, authorization, and the handling of sensitive data. MASVS-AUTH-1 emphasizes the importance of following best practices in the secure use of authentication and authorization protocols within apps, especially when connecting to remote endpoints, which aligns with the ENISA's guideline that stresses the need for proper classification and control of sensitive data, including the use of secure API calls for sensitive data. Both are concerned with ensuring that sensitive data, such as passwords and personal information, is protected accordingly throughout the design and implementation phases.
- MASVS-AUTH-2: Both ”MASVS-AUTH-2,” which discusses the correct implementation of authentication mechanisms including biometrics and local PIN codes, and the ENISA Guideline on classifying data storage according to sensitivity and applying appropriate controls, relate to the principle of protecting sensitive data. MASVS-AUTH-2 is focused on ensuring that the authentication mechanisms to access sensitive data are robust, whereas the ENISA Guideline emphasizes classifying and handling data based on its sensitivity, including enforcing secure API calls. Both guidelines aim to ensure that sensitive data is accessed securely and handled properly, making them correlated in the broader context of data security and protection.
- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3” and the ENISA guideline is that both emphasize the importance of additional security measures for sensitive actions or data within the app. ”MASVS-AUTH-3” calls for extra forms of authentication for sensitive actions, while the ENISA guideline suggests classifying data by sensitivity and applying appropriate controls, which includes secure processing and storage, as well as secure API calls for sensitive data. Both guidelines align on the principle that sensitive information requires enhanced security measures.
- MASVS-CODE-4: ”MASVS-CODE-4” and the ENISA Guideline both emphasize the importance of treating data as potentially untrusted input that needs to be handled with care, especially when it might be modified by untrusted sources. They both focus on ensuring that data is verified, sanitized, or processed with security in mind. ”MASVS-CODE-4” speaks specifically to the multiple entry points for data and the need to validate and sanitize to prevent attacks, while the ENISA Guideline emphasizes the need for classification, control application, and secure processing of data based on its sensitivity. Both pertain to the security considerations during design and implementation phases to mitigate potential data-related vulnerabilities.
- MASVS-CRYPTO-1: Both ”MASVS-CRYPTO-1” and the ENISA Guideline emphasize the importance of appropriately handling sensitive data in mobile applications. ”MASVS-CRYPTO-1” highlights the significance of cryptography best practices in securing user data, especially in scenarios where physical access to the device is possible. It underscores

the use of cryptography for data protection. On the other hand, the ENISA Guideline stresses the classification of data storage based on sensitivity and the application of corresponding security controls. While it mentions controls broadly (which includes cryptography as a control mechanism), it essentially aligns with the MASVS requirement by advocating for measures that ensure sensitive data is processed, stored, and used securely, including the secure validation of API calls handling sensitive data. Both guidelines converge on the principle of classifying and protecting sensitive data according to its level of sensitivity, and using suitable security controls, including cryptography, to safeguard against unauthorized access or disclosure.

- MASVS-CRYPTO-2: Both MASVS-CRYPTO-2 and the ENISA guideline emphasize the importance of properly handling sensitive data according to its classification. MASVS-CRYPTO-2 focuses on the lifecycle management of cryptographic keys, including secure generation, storage, and protection of keys, which are critical for maintaining the confidentiality and integrity of sensitive data when encryption is applied. The ENISA guideline advises on classifying data by sensitivity and applying appropriate controls while also ensuring the security of API calls when dealing with sensitive data. Both stress the safeguarding of sensitive information, albeit from slightly different angles—MASVS-CRYPTO-2 from the perspective of cryptographic key management specifically, and the ENISA guideline from a broader design and data classification perspective.
- MASVS-NETWORK-1: Both ”MASVS-NETWORK-1” and the ENISA Guideline emphasize the importance of protecting sensitive data through the application of appropriate security measures. ”MASVS-NETWORK-1” focuses on ensuring data privacy and integrity for data in transit, typically through encryption and endpoint authentication, warning against disabling secure defaults or bypassing them. The ENISA Guideline stresses the importance of classifying data according to sensitivity during the design phase and applying corresponding controls such as encryption for passwords and personal data. Both guidelines are concerned with validating security measures, although in different contexts: ”MASVS-NETWORK-1” with respect to secure connections and the ENISA Guideline with respect to API calls interacting with sensitive data. They share a common goal of upholding data security, whether the data is in transit or being processed by APIs.
- MASVS-PLATFORM-1: The correlation exists because ”MASVS-PLATFORM-1” focuses on secure interaction with IPC mechanisms which often involves API calls and data handling. Similarly, the ENISA Guideline emphasizes the importance of data classification and the secure processing, storage, and use of sensitive data. Both guidelines aim to ensure that data and functionality exposed through platform mechanisms or APIs are secured according to the sensitivity of the data involved. Secure IPC mechanisms are essential for preventing unauthorized access or leakage of sensitive data, which is in line with the ENISA Guideline’s recommendation to validate the security of API calls when they are applied to sensitive data.
- MASVS-PLATFORM-2: There is a correlation between ”MASVS-PLATFORM-2” and the ENISA guideline you provided. The MASVS-PLATFORM-2 requirement refers to ensuring that WebViews are configured securely to prevent sensitive data leakage and the exposure of sensitive functionality. This aligns well with the ENISA guideline, which suggests classifying data according to sensitivity and applying appropriate controls, as well as validating the security of API calls for sensitive data. Secure configuration of WebViews, as referred to in MASVS-PLATFORM-2, is a part of the process to protect sensitive data and could be considered as a specific control measure and validation procedure, which is what the ENISA guideline recommends. Both are concerned with protecting sensitive data throughout its handling processes.

- MASVS-PLATFORM-3: The correlation between ”MASVS-PLATFORM-3” and the ENISA Guideline lies in the emphasis on protecting sensitive data according to its level of sensitivity. ”MASVS-PLATFORM-3” focuses on preventing accidental data leakage through UI mechanisms like auto-generated screenshots and user behavior such as shoulder surfing. Similarly, the ENISA Guideline advises classifying data by sensitivity in the design phase and applying appropriate controls, including secure processing, storage, and API interactions for sensitive data like passwords and personal information. Both aim to ensure sensitive data is safeguarded throughout its lifecycle and through user interactions.
- MASVS-PRIVACY-1: ”MASVS-PRIVACY-1” and the ENISA Guideline both emphasize the importance of being mindful about data collection, data storage, and data processing. Both recommend only using data that is absolutely necessary for the app’s functionality, ensuring informed user consent, and applying appropriate security controls based on the sensitivity of the data. They align in the principle of data minimization and secure handling of sensitive data.
- MASVS-PRIVACY-2: Both MASVS-PRIVACY-2 and the ENISA Guideline emphasize on the importance of handling user data with care to protect user identity and privacy. MASVS-PRIVACY-2 focuses on unlinkability techniques and ensuring data streams are used only for their intended function, while the ENISA Guideline advocates for data classification according to sensitivity and applying appropriate controls for data processing, storage, and usage, along with validating the security of API calls for sensitive data. Both sets of guidelines aim to safeguard user privacy through structured handling of data and careful consideration of data sensitivity in the mobile app development process.
- MASVS-PRIVACY-3: The correlation between ”MASVS-PRIVACY-3” and the ENISA guideline is evident as both emphasize the importance of transparent data handling and classification according to sensitivity. MASVS-PRIVACY-3 focuses on user awareness and adherence to platform guidelines, while the ENISA guideline details the design phase responsibility to classify data storage by sensitivity and implement appropriate controls. Both guidelines aim to ensure the secure and transparent handling of user data in mobile applications.
- MASVS-PRIVACY-4: The given description of ”MASVS-PRIVACY-4” emphasizing user control over their data aligns with the ENISA guideline on data classification and applying appropriate controls. Both stress the importance of protecting sensitive data and ensuring users have mechanisms to manage their privacy and consent, reflecting a shared focus on privacy and security in the handling of user data.
- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the described ENISA Guideline is evident in their shared emphasis on the security and trustworthiness of the platform on which applications run. MASVS-RESILIENCE-1 underlines the importance of ensuring that the operating system has not been compromised, as a compromised OS could undermine security measures like secure storage and sandboxing that are essential for protecting sensitive app data. Similarly, the ENISA Guideline stresses the necessity of classifying data based on its sensitivity and applying appropriate controls to ensure its security during the design phase. Both directives focus on safeguarding sensitive data by way of applying stringent controls and validating the security posture of the platform, whether through directly assessing the integrity of the OS (as per MASVS-RESILIENCE-1) or through securely handling data according to its classification and employing secure API calls (as advised by the ENISA Guideline).
- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the ENISA guideline exists because both refer to the handling of sensitive data with appropriate security measures. ”MASVS-STORAGE-1” emphasizes that apps should protect sensitive

data regardless of the storage location, ensuring that if data needs to be stored locally, it is protected properly. Similarly, the ENISA guideline advises classifying data storage by sensitivity and applying appropriate controls, which includes processing, storing, and using data in accordance with its classification and validating the security of API calls dealing with sensitive data. Both are concerned with securing sensitive data within the storage mechanisms used by an app.

- MASVS-STORAGE-2: Both the MASVS-STORAGE-2 description and the ENISA guideline highlight the importance of being aware of how sensitive data is stored and handled. They both imply the necessity for appropriate controls based on the sensitivity of the data to prevent unintentional exposure. MASVS-STORAGE-2 focuses on avoiding accidental leaks due to misuse of APIs or system features, while the ENISA guideline advises on classifying data by sensitivity and securing API calls dealing with such data. Both stress the significance of secure data handling in the design phase to protect sensitive information.

2.2 Implementation Guidance (ENISA 1.2):

ENISA Secure Smartphone Development Guidance (1.2): Store and process sensitive data on the server instead of the client-end device. The relative security of client vs. server-side security also needs to be assessed on a case-by-case basis (see ENISA cloud risk assessment or the OWASP Cloud top 10 for decision support). Highly sensitive data (e.g., biometric data, private keys) should not be transported from the component that were initially created.

2.2.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA guideline is that both emphasize the security of sensitive data and the use of best practices in handling authentication and authorization. MASVS-AUTH-1 discusses the need for secure use of protocols for apps requiring user authentication and communication with a remote endpoint, while the ENISA guideline advises on limiting sensitive data processing to the server side rather than the client end, thus also implying the need for robust authentication and authorization procedures. The concepts of not exposing sensitive data on the client and ensuring robust protocols intersect in both recommendations.
- MASVS-AUTH-2: The correlation between ”MASVS-AUTH-2” and the ENISA guideline is found in the emphasis on proper implementation of authentication mechanisms, including biometrics or local PIN codes, and the security considerations for processing and storing sensitive data. MASVS-AUTH-2 suggests that apps need to correctly implement local authentication, including biometrics and PIN codes. On the other hand, the ENISA guideline advises against storing and processing highly sensitive data, like biometric data, on the client-end device, advocating for server-side storage and processing instead. Both highlight the importance of security in handling sensitive authentication data, albeit from slightly different angles: MASVS-AUTH-2 focuses on correct implementation, while ENISA suggests server-side processing as a more secure option.
- MASVS-AUTH-3: The MASVS-AUTH-3 refers to implementing additional forms of user authentication securely, which may involve using sensitive data like biometrics or MFA codes. The ENISA guideline suggests that sensitive data should be processed on the server rather than the client-end device to enhance security. Both point towards the importance of securing sensitive data and actions, which suggests that there is a correlation between the two guidelines in terms of where and how sensitive data and authentication mechanisms should be handled securely.
- MASVS-CODE-2: While MASVS-CODE-2 is about ensuring that there is a mechanism to force updates to the app when critical vulnerabilities are found, the ENISA Guideline emphasizes the importance of storing and processing sensitive data on the server side rather than the client-end device to maintain security. These two guidelines correlate in that both are measures to protect sensitive data and maintain the security of the application. MASVS-CODE-2 addresses the scenario where an already deployed application has vulnerabilities that could potentially be exploited, and by forcing updates, it steps towards mitigating those risks. The ENISA Guideline complements this by advocating for sensitive data processing to occur on the more secure server side, which could inherently reduce the risk of exposing such data through vulnerabilities in the client-end application that MASVS-CODE-2 seeks

to address with forced updates. Both guidelines contribute to an overarching security strategy to safeguard sensitive data in mobile applications.

- MASVS-CODE-3: Both MASVS-CODE-3 and the ENISA Guideline emphasize the importance of assessing security risks related to third-party components and sensitive data handling. MASVS-CODE-3 highlights the challenge of fully assessing all app components, especially third-party ones, and mitigating known vulnerabilities. Similarly, the ENISA Guideline advises on the need to store and process sensitive data server-side rather than client-side, also urging risk assessment for deciding on security measures. Both guidelines are concerned with identifying and mitigating potential vulnerabilities to ensure data security.
- MASVS-CODE-4: The ”MASVS-CODE-4” description emphasizes the importance of treating all incoming data as untrusted input, which necessitates proper verification and sanitation before usage to prevent security vulnerabilities such as SQL injection, XSS, and insecure deserialization. Similarly, the ENISA guideline advises against processing and storing sensitive data on the client-end device, suggesting that it should rather be done server-side, where it is typically more secure. Both principles are aligned in advocating the secure handling of data to mitigate risks and suggest greater trust is placed in server-side processing due to the relative security advantages servers have over client devices. This correlation indicates a common security approach that prioritizes the validation, sanitation, and secure management of input data to avoid the exploitation of security weaknesses.
- MASVS-CRYPTO-1: The MASVS-CRYPTO-1 description emphasizes the importance of cryptography in protecting user data, especially on mobile devices where physical access is a more common threat. It advises adherence to general cryptography best practices as defined in external standards. The ENISA Guideline also addresses the secure handling of sensitive data, recommending server-side storage and processing over client-end (mobile) devices due to security considerations, which aligns with the need for strong cryptographic practices recommended by MASVS-CRYPTO-1. Both highlight the importance of secure handling of highly sensitive data, such as biometric data and private keys, which is part of cryptographic best practices. Therefore, there is a correlation in the focus on protecting sensitive data through secure practices and appropriate data storage locations.
- MASVS-CRYPTO-2: Both MASVS-CRYPTO-2 and the ENISA guideline emphasize the importance of proper handling and protection of sensitive data and cryptographic keys. MASVS-CRYPTO-2 focuses on the management of cryptographic keys throughout their lifecycle to ensure strong cryptography isn't undermined by poor key management practices. The ENISA guideline advises against storing and processing sensitive data on the client-end device and suggests that highly sensitive data, such as private keys, should remain on the server or their initial component, which aligns with the goal of protecting key material from exposure and mishandling as indicated in MASVS-CRYPTO-2. Both sources advocate for the careful consideration and protection of sensitive data, highlighting the risks associated with client-side storage and the heightened security typically associated with server-side storage.
- MASVS-NETWORK-1: The MASVS-NETWORK-1 requirement and the ENISA guideline both share a common theme of protecting sensitive data, particularly in transit. MASVS-NETWORK-1 focuses on the importance of securing connections and maintaining data privacy and integrity when data is being transmitted over the network. This involves using encryption and authentication methods like TLS to safeguard data exchanges. The ENISA guideline advises against storing and processing sensitive data on the client-end device and suggests that such operations should be performed on the server side, where security can be more tightly controlled. This guideline implicitly acknowledges

the risks associated with transmitting sensitive data – if it must happen, it should be done securely, aligning with the intent of MASVS-NETWORK-1 to enforce secure connections and data transmission practices. Both directives aim to prevent unauthorized access and/or tampering with sensitive data and imply a need for rigorous data protection measures, especially when data is not within the confines of a secure server environment. Therefore, there is a correlation in that both are concerned with maintaining the security and integrity of sensitive data, particularly when it leaves a more secure server-side environment.

- MASVS-NETWORK-2: The correlation between "MASVS-NETWORK-2" and the provided ENISA Guideline is that both address concerns related to the security and handling of sensitive data within mobile applications. MASVS-NETWORK-2 refers to the practice of certificate pinning or public key pinning, which is an approach that enhances the security of HTTPS connections by trusting only specific certificate authorities (CAs), thus preventing man-in-the-middle attacks by ensuring that the app communicates only with the intended server. The ENISA Guideline, on the other hand, advises that sensitive data be stored and processed on the server rather than the client device, and it recognizes that the security of data should be evaluated in each specific context, pointing to resources like the OWASP Cloud top 10. The correlation here is both recommendations aim to protect sensitive data. Certificate pinning ensures data in transit is secured by validating certificates, which indirectly helps to ensure that sensitive data remains secure while it is transmitted to the server where it is supposed to be processed and stored, aligning with the ENISA Guideline's emphasis on server-side data handling for increased security. The rationale is that by doing so, risks associated with compromised client-side environments are mitigated.
- MASVS-PLATFORM-1: The MASVS-PLATFORM-1 control and the ENISA Guideline both emphasize security in the context of data interaction and processing. MASVS-PLATFORM-1 focuses on ensuring secure IPC (Inter-Process Communication) mechanisms in apps, which is aligned with the ENISA guideline that recommends sensitive data should be processed on the server rather than the client device. Both are concerned with the secure handling of sensitive data to prevent unauthorized access or leakage, although through different aspects – MASVS-PLATFORM-1 through secure app interactions, and ENISA by advocating for server-side processing. This reflects a correlation in their intent to enhance data security.
- MASVS-PLATFORM-2: The MASVS-PLATFORM-2 description emphasizes the secure configuration of WebViews to prevent sensitive data leakage and exposure of sensitive functionality. This aligns with the ENISA Guideline which advises that sensitive data should be processed on the server instead of the client-end device to enhance security. Both guidelines stress the importance of protecting sensitive data, and invariably, following MASVS-PLATFORM-2's guidance on securely using WebViews is a supportive action towards the ENISA principle of handling sensitive data server-side when possible.
- MASVS-PLATFORM-3: There is a correlation between the MASVS-PLATFORM-3 description and the ENISA guideline. Both the Mobile Application Security Verification Standard (MASVS) mentioned and the ENISA guideline emphasize the importance of handling sensitive data properly to prevent unintentional leaks and enhance security. MASVS-PLATFORM-3 specifically addresses avoiding leaks through UI mechanisms and insecure handling on the device, while the ENISA guideline suggests keeping sensitive data processing and storage on the server side rather than the client device, which inherently also protects against the types of leaks that MASVS-PLATFORM-3 is concerned with, such as screenshots or shoulder surfing. Furthermore, assessing the security needs on a case-by-case basis, as mentioned in the ENISA guideline, complements the goal of

MASVS-PLATFORM-3 to protect sensitive data from being leaked by considering the security implications of client vs. server-side processing and storage.

- MASVS-PRIVACY-1: Both ”MASVS-PRIVACY-1” and the ENISA guideline emphasize the principle of data minimization and secure data handling. ”MASVS-PRIVACY-1” requires that apps should request only the data necessary for their functionality with informed user consent, aligning with data protection principles. The ENISA guideline highlights the best practice of storing and processing sensitive data on the server side rather than the client end device and assessing the security of client vs. server-side security case by case. Both stress the importance of protecting sensitive data and reducing the risk of data breaches or leaks, with ”MASVS-PRIVACY-1” also addressing third-party SDKs and supply chain responsibility, which complements the ENISA guideline’s advice on secure data handling and locations.
- MASVS-PRIVACY-2: The correlation between MASVS-PRIVACY-2 and the ENISA guideline is that both emphasize protecting user identity and sensitive data. MASVS-PRIVACY-2 advises on techniques to avoid user identification and tracking, such as data abstraction and pseudonymization, and cautions on the isolated use of fingerprint-like signals for their intended purposes. Similarly, the ENISA guideline recommends storing and processing sensitive data on the server instead of the client-end device, which is also a measure to protect sensitive data and user privacy by not exposing it on less secure devices. Both guidelines seek to limit the risk to user privacy by recommending strategies that minimize exposure of sensitive data and potential misuse.
- MASVS-PRIVACY-3: Both the MASVS-PRIVACY-3 requirement and the ENISA Guideline emphasize the importance of data protection and privacy. MASVS-PRIVACY-3 focuses on the transparency of how apps handle user data, including collection, storage, and sharing practices. It ensures apps inform users about data usage that may be unexpected, such as background data collection, and comply with platform data declaration guidelines. On the other hand, the ENISA Guideline recommends storing and processing sensitive data on the server-side rather than the client-end device, highlighting the risks associated with client-side data storage and arguing for the importance of assessing security for sensitive data on a case-by-case basis. Both guidelines aim to protect users by ensuring sensitive data is handled correctly, with MASVS-PRIVACY-3 focusing on user awareness and consent and the ENISA Guideline on the technical aspects of data security.
- MASVS-RESILIENCE-1: The MASVS-RESILIENCE-1 control and the ENISA guideline both emphasize the importance of operating in a secure environment to protect sensitive data. MASVS-RESILIENCE-1 stresses the risks of running an app on a compromised platform and the necessity of trusting the platform’s security features, such as secure storage and sandboxing. The ENISA guideline recommends that sensitive data be processed on the server rather than the client-side device due to the potential risks associated with client-end device security. Both sources advocate for the protection of highly sensitive information within a secured and trusted environment. This shows a correlation between the two in terms of their approach to managing sensitive data securely and the consideration of the security of the underlying platform.
- MASVS-RESILIENCE-2: The Mobile Application Security Verification Standard (MASVS) Resilience requirement MASVS-RESILIENCE-2 and the ENISA guideline both recognize that client-side (user-controlled devices) security is inherently weaker compared to server-side security because of the user’s ability to modify or tamper with the application and the data stored on the device. MASVS-RESILIENCE-2 aims to protect the integrity of the app by preventing unauthorized modifications, which aligns with the ENISA guideline’s recommendation to process and store sensitive data server-side to mitigate the risks asso-

ciated with client-side storage and processing. Both prioritize securing the application by considering the security implications of where data is processed and stored.

- MASVS-RESILIENCE-3: The correlation between ”MASVS-RESILIENCE-3” and the ENISA Guideline is that both are focused on protecting sensitive data and impeding unauthorized access or modification of the mobile application. MASVS-RESILIENCE-3 is about making static analysis difficult to prevent understanding and tampering with the app, while the ENISA Guideline advises on processing and storing sensitive data on the server rather than the client-end device, which aligns with the goal of preventing tampering and protecting sensitive information since server environments typically have stronger security controls. Both guidelines aim to limit the exposure of sensitive data and reduce the risk of compromise by considering the security of the environment where the data is handled.
- MASVS-RESILIENCE-4: The correlation between MASVS-RESILIENCE-4 and the described ENISA Guideline exists in their shared goal of enhancing security against dynamic analysis and manipulation. MASVS-RESILIENCE-4 focuses on making dynamic analysis and runtime instrumentation difficult, which is aligned with the ENISA guideline's recommendation to store and process sensitive data on the server rather than the client device. By doing so, sensitive data is less exposed to dynamic analysis and potential manipulation on the less secure, client-end device. Both guidelines aim to prevent attackers from modifying or accessing sensitive data at runtime by advocating for server-side processing and storage, where security measures can be more robustly enforced.
- MASVS-STORAGE-1: The correlation exists because both the MASVS-STORAGE-1 guideline and the ENISA recommendation emphasize the importance of proper handling and protection of sensitive data. MASVS-STORAGE-1 specifically addresses where sensitive data can be stored by an app, whether in private or public storage locations, and the need for proper protection mechanisms. The ENISA guideline advises against storing and processing sensitive data on the client-end device, which complements the MASVS-STORAGE-1 by suggesting that storing such data on the server is generally safer and also indicates the need for a case-by-case assessment of the security on client versus server sides. Both guidelines ultimately aim at enhancing the security of sensitive information, with MASVS-STORAGE-1 focusing on local storage within the app and ENISA promoting server-side storage and processing.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA Guideline is that both emphasize the importance of protecting sensitive data from unintentional leaks and exposure. MASVS-STORAGE-2 addresses the prevention of unintentional storage of sensitive data in publicly accessible locations due to the misuse of APIs or system capabilities, while the ENISA Guideline advises against storing and processing sensitive data on the client-end device, suggesting that it should be done on the server for better security. Both guidelines share the common goal of ensuring that highly sensitive data is kept secure and not exposed through client-side vulnerabilities or mishandling.

2.3 Implementation Guidance (ENISA 1.3):

ENISA Secure Smartphone Development Guidance (1.3): When storing sensitive data on the device, use a file encryption API provided by the OS or other trusted source. Some platforms (e.g., iOS and Android) provide file encryption API's which use a secret key protected by the device unlock code and deletable on remote wipe. If this is available, it should be used as it increases the security of the encryption without creating extra burden on the end-user. It also makes stored data safer in the case of loss or theft. However, it should be borne in mind that even when protected by the device unlock key, if data is stored on the device, its security is dependent on the security of the device unlock code if remote deletion of the key is for any reason not possible.

2.3.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: The MASVS-AUTH-2 guideline mentions the importance of correctly implementing biometrics or local PIN code authentication mechanisms, which may rely on local app authentication without a remote endpoint. The ENISA Guideline advises using file encryption APIs provided by the OS that are protected by the device unlock code, which is often the same security mechanism used for biometric or PIN code authentication. Both guidelines emphasize the importance of securing local authentication methods and the data they protect, showing a correlation between the proper implementation of local authentication (MASVS-AUTH-2) and the use of OS-provided encryption APIs (ENISA Guideline) that leverage the device unlock code.
- MASVS-CRYPTO-1: The ENISA Guideline recommends using file encryption APIs provided by the OS or another trusted source, which aligns with the description of "MASVS-CRYPTO-1" in emphasizing the importance of implementing cryptographic best practices to protect user data, especially in scenarios where physical access to a device is possible. Both stress the significance of leveraging built-in platform features for encryption to enhance security and reduce user burden, demonstrating a correlation between the MASVS standard and the ENISA recommendation.
- MASVS-CRYPTO-2: The MASVS-CRYPTO-2 guideline emphasizes the importance of proper cryptographic key management, which implies secure key generation, storage, and protection. The ENISA guideline stresses using a file encryption API provided by the OS, which inherently involves the OS handling the key management, such as protecting the key with the device unlock code and allowing for remote wipe of the key. Both guidelines recognize that the strength of cryptography is heavily reliant on secure key management practices, which includes protecting the key and considering the security context, such as the device's unlock code security. There is a correlation as both guidelines aim to ensure that encryption keys are managed and protected securely to maintain the overall security of the encrypted data.
- MASVS-PRIVACY-1: Both the MASVS-PRIVACY-1 description and the ENISA Guideline emphasize the importance of secure data handling practices. MASVS-PRIVACY-1 focuses on data minimization, user consent, and careful sharing of data, especially with third parties, which aligns with the ENISA Guideline's recommendation to utilize file encryption APIs provided by the OS to secure sensitive data stored on the device. Both stress the importance of protecting user data and reducing the risk associated with data leaks or breaches.

- MASVS-PRIVACY-2: While the MASVS-PRIVACY-2 control focuses on techniques to avoid user identification and tracking, and the ENISA Guideline provides recommendations on how to securely store sensitive data, both share the underlying goal of protecting user privacy and data security. MASVS-PRIVACY-2's use of data abstraction, anonymization, and pseudonymization can be seen as complementary to the ENISA Guideline's recommendation to use file encryption APIs that are tied to the device unlock code. Both controls aim to ensure that user data is safeguarded against unauthorized access and potential privacy breaches.
- MASVS-PRIVACY-3: The Mobile Application Security Verification Standard (MASVS) PRIVACY-3 guideline focuses on informing users about how their data is used, including data collection, storage, and sharing practices. It aligns with the ENISA guideline on securely storing sensitive data on the device using the file encryption APIs provided by the operating system, as both stress the importance of secure data handling practices. While MASVS-PRIVACY-3 emphasizes transparency and user awareness, the ENISA guideline provides a technical method to ensure the security of data stored on devices, which is a complementary aspect of protecting users' privacy rights as outlined in MASVS-PRIVACY-3. Together, they contribute to the overall objective of safeguarding user data and trust by ensuring proper usage and security measures are in place.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the ENISA Guideline is that both emphasize the importance of relying on the underlying platform's security features for protecting sensitive data. MASVS-RESILIENCE-1 mentions that running on a compromised platform can disable security features such as secure storage, which is where the ENISA Guideline suggests using the file encryption API provided by the OS. Both acknowledge that the security of the app and its data is dependent on the integrity and trust of the platform. The platform's encryption APIs are considered a trusted source as per the ENISA Guideline, which aligns with the MASVS requirement of trusting the platform's security features.
- MASVS-STORAGE-1: The description of "MASVS-STORAGE-1" emphasizes the importance of proper protection for sensitive data stored by an app, regardless of whether it is stored in private or public locations. The ENISA Guideline also stresses the need for using file encryption APIs that are protected by the device unlock code, providing a higher level of security for the stored sensitive data. Both are concerned with the security of sensitive data on the device and recommend measures to protect it, reflecting a correlation between the MASVS control and the ENISA Guideline.
- MASVS-STORAGE-2: The ENISA guideline emphasizes the utilization of file encryption APIs provided by the operating system or another trusted source to protect sensitive data stored on the device. This aligns with the description of "MASVS-STORAGE-2," which focuses on preventing unintentional leaks of sensitive data due to the misuse of APIs or system capabilities. Both the MASVS requirement and the ENISA guideline acknowledge the risks associated with storing sensitive information and provide measures to secure it—specifically, through the use of appropriate encryption APIs to prevent unauthorized access in case of device loss or theft.

2.4 Implementation Guidance (ENISA 1.4):

ENISA Secure Smartphone Development Guidance (1.4): Verify that OS level storage encryption is enabled and the device is protected by a PIN or passphrase.

2.4.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: The correlation between MASVS-AUTH-2, which discusses the correct implementation of biometric or local PIN code authentication mechanisms, and the ENISA Guideline about enabling OS level storage encryption and device protection through a PIN or passphrase, is that both guidelines are concerned with strengthening the authentication mechanisms and securing user access control. MASVS-AUTH-2 emphasizes the importance of properly implementing local authentication methods, such as biometrics or PIN codes, which can be seen as the first line of defense in app security. On the other hand, the ENISA Guideline draws attention to the need for securing the device through encryption and strong access controls (PIN/passphrase), which serve to protect the data at rest on the device level and prevent unauthorized access. This correlation implies that both local application security, as stated in MASVS, and device-level security, as recommended by ENISA, are necessary to provide a comprehensive security posture and mitigate the risk of unauthorized access or data breaches.
- MASVS-CRYPTO-1: The ”MASVS-CRYPTO-1” description addresses the importance of cryptography in securing user data, particularly in mobile environments where physical access to a user’s device is a plausible threat. It emphasizes adherence to general cryptography best practices. The ENISA guideline ”Verify that OS level storage encryption is enabled and the device is protected by a PIN or passphrase” is directly correlated with these best practices. It outlines a specific mechanism (OS level storage encryption) for protecting data at rest on the device, and a method (PIN or passphrase) for ensuring that only authenticated users can access the encrypted data. Both statements are concerned with securing user data on mobile devices through encryption and access controls.
- MASVS-CRYPTO-2: The correlation between ”MASVS-CRYPTO-2,” which emphasizes the importance of managing cryptographic keys throughout their lifecycle, including key generation, storage, and protection, and the ENISA Guideline, which requires verification that OS level storage encryption is enabled and the device is protected by a PIN or passphrase, is evident. Both requirements focus on ensuring the security of cryptographic assets. MASVS-CRYPTO-2 underlines the overall importance of secure key management practices to prevent compromise, while the ENISA Guideline specifically looks at the implementation of such practices at the OS level, ensuring that keys used for storage encryption are well-protected by user authentication mechanisms like PINs or passphrases. This shows a concern for protecting cryptographic keys (which are critical to storage encryption) against unauthorized access, which aligns with the broader goal of secure key lifecycle management.
- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the ENISA Guideline regarding OS-level storage encryption and device protection is evident in their shared emphasis on the security of the underlying platform. The MASVS-RESILIENCE-1 control highlights the risks of running apps on tampered platforms where key security features could be disabled, thereby compromising app data. This is aligned with the ENISA Guideline’s focus on ensuring that OS-level storage encryption is enabled

and that the device is safeguarded with a PIN or passphrase, both measures aimed at maintaining the integrity and trust of the operating system's security features. Both documents underscore the importance of platform security as a foundation for protecting an application's data and the overall system.

- MASVS-RESILIENCE-3: The correlation between "MASVS-RESILIENCE-3" and the ENISA guideline on OS level storage encryption with PIN or passphrase protection lies in the context of securing an app against unauthorized access and tampering. Although they approach security from different angles, both are part of a defensive strategy to protect sensitive data and the integrity of the app. MASVS-RESILIENCE-3 is about making static analysis difficult for an attacker trying to understand or tamper with an app, thus protecting its internals from being comprehended and potentially modified. The ENISA guideline about ensuring OS-level storage encryption and device protection via a PIN or passphrase is a protection mechanism for the data at rest. It ensures that the data stored by the app on the device is encrypted, making it unreadable without proper authentication, and therefore protects against data extraction and manipulation if the device is lost or stolen. In essence, while MASVS-RESILIENCE-3 is a proactive measure within the app to prevent understanding and tampering of the code, the ENISA guideline is a broader security measure for protecting the data on the system where the app resides. Both contribute to a layered framework of security controls that harden the app and its environment against compromise.
- MASVS-RESILIENCE-4: MASVS-RESILIENCE-4 focuses on making dynamic analysis and instrumentation difficult for an attacker. This would include efforts to prevent attackers from modifying code during runtime, which is often done by implementing various security measures. The ENISA Guideline on ensuring that OS-level storage encryption is enabled and the device is protected by a PIN or passphrase does not correlate directly with hindering dynamic analysis and runtime manipulation but is connected to the overall aim of reducing the risk of unauthorized access to application data and code. If an attacker cannot bypass the PIN or passphrase, they are less likely to access the app's runtime environment and perform dynamic analysis or instrumentation. While the two guidelines address different aspects of security, both contribute to a resilient security posture by protecting against different types of threats.
- MASVS-STORAGE-1: Both "MASVS-STORAGE-1" and the ENISA guideline mentioned focus on ensuring the protection of sensitive data on mobile devices. "MASVS-STORAGE-1" addresses the proper handling and storage of sensitive data by the app, regardless of the storage location (private or public), which implicitly requires protection mechanisms to be in place. The ENISA guideline directly specifies the use of OS-level storage encryption and device protection via PIN or passphrase as methods to safeguard data. This aligns with the intent of "MASVS-STORAGE-1" to secure sensitive data stored locally on the device, as encryption and access control are fundamental aspects of data protection.

2.5 Implementation Guidance (ENISA 1.5):

ENISA Secure Smartphone Development Guidance (1.5): Do not store/cache sensitive data (including keys) unless they are encrypted and if possible stored in a platform supported tamper-proof area.

2.5.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation here is that the MASVS-AUTH-1 standard is focused on ensuring that user authentication and authorization are implemented according to best practices to secure the protocols used for communication with a remote endpoint. This implies diligence in handling sensitive data, such as credentials and session tokens, which should not be improperly stored or cached. The ENISA Guideline's advice not to store/cache sensitive data unless encrypted and stored securely complements the MASVS-AUTH-1 by outlining the standards for the secure storage of such data, which is a part of following best practices in authorization and authentication mechanisms. Therefore, following the ENISA Guideline would be a key aspect of adhering to MASVS-AUTH-1 concerning the secure use of authentication and authorization protocols.
- MASVS-AUTH-2: The correlation between "MASVS-AUTH-2" which requires that biometrics or a local PIN code are correctly implemented, and the ENISA Guideline about not storing or caching sensitive data unless encrypted and if possible stored in a platform-supported tamper-proof area, is that both are emphasizing the need for secure authentication mechanisms and the protection of sensitive information. MASVS-AUTH-2 implies a correct implementation which would include not storing any sensitive auth-related data in a way that is easily accessible or unencrypted, which is in line with the ENISA Guideline's recommendation for encryption and the use of secure storage areas for any sensitive data, to prevent unauthorized access or tampering.
- MASVS-AUTH-3: While MASVS-AUTH-3 discusses the implementation of additional authentication methods for sensitive actions, it implies secure implementation which would include not storing or caching sensitive data insecurely. The ENISA Guideline advises against storing or caching sensitive data unless it is encrypted and stored in a secure area, aligning with the secure implementation aspects touched upon in MASVS-AUTH-3. Both guidelines are concerned with the protection of sensitive data and actions within the app, therefore they are correlated.
- MASVS-CRYPTO-1: The correlation is that both the MASVS-CRYPTO-1 description and the ENISA Guideline emphasize the importance of cryptography for the protection of sensitive data, especially in a mobile environment where physical access to a device can enable attacks. MASVS-CRYPTO-1 speaks to the broader application of general cryptography best practices for securing user data, while the ENISA guideline provides a specific recommendation regarding the storage of sensitive data, suggesting encryption and the use of tamper-proof areas supported by the platform. Both are focused on mitigating the threat of unauthorized access to sensitive data on mobile devices.
- MASVS-CRYPTO-2: The correlation between "MASVS-CRYPTO-2" and the ENISA Guideline is clear. The MASVS-CRYPTO-2 description clearly emphasizes the importance of proper management of cryptographic keys throughout their lifecycle. This includes concerns about how keys are generated, stored, and protected. The ENISA Guideline speaks to a similar concern, stating that sensitive data (which includes cryptographic

keys) should not be stored or cached unless they are encrypted and, if possible, stored in a platform-supported tamper-proof area. Both guidelines are aimed at preventing the compromise of sensitive data, specifically cryptographic keys, by enforcing strict storage and protection protocols.

- MASVS-PLATFORM-1: The ”MASVS-PLATFORM-1” requirement and the ENISA guideline both address the secure handling of data within a mobile application. Specifically, ”MASVS-PLATFORM-1” focuses on the secure interaction involving IPC (Inter-Process Communication) mechanisms on the platform, suggesting a need to ensure that any data or functionality exposed through these mechanisms are secured. On the other hand, the ENISA guideline emphasizes the secure storage and caching of sensitive data by recommending encryption and the use of platform-supported tamper-proof areas. While ”MASVS-PLATFORM-1” is more broadly about secure IPC interaction, both aspects complement each other in the broader context of securing sensitive data against unauthorized access and tampering, whether via IPC or storage on the device. Ensuring IPC is secure by design implicitly covers the safe handling of possibly cached or stored sensitive data during IPC operations, aligning with the guideline’s directive to protect sensitive data at rest.
- MASVS-PLATFORM-3: Both ”MASVS-PLATFORM-3” and the ENISA Guideline express concerns regarding the secure handling of sensitive data on mobile platforms. ”MASVS-PLATFORM-3” focuses on preventing unintentional leaks of sensitive data that is displayed in the UI due to platform mechanisms or user behavior (like shoulder surfing). The ENISA Guideline emphasizes not storing or caching sensitive data unless it is encrypted and stored in a secure area supported by the platform. The correlation lies in the objective to protect sensitive data from being exposed or compromised, albeit through different aspects of mobile application security – display and storage respectively. Both guidelines aim to enhance the security posture by outlining measures to safeguard sensitive data within mobile applications.
- MASVS-PRIVACY-3: The ENISA guideline on not storing or caching sensitive data unless encrypted and in a secure area is correlated with MASVS-PRIVACY-3, which emphasizes clear information about data practices and adherence to platform guidelines. Both stress the importance of transparent and secure data handling to protect user privacy.
- MASVS-PRIVACY-4: While ”MASVS-PRIVACY-4” is focused on user control over their data including management and consent mechanisms, the ENISA Guideline emphasizes not storing or caching sensitive data unless it’s encrypted and ideally placed in a secure area. The correlation lies in the emphasis on protecting user data and ensuring privacy. Both require measures to prevent unauthorized access or mishandling of user data, albeit from different action points: user control and transparency for MASVS-PRIVACY-4 versus encryption and secure storage for the ENISA Guideline. The essence of both is to uphold data privacy and security.
- MASVS-RESILIENCE-1: The MASVS-RESILIENCE-1 requirement and the ENISA guideline both emphasize the importance of a secure underlying platform for the proper functioning of security controls in mobile apps. MASVS-RESILIENCE-1 focuses on validating that the operating system has not been compromised to ensure the reliability of its security features such as secure storage and biometrics, which is necessary for protecting the app’s data. The ENISA guideline dictates not to store or cache sensitive data unless it is encrypted and preferably stored in a platform-supported tamper-proof area, further underlining the necessity of a secure and unaltered platform for the protection of sensitive data. Both statements recognize that a tampered platform can undermine app security by disabling or evading critical security mechanisms.

- MASVS-RESILIENCE-2: The correlation between "MASVS-RESILIENCE-2" which emphasizes preventing modifications to the app's original code and resources to maintain integrity and the ENISA guideline on not storing or caching sensitive data unencrypted is clear. Both controls are concerned with ensuring the security and integrity of the application and its data on user-controlled devices. "MASVS-RESILIENCE-2" aims to protect the app from being modified or tampered with, which could include the insertion of malicious code or enabling unauthorized features. Similarly, the ENISA guideline recommends encryption for any sensitive data stored on the device to protect it from unauthorized access or modification, thus promoting the resilience of the app against such attacks. Hence, both controls contribute to protecting the application from tampering and preserving its intended functionality.
- MASVS-RESILIENCE-3: "MASVS-RESILIENCE-3" and the ENISA guideline both emphasize the importance of obstructing unauthorized access to sensitive information in mobile applications. The MASVS guideline focuses on impeding comprehension of the app's internals, thus making it difficult to tamper with the application through static analysis. This objective aligns with the ENISA guideline which advises not to store or cache sensitive data unless it is encrypted and, if possible, stored in a tamper-proof area supported by the platform. Both are concerned with enhancing app resilience against reverse engineering and tampering, and they complement each other by addressing different layers of security: MASVS-RESILIENCE-3 targets the difficulty of analysis and ENISA targets the security of storage mechanisms for sensitive data.
- MASVS-RESILIENCE-4: The correlation between "MASVS-RESILIENCE-4," which describes the goal of making dynamic analysis and instrumentation difficult, and the ENISA guideline advising against storing sensitive data unless it's encrypted and in a tamper-proof area, is that both aim at increasing the application's resistance to tampering and unauthorized analysis. MASVS-RESILIENCE-4 targets the prevention of real-time code modification and examination, whereas the ENISA guideline focuses on the protection of sensitive data at rest. Both contribute to the resilience of mobile applications against attacks that could exploit stored data or real-time application behavior.
- MASVS-STORAGE-1: Both the MASVS-STORAGE-1 requirement and the ENISA Guideline emphasize the importance of protecting sensitive data that is intentionally stored by the app. They concur on the principle that if sensitive data must be stored, it should be encrypted to prevent unauthorized access. Additionally, the ENISA Guideline mentions the preference for storage in a platform-supported tamper-proof area, which aligns with MASVS-STORAGE-1's emphasis on proper protection, regardless of the storage location. The correlation lies in the shared objective of securing sensitive data against illicit access or tampering.
- MASVS-STORAGE-2: Both "MASVS-STORAGE-2" and the ENISA guideline focus on the proper handling of sensitive data to prevent unintentional leaks. MASVS-STORAGE-2 mentions the unwanted storage or exposure of sensitive data due to the misuse of APIs and system capabilities, while the ENISA guideline advises against storing or caching sensitive data unless it is encrypted and ideally kept in a secure area supported by the platform. Both stress the importance of developers taking preventive measures to safeguard sensitive data.

2.6 Implementation Guidance (ENISA 1.6):

ENISA Secure Smartphone Development Guidance (1.6): Consider re-evaluating access authorization to sensitive data based on contextual information such as location (e.g., require further authentication if location data shows device is outside of expected region).

2.6.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation exists because both "MASVS-AUTH-1" and the ENISA guideline emphasize the importance of secure authentication and authorization practices within mobile apps, particularly when they are accessing remote endpoints. "MASVS-AUTH-1" outlines the requirement for apps to follow best practices for secure protocol use related to authentication and authorization. The ENISA guideline similarly suggests enhancing security by re-evaluating access authorization based on contextual information, such as location, which is a specific example of how an app could enforce authorization policies to secure sensitive data. Essentially, both are focused on strengthening the authentication and authorization mechanisms to protect user data and access to sensitive resources.
- MASVS-AUTH-2: The correlation between "MASVS-AUTH-2" which discusses the implementation of local authentication mechanisms, such as biometrics or PIN codes, and the ENISA Guideline about re-evaluating access authorization based on contextual information like location is that both relate to enhancing the security of the authentication process. MASVS-AUTH-2 focuses on the integrity and correct implementation of local authentication methods, while the ENISA Guideline suggests an additional layer of security by considering the user's context (like location) to determine if further authentication is needed. Both aim to protect sensitive data by ensuring that the user is authorized to access the data, potentially combining multiple factors or conditions to grant access.
- MASVS-AUTH-3: The guideline from ENISA recommending re-evaluation of access authorization based on contextual information like location directly correlates with MASVS-AUTH-3, which advocates for additional forms of authentication for sensitive actions within an app. Both directives emphasize the need for enhanced security measures—such as MFA, biometric verification, or contextual checks like location—to secure sensitive transactions or data access in mobile applications.
- MASVS-CRYPTO-1: Both "MASVS-CRYPTO-1" and the ENISA guideline highlight the importance of security measures in mobile environments where physical access to a device can potentially compromise sensitive data. While MASVS-CRYPTO-1 underscores the role of cryptography in protecting user data in such scenarios by adhering to best practices, the ENISA guideline extends the concept by suggesting the re-evaluation of access to sensitive data based on contextual information like location. Both advise integrating additional protective layers in response to contextual risk, aligning in their regard for enhancing data security through contextual and cryptographic measures.
- MASVS-PLATFORM-2: Although "MASVS-PLATFORM-2" and the ENISA guideline address different aspects of mobile app security, there's an indirect correlation. "MASVS-PLATFORM-2" focuses on the secure configuration of WebViews, which can include mechanisms to protect against sensitive data leakage. The ENISA guideline suggests re-evaluating access to sensitive data based on contextual information like location, which also

relates to preventing sensitive data exposure. Both are components of a larger strategy to protect sensitive information within mobile applications, and measures ensuring WebView security could potentially include context-aware access controls suggested by ENISA.

- MASVS-PRIVACY-1: There is a correlation between "MASVS-PRIVACY-1" and the ENISA Guideline mentioned. Both emphasize the importance of limiting and controlling access to sensitive data. "MASVS-PRIVACY-1" advocates for data minimization and informed user consent, ensuring that third-party SDKs adhere to consent requirements, which is in line with the ENISA Guideline's call for re-evaluating access to sensitive data based on context, such as location, to enhance security measures when necessary. Both guidelines aim to protect user privacy and data integrity in different contexts by implementing access controls based on necessity and contextual information.
- MASVS-PRIVACY-2: Both "MASVS-PRIVACY-2" and the mentioned ENISA Guideline focus on protecting user privacy by considering contextual information and employing techniques to limit the identification and tracking of users. MASVS-PRIVACY-2 emphasizes the importance of using unlinkability methods such as data abstraction, anonymization, and pseudonymization to protect user identity and also stresses the need to create technical boundaries when using identifying data for specific purposes, whereas the ENISA Guideline suggests that access to sensitive data may need further authentication based on the context, such as location, which is a form of contextual information control to safeguard user data and privacy. Both align with the overarching goal of reducing privacy risks associated with user data processing.
- MASVS-PRIVACY-3: Both "MASVS-PRIVACY-3" and the ENISA Guideline presented emphasize the importance of proper management and transparency regarding the use of sensitive user data. The MASVS-PRIVACY-3 focuses on the need to inform users about data usage practices, including unexpected behaviors like background data collection, adhering to platform guidelines. The ENISA Guideline suggests re-evaluating access to sensitive data based on contextual factors like location to enhance security, which complies with the principle of providing clear information and managing data access responsibly, as per the user's expectations outlined in MASVS-PRIVACY-3. Both guidelines promote user awareness and data protection.
- MASVS-PRIVACY-4: Both the MASVS-PRIVACY-4 guideline and the ENISA guideline emphasize the importance of user control over their own data. MASVS-PRIVACY-4 focuses on providing users with the ability to manage their data and modify privacy settings while ensuring they are informed of and consent to any changes regarding the use of their data. The ENISA guideline complements this by suggesting re-evaluation of access authorization based on contextual information, which aligns with the principle of adapting privacy measures to ensure users' data control in changing circumstances, such as when the user's device is located in an unexpected region. Both presuppose mechanisms that ensure user data is protected and that users maintain control over access to their sensitive data under varying contexts.
- MASVS-STORAGE-1: The correlation between "MASVS-STORAGE-1" and the ENISA guideline on re-evaluating access authorization to sensitive data based on contextual information such as location lies in their shared focus on protecting sensitive data. MASVS-STORAGE-1 addresses the need for apps to protect sensitive data regardless of where it is stored, which includes taking measures to ensure that sensitive data is not exposed or accessible by unauthorized parties. The ENISA guideline complements this by suggesting that access to sensitive data should be dynamically controlled based on context, such as location, which is a strategy to enhance protection measures for sensitive data, especially when the context indicates a higher risk of unauthorized access (e.g., when the device

is outside an expected region). Both highlight the importance of securing sensitive data throughout its lifecycle, from storage to access control.

2.7 Implementation Guidance (ENISA 1.7):

ENISA Secure Smartphone Development Guidance (1.7): Do not store historical location data or other sensitive information on the device beyond the period required by the application. Assume that shared storage is untrusted - information may easily leak in unexpected ways through any shared storage. In particular: (A) Be aware of caches and temporary storage as a possible leakage channel, when shared with other apps. (B) Be aware of shared storage such as address book, media gallery, audio files, as a possible leakage channel. For example, storing images with location metadata in the media-gallery allows that information to be shared in unintended ways. (C) Do not store temporary cached data in a world readable directory

2.7.1 OWASP MASVS MAPPING

- MASVS-AUTH-3: The correlation can be seen in the emphasis on security for sensitive user actions in MASVS-AUTH-3 and the concern for secure storage and handling of sensitive information in the ENISA guideline. MASVS-AUTH-3 focuses on securing methods for additional authentication within the app, ensuring that the means of authentication like biometric, PIN, or MFA code generators are implemented securely to protect sensitive actions. Correspondingly, the ENISA guideline is concerned with securely storing sensitive data and not keeping such data longer than necessary, particularly highlighting the risk of data leakage through shared or temporary storage. Both points stress the importance of security measures to protect sensitive data and actions within the app, highlighting potential risks of data leakage and the need for secure implementation practices.
- MASVS-CODE-4: While MASVS-CODE-4 primarily concerns itself with ensuring that incoming data is treated as untrusted and is verified and sanitized, the ENISA Guideline focuses on the principle that data stored on the device should be considered exposed to untrusted actors, which aligns with the principle of treating incoming data as potentially malicious or untrusted in MASVS-CODE-4. Both are focused on data security and the assumption that various entry points (storage, UI, IPC, network, file system, etc.) can be exploited and therefore should not be fully trusted. MASVS-CODE-4's emphasis on data validation and sanitation before usage complements the ENISA Guideline's concern about historical and sensitive data leakage through shared storage, caches, and temporary directories. Both guidelines aim to prevent the misuse of data by untrusted actors.
- MASVS-CRYPTO-1: The ENISA guideline advises against storing sensitive data, like historical location data, on a device beyond what is necessary, and highlights the risk of data leakage through shared or temporary storage. The MASVS-CRYPTO-1 description emphasizes the importance of cryptography for securing user data, especially on mobile devices where physical access is a plausible threat. The correlation exists because both stress the potential risks of data exposure and the necessity of protecting data at rest. Implementing robust cryptography measures as recommended by MASVS-CRYPTO-1 would mitigate the risks outlined by the ENISA guideline, thus reinforcing the guideline's emphasis on not trusting shared storage and being cautious with how sensitive information is managed on mobile devices.
- MASVS-CRYPTO-2: The correlation between "MASVS-CRYPTO-2" and the mentioned ENISA Guideline is that both emphasize the importance of proper management of sensitive information. While MASVS-CRYPTO-2 focuses on cryptographic key lifecycle manage-

ment to ensure they are generated, stored, and protected securely, the ENISA Guideline warns against storing sensitive data (such as historical location data) beyond what is necessary and highlights that shared storage can be untrusted. Both highlight risks associated with improper storage practices that could lead to data leakage, although they focus on different aspects of data handling (cryptographic keys vs. sensitive information).

- MASVS-PLATFORM-1: The MASVS-PLATFORM-1 requirement from the Mobile Application Security Verification Standard (MASVS) relates to ensuring the secure interaction of apps with their environment, which includes the proper handling of Inter-Process Communication (IPC) mechanisms. This correlates with the ENISA Guideline that emphasizes not storing sensitive information on the device beyond the required period and assuming that shared storage is untrusted, which includes being cautious of information leakage through caches, temporary storage, and shared resources such as the media gallery. Both MASVS-PLATFORM-1 and the ENISA guideline aim to prevent unintended data exposure through shared resources, which is consistent with the secure handling of IPC mechanisms.
- MASVS-PLATFORM-3: The correlation between "MASVS-PLATFORM-3" and the described ENISA Guideline revolves around the issue of sensitive data leakage. "MASVS-PLATFORM-3" focuses on preventing unintentional data leakage through platform mechanisms such as screenshots or via physical means like shoulder surfing. Similarly, the ENISA Guideline advises against storing sensitive data on the device beyond necessary and highlights potential leakage channels, including caches, temporary storage, and shared storage. Both emphasize the importance of handling sensitive data cautiously to prevent unintended disclosure.
- MASVS-PRIVACY-1: Both the MASVS-PRIVACY-1 description and the ENISA Guideline focus on the principle of data minimization and the protection of sensitive information. They both emphasize the need to limit the collection, access, and storage of user data to what is strictly necessary for the app's functionality. Furthermore, they both address the risks associated with data sharing, storage, and potential leaks, especially in relation to shared or temporary storage access by third parties or other apps.
- MASVS-PRIVACY-2: The ENISA Guideline's recommendation to avoid storing historical location data or other sensitive information on the device aligns with the principles described in MASVS-PRIVACY-2, which emphasizes the use of unlinkability techniques to prevent user identification and tracking. Both guidelines are concerned with mitigating the risk of sensitive data leakage and ensuring that user privacy is maintained by limiting data retention and exposure.
- MASVS-PRIVACY-3: The ENISA guideline emphasizes on not storing sensitive information, including historical location data, on the device for longer than necessary and recognizes that shared storage can be untrusted. It also warns about the risks of sensitive data leakage through caches, temporary storage, and other shared storage like media galleries. This is directly related to "MASVS-PRIVACY-3," which is about providing users clear information about data collection, storage, and sharing practices. Both stress the importance of user awareness and control over data handling, especially for unexpected or non-transparent background data collection and sharing, thus indicating a correlation in their emphasis on protecting user privacy and secure data management practices.
- MASVS-PRIVACY-4: Both MASVS-PRIVACY-4 from the Mobile Application Security Verification Standard (MASVS) and the stated ENISA Guideline emphasize the importance of user control over their data and minimizing the risk of data leakage. MASVS-PRIVACY-4 focuses on giving users mechanisms to manage, delete, and modify their data as well as to be transparent about data use and consent. The ENISA Guideline advises against storing sensitive data beyond what the application requires and to consider shared storage

as untrusted. Both address concerns regarding the protection of user data, data leakage through shared storage, and the management and privacy of user data.

- MASVS-RESILIENCE-1: The described MASVS-RESILIENCE-1 deals with the security of the platform itself, including the trust in the platform's security features such as secure storage and sandboxing which are essential to prevent data leakage. The given ENISA Guideline advises against storing sensitive information beyond the necessary period and highlights the risks associated with shared and temporary storage, which can be compromised if the platform's security features are disabled or tampered with. Both the MASVS-RESILIENCE-1 and the ENISA guideline emphasize the importance of ensuring the integrity and security of the platform to protect sensitive data from being leaked through shared or insecure storage mechanisms.
- MASVS-RESILIENCE-3: The correlation between "MASVS-RESILIENCE-3" which focuses on preventing understanding of the app's internals to impede tampering, and the ENISA Guideline on not storing sensitive information beyond the required period, particularly with regards to shared storage, lies in the overarching principle of safeguarding the app's data and operation from unauthorized access or manipulation. Both are concerned with security measures to maintain the app's integrity and protect user data. While MASVS-RESILIENCE-3 is about making the app's functioning opaque, the ENISA Guideline is about preventing data leakage (including from caches and shared storage), which could give an attacker insights into the app's behavior or user data, thus facilitating tampering or reverse engineering.
- MASVS-STORAGE-1: The MASVS-STORAGE-1 description emphasizes the proper protection of sensitive data stored by the app, regardless of whether it is kept in private or public locations. The ENISA Guideline advises against storing sensitive information like historical location data for longer than necessary and warns about the risks of shared storage, explicitly mentioning caches, temporary storage, and shared resources like media galleries as potential leakage channels. Both the MASVS control and ENISA guideline highlight the importance of treating shared storage as untrusted and the necessity of controlling where and how sensitive data is stored to avoid unintended leakage. Consequently, there is a clear correlation between MASVS-STORAGE-1 and the ENISA guideline concerning the management and protection of sensitive data on mobile devices.
- MASVS-STORAGE-2: Both the MASVS-STORAGE-2 and the ENISA Guideline caution against unintentional storage or exposure of sensitive data in publicly accessible or shared locations. The MASVS control is about preventing unintentional leaks which developers can avoid, and the ENISA Guideline provides specific examples of where leaks can occur: caches, temporary storage, shared storage such as an address book or media gallery, and how storing sensitive data like historical location data can lead to unintended sharing. Both stress the importance of not keeping sensitive information beyond what is necessary and handling it with the awareness that shared storage is untrusted.

2.8 Implementation Guidance (ENISA 1.8):

ENISA Secure Smartphone Development Guidance (1.8): For sensitive personal data, deletion should be scheduled according to a maximum retention period, (to prevent e.g. data remaining in caches indefinitely).

2.8.1 OWASP MASVS MAPPING

- MASVS-CRYPTO-2: While MASVS-CRYPTO-2 focuses on the management of cryptographic keys throughout their lifecycle, including generation, storage, and protection, the ENISA Guideline addresses the retention and scheduled deletion of sensitive personal data. Both controls imply a time-bound aspect to data protection—MASVS-CRYPTO-2 through the lifecycle of cryptographic keys which must be securely managed to maintain their integrity and the ENISA Guideline through ensuring data does not exceed its intended retention period. Proper key management as outlined in MASVS-CRYPTO-2 could include policies for how keys should be retired or replaced in order to prevent unauthorized access after the maximum retention period for the encrypted data has expired. Thus, there's a correlation in the sense that both are concerned with preventing unauthorized access to sensitive data over time, albeit through different mechanisms—one through key management and the other through data retention policies.
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA guideline mentioned is that both address the concern of protecting sensitive data within mobile applications. "MASVS-PLATFORM-2" emphasizes secure configuration of WebViews to prevent data leakage and exposure of sensitive functionality, while the ENISA guideline specifies that sensitive personal data should have a scheduled deletion according to a maximum retention period to prevent it from remaining in caches indefinitely. Both advise on measures to safeguard sensitive data, but from different angles; one focusing on UI components and their secure use while the other on data retention policies.
- MASVS-PRIVACY-1: Both the MASVS-PRIVACY-1 description and the ENISA Guideline emphasize the importance of limiting the access, retention, and potential misuse of sensitive user data. The MASVS-PRIVACY-1 stresses the need for informed consent, data sharing with third parties only when necessary, and careful integration of third-party SDKs to prevent unauthorized data collection or ignoring consent signals. Similarly, the ENISA Guideline focuses on the deletion of sensitive personal data after a maximum retention period to prevent indefinite data caching. Both guidelines advocate for strict data management practices to protect user privacy and security.
- MASVS-PRIVACY-3: MASVS-PRIVACY-3 and the described ENISA Guideline both focus on user privacy and the management of personal data. MASVS-PRIVACY-3 emphasizes the importance of transparency regarding data use, which includes informing users about data collection, storage, and sharing practices. The ENISA Guideline complements this by stating that sensitive personal data should have a maximum retention period and be scheduled for deletion, which is a part of managing how data is stored and ensuring that it is not kept indefinitely, especially in places like caches. Both aim to protect user privacy and limit the risks related to personal data handling.
- MASVS-PRIVACY-4: The statement "For sensitive personal data, deletion should be scheduled according to a maximum retention period (to prevent e.g. data remaining in caches indefinitely)" from the ENISA guidelines and the description of "MASVS-

PRIVACY-4" both emphasize user control over their personal and sensitive data, including the need for mechanisms to manage, delete, and modify their data. The ENISA guideline focuses on setting a maximum retention period for sensitive data, which is aligned with the MASVS requirement that users should have the ability to manage and delete their data. Thus, there is a correlation between MASVS-PRIVACY-4 and the ENISA guideline concerning the control users have over their private and sensitive data, specifically regarding the deletion and retention of such data.

- MASVS-STORAGE-1: Both the MASVS-STORAGE-1 requirement and the ENISA Guideline emphasize the importance of handling sensitive data with care. MASVS-STORAGE-1 focuses on the protection of sensitive data, regardless of where it is stored, while the ENISA guideline addresses the retention period and scheduled deletion of such data to avoid indefinite storage in caches. Both are concerned with ensuring sensitive data is not exposed or accessible beyond its intended use or timeframe.
- MASVS-STORAGE-2: The correlation exists because both "MASVS-STORAGE-2" and the ENISA guideline emphasize the management of sensitive data to prevent unintended storage or prolonged retention. MASVS-STORAGE-2 focuses on avoiding unintentional leaks due to developer oversight related to APIs and system features like backups or logs, while the ENISA guideline suggests scheduling deletion of sensitive personal data to avoid indefinite retention, which could include unintended storage in caches or other publicly accessible locations. Both address the principle of limiting exposure and access to sensitive data through appropriate data lifecycle management.

2.9 Implementation Guidance (ENISA 1.9):

ENISA Secure Smartphone Development Guidance (1.9): There is currently no standard secure deletion procedure for flash memory (unless wiping the entire medium/card). Therefore, data encryption and secure key management are especially important.

2.9.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: The correlation between ”MASVS-AUTH-2,” which concerns the correct implementation of authentication mechanisms in apps, and the ENISA guideline on the importance of data encryption and secure key management is that both focus on aspects of securing user data. Implementing biometric or PIN code authentication mechanisms properly helps prevent unauthorized access to an app, which is crucial if the app does not rely on remote authentication and stores sensitive information locally. The ENISA guideline emphasizes the need for encryption and secure key management since secure deletion is problematic on flash memory; this is in line with MASVS-AUTH-2's concern with local app security. If the local authentication mechanisms are compromised (which MASVS-AUTH-2 seeks to prevent), inadequate key management or encryption would result in easier unauthorized access to data stored on the device, aligning both statements in their purpose of offering secure data management and access control.
- MASVS-CODE-1: While MASVS-CODE-1 does not directly address data deletion or encryption, it is related to maintaining a secure environment by ensuring the app is running on an up-to-date mobile OS which includes the latest security patches and features. The ENISA guideline emphasizes the importance of data encryption and secure key management in the absence of a secure deletion standard for flash memory. Both the MASVS-CODE-1 requirement and the ENISA guideline are concerned with protecting data against known threats, albeit from different angles: one through updating the OS to benefit from security improvements, and the other through ensuring data protection measures like encryption are in place due to the challenges with secure deletion on flash memory. Both approaches ultimately aim to enhance the security of data on mobile devices.
- MASVS-CRYPTO-1: Both ”MASVS-CRYPTO-1” and the ENISA Guideline emphasize the importance of cryptography to protect user data, especially in scenarios where physical access to a device is possible. ”MASVS-CRYPTO-1” discusses the role of cryptography in a mobile environment and general best practices, which implies secure key management as an essential part. The ENISA Guideline emphasizes the lack of a standard secure deletion process for flash memory, highlighting the need for strong data encryption and secure key management. These statements correlate as they both identify cryptography and key management as critical elements for securing data on devices that are susceptible to physical access by attackers.
- MASVS-CRYPTO-2: Both ”MASVS-CRYPTO-2” and the ENISA Guideline emphasize the importance of secure key management in the context of cryptography. MASVS-CRYPTO-2 covers the management of cryptographic keys during their entire lifecycle, highlighting that even strong cryptography can be undermined by poor key management practices. Similarly, the ENISA Guideline notes the lack of a standard secure deletion procedure for flash memory and underscores the significance of data encryption coupled with secure key management as crucial for data security. The common thread in both

is the pivotal role of managing cryptographic keys securely to maintain the integrity of cryptographic operations.

- MASVS-NETWORK-1: Both "MASVS-NETWORK-1" and the ENISA Guideline emphasize the importance of data privacy and integrity for information that could be exposed. The "MASVS-NETWORK-1" is focused on ensuring secure data transit by preventing the bypass of secure defaults and emphasizing proper encryption and authentication of connections. On the other hand, the ENISA Guideline addresses the absence of secure deletion procedures for flash memory, pointing out the necessity of encryption and key management for data at rest. While one deals with data in transit and the other with data at rest, both underline encryption as a critical measure for protecting sensitive data.
- MASVS-PLATFORM-3: The correlation between MASVS-PLATFORM-3 and the ENISA guideline in question lies in the focus on protecting sensitive data. MASVS-PLATFORM-3 addresses the risk of unintended data leakage through platform mechanisms like screenshots or direct observation, suggesting a need for precautions when displaying sensitive data. Similarly, the ENISA guideline implies that since secure deletion is not straightforward for flash memory, encryption and secure key management become critical in preventing unauthorized access to sensitive data. Both are concerned with preventing the exposure of sensitive information, albeit through different aspects of data handling and storage.
- MASVS-PRIVACY-2: The MASVS-PRIVACY-2 control and the ENISA guideline both emphasize the importance of protecting user identity and data. MASVS-PRIVACY-2 suggests using techniques like data abstraction, anonymization, and pseudonymization to prevent user identification and tracking. Similarly, the ENISA guideline recognizes that secure deletion procedures for flash memory are not standardized and hence stresses the importance of data encryption and secure key management as a means to protect sensitive information. Both are focused on ensuring that user data is not compromised and that user privacy is maintained through technical measures, thus showing a correlation in their objectives of enhancing user privacy and data security.
- MASVS-PRIVACY-3: Both the MASVS-PRIVACY-3 guideline and the ENISA Guideline highlight the importance of managing user data with care. MASVS-PRIVACY-3 focuses on transparency around data use and adherence to data policies, whereas the ENISA Guideline emphasizes the technical aspect of securing data (through encryption and key management) due to the lack of a standard secure deletion process for flash memory. Both guidelines complement each other in aiming to protect user privacy and ensure the security of user data.
- MASVS-PRIVACY-4: The MASVS-PRIVACY-4 guideline focuses on giving users control over their data, including the ability to manage, delete, and modify their data. The ENISA Guideline's emphasis on the lack of a standard secure deletion procedure for flash memory highlights the importance of encryption and secure key management as a way to protect user data, which aligns with the control and privacy management aspects mentioned in MASVS-PRIVACY-4. If the data is encrypted, managing keys effectively can render data inaccessible upon deletion, indirectly providing a form of user control over their data even in the absence of secure deletion methods for flash memory.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the ENISA Guideline on secure deletion procedure is present in the emphasis on the integrity and trust of the platform's security features. MASVS-RESILIENCE-1 highlights the dangers of running apps on tampered platforms which may compromise security features that protect the app's data. The ENISA Guideline complements this by stating the importance of data encryption and secure key management, especially when standard secure deletion is not possible. Both point to a reliance on the underlying platform's security mechanisms

to safeguard data. If the platform is tampered with, it undermines secure storage and key management measures, thus making the correlation evident.

- MASVS-RESILIENCE-3: The correlation between ”MASVS-RESILIENCE-3” and the ENISA guideline regarding secure deletion procedures for flash memory is that both are concerned with protecting the app and its data from unauthorized access or tampering. ”MASVS-RESILIENCE-3” focuses on making it difficult to understand an app’s internals through obfuscation to prevent static analysis, while the ENISA guideline emphasizes the importance of data encryption and secure key management as methods to protect data, especially when secure deletion is not feasible. Despite focusing on different aspects of security (obfuscation vs. encryption), both guidelines aim to increase the resilience of mobile applications against reverse engineering, unauthorized data access, and tampering.
- MASVS-RESILIENCE-4: The ”MASVS-RESILIENCE-4” refers to making it difficult to perform dynamic analysis or dynamic instrumentation of an app. This control is related to protecting the app’s runtime environment and ensuring that its behavior cannot be easily modified or understood by attackers. On the other hand, the ENISA guideline about the lack of a standard secure deletion procedure for flash memory implies that since data cannot be securely deleted on a granular level, it is important to encrypt data and manage encryption keys securely. Both statements are concerned with safeguarding data integrity and impeding unauthorized access or modification. The MASVS control focuses on runtime protection which includes memory where encryption keys might reside, while the ENISA guideline emphasizes the need for encryption and key management as a compensatory control for the inability to securely delete data. Both contribute to an overall security posture that protects against dynamic threats.
- MASVS-STORAGE-1: The correlation exists because both the MASVS-STORAGE-1 and the ENISA Guideline emphasize the importance of protecting sensitive data that is stored on mobile devices. MASVS-STORAGE-1 highlights the need for proper protection of sensitive data stored by apps, regardless of location, which includes managing how the data is handled and potentially encrypted. The ENISA Guideline stresses that in the absence of a standard secure deletion procedure for flash memory, encryption and managing encryption keys securely is vital. Both are concerned with the security of sensitive data at rest on a device’s storage, and the recommended approach for securing such data involves encryption and key management practices.
- MASVS-STORAGE-2: The ENISA guideline’s emphasis on the importance of data encryption and secure key management is tightly correlated with MASVS-STORAGE-2’s focus on preventing unintentional leaks of sensitive data. Both address the risk of sensitive information being exposed due to improper handling or misuse of APIs and system capabilities, and both suggest that proactive measures must be taken by developers to avoid such leaks. In the context of flash memory, since no standard secure deletion exists, encrypting data becomes crucial to prevent exposure if the data is unintentionally stored in an accessible location.

2.10 Implementation Guidance (ENISA 1.10):

ENISA Secure Smartphone Development Guidance (1.10): Consider the security of the whole data lifecycle in writing your application (collection over the wire, temporary storage, caching, backup, deletion, etc.).

2.10.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: MASVS-AUTH-1 is concerned with ensuring that mobile apps follow best practices for secure authentication and authorization when connecting to remote endpoints. This includes considerations for secure protocol use, which is relevant to the security of data transmitted over the wire—a part of the data lifecycle mentioned in the ENISA guideline. The guideline emphasizes the importance of securing data throughout its entire lifecycle, which includes collection (similar to ”connecting to a remote endpoint” as in MASVS-AUTH-1), storage, caching, and deletion, among other aspects. Authentication and authorization practices are a subset of the broader set of considerations necessary for securing the whole data lifecycle as mentioned by ENISA.
- MASVS-AUTH-3: Both ”MASVS-AUTH-3” and the ENISA guidelines emphasize the necessity of implementing robust security measures. ”MASVS-AUTH-3” focuses on the additional forms of authentication for sensitive actions to increase security, which is a part of ensuring safe data handling during user interactions. The ENISA guideline broadens this by addressing the security of the entire data lifecycle, which inherently includes secure collection and transmission, where strong authentication measures can play a crucial role, especially in sensitive actions highlighted in ”MASVS-AUTH-3.” Thus, there is a correlation as both are concerned with secure data handling, albeit at different stages and scopes.
- MASVS-CODE-1: Both ”MASVS-CODE-1” and the ENISA Guideline emphasize the importance of up-to-date security measures throughout different stages. ”MASVS-CODE-1” focuses on ensuring that the app runs on an updated platform, leveraging the latest security protections provided by mobile OS updates, which can include fixes for vulnerabilities that could be exploited during data lifecycle processes such as collection and storage. Meanwhile, the ENISA Guideline advises considering security throughout the entire data lifecycle, which implicitly includes operating within a secure environment provided by the latest platform versions. Both guidelines are convergent in that they advocate for rigorous attention to security at all stages to minimize vulnerabilities.
- MASVS-CODE-2: The correlation between ”MASVS-CODE-2” and the ENISA Guideline is that they both emphasize the importance of maintaining the security of an application throughout its lifecycle. ”MASVS-CODE-2” specifically addresses the need to have a mechanism in place to respond to critical vulnerabilities that are discovered post-production by enforcing updates. This is a form of incident response and remediation to maintain security. The ENISA Guideline advises considering the security of the entire data lifecycle, which includes handling vulnerabilities and ensuring that data remains protected. Both advocate for ongoing security considerations beyond initial development, whether it’s through updating mechanisms or secure data handling practices.
- MASVS-CODE-3: MASVS-CODE-3 from the OWASP Mobile Application Security Verification Standard (MASVS) focuses on ensuring applications use software components without known vulnerabilities, primarily targeting detected vulnerabilities through scan-

ning libraries. The ENISA Guideline emphasizes securing the entire data lifecycle, including collection, transmission, temporary storage, caching, backup, and deletion of data within applications. The correlation between these two lies in the principle of securing applications by recognizing and mitigating potential vulnerabilities in software components (reflected in MASVS-CODE-3) and adhering to a holistic security approach throughout the data lifecycle (as highlighted by ENISA). The need to scan libraries for known vulnerabilities as per MASVS is a part of securing the temporary storage and caching stages within the data lifecycle, which aligns with the ENISA guideline to consider security at all stages. Hence, there is a connection as both guidelines embed the practice of securing app components and data flow from vulnerabilities to enhance overall mobile application security.

- MASVS-CODE-4: The MASVS-CODE-4 requirement that apps should properly verify and sanitize all incoming data relates to the ENISA guideline to consider security throughout the whole data lifecycle. Both concepts emphasize ensuring data integrity and security from the point of collection (data entry) to subsequent processes such as storage, caching, and deletion. MASVS-CODE-4 specifically focuses on treating data as untrusted and properly handling it to prevent security vulnerabilities, which is in line with the ENISA guideline's broader perspective on safeguarding data during all stages of its lifecycle within an application.
- MASVS-CRYPTO-1: The correlation between "MASVS-CRYPTO-1" and the ENISA Guideline about considering the security of the whole data lifecycle is evident in the emphasis on securing user data through cryptographic best practices. "MASVS-CRYPTO-1" specifically mentions the importance of cryptography for protecting user data on mobile devices where physical access by attackers is plausible. It aligns with the ENISA guideline that emphasizes securing data throughout its entire lifecycle, including transmission, storage, and deletion processes. Cryptography is a key tool in ensuring the confidentiality, integrity, and availability of data across all these stages.
- MASVS-CRYPTO-2: The correlation between "MASVS-CRYPTO-2" and the ENISA Guideline is evident, as both refer to the importance of secure data handling throughout its lifecycle. MASVS-CRYPTO-2 emphasizes the need for secure management of cryptographic keys, which includes aspects of their generation, storage, and protection at all stages. The ENISA Guideline broadly covers the security of the data lifecycle, which would inherently include cryptographic key management since keys are essential for ensuring data security during collection, transmission, temporary storage, caching, backup, and deletion. Both guidelines underscore the idea that security should not only focus on the strength of cryptographic methods but also on the secure management of keys and data which they protect.
- MASVS-NETWORK-1: Both the Mobile Application Security Verification Standard (MASVS) "MASVS-NETWORK-1" and the ENISA Guideline focus on the security of data throughout its entire lifecycle, especially during transit. MASVS-NETWORK-1 emphasizes the importance of encrypting data and authenticating the remote endpoint to ensure data privacy and integrity during network communication, which is an integral part of the data lifecycle as mentioned in the ENISA guideline. Both guidelines are concerned with the secure handling of data from the point of collection, during transmission, and through to other stages like storage and deletion.
- MASVS-NETWORK-2: The correlation between "MASVS-NETWORK-2" and the ENISA Guideline about considering the security of the whole data lifecycle is evident. MASVS-NETWORK-2 focuses on ensuring trust in communications by explicitly trusting specific Certificate Authorities (CAs) and implementing certificate or public key pinning. This

directly relates to the ENISA Guideline which mentions securing data during collection over the wire. Certificate pinning enhances the security of data in transit by reducing the risk of man-in-the-middle attacks, which is a crucial aspect of the data lifecycle mentioned by ENISA. Hence, both concepts aim at enhancing the security measures during the initial stages of the data lifecycle: the collection and transmission of data.

- MASVS-PLATFORM-1: Both "MASVS-PLATFORM-1" and the ENISA Guideline emphasize the importance of securing the data and functionality of an app throughout its entire lifecycle, including interactions with other apps or the environment (IPC mechanisms) and stages such as collection, storage, caching, and deletion. MASVS-PLATFORM-1 focuses on secure interactions involving IPC mechanisms, which are integral to the safe handling of data within and across applications. The ENISA Guideline extends the concept by covering security considerations for the entire data lifecycle. Both highlight the necessity to consider security holistically in the context of app development and data management.
- MASVS-PLATFORM-2: MASVS-PLATFORM-2 discusses the secure configuration of WebViews to prevent sensitive data leakage and exposure of sensitive functionality. This aligns with the ENISA guideline that emphasizes considering security throughout the entire data lifecycle, including collection, storage, and deletion. Securely configuring WebViews directly relates to the temporary storage and handling of data within the application, ensuring data is managed securely as it is collected, displayed, and interacted with over the wire.
- MASVS-PLATFORM-3: The correlation exists because both the MASVS-PLATFORM-3 and the ENISA Guideline address the concern of protecting sensitive data throughout its lifecycle within a mobile application. MASVS-PLATFORM-3 specifically mentions protecting sensitive data from unintended leaks via platform mechanisms like auto-generated screenshots, which is a part of the data lifecycle mentioned in the ENISA Guideline—particularly, the display, temporary storage, and potential caching of sensitive data. Both aim to minimize the risk of sensitive information becoming accessible to unauthorized parties through various stages of data handling in a mobile application.
- MASVS-PRIVACY-1: The correlation between "MASVS-PRIVACY-1" and the ENISA Guideline exists as both emphasize the importance of handling user data responsibly throughout the entire data lifecycle. "MASVS-PRIVACY-1" specifically relates to minimizing data access, informed consent, and controlling third-party SDKs in respect to user consent, which aligns with the ENISA Guideline's focus on considering the security of the whole data lifecycle. This includes aspects such as data collection, storage, transmission, and deletion, all of which are integral to ensuring data privacy and security from the moment of collection to deletion. Both guidelines advocate for a comprehensive approach to data handling that enhances privacy and security measures in application development.
- MASVS-PRIVACY-2: The correlation between "MASVS-PRIVACY-2" and the cited ENISA Guideline lies in the overarching goal of ensuring user privacy and security throughout the entire data lifecycle. "MASVS-PRIVACY-2" details the importance of protecting user identity using techniques such as data abstraction, anonymization, and pseudonymization, which aligns with the ENISA Guideline's emphasis on considering the security of data from collection to deletion. Both directives advocate for the protection of user data against identification and tracking, and the concept of handling data securely over its entire lifecycle inherently includes the protection of user identity as highlighted in "MASVS-PRIVACY-2".
- MASVS-PRIVACY-3: Both "MASVS-PRIVACY-3" and the ENISA Guideline emphasize the importance of transparency and management of the data lifecycle. "MASVS-PRIVACY-3" stresses that users have the right to understand how their data is utilized, highlighting

the need for clarity regarding data practices and adherence to platform guidelines. Similarly, the ENISA Guideline focuses on considering the security aspects of the entire data lifecycle, which inherently includes the practices of data collection, storage, sharing, and deletion—all of which are critical in enabling users to have insight into how their data is handled. Hence, there is a correlation as both promote the principle of safeguarding user data and ensuring transparency in its handling throughout its entire lifecycle.

- MASVS-PRIVACY-4: Both the MASVS-PRIVACY-4 and the ENISA Guideline emphasize the importance of user control over their personal data throughout the entire data lifecycle. MASVS-PRIVACY-4 specifically mentions the necessity for users to manage, delete, and modify their data, which aligns with the ENISA recommendation of considering security during various stages including collection, temporary storage, caching, backup, and deletion. Both guidelines aim to enhance user privacy and secure data management in mobile applications.
- MASVS-RESILIENCE-1: While MASVS-RESILIENCE-1 focuses on ensuring that the app is running on an uncompromised platform, thus maintaining the integrity and trustworthiness of the operating system's security features, the ENISA Guideline emphasizes considering the security of the entire data lifecycle within the application. There is a correlation because in both instances, the underlying principle is about safeguarding the app's data integrity and security. If the platform is compromised (as per MASVS-RESILIENCE-1 concern), it can negatively impact the data lifecycle security as emphasized by the ENISA Guideline. Without a secure platform, data collected, transmitted, stored, or deleted by the application could be at risk, which highlights the importance of a trusted foundation for maintaining the overall security throughout the data lifecycle.
- MASVS-RESILIENCE-2: The correlation exists in that both the MASVS-RESILIENCE-2 and the ENISA guideline emphasize the need to protect the integrity and security of data and app functionality throughout their respective lifecycles. MASVS-RESILIENCE-2 is concerned with ensuring that the original code and resources of an app are not modified, thus preserving its intended functionality and preventing security threats such as backdooring. The ENISA guideline advises consideration of security for the entire data lifecycle, which includes protection during collection, transmission, storage, and deletion phases. Implementing security controls for the integrity of the app's functionality implicitly covers several aspects of the data lifecycle, such as preventing unauthorized modifications that could affect data storage and handling. Both share the underlying principle that developers should be vigilant against unauthorized changes and compromises at any stage of the app or data's lifecycle.
- MASVS-STORAGE-1: The description of "MASVS-STORAGE-1" is correlated with the ENISA guideline about considering the security of the whole data lifecycle in writing an application. Both statements emphasize the importance of managing sensitive data securely throughout its existence within an application, from the point of collection, through any form of storage including temporary storage and caching, to backup and deletion. They each highlight the necessity of protecting sensitive data regardless of where or how it is stored within the application's lifecycle.
- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" and the ENISA Guideline is evident as both emphasize the importance of securing sensitive data throughout its entire lifecycle within an application. MASVS-STORAGE-2 focuses on the need to prevent unintentional exposure of sensitive data in publicly accessible locations, which can occur at various stages such as temporary storage, caching, or through system capabilities like backups or logs. The ENISA Guideline similarly advises consideration of security practices for the entire data lifecycle, including data collection, transmission, temporary

storage, caching, backups, and deletion. Both aim to ensure developers implement measures to protect data at all stages, preventing accidental leaks and unauthorized access.

2.11 Implementation Guidance (ENISA 1.11):

ENISA Secure Smartphone Development Guidance (1.11): Ensure that during application removal (uninstall operation), any confidential user data and the corresponding app-specific credentials are deleted from the execution environment, the device, and any other storage medium.

2.11.1 OWASP MASVS MAPPING

- MASVS-CRYPTO-2: The correlation between ”MASVS-CRYPTO-2” and the ENISA Guideline on removing confidential data during app uninstallation is that both of them pertain to the proper handling of sensitive information. ”MASVS-CRYPTO-2” emphasizes the importance of managing cryptographic keys throughout their lifecycle, including during storage and protection. Proper key management would inherently include ensuring that cryptographic keys are not left behind in an insecure state when an application is uninstalled. The ENISA Guideline specifically addresses the need to delete confidential user data and app-specific credentials, which would include cryptographic keys, from all storage mediums during uninstallation to prevent unauthorized access or use of the keys. Both controls aim to protect sensitive data from being compromised.
- MASVS-PRIVACY-1: Both the MASVS-PRIVACY-1 description and the ENISA Guideline focus on protecting user privacy and data security. MASVS-PRIVACY-1 emphasizes the importance of minimal data access, informed consent, and control over third-party SDKs in the data handling process, which correlates with the ENISA Guideline’s requirement to delete confidential user data upon application removal. Both guidelines ensure users’ data is handled responsibly and align with the principles of data minimization and user consent.
- MASVS-PRIVACY-3: Both MASVS-PRIVACY-3 and the ENISA Guideline on application removal focus on protecting user data and ensuring transparency regarding data usage and removal. MASVS-PRIVACY-3 highlights the importance of informing users about how their data is used, implying adherence to data handling practices which should also encompass data deletion upon app uninstallation. The ENISA Guideline specifically addresses the need for confidential user data and credentials to be deleted during the uninstall process, which is in line with the overarching theme of responsible data management outlined in MASVS-PRIVACY-3. Both guidelines aim to enhance user privacy and establish trust by ensuring that user data is not misused or left behind unintentionally.
- MASVS-PRIVACY-4: Both ”MASVS-PRIVACY-4” and the ENISA Guideline emphasize the importance of user control over their data. While MASVS-PRIVACY-4 broadly addresses user control in terms of managing, deleting, modifying data, and privacy settings, the ENISA Guideline provides specific instructions to ensure that confidential data and credentials are deleted upon application uninstallation. Both guidelines are aligned in their objective to protect user privacy by allowing users to manage the lifecycle of their data, including its final deletion.
- MASVS-STORAGE-1: Both ”MASVS-STORAGE-1” and the ENISA Guideline emphasize the protection and proper handling of sensitive data within a mobile application context. ”MASVS-STORAGE-1” is concerned with ensuring that sensitive data is protected no matter where it is stored, indicating a need for secure storage practices and data protection measures. The ENISA Guideline complements this by stating that upon appli-

cation removal, any confidential data and credentials should be deleted. Both guidelines aim to prevent unauthorized access to sensitive information and ensure data privacy and security, with ”MASVS-STORAGE-1” focusing on protection during usage and storage, and the ENISA Guideline focusing on ensuring data is securely removed when the app is uninstalled.

- MASVS-STORAGE-2: The correlation exists between ”MASVS-STORAGE-2” and the ENISA Guideline mentioned. The MASVS-STORAGE-2 focuses on preventing unintentional data leaks which can occur when sensitive data gets stored in publicly accessible locations. This can be because of the misuse of certain APIs or system capabilities. Similarly, the ENISA Guideline emphasizes the importance of ensuring that all confidential user data and app-specific credentials are deleted from all storage mediums upon application uninstallation. Both are concerned with the secure handling of sensitive data to avoid leaks and exposures outside of the intended storage contexts. They both stress developer responsibility in managing the lifecycle of sensitive data, whether during the use or termination (uninstall) of the app.

2.12 Implementation Guidance (ENISA 1.12):

ENISA Secure Smartphone Development Guidance (1.12): Apply the principle of minimal disclosure - only collect and disclose data which is required for business use of the application. Identify in the design phase what data is needed, its sensitivity and whether it is appropriate to collect, store and use each data type.

2.12.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The description of ”MASVS-AUTH-1” is fundamentally related to the principle of minimal disclosure as stated in the ENISA Guideline. Both emphasize the importance of security and privacy considerations in the design and implementation of user authentication and authorization components of mobile apps. ”MASVS-AUTH-1” is about following best practices for secure communication protocols and user authentication to protect sensitive data, which aligns with the ENISA Guideline’s directive of collecting and disclosing only the necessary data for the business use. The security practices in authentication and authorization serve to limit the exposure of user data, which is a direct application of minimal disclosure principles.
- MASVS-AUTH-2: The correlation here is that both MASVS-AUTH-2 and the ENISA Guideline emphasize the importance of proper implementation of authentication mechanisms and the careful consideration of data management. MASVS-AUTH-2 focuses on the security aspects of biometric and local PIN code implementations for user authentication in mobile apps, which involves handling sensitive user authentication data. Similarly, the ENISA Guideline advises on the principle of minimal disclosure for the data collected and disclosed by the application, which includes sensitive data used for business purposes and would encompass authentication data. Both stress the need to consider what sensitive data (such as biometric or PIN code data) is necessary for the functioning of the application while ensuring that its handling is secure and respects user privacy, aligning them in their concern for secure and privacy-conscious data practices in mobile app development.
- MASVS-AUTH-3: While MASVS-AUTH-3 discusses the additional forms of authentication needed for sensitive actions, which implies implementing security features, and the ENISA guideline refers to the principle of minimal disclosure in terms of data collection and disclosure, there is an indirect correlation between the two. The rationale is that both focus on the security and privacy aspects of mobile application design. MASVS-AUTH-3’s push for secure implementation of authentication methods supports the protection of sensitive user data, which aligns with ENISA’s guideline to minimize data disclosure and only use data necessary for the application’s business purposes. Implementing secure authentication methods is part of ensuring that sensitive data is not unnecessarily exposed, thus adhering to the principle of minimal disclosure.
- MASVS-CODE-4: The correlation between ”MASVS-CODE-4,” which focuses on proper verification and sanitization of incoming data from various entry points to prevent injection attacks and other vulnerabilities, and the ENISA Guideline on applying the principle of minimal disclosure is founded on the shared principle of data security and minimal exposure. Both concepts aim to protect sensitive data, albeit through different means: MASVS-CODE-4 emphasizes active sanitization and validation of input data to prevent exploitation, while the ENISA guideline advocates for the minimization of data collection and disclosure to reduce the potential attack surface and limit exposure of sensitive infor-

mation. Thus, they complement each other in enhancing the overall security posture of the application.

- MASVS-CRYPTO-1: Both ”MASVS-CRYPTO-1” and the ENISA guideline emphasize the protection of user data, with a focus on best practices and minimal necessary usage. ”MASVS-CRYPTO-1” highlights the importance of cryptography in securing user data, especially when physical access to a device by attackers is possible. The ENISA guideline complements this by advocating for minimal data collection and usage, which inherently reduces the amount of data that needs to be protected via cryptographic measures. Both are aimed at limiting the exposure of sensitive user information and employing prudent security measures during the design and application phases.
- MASVS-CRYPTO-2: While MASVS-CRYPTO-2 specifically addresses the lifecycle management of cryptographic keys, including their generation, storage, and protection, the ENISA guideline focuses on the principle of minimal disclosure concerning the collection and handling of data necessary for the application's business use. The correlation lies in the foundational principle of data minimization and security that both embrace. Proper cryptographic key management ensures that sensitive data protected by encryption remains confident and accessible only to authorized parties. Adhering to the ENISA guideline ensures that only necessary data is collected and stored, reducing the potential impact of a data breach or improper data handling. Both contribute to the overall security posture by minimising the risks associated with data processing and management.
- MASVS-NETWORK-1: Both ”MASVS-NETWORK-1” and the ENISA Guideline emphasize the importance of protecting data. ”MASVS-NETWORK-1” focuses on ensuring data privacy and integrity for data in transit through encryption and secure connections, which directly relates to protecting collected and disclosed data, as stated by the ENISA guideline about minimal disclosure and handling of sensitive data. Both aim to ensure the security and privacy of user data within apps, and while they focus on different aspects (data in transit versus data collection), they share the common goal of minimizing the risk of unauthorized data exposure and enhancing data security practices.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the indicated ENISA guideline centers on the concept of secure data handling and controlled data exposure. MASVS-PLATFORM-1 highlights the importance of secure interactions involving IPC (Inter-Process Communication) mechanisms in mobile applications, emphasizing that data and functionality exposed through these avenues should be done so safely. This aligns with the ENISA principle of minimal disclosure, which advises only collecting and disclosing data necessary for business use. Both focus on ensuring that only required data is available and that it is securely managed and disclosed, thus preventing unwanted or insecure access to sensitive information.
- MASVS-PLATFORM-2: The correlation between ”MASVS-PLATFORM-2” and the ENISA guideline regarding minimal disclosure exists as both are focused on ensuring sensitive data is protected. MASVS-PLATFORM-2 speaks to configuring WebViews securely to prevent sensitive data leakage, which aligns with the ENISA guideline's aim to collect and disclose only the data required for business use, highlighting the focus on data sensitivity and appropriateness of data handling. Both are concerned with the principle of least privilege and minimal exposure of sensitive information.
- MASVS-PLATFORM-3: Both the Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-PLATFORM-3” and the ENISA Guideline on minimal disclosure relate to the protection of sensitive data within mobile applications. MASVS-PLATFORM-3 focuses on ensuring that sensitive data is not inadvertently leaked through platform mechanisms, such as screenshots or shoulder surfing. The ENISA guideline

emphasizes the principle of minimal disclosure, which aligns with the concept of only displaying necessary sensitive data in the UI, ensuring that data collection, storage, and use are appropriate and limited to what is required for business purposes. Both guidelines aim to minimize the exposure and potential misuse of sensitive information.

- MASVS-PRIVACY-1: Both ”MASVS-PRIVACY-1” and the ENISA Guideline emphasize the importance of data minimization, which entails only requesting, collecting, and sharing data that is strictly necessary for the app’s functionality. Both stress the importance of user consent before accessing data, and highlight the responsibility of app developers to manage data access and sharing deliberately and securely, with a focus on minimizing potential privacy risks.
- MASVS-PRIVACY-2: The correlation between ”MASVS-PRIVACY-2” and the ENISA Guideline on ”minimal disclosure” is that both emphasize the importance of protecting user identity and limiting the exposure of personal information. MASVS-PRIVACY-2 suggests techniques like data abstraction, anonymization, and pseudonymization to prevent user identification and tracking, which aligns with the ENISA principle of collecting only the data that is necessary for the business use of the application. Both advocate for the careful consideration of data use and the implementation of measures to ensure that user data is not misused or unnecessarily exposed.
- MASVS-PRIVACY-3: Both ”MASVS-PRIVACY-3” and the ENISA Guideline stress the importance of transparency and minimal data collection. MASVS-PRIVACY-3 focuses on user awareness of data use, while ENISA underscores collecting only necessary data. They correlate by advocating for user rights and responsible data handling.
- MASVS-PRIVACY-4: Both ”MASVS-PRIVACY-4” and the ENISA Guideline emphasize the principle of minimal data disclosure and user control over their personal data. MASVS-PRIVACY-4 stresses that users should have mechanisms to manage their data and that apps should obtain consent when more data is required than initially specified. The ENISA Guideline similarly advocates for collecting only the data necessary for the app’s business use, determining the sensitivity of the data, and deciding on the appropriateness of collecting, storing, and using each type of data in the design phase. Both guidelines are aligned in their objective to ensure that users’ privacy is respected by minimizing data collection and providing transparency and control over personal data.
- MASVS-STORAGE-1: Both ”MASVS-STORAGE-1” and the ENISA guideline emphasize the importance of handling sensitive data with care. ”MASVS-STORAGE-1” focuses on ensuring that any sensitive data stored by an app is adequately protected, regardless of the storage location. The ENISA guideline advises on minimal data disclosure, suggesting that data collection and sharing should be limited to what is necessary for the application’s business use and that data’s sensitivity should be considered in the design phase. Both place a strong emphasis on the security and minimal disclosure of sensitive data collected and stored by mobile applications.
- MASVS-STORAGE-2: The reference to ”MASVS-STORAGE-2” pertains to precautionary measures regarding the unintended or accidental exposure of sensitive data due to the use of certain APIs or system capabilities. The principle of minimal disclosure recommended by the ENISA guideline aligns with the intention behind ”MASVS-STORAGE-2” by suggesting that developers should identify and minimize the collection, storage, and usage of sensitive data to what is strictly necessary for the application’s business use. Both highlight the need for proactive measures to prevent the exposure of information that may lead to privacy breaches or security issues.

2.13 Implementation Guidance (ENISA 1.13):

ENISA Secure Smartphone Development Guidance (1.13): Use non-persistent identifiers which are not shared with other apps wherever possible (e.g., do not use the device unique hardware identifiers such as IMEI or UDID as an identifier).

2.13.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the described ENISA Guideline is that both are concerned with the secure handling of user authentication and identifier management within an app. MASVS-AUTH-1 refers to the implementation of authentication and authorization best practices, which implicitly includes using secure and non-persistent identifiers that minimize privacy risks and reduce the chance of unauthorized tracking or access. The ENISA Guideline specifically advises against using persistent unique hardware identifiers, which aligns with the principles outlined in MASVS-AUTH-1 regarding the importance of securing user authentication protocols.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the ENISA guideline about using non-persistent identifiers is based on the principle of minimizing the exposure of sensitive data. MASVS-CRYPTO-1 emphasizes the importance of cryptography for securing user data on mobile devices where physical access by attackers is more likely. The use of strong cryptography would be part of ensuring that identifiers, such as IMEI or UDID, are not easily accessible or misused if stored. Similarly, the ENISA guideline advises against using persistent, unique hardware identifiers that can be accessed by other apps, as this could lead to tracking or compromising the user’s privacy. Both stress on best practices to protect sensitive user data, with MASVS-CRYPTO-1 providing a general blanket statement about cryptography and ENISA providing a specific example within that realm.
- MASVS-CRYPTO-2: The correlation between ”MASVS-CRYPTO-2” and the ENISA Guideline is that both emphasize the importance of managing sensitive information securely. ”MASVS-CRYPTO-2” focuses on the management of cryptographic keys to ensure that strong cryptography is not compromised by key management issues. The ENISA Guideline addresses the management of identifiers, recommending the use of non-persistent and non-shared identifiers to protect user privacy and security. Both guidelines are aimed at preventing unauthorized access and ensuring that sensitive data, whether it be cryptographic keys or device identifiers, are handled in a secure manner to protect the overall security of the system.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the ENISA Guideline about using non-persistent identifiers is that both are concerned with the secure interaction between applications and platform components. MASVS-PLATFORM-1 emphasizes securely using IPC (Inter-Process Communication) mechanisms provided by the platform, which would include how identifiers are shared and utilized between apps and system components. The ENISA guideline advises against using persistent, unique identifiers that could be shared across apps, potentially leading to tracking or exposing user identity. Both are aligned in their aim to enhance privacy and security in app interactions.
- MASVS-PLATFORM-2: While ”MASVS-PLATFORM-2” does not explicitly mention the use of non-persistent identifiers, it is related to ensuring that WebViews are configured securely to prevent sensitive data leakage. The ENISA guideline’s advice to use non-persistent

identifiers that are not shared with other apps is aligned with preventing data leakage, which can include leakage of hardware identifiers like IMEI or UDID. Both statements share a common goal of protecting sensitive information, and secure configuration of WebViews can be one aspect of not exposing device unique identifiers.

- MASVS-PRIVACY-1: The description of ”MASVS-PRIVACY-1” aligns with the ENISA Guideline as both advocate for minimal access to user data and informed consent. MASVS-PRIVACY-1 emphasizes data minimization, user consent, and cautious sharing with third parties, which correlates with the ENISA recommendation of using non-persistent identifiers and avoiding sharing device-unique identifiers with other apps. Both guidelines prioritize user privacy and limiting unnecessary data access and sharing.
- MASVS-PRIVACY-2: Both the MASVS-PRIVACY-2 description and the ENISA Guideline emphasize the importance of preserving user identity by avoiding the use of persistent, unique, and identifying information. MASVS-PRIVACY-2 talks about unlinkability techniques to protect user identification and tracking, which aligns with the ENISA guideline's recommendation to use non-persistent identifiers and avoid unique hardware identifiers shared across different apps. Both aim to enhance user privacy by implementing measures that prevent persistent tracking and identification.
- MASVS-PRIVACY-3: Both MASVS-PRIVACY-3 and the ENISA Guideline emphasize the importance of transparent and ethical handling of user data. MASVS-PRIVACY-3 requires that users be informed about how their data is used, including data collection, storage, and sharing practices. The ENISA guideline complements this by recommending the use of non-persistent identifiers that are not shared with other apps, thereby avoiding the use of unique hardware identifiers that could compromise user privacy. Both guidelines are concerned with protecting user privacy by controlling how personally identifiable information is collected and used.
- MASVS-PRIVACY-4: The correlation between ”MASVS-PRIVACY-4” and the ENISA guideline mentioned is present because both focus on safeguarding user data privacy. ”MASVS-PRIVACY-4” emphasizes the need for users to have control over their data, which includes being able to manage their identifiers and consent. The ENISA guideline specifically advises against using persistent identifiers that can be exploited for tracking purposes, thereby supporting the principle that users should control how their data is managed and shared, in alignment with ”MASVS-PRIVACY-4”. Both stress that an app should minimize the potential for privacy infringement by limiting the use of identifiers that could compromise user data control and privacy.
- MASVS-STORAGE-1: Both ”MASVS-STORAGE-1” and the described ENISA Guideline focus on the proper handling and protection of sensitive data on mobile devices. ”MASVS-STORAGE-1” emphasizes the secure storage of sensitive data in appropriate locations, ensuring protection regardless of whether the data is in private or public areas. The ENISA Guideline advises against using persistent, globally unique identifiers that can be accessed by other apps on the device, such as IMEI or UDID. Both are concerned with preventing unintended access and potential misuse of sensitive information by ensuring that it is managed in a secure and controlled manner.
- MASVS-STORAGE-2: The control MASVS-STORAGE-2, which involves preventing unintentional storage or exposure of sensitive data, correlates with the ENISA guideline advising against using persistent identifiers like IMEI or UDID. This is because using such identifiers could lead to sensitive data leaks if they are stored or logged inadvertently by the app, which MASVS-STORAGE-2 aims to mitigate. Hence, both are concerned with avoiding unintended disclosure of sensitive information.

2.14 Implementation Guidance (ENISA 1.14):

ENISA Secure Smartphone Development Guidance (1.14): Applications on managed devices should leverage remote wipe and kill switch APIs to remove sensitive information from the device in the event of theft or loss.

2.14.1 OWASP MASVS MAPPING

- MASVS-CODE-1: The correlation is that both MASVS-CODE-1 and the ENISA Guideline focus on ensuring the security of mobile applications and data on devices. MASVS-CODE-1 emphasizes running apps on up-to-date platform versions to make use of the latest security protections. The ENISA Guideline advises leveraging remote wipe and kill switch APIs on managed devices to protect sensitive information in case of device theft or loss. Both sets of guidance are concerned with mitigating risks associated with known vulnerabilities, though they focus on different aspects of security—system updates in the case of MASVS and data protection controls for ENISA.
- MASVS-CODE-2: Both ”MASVS-CODE-2” and the ENISA Guideline refer to security measures that can be taken to mitigate risks associated with post-deployment scenarios. While ”MASVS-CODE-2” specifically focuses on forcing users to update the application to address vulnerabilities that arise after the app is in production, the ENISA Guideline addresses the capability to remotely wipe or disable the application in cases such as theft or loss of the device. Both measures are proactive controls intended to protect sensitive information and maintain the security of the application in response to potentially harmful events. Although ”MASVS-CODE-2” is about updates and the ENISA Guideline is about remote wipe/kill switch, they correlate in their overall aim to safeguard apps and data in dynamic real-world scenarios where security threats might manifest.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the specified ENISA guideline is that both address the need to protect sensitive user data on mobile devices, especially in the event of physical access by unauthorized parties such as through theft or loss. ”MASVS-CRYPTO-1” mentions cryptography as a means to secure user data, which is relevant since adequately implemented cryptographic measures can ensure that data remains confidential and tamper-proof. The ENISA guideline suggests that applications use remote wipe and kill switch APIs to remove sensitive information if a device is compromised. Both of these recommendations aim to mitigate the risks associated with unauthorized physical access to a device by ensuring that sensitive data is either encrypted or can be securely erased. Therefore, there is a correlation between the two as they both serve the purpose of protecting sensitive data on mobile devices under the threat model of physical compromise.
- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the ENISA guideline on leveraging remote wipe and kill switch APIs exists in the realm of trust and protection of the app’s environment and data. MASVS-RESILIENCE-1 highlights the importance of running an app on a secure platform that hasn’t been tampered with, as a compromised OS may disable security features that are crucial for protecting data. Similarly, the ENISA guideline suggests using remote wipe and kill switch APIs to safeguard sensitive information in the event of device theft or loss. Both are concerned with mitigating risks that could compromise app data if the device’s security is breached, albeit in different contexts:

one is through maintaining OS integrity, and the other through proactive countermeasures in case of a security incident.

- MASVS-STORAGE-1: Both "MASVS-STORAGE-1" and the described ENISA Guideline address the protection of sensitive data on mobile devices. "MASVS-STORAGE-1" focuses on ensuring that sensitive data is stored securely regardless of its location, which implies proper data handling and storage practices to safeguard information. The ENISA Guideline complements this by prescribing that in the event of a device theft or loss, sensitive data should be removable through remote wipe and kill switch APIs, highlighting a method of protection post-incident. Both are concerned with mitigating the risk associated with sensitive data storage on mobile devices.
- MASVS-STORAGE-2: The correlation exists in the context of managing sensitive data and preventing unintentional exposure. "MASVS-STORAGE-2" emphasizes the importance of avoiding unintentional storage or exposure of sensitive data by leveraging proper APIs and practices. Similarly, the ENISA guideline advises using remote wipe and kill switch APIs to safeguard sensitive information on managed devices in case of theft or loss. Both advocate for proactive control measures by developers to protect sensitive data, addressing different aspects of data exposure risks. While MASVS-STORAGE-2 focuses on preventing leaks through proper use of APIs and understanding system capabilities, the ENISA guideline highlights remediation measures in an adverse event but both contribute to the overall security posture regarding data management on devices.

2.15 Implementation Guidance (ENISA 1.15):

ENISA Secure Smartphone Development Guidance (1.15): Application developers may want to incorporate an application-specific ”data kill switch” into their products, to allow the per-app deletion of their application’s sensitive data when needed (strong authentication is required to protect misuse of such a feature).

2.15.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: Although ”MASVS-AUTH-1” and the ENISA guideline pertain to different aspects of app security, they both emphasize the need for strong security controls around authentication and authorization mechanisms. ”MASVS-AUTH-1” focuses on the secure use of authentication and authorization protocols in apps to protect access to remote endpoints, while the ENISA guideline addresses the need to secure a feature (data kill switch) that could potentially be misused if not protected by strong authentication. Both stress the importance of ensuring that authentication is not only properly enforced but also protected from misuse, reflecting a correlation in the underlying principle of securing sensitive actions and features within apps.
- MASVS-AUTH-2: The correlation exists in that both MASVS-AUTH-2 and the ENISA Guideline emphasize the importance of implementing proper authentication mechanisms to protect sensitive data. MASVS-AUTH-2 discusses the need for correctly implementing biometric or local PIN code authentication, which are methods to ensure that only the legitimate user can access the app’s data or functions. Similarly, the ENISA Guideline suggests incorporating a ”data kill switch” that requires strong authentication to prevent misuse. Both guidelines focus on the security of the authentication process to safeguard sensitive data within mobile applications.
- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3,” which emphasizes the need for additional forms of secure authentication for sensitive actions within the app, and the ENISA guideline, which suggests incorporating a ”data kill switch” that should be protected by strong authentication to prevent misuse, is present. Both are concerned with enhancing the security of sensitive operations within an application through the use of additional secure authentication mechanisms.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the ENISA guideline about the ”data kill switch” is that both are focused on the protection and secure management of user data, especially in scenarios where physical access to the device is possible. ”MASVS-CRYPTO-1” emphasizes the importance of cryptography best practices in securing user data, while the ENISA guideline suggests an additional feature - an application-specific ”data kill switch” which would allow users to delete sensitive data when necessary. Implementing such a feature would require strong authentication to prevent misuse, aligning with the concept of using robust cryptographic measures to ensure data security and privacy.
- MASVS-CRYPTO-2: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-CRYPTO-2” and the quoted ENISA Guideline both emphasize secure management of sensitive data within mobile applications. MASVS-CRYPTO-2 focuses on the importance of managing cryptographic keys properly as part of their lifecycle to ensure that encryption remains effective. The ENISA guideline suggests that app developers should add a mechanism (data kill switch) that can securely delete sensitive data

when necessary—as would be the case for cryptographic keys—to prevent misuse or compromise. This function requires strong authentication to prevent unauthorized use, aligning with MASVS-CRYPTO-2's emphasis on secure management and protection of keys. Both guidelines support the broader principle of ensuring data security through proper handling and control mechanisms.

- MASVS-PRIVACY-2: The correlation between "MASVS-PRIVACY-2" and the description from the ENISA Guideline about incorporating a "data kill switch" is that both aim to enhance user privacy and control over personal data. MASVS-PRIVACY-2 discusses techniques such as data abstraction, anonymization, and pseudonymization to prevent user identification and tracking, which aligns with the concept of allowing users to manage the deletion of their sensitive data through a "data kill switch". Furthermore, the emphasis on establishing technical barriers for sensitive data streams, as mentioned in MASVS-PRIVACY-2, complements the need for strong authentication in the ENISA Guideline to prevent misuse of the "data kill switch". Both controls serve to empower user privacy by providing mechanisms to manage and secure sensitive information.
- MASVS-PRIVACY-3: Both "MASVS-PRIVACY-3" and the ENISA guideline emphasize the importance of users having control over how their data is managed. MASVS-PRIVACY-3 underlines the necessity for transparency in data use, as well as user awareness and consent to any unexpected data collection practices. The ENISA guideline takes this a step further by suggesting that developers provide a mechanism ("data kill switch") for users to proactively manage the deletion of their sensitive data. This feature would provide users with more direct control and aligns with the concept of user rights and transparency indicated in MASVS-PRIVACY-3, though it is more specific about the means of granting users control over their data.
- MASVS-PRIVACY-4: The correlation exists because both MASVS-PRIVACY-4 and the ENISA guideline emphasize the importance of user control over their data. MASVS-PRIVACY-4 specifically mentions that users should have mechanisms to manage, delete, and modify their data—similar to the ENISA guideline's recommendation of incorporating a "data kill switch" that enables users to delete an application's sensitive data. Both also imply the need for user consent and secure authentication to protect against misuse.
- MASVS-RESILIENCE-1: The MASVS-RESILIENCE-1 control which emphasizes the importance of running on an uncompromised platform correlates with the ENISA guideline on incorporating a "data kill switch". Both focus on protecting the app's sensitive data under adverse conditions. MASVS-RESILIENCE-1 is concerned with ensuring the underlying platform's integrity so that the app's security features remain effective, while the ENISA guideline suggests an active measure (kill switch) that can be triggered to protect data if the underlying platform's security is breached. Hence, both are mechanisms aimed at safeguarding sensitive data albeit from different angles—one preventive by ensuring a secure platform, the other reactive by providing a method to mitigate the impact of a security failure.
- MASVS-STORAGE-1: The concept of "MASVS-STORAGE-1" relates to the secure handling and storage of sensitive data by mobile apps, ensuring data is properly protected regardless of the storage location. The ENISA guideline's recommendation to incorporate a "data kill switch" is also focused on the secure handling of sensitive data by allowing for its deletion under certain conditions. Both the MASVS standard and the ENISA guideline are concerned with the protection of sensitive data within mobile applications, therefore showing a correlation between the two.
- MASVS-STORAGE-2: The correlation exists. "MASVS-STORAGE-2" covers instances where sensitive data might inadvertently be stored or exposed in ways that are accessible,

due to usage of certain APIs or system features that might unintentionally leak information if not handled properly by the developer. The ENISA guideline recommends the inclusion of an application-specific ”data kill switch” to enable the deletion of sensitive data from the app when necessary. Both are concerned with the proper handling of sensitive data within mobile applications, with the emphasis on preventing unintentional data exposure. The MASVS requirement deals with avoiding leaks in the first place, while the ENISA guideline provides a remediation measure if sensitive data is present on the device. Both measures are complementary: one focuses on prevention while the other on mitigation.

2.16 Implementation Guidance (ENISA 1.16):

ENISA Secure Smartphone Development Guidance (1.16): Do not leak permission-protected data to other applications. This occurs when specific permissions are required to access the data, however an app that has been granted these permissions makes the data available to all other apps without restrictions (e.g., over IPC).

2.16.1 OWASP MASVS MAPPING

- MASVS-PLATFORM-1: The MASVS-PLATFORM-1 requirement and the ENISA Guideline both emphasize the secure handling of interprocess communication (IPC) mechanisms. MASVS-PLATFORM-1 requires secure interactions involving IPC mechanisms, implying that permission-protected data should not be exposed improperly. The ENISA Guideline specifically warns against leaking permission-protected data via IPC mechanisms to other applications without proper restrictions. Both statements are focused on preventing unauthorized or insecure data sharing through IPC, demonstrating a clear correlation between them.
- MASVS-PLATFORM-2: The MASVS-PLATFORM-2 guideline about securely configuring WebViews to prevent sensitive data leakage and functionality exposure correlates with the ENISA guideline about not leaking permission-protected data to other applications. Both guidelines concern the need to prevent the unauthorized access and sharing of sensitive or permission-protected data to preserve the app's security integrity. MASVS-PLATFORM-2 focuses on secure configuration of WebViews, which could be a vector for data leakage, while the ENISA guideline directly addresses the issue of an application that has certain permissions leaking data it has access to other applications, potentially through improperly secured IPC mechanisms. Both are concerned with the broader theme of controlling and protecting sensitive data within mobile apps.
- MASVS-PLATFORM-3: The correlation between "MASVS-PLATFORM-3" and the ENISA guideline is that both are concerned with preventing the unintentional leakage of sensitive data. "MASVS-PLATFORM-3" is focused on ensuring that sensitive data displayed in the UI, such as passwords or credit card details, does not get leaked through platform mechanisms or user behaviors. Similarly, the ENISA guideline addresses the issue of apps with special permissions making sensitive data available to other apps without proper restrictions. Although the specifics of how data might be leaked differ, the underlying principle of protecting sensitive data from unauthorized access or exposure is a common theme in both statements.
- MASVS-PRIVACY-1: The correlation between "MASVS-PRIVACY-1" and the ENISA guideline is clear. Both are oriented towards ensuring that an app only accesses the data it needs, with user consent, and that it does not inadvertently expose or share this data with other applications or parties without restrictions. The MASVS-PRIVACY-1 emphasizes data minimization and informed consent, similar to the ENISA guideline's focus on preventing the leak of permission-protected data to other applications. This common goal supports privacy and security best practices by controlling access to sensitive data and preventing unauthorized data sharing, aligning with principles of least privilege and responsible data handling.
- MASVS-PRIVACY-2: The correlation exists because both MASVS-PRIVACY-2 and the ENISA guideline address the principle of safeguarding user privacy. MASVS-PRIVACY-2

emphasizes the use of techniques to prevent user identification and tracking, ensuring that data streams are isolated and used only for their intended purpose. Similarly, the ENISA guideline warns against the leaking of permission-protected data to other applications, which could also lead to user identification or tracking. Both stress the importance of data protection mechanisms to maintain user privacy, with MASVS-PRIVACY-2 focusing on the broader perspective of unlinkability and the ENISA guideline concentrating on the specific case of inter-app data sharing security.

- MASVS-PRIVACY-3: The correlation between "MASVS-PRIVACY-3" and the ENISA guideline is clear: both emphasize the necessity of protecting user data and ensuring that apps are transparent and responsible about data handling. MASVS-PRIVACY-3 underlines users' right to know about data usage, highlighting the requirement for clarity around data practices, which aligns with the ENISA guideline's warning against the unrestricted sharing of data with other applications, particularly permission-protected data. This reflects a shared objective of both guidelines to prevent unexpected or unauthorized data distribution, thus preserving user privacy.
- MASVS-PRIVACY-4: MASVS-PRIVACY-4 and the ENISA Guideline both emphasize the importance of users being in control of their data and restricting unauthorized access. While MASVS-PRIVACY-4 focuses more broadly on user control over data and consent, the ENISA guideline addresses a specific case where data should not be leaked to other applications without proper permission. The concepts are related in that they both aim to protect user data from being accessed or manipulated without consent, providing a layer of security and privacy control. MASVS-PRIVACY-4's description encompasses the principles of the ENISA guideline within a broader context of user data control and privacy settings.
- MASVS-STORAGE-1: The Mobile Application Security Verification Standard (MASVS) requirement "MASVS-STORAGE-1" and the ENISA Guideline mentioned both emphasize the importance of protecting sensitive data from unauthorized access. MASVS-STORAGE-1 requires that sensitive data, regardless of its source or storage location, is properly protected by the app. This aligns with the ENISA Guideline's caution against leaking permission-protected data to other applications, which would occur if an app with proper permissions inappropriately shares that data with other apps that do not have such permissions. Both guidelines are focused on ensuring that sensitive data is not exposed to unauthorized entities, indicating a correlation between the two standards.
- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" and the ENISA guideline is that both are concerned about preventing sensitive data from being unintentionally exposed or leaked to locations or apps that should not have access to it. MASVS-STORAGE-2 addresses the unintentional storage or exposure of sensitive data in publicly accessible areas due to the use of certain APIs or system capabilities, suggesting the developer should prevent these leaks. The ENISA guideline warns against leaking permission-protected data to other applications, which can happen when an app with the required permissions shares the data irresponsibly. Both guidelines are focused on ensuring sensitive data remains protected and only accessible to authorized parties.

2.17 Implementation Guidance (ENISA 1.17):

ENISA Secure Smartphone Development Guidance (1.17): Restrict the data that is shared with other applications (e.g., by implementing an Android Content Provider). This can be accomplished using fine-grained permissions (ensure permissions are protected using signature protection level on Android).

2.17.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between "MASVS-AUTH-1" and the described ENISA guideline is that both aim to enhance the security of data and user credentials within mobile applications. "MASVS-AUTH-1" emphasizes the importance of secure authentication and authorization mechanisms, which should be enforced on the remote endpoint and followed by best practices in the app. The ENISA guideline suggests restricting the sharing of data with other applications and using fine-grained permissions to achieve this security enhancement. Implementing secure content providers and permissions management is indeed part of the best practices to secure the communication protocols and sensitive data which is what MASVS-AUTH-1 is suggesting. Both guidelines are concerned with mitigating unauthorized access and protecting user data within the app ecosystem.
- MASVS-CODE-4: The "MASVS-CODE-4" description emphasizes treating all incoming data—from the UI, IPC, network, file system, and other entry points—as untrusted input that must be verified and sanitized. This correlates with the ENISA guideline of restricting the data shared with other applications and implementing fine-grained permissions to protect this data, as it aligns with the principle of reducing the attack surface by carefully controlling access to data and ensuring that any data that is accessed by other applications is treated as untrusted, verified, and sanitized to prevent security vulnerabilities such as injection attacks. Both relate to enhancing the security of the application by managing how data is exposed to and from other applications.
- MASVS-CRYPTO-2: Both relate to security and protection measures. By ensuring proper key management as noted in MASVS-CRYPTO-2, it contributes to the overall security posture which restricts data sharing with other applications as advised by the ENISA Guideline, even if the two statements approach the topic from different angles.
- MASVS-NETWORK-1: The correlation between "MASVS-NETWORK-1" and the ENISA Guideline exists in the context of information security principles for mobile applications. Both address the need to protect data privacy and integrity. "MASVS-NETWORK-1" emphasizes the importance of securing data in transit, particularly through encryption and authentication to prevent compromise when data is communicated over a network. The ENISA Guideline focuses on data sharing restrictions between applications and recommends fine-grained permissions to control access to shared data. While "MASVS-NETWORK-1" deals with network communications, and the ENISA Guideline with inter-app data sharing, they both serve the broader purpose of safeguarding sensitive information within the mobile app ecosystem. Implementing network security controls and restricting data shared with other applications are complementary strategies that contribute to overall data protection within mobile apps.
- MASVS-PLATFORM-1: The Mobile Application Security Verification Standard (MASVS) requirement "MASVS-PLATFORM-1" which focuses on secure interactions involving Inter-Process Communication (IPC) mechanisms and the ENISA Guideline advising to re-

strict data shared with other applications via mechanisms like Android Content Providers with fine-grained permissions are correlated. Both emphasize the need for security in app components that expose data or functionality to other apps or the user, ensuring secure sharing and interaction through IPC. MASVS-PLATFORM-1 pertains to securing these interactions as a whole, and the ENISA Guideline provides a specific method for achieving this on the Android platform through permission restrictions and signature protection levels.

- MASVS-PLATFORM-2: The correlation between ”MASVS-PLATFORM-2” and the ENISA guideline exists because both concern the protection of sensitive data within mobile applications. ”MASVS-PLATFORM-2” focuses on ensuring that WebViews are configured securely to prevent data leakage and unwanted exposure of sensitive functionality, such as interaction with native code through JavaScript bridges. This is about controlling how the application presents and manages web content within the app, an essential aspect when web content can interact with the app and potentially access sensitive information. The ENISA guideline emphasizes restricting the data shared with other applications, which can be achieved by using fine-grained permissions, and ensuring those permissions are adequately protected, especially on Android, with signature protection levels. Although this guideline is more about inter-app communication and data sharing, this intersects with ”MASVS-PLATFORM-2” in the context of securing data within the app environment and controlling which parts of the app (or other apps) can access sensitive data, either through UI elements like WebViews or through inter-app communication mechanisms like Android Content Providers. Both are about securing different aspects of app data handling to minimize the potential for sensitive data leakage.
- MASVS-PLATFORM-3: The correlation between ”MASVS-PLATFORM-3” and the described ENISA Guideline is that both are concerned with the protection of sensitive data within the app ecosystem. MASVS-PLATFORM-3 focuses on ensuring sensitive data is not leaked through platform mechanisms such as screenshots or shoulder surfing—essentially addressing inadvertent data exposure. The ENISA Guideline advises on restricting data shared with other applications and using fine-grained permissions to protect data—targeting intentional or unauthorized data sharing. Both guidelines are aimed at preventing sensitive data from being compromised, but approach the problem from different angles; one is more user-centric and the other is system-centric. However, they both contribute to a broader strategy of secure data handling within mobile applications.
- MASVS-PRIVACY-1: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-PRIVACY-1” and the ENISA Guideline both emphasize the principle of data minimization and controlled access to data within mobile applications. They both advise restricting access to only the data that the application absolutely needs for its functionality, and doing so with user consent. Additionally, they both mention the need for careful management of shared data with third parties, such as through the use of fine-grained permissions or their equivalent. The MASVS text specifically mentions the need for third-party SDKs to operate based on user consent and to be cautious about the data shared down their ”supply chain” of dependencies, which is conceptually in line with ENISA’s recommendation to restrict the data shared with other applications and leveraging signature-level protection for permissions on Android. The focus on limiting data access by both sources shows a clear correlation in their guidance for enhancing privacy and security in mobile apps.
- MASVS-PRIVACY-2: The correlation between ”MASVS-PRIVACY-2” and the ENISA guideline about restricting data shared with other applications is apparent as both emphasize the protection of user data. MASVS-PRIVACY-2 advocates for unlinkability techniques

to protect user identity, which aligns with the ENISA guideline's recommendation to use fine-grained permissions and restrict data sharing. Implementing such restrictions and permissions on Android, for example, contributes to the creation of technical barriers mentioned in MASVS-PRIVACY-2, serving to prevent the misuse of sensitive data like device IDs or behavioral patterns, thereby enhancing user privacy.

- MASVS-PRIVACY-3: Both the MASVS-PRIVACY-3 guideline and the ENISA Guideline highlight the importance of user awareness and control over their data. MASVS-PRIVACY-3 focuses on informing users about data practices and adhering to data declaration guidelines, while the ENISA Guideline provides a method (restricting data shared with other apps using fine-grained permissions) that supports the principle of giving users knowledge and control over their data as described in MASVS-PRIVACY-3. Both emphasize protecting user data and ensuring users are informed about how their data is used.
- MASVS-PRIVACY-4: The Mobile Application Security Verification Standard (MASVS) PRIVACY-4 guideline and the ENISA guideline both emphasize the principle of user control and data protection. MASVS-PRIVACY-4 focuses on providing users with mechanisms to manage their data and privacy settings, including re-prompting for consent and updating transparency disclosures. The ENISA guideline complements this by suggesting technical measures such as restricting data sharing with other applications and using fine-grained permissions to achieve this level of user control and privacy. Both guidelines ultimately aim to enhance user privacy and control over personal data within mobile applications.
- MASVS-STORAGE-1: The correlation exists because both "MASVS-STORAGE-1" and the ENISA guideline emphasize the secure handling of sensitive data by restricting access and ensuring proper protection measures. "MASVS-STORAGE-1" mentions the importance of protecting sensitive data regardless of the storage location, including data shared with other applications. The ENISA guideline specifically advises on restricting data shared with apps and using fine-grained permissions for better control over access, aligning with the principle of protecting sensitive data as outlined in "MASVS-STORAGE-1". Both sources are concerned with safeguarding sensitive data within the app ecosystem.
- MASVS-STORAGE-2: The ENISA Guideline's recommendation to restrict data shared with other applications, including through the use of fine-grained permissions on Android, correlates with the principle outlined in MASVS-STORAGE-2. The MASVS-STORAGE-2 requirement addresses the risk of sensitive data being unintentionally stored or exposed in publicly accessible locations, which can be mitigated by restricting shared data as advised by ENISA. Implementing fine-grained permissions and ensuring permissions are protected can help prevent unauthorized access to sensitive information, thereby aligning with the control to prevent unintentional data leaks stated in MASVS-STORAGE-2.

2.18 Implementation Guidance (ENISA 1.18):

ENISA Secure Smartphone Development Guidance (1.18): Restrict broadcast messages (e.g., Android Broadcast Intents) to authorized applications and audit the application's broadcast messages for sensitive content.

2.18.1 OWASP MASVS MAPPING

- MASVS-CODE-4: Both "MASVS-CODE-4" and the ENISA guideline emphasize the importance of handling incoming data securely. "MASVS-CODE-4" discusses the need to treat all data entry points as untrusted and to properly verify and sanitize incoming data to prevent security issues like injection attacks. Meanwhile, the ENISA guideline focuses on restricting and auditing broadcast messages to authorized applications to prevent sensitive information leakage or manipulation by untrusted parties. Both advocate for control measures on how incoming data is processed to maintain the security and integrity of the app.
- MASVS-PLATFORM-1: The correlation exists because "MASVS-PLATFORM-1" addresses the secure interaction through IPC (Inter-Process Communication) mechanisms provided by the platform. This directly relates to the ENISA guideline to restrict broadcast messages to authorized applications and audit them for sensitive content, as broadcast messages are a form of IPC. Both are concerned with ensuring that data and functionality exposed through IPC are secure and only accessed by intended and authorized entities.
- MASVS-PLATFORM-2: While MASVS-PLATFORM-2 and the ENISA guideline mentioned do not address the exact same security measures within mobile applications, there is a correlation in the underlying principle of protecting sensitive data and functionality from unauthorized access. MASVS-PLATFORM-2 pertains to the secure configuration of WebViews to prevent data leakage and exposure of sensitive functionality. Similarly, the ENISA guideline emphasizes the necessity of restricting broadcast messages to prevent unintended or unauthorized access to sensitive contents. Both guidelines aim to shield sensitive data and functions within mobile apps from security risks, highlighting the importance of implementing strict controls and reviewing interactions that could potentially be exploited by malicious entities.
- MASVS-PLATFORM-3: Both "MASVS-PLATFORM-3" and the ENISA guideline focus on preventing the unintentional leakage of sensitive data. While MASVS-PLATFORM-3 refers to protecting sensitive data from being captured in auto-generated screenshots or exposed to onlookers, the ENISA guideline addresses the restriction on broadcasting messages to prevent sensitive data from being sent to unauthorized applications. Both controls are concerned with the secure handling of sensitive information within the app's user interface and the platform's communication mechanisms.
- MASVS-PRIVACY-2: The correlation exists in the emphasis on protecting user data and restricting access to sensitive information. Both MASVS-PRIVACY-2 and the ENISA guideline focus on safeguarding user privacy through technical measures. MASVS-PRIVACY-2 discusses using unlinkability techniques to prevent identification and tracking, while the ENISA guideline advises on restricting broadcast messages to authorized applications to prevent inadvertent disclosure of sensitive information. Both controls aim to minimize the risk of privacy breaches and unauthorized data access or tracking, demonstrating a shared concern for user identity protection and data security.

- MASVS-PRIVACY-3: The ENISA Guideline about restricting broadcast messages to authorized applications and auditing them for sensitive content is related to ”MASVS-PRIVACY-3” because both address the issue of protecting sensitive user data and ensuring transparency in its use. ”MASVS-PRIVACY-3” focuses on users’ rights to understand how their data is used, including unexpected data collection practices, while the ENISA guideline ensures that sensitive data isn’t inadvertently shared through broadcast mechanisms, which aligns with preventing unauthorized data sharing and maintaining user awareness of data practices. Both guidelines seek to uphold privacy standards and prevent leaks of sensitive information, hence there is a correlation between them.
- MASVS-RESILIENCE-3: The ENISA guideline on restricting broadcast messages aligns with MASVS-RESILIENCE-3, as both aim to protect an application from being easily understood or tampered with. Restricting broadcast messages to authorized applications and auditing them for sensitive content are specific tactics that contribute to making it more difficult for attackers to perform static analysis, understand the app’s internals, and modify its behavior. The ENISA guideline thus supports the goal of ”impeding comprehension” of the app’s workings as stated in MASVS-RESILIENCE-3.
- MASVS-RESILIENCE-4: The correlation between ”MASVS-RESILIENCE-4” and the described ENISA Guideline exists in the context of enhancing an app’s resilience to reverse engineering or tampering. MASVS-RESILIENCE-4 is about making dynamic analysis and instrumentation difficult, which would involve monitoring and interfering with the app’s execution. By restricting broadcast messages to authorized applications, as the ENISA Guideline suggests, the app can prevent unauthorized access to sensitive information or operations which could be leveraged during dynamic analysis for reverse engineering or tampering purposes. This ENISA guideline contributes to the overall goal of MASVS-RESILIENCE-4 by limiting the attack surface that could be used for dynamic instrumentation.
- MASVS-STORAGE-2: Both ”MASVS-STORAGE-2” and the ENISA Guideline focus on preventing unintentional exposure of sensitive data. ”MASVS-STORAGE-2” refers to safeguards against accidental storage or exposure in publicly accessible locations, which can happen through various APIs or system capabilities. The ENISA Guideline specifically targets restricting broadcast messages to authorized applications to avoid broadcasting sensitive content to other apps that should not have access to it. Both are concerned with the controlled access to sensitive data and ensuring it is not leaked unintentionally through the app’s functionalities.

2.19 Implementation Guidance (ENISA 1.19):

ENISA Secure Smartphone Development Guidance (1.19): Do not allow third party keyboards to be used for inputs that may contain sensitive data (e.g., credentials, credit card information). Prefer a custom keyboard for such inputs instead.

2.19.1 OWASP MASVS MAPPING

- MASVS-CODE-3: Both ”MASVS-CODE-3” and the ENISA guideline underscore the importance of security assessments for third-party components within an app’s ecosystem. While MASVS-CODE-3 emphasizes checking third-party libraries for known vulnerabilities, as part of broader whitebox assessment, the ENISA guideline specifically warns against the use of third-party keyboards when dealing with sensitive data. Despite focusing on different aspects, the correlation lies in the overall concern for the risks associated with third-party components and the steps to mitigate those risks in the mobile application security context.
- MASVS-CODE-4: The correlation between ”MASVS-CODE-4” and the ENISA guideline regarding third-party keyboards exists in the context of data entry points and the handling of untrusted input. ”MASVS-CODE-4” discusses the various entry points through which data can enter an app, highlighting that this data can be manipulated by untrusted actors and may compromise security checks, leading to vulnerabilities such as injection attacks. It emphasizes the importance of verifying and sanitizing incoming data before usage. The ENISA guideline specifically addresses the risk associated with third-party keyboards when entering sensitive information. It suggests that to prevent potential interception or alteration of sensitive data by untrusted sources, a custom keyboard should be used for sensitive inputs instead of allowing third-party keyboards. Both the ”MASVS-CODE-4” and the ENISA guideline are focused on the concept of treating data from external sources as potentially harmful and taking measures to ensure that the application handles such data securely. Using a custom keyboard for sensitive inputs is a practical example of applying the principle of treating external input as untrusted, which is in line with the recommendation for data verification and sanitation in ”MASVS-CODE-4”.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1”, which deals with general cryptography best practices, and the ENISA guideline about not allowing third-party keyboards for sensitive input is grounded in the underlying purpose of securing the user’s data. Both guidelines aim to mitigate the risk of attackers accessing sensitive user data by recommending security best practices in mobile environments. ”MASVS-CRYPTO-1” implicitly includes the protection of sensitive user input, while the ENISA guideline offers a specific recommendation to use custom keyboards to prevent potential data leakage through third-party keyboards that could intercept or compromise sensitive information like credentials and credit card details. Both are steps towards the same goal of enhancing data protection in mobile applications.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the ENISA Guideline is that both are focused on secure interactions within the app and preventing unauthorized access or leakage of sensitive data. ”MASVS-PLATFORM-1” emphasizes on secure interactions involving IPC mechanisms, which include safeguarding the data exchanged between different components or applications. The ENISA Guideline’s recommendation to not allow third party keyboards for sensitive data inputs aligns with

this concept by reducing the risk of sensitive information being intercepted or misused by untrusted third-party components, thus ensuring secure data handling within the app's ecosystem. This reflects the overarching principle of controlling the security of data and functionality exposure via app interactions and adhering to secure design principles.

- MASVS-PLATFORM-2: Both "MASVS-PLATFORM-2" and the ENISA Guideline are concerned with protecting sensitive data within mobile applications. While MASVS-PLATFORM-2 focuses on securing WebViews and their configurations to prevent data leakage and unintended exposure of functionality, the ENISA Guideline aims to safeguard sensitive input by recommending the use of custom keyboards over third-party keyboards. Both recommendations are measures to control the input and output mechanisms within an app to mitigate the risk of sensitive information being compromised.
- MASVS-PLATFORM-3: The correlation between "MASVS-PLATFORM-3" and the ENISA guideline arises from their mutual focus on protecting sensitive data displayed in the user interface from unintended leakage through platform mechanisms or third-party access. MASVS-PLATFORM-3 addresses concerns about sensitive data being leaked due to platform mechanisms like screenshots or accidental disclosure, such as shoulder surfing or device sharing. Similarly, the ENISA guideline advises against using third-party keyboards for entering sensitive information to prevent interception by such keyboards, which may not adhere to the same security standards as the platform or application. Both guidelines seek to mitigate the risk of sensitive data exposure through the UI components and interactions of mobile applications.
- MASVS-PRIVACY-1: Both "MASVS-PRIVACY-1" and the ENISA guideline focus on minimizing the exposure of sensitive user data. "MASVS-PRIVACY-1" emphasizes data minimization, informed user consent, and careful integration with third-party SDKs, especially around consent and data sharing. The ENISA guideline advises against the use of third-party keyboards for sensitive inputs because these keyboards could be a vector for data leakage. Both standards aim to limit the access of potentially untrusted third parties to sensitive information, with an emphasis on protecting user data and ensuring privacy.
- MASVS-PRIVACY-3: Both MASVS-PRIVACY-3 and the ENISA guideline are concerned with protecting user privacy and ensuring users are aware of how their data is handled. MASVS-PRIVACY-3 emphasizes that users should be informed about data practices that could include background data collection, which could be seen as an unexpected behavior. The ENISA guideline focuses on preventing sensitive data exposure through third-party keyboards, which is a specific instance of ensuring that data collection practices are secure and do not surprise the user, aligning with the principle of MASVS-PRIVACY-3 that users should be aware of how their data is used. Both aim to mitigate the risk of unauthorized data access or leakage.
- MASVS-PRIVACY-4: MASVS-PRIVACY-4 emphasizes user control over their data, ensuring mechanisms are in place for users to manage and modify their data and privacy settings. The ENISA guideline to avoid third-party keyboards for sensitive input seeks to protect user privacy and data from potential third-party access or misuse, aligning with the principle of giving users control over their data by mitigating the risks associated with third-party components. Both reflect a commitment to safeguarding user data.
- MASVS-RESILIENCE-1: Both the MASVS-RESILIENCE-1 guideline and the ENISA guideline concern the security and trust of the underlying platform on which the mobile application runs. MASVS-RESILIENCE-1 discusses the risks associated with running an app on a compromised platform and emphasizes the importance of ensuring that the operating system's security features can be trusted, as these are integral to safeguards like secure storage and sandboxing. Similarly, the ENISA guideline advises against using third-

party keyboards for sensitive data input because third-party components may undermine the security of the platform—potentially leading to data compromise—thus correlating with the importance of platform integrity expressed in MASVS-RESILIENCE-1.

- MASVS-RESILIENCE-4: The ”MASVS-RESILIENCE-4” requirement deals with hardening the app against dynamic analysis and instrumentation, which could enable an attacker to alter or interact with the app during runtime. The ENISA guideline on not allowing third-party keyboards similarly aims to limit the ability of potentially malicious third-party code to intercept or manipulate sensitive data at runtime. Both requirements share the common goal of preventing dynamic manipulation or interference that could lead to security breaches, making them correlated in terms of their purpose in protecting the app against runtime threats.
- MASVS-STORAGE-1: Both the MASVS-STORAGE-1 requirement and the ENISA guideline emphasize the protection of sensitive data on mobile devices. MASVS-STORAGE-1 focuses on ensuring that any sensitive data stored by the app is properly protected, regardless of the storage location. This implies that the app must consider various data sources and storage options, ensuring that sensitive information is not exposed to unauthorized entities. Similarly, the ENISA guideline advises against using third-party keyboards for sensitive data input because these keyboards could potentially capture and misuse the data, including credentials and payment information. Instead, it recommends using a custom keyboard that the app developer controls, further mitigating the risk of sensitive data exposure. Both guidelines are concerned with securing sensitive data within the app’s ecosystem, although they address different aspects of data handling and input security.

2.20 Implementation Guidance (ENISA 1.20):

ENISA Secure Smartphone Development Guidance (1.20): Disable Auto Correction and Autosuggestion for inputs that contain sensitive data.

2.20.1 OWASP MASVS MAPPING

- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3” and the ENISA guideline ”Disable Auto Correction and Autosuggestion for inputs that contain sensitive data” exists in the context of implementing additional authentication measures securely. MASVS-AUTH-3 emphasizes the need for secure implementation of extra authentication forms for sensitive actions within an app, which include measures such as biometric authentication, pins, and other types of multi-factor authentication. A key aspect of secure implementation involves ensuring that sensitive information entry fields are protected from vulnerabilities that could expose the information to unauthorized parties. Disabling auto-correction and autosuggestion for sensitive data inputs is a security best practice that helps to mitigate the risk of accidental exposure of sensitive data through keyboard input prediction features, thereby aligning with the secure implementation principle outlined in MASVS-AUTH-3.
- MASVS-CODE-4: The MASVS-CODE-4 guideline emphasizes treating all user data as untrusted input, which implies that the data should be verified and sanitized before use. Auto-correction and autosuggestion features, which may make assumptions about user input or generate suggestions based on it, potentially bypass the necessary validation and sanitation steps, leading to the unintentional acceptance of modified data. This could result in security vulnerabilities, aligning with the concern in MASVS-CODE-4 about the inadvertent modification of data by untrusted actors and the need for proper data verification and sanitation to prevent injection attacks and other security breaches. Therefore, there is a correlation between MASVS-CODE-4 and the ENISA guideline to disable auto-correction and autosuggestion for sensitive data inputs.
- MASVS-PLATFORM-1: The ENISA guideline ”Disable Auto Correction and Autosuggestion for inputs that contain sensitive data” relates to data exposure prevention through IPC mechanisms. ”MASVS-PLATFORM-1” emphasizes securing interactions involving IPC mechanisms. Auto correction and autosuggestion features can indirectly expose sensitive data through IPC, as they may share input patterns with other apps or the system keyboard, leading to potential leaks. Therefore, both are concerned with secure data handling and interaction security involving IPC, showing a correlation.
- MASVS-PLATFORM-2: Both ”MASVS-PLATFORM-2” and the ENISA guideline concern the secure handling of sensitive data within a mobile application. ”MASVS-PLATFORM-2” emphasizes the need for securely configured WebViews, which includes preventing sensitive data leakage. The ENISA guideline specifically mentions disabling features like auto correction and autosuggestion on inputs dealing with sensitive data to prevent unintended data exposure. Both guidelines aim to protect sensitive data from being compromised within an app’s user interface.
- MASVS-PLATFORM-3: The ”MASVS-PLATFORM-3” requirement is related to preventing unintentional data leaks due to platform mechanisms, which can include scenarios where sensitive information might be captured or displayed inadvertently, thereby leading to potential data exposure. The ENISA guideline to ”Disable Auto Correction and Autosuggestion for inputs that contain sensitive data” falls under the same category of

protecting sensitive information from being leaked through automated platform features. Both are concerned with ensuring that sensitive data is not exposed accidentally by mechanisms that are intended to enhance user experience but can compromise security in cases involving sensitive inputs.

- MASVS-PRIVACY-1: MASVS-PRIVACY-1 is concerned with apps requesting and accessing only the data they need, with a focus on obtaining informed consent from the user and practicing data minimization. This includes being cautious about the sharing and handling of data by third-party SDKs. The ENISA guideline to ”Disable Auto Correction and Autosuggestion for inputs that contain sensitive data” is in line with the goal of data minimization because it reduces the risk of sensitive data being exposed or inadvertently stored, thereby helping to ensure that apps do not access more data than is necessary. Both advocate for better control of user data and prevention of unnecessary data exposure.
- MASVS-PRIVACY-3: Both ”MASVS-PRIVACY-3” and the ENISA guideline regarding the disabling of auto correction and autosuggestion for sensitive data inputs are concerned with protecting user privacy and ensuring user data is handled securely. While MASVS-PRIVACY-3 focuses on transparency and user consent regarding data usage, the ENISA guideline provides a specific recommendation for how to protect sensitive user data during input. Implementing both measures helps in building trust and safeguarding privacy in mobile applications. The correlation exists in the broader context of privacy considerations in app development.
- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the mentioned ENISA Guideline is that both are concerned with the protection of sensitive data within a mobile application context. ”MASVS-STORAGE-1” highlights the importance of properly protecting sensitive data stored by the app, regardless of the storage location. The ENISA Guideline ”Disable Auto Correction and Autosuggestion for inputs that contain sensitive data” is a specific measure that prevents sensitive data from being exposed through keyboard input features that might inadvertently store or suggest sensitive information. Both are addressing the overarching theme of safeguarding sensitive data from unauthorized access or leakage.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA Guideline ”Disable Auto Correction and Autosuggestion for inputs that contain sensitive data” is that both concern preventing the unintentional exposure of sensitive data. MASVS-STORAGE-2 addresses the broader category of unintentional data leaks due to certain APIs and system features like backups or logs. In contrast, the ENISA guideline specifically addresses the feature of auto-correction and autosuggestion in the context of sensitive data entry. Both controls are aimed at providing developers with measures to protect sensitive data and prevent inadvertent storage or exposure through different avenues in a mobile application environment.

2.21 Implementation Guidance (ENISA 1.21):

ENISA Secure Smartphone Development Guidance (1.21): Disable cut, copy and paste functionalities for inputs that may contain sensitive data or restrict the pasteboard to be accessible only from this application.

2.21.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: The correlation between MASVS-AUTH-2, which deals with ensuring correct implementation of local authentication mechanisms like biometrics or PIN codes, and the ENISA guideline about disabling cut, copy, and paste functionalities for sensitive data inputs lies in their common goal of protecting sensitive user authentication data. Both seek to mitigate the risk of unauthorized access or data breach by enforcing secure practices around user authentication data. Although the ENISA guideline specifically addresses the clipboard interactions to prevent sensitive data from being accessible by other applications, it complements MASVS-AUTH-2, which emphasizes the security of local authentication mechanisms that might also include protecting sensitive data such as biometrics or PIN codes from being exposed or misused. Each guideline contributes to a comprehensive security model that protects user authentication within an application.
- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3” and the ENISA guideline about disabling cut, copy, and paste functionality is that both are concerned with the secure handling of sensitive actions and data within a mobile application. ”MASVS-AUTH-3” refers to implementing additional forms of authentication to secure sensitive actions. The ENISA guideline advises on a specific security measure to mitigate the risk of sensitive data being leaked or compromised through the pasteboard (clipboard). Implementing additional authentication methods (as suggested by MASVS-AUTH-3) and restricting the pasteboard (as per the ENISA guideline) are complementary strategies to enhance the security of mobile apps, particularly for features that handle sensitive data or perform sensitive operations.
- MASVS-CODE-4: The MASVS-CODE-4 control is about treating all incoming data as untrusted input, which includes ensuring that it is properly verified and sanitized before use. The ENISA guideline of disabling cut, copy, and paste functionalities—or restricting the pasteboard access—is a measure to control data entry points, thus reducing the risk of untrusted data being introduced into the application from external sources. Both controls aim to prevent unauthorized or malicious data manipulation, and although they approach the problem from different angles, they are correlated in their goal to secure data entry points in mobile applications.
- MASVS-CRYPTO-1: The correlation between MASVS-CRYPTO-1 and the ENISA guideline regarding clipboard functionalities lies in the underlying principle of protecting sensitive user data from unauthorized access. MASVS-CRYPTO-1 emphasizes the importance of cryptography for securing user data, especially when physical access to a device is possible. Disabling or restricting clipboard functionalities for sensitive inputs as per the ENISA guideline is a measure to ensure that sensitive data, which could otherwise be protected by cryptographic means, is not compromised through the clipboard—a common vector for data leakage. Both are preventive measures against data exposure and unauthorized access.
- MASVS-CRYPTO-2: The ENISA guideline of ”Disable cut, copy and paste functionalities for inputs that may contain sensitive data or restrict the pasteboard to be accessible only

from this application" is related to the concept of "MASVS-CRYPTO-2" which focuses on key management, including protection of the keys. Disabling cut, copy, and paste for sensitive data is part of protecting cryptographic keys (assuming they could be inputted or displayed) from being exposed through the clipboard, which is an insecure temporary storage potentially accessible by other applications. It is a practical measure in the broader context of securely managing and storing cryptographic keys and preventing their leakage, thereby supporting good key management practices as outlined in "MASVS-CRYPTO-2".

- MASVS-PLATFORM-1: The MASVS-PLATFORM-1 control's description emphasizes the secure interaction with IPC mechanisms, which include various ways an application can communicate and share data with other applications or the system. The ENISA guideline regarding disabling cut, copy, and paste functionalities or restricting the pasteboard access aligns with this control by mitigating risks associated with insecure data sharing through IPC mechanisms, specifically the pasteboard or clipboard, which is a common IPC facility. Both seek to protect sensitive data from being leaked or accessed by unauthorized entities during inter-process communication.
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA Guideline is based on the shared goal of preventing sensitive data leakage from mobile applications. "MASVS-PLATFORM-2" focuses on the secure configuration of WebViews, which may involve controls to protect against data leakage through such components. The ENISA Guideline advises disabling cut, copy, and paste functionalities for sensitive data inputs and restricting pasteboard access, which is another method to prevent data leakage. Both guidelines aim to enhance the security of sensitive information within mobile applications by imposing restrictions on how data can be handled within the app and potentially shared with other apps or through the clipboard.
- MASVS-PLATFORM-3: Both MASVS-PLATFORM-3 and the ENISA Guideline focus on protecting sensitive data from being exposed through platform or application mechanisms. MASVS-PLATFORM-3 addresses unintended data leaks through mechanisms like auto-generated screenshots or shoulder surfing, while the ENISA Guideline specifically mentions disabling or restricting clipboard functionalities for sensitive inputs, which is another vector that could lead to unintentional data exposure. The underlying principle is the same: sensitive data should be protected from inadvertent disclosure through the user interface or other interactions with the device.
- MASVS-RESILIENCE-3: The correlation between "MASVS-RESILIENCE-3" and the ENISA guideline regarding disabling cut, copy, and paste functionalities is found in the concept of impeding a potential attacker's ability to understand and manipulate sensitive information in the application. MASVS-RESILIENCE-3 focuses on hindering the attacker's comprehension of the app to prevent tampering through obfuscating code and making static analysis difficult. Meanwhile, the ENISA guideline is aimed at protecting sensitive data at the user interface level by disabling cut, copy, and paste functionalities or by sandboxing the pasteboard. Both measures are inherently designed to increase resilience against attacks by adding obstacles that complicate the attacker's efforts to analyze or extract sensitive information from the application.
- MASVS-STORAGE-1: The "MASVS-STORAGE-1" requirement deals with how mobile apps should handle sensitive data, including its storage. This includes protecting data whether it is stored in private or public locations on a device. The ENISA Guideline's recommendation to disable cut, copy and paste functionalities, or to restrict the pasteboard access, complements the security measures described in "MASVS-STORAGE-1." Both pieces of guidance aim to protect sensitive data from unintended exposure or leakage. By restricting clipboard functionalities, an app can prevent sensitive data from being copied

to locations that are outside the control of the app's storage management, thus adhering to the principle of "properly protected" storage as mentioned in MASVS-STORAGE-1.

- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" and the ENISA guideline about disabling cut, copy, and paste functionalities for sensitive data inputs relates through the broader goal of preventing unintentional leakage of sensitive data. "MASVS-STORAGE-2" addresses the issue of unintentional storage or exposure of sensitive data through system capabilities that developers can prevent, while the ENISA guideline specifically aims to mitigate the risk of sensitive data being copied to the clipboard, which is easily accessible and can lead to data leakage. Both controls are concerned with the proper handling of sensitive data to prevent unintended exposure, hence they share a common security objective.

2.22 Implementation Guidance (ENISA 1.22):

ENISA Secure Smartphone Development Guidance (1.22): Disable screen capture for interfaces that contain sensitive data. If the platform does not support this option (e.g., iOS), notify the user about potential security implications of storing a screenshot in unprotected storage.

2.22.1 OWASP MASVS MAPPING

- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA Guideline on disabling screen capture for interfaces containing sensitive data is that both relate to securing sensitive user data within the app. "MASVS-PLATFORM-2" mentions ensuring that WebViews are configured securely to prevent sensitive data leakage. Securing WebViews can include measures like disabling screen capture, which aligns with the ENISA Guideline's recommendation to prevent sensitive data from being stored in unprotected storage through screenshots. Both focus on enhancing the security of the application to protect against potential data breaches or unauthorized access to sensitive information.
- MASVS-PLATFORM-3: The correlation between "MASVS-PLATFORM-3" and the described ENISA guideline is that both pertain to the prevention of sensitive data exposure through the UI. "MASVS-PLATFORM-3" addresses the need to protect against unintentional leakage of sensitive data through platform mechanisms such as auto-generated screenshots, whereas the ENISA guideline specifically recommends disabling screen capture for interfaces that display sensitive information, or notifying users about the security risks if the platform does not support such a feature. Both stipulations aim to safeguard against data leaks via screenshots, whether they be accidental or due to platform limitations.
- MASVS-PRIVACY-3: MASVS-PRIVACY-3 emphasizes that users should be well-informed about the data handling practices of an application, which includes being aware of any unexpected data collection behaviors. While the ENISA Guideline specifically addresses the need to disable screen capture for interfaces that show sensitive data or, if that's not possible, to inform users about the security implications, it is in harmony with the spirit of MASVS-PRIVACY-3. Both aim to protect user data and ensure users are aware of how their data could be potentially compromised or used. In this context, the act of taking a screenshot and the storage of that screenshot can be considered as a form of data 'collection' and 'storage', and hence knowing about it aligns with the right to understand data use as described in MASVS-PRIVACY-3.
- MASVS-RESILIENCE-4: The MASVS-RESILIENCE-4 guideline regarding making it difficult to perform dynamic analysis and preventing dynamic instrumentation correlates with the ENISA Guideline on disabling screen capture. Both guidelines aim at increasing app resilience against attacks that compromise sensitive data during runtime. Disabling screen capture is a specific example of a measure to prevent dynamic analysis by ensuring that sensitive information is not easily captured or misused through screen capture functionality, which attackers could exploit if allowed.
- MASVS-STORAGE-2: The correlation exists because both the MASVS-STORAGE-2 description and the ENISA Guideline are concerned with the prevention of unintended exposure of sensitive data. MASVS-STORAGE-2 addresses the issue of sensitive data being stored or exposed in publicly accessible locations due to the side-effects of using

certain APIs or system capabilities. The ENISA Guideline specifically mentions disabling screen capture on interfaces with sensitive data to prevent sensitive information from being stored in unprotected storage, which could be considered a side-effect of the system capability that allows taking screenshots. Both statements emphasize the importance of developers taking action to mitigate the risk of unintentional data leaks.

2.23 Implementation Guidance (ENISA 1.23):

ENISA Secure Smartphone Development Guidance (1.23): Disable backgrounding or use a blurry screen when the application transitions to the background in platforms that maintain a screenshot of the visible screen in the local storage (e.g., iOS).

2.23.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: The correlation between ”MASVS-AUTH-2” and the ENISA guideline is that both address securing the authentication mechanisms and sensitive information within mobile applications. MASVS-AUTH-2 implies that biometrics, PIN codes, or other local authentication methods must be implemented correctly to ensure the security of an application that relies on local authentication without remote endpoints. The ENISA guideline focuses on mitigating the risk of sensitive information leakage when an app moves into the background and a screenshot is saved to local storage by the operating system (as is the case with iOS). The connection here is the emphasis on proper implementation of security measures specifically related to local aspects of the app: MASVS-AUTH-2 concerns the local authentication, while the ENISA guideline concerns protecting the local display of information when transitioning to the background. Both seek to prevent unauthorized access to sensitive data on the mobile device. They both contribute to a holistic security approach, where local authentication mechanisms and protection of sensitive data on-screen when the app is in the background are crucial factors for securing mobile applications.
- MASVS-PLATFORM-1: ”MASVS-PLATFORM-1” discusses securing interactions involving IPC (Inter-Process Communication) mechanisms, which are a part of how apps expose data and functionality to other apps and the user. Although the ENISA Guideline specifically addresses preventing sensitive information leaks via screenshots when an app transitions to the background, both the MASVS requirement and the ENISA Guideline are concerned with ensuring that sensitive data is not exposed through the normal functioning of the platform’s features. In this case, the IPC mechanisms may not be directly related to the handling of screenshots, but both are platform-provided features that could potentially leak sensitive data if not securely managed. The correlation lies in the broader concept of secure interaction with platform features to protect sensitive data.
- MASVS-PLATFORM-3: The MASVS-PLATFORM-3 control that mentions the need to prevent sensitive data from being unintentionally leaked due to platform mechanisms like auto-generated screenshots aligns with the ENISA Guideline’s recommendation to disable backgrounding or use a blurry screen when the application transitions to the background, to avoid storing a screenshot of visible sensitive data in local storage. Both are concerned with protecting sensitive information from being captured and stored by the platform’s default behavior when apps are sent to the background.
- MASVS-PRIVACY-3: The correlation between ”MASVS-PRIVACY-3” and the ENISA guideline exists because both are concerned with the privacy of user data and ensuring users are aware of how their data is being utilized. The MASVS control emphasizes the importance of transparency regarding data collection, storage, and sharing, including unexpected behaviors like background data collection. The ENISA guideline recommends practices to prevent sensitive information from being stored in screenshots when an app is

backgrounded, which is a form of data protection. Both aim to enhance user privacy and protect against unauthorized use or exposure of personal data.

- MASVS-PRIVACY-4: There is a correlation between ”MASVS-PRIVACY-4” which is about users having control over their data, including management, deletion, and modification rights, as well as privacy consent, and the ENISA guideline recommending the disabling of backgrounding or using a blurry screen. Both are related to ensuring user privacy and control over their data, especially in the context of potential unintentional data leakage. The ENISA guideline addresses the specific risk of sensitive information being captured in screenshots stored locally when an app transitions to the background, which is indirectly related to giving users control over their data by preventing unauthorized data capture and storage.
- MASVS-RESILIENCE-4: MASVS-RESILIENCE-4 relates to making dynamic analysis and manipulation harder for an attacker by resisting runtime observation and code modification. The ENISA guideline of disabling backgrounding or using a blurry screen aligns with this objective because it aims to prevent unintended information disclosure through screenshots stored on local storage when an app transitions to background. This is a technique that could potentially be used for dynamic analysis if sensitive data is exposed in screenshots. By implementing the ENISA guideline, an app improves its resilience to dynamic analysis and instrumentation, which correlates with the aim of MASVS-RESILIENCE-4.
- MASVS-STORAGE-2: Both ”MASVS-STORAGE-2” and the ENISA guideline address the concept of protecting sensitive data from being unintentionally exposed or stored. ”MASVS-STORAGE-2” deals with the broader concept of unintended data storage or exposure through logs, backups, or other publicly accessible locations due to API usage or system capabilities, while the ENISA guideline is more specific, suggesting a protective measure against the automatic screenshot feature in some operating systems. Both controls aim to prevent sensitive information leakage through unintended means. The ENISA guideline is a specific instance of the general principle outlined in ”MASVS-STORAGE-2”, where the developer has the opportunity to intervene and secure the application’s data.

2.24 Implementation Guidance (ENISA 1.24):

ENISA Secure Smartphone Development Guidance (1.24): Introduce input field masking for inputs that contain sensitive data (e.g., passwords).

2.24.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The Mobile Application Security Verification Standard (MASVS) AUTH-1 requirement focusing on the necessity for apps that connect to remote endpoints to enforce user authentication and adhere to best practices for secure protocol use aligns with the ENISA guideline of introducing input field masking for sensitive data such as passwords. Both aim to enhance the security of authentication mechanisms within mobile apps. Input field masking is a security best practice that helps prevent unauthorized observation and access to sensitive data entered into an app, which falls under the umbrella of ensuring a secure use of involved protocols as mentioned in MASVS-AUTH-1.
- MASVS-PLATFORM-3: The MASVS-PLATFORM-3 requirement is correlated with the ENISA guideline on input field masking for sensitive data inputs. The MASVS statement addresses the need to protect sensitive data displayed in the UI from unintended leaks and exposures, such as through auto-generated screenshots or shoulder surfing. Similarly, the ENISA guideline suggests masking input fields for sensitive data, like passwords, which is a specific instance of protecting sensitive information from being displayed and potentially leaked. Both guidelines aim to safeguard sensitive information from unintended disclosure.

2.25 Implementation Guidance (ENISA 1.25):

ENISA Secure Smartphone Development Guidance (1.25): Leverage the hardware-level encryption support for files at the highest supported security level. If possible request application's files to be protected after the device is locked.

2.25.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: The MASVS-AUTH-2 guideline focuses on how authentication mechanisms like biometrics or a local PIN code should be correctly implemented, ensuring secure authentication, while the ENISA guideline advises using hardware-level encryption to protect files at the highest supported security level after the device is locked. Although they approach security from different angles—one from the perspective of access control and the other from data at rest protection—they both emphasize enhancing the security of user data and preventing unauthorized access. The implementation of strong local authentication mechanisms complements hardware-level encryption by ensuring that even if device-level encryption is robust, access to sensitive operations and data is additionally safeguarded by reliable authentication methods. This creates a multi-layered security approach that aligns with both guidelines' objectives to protect user data from unauthorized access.
- MASVS-CRYPTO-1: The correlation between "MASVS-CRYPTO-1" and the ENISA guideline exists because both emphasize the importance of robust cryptography practices to secure user data on mobile devices, particularly against physical access threats. "MASVS-CRYPTO-1" advocates for general cryptography best practices, which would include, as the ENISA guideline specifies, leveraging hardware-level encryption to protect files, especially when the device is locked, thus aligning with the concept of utilizing strong encryption methods to safeguard sensitive data.
- MASVS-PLATFORM-1: The correlation between "MASVS-PLATFORM-1" which is focused on secure interaction via IPC (Inter-process communication) mechanisms and the ENISA guideline about leveraging hardware-level encryption for file protection is based on the underlying principle of protecting data and functionality from unauthorized access. While MASVS-PLATFORM-1 does not explicitly mention hardware-level encryption, it encompasses a broader set of security measures aimed at ensuring that IPC interactions are secure. The ENISA guideline complements this by specifically advising the use of the highest level of file encryption possible, which is an integral part of securing data at rest and protecting IPC mechanism interactions from being compromised, especially when the device is locked. Together, they both aim to mitigate the risk of sensitive data exposure through secure configuration and system-level controls.
- MASVS-PLATFORM-3: While the MASVS-PLATFORM-3 requirement specifically addresses concerns around the unintended leakage of sensitive data through UI elements and platform mechanisms, the ENISA Guideline focuses on leveraging hardware-level encryption for file protection, especially after the device is locked. The correlation here is that both requirements aim to protect sensitive data, albeit through different methods. MASVS-PLATFORM-3 is about safeguarding data displayed in the UI to prevent leaks through screenshots or observation, while the ENISA Guideline emphasizes securing data at rest through encryption to prevent unauthorized access, particularly when the device is

locked. Both contribute to a comprehensive strategy for securing sensitive information in mobile applications.

- MASVS-PRIVACY-3: Although "MASVS-PRIVACY-3" and the ENISA guideline you've mentioned address different aspects of mobile application security and privacy, there is a correlation between them in the broader context of protecting user data. "MASVS-PRIVACY-3" emphasizes the users' right to be informed about how their data is used, advocating for transparency regarding data handling practices. This requirement aligns with providing users with enough information to understand if and how their data might be encrypted, collected, stored, or shared, including the use of hardware-level encryption when the device is locked. The ENISA guideline on leveraging hardware-level encryption for files to the highest security level correlates with "MASVS-PRIVACY-3" by specifying a technical measure that should be implemented to protect user data. If applications follow this guideline, they enhance the security of the data they handle, which supports the principles outlined in "MASVS-PRIVACY-3" regarding clear information on data handling and adherence to platform guidelines, including those related to user data protection. In summary, the adherence to the ENISA guideline can be seen as part of the commitment to "MASVS-PRIVACY-3" by ensuring the technical protection of user data aligns with the broader obligation to inform and protect users' privacy rights.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the ENISA Guideline on leveraging hardware-level encryption support relates to the overarching concern of ensuring the integrity and security of the platform on which the app runs. "MASVS-RESILIENCE-1" emphasizes the importance of ensuring that the platform has not been tampered with, as a compromised OS could undermine security features critical for protecting the app's data, such as secure storage and sandboxing. Similarly, the ENISA Guideline recommends using the highest level of hardware-level encryption for files and protecting them especially when the device is locked. Both guidelines converge on the premise that the security of the platform is fundamental, and they propose measures to ensure that the data managed by applications remains secure even in scenarios where the device's physical security might be compromised. By leveraging hardware encryption and integrity checks of the OS, apps can offer a higher degree of resilience against threats that aim to subvert platform-level security features.
- MASVS-RESILIENCE-3: The correlation between "MASVS-RESILIENCE-3" and the ENISA guideline is that both aim to enhance the security of mobile apps through different means that eventually raise the bar for potential attackers. Masvs-resilience-3 focuses on making it difficult for attackers to understand the internals of an app, thus impeding both static and dynamic analysis, which is a common step in tampering with applications. This can involve techniques such as obfuscation or anti-tampering measures. The ENISA guideline advises leveraging hardware-level encryption for file protection especially when the device is locked, to protect application data at rest and prevent unauthorized access. While MASVS-RESILIENCE-3 is more about preventing attackers from understanding an app's functionality, which could lead to exploitation or reverse engineering, the ENISA guideline ensures that data remains secure even if an attacker gains physical access to the device. Both contribute to the resilience of an application against various types of attacks.
- MASVS-RESILIENCE-4: The description of "MASVS-RESILIENCE-4" emphasizes making it difficult for attackers to perform dynamic analysis or modify code at runtime, which aligns with the ENISA guideline to use hardware-level encryption for files at the highest security level, especially when the device is locked. Both aim to increase application resilience against attacks by enhancing the security of the app's runtime environment and data storage.

- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the ENISA Guideline is that both are concerned with the proper protection of sensitive data stored by applications. MASVS-STORAGE-1 emphasizes the need for apps to handle various sources of sensitive data and to ensure that data is protected regardless of where it is stored, be it in private app storage or public folders. Similarly, the ENISA Guideline outlines the use of hardware-level encryption for files and advises protection of application files, especially after the device is locked. Both guidelines are focused on securing sensitive data in storage by using appropriate protection mechanisms, with MASVS-STORAGE-1 providing a broader directive and the ENISA Guideline specifying the use of hardware encryption features.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA guideline is that they both address the protection of sensitive data. MASVS-STORAGE-2 specifically focuses on preventing unintentional storage or exposure of sensitive data in publicly accessible locations due to developer oversight, while the ENISA guideline advises utilizing hardware-level encryption to protect files, especially when the device is locked. Both controls are aimed at safeguarding data and mitigating risks associated with data leakage or unauthorized access. They mutually reinforce the principle that developers should take proactive measures to secure sensitive information by leveraging available security features and being aware of potential data exposure through application behavior. The correlation lies in the shared goal of enhancing data security in mobile applications.

2.26 Implementation Guidance (ENISA 1.26):

ENISA Secure Smartphone Development Guidance (1.26): If the application while the device is locked needs to write data to a file, use temporary caches instead of weakening the encryption mode. Swap the file content when the device is unlocked and the original file is accessible again.

2.26.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between "MASVS-AUTH-1" and the ENISA guideline is based on the concept of ensuring secure data handling within an application. Specifically, MASVS-AUTH-1 focuses on following best practices for secure authentication and authorization, especially when connecting to remote endpoints. The ENISA guideline addresses a particular best practice regarding securely writing data to a file when the device is locked, which is within the broader scope of maintaining data security as implied by MASVS-AUTH-1. Both are concerned with preserving the security of data and its access control in different states (during transmission for MASVS-AUTH-1 and when the device is locked for the ENISA guideline). The use of temporary caches as suggested by the ENISA guideline is a practical implementation of ensuring security, while MASVS-AUTH-1 outlines the overarching principle of adhering to security best practices.
- MASVS-CODE-4: The correlation exists in the sense that "MASVS-CODE-4" emphasizes on treating all incoming data as untrusted, and ensuring its validation and sanitization. The ENISA guideline, on the other hand, discusses a specific data entry point scenario involving file writes during the device lock state. Both relate to the handling of untrusted input and maintaining security integrity, with the MASVS principle being broader while the ENISA guideline is a specific instance or application of such a principle, particularly addressing the handling of data in a temporarily less secure state (device locked) and ensuring it's security is reinforced when possible (device unlocked).
- MASVS-CRYPTO-1: The "MASVS-CRYPTO-1" description emphasizes the importance of cryptography in securing a user's data on mobile devices, especially considering the high probability of attackers gaining physical access to a user's device. The suggested practice of avoiding weak encryption modes when a device is locked is in harmony with the MASVS guideline that calls for adherence to general cryptography best practices. The ENISA guideline's recommendation to use temporary caches while the device is locked follows the spirit of MASVS-CRYPTO-1 by ensuring data remains secure even in a locked state. When the device is unlocked, the data can then be transferred back to the original file with proper encryption, ensuring that general best practices for cryptography, as addressed by MASVS-CRYPTO-1, are maintained throughout the process.
- MASVS-CRYPTO-2: The correlation between "MASVS-CRYPTO-2" and the described ENISA Guideline can be found in the focus on maintaining strong encryption through proper management of the cryptographic process. MASVS-CRYPTO-2 emphasizes the importance of managing cryptographic keys throughout their lifecycle, which includes protecting keys while data is being written and stored. The ENISA Guideline specifically addresses a scenario where data needs to be written while the device is locked. It suggests using temporary caches in lieu of weakening encryption, which aligns with the principle of protecting keys (and by extension, the data being encrypted) when the main encrypted data is not accessible, such as when the device is locked. When the device is unlocked, the

data can be securely transferred back to the original file without compromising encryption, thus reflecting proper key management practices as per MASVS-CRYPTO-2.

- MASVS-NETWORK-1: The correlation between "MASVS-NETWORK-1" and the ENISA Guideline is that both are concerned with maintaining the confidentiality and integrity of data. "MASVS-NETWORK-1" focuses on protecting data in transit by establishing secure connections and preventing the misuse of secure defaults or bypass mechanisms. The ENISA Guideline complements this by providing a strategy for protecting data at rest (stored on the device), especially in situations where the device is locked. It suggests using temporary caches to securely handle data until the device is unlocked and the original, more secure storage is available. Both guidelines aim to ensure that sensitive data is not exposed or compromised regardless of its state (in transit or at rest).
- MASVS-PLATFORM-1: The control described by "MASVS-PLATFORM-1" relates to secure interactions involving IPC (Inter-Process Communication) mechanisms, which includes ensuring that data exchanges and functionality exposures are handled securely. The ENISA Guideline about using temporary caches to write data while the device is locked, instead of weakening encryption, aligns with the emphasis on secure interactions. It specifically addresses a secure method for data handling during a state (device locked) where certain IPC interactions may still occur or be necessary. Both stress the importance of maintaining security during different states of app interaction, including when the device is locked.
- MASVS-PLATFORM-3: The correlation exists because both statements are concerned with the security of sensitive data on mobile platforms. MASVS-PLATFORM-3 focuses on the protection of sensitive data displayed in the UI to prevent unintentional leaks through screenshots or over-the-shoulder attacks. The ENISA Guideline addresses the secure handling of sensitive data while the device is locked, suggesting the use of temporary caches to avoid weakening encryption. Both reflect the principle of minimizing exposure of sensitive information under particular conditions (UI display and device lock state) and ensuring proper data handling to maintain security.
- MASVS-PRIVACY-2: Although the statement from "MASVS-PRIVACY-2" and the ENISA Guideline address different aspects of privacy and security, there is a correlation in their underlying principle of protecting user information. The MASVS-PRIVACY-2 focuses on the importance of using techniques such as data abstraction, anonymization, and pseudonymization to prevent user identification and tracking. It emphasizes creating technical barriers to ensure that user data collected for one purpose is not repurposed for another, thereby maintaining the privacy and integrity of user data. The ENISA Guideline advises on a specific technical measure to protect data when a device is locked by recommending the use of temporary caches instead of permanent storage with weaker encryption. This guideline is directed at maintaining the data's confidentiality by ensuring it is only accessible when the device is unlocked, which minimizes exposure to potential attackers while the device is in a locked state. Both the MASVS-PRIVACY-2 and the ENISA Guideline share a common goal of implementing precautions to secure user data against unauthorized access and misuse, thus respecting and safeguarding user privacy. The correlation lies in their commitment to employing technical strategies to preserve the confidentiality and integrity of user data, which is central to protecting user identity and preventing user tracking.
- MASVS-PRIVACY-3: The correlation between "MASVS-PRIVACY-3" and the ENISA Guideline about using temporary caches when the device is locked can be seen in their shared focus on protecting user data. MASVS-PRIVACY-3 emphasizes transparency in how user data is used, including background data collection practices, aligning with the

ENISA Guideline's attention to secure data handling methods, even in the locked state, to prevent unnecessary exposure or data leaks. Both aim to maintain user privacy and data security, adhering to expected privacy standards and platform guidelines.

- MASVS-RESILIENCE-1: The MASVS-RESILIENCE-1 guideline and the ENISA guideline both stress the importance of a secure and unaltered platform to ensure the safety of the application's data. While MASVS-RESILIENCE-1 focuses on validating that the operating system has not been compromised to maintain trust in its security features, the ENISA guideline provides a specific example of maintaining data security by using temporary caches to write data while the device is locked, rather than weakening the encryption. Both guidelines implicitly rely on the integrity and security features of the underlying platform to protect the application's data.
- MASVS-RESILIENCE-2: The correlation between "MASVS-RESILIENCE-2" and the described ENISA Guideline is that both relate to maintaining the integrity and security of the app and its data. "MASVS-RESILIENCE-2" emphasizes preventing modifications to the original code and resources to maintain app functionality integrity, which is part of securing the app against unauthorized changes and ensuring the app behaves as intended, even on user-controlled devices. The ENISA Guideline suggests using temporary caches to write data when the device is locked, thereby avoiding the weakening of encryption mode, which aligns with the objective of "MASVS-RESILIENCE-2" in terms of ensuring that data and functionality remain secure and unaltered. By using temporary caches and then swapping the content when the device is unlocked, the guideline supports the resilience control by protecting the app's data integrity when it's in a vulnerable state (locked), similar to protecting the integrity of code and resources.
- MASVS-STORAGE-1: The correlation exists between "MASVS-STORAGE-1" and the mentioned ENISA Guideline. The description for "MASVS-STORAGE-1" focuses on ensuring that sensitive data stored by the app is protected properly, regardless of the location. The ENISA guideline is in alignment with this principle, as it recommends using temporary caches (which are presumably more secure and less permanent) instead of weakening encryption when the device is locked, and then swapping back when the device is unlocked and secured storage is accessible. Both are concerned with maintaining the integrity and security of sensitive data during storage.
- MASVS-STORAGE-2: The "MASVS-STORAGE-2" control and the cited ENISA guideline both address the concern of preventing unintentional leakage of sensitive data. MASVS-STORAGE-2 calls for caution with APIs and system capabilities that might accidentally store sensitive data in publicly accessible locations. Similarly, the ENISA guideline advises using temporary caches to store data when the device is locked, rather than compromising encryption, and swapping content once the device is unlocked and the original file is accessible, to prevent data leakage. Both emphasize the importance of managing sensitive data appropriately to avoid exposing it.

2.27 Implementation Guidance (ENISA 1.27):

ENISA Secure Smartphone Development Guidance (1.27): Prefer using framework functionality (e.g., Android Content Provider) for data sharing instead of using file system permissions or a custom access scheme on platforms that support this (e.g., Android).

2.27.1 OWASP MASVS MAPPING

- MASVS-CODE-4: There is a correlation between ”MASVS-CODE-4” and the described ENISA guideline. MASVS-CODE-4 emphasizes that all incoming data should be treated as untrusted input and that there must be proper verification and sanitation before it's used. This is to prevent security issues such as injection attacks or bypassing security checks. The ENISA guideline advises the use of framework functionalities like Android Content Provider over file system permissions or custom access schemes for data sharing, which is a specific instance of treating data as untrusted. By using framework functionalities, the potential for modification by untrusted actors is reduced since these frameworks are designed to handle data more securely and often include built-in validation and sanitization methods. Hence, both point towards the necessity of careful handling of incoming data to maintain the app's security integrity.
- MASVS-PLATFORM-1: The correlation exists as both the Mobile Application Security Verification Standard (MASVS) PLATFORM-1 and the ENISA guideline emphasize on using secure, platform-provided inter-process communication (IPC) mechanisms for data sharing and functionality exposure. MASVS-PLATFORM-1 requires that all interactions involving IPC mechanisms are conducted securely, which includes using standard components that the platform offers for such purposes. The ENISA guideline specifically advises using framework features like Android Content Provider over file system permissions or custom access schemes for data sharing, which aligns with the secure interaction MASVS-PLATFORM-1 refers to. Thus, both are advocating for leveraging the platform's built-in secure methods for IPC.
- MASVS-PRIVACY-1: The correlation between ”MASVS-PRIVACY-1” and the ENISA Guideline regarding the use of framework functionality for data sharing is that both emphasize minimizing access to user data and enhancing user privacy. MASVS-PRIVACY-1 calls for data minimization and informed consent, restricting access control and sharing with third parties, and managing third-party SDKs in accordance with user consent. Likewise, the ENISA Guideline advises using existing platform functionalities like Android Content Provider for data sharing, which is designed to handle data in a secure and privacy-conscious manner, instead of using less secure methods like file system permissions or custom schemes. Both guidelines aim to protect user data by suggesting the use of safer, more controlled methods of data access and sharing, and aligning with best practices for app development and regulatory requirements.
- MASVS-PRIVACY-3: The MASVS-PRIVACY-3 requirement and the ENISA Guideline both emphasize the importance of transparency and proper management of user data. MASVS-PRIVACY-3 focuses on informing users about data collection and following platform guidelines, while the ENISA Guideline recommends using built-in framework functionalities for data sharing, which typically come with their own set of permissions and disclosures that align with platform guidelines and enhance transparency for the user

about how their data is handled. Both aim to protect user privacy and ensure that users are aware of and can consent to how their data is being used and shared.

- MASVS-PRIVACY-4: The correlation between "MASVS-PRIVACY-4" and the ENISA guideline is rooted in the principle of data privacy and user control over their data. "MASVS-PRIVACY-4" emphasizes users' control over their data, implying that mechanisms should be in place for users to manage their data, which includes sharing it securely. The ENISA guideline promotes utilizing framework functionality (like Android Content Provider) for data sharing, which aligns with having a secure mechanism for managing data sharing as part of user data control, ensuring that data is shared in a controlled and intended manner, enhancing privacy and user agency. Both are united by the underlying objective to safeguard user data and respect user consent.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the ENISA guideline is that both are focused on the importance of relying on the security features of the underlying platform and framework to protect app data and ensure its integrity. "MASVS-RESILIENCE-1" stresses the necessity of operating on a secure platform that has not been compromised to maintain the effectiveness of security controls such as secure storage and sandboxing. Similarly, the ENISA guideline advocates for using framework-provided functionalities like Android's Content Provider for data sharing, rather than custom or file system-based schemes, which relies on the platform's built-in security features. Both recognize the value of the platform's security mechanisms and the risks associated with bypassing them.
- MASVS-STORAGE-1: The correlation between "MASVS-STORAGE-1" and the ENISA guideline is that both are concerned with the secure handling of sensitive data within mobile applications. "MASVS-STORAGE-1" emphasizes that sensitive data should be protected regardless of where it is stored, whether in private or public locations. The ENISA guideline complements this by suggesting the use of framework functionality such as Android Content Providers for data sharing, instead of relying on file system permissions or custom access schemes. Using standard framework functionalities like Content Providers can enhance the security of the data as they are designed to encapsulate the data and provide controlled access to it, thereby aligning with the intent of "MASVS-STORAGE-1" to ensure sensitive data is properly protected.
- MASVS-STORAGE-2: The correlation exists between "MASVS-STORAGE-2" and the ENISA guideline because both are concerned with the proper handling and protection of sensitive data within mobile applications. "MASVS-STORAGE-2" emphasizes that developers must prevent unintentional leaks of sensitive data due to the usage of certain APIs or system features that may expose data to publicly accessible locations. The ENISA guideline advises using framework features, such as the Android Content Provider, for data sharing instead of less secure methods like file system permissions or custom access schemes. This suggests a shared objective of encouraging developers to use proven, secure methods provided by the platform's framework to manage sensitive data and avoid accidental exposure.

2.28 Implementation Guidance (ENISA 1.28):

ENISA Secure Smartphone Development Guidance (1.28): Inspect application-initiated custom notification messages for sensitive content: (A) Allow the user to disable notifications, (B) Allow the user to disable showing content in notifications

2.28.1 OWASP MASVS MAPPING

- MASVS-AUTH-3: While the MASVS-AUTH-3 requirement from the Mobile Application Security Verification Standard (MASVS) and the ENISA Guideline on application-initiated custom notification messages are addressing different aspects of mobile app security, there is a correlation in terms of providing secure user interaction measures. MASVS-AUTH-3 emphasizes the need for implementing additional forms of authentication securely, especially for sensitive actions. It suggests various methods such as biometric, pin, multi-factor authentication (MFA) code generators, and more. The ENISA Guideline advises securing notifications, particularly with respect to sensitive content. It provides measures to protect user privacy and prevent sensitive data exposure through notifications (such as A: disabling notifications, and B: hiding notification content). The correlation lies in the broader goal of protecting sensitive data and actions within the app. MASVS-AUTH-3 deals directly with securing the authentication process for sensitive actions, while the ENISA Guideline deals with the aftermath, ensuring that sensitive information isn't accidentally revealed through notifications after such actions have occurred. Both contribute to a comprehensive security strategy that safeguards user data and actions in different situations.
- MASVS-PLATFORM-1: "MASVS-PLATFORM-1" addresses secure interactions through inter-process communication (IPC) mechanisms, which implies controlling how the app exchanges data and functionality with other apps or the user. The ENISA Guideline's focus on inspecting custom notification messages for sensitive content and allowing users to control notifications aligns with this, as notifications can be a form of IPC. Ensuring that notifications do not expose sensitive information and that users can manage their preferences for notifications contributes to secure IPC practices.
- MASVS-PLATFORM-3: The ENISA guideline to "Inspect application-initiated custom notification messages for sensitive content" and its subparts (A) and (B), which suggest allowing the user to disable notifications and showing content in them, are indeed correlated with "MASVS-PLATFORM-3". Both are focused on ensuring that sensitive data such as passwords, credit card details, or OTP codes are not unintentionally leaked through the user interface or other platform mechanisms. MASVS-PLATFORM-3 addresses the broader issue of protecting sensitive data within the UI from unintentional leaks, including auto-generated screenshots and shoulder surfing, while the ENISA guideline focuses more specifically on controlling the content of notifications to avoid exposure of sensitive information. Both directives serve the common goal of enhancing privacy and security by managing how sensitive information is handled and displayed.
- MASVS-PRIVACY-3: The MASVS-PRIVACY-3 guideline and the ENISA guideline both focus on user privacy and control over their data. MASVS-PRIVACY-3 emphasizes users' rights to be informed about how their data is used, which includes being made aware of any unexpected background data collection. The ENISA guideline complements this by ensuring that notifications, which could potentially leak sensitive information, can be

controlled by the user. They can disable notifications altogether or just the content shown in notifications, further contributing to the user's control over their private data and how it is shared or displayed. Both guidelines ultimately serve to protect user privacy and provide transparency into an app's data practices.

- MASVS-PRIVACY-4: The correlation between MASVS-PRIVACY-4 and the ENISA guideline mentioned is that both are concerned with providing users with control over their data and privacy settings. MASVS-PRIVACY-4 emphasizes the importance of giving users mechanisms to manage, delete, modify their data, and change privacy settings, which aligns with the ENISA guideline that suggests users should have the ability to disable notifications or showing content in them, thereby exercising control over their privacy and the data presented to them through notifications. Both guidelines are aimed at enhancing user privacy and data management capabilities within mobile applications.

2.29 Implementation Guidance (ENISA 1.29):

ENISA Secure Smartphone Development Guidance (1.29): Exclude sensitive application files from device backups and cloud synchronization services. If this option is not available in the in use platform (e.g., Android), exclude the whole application from device backups.

2.29.1 OWASP MASVS MAPPING

- MASVS-CRYPTO-1: The MASVS-CRYPTO-1 guideline emphasizes the importance of cryptography in protecting user data, especially on mobile devices where physical access by attackers is a significant risk. It suggests adherence to best practices in cryptography, which includes protecting sensitive data. The ENISA guideline specifically recommends excluding sensitive application files from device backups and cloud synchronization services to prevent unauthorized access, which aligns with the notion of protecting data at rest using cryptographic measures. Both guidelines aim to safeguard sensitive data on mobile devices, thus showing a correlation.
- MASVS-CRYPTO-2: MASVS-CRYPTO-2 focuses on managing cryptographic keys throughout their lifecycle, including protection and storage. The ENISA Guideline advises excluding sensitive application files from device backups and cloud synchronization services to prevent unauthorized access. Both controls are concerned with securing sensitive data: MASVS-CRYPTO-2 does this by managing cryptographic keys effectively to maintain the integrity of encryption, while the ENISA Guideline does it by preventing sensitive data from being included in potentially insecure backups. Both suggest methods to ensure sensitive data is not exposed due to poor security practices, hence there is a correlation between the two.
- MASVS-PLATFORM-1: The ENISA guideline to ”Exclude sensitive application files from device backups and cloud synchronization services” is correlated with MASVS-PLATFORM-1, which emphasizes securing interactions involving IPC (Inter-Process Communication) mechanisms. Device backups and cloud synchronization services can potentially be used as IPC mechanisms if they transfer sensitive data between processes or systems. Ensuring sensitive application files are excluded from backups aligns with the intention of securing IPC interactions per MASVS-PLATFORM-1. By excluding sensitive data from backups or cloud sync, the app minimizes the risk of data leaks or unauthorized access through these vectors, which is a measure to ensure that interactions involving data transfer are secure.
- MASVS-PRIVACY-1: The MASVS-PRIVACY-1 guideline emphasizes the importance of data minimization and restricted access control, which includes obtaining informed consent from users before accessing their data and sharing with third parties. This focuses on ensuring that user consent is respected and that unnecessary data access or sharing is avoided, especially through third-party SDKs. The ENISA guideline advises excluding sensitive application files from backups and cloud synchronization to protect user data. Both guidelines aim at reducing the risk of unauthorized access to user data by implementing measures that limit data exposure and enhance privacy, representing a correlation in their goals for protecting user privacy and data security.
- MASVS-PRIVACY-3: Both ”MASVS-PRIVACY-3” and the ENISA guideline emphasize the protection of user data. MASVS-PRIVACY-3 is about transparent information to users

regarding data usage, ensuring that unexpected data collection practices are clearly disclosed. The ENISA guideline address the need to protect sensitive application files from backups and cloud synchronization, which is a specific action that supports MASVS-PRIVACY-3's more general principle of safeguarding user data and being clear about data use. Both ultimately contribute to protecting user's privacy rights.

- MASVS-PRIVACY-4: While MASVS-PRIVACY-4 emphasizes user control over their data, including the ability to manage, delete, and modify their data and change privacy settings, the ENISA guideline on excluding sensitive application files from backups complements this by ensuring that the user's data is not inadvertently exposed through device backups and cloud synchronization services. Both guidelines aim to protect user data privacy by providing mechanisms of control and minimizing unintended data sharing. The user's control over their data implicitly includes deciding how their data should be handled in terms of backups and synchronization.
- MASVS-RESILIENCE-1: The MASVS-RESILIENCE-1 control discussed the importance of trusting the platform to ensure the security of an application, mentioning aspects such as secure storage, sandboxing, and other platform-dependent security features. The ENISA Guideline advises excluding sensitive application files from device backups and cloud synchronization services due to potential risks in scenarios where the platform may be compromised or not fully secure. This guideline aligns with the MASVS-RESILIENCE-1 intent, as excluding sensitive data from backups helps maintain the application's data integrity even if the platform's security features—like secure backups—are compromised. Therefore, there is a correlation as both emphasize safeguarding app data in cases where the platform's security cannot be assured.
- MASVS-RESILIENCE-2: Both MASVS-RESILIENCE-2 and the ENISA guideline address the concern of preventing unauthorized access or modifications to the application's data or code. MASVS-RESILIENCE-2 suggests implementing proper protections to maintain the integrity of the app, by preventing modifications to its code and resources. The ENISA guideline complements this by recommending that sensitive application files should be excluded from device backups and cloud synchronization services to prevent their potential exposure or tampering through these mediums. Both recommendations aim to mitigate the risks associated with user-controlled devices where the application resides, although they approach the problem from slightly different angles.
- MASVS-STORAGE-1: "MASVS-STORAGE-1" and the described ENISA Guideline both emphasize the protection of sensitive data managed by mobile applications. MASVS-STORAGE-1 focuses on ensuring that any sensitive data stored by the app is protected, regardless of the location (private or public storage areas), while the ENISA Guideline specifically addresses the need to exclude sensitive application files from device backups and cloud synchronization to prevent unauthorized access or leakage of such data. Both aim to safeguard sensitive data against potential threats that could arise from improper handling or storage practices.
- MASVS-STORAGE-2: Both "MASVS-STORAGE-2" and the ENISA guideline emphasize the importance of preventing sensitive data from being unintentionally stored or exposed, especially in locations that are publicly accessible or could be included in device backups and cloud synchronization services. The MASVS-STORAGE-2 is focused on controlling unintentional leaks of sensitive data as a side-effect of using certain APIs or system capabilities, while the ENISA guideline provides a specific strategy to exclude sensitive application files from device backups and cloud synchronization as a means to prevent such leaks. They both convey the same principle of protecting sensitive data by

managing how it is stored and ensuring that it is not unintentionally exposed or backed up in an insecure manner.

2.30 Implementation Guidance (ENISA 1.30):

ENISA Secure Smartphone Development Guidance (1.30): If the application allows the arbitrary selection of files from the device storage, consider the use of a white-list to restrict access only to the intended (absolute) file paths.

2.30.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The ENISA Guideline relates to ensuring secure protocols and practices are followed when an app interacts with a remote endpoint, especially in terms of file access and user data. ”MASVS-AUTH-1” talks about enforcing best practices for secure usage of protocols in authentication and authorization, which implies that the app should include mechanisms to prevent unauthorized file access. A whitelist, as suggested by the ENISA Guideline, is a best practice to ensure that only intended and authorized file paths are accessible, thus maintaining secure protocol usage and adhering to the requirements laid out in ”MASVS-AUTH-1”.
- MASVS-CODE-4: The correlation exists because the Mobile Application Security Verification Standard (MASVS) CODE-4 emphasizes that applications have multiple data entry points including the file system, and this data should be considered untrusted and properly verified and sanitized. The ENISA Guideline similarly addresses the file system as a potential vector for untrusted input, suggesting the use of a whitelist to restrict file access paths, which is a method of verifying and ensuring that only intended paths are used, thus preventing the exploitation of the file system for injection attacks or bypassing security checks. Both guidelines aim to prevent security breaches by controlling how data from various sources, including the file system, is handled.
- MASVS-PLATFORM-1: The correlation exists because both ”MASVS-PLATFORM-1” and the ENISA Guideline address the security of interactions with the system, though in different contexts. MASVS-PLATFORM-1 focuses on securing IPC (Inter-Process Communication) mechanisms to ensure that data and functionality exposed intentionally by apps are interacted with securely. The ENISA Guideline suggests the use of a whitelist to restrict file path access when an app allows arbitrary file selection, which is a way to secure interactions involving file access. Both are concerned with providing secure methods for apps to interact with other apps, the system, or the user, in order to prevent unauthorized or unintended access to data or functionality.
- MASVS-PRIVACY-1: The correlation between ”MASVS-PRIVACY-1” and the ENISA guideline is that both emphasize restricting access to only what is necessary for the app’s functionality. MASVS-PRIVACY-1 focuses on data minimization and informed user consent, ensuring that apps request access only to data they need and share user data judiciously. Similarly, the ENISA guideline suggests restricting file access via a whitelist, which aligns with the principle of granting permissions exclusively to the app’s required data sets—which is a form of data minimization and access control.
- MASVS-PRIVACY-3: Both ”MASVS-PRIVACY-3” and the ENISA Guideline focus on protecting the user’s privacy and data. MASVS-PRIVACY-3 emphasizes the right of users to be informed about how their data is used, which includes being transparent about data collection practices. The ENISA Guideline suggests restricting file access only to intended paths, which is a method to prevent unauthorized or unexpected data access. Implementing

a whitelist as suggested by ENISA is in line with providing users with clear information on data practices and adhering to data usage expectations as stated in MASVS-PRIVACY-3.

- MASVS-PRIVACY-4: The correlation here is that both MASVS-PRIVACY-4 and the ENISA Guideline emphasize the importance of user control over their data. MASVS-PRIVACY-4 underlines the need for mechanisms that let users manage their data, which includes modifying and deleting it, as well as updating privacy settings. Similarly, the ENISA Guideline advises on a method (using a white-list) to control which files an app can access on a device, which is a way to prevent unauthorized access to user data. Both guidelines aim to protect user data and ensure that the user has authority over how their data is accessed and used by apps.
- MASVS-RESILIENCE-2: The correlation between ”MASVS-RESILIENCE-2” and the ENISA guideline about restricting file path selection ties into the broader theme of ensuring application integrity and protecting against modifications or misuses. ”MASVS-RESILIENCE-2” emphasizes the importance of preserving the original functionality of the app by preventing code and resource modifications, which can be circumvented by local alterations or backdoored versions in third-party app stores. Similarly, the ENISA guideline addresses a specific aspect of resilience by suggesting the use of a whitelist for file path selection to prevent arbitrary access to device storage, which could lead to the execution of unauthorized or modified code. Both measures are proactive steps to maintain the integrity and intended functionality of an application while operating in an untrusted, user-controlled environment.
- MASVS-STORAGE-1: Both the Mobile Application Security Verification Standard (MASVS) reference to ”MASVS-STORAGE-1” and the ENISA Guideline touch upon the security measures that must be observed when mobile applications handle sensitive data storage. ”MASVS-STORAGE-1” describes the need for apps to protect sensitive data, regardless of where it is stored—be it private to the app or in public folders. Meanwhile, the ENISA Guideline suggests restricting file path access to prevent arbitrary file selection from the device storage, which aligns with the MASVS idea of protecting stored sensitive data by potentially employing a whitelist approach. Both are concerned with ensuring that sensitive data on the device is accessed and manipulated in a controlled and secure manner.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the described ENISA Guideline exists as both are concerned with the prevention of unintentional exposure of sensitive data due to the misuse or misconfiguration of storage APIs, system capabilities, or file access permissions. MASVS-STORAGE-2 addresses the risk of sensitive data being unintentionally stored or exposed in publicly accessible locations, which could happen through arbitrary file selection or poor handling of file paths. The ENISA guideline suggests mitigating this risk by restricting file access via a whitelist, ensuring that only intended file paths are accessible, directly addressing the concern mentioned in MASVS-STORAGE-2 about the developer having ways to prevent unintentional leaks. Both focus on ensuring sensitive data stays within controlled and secure storage boundaries and that it is not inadvertently exposed through the application’s functionality.

2.31 Implementation Guidance (ENISA 1.31):

ENISA Secure Smartphone Development Guidance (1.31): Delete application caches on app termination.

2.31.1 OWASP MASVS MAPPING

- MASVS-CRYPTO-1: ”MASVS-CRYPTO-1” relates to implementing cryptography best practices for securing user data, and one aspect of data protection is ensuring sensitive information is not recoverable after the application is closed. Deleting application caches upon termination, as advised by the ENISA guideline, is a practice that complements cryptographic controls by removing temporary data that could potentially be accessed by attackers with physical access to the device. Together, both guidelines aim to protect user data from unauthorized access.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the ENISA guideline ”Delete application caches on app termination” lies in the focus on securing data and app functionalities, especially when app interactions involve IPC (Inter-Process Communication) mechanisms. MASVS-PLATFORM-1 encompasses the broader security approach that includes ensuring IPC is handled securely, and one aspect of securing IPC can be managing application caches. While the ENISA guideline specifically mentions deleting caches upon app termination to remove sensitive data, it is a part of the larger security goal outlined by MASVS-PLATFORM-1 which is to prevent unintended data exposure through any IPC mechanisms. By deleting caches, the app reduces the risk of sensitive data being intercepted or accessed by unauthorized entities during or after IPC.
- MASVS-PLATFORM-2: The correlation between ”MASVS-PLATFORM-2,” which involves securing WebViews to prevent sensitive data leakage and exposure of sensitive functionality, and the ENISA Guideline about deleting application caches on app termination, is that both are measures aimed at protecting sensitive data within a mobile application. The MASVS guidance is focused on the secure configuration of WebViews to prevent data leaks while in use, and the ENISA Guideline is focused on ensuring that data is not left behind in application caches when the app is terminated. Both guidelines contribute to reducing the risk of sensitive data being compromised, thus there is a correlation between the two in terms of their objectives to enhance data security in mobile applications.
- MASVS-PLATFORM-3: The correlation between ”MASVS-PLATFORM-3” and the ENISA Guideline ”Delete application caches on app termination” can be seen in the context of preventing unintentional leakage of sensitive data. MASVS-PLATFORM-3 is concerned with ensuring that sensitive data, when displayed in the user interface, does not become vulnerable to unintentional leaks through platform mechanisms or user actions. Similarly, deleting application caches upon app termination, as recommended by ENISA, is a measure to prevent sensitive data, which might have been temporarily stored or cached during app operation, from persisting on the device after the app is closed, thereby reducing the risk of sensitive data exposure. Both guidelines aim to enhance the privacy and security of user data within mobile applications.
- MASVS-PRIVACY-1: The ENISA guideline ”Delete application caches on app termination” relates to the principle of data minimization referenced in ”MASVS-PRIVACY-1” because it involves active measures to reduce the amount of data that could be potentially exposed in case of a breach or leak. By deleting caches upon termination, the applica-

tion limits the amount of residual data that could be accessed by unauthorized parties, enhancing user privacy and aligning with the requirement for apps to access and handle only necessary data. Both ”MASVS-PRIVACY-1” and the ENISA guideline highlight the importance of protecting users’ data by minimizing its footprint and ensuring that data handling follows strict guidelines of necessity and consent.

- MASVS-PRIVACY-2: Although MASVS-PRIVACY-2 and the ENISA guideline ”Delete application caches on app termination” are not identical, there is a correlation between the two. MASVS-PRIVACY-2 emphasizes the importance of techniques like data abstraction, anonymization, and pseudonymization to prevent user identification and tracking, which aligns with the concept of minimizing data retention and exposure as suggested by the ENISA guideline. By deleting application caches upon app termination, it lowers the risk of sensitive information being recovered from cache and aligns with the broader goal of protecting user identity and privacy. It contributes to the reduction of fingerprint-like signals being stored persistently on the device, thereby supporting the principle of unlinkability and isolation of data streams for their intended purposes.
- MASVS-PRIVACY-3: The correlation between ”MASVS-PRIVACY-3” and the ENISA Guideline ”Delete application caches on app termination” is based on the principle of user privacy protection. MASVS-PRIVACY-3 emphasizes users’ right to be informed about how their data is used. This includes being transparent about data collection, storage, and sharing practices. Deleting application caches on app termination is a practice that aligns with these principles. It ensures that sensitive data is not left behind after the app is closed, which could otherwise be accessed by unauthorized parties or other apps. This practice supports the notion of providing clear information about data handling by actively managing the data lifecycle, including its secure disposal, to avoid unexpected data retention or exposure.
- MASVS-STORAGE-1: The MASVS-STORAGE-1 requirement is about handling sensitive data properly, which includes ensuring that it is properly protected regardless of where it is stored. This correlates with the ENISA Guideline of deleting application caches on termination because one of the aspects of handling sensitive data securely is to make sure it is not left unprotected after the application is closed. Deleting caches can prevent unauthorized access to any sensitive data that might have been temporarily stored in the app’s cache while it was running.
- MASVS-STORAGE-2: The description of ”MASVS-STORAGE-2” indicates a concern with sensitive data being unintentionally stored or exposed in locations that could potentially be publicly accessible, including as a side-effect of the use of certain APIs or system capabilities like backups or logs. The ENISA Guideline ”Delete application caches on app termination” corresponds to this point because application caches can be one of the locations where sensitive data might be unintentionally stored. If an app does not properly delete its caches upon termination, sensitive information stored in these caches could be exposed or leaked, leading to a security risk that ”MASVS-STORAGE-2” aims to address. Thus, adherence to the ENISA guideline would be a way to prevent the kind of unintentional leaks of sensitive data that ”MASVS-STORAGE-2” refers to.

2.32 Implementation Guidance (ENISA 1.32):

ENISA Secure Smartphone Development Guidance (1.32): Database files that contain sensitive data (e.g., iOS WebView caches) must be manually removed from the file system. Deleting records using the database API will not necessary lead to complete data removal from database structure.

2.32.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation can be seen in how both the MASVS-AUTH-1 and the ENISA Guideline address the importance of safeguarding sensitive data through proper management and security best practices. MASVS-AUTH-1 refers to user authentication and authorization best practices for apps communicating with remote endpoints. These best practices are likely to include secure handling of sensitive data such as credentials and tokens, perhaps stored in database files. The ENISA Guideline highlights the importance of properly removing sensitive data from the file system, such as database files that may contain sensitive information like cached user data. Therefore, both are concerned with preventing unauthorized access to sensitive data - either through network communication in the case of MASVS or through residual data in file systems as mentioned by ENISA.
- MASVS-PLATFORM-1: Both the MASVS-PLATFORM-1 requirement from the Mobile Application Security Verification Standard (MASVS) and the ENISA guideline on database file handling are concerned with secure data handling and interaction between the application and the user or other applications. MASVS-PLATFORM-1 focuses on ensuring that Inter-Process Communication (IPC) mechanisms provided by the platform are used securely, which implies that any data exposed through IPC must be handled securely to prevent unauthorized access or leakage. The ENISA guideline addresses the secure deletion of sensitive data, emphasizing that simply using database APIs to delete records may not be enough to ensure the data is entirely removed from the system. Both guidelines are correlated as they highlight the necessity of robust security measures when dealing with data exposure and deletion to protect sensitive information within the mobile application's ecosystem.
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the given ENISA Guideline is that both pertain to the secure handling of sensitive data within the context of mobile application features that provide extended functionality such as WebViews. MASVS-PLATFORM-2 refers to the need for securely configuring WebViews to prevent data leakage, which aligns with the ENISA Guideline advising that sensitive data within database files (such as iOS WebView caches) must be thoroughly removed from the file system to ensure data is not inadvertently retained or recoverable. Both highlight the importance of taking additional measures to ensure the security and privacy of sensitive information within mobile apps.
- MASVS-PRIVACY-1: The correlation between "MASVS-PRIVACY-1" and the ENISA Guideline is that both address the importance of protecting sensitive data from unnecessary exposure. "MASVS-PRIVACY-1" advises apps to minimize data access and operate with user consent, while the ENISA Guideline emphasizes the need to ensure sensitive data is completely removed from the file system, as merely using the database API for deletions may not be sufficient. Both guidelines highlight the need for thorough data handling practices to prevent unauthorized access and enhance data privacy protection.

- MASVS-PRIVACY-2: The MASVS-PRIVACY-2 emphasizes the importance of protecting user identity through techniques such as data abstraction, anonymization, and pseudonymization. It also mentions establishing technical barriers to prevent the repurposing of 'fingerprint-like' signals for different objectives beyond their intended function. The ENISA Guideline about manually removing database files that contain sensitive data aligns with the principle of MASVS-PRIVACY-2. It ensures that any sensitive data, such as cached identities or 'fingerprint-like' information, is thoroughly removed from persistent storage, reducing the risk of user identification and tracking, consistent with the goals outlined in MASVS-PRIVACY-2. This correlation lies in the underlying objective of both guidelines to safeguard user privacy by eliminating potential traces of identity and sensitive information.
- MASVS-PRIVACY-3: The MASVS-PRIVACY-3 guideline emphasizes the importance of users being informed about how their data is used, which includes data collection, storage, and sharing practices. The ENISA guideline on manually removing database files that contain sensitive data aligns with this principle by ensuring that when sensitive data is deleted, it is done effectively and thoroughly. This practice supports transparency and assures users that their data isn't lingering or being used in unexpected ways after supposed deletion, thus maintaining their privacy in accordance with the principles outlined in MASVS-PRIVACY-3.
- MASVS-PRIVACY-4: While MASVS-PRIVACY-4 from the Mobile Application Security Verification Standard (MASVS) focuses on giving users control over their data with the capacity to manage, delete, and modify it, along with the ability to change privacy settings, ENISA's guideline emphasizes the technical side of ensuring sensitive data is properly removed from the file system. While they address different aspects, they both correlate in the broader theme of safeguarding user privacy and ensuring the proper handling and deletion of user data. MASVS deals more with user-facing features and the principle of user consent and control, while the ENISA guideline provides a specific implementation measure that supports that principle by ensuring that when users take action to control their data, the deletion is thorough and complete on a technical level.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the mentioned ENISA Guideline is that both involve the safeguarding of sensitive data through ensuring the integrity and security of the platform environment. "MASVS-RESILIENCE-1" emphasizes the importance of running the app on a secure and uncompromised operating system, as this underpins many critical security features like secure storage and sandboxing, which in turn contribute to protecting sensitive data within the app. The ENISA guideline directly addresses the need to properly remove sensitive data from database files on the file system, which can be a part of the secure data handling practices that are predicated on a trusted platform. Both focus on the prevention of data leaks or breaches due to platform tampering or inadequate data removal practices.
- MASVS-RESILIENCE-3: The correlation exists in the principle of protecting sensitive data from being easily understood or accessed by unauthorized parties. The "MASVS-RESILIENCE-3" talks about impeding the understanding of how an app works to prevent tampering, which includes techniques like code obfuscation or anti-tampering mechanisms. The ENISA guideline about manually removing sensitive data from database files complements this by ensuring that data remnants are not left over after records are deleted, which could potentially provide insights into the app's data structure or operations if not properly handled. Both guidelines seek to increase the resilience of the app against static analysis by making it more difficult for an attacker to analyze or manipulate an app's data or codebase.

- MASVS-STORAGE-1: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-STORAGE-1” and the ENISA guideline about handling sensitive data in database files both emphasize on the proper management and protection of sensitive data stored locally on a device. MASVS-STORAGE-1 specifies that apps need to handle sensitive data securely regardless of the storage location, which implicitly includes appropriate deletion and protection mechanisms. The ENISA guideline complements this by specifically addressing the issue of complete data removal from database files, which is a facet of securing sensitive data at rest. Both point towards the same goal of ensuring sensitive data is not exposed due to inadequate storage and removal practices.
- MASVS-STORAGE-2: The ENISA Guideline’s recommendation to manually remove database files that contain sensitive data from the file system correlates with the description of ”MASVS-STORAGE-2,” which addresses the prevention of unintentional leaks of sensitive data due to side effects of using certain APIs or system capabilities. The guideline directly relates to preventing sensitive data within database files from being accessible, which is in line with the intention of control MASVS-STORAGE-2 to avoid such sensitive data exposure through due diligence.

2.33 Implementation Guidance (ENISA 1.33):

ENISA Secure Smartphone Development Guidance (1.33): Disable application logging and debug messages in production releases. All exceptions should be handled securely.

2.33.1 OWASP MASVS MAPPING

- MASVS-PRIVACY-3: The correlation between ”MASVS-PRIVACY-3” and the ENISA Guideline about disabling application logging and debug messages in production releases is that both are concerned with protecting user privacy and ensuring secure data handling practices. MASVS-PRIVACY-3 emphasizes the users' right to know how their data is used, including unexpected data collection practices, which implies a need for transparency and secure handling of user data. The ENISA Guideline's directive to disable logging and debug messages in production releases aligns with this by preventing potential leaks of sensitive information through logs, thereby supporting the privacy and security aspects addressed in MASVS-PRIVACY-3. Both guidelines indirectly support the principle of least privilege by limiting exposure of user data and system information to only what is necessary and expected.
- MASVS-RESILIENCE-2: The correlation between ”MASVS-RESILIENCE-2” and the ENISA guideline regarding disabling application logging and debug messages in production releases is that both are concerned with securing the application against modifications or exploitations that could compromise its integrity or functionality. ”MASVS-RESILIENCE-2” emphasizes that applications need protections to prevent their original code and resources from being modified, which includes preventing cheating or enabling unauthorized features. Similarly, the ENISA guideline advises disabling logging and debug messages to prevent potential attackers from gaining insights into the application's behavior or points of failure, which could be used to modify or exploit the application. Both controls are aimed at ensuring the resilience and integrity of the application in a user-controlled environment.
- MASVS-RESILIENCE-3: The correlation between ”MASVS-RESILIENCE-3” and the ENISA guideline regarding disabling application logging and debug messages in production releases is focused on the principle of minimizing the information available to an attacker trying to understand or tamper with an app. ”MASVS-RESILIENCE-3” emphasizes the importance of impeding comprehension of an app's internals through obfuscation or other means to prevent static analysis, while the ENISA guideline focuses on ensuring that log and debug information isn't available in production apps as it can provide insights into the app's behavior and potential vulnerabilities. Both controls are about reducing the risk of reverse engineering and subsequent tampering, thus showing a correlation in their intent to enhance the resilience of the application.
- MASVS-RESILIENCE-4: The correlation between ”MASVS-RESILIENCE-4” and the ENISA guideline regarding disabling application logging and debug messages in production releases lies in the objective of preventing an attacker from gaining insights into the application's behavior or structure that could facilitate dynamic analysis or dynamic instrumentation. The MASVS-RESILIENCE-4 control aims to make it difficult for an attacker to perform dynamic analysis or modify the code at runtime, which includes any form of debugging and real-time manipulation. Disabling logging and debug messages as recom-

mended by ENISA directly supports this goal by removing verbose output that could be used to understand internal application states, flow, and possibly sensitive data, which in turn could be used to identify points of failure or vulnerabilities during a dynamic analysis. Handling exceptions securely further ensures that unexpected behaviors don't provide additional information to attackers. Therefore, both the MASVS-RESILIENCE-4 and the ENISA guideline contribute to resilience against runtime analysis and manipulation.

- MASVS-STORAGE-2: The Mobile Application Security Verification Standard (MASVS) control "MASVS-STORAGE-2" and the ENISA guideline both address the prevention of unintentional sensitive data exposure. MASVS-STORAGE-2 focuses on cases where sensitive data might be unintentionally stored or exposed due to the use of certain APIs or system capabilities, which includes logs. Similarly, the ENISA guideline advises disabling application logging and debug messages in production, which is a specific instance where sensitive data could be unintentionally leaked if not properly handled. Both guidelines aim to ensure that developers implement measures to prevent accidental exposure of sensitive information, hence there is a correlation between them.

2.34 Implementation Guidance (ENISA 1.34):

ENISA Secure Smartphone Development Guidance (1.34): In the case that the application includes embedded web browsing capabilities (e.g., WebViews), clear stored cookies on app termination or use in-memory cookie storage.

2.34.1 OWASP MASVS MAPPING

- MASVS-PLATFORM-2: The correlation exists because both ”MASVS-PLATFORM-2” and the ENISA Guideline focus on the security aspects of using WebViews within mobile applications. ”MASVS-PLATFORM-2” emphasizes the importance of securely configuring WebViews to prevent data leakage and exposure of sensitive functionalities, which could include the handling of cookies. The ENISA Guideline specifically addresses the management of cookies within WebViews, suggesting measures such as clearing stored cookies on app termination or using in-memory storage to reduce the risk of sensitive data leakage. Both statements highlight the need for careful management of WebViews to protect sensitive information.
- MASVS-PRIVACY-3: Both MASVS-PRIVACY-3 and the ENISA Guideline emphasize the importance of transparent data handling practices and users' rights to understand how their data is used. MASVS-PRIVACY-3 underlines the need for apps to clearly inform users about data collection, storage, and sharing practices, which is in line with the ENISA guideline's specific requirement to manage cookie storage in a way that respects user privacy, such as clearing stored cookies or opting for in-memory storage which is less persistent. Both controls contribute to the broader goal of protecting user privacy and ensuring users are aware of how their data is handled within applications.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA guideline concerning the handling of cookies in embedded web browsing scenarios lies in the focus on preventing the unintentional storage of sensitive data where the developer can take measures to avoid such leaks. ”MASVS-STORAGE-2” addresses the risk of sensitive data being accidentally stored or exposed in publicly accessible places, which could occur if cookies with sensitive information are not managed properly in embedded web browsers. The ENISA guideline specifically advises clearing stored cookies or using in-memory storage to mitigate this risk, thus aligning with the objective of ”MASVS-STORAGE-2” to prevent unintended leaks of sensitive data through proper management and control by the developer.

Chapter 3
Implement user authentication, authorization and session management correctly

Mobile devices are often shared temporarily, lost or stolen. Mobile applications can be undermined by an insecure authentication or authorization control. Unauthorized individuals may obtain access to sensitive data or sensitive systems by circumventing authentication (logins) or by reusing valid tokens or cookies. Mobile applications must implement secure session management to prevent unauthorized access to the application and it's data.

3.1 Implementation Guidance (ENISA 2.1):

ENISA Secure Smartphone Development Guidance (2.1): Do not rely on client side security controls. Application controls can be easily tampered by an adversary. Both authentication and authorization controls should be implemented on the server side.

3.1.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA Guideline is clear. Both are emphasizing that security controls, particularly for authentication and authorization, should not be solely enforced on the client side as they can be circumvented by attackers. MASVS-AUTH-1 suggests that while the app must follow best practices, the actual enforcement of these mechanisms must be handled on the server side. Similarly, ENISA’s guideline advises against relying on client-side security controls and states that authentication and authorization should be implemented server-side. This aligns with the principle that security measures are more robust when applied in the more controlled server environment, where the risk of tampering is significantly reduced compared to the client side.
- MASVS-AUTH-2: The correlation between ”MASVS-AUTH-2,” which discusses the proper implementation of biometrics or local PIN code authentication, and the ENISA guideline advising not to rely solely on client-side security controls due to potential tampering by adversaries, is that both stress the importance of server-side authentication and authorization measures. While MASVS-AUTH-2 recognizes the use of local authentication mechanisms, it implicitly suggests that they should not be the only line of defense, which aligns with the ENISA guideline’s emphasis on server-side security controls to mitigate the risks associated with relying only on client-side authentication.
- MASVS-AUTH-3: The description of ”MASVS-AUTH-3” mentions the implementation of additional forms of authentication for sensitive actions inside the app, indicating the use of client-side security measures. However, the ENISA guideline advises against solely relying on client-side controls, as they can be tampered with by adversaries, and stresses the importance of server-side controls for both authentication and authorization. The correlation is that while MASVS-AUTH-3 suggests the use of additional authentication methods, it does not contradict the ENISA guideline, which implies that these should be supported by robust server-side implementation to ensure security. MASVS-AUTH-3 does not state that these controls should only be client-side, and it’s implied that secure implementation inherently involves proper server-side controls.
- MASVS-CODE-2: Both the MASVS-CODE-2 statement and the ENISA guideline emphasize the importance of not solely relying on client-side security measures. MASVS-CODE-2 suggests having a mechanism to force updates on the client’s app, which implicitly acknowledges that the client-side environment can be insecure or become outdated, leading to vulnerabilities. The ENISA guideline explicitly states that client-side controls can be tampered with, recommending that critical controls like authentication and authorization be server-side. The correlation is that both are advocating for robust security measures that are not wholly dependent on the client side, which may be more susceptible to attacks.
- MASVS-CODE-3: The MASVS-CODE-3 description alludes to the idea that not all components can be thoroughly assessed, like third-party components, and hence one cannot rely solely on client-side security controls. This correlates with the ENISA Guideline, which

emphasizes that relying on client-side security controls is not sufficient as they can be easily tampered with. Therefore, authorization and authentication should be enforced server-side to enhance security, which is a sentiment mirrored indirectly by MASVS-CODE-3 as it points out the limitations of partial security assessments, which include those relating to client-side controls.

- MASVS-CODE-4: The correlation between "MASVS-CODE-4" and the ENISA Guideline is evident in their shared emphasis on not trusting client-side input and ensuring that security controls are implemented to validate and sanitize incoming data. "MASVS-CODE-4" specifically addresses the various entry points for data in an application and the necessity to treat all incoming data as potentially compromised by attackers, which can lead to security vulnerabilities such as injection attacks. Similarly, the ENISA Guideline advises not to rely on client-side security controls as they can be manipulated by an adversary, stressing the importance of implementing authentication and authorization controls on the server side. Both imply that robust security requires data validation and the enforcement of security policies beyond the client's scope, thus affirming the principle that client-side input should be considered untrusted.
- MASVS-CRYPTO-1: The correlation between "MASVS-CRYPTO-1" and the ENISA Guideline is that both emphasize the importance of robust security controls to protect user data, especially in scenarios where an attacker might gain physical access to a device, like in mobile environments. "MASVS-CRYPTO-1" underscores the need for strong cryptographic practices to safeguard data, while the ENISA Guideline asserts that security controls, including authentication and authorization, should not be solely relied upon on the client side (which in the case of mobile apps, would be the device itself) because they can be manipulated by an attacker. Instead, such controls should be enforced on the server side, where they are less susceptible to tampering. Both statements recognize that security measures need to be designed with the assumption that client-side environments can be compromised.
- MASVS-CRYPTO-2: The correlation exists in the sense that both statements are concerning the proper management and security of sensitive operations. "MASVS-CRYPTO-2" highlights the importance of secure key management for cryptography, which would involve how keys are generated, stored, and protected to ensure that they cannot be compromised. The ENISA Guideline stresses that critical security controls should not be exclusively enforced on the client side because they can be manipulated by an adversary. Both are aligned in the principle that security-relevant operations, like cryptography implementation (which includes key management) and authentication/authorization mechanisms, need to be carefully managed and often require a server-side component to reduce the risk of client-side tampering and to maintain overall system security integrity.
- MASVS-NETWORK-1: Both the MASVS-NETWORK-1 description and the ENISA Guideline emphasize the importance of implementing security controls that cannot be easily bypassed or tampered with by an adversary. The MASVS-NETWORK-1 is focused on ensuring the privacy and integrity of data in transit, which includes setting up secure connections to prevent interception or alteration of data, a responsibility that generally lies within the domain of server-side and network transmission protocols like TLS. The ENISA guideline advises against relying solely on client-side security, suggesting that critical controls like authentication and authorization be enforced server-side to enhance security. Both sources advocate for robust security measures beyond the client-side, which may be more vulnerable to tampering.
- MASVS-NETWORK-2: The correlation between "MASVS-NETWORK-2" and the described ENISA Guideline is that both touch upon the topic of not relying solely on client-

side security mechanisms. While MASVS-NETWORK-2 focuses on enhancing transport security by specifically trusting certain CAs, thus reducing the risk of man-in-the-middle attacks through more stringent trust management, the ENISA Guideline emphasizes that security controls should be enforced on the server side to avoid the risk of tampering by an adversary. Implementing certificate or public key pinning, as recommended by MASVS-NETWORK-2, is a client-side control but it complements server-side authentication and authorization controls, not replacing them. It is a part of a defense-in-depth strategy where multiple layers of security controls, including those on both client and server sides, contribute to the overall robustness of security.

- MASVS-PLATFORM-1: Both the MASVS-PLATFORM-1 description and the ENISA Guideline emphasize the importance of secure interactions and the limitation of relying solely on client-side security controls. MASVS-PLATFORM-1 suggests secure handling of inter-process communication (IPC), implying that data exchange and functionality exposure should be carefully managed to prevent unauthorized access or misuse. The ENISA Guideline reinforces this by stating that security controls, especially authentication and authorization, are more reliable when implemented server-side, as client-side controls can be manipulated. Both guidelines suggest a distrust of client-side security measures and a preference for robust, server-side implementations to ensure the security of interactions, which includes IPC mechanisms.
- MASVS-PLATFORM-2: MASVS-PLATFORM-2 addresses the secure configuration of WebViews to prevent sensitive data leakage and exposure of sensitive functionality, while the ENISA guideline emphasizes not relying solely on client-side security controls (which includes controls within WebViews) and suggests implementing critical controls server-side. Although both are discussing different aspects of security, the correlation lies in the principle that simply trusting the client-side environment (like WebViews) is not adequate for securing sensitive operations and data, therefore they are complementary in advocating for thorough and layered security measures.
- MASVS-RESILIENCE-1: The correlation exists in that both the MASVS-RESILIENCE-1 and the ENISA Guideline emphasize not solely relying on the security features of the client side, recognizing that such controls can be compromised. MASVS-RESILIENCE-1 highlights the dangers of running apps on a tampered platform, which can undermine the trust in the platform's security features like secure storage and sandboxing. Similarly, the ENISA Guideline advises that client-side security controls are not enough on their own as they can be bypassed by an adversary, and it emphasizes the importance of server-side implementations of authentication and authorization controls. Both sources acknowledge the limitations and vulnerabilities of depending on client-side security alone.
- MASVS-RESILIENCE-2: The correlation between "MASVS-RESILIENCE-2" and the ENISA Guideline is evident in their common emphasis on not relying solely on the client side for security controls. Both acknowledge the ease with which client-side controls can be bypassed or tampered with by an adversary, thereby compromising the app's functionality and integrity. The MASVS-RESILIENCE-2 description highlights the importance of protecting the app from modifications, while the ENISA Guideline specifically states that critical controls like authentication and authorization should be implemented server-side to mitigate this risk.
- MASVS-RESILIENCE-3: Both the MASVS-RESILIENCE-3 statement and the ENISA guideline emphasize the limitations of client-side security controls. The MASVS-RESILIENCE-3 mentions making it difficult to understand the app's internals to prevent tampering, which aligns with the ENISA guideline that suggests not to rely on client-side controls as they

can be tampered with. Both suggest that stronger security measures, such as server-side controls, are necessary to improve resilience against adversarial actions.

- MASVS-RESILIENCE-4: The correlation between ”MASVS-RESILIENCE-4” and the described ENISA Guideline is that both emphasize the limitation of relying solely on client-side security measures. MASVS-RESILIENCE-4 is about making dynamic analysis and instrumentation—typically client-side security considerations—more difficult, which suggests that client-side controls can be circumvented or manipulated. This is in line with the ENISA Guideline’s warning against relying on client-side controls, since they can be tampered with by an adversary, hence the guidance to implement critical security controls like authentication and authorization on the server side. Both sources recommend strengthening security beyond the client side to improve resistance to attacks.
- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the ENISA Guideline about not relying on client-side security controls is that both emphasize the importance of proper handling and protection of sensitive data. MASVS-STORAGE-1 speaks to the need for appropriate protection of sensitive data irrespective of where it is stored on the client side, while the ENISA Guideline advises that because client-side security controls can be compromised, it’s essential to have strong server-side controls for authentication and authorization. Both statements underscore that client-side data, alone, is not sufficiently secure and that robust measures must be in place to protect sensitive information, which may include server-side controls as per ENISA’s guidance.
- MASVS-STORAGE-2: Both ”MASVS-STORAGE-2” and the ENISA guideline emphasize the importance of handling sensitive data correctly. ”MASVS-STORAGE-2” addresses the risk of accidentally storing sensitive data in publicly accessible locations due to the misuse of APIs and system capabilities, which can be seen as a subset of client-side security controls. The ENISA guideline specifically warns against solely relying on client-side security controls because they can be easily manipulated by an adversary, which could lead to sensitive data exposure. Therefore, there is a correlation as both are concerned with the prevention of unintentional data leaks and the proper use of security controls, advocating a defense-in-depth approach where sensitive operations, especially those related to authentication and authorization, should be enforced on the server-side.

3.2 Implementation Guidance (ENISA 2.2):

ENISA Secure Smartphone Development Guidance (2.2): Consider using asymmetric cryptography for authentication and authorization purposes. Generate and use the private key directly within a platform supported secure hardware (e.g., Trusted Execution Environment (TEE), Secure Element (SE)).

3.2.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1,” which emphasizes the need for implementing best practices in apps for secure user authentication and authorization, and the ENISA guideline advising the use of asymmetric cryptography for authentication and authorization purposes is clear. The ENISA guideline further recommends generating and using the private key within a secure hardware environment. Both focus on reinforcing the security mechanisms for authentication and authorization in mobile apps, thereby ensuring that the app adheres to robust security protocols and practices. The ENISA guideline details a specific best practice (use of asymmetric cryptography and secure hardware) that would contribute to fulfilling the requirements of ”MASVS-AUTH-1” by ensuring the secure management of authentication and authorization procedures within the app.
- MASVS-AUTH-2: The correlation between ”MASVS-AUTH-2” and the ENISA guideline exists in the context of implementing robust authentication mechanisms. MASVS-AUTH-2 emphasizes the need for correctly implementing biometrics or local PIN codes as authentication methods, which can be seen as a subset of the broader requirement from ENISA to use secure hardware like TEE or SE for cryptographic operations related to authentication and authorization. By utilizing secure hardware, the integrity and confidentiality of the private key used in asymmetric cryptography for authentication are ensured, which aligns with the goal of MASVS-AUTH-2 to have secure local authentication mechanisms. Hence, both are focused on enhancing the security of authentication methods, albeit at different abstraction levels.
- MASVS-AUTH-3: The correlation exists between ”MASVS-AUTH-3” and the ENISA guideline in that they both emphasize the need for additional forms of secure authentication. ”MASVS-AUTH-3” references the general concept of adding extra layers of authentication for sensitve in-app actions, which can include various methods such as biometric, PIN, or MFA code generators. The ENISA guideline elaborates on one specific form of secure authentication, which is the use of asymmetric cryptography within secure hardware like TEE or SE. Both guidelines are aligned in their objective to enhance security for authentication and authorization processes within apps.
- MASVS-CRYPTO-1: The correlation is present because ”MASVS-CRYPTO-1” emphasizes the importance of cryptography in protecting user data, particularly in mobile environments where physical access to devices can increase risk. The ENISA Guideline complements this by advising the use of asymmetric cryptography for authentication and authorization, which is a cryptography best practice meant to enhance security, especially when the private keys are generated and used within secure hardware like TEE or SE, aligning with general cryptography best practices referenced in ”MASVS-CRYPTO-1”.
- MASVS-CRYPTO-2: The correlation exists because MASVS-CRYPTO-2 emphasizes the importance of proper key management throughout the lifecycle of cryptographic keys,

which implicitly includes the aspects of key generation, storage, and protection. The ENISA Guideline's recommendation to use asymmetric cryptography for authentication and authorization and to generate and use the private key within secure hardware directly aligns with the concept of secure key management. Both guidelines aim to ensure that cryptographic keys are handled in a secure and robust manner, which is essential for maintaining the integrity and trustworthiness of cryptographic systems. Secure hardware like TEE or SE provides a fortified environment for key generation and usage, aligning with the lifecycle management outlined in MASVS-CRYPTO-2.

- MASVS-NETWORK-1: Both "MASVS-NETWORK-1" and the ENISA Guideline emphasize the importance of ensuring the privacy and integrity of data in transit through secure cryptographic practices. MASVS-NETWORK-1 highlights the need for secure connections, while the ENISA guideline recommends using asymmetric cryptography within a secure hardware environment to further enhance authentication and authorization security. Both are aligned in their goal to protect data from being compromised during network communication.
- MASVS-NETWORK-2: The correlation between "MASVS-NETWORK-2" and the mentioned ENISA Guideline exists in the focus on enhancing security through specific cryptographic practices. MASVS-NETWORK-2 emphasizes trusting only specific Certificate Authorities (CAs) and employing certificate or public key pinning to prevent man-in-the-middle (MITM) attacks by ensuring the app communicates only with the intended server. Meanwhile, the ENISA Guideline advises using asymmetric cryptography within secure hardware like a Trusted Execution Environment (TEE) or Secure Element (SE) for authentication and authorization to protect the private key and ensure secure operations. Both recommendations aim to strengthen the security posture by implementing cryptographic measures that protect the integrity and confidentiality of the communication channel between the client and the server.
- MASVS-PLATFORM-1: The correlation between "MASVS-PLATFORM-1" and the ENISA guideline on using asymmetric cryptography for authentication and authorization purposes is that both are concerned with ensuring secure interactions within the application environment. "MASVS-PLATFORM-1" emphasizes the importance of secure inter-process communication (IPC) mechanisms provided by the platform, which can include secure handling of authentication and authorization data. The ENISA guideline suggests using asymmetric cryptography and secure hardware such as TEE or SE to strengthen security which directly relates to the secure interaction of IPC mechanisms as per "MASVS-PLATFORM-1" requirements by ensuring that sensitive operations such as key generation and usage are protected in a secure environment. Secure IPC would naturally benefit from such practices, as it would be less vulnerable to unauthorized access or manipulation.
- MASVS-PRIVACY-2: The MASVS-PRIVACY-2 control emphasizes techniques for protecting user identity through unlinkability, which includes practices such as data abstraction, anonymization, and pseudonymization to prevent user identification and tracking. The recommendation in the ENISA Guideline to use asymmetric cryptography for authentication and authorization, with the private key generated and used within secure hardware like TEE or SE, aligns with the principle of protecting user identity. Using secure hardware ensures that the cryptographic operations are isolated and secure from external threats, thereby supporting the goal of user privacy and preventing the misuse of identity-related information. It stands as a technical barrier against the extraction or compromise of the private key, which is consistent with the intention behind MASVS-PRIVACY-2 to employ technical measures that safeguard user privacy.

- MASVS-PRIVACY-3: The correlation between ”MASVS-PRIVACY-3” and the ENISA Guideline about using asymmetric cryptography for authentication and authorization within secure hardware is related to the overall goal of ensuring secure and privacy-respecting data handling practices. While MASVS-PRIVACY-3 focuses on the user's rights to understand how their data is used, with an emphasis on transparency regarding data collection, storage, and sharing practices, the ENISA guideline addresses the technical measures for maintaining the confidentiality and integrity of user data, especially in authentication and authorization processes. By utilizing asymmetric cryptography within secure hardware like TEE or SE, apps can strengthen the security around user data, which aligns with the spirit of MASVS-PRIVACY-3 by safeguarding data against unwanted or unexpected access and usage, thereby contributing to transparent and secure data handling practices as outlined in MASVS-PRIVACY-3. This technical means of protection can be considered part of the broader ethical and regulatory expectation that users' data be treated in a secure and transparent manner, which includes clear disclosure of data practices – indeed, secure handling of authentication/authorization can be a part of what an informed user would expect, contributing to the app's adherence to the platform guidelines on data declarations mentioned in MASVS-PRIVACY-3.
- MASVS-RESILIENCE-1: There is a correlation between ”MASVS-RESILIENCE-1” and the ENISA Guideline mentioned. MASVS-RESILIENCE-1 emphasizes the importance of running apps on secure, untampered platforms because a compromised OS can undermine the effectiveness of security features like secure storage, biometrics, and sandboxing that are critical for protecting app data. The validity of the system's security features is crucial for the app's own resilience. The ENISA Guideline recommends the use of asymmetric cryptography for authentication and authorization, and specifically mentions generating and using private keys within platform-supported secure hardware such as TEE or SE. This essentially aligns with the MASVS control since the use of secure hardware modules like TEE or SE is a way of ensuring that the cryptographic operations, and by extension, the platform, are secure and have not been compromised. By doing so, it indirectly supports the precaution suggested by MASVS-RESILIENCE-1, which is to establish trust in the platform's integrity and security features.
- MASVS-RESILIENCE-2: The correlation between ”MASVS-RESILIENCE-2” and the ENISA guideline regarding the use of asymmetric cryptography within secure hardware lies in the focus on enhancing the security and integrity of an app's functionality on a user-controlled device. ”MASVS-RESILIENCE-2” highlights the importance of protecting an app against modifications to its original code and resources, which could lead to cheating, unauthorized access to premium features, or the introduction of malicious backdoors. The ENISA guideline complements this by recommending the use of secure hardware to generate and use private keys for authentication and authorization, which would contribute to the app's resilience against tampering and reverse engineering. By utilizing secure hardware like TEE or SE, an app can protect cryptographic operations and sensitive data, thereby aligning with the goal of ”MASVS-RESILIENCE-2” to maintain the app's intended functionality and integrity.
- MASVS-STORAGE-1: The correlation exists because both the MASVS-STORAGE-1 description and the ENISA guideline emphasize the protection of sensitive data. MASVS-STORAGE-1 focuses on ensuring that sensitive data handled and stored by the app is properly protected regardless of its location. The ENISA guideline complements this by recommending the use of secure hardware mechanisms like TEE or SE for generating and handling private keys for authentication and authorization, which is a subset of protecting

sensitive data. Both guidelines aim to ensure that sensitive information remains secure and is not exposed to unauthorized access.

3.3 Implementation Guidance (ENISA 2.3):

ENISA Secure Smartphone Development Guidance (2.3): If a password based authentication mechanism is used, ensure that a strong a password policy is being followed. Consider enforcing restrictions about password length and formation, reuse of old user passwords, use of common passwords, password duration, etc. It may also be useful to provide feedback on the strength of the password when it is being entered for the first time. However, do not maintain any representation of the password strength in application storage or the back-end server as it may expose the password in preimage attacks.

3.3.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The description for ”MASVS-AUTH-1” emphasizes the importance of implementing best practices for secure user authentication and authorization when an app connects to a remote endpoint. The ENISA Guideline complements this by providing specific recommendations for strong password policies when using password-based authentication mechanisms. Both focus on enhancing the security of authentication processes within applications, thus showing a correlation.
- MASVS-AUTH-3: Both ”MASVS-AUTH-3” and the ENISA Guideline emphasize on strengthening the authentication mechanisms to ensure the security of sensitive actions within an application. ”MASVS-AUTH-3” suggests that additional forms of authentication such as biometrics, PINs, MFA code generators, etc., should be implemented securely to protect sensitive actions. Meanwhile, the ENISA Guideline focuses on having a strong password policy when password-based authentication is used, including recommendations on password length, complexity, and prohibitions against using common passwords or reusing old ones. Both refer to enhancing security during the authentication process, albeit with different focuses—one on additional authentication forms and the other on password strength and management policies.
- MASVS-STORAGE-1: The correlation exists between ”MASVS-STORAGE-1” and the given ENISA Guideline in the sense that both are concerned with the secure handling and storage of sensitive data. ”MASVS-STORAGE-1” emphasizes the protection of sensitive data stored locally on the device, regardless of the location, which includes password policies as part of the broader category of sensitive data. The ENISA Guideline specifically addresses strong password policies, highlighting the need to protect passwords, which are a type of sensitive data. Both aim to mitigate the risks associated with sensitive data exposure and ensure that such data, including passwords, is stored in a secure and protected manner, with consideration for potential vulnerabilities like preimage attacks.

3.4 Implementation Guidance (ENISA 2.4):

ENISA Secure Smartphone Development Guidance (2.4): Do not reveal registered usernames and remove any fingerprint of their existence from verbose error messages.

3.4.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” which refers to the use of secure authentication and authorization protocols that adhere to best practices and the ENISA guideline to ”Do not reveal registered usernames and remove any fingerprint of their existence from verbose error messages” is that both are concerned with security best practices related to authentication mechanisms. Specifically, these standards guide against the leakage of sensitive information that could be used to compromise user accounts or infer the existence of user accounts, such as revealing whether usernames are registered through error messages. This falls under the umbrella of following best practices for secure use of authentication and authorization protocols, as both help prevent unauthorized access and potential security breaches.
- MASVS-AUTH-2: The correlation between ”MASVS-AUTH-2” and the ENISA Guideline can be established through the theme of secure authentication mechanisms. While MASVS-AUTH-2 emphasizes the correct implementation of local authentication mechanisms such as biometrics or a PIN code, which are critical for ensuring that only legitimate users access the application, the ENISA Guideline focuses on avoiding the leakage of sensitive information about user accounts. Both pieces of guidance are concerned with strengthening the security posture around user authentication - MASVS-AUTH-2 through the proper functioning of authentication measures, and the ENISA Guideline through the proper handling of error messages to prevent username enumeration or other forms of information leakage. Together, they contribute to the broader principle of protecting user identity and authentication credentials within an application.
- MASVS-AUTH-3: While ”MASVS-AUTH-3” focuses on the implementation of additional authentication measures for sensitive actions within an app, and the ENISA Guideline emphasizes the importance of not revealing registered usernames or giving any indication of their existence in error messages, both are concerned with enhancing authentication mechanisms and protecting user identity. ”MASVS-AUTH-3” suggests adding layers to authentication, which could include measures to prevent usernames from being revealed, while the ENISA Guideline explicitly states that such information should not be leaked. Their correlation lies in the overarching objective to secure user authentication processes and safeguard user identity.
- MASVS-CRYPTO-1: The controls and guidelines from MASVS-CRYPTO-1 and ENISA both seek to protect sensitive user information through secure practices. MASVS-CRYPTO-1 emphasizes the use of cryptography to secure user data, particularly in mobile environments where physical device access is a risk. The ENISA guideline regarding not revealing registered usernames and avoiding the disclosure of their existence in error messages aligns with this goal by protecting user credentials and reducing the risk of information leakage. Both are concerned with mitigating risks that could compromise user security.
- MASVS-PLATFORM-3: Both ”MASVS-PLATFORM-3” and the ENISA Guideline are concerned with the protection of sensitive data within the user interface to prevent uninten-

tional leaks or disclosure. ”MASVS-PLATFORM-3” addresses the need to ensure sensitive data like passwords, credit card details, and OTP codes are not leaked through platform mechanisms, while the ENISA Guideline focuses on avoiding the reveal of registered usernames and removing their evidence from verbose error messages. Both guidelines aim to protect sensitive user data from being exposed to unauthorized parties, although they focus on different elements and scenarios within the UI and error messaging contexts.

- MASVS-PRIVACY-2: The correlation between ”MASVS-PRIVACY-2” and the ENISA guideline about not revealing registered usernames and removing any fingerprint of their existence from verbose error messages exists in the emphasis on protecting user identity and privacy. Both highlight the importance of preventing user identification and tracking through different means: ”MASVS-PRIVACY-2” suggests unlinkability techniques, while the ENISA guideline advises against revealing identifiers in error messages. Both are concerned with technical measures to safeguard user identity.
- MASVS-PRIVACY-4: Both ”MASVS-PRIVACY-4” and the ENISA guideline emphasize protecting user privacy and data control, but they approach it from different angles. MASVS-PRIVACY-4 focuses on user agency over their own data, ensuring users can manage their data and consent. The ENISA guideline tackles privacy by advising against disclosing usernames or hints thereof in error messages, which could lead to unauthorized data access or user identification. Both are concerned with maintaining user privacy and limiting data exposure.
- MASVS-RESILIENCE-3: Both ”MASVS-RESILIENCE-3” and the ENISA Guideline focus on reducing the amount of information that an attacker can obtain, which could be used to compromise the system. In the context of MASVS-RESILIENCE-3, the goal is to obfuscate or complicate the understanding of app internals to prevent tampering through static analysis. The ENISA Guideline emphasizes not revealing user information or leaving traces in error messages that could be used to deduce system behavior or user details. Both controls aim to minimize the risk of unauthorized access or modification by obscuring sensitive details, thus they are correlated in their intent to harden the system against information leakage that could benefit an attacker.
- MASVS-STORAGE-2: Both ”MASVS-STORAGE-2” and the ENISA guideline emphasize the importance of not exposing sensitive data unintentionally. ”MASVS-STORAGE-2” is concerned with preventing leaks of sensitive data to publicly accessible locations as a side-effect of using APIs or system capabilities. The ENISA guideline specifically addresses not revealing registered usernames and avoiding fingerprints of their existence in error messages. Both are focused on the secure handling of sensitive information to prevent unintended disclosures.

3.5 Implementation Guidance (ENISA 2.5):

ENISA Secure Smartphone Development Guidance (2.5): Introduce a bruteforce protection mechanism for the authentication controls (e.g., password change/reset). Consider enforcing account lockout for a specific duration, extended questions about the user, notifying the user through another channel and completely automated public Turing tests (captcha) in case of multiple failed attempts.

3.5.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The MASVS-AUTH-1 requirement and the ENISA Guideline are correlated, as both are concerned with ensuring secure authentication practices in mobile applications. MASVS-AUTH-1 discusses the best practices for secure authentication and authorization, which involve protecting the app against threats by implementing secure use of protocols. The ENISA Guideline specifically recommends brute force protection mechanisms, which is a subset of the broader security practices mentioned in MASVS-AUTH-1. Both emphasize the importance of securing authentication controls to prevent unauthorized access.
- MASVS-AUTH-2: MASVS-AUTH-2 and the ENISA guideline both address the robustness of authentication mechanisms in apps. MASVS-AUTH-2 focuses on the proper implementation of local authentication methods such as biometrics and PIN codes. The ENISA guideline suggests protection against brute-force attacks through various means, including account lockouts and additional verification methods. Both are concerned with preventing unauthorized access to an application's protected functions and data. In essence, while MASVS-AUTH-2 speaks to the implementation of the mechanisms themselves, the ENISA guideline speaks to the protective measures that prevent abuse of those mechanisms, such as brute-force attempts. The correlation lies in their shared objective of securing authentication processes.
- MASVS-AUTH-3: The correlation between "MASVS-AUTH-3" and the ENISA guideline is that both emphasize additional security measures for authentication within an application. "MASVS-AUTH-3" suggests various forms of additional authentication methods like biometric, pin, MFA code generator, etc., which are intended to enhance the security for sensitive actions inside the app. The ENISA guideline recommends a brute force protection mechanism, such as account lockouts or additional verification processes after multiple failed attempts, which also serves to strengthen the authentication controls and thus protect sensitive actions in a similar way to the additional authentication methods mentioned in "MASVS-AUTH-3". Both are focused on securing the authentication process beyond just the primary means of credential input.
- MASVS-CODE-2: The correlation is that both the "MASVS-CODE-2" description and the ENISA Guideline focus on implementing security controls to protect users and the application's integrity when threats are identified. "MASVS-CODE-2" mentions a mechanism to force app updates in response to critical vulnerabilities, which can prevent the exploitation of such weaknesses, while the ENISA guideline suggests introducing brute force protection mechanisms for authentication controls, which is also a preventive measure against security threats. Both controls are proactive measures to enhance security.

3.6 Implementation Guidance (ENISA 2.6):

ENISA Secure Smartphone Development Guidance (2.6): Ensure that the session management is handled securely after the initial authentication, using appropriate secure protocols.

3.6.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: ”MASVS-AUTH-1” and the ENISA Guideline both emphasize the importance of secure session management post-initial authentication. MASVS-AUTH-1 focuses on the application ensuring the secure use of protocols involved in user authentication and authorization, while the ENISA Guideline specifically mentions the security of session management after initial authentication using proper secure protocols. Both touch upon securing the communication and access control mechanisms in place after a user has been authenticated, pointing toward a correlation between the two statements.
- MASVS-AUTH-3: The Mobile Application Security Verification Standard (MASVS-AUTH-3) which mentions the need for additional forms of authentication for sensitive actions inside the app correlates with the ENISA Guideline regarding secure session management after initial authentication. MASVS-AUTH-3 is concerned with adding layers of security such as biometric, pin, MFA code generator, etc., to enhance the security of an authenticated session. The ENISA Guideline emphasizes that post-initial authentication, the session should be managed securely. Both guidelines aim at improving the security measures after a user is initially authenticated, ensuring that the session remains secure throughout its duration, especially during sensitive transactions.
- MASVS-NETWORK-1: The MASVS-NETWORK-1 requirement focuses on ensuring data privacy and integrity for data in transit, which aligns with the ENISA guideline that emphasizes secure session management after authentication. Both standards necessitate the use of secure protocols to maintain the confidentiality and integrity of data exchanges within mobile applications, highlighting the importance of using encryption and proper endpoint authentication for network communications. Secure session management is an aspect of maintaining secure connections, which is the core of MASVS-NETWORK-1.
- MASVS-NETWORK-2: The guideline from ENISA ”Ensure that the session management is handled securely after the initial authentication, using appropriate secure protocols” relates to the practice of securing the communication channel to prevent unauthorized access and session hijacking after the user has been authenticated. By employing certificate pinning, as described in ”MASVS-NETWORK-2,” an app can ensure that it is communicating with the intended server by validating its certificate against a known set of public keys or certificates, rather than trusting all default root CAs. This adds a layer of security to session management by preventing man-in-the-middle attacks, where an attacker might intercept and manipulate communications. Thus, the MASVS-NETWORK-2 requirement is consistent with and supportive of the ENISA guideline to manage sessions securely using appropriate secure protocols.
- MASVS-PLATFORM-1: MASVS-PLATFORM-1 concerns the secure use of Inter-Process Communication (IPC) mechanisms provided by the mobile platform to expose data or functionality. The ENISA guideline on secure session management after initial authentication is related because IPC mechanisms are a way that apps interact with users and other apps, potentially affecting session management. Ensuring secure IPC mechanisms aligns with

securing the management of a user's session post-authentication, as both revolve around maintaining the integrity and confidentiality of data exchanges during a session. Secure IPC controls can prevent unauthorized access or interception of session data, aligning with the ENISA guideline for secure session management.

- MASVS-PLATFORM-2: There is a correlation. The description of "MASVS-PLATFORM-2" focuses on the secure configuration of WebViews to prevent sensitive data leakage and exposure of sensitive functionalities, which can be related to session management security. The ENISA guideline "Ensure that the session management is handled securely after the initial authentication, using appropriate secure protocols," is also concerned with preventing sensitive data leakage through secure management of sessions in the application. Both statements highlight the importance of secure practices to safeguard sensitive data and functionalities - MASVS-PLATFORM-2 through the configuration of WebView components, and the ENISA guideline through secure session management protocols. Although MASVS-PLATFORM-2 is specifically about WebViews, and the ENISA Guideline addresses session management more broadly, both contribute to the overarching goal of protecting sensitive data within an app. Securely configured WebViews help ensure session management within embedded browser interfaces is fortified against common vulnerabilities.
- MASVS-PLATFORM-3: The description of "MASVS-PLATFORM-3" refers to the protection of sensitive data displayed in the UI from being leaked or unintentionally disclosed. The ENISA Guideline's focus on secure session management after initial authentication intersects with this description because session tokens and other session-specific sensitive data need to be safeguarded during the entire session, not just at the point of authentication. This includes when such information is displayed in the UI, making the secure handling of this information part of the broader issue of maintaining secure session management as per ENISA recommendations. Thus, the two guidelines correlate in their concern for secure handling of sensitive data during active sessions.

3.7 Implementation Guidance (ENISA 2.7):

ENISA Secure Smartphone Development Guidance (2.7): Require authentication credentials or tokens to be passed with any subsequent request (especially those granting privileged access or modification).

3.7.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA guideline is that both are centered around the principle of secure authentication and authorization in applications that connect to remote endpoints. The MASVS-AUTH-1 talks about apps needing to ensure they follow best practices for secure use of protocols in user authentication and authorization. Similarly, the ENISA guideline insists on the necessity of passing authentication credentials or tokens with each request, especially those that grant privileged access or allow modification, which is a best practice for maintaining secure authenticated sessions and enforcing authorization checks. Both statements underscore the importance of continuous authentication and authorization checks as part of a secure app communication protocol.
- MASVS-AUTH-2: Both ”MASVS-AUTH-2” and the ENISA guideline are concerned with the secure implementation and use of authentication mechanisms to protect access to an application’s features and data. ”MASVS-AUTH-2” focuses on the proper implementation of local authentication mechanisms such as biometrics or a PIN code, which may be crucial for apps that completely rely on local authentication without remote endpoints. This ensures that access to the app and its sensitive functionalities is securely gated behind these authentication methods. The ENISA guideline emphasizes that authentication credentials or tokens should accompany any subsequent request after the initial login, especially for those that offer privileged access or the ability to modify data. This is to ensure that each request is made by an authenticated and authorized user and to prevent unauthorized access. Both standards are promoting the principle that access to sensitive features and data should be appropriately protected using authentication — the former focuses on the correct implementation of local authentication mechanisms, while the latter insists on continuous verification of authentication credentials. The underlying objective is to uphold security in the access control process of the applications.
- MASVS-AUTH-3: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-AUTH-3” and the ENISA Guideline both emphasize the need for additional authentication mechanisms, especially for sensitive actions or privileged access within an app. MASVS-AUTH-3 suggests various methods like biometric, PIN, multi-factor authentication (MFA) code generators, email, and deep links for added security, while the ENISA Guideline specifies that authentication credentials or tokens should accompany any request that grants privileged access or allows modification. Both are aligned in their recommendation to enhance security through proper authentication for critical functions within mobile applications.
- MASVS-CRYPTO-2: There is a correlation between ”MASVS-CRYPTO-2” which discusses the importance of secure management of cryptographic keys throughout their lifecycle and the ENISA Guideline that requires authentication credentials or tokens to be passed with any subsequent request. The correlation is that both guidelines are aimed at securing sensitive operations and data. The MASVS-CRYPTO-2 specifically focuses on

ensuring that cryptographic keys, which are used to encrypt data and authenticate users, are managed securely. Poor key management could lead to the compromise of these keys and by extension the data or operations they protect. Similarly, the ENISA Guideline ensures that every critical request is accompanied by proper authentication credentials or tokens to maintain the security of privileged access or actions, which often rely on cryptographic mechanisms that make use of such keys. Thus, both guidelines emphasize the necessity of proper security measures to protect authentication mechanisms and by extension the integrity and confidentiality of data and actions within a system.

- MASVS-NETWORK-1: The MASVS-NETWORK-1 control focuses on ensuring data privacy and integrity for data in transit, which involves encryption and authentication. This aligns with the ENISA guideline which calls for authentication credentials or tokens to be passed with any request, as such a measure is part of setting up secure connections. Both involve elements of authentication to ensure that the data traffic is associated with an authenticated entity and to protect against unauthorized access or modification.
- MASVS-PLATFORM-1: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-PLATFORM-1” which states that ”Apps typically use platform provided IPC mechanisms to intentionally expose data or functionality. Both installed apps and the user are able to interact with the app in many different ways. This control ensures that all interactions involving IPC mechanisms happen securely,” is related to the ENISA guideline ”Require authentication credentials or tokens to be passed with any subsequent request (especially those granting privileged access or modification).” The correlation exists because secure interactions via IPC mechanisms inherently require proper authentication and authorization controls to ensure that only legitimate and intended actors can interact with the app’s functionalities or access its data. The ENISA guideline addresses the need for authentication credentials or tokens, which is critical in securing IPC mechanisms by verifying the identity of entities before granting access or allowing actions to be performed.
- MASVS-PLATFORM-2: MASVS-PLATFORM-2 and the ENISA guideline both emphasize the importance of securing sensitive data and functionality within mobile applications. MASVS-PLATFORM-2 specifically addresses the secure configuration of WebViews to prevent data leakage and exposure of sensitive functionality, such as through JavaScript bridges. This can include ensuring that WebViews handle data securely and do not inadvertently expose sensitive information. The ENISA guideline complements this by recommending that authentication credentials or tokens are required for requests that could grant privileged access or allow data modifications. Both are focused on the secure management of sensitive data within the app’s functionality to protect against unauthorized access and potential security breaches. Ensuring WebViews are securely configured is one aspect of protecting sensitive data, which is aligned with the broader objective of securing authentication tokens as emphasized by ENISA.

3.8 Implementation Guidance (ENISA 2.8):

ENISA Secure Smartphone Development Guidance (2.8): Use unpredictable session identifiers with high entropy. Note that random number generators generally produce random but predictable output for a given seed (e.g., the same sequence of random numbers is produced for each seed). Therefore, it is important to provide an unpredictable seed for the random number generator. The standard method of using the date and time is not secure. It can be improved, for example using a combination of the date and time, the phone temperature sensor, and the data from the gyroscope sensor (x, y, and z axis). Combining multiple values and using well-tested algorithms that maximise entropy should be chosen.

3.8.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The MASVS-AUTH-1 guideline discusses the importance of secure authentication and authorization practices in apps that connect to a remote endpoint. The ENISA guideline on using unpredictable session identifiers with high entropy complements this by outlining how to securely generate session identifiers that are required during the authentication process. Both guidelines aim to enhance security during the user authentication phase to protect against potential attacks, making them aligned in their intent to ensure the secure use of protocols in mobile apps.
- MASVS-CRYPTO-2: While MASVS-CRYPTO-2 addresses the overarching concept of cryptographic key management across its lifecycle, the ENISA guideline emphasizes the importance of utilizing high entropy in session identifiers, which is a specific aspect of key generation—a critical part of key management. Both underscore the significance of avoiding predictability in the generation of cryptographic elements. High entropy and unpredictable seeds, as suggested by ENISA, contribute to the strength and security of key management practices, aligning it with the principles outlined in MASVS-CRYPTO-2.
- MASVS-PRIVACY-2: Both MASVS-PRIVACY-2 and the ENISA Guideline emphasize the importance of unpredictability and entropy in the context of user privacy. MASVS-PRIVACY-2 focuses on techniques such as anonymization and pseudonymization to prevent user identification and tracking, and mentions the importance of technical barriers to ensure data streams serve their intended function without compromising privacy. The ENISA Guideline outlines the need for unpredictable session identifiers with high entropy, and suggests combining multiple values from various sensors to enhance unpredictability and avoid predictability from number generators. Both highlight the necessity of protecting user identity by enhancing the randomness and entropy of methods used in systems.

3.9 Implementation Guidance (ENISA 2.9):

ENISA Secure Smartphone Development Guidance (2.9): Use context to add security to authentication (e.g., geo location, IP location, etc). Ensure that any collected data is in compliance with the local laws and regulatory requirements.

3.9.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: "MASVS-AUTH-1" discusses ensuring that mobile applications follow best practices for secure protocol use in user authentication and authorization when connecting to remote endpoints. The description explicitly mentions the necessity of the application to uphold security standards in the use of protocols for authentication. This correlates with the ENISA Guideline, which emphasizes enhancing authentication security using context-based measures such as geolocation or IP location. Both stress the importance of adhering to security best practices in the context of authentication, though from different angles—the MASVS focusing on general protocol use, and the ENISA adding context-based security to strengthen authentication. Moreover, both directives include compliance with local laws and regulatory requirements, reinforcing the correlation between the two guidelines.
- MASVS-AUTH-2: The correlation between "MASVS-AUTH-2" and the ENISA guideline exists in the sense that both are concerned with strengthening user authentication mechanisms in mobile applications. While MASVS-AUTH-2 emphasizes the correct implementation of local authentication methods such as biometrics or PIN codes, the ENISA guideline suggests enhancing authentication security by considering additional contextual factors such as geolocation or IP location. Both standards aim to increase the robustness of authentication procedures in mobile apps, albeit using different complementary approaches. Moreover, they both take into consideration the compliance with local laws and regulatory requirements when handling user data, which shows an awareness of legal and privacy aspects in the authentication process.
- MASVS-AUTH-3: The Mobile Application Security Verification Standard (MASVS) requirement MASVS-AUTH-3 and the ENISA Guideline both emphasize the importance of enhanced security measures for user authentication within a mobile application. MASVS-AUTH-3 advocates for additional forms of authentication for sensitive actions, which could include context-aware security measures like geo-location or IP location checks as mentioned in the ENISA Guideline. These context-based measures help to strengthen the security by ensuring that authentication requests are consistent with the user's typical patterns and behaviors. Both sources are aligned in promoting a multi-layered approach to authentication and ensuring compliance with relevant laws and regulations.
- MASVS-NETWORK-1: While "MASVS-NETWORK-1" focuses on securing data in transit primarily through encryption and endpoint authentication to ensure data privacy and integrity, the ENISA Guideline complements it by suggesting the use of additional context for security, such as geo-location or IP location, during authentication processes. Both controls point towards a broader concept of ensuring data security and user authentication in compliance with local laws and regulations. Although the MASVS requirement directly addresses secure connections and the ENISA guideline focuses more on authentication security, they converge on the principle of protecting user data and respecting legal frameworks within networked environments.

- MASVS-PRIVACY-1: Both MASVS-PRIVACY-1 and the ENISA Guideline emphasize the importance of data minimization and compliance with local laws and regulatory requirements. MASVS-PRIVACY-1 discusses requesting only necessary data with informed consent and managing third-party SDKs in alignment with user consent, while the ENISA Guideline focuses on collecting data in a manner that complies with local laws during authentication. Both guidelines prioritize responsible data handling and legal compliance, indicating a correlation.
- MASVS-PRIVACY-2: Both MASVS-PRIVACY-2 and the ENISA Guideline emphasize protecting user privacy through careful consideration of user-related information collection. MASVS-PRIVACY-2 promotes unlinkability techniques to prevent user identification and tracking and cautions against repurposing uniquely identifiable data, such as fingerprints for different use-cases. Similarly, the ENISA Guideline suggests using context (like geo-location, IP address) to enhance security but emphasizes that collected data should comply with local laws and regulatory requirements, which inherently includes privacy considerations. Both aim to safeguard user privacy while utilizing user data for specific legitimate purposes and ensuring compliance with privacy regulations.
- MASVS-PRIVACY-3: The correlation between "MASVS-PRIVACY-3" and the ENISA Guideline is that both focus on the proper handling and transparency of user data. "MASVS-PRIVACY-3" emphasizes the user's right to be informed about data usage practices, including unexpected data collection, which aligns with the ENISA guideline's requirement for compliance with local laws and regulations when collecting data such as geo-location for adding security to authentication. Both propose the protection and lawful processing of user information, highlighting transparency and legal compliance.
- MASVS-RESILIENCE-1: The MASVS-RESILIENCE-1 control, which focuses on ensuring that an app runs on a secure, untampered platform, shares a correlation with the ENISA guideline about using the context to enhance authentication security. Both are about leveraging the underlying platform's integrity and capabilities to increase security. The validation of the OS's integrity in MASVS-RESILIENCE-1 ensures that security features like secure storage and biometrics (which can include context-based parameters like geo-location and IP location mentioned by ENISA) can be trusted. Additionally, both touch on the aspect of compliance with local laws and regulatory requirements, emphasizing the need to securely manage sensitive data in agreement with legal expectations.
- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" and the ENISA guideline is that both emphasize the importance of secure handling of sensitive data. MASVS-STORAGE-2 deals with the unintended exposure of sensitive data due to the misuse of APIs or system capabilities, which implies that developers should be aware of and use context-aware data handling strategies to prevent leaks. The ENISA guideline suggests using context such as geo-location or IP location to enhance authentication security, also mentioning the necessity of compliance with data protection laws. Both are concerned with the protection of sensitive information, although in different aspects of security—MASVS-STORAGE-2 focuses on data storage and leakage prevention, while ENISA addresses secure authentication and legal compliance. Nonetheless, they are connected through the overarching principle of safeguarding sensitive data.

3.10 Implementation Guidance (ENISA 2.10):

ENISA Secure Smartphone Development Guidance (2.10): Consider using additional authentication factors for applications giving access to sensitive data or interfaces where possible: (A) Knowledge factors, (B) Something you know (i.e. user's secret question), (C) Possession factors, (D) Something you have (i.e. hardware token, grid, sim card), (E) Inherence factors, (F) Something you are (i.e. fingerprint, voice, facial, retina recognition)

3.10.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation exists because both the MASVS-AUTH-1 guideline and the ENISA Guideline emphasize the importance of secure user authentication in applications. MASVS-AUTH-1 points out the necessity for apps to follow best practices in implementing secure authentication and authorization when connecting to remote endpoints. The ENISA Guideline suggests enhancing security by using additional authentication factors, which is consistent with the best practices mentioned in MASVS-AUTH-1. The different factors listed by ENISA (knowledge, possession, and inherence) are methods to strengthen the authentication process that applications should consider as part of the best practices in user security.
- MASVS-AUTH-2: The "MASVS-AUTH-2" requirement addresses the proper implementation of local authentication mechanisms such as biometrics or local PIN code, which directly correlates with the ENISA Guideline's mention of inherence factors, specifically "Something you are (i.e. fingerprint, voice, facial, retina recognition)." Both discuss the use of biometric authentication as part of a secure authentication system for apps handling sensitive data or interfaces.
- MASVS-AUTH-3: The correlation exists in that both "MASVS-AUTH-3" and the ENISA guideline emphasize the importance of using additional methods of authentication, particularly for actions within an app that involve sensitive information or functionalities. "MASVS-AUTH-3" mentions using a variety of methods such as biometric, PIN, MFA code generators, email, and deep links, which should be securely implemented. Similarly, the ENISA guideline encourages the use of additional authentication factors including knowledge factors (something you know), possession factors (something you have), and inherence factors (something you are), which cover a similar range of authentication methods as those discussed in "MASVS-AUTH-3," and are also focused on security and sensitivity.
- MASVS-CRYPTO-1: The MASVS-CRYPTO-1 guideline emphasizes the importance of cryptography to protect user data, especially on mobile devices where physical access by attackers is possible. Cryptography is a part of a multi-factor authentication approach that protects data by ensuring that only authorized users can access it. The ENISA Guideline supports this by suggesting multiple factors of authentication, which could include cryptographic mechanisms along with knowledge, possession, and inherence factors to enhance security. Both guidelines underscore the necessity of securing sensitive data against unauthorized access, and cryptography can be a crucial component of the authentication factors mentioned by ENISA.
- MASVS-RESILIENCE-1: Both MASVS-RESILIENCE-1 and the ENISA Guideline emphasize the importance of platform and authentication security. MASVS-RESILIENCE-1 focuses on the integrity of the operating system, ensuring security features such as secure

storage and sandboxing are not compromised, which is fundamental for protecting the data within an app. The ENISA Guideline suggests using additional authentication factors, which are dependent on the security and trustworthiness of the platform to function effectively. In other words, strong authentication mechanisms are part of the broader security posture that includes platform integrity as outlined by MASVS-RESILIENCE-1. If the platform is compromised, the additional authentication factors may also be rendered ineffective, indicating a correlation between the two guidelines.

- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the ENISA guideline is that both are concerned with the protection of sensitive data. MASVS-STORAGE-1 emphasizes the proper protection of sensitive data stored locally by the app, regardless of whether the storage location is private or public. The ENISA guideline suggests using additional authentication factors to safeguard access to sensitive data, which complements the storage security by adding a layer of protection to ensure that only authenticated users can access the data. Together, they address both the storage and access aspects of sensitive data protection.

3.11 Implementation Guidance (ENISA 2.11):

ENISA Secure Smartphone Development Guidance (2.11): Do only rely on not adequately secure channels for multi factor authentication (phone numbers and voice mails can be hijacked, see Section 4-10)

3.11.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA guideline is that both emphasize the importance of secure communication channels for authentication. ”MASVS-AUTH-1” stresses the need for apps to follow best practices for secure protocol usage, which aligns with the ENISA warning against relying on insufficiently secure channels (like phone numbers and voicemails) for multi-factor authentication, as they can be compromised. Using secure channels is part of the best practices mentioned in MASVS-AUTH-1.
- MASVS-AUTH-2: Both ”MASVS-AUTH-2” and the ENISA guideline underline the importance of implementing secure authentication mechanisms. MASVS-AUTH-2 mentions the need for correct implementation of biometric or local PIN code authentication, and implies the potential risks associated with relying solely on local app authentication without a remote endpoint. Similarly, the ENISA guideline warns against relying on inadequately secure channels for multi-factor authentication. Both point towards the dangers of using authentication channels that can be compromised, hence there is a correlation between the two in emphasizing robust authentication practices.
- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3” and the ENISA guideline exists in the context of implementing additional forms of authentication securely. MASVS-AUTH-3 suggests the use of various methods such as biometrics, PIN, MFA code generators, email, and deep links for sensitive actions within an app, which directly ties into the ENISA guideline advising against reliance on inadequately secure channels for multi-factor authentication. ENISA specifically mentions the vulnerabilities of phone numbers and voice mails that can be hijacked, implying that secure implementation of authentication means is crucial—which is the point MASVS-AUTH-3 is making about ensuring the secure implementation of these authentication forms.
- MASVS-NETWORK-1: The MASVS-NETWORK-1 guideline focuses on the necessity for data privacy and integrity for data in transit by enforcing secure connections. It emphasizes the risk of developers disabling secure defaults or bypassing them, which could lead to insecure data transmissions. The ENISA guideline advises against relying on insecure channels for multi-factor authentication because they can be compromised, which falls directly under the concerns raised by MASVS-NETWORK-1 about ensuring secure communication channels to protect data from being hijacked or tampered with in transit. Both guidelines stress the importance of secure communication to protect sensitive data.
- MASVS-RESILIENCE-1: The ”MASVS-RESILIENCE-1” mentions the importance of running apps on platforms that have not been tampered with, emphasizing the reliance on the platform’s security features like secure storage, biometrics, sandboxing, etc. This aligns with the ENISA guideline that advises against relying on inadequately secure channels for multi-factor authentication. Both points stress the significance of a secure and trustworthy platform for implementing security controls to protect the app’s data and to assure the

integrity of authentication processes. Relying on compromised channels or platforms could lead to security breaches, which is a concern reflected in both statements.

- MASVS-STORAGE-2: Both ”MASVS-STORAGE-2” and the mentioned ENISA Guideline relate to the concept of securing sensitive data and ensuring that it is not unintentionally exposed or transmitted via insecure channels. MASVS-STORAGE-2 concerns the prevention of sensitive data leaks through careful use of APIs and system capabilities, while the ENISA Guideline advises against relying on insecure channels for critical processes like multi-factor authentication, which could lead to the hijacking of phone numbers or voicemails. Both emphasize the protection of sensitive information from being compromised.

3.12 Implementation Guidance (ENISA 2.12):

ENISA Secure Smartphone Development Guidance (2.12): Use authentication that ties back to the end user identity (rather than only to the device identity).

3.12.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The requirement ”MASVS-AUTH-1” is about ensuring that mobile apps follow best practices for secure user authentication and authorization when connecting to a remote endpoint. This is correlated with the ENISA guideline, which states that authentication should tie back to the end user identity rather than just the device identity. Both guidelines emphasize the importance of linking the authentication process to the actual user to ensure that the access control mechanisms are robust and provide accountability and proper access restrictions based on user identity. This ensures that even if a device is compromised, the authentication tied to a user identity can offer an additional layer of security.
- MASVS-AUTH-2: The statement ”MASVS-AUTH-2” describing the need for apps to implement biometric or PIN code authentication correctly correlates with the ENISA Guideline that specifies using authentication tied to the user’s identity rather than solely to the device. Both statements emphasize the importance of establishing authentication mechanisms strongly connected to the individual user’s identity to enhance security. By correctly implementing biometrics or local PIN codes, the application is ensuring that the authentication mechanism is tied to the physical user (via biometric data) or a secret that the user knows (PIN code), meeting the ENISA guideline’s requirement for user-centric authentication rather than device-centric.
- MASVS-AUTH-3: The MASVS-AUTH-3 guideline and the ENISA guideline both emphasize the importance of tying authentication methods to the user’s identity rather than just the device identity. MASVS-AUTH-3 suggests using additional forms of authentication for sensitive actions (like biometrics, pin, MFA code generator, email, deep links, etc.), which are user-specific rather than device-specific. The ENISA guideline explicitly calls for authentication that ties back to the user’s identity, reinforcing the concept that security should be focused on verifying the user rather than just the device they are using. Both guidelines aim to ensure that sensitive actions are performed by the legitimate user and not by anyone who merely has access to the user’s device.
- MASVS-CODE-3: While MASVS-CODE-3 primarily addresses the need for a comprehensive security assessment of app components, including the identification of known vulnerabilities in libraries, this can indirectly relate to the ENISA Guideline for user-centric authentication. If third-party components or libraries are used for authentication mechanisms and have known vulnerabilities, they could potentially undermine the security of authentication processes that should tie back to the user’s identity. Therefore, MASVS-CODE-3 is relevant as it emphasizes the importance of assessing components that might be responsible for implementing the requirements stated in the ENISA Guideline, even though it might not directly prescribe user-centric authentication methods.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the ENISA guideline regarding user identity-based authentication is present due to the emphasis on security in scenarios where attackers may have physical access to a device. Both guidelines focus on protecting user data and ensuring that authentication mechanisms are robust

against unauthorized access. ”MASVS-CRYPTO-1” discusses general cryptography best practices, which would inherently include secure implementation of authentication mechanisms. The ENISA guideline specifically advocates for authentication tied to user identity rather than just the device, implying the use of cryptographic measures that link authentication credentials to individual users to prevent unauthorized access even if the device itself is compromised. Thus, adoption of MASVS-CRYPTO-1's principles would support the ENISA guideline's recommendation for user-centric authentication, showcasing their correlation.

- MASVS-CRYPTO-2: The correlation between ”MASVS-CRYPTO-2” and the ENISA guideline about using authentication that ties back to the end-user identity is that both are emphasizing the importance of secure management practices around sensitive elements that can impact security. While MASVS-CRYPTO-2 specifically addresses the lifecycle management of cryptographic keys, including how they are generated, stored, and protected, the ENISA guideline focuses on ensuring that authentication mechanisms are linked to the user's identity, not just the device. Proper key management is a crucial component of secure authentication systems because if keys are compromised, the authentication can be subverted, potentially allowing an attacker to impersonate a user. Thus, both controls are related to safeguarding the identity of end-users and maintaining the integrity of the authentication process.
- MASVS-NETWORK-1: The correlation between ”MASVS-NETWORK-1” and the ENISA guideline on authentication is that both emphasize the importance of maintaining the integrity and privacy of data that is transmitted over a network. ”MASVS-NETWORK-1” focuses on the need for secure connections to protect data in transit, suggesting that encryption and endpoint authentication are critical. This implies that the identity interacting with the app, which would be the end user, should have their interactions secured and authenticated, aligning with the ENISA guideline that recommends authentication be tied to the end user's identity. Both controls are aimed at preventing interception or tampering with the data by unauthorized entities, and user-based authentication is a way of ensuring that only authorized users can access or transmit sensitive information.
- MASVS-PLATFORM-1: The MASVS-PLATFORM-1 description mentions that apps use inter-process communication (IPC) mechanisms to expose data or functionality, and it emphasizes the security of these interactions. The ENISA guideline advises the use of authentication tied to the end user identity rather than to the device identity to ensure secure usage and access control. There is a correlation because both the MASVS-PLATFORM-1 and the ENISA guideline seek to ensure secure interactions with the app, preventing unauthorized access and ensuring that only authenticated and authorized entities can interact with exposed IPC mechanisms or functionalities. Secure IPC is an essential part of authenticating end user identity as it often involves transmitting sensitive information that should be accessible only to authenticated users.
- MASVS-PLATFORM-2: The correlation between ”MASVS-PLATFORM-2” and the ENISA guideline can be established in terms of security and protection against sensitive data leakage. The MASVS-PLATFORM-2 mentions the secure configuration of WebViews to prevent sensitive data leakage and exposure, which aligns with the ENISA guideline's emphasis on using authentication tied to the end user identity. Both aim to safeguard sensitive information and ensure it is not compromised, with MASVS focusing on secure UI elements like WebViews, and ENISA enforcing robust authentication methods that relate to user rather than device identity, which could also be compromised through insecure WebView implementations.

- MASVS-PRIVACY-3: While the ENISA guideline focuses specifically on authentication methods tied to the user's identity rather than the device, MASVS-PRIVACY-3 is about overall data privacy and user awareness of data use. Thus, ensuring that users are informed about how their data, including information related to their identity and its use in authentication, is in line with both provisions. Both guidelines emphasize the importance of transparency and user control over their personal data, which includes understanding how their identity is managed and authenticated by the application.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the ENISA guideline about using authentication ties to the end-user identity is that both stress the importance of operating in a secure environment. "MASVS-RESILIENCE-1" emphasizes the risk of operating on a tampered platform, which can undermine platform-dependent security controls like secure storage, biometrics, and sandboxing. Meanwhile, the ENISA guideline suggests that authentication should be linked to the user's identity instead of just the device identity. Both points underscore a foundational security principle: ensuring that the operating environment (whether it be the platform's integrity or the authentication context) is secure and reliable. This is because compromising the platform could potentially circumvent user-based authentication mechanisms as well, putting the data at risk.
- MASVS-RESILIENCE-2: The correlation between "MASVS-RESILIENCE-2" and the cited ENISA guideline lies in their mutual focus on preserving the integrity of an application's environment and functionality. MASVS-RESILIENCE-2 emphasizes the importance of protecting an app from modifications to its original code and resources, ensuring its intended functionality remains intact. This is closely related to the ENISA principle of employing user-centric authentication, as both mechanisms intend to safeguard against unauthorized access or alterations that could undermine the security and intended use of the application. By tying authentication to user identity, it is possible to better ensure that only legitimate users can access and use the app, while preventing device-centric methods that could be exploited by attackers if they gain control of a device. Thus, both are part of a broader strategy to preserve the security and intended use of an application in a potentially hostile user-controlled environment.

3.13 Implementation Guidance (ENISA 2.13):

ENISA Secure Smartphone Development Guidance (2.13): Authentication should not be used as a replacement of authorization security controls. Authorization verifies the permissions of a user and presupposes strong authentication.

3.13.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: Both ”MASVS-AUTH-1” and the ENISA guideline emphasize the distinct roles of authentication and authorization in the security of applications. ”MASVS-AUTH-1” acknowledges the need for secure user authentication and also the enforcement of authorization, indicating that authentication should be accompanied by appropriate authorization mechanisms. The ENISA guideline also asserts that authentication should not replace authorization, underscoring the necessity for authorization to verify user permissions after successful authentication. Both stress that while authentication is crucial, it is not sufficient on its own to ensure security without proper authorization controls.
- MASVS-AUTH-2: The correlation between ”MASVS-AUTH-2” and the ENISA guideline pertains to the context of user authentication mechanisms. MASVS-AUTH-2 discusses the proper implementation of local authentication mechanisms such as biometrics or PIN codes. The ENISA guideline advises that authentication should not be used as a substitute for authorization, implying that even with strong authentication (as prescribed in MASVS-AUTH-2), appropriate authorization controls are still necessary. Both statements highlight the need for robust security but address different, though related, aspects of it. MASVS-AUTH-2 focuses on the authenticity of the user, whereas the ENISA guideline emphasizes that proper permissions (authorization) are also essential after authentication is established.
- MASVS-NETWORK-1: The ”MASVS-NETWORK-1” description and the ENISA Guideline share a correlation regarding the importance of proper security measures in the context of authentication and data protection. ”MASVS-NETWORK-1” emphasizes the need for securing data in transit, specifically through encryption and endpoint authentication, to maintain data privacy and integrity. Meanwhile, the ENISA Guideline highlights the distinct roles of authentication and authorization, implying that strong authentication (which is part of securing data in transit and ensuring privacy and integrity as per MASVS-NETWORK-1) is crucial but not sufficient on its own without proper authorization controls. Both stress the need for comprehensively securing network communication and access rights to safeguard user data and system integrity.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the ENISA Guideline is that both emphasize the need for secure interaction mechanisms within an app’s environment. MASVS-PLATFORM-1 talks about ensuring secure interactions involving Inter-Process Communication (IPC) mechanisms, which implies that authorization checks should be in place to make sure that only allowed entities can interact with the exposed data or functionality. The ENISA Guideline stresses that authentication (verifying the identity of a user) should not be used in place of authorization (verifying the permissions of a user). Both are aligned in the sense that a secure app should employ proper authorization measures to control access to its IPC mechanisms, in addition to strong authentication practices.
- MASVS-PRIVACY-2: The correlation between ”MASVS-PRIVACY-2” and the ENISA guideline is that both emphasize the importance of maintaining the privacy and security of

the user's identity and data. "MASVS-PRIVACY-2" focuses on techniques to prevent user identification and tracking, including ensuring that data used for one purpose (like fraud detection) is not repurposed for another (like analytics), thereby maintaining data stream isolation. The ENISA guideline also delineates between authentication and authorization, implying that strong authentication should be upheld without compromising authorization controls. The underlying principle in both the MASVS-PRIVACY-2 control and the ENISA guideline is that user identity and permissions should be managed with care to prevent misuse and maintain privacy. While MASVS-PRIVACY-2 provides methods for protecting identity, the ENISA guideline outlines the distinction and relationship between authentication and authorization processes. Both contribute to the broader context of user privacy and data protection.

- MASVS-RESILIENCE-1: The description of "MASVS-RESILIENCE-1" mentions the importance of running applications on a secure platform, which hasn't been tampered with, to ensure that security features such as secure storage, biometrics, sandboxing, etc., can be trusted. The ENISA guideline you mentioned emphasizes that strong authentication should be in place before authorization is performed. The correlation is that both emphasize the need for a secure and trustworthy environment before sensitive operations (like authentication and subsequently authorization) can be correctly and safely carried out. The secure platform is a precondition for secure authentication, which is, in turn, a precondition for correct authorization, as implied by the ENISA guideline.

3.14 Implementation Guidance (ENISA 2.14):

ENISA Secure Smartphone Development Guidance (2.14): Apps that support user authentication must have a logout function which terminates the authenticated session. Upon logout, session should also be invalidated on the server side.

3.14.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the described ENISA Guideline is evident in their mutual focus on secure user authentication and session management. MASVS-AUTH-1 implies that best practices for secure protocol use must be followed in apps that connect to remote endpoints and handle authentication, which would encompass both client-side and server-side considerations, such as session termination and invalidation upon user logout as specified by the ENISA Guideline.
- MASVS-AUTH-3: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-AUTH-3” and the ENISA Guideline both address the security aspects of user authentication within mobile applications. MASVS-AUTH-3 calls for additional forms of authentication for sensitive actions inside the app, which implies a multi-layered security approach for user sessions that could include multifactor authentication (MFA), biometrics, and other measures. The ENISA Guideline emphasizes the necessity of a secure logout function to terminate the authenticated session both on the client and server sides, which is complementary to MASVS-AUTH-3 as it addresses the secure management of active sessions. Both guidelines are convergent in their purpose to enhance and protect authenticated sessions against unauthorized access or session hijacking, hence there is a correlation in ensuring a secure authentication lifecycle.

3.15 Implementation Guidance (ENISA 2.15):

ENISA Secure Smartphone Development Guidance (2.15): Clear any maintained sensitive data on session termination. Reset the application state and request for user re-authentication.

3.15.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation exists because ”MASVS-AUTH-1” focuses on the importance of secure user authentication and the enforcement of authorization mechanisms, which are integral to secure communication with remote endpoints. The ENISA guideline of clearing maintained sensitive data on session termination and requiring re-authentication aligns with these best practices by ensuring that user data is protected, especially at the end of a session, thereby preventing unauthorized access and potential misuse. Both highlight the need for robust security measures in the authentication and authorization processes of an application.
- MASVS-AUTH-2: MASVS-AUTH-2 discusses the correct implementation of local authentication mechanisms such as biometrics or a PIN code, emphasizing the importance of proper security practices where local authentication is significant, particularly when there is no remote endpoint. The ENISA guideline advising to clear any sensitive data on session termination and to reset the application state for user re-authentication correlates with the secure implementation principle in MASVS-AUTH-2. Ensuring sensitive data is cleared and the application state is reset after a session ends is part of a secure implementation strategy for authentication mechanisms, which should protect data both during a session and when transitioning out of one.
- MASVS-AUTH-3: Both ”MASVS-AUTH-3” and the ENISA guideline relate to enhancing the security of user authentication within applications. ”MASVS-AUTH-3” suggests using additional forms of authentication for sensitive actions, advocating for more stringent security measures. Similarly, the ENISA guideline recommends clearing sensitive data upon session termination and requiring re-authentication, which aligns with the goal of MASVS-AUTH-3 to provide secure authentication mechanisms. Both standards aim to protect sensitive data and authenticate users securely, suggesting a correlation between their objectives.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the ENISA guideline on clearing sensitive data upon session termination is based on the underlying principle of protecting sensitive user data. MASVS-CRYPTO-1 emphasizes the use of cryptography to secure user data, particularly in situations where physical access to the device is possible. This includes implementing encryption and other cryptographic controls to safeguard data both at rest and in transit. The ENISA guideline complements this by stating that any sensitive data should be cleared when the session ends, and the application should be reset, with a new request for user authentication. This is a measure to prevent unauthorized access to sensitive data after a user has finished their session. If cryptographic measures are in place and a session ends, any encrypted data would be less useful to an attacker without the necessary keys to decrypt it. Both statements are concerned with ensuring that sensitive data remains secure, particularly when the session has ended or when the device could be compromised. Implementing proper cryptographic practices

along with clearing sensitive data on session termination are part of a layered security approach to protect sensitive information on mobile devices.

- MASVS-NETWORK-1: There is a correlation between "MASVS-NETWORK-1" and the described ENISA Guideline. Reasoning: While "MASVS-NETWORK-1" focuses on ensuring data privacy and integrity for data in transit by establishing secure connections, the ENISA Guideline emphasizes the importance of clearing sensitive data when a session terminates and resetting the application state, which includes re-authenticating the user. Both controls relate to the overarching theme of maintaining the security and integrity of users' sensitive data. "MASVS-NETWORK-1" is about protecting data as it moves between the app and other network endpoints, while the ENISA Guideline is about ensuring that any sensitive data does not persist beyond the lifetime of a session, which could include data in transit if a session is terminated. The correlation exists in the shared goal of preventing unauthorized access to sensitive data whether it is being transported or stored temporarily during a session. Additionally, part of securing a session includes ensuring that the initial connection setup is secure, which relates back to the MASVS-NETWORK-1 requirement for secure connections.
- MASVS-PLATFORM-1: The correlation exists because both MASVS-PLATFORM-1 and the ENISA guideline mentioned refer to securing the interaction with the application and ensuring that sensitive data is handled properly. MASVS-PLATFORM-1 focuses on secure interactions involving Inter-Process Communication (IPC) mechanisms, which could be used by other apps or users to interact with the app's data or functionality. The ENISA guideline addresses the importance of clearing sensitive data upon session termination and resetting the application state, which is a part of ensuring that IPC interactions do not expose sensitive information when a session ends. Both points emphasize the need for secure management of sensitive data within an application's lifecycle, including IPC and session management.
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA Guideline regarding clearing sensitive data on session termination is that both are concerned with preventing sensitive data leakage. "MASVS-PLATFORM-2" emphasizes the need for secure configuration of WebViews to prevent sensitive data from being exposed, which aligns with the ENISA guideline's requirement to clear sensitive data when a session ends to avoid unauthorized access or data persistence that could lead to a leak. Both guidelines aim to safeguard sensitive information within the context of an app's session management and user interface components.
- MASVS-PLATFORM-3: The correlation exists because both "MASVS-PLATFORM-3" and the ENISA guideline regarding sensitive data address the concern of protecting sensitive information from unintended leakage or exposure. "MASVS-PLATFORM-3" mentions protection mechanisms to prevent sensitive data displayed in the UI from being leaked through platform mechanisms or accidental disclosures. Similarly, the ENISA guideline advises clearing maintained sensitive data upon session termination, which is also a measure to prevent sensitive data leakage by ensuring that such information is not left accessible after the user's session ends. Both sets of guidelines aim to enhance the privacy and security of sensitive data within the application's lifecycle.
- MASVS-PRIVACY-2: The correlation exists in the emphasis on protecting user privacy and data. While "MASVS-PRIVACY-2" focuses on the need for unlinkability and preventing user identification and tracking by using techniques like data abstraction and pseudonymization, the ENISA guideline stresses the importance of clearing sensitive data upon session termination to protect user privacy and prevent unauthorized access. Both controls are concerned with safeguarding user data and ensuring that personal information

is not misused or easily accessed by unauthorized parties. The end goal of both controls is to enhance user privacy and data protection within the application lifecycle.

- MASVS-PRIVACY-3: Both ”MASVS-PRIVACY-3” and the cited ENISA Guideline emphasize transparency and the protection of user data. ”MASVS-PRIVACY-3” highlights the need for clear communication with users about data usage practices, which aligns with the ENISA Guideline’s emphasis on clearing sensitive data when a session ends, thereby ensuring that user data is not stored beyond the necessary session scope. Both aim to secure user data and reinforce user trust through responsible data management and clear policies.
- MASVS-PRIVACY-4: Both ”MASVS-PRIVACY-4” and the ENISA Guideline concerning the clearance of sensitive data on session termination share a common ground in terms of user data control and protection. ”MASVS-PRIVACY-4” emphasizes giving users the capability to manage their data, implying that data should not persist without the user’s consent and should be manageable upon session termination. The ENISA Guideline specifically addresses the need to clear sensitive data when a session ends, which aligns with the principle of user control over data by ensuring that users’ sensitive information does not remain at risk after they’ve finished using the app. Both guidelines aim to enhance privacy and security by putting the control of sensitive data in users’ hands.
- MASVS-STORAGE-1: The correlation exists because ”MASVS-STORAGE-1” is concerned with the handling and storage of sensitive data by an app, ensuring that such data is protected regardless of where it is stored. This correlates with the ENISA guideline, which states that upon session termination, any maintained sensitive data should be cleared. Both are focused on protecting sensitive data—MASVS-STORAGE-1 by secure handling and storage during the app’s operation, and the ENISA guideline by ensuring sensitive data is not left behind after a session ends and prompting for user re-authentication, which is another layer of data protection.
- MASVS-STORAGE-2: The control ”MASVS-STORAGE-2” and the ENISA Guideline both address the management of sensitive data within the application’s context. ”MASVS-STORAGE-2” mentions the risk of sensitive data being unintentionally stored or exposed in publicly accessible locations due to certain API or system capabilities, and implies that developers should prevent this leakage. The ENISA Guideline advises clearing any sensitive data upon session termination and resetting the application state, which aligns with preventing unintentional leaks by ensuring sensitive data does not persist beyond the necessary scope of a user session. Both guidelines are concerned with the proper handling and lifecycle of sensitive data to protect user privacy and security.

3.16 Implementation Guidance (ENISA 2.16):

ENISA Secure Smartphone Development Guidance (2.16): Clear any maintained sensitive data and attempt to also terminate any server side session after application state change (e.g., termination, backgrounding). Consider a user request for application termination as a request to logout.

3.16.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: Both ”MASVS-AUTH-1” and the ENISA Guideline emphasize the importance of protecting user data and session information in the context of mobile app security. ”MASVS-AUTH-1” suggests that apps must follow best practices for secure communication with a remote endpoint, which includes proper authentication and authorization. The ENISA Guideline complements this by recommending the clearance of sensitive data and the termination of server-side sessions when the app state changes (like being terminated or moved to the background), which is a specific best practice to ensure secure protocol use mentioned in ”MASVS-AUTH-1”. Both guidelines advocate for active session management and protection of sensitive information, which correlates with each other.
- MASVS-AUTH-2: The mentioned MASVS-AUTH-2 and the ENISA Guideline both focus on the security of authentication mechanisms within apps, although they approach it from different angles. MASVS-AUTH-2 emphasizes the importance of correctly implementing authentication mechanisms such as biometrics or a local PIN code. It highlights the need for robust local authentication especially in apps that do not connect to a remote endpoint and rely on local measures for user verification. On the other hand, the ENISA Guideline is concerned with the proper management of sensitive data and session termination when the app’s state changes—for instance, when it is terminated or moved to the background. While MASVS-AUTH-2 does not explicitly mention session management or sensitive data clearance, it is implied within the broader context of proper authentication implementation. After all, part of implementing a secure authentication mechanism involves ensuring that sensitive data is not exposed when the app’s state changes and that local authentication does not become a vulnerability. Therefore, there is a correlation in that both guidelines aim to protect sensitive information and maintain the integrity of the app’s authentication mechanisms, although one is more specific to the implementation of local authentication while the other deals with session and data management upon state changes.
- MASVS-CRYPTO-2: The correlation between ”MASVS-CRYPTO-2” and the ENISA guideline provided is related to the security and management of sensitive information. While ”MASVS-CRYPTO-2” focuses on the management of cryptographic keys throughout their lifecycle, ensuring that keys are generated, stored, and protected properly to maintain the strength of the cryptography, the ENISA guideline deals with maintaining the security of sensitive data by clearing it from memory and terminating server sessions when the state of an application changes (e.g., when the app is terminated or sent to the background). Both controls are concerned with preventing unauthorized access to sensitive data and ensuring that such data does not persist in an insecure state. Poor key management could lead to sensitive information being accessible after application state changes, just as failure to clear sensitive data or terminate sessions upon state changes could lead to security breaches. Therefore, there is a connection in that both are addressing aspects of

secure information handling and lifecycle management to prevent potential cryptographic and data-related vulnerabilities.

- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA Guideline is that both are concerned with the prevention of sensitive data leakage. While "MASVS-PLATFORM-2" focuses on ensuring that WebViews are configured securely to prevent sensitive data leakage and exposure of sensitive functionality, the ENISA Guideline addresses the need to clear sensitive data and terminate server-side sessions upon application state changes such as termination or going into the background. Both guidelines aim to enhance the security of mobile applications by protecting sensitive data and recommend steps to mitigate risks associated with data exposure.
- MASVS-PLATFORM-3: Both "MASVS-PLATFORM-3" and the ENISA Guideline address the management of sensitive data within the context of an application's state changes. While "MASVS-PLATFORM-3" focuses on preventing sensitive data from being inadvertently leaked through platform mechanisms during display operations, the ENISA Guideline emphasizes the importance of clearing out sensitive data and terminating server-side sessions when the application's state changes, like termination or backgrounding. Both controls are concerned with securing sensitive data during different aspects of app lifecycle management to prevent unauthorized access or data leakage.
- MASVS-PRIVACY-1: Both the MASVS-PRIVACY-1 guideline and the ENISA guideline emphasize the importance of minimizing the access and retention of sensitive data. MASVS-PRIVACY-1 speaks to the need for applications to request only the data that is necessary for their function and with user consent, while the ENISA guideline suggests that sensitive data should be cleared and server sessions should be terminated when there is a change in the application's state. Both guidelines aim to enhance privacy and reduce the risk of data breaches by controlling how data is handled within apps, ensuring user consent, and managing the lifecycle of sensitive information.
- MASVS-PRIVACY-3: The correlation between "MASVS-PRIVACY-3" and the ENISA Guideline lies in the focus on data privacy and transparency. MASVS-PRIVACY-3 emphasizes the importance of informing users about how their data is used, which includes practices that users would not normally expect, such as background data collection. The ENISA Guideline complements this by suggesting that any sensitive data should be cleared and server-side sessions terminated when the application undergoes a state change. Both aim to protect user data and ensure users are aware of and in control of their data usage, although MASVS-PRIVACY-3 is more about informing users, while ENISA is about active data management in response to application state changes.
- MASVS-PRIVACY-4: Both MASVS-PRIVACY-4 and the ENISA guideline emphasize the importance of user control over their data and application behavior in respect to privacy. MASVS-PRIVACY-4 advocates for user mechanisms to manage, delete, and modify data, and addresses the need for re-prompting consent if more data is required, implying an ongoing control over data privacy. The ENISA guideline complements this by recommending the clearing of sensitive data and termination of server-side sessions, which is another form of giving users control by safeguarding their data when the app's state changes. Both guidelines are aligned in enhancing user privacy and control.
- MASVS-STORAGE-1: The correlation between "MASVS-STORAGE-1" and the ENISA Guideline is that both are concerned with the protection of sensitive data within a mobile application environment. "MASVS-STORAGE-1" focuses on ensuring that any sensitive data stored by the app, regardless of where it is stored, is protected appropriately. The ENISA Guideline complements this by suggesting that sensitive data should be cleared and server-side sessions terminated when the application state changes, such as when it is

terminated or sent to the background. This includes a user-triggered application termination being treated as a request to log out. Both guidelines emphasize the importance of actively managing sensitive data to prevent unauthorized access or disclosure.

- MASVS-STORAGE-2: The correlation exists because both ”MASVS-STORAGE-2” and the ENISA Guideline emphasize the importance of handling sensitive data securely within a mobile application. Specifically, MASVS-STORAGE-2 mentions the risk of sensitive data being unintentionally stored or exposed in publicly accessible locations due to developer oversights in using certain APIs or system capabilities. Similarly, the ENISA Guideline advises on the proper management of sensitive data by clearing any maintained sensitive information and attempting to terminate server-side sessions when the application's state changes, such as during termination or backgrounding. Both are focused on preventing unintended leaks of sensitive information and ensuring that sensitive data does not persist beyond the required lifecycle.

3.17 Implementation Guidance (ENISA 2.17):

ENISA Secure Smartphone Development Guidance (2.17): For applications that contain sensitive data, is also recommended to request for user re-authentication when the application state changes to background or verify that the device is secured with PIN, pattern or password.

3.17.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: Both ”MASVS-AUTH-1” and the ENISA Guideline emphasize the importance of proper authentication and potentially additional security measures when dealing with sensitive data and state changes. ”MASVS-AUTH-1” refers to ensuring that apps follow best practices for secure protocol use associated with user authentication and authorization. The ENISA Guideline specifically suggests re-authentication or checks for device security mechanisms when the application state changes, which is aligned with implementing secure practices recommended by ”MASVS-AUTH-1”. Both guidelines aim to mitigate unauthorized access and enhance the security of sensitive data.
- MASVS-AUTH-2: ”MASVS-AUTH-2” refers to the correct implementation of local biometric or PIN code authentication mechanisms in apps, which is crucial when an app doesn’t have a remote endpoint and relies completely on local authentication methods. The ENISA Guideline emphasizes the importance of re-authentication when the app goes into the background and ensuring that the device is secured with a PIN, pattern, or password. Both the MASVS standard and the ENISA Guideline focus on the security of local authentication measures and the protection of sensitive data, which shows a clear correlation in their objectives to enhance mobile app security through user authentication practices.
- MASVS-AUTH-3: Both the MASVS-AUTH-3 description and the ENISA Guideline emphasize the importance of additional authentication measures for sensitive actions or data within the app. MASVS-AUTH-3 suggests various secure methods for additional authentication such as biometric, pin, MFA code generator, etc., while the ENISA Guideline recommends re-authentication when the application state changes to the background or ensuring that the device is secured with a PIN, pattern, or password. Both references are concerned with enhancing security for sensitive functions and data by utilizing more than just the initial authentication mechanism.
- MASVS-CRYPTO-1: The correlation exists because both statements emphasize the importance of securing user data in mobile environments where physical access to the device is a possible threat vector. ”MASVS-CRYPTO-1” highlights the role of cryptography in safeguarding user data, while the ENISA guideline recommends additional security measures like re-authentication when the app goes into the background or ensuring the device itself is secured. Both are concerned with reinforcing the protection of sensitive information on mobile devices against unauthorized access.
- MASVS-CRYPTO-2: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-CRYPTO-2” directly pertains to the management of cryptographic keys through their entire lifecycle, which includes the generation, storage, and protection of these keys. Good key management practices are crucial to maintaining the effectiveness of the applied cryptography within an application. The ENISA Guideline’s recommendation to request user re-authentication when the app goes into the background or to ensure the device is secured with a PIN, pattern, or password is a measure aimed at protecting sensitive

data, likely including cryptographic keys, either in use or stored, alongside other sensitive information. While the ENISA Guideline does not specifically mention key management, requiring user re-authentication or device security indirectly supports the objective of "MASVS-CRYPTO-2" by adding an additional layer of security that prevents unauthorized access to cryptographic keys and other sensitive data if the user is not present or the device is found unsecured. Thus, the correlation exists as both are concerned with safeguarding sensitive data, where proper key management as per "MASVS-CRYPTO-2" is a critical component to ensure the cryptography's strength is not undermined by poor security practices, and the ENISA Guideline complements it by recommending authentication measures that protect the application state and the device, which contributes to the security of the cryptographic keys among other sensitive data.

- MASVS-PLATFORM-1: The correlation between "MASVS-PLATFORM-1" and the ENISA guideline centers on secure interaction with the application, specifically in terms of inter-process communication (IPC) and state changes. "MASVS-PLATFORM-1" addresses securing IPC mechanisms to ensure that data and functionality exposure is intentional and secure, which aligns with the ENISA guideline's emphasis on re-authentication or device-level security when the app state changes (e.g., going to the background). Both guidelines aim to protect sensitive data by ensuring that transitions in app interaction are safeguarded.
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the described ENISA Guideline centers around the theme of protecting sensitive data within mobile applications. "MASVS-PLATFORM-2" mentions the secure configuration of WebViews to prevent sensitive data leakage, which is in line with the ENISA Guideline's recommendation for re-authentication or device security measures when the application state changes to background. Both statements emphasize the need to implement security measures to safeguard sensitive information within mobile applications, demonstrating a clear correlation.
- MASVS-PLATFORM-3: The description of "MASVS-PLATFORM-3" suggests that the control is aimed at preventing unintentional data leakage through platform mechanisms, which could occur when sensitive data is displayed in the UI. The ENISA guideline recommends re-authentication or verification of device security when the app transitions to the background. Both the MASVS control and the ENISA guideline are concerned with protecting sensitive information in situations where there could be an increased risk of unauthorized access or disclosure, such as when the app's state changes or when the device might be shared. Thus, they both address the broader objective of ensuring sensitive data protection, albeit through different specific mechanisms.
- MASVS-PRIVACY-1: The correlation between "MASVS-PRIVACY-1" and the description related to access, user consent, and data sharing with third parties aligns with the ENISA guideline which recommends re-authentication when the app state changes or verifying device security. Both focus on protecting user data and ensuring secure access management.
- MASVS-PRIVACY-3: The correlation between "MASVS-PRIVACY-3" and the described ENISA guideline exists in the context of informing users about data handling practices and ensuring security of sensitive data. MASVS-PRIVACY-3 underlines the importance of transparency in how an app uses personal data, which includes practices like background data collection. The ENISA guideline complements this by recommending re-authentication for sensitive data when an app goes to the background or confirming device security. Both contribute to ensuring users' privacy and data security by advocating for clear communication about data use and additional security measures.

- MASVS-PRIVACY-4: Both MASVS-PRIVACY-4 and the quoted ENISA Guideline emphasize the importance of giving users control over their data and ensuring their privacy and security within the application context. MASVS-PRIVACY-4 asserts that users should have the ability to manage their data and their privacy settings, including the right to modify, delete, and control their information. It also touches on the need for re-prompting consent if more data is required than initially agreed upon. This reflects a concern for continued user awareness and control over their data. Similarly, the ENISA Guideline recommends re-authentication when an app goes to the background and encourages device-level security measures like PINs or passwords. This is also about ensuring that the user's data remains secure and under their control, especially in the case of potentially unauthorized access when the app's state changes. Both guidelines promote user control and security measures to protect user data, implying a shared goal of elevating user privacy and data integrity.
- MASVS-RESILIENCE-1: Both statements are concerned with the security and integrity of the platform upon which the application is running. "MASVS-RESILIENCE-1" emphasizes the importance of running apps on a secure and uncompromised operating system. This is because the integrity of the platform directly impacts the effectiveness of the app's security measures like secure storage and biometrics. The ENISA guideline suggests re-authentication or verification that the device is secured with a PIN, pattern, or password. Although the focus of the guideline is on user authentication when an app moves to the background, it inherently relies on the premise that the operating system's security features are intact and can be trusted. Hence, there is a correlation between the importance of platform integrity in MASVS-RESILIENCE-1 and the recommendation by ENISA for additional user authentication measures in response to state changes in an application. This ensures that sensitive data is protected even in cases where the application state changes, assuming that the platform's security features are operational.
- MASVS-STORAGE-1: The correlation between "MASVS-STORAGE-1" and the ENISA Guideline is that both are concerned with the secure handling of sensitive data within mobile applications. Specifically, "MASVS-STORAGE-1" emphasizes the importance of protecting sensitive data regardless of where it is stored, while the ENISA Guideline focuses on the security measures that should be in place when the application state changes or when verifying that the device itself is secure with a PIN, pattern, or password. Both guidelines aim to ensure that sensitive data is not compromised, whether it's at rest within the app's storage or when the app's state changes, thereby requiring certain security actions to be taken by the user or the system.
- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" and the ENISA guideline lies in the concern for the protection of sensitive data within mobile applications. "MASVS-STORAGE-2" pertains to the prevention of unintentional leaks of sensitive data due to the use of certain APIs and system capabilities, which includes ensuring that data is not inadvertently stored in publicly accessible locations or exposed through mechanisms like backups or logs. The ENISA guideline complements this concern by recommending additional security measures, such as user re-authentication when the app goes into the background, or verifying that the device has proper security controls like PIN, pattern, or password. Both guidelines are aimed at reducing the risk of sensitive data exposure, though MASVS-STORAGE-2 focuses on the development side and the ENISA guideline emphasizes device and user session security.

3.18 Implementation Guidance (ENISA 2.18):

ENISA Secure Smartphone Development Guidance (2.18): For platforms that support application component history stack (e.g., Android), always clear the stack on session or app termination and user's request to logout.

3.18.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between "MASVS-AUTH-1" and the ENISA guideline about clearing the stack on session or app termination and user's request to logout is that they both relate to ensuring the security of user sessions within mobile applications. "MASVS-AUTH-1" underlines the importance of following best practices for secure user authentication and authorization, which would encompass proper session management. This includes securely handling session terminations, as suggested by ENISA, to prevent unauthorized access or leakage of sensitive information after a user has logged out or ended their session. Clearing the history stack is a specific example of how an app can maintain a secure user environment in line with the principles described in "MASVS-AUTH-1".
- MASVS-AUTH-2: The mobile application security verification standard (MASVS) requirement AUTH-2, which emphasizes the need for correct implementation of local authentication mechanisms, such as biometrics or local PIN code, can be correlated with the ENISA guideline pertaining to application component history stack management on platforms like Android. While MASVS-AUTH-2 is focused on the integrity and security of local authentication methods, the ENISA guideline addresses the need to maintain session integrity by clearing the stack to prevent any potential residue of sensitive information or authentication states after session or app termination or when a user logs out. Both guidelines deal with maintaining the security boundaries of a user session and protecting against unauthorized access. Clearing the stack as per the ENISA guideline complements the secure implementation demanded by MASVS-AUTH-2, ensuring that after authentication, the session does not leave traces that could be exploited.
- MASVS-NETWORK-1: The correlation between "MASVS-NETWORK-1" and the ENISA guideline about clearing the application component history stack is centered on the theme of protecting the confidentiality and integrity of user data. While MASVS-NETWORK-1 specifically addresses the need for secure network communications by ensuring data privacy and integrity in transit through encryption and endpoint authentication (such as with TLS), the ENISA guideline focuses on protecting sensitive information that may reside in the application's history stack by advising to clear it to prevent data leakage upon session or app termination. Both controls are concerned with different aspects of protecting sensitive information within the scope of a mobile application's lifecycle. MASVS-NETWORK-1 deals with data as it moves across the network, whereas the ENISA guideline deals with data at rest within the application's history stack. They are complementary in a comprehensive mobile app security strategy that considers both data in transit and data at rest.
- MASVS-PLATFORM-1: "MASVS-PLATFORM-1" refers to secure interaction with Inter-Process Communication (IPC) mechanisms, which includes protecting data and functionality exposed through these mechanisms. The ENISA guideline about clearing the application component history stack upon session or app termination aligns with MASVS-PLATFORM-1, as it is a measure to protect against potential IPC vulnerabilities by ensuring that sensitive information is not inadvertently exposed through historical data that could

be accessed by other applications or the user after a session ends. Both concepts relate to safeguarding interactions within the app's environment and between other apps or the system.

- MASVS-PRIVACY-2: Both "MASVS-PRIVACY-2" and the ENISA Guideline focus on the protection of user privacy. MASVS-PRIVACY-2 emphasizes the importance of using unlinkability techniques to prevent user identification and tracking, which includes the handling of user data and session information in a way that preserves anonymity. The ENISA guideline similarly addresses privacy by recommending the clearing of the application component history stack upon session or app termination and upon a user's request to logout, ensuring that past user actions are not traceable, thus contributing to the user's unlinkability and privacy. Both guidelines aim to minimize the risk of user identification and tracking through the proper management of sensitive information.
- MASVS-PRIVACY-4: Both the "MASVS-PRIVACY-4" requirement and the mentioned ENISA Guideline aim to enhance user privacy and control over their data. MASVS-PRIVACY-4 focuses on providing users with mechanisms to manage their data and privacy settings, which aligns with the ENISA guideline that emphasizes the importance of clearing the history stack on session termination or logout to prevent data leaks or unintended persistence of sensitive information. Both advocate for proactive measures in protecting user data privacy within the app's lifecycle.
- MASVS-RESILIENCE-1: Although "MASVS-RESILIENCE-1" and the ENISA Guideline provided address different aspects of mobile app security, they are correlated because they both aim to strengthen the security posture of mobile applications by ensuring the application operates in a secure environment and manages its data securely. "MASVS-RESILIENCE-1" focuses on ensuring that the operating system platform has not been tampered with, maintaining the integrity of security features like secure storage, biometrics, and sandboxing, all of which rely on the platform being secure and trustworthy. If the platform is compromised, the application's data could be at risk. The ENISA Guideline emphasizes the need for proper session management by clearing the application component history stack upon session or app termination, or upon the user's request to logout. This practice helps to prevent unauthorized access to sensitive data left in the history stack and ensures that leftover data from previous sessions is not accessible to potentially malicious actors. Both controls contribute to the resilience of mobile applications by mitigating risks associated with compromised platform security and improper session handling. They complement each other in that ensuring the platform is secure (MASVS-RESILIENCE-1) provides a foundation upon which specific security measures like proper session management (ENISA Guideline) can effectively protect application data.
- MASVS-STORAGE-2: The ENISA guideline mentioned is related to preventing unintentional leakage of sensitive data when an app's state is stored in a publicly accessible location such as component history stacks. The MASVS-STORAGE-2 control addresses the prevention of leaks that can occur when sensitive data is unintentionally stored or exposed to public locations due to use of certain APIs or system capabilities, which may include history stacks. Both are concerned with the secure handling of sensitive data to prevent accidental exposure and align in the context of managing sensitive information upon session or app termination.

3.19 Implementation Guidance (ENISA 2.19):

ENISA Secure Smartphone Development Guidance (2.19): Ensure that the app runs with user privileges (unprivileged) on the end user device (does not require a rooted or a jailbroken device). Verify that it does not request more access authorizations to system resources and rights in the execution environment than the absolutely necessary (least privilege principle)

3.19.1 OWASP MASVS MAPPING

- MASVS-PLATFORM-1: Both the MASVS-PLATFORM-1 requirement and the ENISA guideline emphasize the principle of least privilege. MASVS-PLATFORM-1 focuses on secure IPC mechanisms, ensuring that only necessary data or functionality is exposed intentionally, whereas the ENISA guideline stresses that the app should run with only the necessary user privileges and should not require rooted or jailbroken devices to operate. Both aim to restrict the app's access to the minimum required for proper functionality, thereby mitigating risks related to excessive permissions that can lead to security vulnerabilities.
- MASVS-PLATFORM-2: The correlation exists because both "MASVS-PLATFORM-2" and the ENISA Guideline address the principle of least privilege. "MASVS-PLATFORM-2" relates to the secure configuration of WebViews to prevent access to sensitive features or data, which aligns with the ENISA Guideline's focus on running the app with user privileges and not requesting excessive system resources and rights. Both aim to minimize unnecessary access to sensitive functionalities, thus enhancing the security posture of the application.
- MASVS-PLATFORM-3: Although "MASVS-PLATFORM-3" and the ENISA Guideline mentioned do not directly reference each other, they both align with the principle of ensuring that sensitive data is protected and that applications adhere to the principle of least privilege. "MASVS-PLATFORM-3" is focused on preventing unintentional leaks of sensitive information via platform mechanisms, while the ENISA Guideline emphasizes that apps should run with user privileges and request only the necessary authorizations to system resources, thereby minimizing the risk of sensitive data exposure due to excessive privileges. Both controls aim to safeguard sensitive data by ensuring it is handled appropriately within their respective scopes. The correlation lies in their shared goal of enhancing security and protecting user data by following best practices.
- MASVS-PRIVACY-1: Both "MASVS-PRIVACY-1" and the ENISA guideline emphasize the principle of requesting the minimal amount of access necessary for functionality. MASVS-PRIVACY-1 focuses on data minimization and informed consent, limiting access to user data and sharing with third parties, which aligns with the least privilege principle mentioned in the ENISA guideline which dictates that an app should not request more access authorizations than necessary. Both advocate for the protection of user data and system resources by restricting app permissions to what is strictly required for app operation.
- MASVS-PRIVACY-3: Both "MASVS-PRIVACY-3" and the ENISA guideline emphasize the principle of minimizing user data exposure and ensuring users are aware and in control of how their data is used. "MASVS-PRIVACY-3" focuses on transparency around data practices and adhering to platform guidelines, while the ENISA guideline concentrates on running the app with the least privileges required, avoiding unnecessary access to system

resources, and ensuring that the app doesn't operate on a compromised device. Both aim to protect user data and enhance privacy by advocating for the least amount of data access and user permissions necessary for app functionality.

- MASVS-RESILIENCE-1: The MASVS-RESILIENCE-1 is directly correlated with the ENISA guideline on ensuring that the app runs with user privileges and does not require a rooted or jailbroken device. The reasoning behind this correlation is simple: Both the MASVS-RESILIENCE-1 and the ENISA guideline emphasize the importance of the app operating in a secure and unmodified OS environment. Running on a tampered platform, like a rooted or jailbroken device, may disable important security features, therefore putting app data at risk as stated in the MASVS-RESILIENCE-1 description. Similarly, the ENISA guideline insists on the app running with unprivileged user rights and adhering to the least privilege principle, meaning it should only request access to system resources that are absolutely necessary for its operation. This is to prevent the app from having more permissions than required, which could potentially be exploited if the device were compromised. Both guidelines are aimed at maintaining the integrity and security of the app and the data it handles by ensuring that the underlying platform remains untampered with and secure.
- MASVS-RESILIENCE-2: The correlation between "MASVS-RESILIENCE-2" and the ENISA Guideline relates to the concept of maintaining the integrity and security context of a mobile application on a user-controlled device. "MASVS-RESILIENCE-2" talks about protecting the app from being modified to ensure its intended functionality is not compromised. This includes preventing running of a modified version of the app or uploading a backdoored version to third-party app stores, which aligns with the ENISA Guideline's emphasis on running the app with user privileges and not requiring rooted or jailbroken devices. Both are concerned with limiting the scope of execution to what is necessary for functionality while protecting against elevation of privileges or modifications that could harm the integrity of the app or device. The practice of "least privilege," as mentioned in the ENISA Guideline, is a fundamental principle of information security that supports the objective of "MASVS-RESILIENCE-2".
- MASVS-RESILIENCE-4: The correlation between "MASVS-RESILIENCE-4" and the ENISA Guideline is evident in that both recommend measures to increase the difficulty for attackers to manipulate the app's behavior. "MASVS-RESILIENCE-4" focuses on making dynamic analysis and the modification of code at runtime more challenging. Preventing dynamic instrumentation is a security practice that aligns with not providing attackers with the level of privilege that would be available if the app required a rooted or jailbroken device. The ENISA Guideline emphasizes running the app with user privileges, adhering to the principle of least privilege, and not requiring more system resources and rights than necessary. By following this guideline, an app limits the ability of an attacker to gain elevated privileges and perform actions like dynamic analysis or code injection, which could compromise the app's integrity. Both the MASVS requirement and the ENISA Guideline aim to limit the attack surface and promote a secure app execution environment, thereby achieving a similar objective of resilience against runtime manipulation and exploitation.
- MASVS-STORAGE-1: Both the Mobile Application Security Verification Standard (MASVS) "MASVS-STORAGE-1" control and the ENISA guideline you mentioned touch on the principle of protecting sensitive data within mobile applications, albeit from different angles. MASVS-STORAGE-1 emphasizes the need for sensitive data that is stored locally by the app to be properly protected regardless of where it is stored. This means implementing security measures to ensure that any sensitive information kept on the device is not easily accessible by unauthorized parties. The ENISA guideline focuses on ensuring that apps run with the least privileges necessary, avoiding the need for rooted or jailbroken

devices. This is to prevent the app from having more access to system resources and rights than needed, which aligns with the least privilege principle. By doing so, it implicitly contributes to the protection of sensitive data by reducing the attack surface that could be used to compromise data stored on the device. In both cases, the underlying goal is to minimize the risks to sensitive data handled by mobile applications, hence there is a correlation between MASVS-STORAGE-1 and the mentioned ENISA guideline.

- MASVS-STORAGE-2: The MASVS-STORAGE-2 description highlights concerns about sensitive data being unintentionally stored or exposed in publicly accessible locations. This is directly related to the principle of least privilege mentioned in the ENISA Guideline, as unnecessarily high privileges or access authorizations could lead to increased risk of such unintentional exposure. Running an app with only the necessary user privileges reduces the risk of sensitive data being accessed or exposed through system capabilities such as backups, logs, or other means. Both statements emphasize the importance of minimizing the app's access rights to what's necessary to prevent unauthorized access to sensitive information.

Chapter 4
Handle authentication and authorization factors securely on the device

User account credentials, if stolen, not only provide unauthorized access to the mobile backend service but potentially to other services and accounts owned by the user. Mobile applications need to designed to protect user credentials to protect the users as well as the application's backend infrastructure.

4.1 Implementation Guidance (ENISA 3.1):

ENISA Secure Smartphone Development Guidance (3.1): Instead of passwords consider using longer term authorization tokens that can be securely stored on the device (as per the OAuth model). Secure the tokens in transit (using TLS). Tokens can be issued by the backend service after verifying the user credentials initially. The tokens should be time bounded to the specific service as well as revocable (if possible server side), thereby minimizing the damage in loss scenarios. Use the latest versions of the authorization standards (such as OAuth 2.0). Make sure that these tokens expire as frequently as practicable.

4.1.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The description of ”MASVS-AUTH-1” mentions that apps connecting to a remote endpoint require user authentication and the enforcement of authorization mechanisms, which aligns with the ENISA Guideline’s recommendation of using authorization tokens, like those in the OAuth model. Both stress the importance of secure transmission (e.g., using TLS) and the app’s role in ensuring secure use of these authentication and authorization protocols. They also converge on the idea that these tokens should be time-bounded and revocable to minimize damage in loss scenarios. The utilization of the latest versions of authorization standards, such as OAuth 2.0, is emphasized by ENISA and is part of following best practices as mentioned in the MASVS authentication requirement.
- MASVS-AUTH-3: MASVS-AUTH-3 suggests additional forms of authentication beyond traditional passwords, which aligns with the ENISA guideline advocating for long-term authorization tokens, like those used in the OAuth model. Both guidelines emphasize enhancing security by not relying solely on passwords and ensuring the secure implementation of authentication measures. The need for tokens to be securely stored, time-bounded, and revocable if possible also corresponds with the MASVS-AUTH-3 requirement for secure implementation. Hence, the MASVS-AUTH-3 correlates with the ENISA guideline in emphasizing security of authentication mechanisms that go beyond basic password protection.
- MASVS-CRYPTO-1: The MASVS-CRYPTO-1 guideline emphasizes the importance of cryptography in securing user data, especially in mobile environments where physical access to a device is a common risk. The ENISA guideline complements this by providing a specific application of cryptography best practices: the use of authorization tokens like OAuth 2.0 to secure user credentials and data. Both guidelines acknowledge the necessity of protecting sensitive information using secure methods both at rest and in transit, which in the ENISA guideline’s case is done by using tokens secured by TLS and adhering to current standards. This correlation signifies a common goal to safeguard user data against unauthorized access, thus aligning with the principles in MASVS-CRYPTO-1.
- MASVS-CRYPTO-2: Both ”MASVS-CRYPTO-2” and the ENISA Guideline emphasize the importance of secure management of authentication credentials, which includes handling cryptographic keys and authorization tokens. ”MASVS-CRYPTO-2” focuses on the lifecycle of cryptographic keys which includes generation, storage, and protection. The ENISA guideline complements this by discussing secure storage and transmission of long-term authorization tokens, as well as ensuring they are time-bounded and revocable to minimize damage in case of loss. Both guidelines are aligned in the viewpoint that security

of credentials, whether they are keys or tokens, is critical and that strong mechanisms should be in place to manage them properly throughout their lifecycle.

- MASVS-NETWORK-1: Both the MASVS-NETWORK-1 description and the ENISA Guideline emphasize the importance of securing data in transit. MASVS-NETWORK-1 discusses ensuring data privacy and integrity by encrypting data and authenticating endpoints, similar to how TLS operates. The ENISA Guideline focuses on using authorization tokens securely, which involves TLS to secure the tokens in transit. Both are concerned with using proper security measures to protect network communications and data transmission, suggesting a correlation between the principles outlined in MASVS-NETWORK-1 and the practices recommended by the ENISA Guideline.
- MASVS-PLATFORM-3: Both the MASVS-PLATFORM-3 description and the ENISA Guideline address concerns regarding the secure handling of sensitive data. The MASVS-PLATFORM-3 focuses on preventing leaks of sensitive data displayed in the UI, while the ENISA Guideline suggests using longer-term authorization tokens instead of passwords for secure storage and transmission, minimizing damage in loss scenarios. Both guidelines are focused on increasing the security of sensitive information, with MASVS-PLATFORM-3 dealing with the display and potential leakage of such data, and ENISA focusing on the secure management and revocation of authorization tokens to limit the impact of compromised credentials. They correlate because both aim to protect sensitive data from unauthorized access or exposure.
- MASVS-STORAGE-1: The correlation is that both ”MASVS-STORAGE-1” and the ENISA guideline emphasize the importance of protecting sensitive data stored on a mobile device. ”MASVS-STORAGE-1” refers to ensuring that any sensitive data intentionally stored by the app is properly protected, regardless of the storage location. The ENISA guideline focuses on using authorization tokens instead of passwords and securing these tokens both in transit (with TLS) and at rest on the device, which is in line with protecting sensitive data storage as per ”MASVS-STORAGE-1”. Both are concerned with minimizing the risk of data exposure and suggest mechanisms for securing data, with an emphasis on proper data storage practices and using secure tokens instead of passwords.
- MASVS-STORAGE-2: The MASVS-STORAGE-2 guideline is concerned with preventing unintentional leaks of sensitive data due to improper handling by developers, which could occur if authorization tokens are not securely stored or are exposed in logs or backups. The ENISA Guideline also emphasizes the secure storage of authorization tokens on the device, secure transmission (using TLS), and additional security measures such as bounding the tokens to a specific service, making them revocable, and expiring them frequently. Both sets of guidelines aim to protect sensitive information, such as authorization tokens, from being compromised, and to minimize the risk of leaks and unauthorized access, thus showing a clear correlation.

4.2 Implementation Guidance (ENISA 3.2):

ENISA Secure Smartphone Development Guidance (3.2): In the case passwords need to be stored on the device, leverage the encryption and key-store mechanisms provided by the mobile OS to securely store passwords, password equivalents and authorization tokens. Never store passwords in clear text. Do not store passwords or long term session IDs without appropriate encryption.

4.2.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA Guideline is evident in the focus on secure user authentication and the proper handling of credentials and authorization tokens. ”MASVS-AUTH-1” emphasizes that apps must follow best practices for secure communication with remote endpoints, which includes how user credentials and authorization procedures are managed. The ENISA Guideline complements this by providing a specific best practice regarding the storage of passwords and authorization tokens on the device, stipulating that they should be securely stored using encryption and key-store mechanisms offered by the mobile OS, and never in clear text. Both guidelines are concerned with preventing unauthorized access and ensuring the confidentiality and integrity of user credentials.
- MASVS-AUTH-2: ”MASVS-AUTH-2”, referring to the correct implementation of authentication mechanisms including biometrics or a local PIN code, implicitly demands secure handling of authentication credentials, which correlates with the ENISA guideline emphasizing the use of encryption and secure storage mechanisms provided by the mobile OS. Although ”MASVS-AUTH-2” does not explicitly mention passwords, the principle of securely implementing authentication mechanisms applies to all forms of credentials, aligning with ENISA’s guideline against storing passwords or equivalent tokens in clear text and advocating for encryption.
- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3” and the ENISA guideline is that both are focused on enhancing the security of authentication mechanisms within mobile applications. MASVS-AUTH-3 deals with implementing additional authentication measures for sensitive actions and ensuring they are implemented securely. This aligns with the ENISA guideline which emphasizes the secure storage of passwords and authorization tokens using encryption and key-store mechanisms provided by the mobile OS, rather than storing them in clear text or without proper encryption. Both are concerned with protecting the credentials and tokens that would grant access to sensitive actions or areas within the app.
- MASVS-CRYPTO-1: Both the ”MASVS-CRYPTO-1” description and the ENISA Guideline emphasize the importance of proper cryptographic measures to protect sensitive data such as passwords and authorization tokens, especially on mobile devices where physical access is a potential risk. They both advise against storing sensitive data in clear text and suggest using the mobile OS’s encryption and key storage mechanisms to ensure data security.
- MASVS-CRYPTO-2: Both ”MASVS-CRYPTO-2” and the ENISA Guideline emphasize the importance of secure key management and the protection of sensitive information. ”MASVS-CRYPTO-2” focuses on the management of cryptographic keys throughout their lifecycle to avoid compromising cryptography due to poor key management. The ENISA

Guideline complements this by providing a specific application of key management, stating that passwords and authorization tokens should be stored securely using encryption and key-store mechanisms provided by the mobile OS, never in clear text. Both sources highlight that proper encryption and secure storage practices are essential for maintaining the confidentiality and integrity of sensitive data.

- MASVS-NETWORK-1: Both the MASVS-NETWORK-1 requirement and the ENISA Guideline emphasize the importance of data security, particularly in the context of network communications and data storage. MASVS-NETWORK-1 focuses on ensuring data privacy and integrity during data transit by advocating for secure connections, encryption, and endpoint authentication. Similarly, the ENISA Guideline stresses the secure storage of sensitive information such as passwords and authorization tokens by suggesting the use of encryption and secure storage mechanisms provided by the mobile OS. Both guidelines aim to prevent the exposure of sensitive data either in transit or at rest and to avert potential security risks associated with inappropriate handling of such data.
- MASVS-PLATFORM-1: Both the MASVS-PLATFORM-1 description and the ENISA Guideline emphasize the importance of using platform-provided mechanisms to ensure the secure handling of sensitive data. MASVS-PLATFORM-1 focuses on secure interactions through IPC mechanisms, whereas the ENISA Guideline specifically addresses the secure storage of passwords and authorization tokens using encryption and key-store features of the mobile operating system. Both reflect the principle of leveraging native security features to protect sensitive data and functionality within the app ecosystem.
- MASVS-PLATFORM-2: Both the MASVS-PLATFORM-2 and the ENISA guideline emphasize the importance of securing sensitive data and functionality. MASVS-PLATFORM-2 focuses on the secure configuration of WebViews to prevent data leakage and exposure of sensitive functionality, which could include password handling. The ENISA guideline specifically addresses the secure storage of passwords and authorization tokens, advocating against clear text storage and in favor of encryption and key-store mechanisms. Thus, there is a correlation in the shared objective of preventing sensitive data from being compromised through proper security measures.
- MASVS-PLATFORM-3: The correlation between ”MASVS-PLATFORM-3” and the ENISA guideline you mentioned is that both relate to the protection of sensitive data in mobile applications. ”MASVS-PLATFORM-3” focuses on the precautions necessary to prevent sensitive data from being leaked through the user interface due to platform features or user behavior, such as auto-generated screenshots or shoulder surfing. The ENISA guideline specifically addresses the secure storage of passwords and authorization tokens, emphasizing the use of encryption and key storage mechanisms provided by the mobile operating system to prevent storing sensitive information in clear text. Both sets of recommendations are concerned with safeguarding sensitive information from unintended disclosure and ensuring that application developers employ secure methods to manage and display such information.
- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the described ENISA Guideline is that they both emphasize the importance of relying on the security features provided by the mobile operating system. The MASVS-RESILIENCE-1 control discusses the risks associated with running an app on a tampered platform that might have compromised security features. Since one of the controls reliant on a secure platform is secure storage (which is required to securely store sensitive data such as passwords and tokens), it aligns with the ENISA Guideline’s directive to use encryption and key-store mechanisms provided by the OS to securely store passwords and other authorization credentials, rather than storing them in clear text or without appropriate encryption. Both

guidelines advocate for leveraging the platform's inherent security mechanisms to protect sensitive information.

- MASVS-STORAGE-2: The control "MASVS-STORAGE-2" from the Mobile Application Security Verification Standard (MASVS) is correlated with the ENISA Guideline regarding the secure storage of sensitive information. Both are focused on preventing sensitive data such as passwords and tokens from being stored or exposed in insecure ways. While the MASVS-STORAGE-2 control addresses unintentional leaks of sensitive data due to the use of certain APIs or system capabilities, the ENISA guideline provides a best practice for intentionally storing such data, emphasizing the use of encryption and secure storage mechanisms provided by the mobile OS. Both guidelines aim to prevent the exposure of sensitive data through proper storage and handling practices.

4.3 Implementation Guidance (ENISA 3.3):

ENISA Secure Smartphone Development Guidance (3.3): Leverage the provided key-store mechanisms at the highest supported security level and only when a device passcode has been set. If possible request key-store items to be protected after the device is locked and to remain only in the current device (e.g., exclude these items from backups and cloud synchronization).

4.3.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: ”MASVS-AUTH-1” talks about ensuring secure usage of protocols for user authentication and authorization, which correlates to the ENISA guideline that suggests leveraging key-store mechanisms at the highest security level and only when a device passcode is set. Both are concerned with improving security during the process of user authentication and protecting sensitive data, implying that key store access should be restricted to authenticated users and that sensitive key store items should be further protected when the device is locked and not be transferred out of the device. This establishes a complementary relationship between the secure user authentication practices mentioned in ”MASVS-AUTH-1” and the key storage and protection measures prescribed in the ENISA guideline.
- MASVS-AUTH-2: The correlation exists because both statements emphasize the importance of proper implementation and security measures related to local authentication methods. The Mobile Application Security Verification Standard (MASVS) ”MASVS-AUTH-2” focuses on the need for correct implementation of biometric authentication and local PIN codes, which is a form of local app authentication that may function without communication to a remote endpoint. It implicitly requires secure storage of authentication data and secure mechanisms. The ENISA guideline also underlines secure practices related to key storage, mentioning that keys should be stored at the highest security level provided, only accessible when a device passcode is set, and protected when the device is locked. Additionally, it suggests that keys should remain only on the current device by excluding them from backups and cloud synchronization to prevent unauthorized access. Both MASVS-AUTH-2 and the ENISA guideline aim to ensure that local authentication methods and key storage mechanisms are secure and resilient against unauthorized access, thus maintaining the overall security of the application and the device.
- MASVS-AUTH-3: While ”MASVS-AUTH-3” discusses the general concept of adding an additional layer of authentication for sensitive actions in the app, the ENISA guideline suggests a specific security measure involving the secure storage of cryptographic keys. Both of these points relate to enhancing the security of sensitive operations on the app by requiring additional authentication beyond the basic login credentials and utilizing secure storage mechanisms, which suggests a correlation in the overall aim of strengthening authentication and security practices.
- MASVS-CRYPTO-1: The ”MASVS-CRYPTO-1” description emphasizes the importance of cryptography in mobile environments, especially considering physical access to devices. It suggests adherence to general cryptography best practices often found in external standards. The ENISA guideline complements this by recommending the use of key-store mechanisms at high security levels, specifically when a device passcode is set, and suggests additional security measures like protecting key-store items after the device is locked and

ensuring they remain only on the current device, excluding them from backups and cloud synchronization. Both statements stress securing user data through proper cryptographic practices and leveraging security features to protect data, suggesting a correlation.

- MASVS-CRYPTO-2: Both ”MASVS-CRYPTO-2” and the mentioned ENISA Guideline emphasize the importance of secure key management in the context of cryptographic implementations. ”MASVS-CRYPTO-2” highlights the need for proper key management throughout keys' lifecycle, which includes secure generation, storage, and protection of cryptographic keys. The ENISA Guideline aligns with this by specifically recommending the use of secure key-store mechanisms, ensuring protection at the highest security level after the device is locked, and restricting keys to the device by avoiding backups and cloud synchronization. Both sources are concerned with mitigating the risks associated with poor key management and thus preserving the efficiency of cryptographic measures.
- MASVS-NETWORK-1: There is a correlation between ”MASVS-NETWORK-1” and the ENISA guideline. The ”MASVS-NETWORK-1” focuses on the security of data in transit by ensuring the proper setup of secure connections, which includes the use of encryption and endpoint authentication—this aims to maintain data privacy and integrity. Similarly, the ENISA guideline stresses the importance of using key-store mechanisms securely, indicating that data protection should extend to data at rest by ensuring that sensitive key-store items are only accessible when the device is unlocked and are not compromised through backups or cloud synchronization. Both the MASVS standard and the ENISA guideline emphasize securing data whether in transit (as in MASVS-NETWORK-1) or at rest (as in ENISA), and while they focus on different aspects of security, the underlying principle of ensuring data privacy and integrity remains the same.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the described ENISA guideline is that they both focus on securing interactions and data within a mobile environment. ”MASVS-PLATFORM-1” addresses the security of Inter-Process Communication (IPC) mechanisms, which could involve the sharing and storage of sensitive data such as keys. The ENISA guideline focuses on utilizing the platform's key-store mechanisms securely, which includes protecting the key-store items when the device is locked and ensuring they are not included in backups or cloud synchronization, which is a form of IPC as keys might be shared or accessed by other processes or applications. Both aim to uphold the security of data, especially in interactions that can involve other apps, the operating system, or potentially malicious actors.
- MASVS-PLATFORM-3: The correlation between ”MASVS-PLATFORM-3” and the ENISA guideline is based on their shared concern for protecting sensitive data from unintentional leaks and ensuring its security. ”MASVS-PLATFORM-3” addresses the risk of exposing sensitive UI data like passwords or OTP codes due to platform mechanisms or accidental disclosures. The ENISA guideline similarly emphasizes using key-store mechanisms to safeguard sensitive items, particularly when device passcodes are set and when the device is locked, to prevent them from being included in backups or cloud synchronization. Both guidelines are focused on enhancing data protection at the platform level and minimizing the risk of sensitive data exposure.
- MASVS-RESILIENCE-1: The correlation is that both ”MASVS-RESILIENCE-1” and the ENISA guideline focus on ensuring that security features provided by the underlying platform are leveraged effectively. MASVS-RESILIENCE-1 emphasizes the necessity of running an app on a secure and uncompromised operating system for the platform's security features to be trusted, while the ENISA guideline specifically advises using key-store mechanisms at the highest security level, particularly when a device passcode is set, and to enhance protection by not including sensitive key-store items in backups and cloud

synchronization. Both guidelines implicitly require a trust in the platform's security, highlighting the importance of the operating system's integrity for safeguarding app data and using platform features like secure storage and sandboxing effectively.

- MASVS-RESILIENCE-2: Both "MASVS-RESILIENCE-2" and the ENISA Guideline cited focus on enhancing the security and integrity of the app and the environment it runs in. "MASVS-RESILIENCE-2" is about preventing unauthorized modifications to the app, which aligns with the ENISA Guideline's emphasis on using key-store mechanisms to protect sensitive information, particularly in a secured environment where a device passcode is set. Both aim to safeguard against tampering and unauthorized use.
- MASVS-STORAGE-1: The correlation exists because both "MASVS-STORAGE-1" and the ENISA guideline are concerned with the secure handling and storage of sensitive data on mobile devices. "MASVS-STORAGE-1" focuses on ensuring that any sensitive data stored by the app is adequately protected, regardless of its location, which includes the use of internal app storage or public folders. The ENISA guideline specifically advises leveraging key-store mechanisms at the highest security level, to protect key material when a device passcode is set, and to ensure such security elements are not compromised through backups or cloud synchronization. Both directives emphasize the importance of robust security measures for data at rest, particularly when it comes to sensitive information. The storage of sensitive data should be done in a manner that precludes unauthorized access, which aligns with the ENISA's advice to use device passcode-linked key-store mechanisms. The measure of protecting key-store items after the device is locked complements the MASVS requirement for proper protection of intentional data storage by apps, as it suggests an additional layer of security for encryption keys that protect this data.
- MASVS-STORAGE-2: Both the MASVS-STORAGE-2 description and the ENISA guideline address the prevention of unintentional leaks of sensitive data due to the use of certain APIs, system capabilities, or configuration settings. MASVS-STORAGE-2 mentions the need to avoid storing sensitive data in publicly accessible locations or where it can be exposed by system features like backups or logs, which aligns with the ENISA recommendation to leverage key-store mechanisms at the highest security level, specifically when device passcode is set, and to exclude sensitive items from backups and cloud synchronization. Both are concerned with ensuring that sensitive data remains secure and is not inadvertently disclosed through common data handling processes.

4.4 Implementation Guidance (ENISA 3.4):

ENISA Secure Smartphone Development Guidance (3.4): Consider purging credentials or keys from memory after use. Avoid automatic memory managed structures (e.g., controlled by garbage collector) and immutable objects for maintaining the keys. Prefer to immediately zero out the memory containing the data after use rather than waiting for the garbage collection mechanism.

4.4.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the mentioned ENISA Guideline exists in the context of proper handling of authentication and authorization mechanisms within an app, specifically in relation to securely managing sensitive data like credentials or keys in memory. ”MASVS-AUTH-1” emphasizes the need for apps to adhere to best practices for secure communication protocols, of which secure handling of credentials and keys is a critical component. The ENISA Guideline further details this by recommending the purging of such sensitive data from memory after use to reduce the risk of unauthorized access. Both guidelines aim to secure user authentication and authorization by advocating for better control and management of sensitive information within the app’s lifecycle.
- MASVS-AUTH-2: The correlation between ”MASVS-AUTH-2” which talks about the importance of correctly implementing authentication mechanisms, particularly for apps that may rely fully on local authentication, and the ENISA Guideline related to purging credentials or keys from memory after use, lies in the realm of securing sensitive authentication data. Both guidelines emphasize the need to implement security measures that protect authentication information. MASVS-AUTH-2 highlights the need for robust implementation of authentication mechanisms such as biometrics or PIN codes, while the ENISA Guideline focuses on the security of credentials or keys in memory, recommending their immediate removal after use to prevent unauthorized access or leakage. They both aim to enhance the security of authentication processes by addressing different aspects of the authentication data lifecycle.
- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3” and the ENISA guideline exists in the context of safeguarding sensitive actions within the application. MASVS-AUTH-3 emphasizes the need for additional forms of authentication to enhance security for sensitive actions. The ENISA guideline provides a specific best practice for handling credentials and keys securely by purging them from memory after use to prevent unauthorized access or exploitation. Both MASVS-AUTH-3 and the ENISA guideline are focused on enhancing the security of authentication mechanisms and protecting sensitive operations within an app from potential threats. Adding extra authentication steps as recommended by MASVS-AUTH-3 means there will be additional credentials or keys to manage, hence the relevance of the ENISA guideline which spells out how to handle these sensitive elements securely, including the avoidance of auto-managed memory structures which might delay the purging of keys.
- MASVS-CRYPTO-1: Both the MASVS-CRYPTO-1 description and the ENISA Guideline emphasize the importance of proper management and handling of cryptographic keys and sensitive data in memory. While MASVS-CRYPTO-1 underscores the importance of cryptography for securing user data, particularly in a mobile environment where physical

device access is a risk, the ENISA Guideline provides specific advice on how to manage cryptographic keys and sensitive data in memory to prevent leakage. This includes purging keys after use and avoiding structures that are not immediately wiped, such as those managed by garbage collectors or immutable objects. Both are essentially aimed at enhancing data security through proper cryptographic practices and memory handling to prevent unauthorized access.

- MASVS-CRYPTO-2: The correlation exists because both ”MASVS-CRYPTO-2” and the ENISA guideline focus on secure key management practices. The MASVS requirement emphasizes managing cryptographic keys throughout their lifecycle, which implicitly includes safely handling keys during use and post-use, aligning with the ENISA guideline's recommendation to purge keys from memory after use to prevent them from being compromised. Avoiding automatic memory-managed structures and preferring immediate memory zeroing after use are specific strategies to ensure keys are not inadvertently exposed, which supports the concept of carefully managing and protecting keys as stated in MASVS-CRYPTO-2.
- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the ENISA Guideline mentioned is that both are focused on ensuring the security and integrity of the application by taking into account the platform and environment where the app operates. ”MASVS-RESILIENCE-1” is about validating that the operating system has not been compromised and that its security features are operational, which directly affects how security mechanisms such as secure storage and sandboxing function. On the other hand, the ENISA guideline addresses the security practice of managing sensitive data (credentials or keys) in memory and the importance of actively clearing such data immediately after use to prevent it from being compromised, which is an action indicative of not entirely trusting the platform or its automatic memory management features such as garbage collection. Both guidelines point towards a defensive stance against potential tampering or exploitation of the app's runtime environment.
- MASVS-STORAGE-2: The correlation exists because both statements are addressing the safeguarding of sensitive data. ”MASVS-STORAGE-2” addresses the issue of sensitive data being unintentionally stored or exposed due to the misuse of APIs or system capabilities, suggesting that developers should be aware of and prevent such leaks. The ENISA guideline also addresses the protection of sensitive data, specifically recommending that credentials or keys be purged from memory after use to avoid exposure through memory management mechanisms or immutable objects. Both are concerned with preventing sensitive data from being available longer than necessary or being unintentionally exposed.

4.5 Implementation Guidance (ENISA 3.5):

ENISA Secure Smartphone Development Guidance (3.5): If the credentials or keys appear in the user interface (UI) components, try to release the UI frames immediately after use.

4.5.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: Both ”MASVS-AUTH-1” and the ENISA guideline mentioned are related to ensuring secure authentication and authorization practices when mobile apps communicate with remote endpoints. ”MASVS-AUTH-1” emphasizes following best practices for secure usage of authentication and authorization protocols, which implicitly includes handling credentials securely and minimizing their exposure. The ENISA guideline specifically recommends releasing UI components that display credentials or keys immediately after use to reduce the risk of exposure. Both points address the safeguarding of authentication data from unnecessary exposure, which contributes to the overall security of the mobile application. Hence, there is a correlation between MASVS-AUTH-1 and the cited ENISA guideline in the context of secure handling of user authentication data in mobile apps.
- MASVS-AUTH-3: The ”MASVS-AUTH-3” requirement suggests the use of additional forms of authentication for sensitive actions, which implies a focus on increasing security measures within an app. The ENISA Guideline’s advice to release UI frames immediately after use is a security best practice that helps prevent sensitive information, such as credentials or keys, from remaining accessible in the UI after they are no longer needed. Both guidelines are concerned with enhancing the security and protection of authentication information within applications. By releasing UI components after use, one can reduce the risk of credentials or keys being compromised, which aligns with the secure implementation of additional authentication forms as mentioned in ”MASVS-AUTH-3.”
- MASVS-CRYPTO-1: Both ”MASVS-CRYPTO-1” and the ENISA guideline emphasize the importance of proper cryptographic practices to protect user data, especially in scenarios where physical device access is possible. ”MASVS-CRYPTO-1” describes general cryptography best practices, which include securing credentials and keys. The ENISA guideline specifically addresses the proper handling of credentials or keys within user interface components, advising that they should be released immediately after use. This is in line with the best practices mentioned in ”MASVS-CRYPTO-1” because promptly releasing UI components that contain sensitive information reduces the window of opportunity for attackers to gain access to this information through UI element inspection or memory dumps. Both guidelines aim to minimize the risk of sensitive cryptographic material being compromised.
- MASVS-CRYPTO-2: The correlation between ”MASVS-CRYPTO-2,” which deals with the management of cryptographic keys throughout their lifecycle, and the ENISA Guideline about releasing UI frames that display credentials or keys immediately after use is that both are concerned with the protection of cryptographic keys. MASVS-CRYPTO-2 emphasizes that strong cryptography can be undermined by poor key management, which includes how keys are generated, stored, and protected. The ENISA guideline complements this by recommending a specific practice to ensure that keys, when displayed in a UI, are

not exposed any longer than necessary, thereby reducing the window of opportunity for unauthorized access or leakage, which is part of protecting keys during their usage phase.

- MASVS-NETWORK-1: The "MASVS-NETWORK-1" description emphasizes the importance of data privacy and integrity during transit, which is largely achieved through secure connections such as those provided by TLS. Part of making sure that connections are secure involves proper handling of sensitive information such as credentials or keys. The ENISA Guideline about releasing UI frames immediately after use relates to reducing the window of opportunity for sensitive data exposure. While it focuses specifically on the UI components, the overarching theme is the safeguarding of sensitive information which aligns with the goals of "MASVS-NETWORK-1" in the context of network security and preventing potential data breaches. Thus, there is a correlation between ensuring secure network communication and the practice of protecting credentials or keys from unnecessary exposure even at the UI level.
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA guideline mentioned lies in the focus on securing UI components to prevent sensitive data leakage. The MASVS-PLATFORM-2 emphasizes the secure configuration of WebViews, which are UI components, to avoid exposure of sensitive data and functionality. Similarly, the ENISA guideline advises that UI components (like frames that may display credentials or keys) should be released immediately after use to prevent sensitive information from being compromised. Both pieces of guidance are concerned with mitigating the risk of sensitive data exposure through UI elements.
- MASVS-PLATFORM-3: The correlation exists because both the MASVS-PLATFORM-3 and the ENISA Guideline address the need to handle sensitive data carefully within the user interface to prevent unintended leakage or disclosure. MASVS-PLATFORM-3 focuses on ensuring sensitive data (like passwords, credit card details, etc.) doesn't leak due to platform mechanisms such as auto-generated screenshots or shoulder surfing. Meanwhile, the ENISA Guideline suggests releasing UI frames immediately after use when they contain credentials or keys. Both statements aim to protect sensitive information from being exposed to unauthorized parties, with a particular focus on how they are managed in the UI context.
- MASVS-RESILIENCE-4: The correlation exists in that both MASVS-RESILIENCE-4 and the ENISA guideline are addressing measures to enhance the security of an application against analysis and tampering. MASVS-RESILIENCE-4 focuses on making it difficult for attackers to perform dynamic analysis and dynamic instrumentation to understand or modify the app behavior at runtime. The ENISA guideline is a specific implementation of this principle, suggesting that if credentials or keys are displayed in the UI, they should be released quickly to minimize the window during which dynamic analysis techniques could capture or manipulate this sensitive information. Both points aim to protect an app's runtime environment and sensitive data it handles from being compromised.
- MASVS-STORAGE-1: The correlation between "MASVS-STORAGE-1" and the ENISA Guideline about releasing UI frames immediately after use lies in the concern for protecting sensitive data. Both MASVS-STORAGE-1 and the ENISA Guideline emphasize the importance of ensuring that sensitive data is handled securely—MASVS-STORAGE-1 by pointing out that any sensitive data intentionally stored by the app must be properly protected, irrespective of its storage location, and the ENISA Guideline by suggesting that any sensitive data displayed or manipulated in UI components (which could include credentials or keys) should have those UI frames released immediately after use in order to minimize the risk of exposure or leakage, potentially through memory dumps or other apps accessing the UI layer.

- MASVS-STORAGE-2: Both ”MASVS-STORAGE-2” and the ENISA Guideline emphasize the importance of handling sensitive data carefully to prevent unintentional storage or exposure. MASVS-STORAGE-2 is concerned with avoiding the accidental storage of sensitive data in publicly accessible locations, which can occur through the misuse of APIs or system features. The ENISA guideline’s recommendation to release UI frames immediately after use is intended to protect credentials or keys shown in the UI from being unintentionally retained or exposed. Both guidelines aim to mitigate risks associated with improper handling of sensitive information that could lead to leaks, which shows a correlation between the two.

4.6 Implementation Guidance (ENISA 3.6):

ENISA Secure Smartphone Development Guidance (3.6): Some devices and add-ons allow developers to use a secure hardware (e.g., TEE, SE) - the number of devices offering this functionality is likely to increase. Developers should make use of such capabilities to store keys, credentials and other sensitive data.

4.6.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: Both MASVS-AUTH-1 and the ENISA Guideline emphasize the importance of security in relation to user authentication and the management of sensitive data. MASVS-AUTH-1 concerns best practices for secure communication with remote endpoints, including handling user authentication and authorization, while the ENISA Guideline suggests utilizing secure hardware (like TEE or SE) for storing sensitive information like keys and credentials. Both are focused on enhancing the security posture of applications by protecting authentication mechanisms and sensitive data.
- MASVS-AUTH-2: Both ”MASVS-AUTH-2” and the ENISA guideline describe the importance of implementing secure authentication mechanisms. ”MASVS-AUTH-2” specifies the necessity for correct implementation of local authentication methods like biometrics or PIN codes, which may rely solely on local app authentication without a remote endpoint. The ENISA guideline advocates for the utilization of secure hardware capabilities such as TEE (Trusted Execution Environment) or SE (Secure Element) to safeguard keys, credentials, and other sensitive data. There is a correlation because both stress using available hardware-based security features to enhance the protection of authentication methods and sensitive information within mobile apps.
- MASVS-AUTH-3: Both ”MASVS-AUTH-3” and the ENISA Guideline emphasize on the need for additional security measures when dealing with sensitive actions or storing sensitive data within an app. MASVS-AUTH-3 mentions the use of additional forms of authentication for sensitive actions, and the ENISA Guideline advises on using secure hardware like TEE (Trusted Execution Environment) or SE (Secure Element) for storing keys, credentials, and other sensitive data. Both focus on enhancing security beyond basic measures, indicating a correlation between the two statements.
- MASVS-CRYPTO-1: Both ”MASVS-CRYPTO-1” and the ENISA Guideline emphasize the importance of securing user data, particularly on mobile devices where physical access to the device is a plausible threat. MASVS-CRYPTO-1 refers to following general cryptography best practices for data protection, while the ENISA Guideline suggests utilizing secure hardware like Trusted Execution Environments (TEE) or Secure Elements (SE) to store sensitive data like keys and credentials. The correlation here is that both guidelines advise leveraging strong security measures to protect sensitive information, with an understanding of the mobile context. The ENISA Guideline’s recommendation to use secure hardware can be seen as a specific instance of the broader best practices mentioned in MASVS-CRYPTO-1.
- MASVS-CRYPTO-2: The correlation between MASVS-CRYPTO-2 and the ENISA Guideline is that they both emphasize the importance of secure key management. MASVS-CRYPTO-2 specifies that proper key management throughout the cryptographic key lifecycle is crucial to prevent compromise, even with strong cryptography. This involves secure key generation, storage, and protection. The ENISA Guideline complements this by sug-

gesting developers utilize secure hardware like Trusted Execution Environments (TEE) or Secure Elements (SE) for storing keys, credentials, and other sensitive data, thereby following best practices for secure key management as recommended by MASVS-CRYPTO-2. As more devices provide such secure hardware options, the guideline encourages their use, which aligns with MASVS-CRYPTO-2's emphasis on the security of key management processes.

- MASVS-NETWORK-1: Both the Mobile Application Security Verification Standard (MASVS) NETWORK-1 control and the ENISA guideline emphasize the importance of securing sensitive data. MASVS-NETWORK-1 focuses on ensuring the privacy and integrity of data in transit, typically achieved through encryption and endpoint authentication, such as with TLS. The ENISA guideline encourages developers to leverage secure hardware features, like Trusted Execution Environments (TEE) or Secure Elements (SE), for storing sensitive data like keys and credentials. While MASVS-NETWORK-1 addresses secure network communication specifically, the use of secure hardware recommended by ENISA can complement this by providing a robust method for storing the cryptographic materials (keys and credentials) used to establish and maintain these secure communications. Thus, both are concerned with safeguarding sensitive information, albeit at different stages (in transit and at rest).
- MASVS-PLATFORM-3: The MASVS-PLATFORM-3 control is focused on preventing unintentional leakage of sensitive data through mechanisms such as auto-generated screenshots, which could occur due to platform features or user behavior (e.g., shoulder surfing, sharing the device). The ENISA guideline emphasizes the use of secure hardware, like Trusted Execution Environments (TEE) or Secure Elements (SE), to protect keys, credentials, and other sensitive data. Both the MASVS control and the ENISA guideline are concerned with the protection of sensitive data against different types of exposure or compromise. The MASVS control addresses the issue at the user interface level, while the ENISA guideline approaches it through the use of hardware-based security measures, but their underlying objective is aligned: enhancing the security of sensitive data on mobile platforms.
- MASVS-PRIVACY-3: The correlation between MASVS-PRIVACY-3 and the ENISA Guideline is centered on the protection and transparency of user data handling. MASVS-PRIVACY-3 emphasizes users' rights to clear information about how their data is used by apps, which includes data collection, storage, and sharing practices. The ENISA Guideline complements this by recommending the use of secure hardware such as TEE or SE to store sensitive data, which is an aspect of how data is stored with the intent to increase security. Adhering to secure storage practices as suggested by ENISA supports the MASVS-PRIVACY-3 requirement by ensuring that data is stored in a secure manner, thereby aligning with the app's commitment to responsibly handle user data, which should be clearly communicated to the users.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the ENISA Guideline about utilizing secure hardware (e.g., TEE, SE) for storing sensitive data is that both are concerned with the security and integrity of the platform on which the mobile app is running. MASVS-RESILIENCE-1 highlights the importance of ensuring that the operating system has not been compromised to trust its security features, like secure storage, which could be provided by secure hardware elements like TEE or SE as indicated by the ENISA Guideline. Both suggest that leveraging the platform's secure elements is essential for protecting sensitive data.
- MASVS-RESILIENCE-2: The correlation between "MASVS-RESILIENCE-2" and the described ENISA Guideline is that both are focused on enhancing the security and integrity

of mobile applications. MASVS-RESILIENCE-2 is about protecting the app from modifications and maintaining the integrity of its original functionality, which could include preventing cheating or unauthorized access to premium features. The ENISA guideline advises developers to utilize secure hardware features like Trusted Execution Environments (TEE) or Secure Elements (SE) to store sensitive information securely. Using these hardware features would support the goal of MASVS-RESILIENCE-2 by making it more difficult for an attacker to tamper with or reverse engineer the app, as the critical components and data would be protected by the hardware's security capabilities. Hence, both are concerned with preserving the security of the app on user-controlled devices.

- MASVS-RESILIENCE-3: The correlation between "MASVS-RESILIENCE-3" and the ENISA guideline about using secure hardware (e.g., TEE, SE) for storing keys, credentials, and other sensitive data is centered around the concept of increasing the resilience and security of the application. MASVS-RESILIENCE-3 emphasizes impeding the understanding of an app through static analysis to prevent tampering, while the use of secure hardware as suggested by ENISA is a method to protect sensitive data from being compromised. Both guidelines aim to safeguard against malicious activities by making it harder for attackers to access or manipulate critical components or data within the app. Using secure hardware can be part of the strategy to obscure how an application works and protects its sensitive data, aligning with the objective of MASVS-RESILIENCE-3 to hinder static analysis.
- MASVS-RESILIENCE-4: The correlation between "MASVS-RESILIENCE-4" and the ENISA Guideline is that both emphasize enhancing the security of mobile applications against dynamic analysis and manipulation. MASVS-RESILIENCE-4 focuses on making it difficult to perform dynamic analysis and prevent runtime code modification, which aligns with the ENISA Guideline's recommendation to use secure hardware like TEE or SE for protecting sensitive data. By utilizing secure hardware, developers can make dynamic analysis and instrumentation more challenging for attackers, thus increasing the resilience of the app in a way that is consistent with MASVS-RESILIENCE-4's objective.
- MASVS-STORAGE-1: The MASVS-STORAGE-1 requirement and the ENISA guideline both emphasize the importance of secure storage for sensitive data. MASVS-STORAGE-1 addresses the need for properly protecting sensitive data regardless of where it is stored, while the ENISA guideline specifically points to utilizing secure hardware like TEE (Trusted Execution Environment) or SE (Secure Element) when available on devices for storing keys, credentials, and other sensitive information. Both statements advocate for heightened security measures to safeguard sensitive data handled by apps, indicating a correlation between the two directives in the context of data protection.

4.7 Implementation Guidance (ENISA 3.7):

ENISA Secure Smartphone Development Guidance (3.7): Provide the ability to the mobile user to change passwords or other authentication tokens.

4.7.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: Both ”MASVS-AUTH-1” and the ENISA guideline ”Provide the ability to the mobile user to change passwords or other authentication tokens” focus on the aspect of user authentication and the security measures around it. ”MASVS-AUTH-1” suggests that applications should adhere to best practices for secure communication and authentication protocols, while the ENISA guideline emphasizes the need for users to have the ability to change passwords or authentication tokens, which is a subset of implementing secure authentication mechanisms. Enabling users to change their authentication credentials is indeed a security best practice, as it allows users to respond to potential security breaches by altering their tokens or passwords, thus maintaining the integrity of the authentication process. Overall, both guidelines aim to enhance the security of the user authentication process within mobile applications.
- MASVS-AUTH-3: Both ”MASVS-AUTH-3” and the ENISA guideline emphasize the importance of robust authentication mechanisms within mobile applications. ”MASVS-AUTH-3” suggests implementing additional forms of authentication for sensitive actions, which could include the ability to change passwords or other authentication tokens as described in the ENISA guideline. This ability enhances security by allowing users to manage their credentials, which is complementary to the intent of providing multi-faceted authentication methods.
- MASVS-CRYPTO-2: The ENISA guideline ”Provide the ability to the mobile user to change passwords or other authentication tokens” aligns with ”MASVS-CRYPTO-2” because both relate to the secure management of authentication credentials. ”MASVS-CRYPTO-2” focuses on the lifecycle of cryptographic keys (which can be considered as a form of authentication token), including their safe generation, storage, and protection. Allowing a mobile user to change passwords or authentication tokens is an important aspect of key management as it ensures that users can take action to maintain the security of their credentials. Thus, while ”MASVS-CRYPTO-2” is more broadly about cryptographic key management, the ability to change passwords or authentication tokens is a subset of this process, and fitting password and token management would indeed be part of proper key management practices.
- MASVS-PLATFORM-3: The correlation between ”MASVS-PLATFORM-3” and the ENISA Guideline regarding the ability for a mobile user to change passwords or other authentication tokens is that both are concerned with the protection of sensitive data. MASVS-PLATFORM-3 is about implementing measures to prevent sensitive data from being leaked through platform mechanisms like auto-generated screenshots or device sharing, which could include passwords and authentication tokens. Similarly, the ENISA Guideline on providing the ability to change passwords or other authentication tokens underlines the importance of giving users control over their authentication data to maintain its security. Both guidelines underscore the significance of securing sensitive information within the app UI and in the context of user interactions.

- MASVS-PRIVACY-4: Both ”MASVS-PRIVACY-4” and the ENISA guideline you referenced emphasize giving control to the user over their personal data and authentication credentials respectively. ”MASVS-PRIVACY-4” encompasses a broad range of control by allowing users to manage their data and privacy settings, while the ability to change passwords or authentication tokens mentioned in the ENISA guideline falls under the umbrella of managing security-related user data. Both guidelines aim to put users in charge of their information and its security, showing a correlation in their purposes of enhancing user control and privacy.

4.8 Implementation Guidance (ENISA 3.8):

ENISA Secure Smartphone Development Guidance (3.8): Ensure passwords and keys are not visible in cache or logs.

4.8.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA Guideline regarding passwords and keys not being visible in cache or logs is that both are focused on the principle of protecting sensitive authentication-related information within a mobile application context. The MASVS-AUTH-1 requirement states the need for authentication and authorization best practices to ensure secure protocol use. This implicitly includes the safe handling of authentication credentials (such as passwords) and authorization tokens or keys. If passwords and keys were visible in cache or logs, it would indicate a failure to follow best practices in securing these sensitive data, thus violating the MASVS-AUTH-1 principle. The ENISA Guideline directly states that passwords and keys should not be visible in cache or logs, highlighting a specific aspect of the best practices implied by MASVS-AUTH-1. Therefore, adherence to the ENISA Guideline would contribute to meeting the MASVS-AUTH-1 requirement, as both aim to protect the integrity and confidentiality of authentication and authorization mechanisms.
- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3” which discusses requiring additional forms of authentication for sensitive actions and the ENISA guideline suggesting that passwords and keys are not visible in cache or logs is that both are related to strengthening authentication measures and safeguarding sensitive authentication credentials. ”MASVS-AUTH-3” is focused on the implementation of secure multi-factor authentication methods, while the ENISA guideline is concerned with protecting credentials (such as passwords and keys) from being exposed in areas like cache or logs, where they may be vulnerable to unauthorized access. Both guidelines aim to enhance the security of authentication processes in a mobile app context.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the ENISA guideline about ensuring passwords and keys are not visible in cache or logs relates to the best practices in cryptography to secure user data. ”MASVS-CRYPTO-1” emphasizes the importance of cryptography in protecting user data, especially on mobile devices that can be physically accessed by attackers. To adhere to this control, it’s essential to use cryptographic best practices, one of which is preventing sensitive information like passwords and keys from being stored in easily accessible places like cache or logs, aligning with the ENISA’s guideline for maintaining the confidentiality and integrity of cryptographic material.
- MASVS-CRYPTO-2: Both ”MASVS-CRYPTO-2” and the ENISA Guideline relate to the safe management and protection of cryptographic keys and sensitive authentication credentials. ”MASVS-CRYPTO-2” addresses comprehensive key management practices including generation, storage, and protection of keys. Similarly, the ENISA Guideline specifically mentions ensuring that passwords and keys are not exposed in vulnerable locations such as cache or logs, which could be considered as aspects of storage and protection in key management. Both are focused on mitigating the risk of sensitive cryptographic material being compromised.
- MASVS-NETWORK-1: There is a correlation between ”MASVS-NETWORK-1” and the ENISA guideline ”Ensure passwords and keys are not visible in cache or logs.” Reasoning:

The essence of ”MASVS-NETWORK-1” focuses on maintaining the confidentiality and integrity of data in transit which involves setting up secure connections, typically achieved through encryption and endpoint authentication. The mention of avoiding the bypassing of secure defaults or misuse of APIs implies attention to potential vulnerabilities that could expose sensitive data. Similarly, the ENISA guideline about not allowing passwords and keys to be visible in cache or logs is about ensuring the confidentiality and integrity of these sensitive pieces of information. Both guidelines are concerned with the prevention of unauthorized access to sensitive data - in one case ’in transit’, and in the other, when ’at rest’. While ”MASVS-NETWORK-1” does not directly mention prevention of exposure in cache or logs, observing this guideline inherently supports the MASVS requirement by minimizing the risk of sensitive data (which could include credentials and keys used for establishing secure connections) being compromised.

- MASVS-PLATFORM-3: Both ”MASVS-PLATFORM-3” and the ENISA guideline describe the need to protect sensitive data from accidental or unintended exposure. MASVS-PLATFORM-3 focuses on preventing sensitive data from being leaked through UI mechanisms such as screenshots or shoulder surfing, while the ENISA guideline specifically emphasizes that passwords and keys should not be visible in cache or logs. Both are addressing the broader principle of securing sensitive data from being exposed to unauthorized parties, albeit through different vectors that may lead to potential data leaks or breaches.
- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1,” which emphasizes the importance of running on a secure and uncompromised platform to maintain the integrity of security features like secure storage and sandboxing, and the ENISA Guideline stating that ”Ensure passwords and keys are not visible in cache or logs,” is that both controls are concerned with protecting sensitive data within the context of a trusted environment. If the platform is tampered with, it may undermine the security mechanisms designed to shield passwords and keys from being exposed in places like cache or logs. Thus, MASVS-RESILIENCE-1’s focus on platform integrity directly relates to the ENISA guideline’s aim of keeping credentials secure by ensuring they do not become visible in areas that could be compromised on an untrusted platform.
- MASVS-RESILIENCE-3: The correlation between MASVS-RESILIENCE-3 and the ENISA guideline regarding the protection of passwords and keys from being visible in cache or logs is that they both aim to enhance the security and resilience of an application by protecting sensitive information. MASVS-RESILIENCE-3 focuses on impeding the understanding of an app’s internals to prevent tampering, which could include methods to obscure or protect passwords and keys from being discovered through static analysis. Similarly, the ENISA guideline explicitly addresses the need to ensure sensitive data like passwords and keys are not exposed in places like cache or logs where an attacker could easily access them using static or dynamic analysis methods. Both controls are concerned with reducing the attack surface and making it more difficult for malicious actors to compromise the application’s security measures.
- MASVS-RESILIENCE-4: While MASVS-RESILIENCE-4 does not directly address handling passwords and keys, it does emphasize making dynamic analysis difficult. Dynamic analysis can include observing an app’s behavior which potentially could expose sensitive information such as passwords and keys if they are not properly protected. If an app implements measures to hinder dynamic analysis, it would be more challenging for an attacker to observe passwords and keys in cache or logs during runtime, which aligns with the ENISA guideline to ensure such sensitive data is not exposed.
- MASVS-STORAGE-1: The MASVS-STORAGE-1 requirement relates to the handling and storage of sensitive data, ensuring that data is properly protected no matter where

it is stored. The ENISA Guideline about ensuring passwords and keys are not visible in cache or logs is a specific example of measures to protect sensitive data. Both relate to the principle of safeguarding sensitive information from unauthorized access, including where data might be unintentionally exposed such as in logs or caches. The MASVS-STORAGE-1 is broader but includes protection mechanisms relevant to the ENISA Guideline.

- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA Guideline ”Ensure passwords and keys are not visible in cache or logs” is that both are concerned with the prevention of unintentional exposure of sensitive information. MASVS-STORAGE-2 addresses the risk of sensitive data being accidentally stored or exposed in publicly accessible locations, possibly due to the misuse of APIs or system capabilities, while the ENISA Guideline specifically mentions that passwords and keys should not be visible in cache or logs. Both point towards the goal of ensuring sensitive data is adequately protected and not exposed through mismanagement or oversight by developers.

4.9 Implementation Guidance (ENISA 3.9):

ENISA Secure Smartphone Development Guidance (3.9): Do not store any passwords or secrets in the application binary. Do not use a generic shared secret for integration with the backend server (like password embedded in code). Mobile application binaries can be easily downloaded and reverse engineered.

4.9.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the cited ENISA Guideline is evident. MASVS-AUTH-1 highlights the importance of secure protocols and practices regarding user authentication and authorization. It implies that the application should not take shortcuts or use insecure methods to handle user credentials and permissions, which aligns with the ENISA Guideline’s emphasis on not storing passwords or secrets in the application binary or using a generic shared secret. Both are concerned with avoiding practices that could compromise security, especially in the context of reverse engineering or exposing sensitive information.
- MASVS-AUTH-3: The MASVS-AUTH-3 requirement about implementing additional forms of authentication securely correlates with the ENISA guideline on not storing passwords or secrets in the application binary or using a generic shared secret for backend integration. Both highlight the importance of secure authentication mechanisms and discourage practices that would allow attackers to retrieve sensitive information through reverse engineering of the application binary. MASVS-AUTH-3’s suggestion to use secure additional authentication aligns with ENISA’s directive to avoid embedding secrets in the code, as both are aimed at enhancing the security of the authentication process.
- MASVS-CODE-3: The correlation between ”MASVS-CODE-3” and the ENISA Guideline described is that both are concerned with security issues that can arise from inadequate protection measures in application components. MASVS-CODE-3 mentions the importance of performing a full whitebox assessment to ensure security, including checking for known vulnerabilities in third-party components, which could be considered as ”low-hanging fruit.” This aligns with the ENISA Guideline, which warns against storing passwords or secrets in the application binary and using generic shared secrets. Both reflect the understanding that leaving sensitive data or weak security practices in the app’s code can lead to serious security vulnerabilities that can be exploited if the application is reverse-engineered.
- MASVS-CODE-4: The Mobile Application Security Verification Standard (MASVS) code ”MASVS-CODE-4” refers to the need for treating all incoming data as untrusted input which must be verified and sanitized, while the ENISA guideline advises against storing passwords or secrets in the application binary and using generic shared secrets for backend integration. Both of these points are underpinned by a common principle which is to protect the application from security breaches that can originate from untrusted or manipulated inputs, whether it be through the UI, IPC, network, file system, or even the code itself when it contains hard-coded secrets that can be exploited. In both cases, the concern is that improperly handled or secured input/data can compromise the security of the application and potentially allow unauthorized access to sensitive information or systems.
- MASVS-CRYPTO-1: The description of ”MASVS-CRYPTO-1” emphasizes the importance of cryptography in protecting user data, especially given the risk of attackers getting physical access to a mobile device. The ENISA guideline advises against storing passwords

or secrets within the application binary to prevent easy access through reverse engineering. Both these statements are concerned with applying strong cryptography practices to safeguard sensitive information in the mobile environment, thus showing a correlation. Both highlight the need to avoid insecure storage and handling of sensitive data, including secrets and passwords, aligning with the goal of preventing exposure through code reverse engineering and achieving a robust cryptographic posture.

- MASVS-CRYPTO-2: The correlation between ”MASVS-CRYPTO-2,” which discusses the importance of proper management of cryptographic keys throughout their lifecycle, and the ENISA Guideline advice against storing passwords or secrets in the application binary is evident. Both emphasize the risks associated with poor handling of sensitive information and stress the need for protecting such data from exposure through secure key management, storage practices, and avoiding hard-coded secrets that can be compromised through reverse engineering.
- MASVS-NETWORK-1: Both ”MASVS-NETWORK-1” and the ENISA guideline emphasize the importance of protecting the confidentiality and integrity of data. ”MASVS-NETWORK-1” deals with the security of data in transit specifically, ensuring secure network connections and protection against man-in-the-middle attacks typically through TLS encryption and endpoint authentication. The ENISA guideline complements this by addressing the security of sensitive information at rest within the app’s codebase. It warns against storing secrets or passwords within app binaries and using generic shared secrets, as this would pose a risk if the app is reverse engineered. Both controls are related to preventing unauthorized access to sensitive data, one while the data is in transit and the other while it is at rest in the application’s compiled code.
- MASVS-PLATFORM-2: Both the ”MASVS-PLATFORM-2” and the ENISA guideline emphasize the importance of secure configuration to prevent sensitive data leakage. MASVS-PLATFORM-2 addresses secure configuration in the context of WebViews and their use in mobile applications, highlighting the need to prevent exposure of sensitive functionality and data, which can include passwords and secrets. The ENISA guideline explicitly mentions not to store passwords or secrets in the application binary as they can be easily extracted through reverse engineering. Both statements concern the protection of sensitive information within mobile applications.
- MASVS-PLATFORM-3: The correlation exists because both ”MASVS-PLATFORM-3” and the ENISA guideline address the concern of protecting sensitive data within a mobile application. ”MASVS-PLATFORM-3” focuses on the safe handling of sensitive data displayed in the UI, aiming to prevent accidental leaks through platform mechanisms or unintended user actions. On the other hand, the ENISA guideline emphasizes that sensitive information, particularly passwords or secrets, should not be stored in the application binary or be hardcoded, as this can lead to security vulnerabilities due to the possibility of reverse engineering. Both statements underline the importance of proper management of sensitive data to mitigate the risk of unauthorized access and exposure.
- MASVS-PRIVACY-2: Although the description of ”MASVS-PRIVACY-2” primarily focuses on the protection of user identity through techniques like anonymization and pseudonymization, and the separation of data streams to prevent repurposing for other uses, it is inherently related to the ENISA Guideline’s emphasis on not storing passwords or secrets in the application binary. Both are concerned with safeguarding user privacy and security. The MASVS-PRIVACY-2’s approach to prevent user identification and tracking complements the ENISA Guideline’s directive to avoid embedding secrets in code, as both practices aim to minimize security risks and protect against unauthorized access or user tracking.

- MASVS-PRIVACY-4: Although ”MASVS-PRIVACY-4” and the ENISA Guideline do not directly reference each other, there is an indirect correlation. ”MASVS-PRIVACY-4” emphasizes that users should have control over their data, which includes the ability to manage privacy settings and consent, while the ENISA guideline advises against storing passwords or secrets in the application binary to prevent unauthorized access and disclosure of user data. Both guidelines aim to protect user data and privacy, ensuring that user information remains secure and under the user's control. The common goal of enhancing data privacy and security creates a correlation between the two principles.
- MASVS-RESILIENCE-1: The MASVS-RESILIENCE-1 description highlights the importance of not running an app on a tampered platform, as this may disable security features and put app data at risk. The ENISA guideline advises against storing any passwords or secrets in the application binary and avoiding generic shared secrets for backend integration. Both statements correlate in that they emphasize the need to protect sensitive information and maintain the integrity of security measures. Running on a compromised OS could lead to bypassing protections and extracting embedded secrets from the binary, which is the exact risk that the ENISA guideline warns against.
- MASVS-RESILIENCE-2: The correlation is that both statements emphasize the importance of protecting the app from being manipulated or reverse-engineered. The MASVS-RESILIENCE-2 mentions the need for protections to prevent modifications to the app's code and resources, which aligns with the ENISA guideline that advises against storing passwords or secrets in the app binary, as well as avoiding the use of generic shared secrets that can be extracted through reverse engineering. Both are concerned with ensuring the security and integrity of the app on the user device and protecting the app's communication with the backend server.
- MASVS-RESILIENCE-3: Both ”MASVS-RESILIENCE-3” and the ENISA guideline emphasize the importance of protecting the app from being understood or reverse-engineered to prevent tampering, code comprehension, and exposure of sensitive information like passwords or secrets. MASVS-RESILIENCE-3 aims to impede the comprehension of an app's internals, and the ENISA guideline specifically advises against storing sensitive information within the app binary, as this would make it vulnerable to reverse engineering, which correlates directly to understanding the app's internals as per MASVS-RESILIENCE-3.
- MASVS-RESILIENCE-4: Both MASVS-RESILIENCE-4 and the ENISA guideline emphasize the risks associated with information that can be obtained through reverse engineering or dynamic analysis of mobile applications. MASVS-RESILIENCE-4 is about making dynamic analysis difficult to protect against runtime code modification and dynamic instrumentation, while the ENISA guideline advises against storing passwords or secrets in application binaries, acknowledging that app binaries are susceptible to being reverse-engineered. The correlation is that both are aiming to prevent attackers from gaining sensitive information or modifying the application's behavior by inspecting and manipulating the app's code.
- MASVS-STORAGE-1: The correlation is apparent as both ”MASVS-STORAGE-1” and the ENISA guideline address the issue of securely handling sensitive data within mobile applications. MASVS-STORAGE-1 emphasizes the proper protection of sensitive data regardless of storage location, while the ENISA guideline specifically advises against storing passwords or secrets in the application binary and using shared secrets for backend integration. This recommendation is based on the fact that mobile application binaries can be downloaded and reverse-engineered, leading to the exposure of such sensitive information if it is not properly secured. Both guidelines highlight the importance of implementing robust security measures to protect sensitive data handled by mobile apps.

- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the described ENISA Guideline is that both are concerned with the security of sensitive data within mobile applications. ”MASVS-STORAGE-2” addresses the issue of sensitive data being unintentionally stored or exposed in locations that are publicly accessible, a scenario that could result from developer error or misuse of APIs and system features. Similarly, the ENISA Guideline warns against the storage of passwords or secrets in the application binary and the use of a generic shared secret, as these practices can lead to sensitive data being compromised. The underlying principle of both guidelines is to avoid the exposure of sensitive information that could be leveraged by malicious actors through techniques such as reverse engineering.

Chapter 5
Ensure sensitive data is protected in transit

Network-based attacks are one of the major threats to smartphone applications, especially since the majority of smartphones contain multiple different networking technologies. Today most smartphones contain at least WiFi and cellular networking technologies such as GPRS, UMTS, CDMA, LTE (and possible others). In addition, Bluetooth and other short distance radio interfaces such as Near Field Communication (NFC) are commonly integrated in modern smartphones. Sensitive data passing through this shared channels can be intercepted and modified"

5.1 Implementation Guidance (ENISA 4.1):

ENISA Secure Smartphone Development Guidance (4.1): Assume that the network layer is not secure. Specifically, WiFi networks must be considered not trustworthy. Modern network layer attacks can defeat network layer encryption.

5.1.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA Guideline about assuming that the network layer is not secure is clear. MASVS-AUTH-1 emphasizes that apps must adhere to best practices for secure communication protocols, which implicitly includes the assumption that the network may be insecure or untrustworthy. The ENISA Guideline explicitly states that network layers, especially WiFi, should not be considered secure due to potential modern network attacks that can defeat encryption. Both statements underscore the importance of ensuring that security measures are in place at the application level, regardless of the security of the underlying network.
- MASVS-CODE-4: The correlation between ”MASVS-CODE-4” and the ENISA guideline is that both emphasize the importance of treating incoming data as untrusted and potentially compromised. MASVS-CODE-4 concerns the need to verify and sanitize all incoming data to prevent security breaches like SQL injection, XSS, or insecure deserialization. Similarly, the ENISA guideline assumes that the network layer, including WiFi networks, is not secure, which implies that data received over the network could be intercepted or altered by attackers. Therefore, both statements advocate for a security approach that does not rely on the integrity of the incoming data and suggest defensive measures to ensure data is appropriately handled to maintain application security.
- MASVS-CRYPTO-1: The Mobile Application Security Verification Standard (MASVS) ”CRYPTO-1” and the ENISA guideline both emphasize the importance of implementing strong cryptographic measures due to the assumption that an attacker may have access to user data through physical means or network layer attacks. While MASVS ”CRYPTO-1” outlines the importance of cryptography to protect data on the physical device, the ENISA guideline extends this concern to network communications, underlining the need to distrust even WiFi networks and ensure encryption beyond the network layer. Both advices are based on the premise that the environment where the application operates cannot be fully trusted, hence highlighting the need for robust cryptography practices.
- MASVS-CRYPTO-2: ”MASVS-CRYPTO-2” concerning the management of cryptographic keys throughout their lifecycle and the ENISA Guideline on considering WiFi networks as not trustworthy both underline the importance of strict security measures beyond just relying on the strength of encryption. They imply that without proper key management and a distrustful approach towards network layer security, even strong cryptography can be vulnerable to attacks.
- MASVS-NETWORK-1: Both the MASVS-NETWORK-1 description and the ENISA Guideline emphasize the importance of data privacy and integrity for transmitted data, especially in potentially insecure network environments. They both acknowledge that although encryption and other security measures are necessary, they can be circumvented or improperly implemented, thus cautioning developers to not rely solely on the network layer for security and to assume it is not secure by default. Both advocate for robust security measures at the application layer to protect data in transit.

- MASVS-NETWORK-2: The correlation between ”MASVS-NETWORK-2” and the mentioned ENISA guideline exists in the context of IoT security. Reasoning: ”MASVS-NETWORK-2” focuses on establishing trust in the network layer by limiting trust to specific Certificate Authorities (CAs) through certificate pinning or public key pinning. This practice enhances the security of network communication by ensuring that the app does not blindly trust all pre-installed root CAs but instead verifies the server's certificate against a known set of certificates or public keys. The ENISA guideline suggests assuming that the network layer is inherently insecure, specifically highlighting the risks associated with WiFi networks. By considering networks untrustworthy, it implies that additional security measures should be implemented to mitigate potential attacks that could compromise network communication, such as Man-in-the-Middle (MITM) attacks that can defeat network layer encryption. Both ”MASVS-NETWORK-2” and the ENISA guideline aim to protect sensitive data during transmission over potentially insecure networks. Certificate pinning aligns with the ENISA guideline by directly addressing the issue of not relying on the security of the network layer and providing a mechanism to trust the secure connection to the intended server, despite potential network vulnerabilities.
- MASVS-PLATFORM-2: Both ”MASVS-PLATFORM-2” and the ENISA Guideline focus on the importance of ensuring security in situations where there could be potential exposure to untrusted environments. ”MASVS-PLATFORM-2” addresses securing WebViews to prevent data leakage and exposure to sensitive functionality, implying awareness of the surrounding risks and mitigating them, akin to the ENISA Guideline, which assumes the network layer, specifically public WiFi, is not secure and advises to design the app with the notion that underlying networks can be compromised. Both reflect an underlying principle of not placing trust in external systems and ensuring robust security regardless of the assumed security of the network layer.
- MASVS-PRIVACY-2: Both ”MASVS-PRIVACY-2” and the ENISA guideline focus on aspects of protecting user privacy and identity. ”MASVS-PRIVACY-2” emphasizes on the use of techniques to prevent user identification and tracking, which also correlates with the ENISA guideline's caution about the insecurity of network layers, particularly WiFi networks, against modern network layer attacks. They both underline the importance of implementing security measures beyond basic network layer encryption to safeguard user privacy.
- MASVS-STORAGE-2: While MASVS-STORAGE-2 discusses avoiding unintentional storage or exposure of sensitive data, and the ENISA guideline advises to assume the network layer is not secure, both point towards a concern for data security either at rest (MASVS-STORAGE-2) or in transit (ENISA guideline). The correlation lies in the emphasis on protecting sensitive data from unintended exposure, whether it be through leakage into logs or backups, or through interception over insecure networks like WiFi. In both cases, the developer is urged to take proactive steps to secure sensitive data against different types of vulnerabilities.

5.2 Implementation Guidance (ENISA 4.2):

ENISA Secure Smartphone Development Guidance (4.2): Applications should enforce the use of an end-to-end secure channel (such as TLS) when sending sensitive information over any network (e.g., using Strict Transport Security - STS). This includes passing user credentials and other authentication equivalents.

5.2.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA Guideline is that both emphasize the importance of secure communication for sensitive information, specifically when authentication and authorization are involved. ”MASVS-AUTH-1” mentions that apps should follow best practices for secure use of protocols when connecting to a remote endpoint with authentication and authorization requirements. Similarly, the ENISA Guideline advises on the use of end-to-end secure channels (like TLS) for transmitting sensitive information, including user credentials and authentication data. The use of TLS, as suggested by the ENISA Guideline, would be part of the best practices mentioned in ”MASVS-AUTH-1” for ensuring secure communications.
- MASVS-CRYPTO-1: The description for ”MASVS-CRYPTO-1” emphasizes the importance of utilizing good cryptographic practices to protect user data, especially given the likelihood of physical device access by attackers. The ENISA Guideline advises the enforcement of secure channels like TLS for transmitting sensitive information over networks. Both relate to safeguarding data by dictating standards and methodologies for secure communication to prevent unauthorized access to sensitive information. Hence, there is a correlation as both guidelines are oriented towards ensuring the security of data transmission and storage through robust cryptographic measures.
- MASVS-NETWORK-1: Both the MASVS-NETWORK-1 requirement and the ENISA guideline emphasize the importance of ensuring data privacy and integrity during data transit in mobile applications. They both recommend the use of secure channels, such as TLS, to protect sensitive information being communicated over a network, highlighting the necessity for encryption and endpoint authentication to prevent data breaches and unauthorized access.
- MASVS-NETWORK-2: The correlation between ”MASVS-NETWORK-2” and the described practice from the ENISA Guideline is that both controls are concerned with ensuring the integrity and security of network communication in mobile applications. ”MASVS-NETWORK-2” specifically refers to certificate pinning or public key pinning, which is a technique used to ensure that a mobile application communicates only with the intended server by validating the server’s certificate against a known set of certificates or public keys embedded in the application. This effectively limits the set of trusted certificates and prevents man-in-the-middle attacks that may occur if the device’s default root certificates are compromised or too broad. The ENISA Guideline advocates for the enforcement of an end-to-end secure channel, such as TLS, for the transmission of sensitive data. While it does not explicitly mention certificate or public key pinning, the use of TLS (with STS ensuring it can’t be bypassed) is part of achieving a secure communication channel. Certificate pinning is an additional, more granular layer of security that can be applied on top of TLS to further ensure that the secure channel is not compromised by untrusted or rogue certificates. Both controls are complementary: TLS provides encryption

and integrity of the data in transit, while certificate pinning ensures trustworthiness of the endpoint being communicated with.

- MASVS-PLATFORM-1: The ENISA guideline you've cited focuses on securing sensitive information during transmission over a network by enforcing end-to-end secure channels like TLS, including techniques like STS. While this guideline is directly related to network communication, the MASVS-PLATFORM-1 is somewhat broader in scope, dealing with secure Inter-Process Communication (IPC) mechanisms. The correlation lies in the fundamental objective of both to ensure secure data exchange. MASVS-PLATFORM-1 addresses the security of IPC mechanisms, which may involve network communication when different apps or components on the same device exchange data. When these IPC mechanisms involve sending sensitive information over a network as part of their operation, the ENISA guideline's recommendation of using secure transmission channels like TLS would be applicable and necessary to achieve the level of security expected by MASVS-PLATFORM-1 standard. In summary, while each guideline targets security at different levels of an application's operation, they both aim to protect sensitive data from unauthorized access or exposure, aligning their intentions regarding information security.
- MASVS-PLATFORM-2: The MASVS-PLATFORM-2 description emphasizes the secure configuration of WebViews to prevent sensitive data leakage and exposure of sensitive functionality, which includes the proper handling of data in transit within a WebView. The ENISA Guideline stresses the importance of using an end-to-end secure channel like TLS to send sensitive information over networks, which would be applicable to WebViews as they may load content over the network. Both guidelines are concerned with the protection of sensitive data, making them correlated.

5.3 Implementation Guidance (ENISA 4.3):

ENISA Secure Smartphone Development Guidance (4.3): For sensitive data, to reduce the risk of man-in-middle attacks (like SSL proxy, SSL strip), a secure connection should only be established after verifying the identity of the remote-end-point (server). This can be achieved by ensuring that TLS is only established with end-points having the trusted certificates in the key chain.

5.3.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA Guideline is that both address the importance of secure authentication and authorization practices when apps communicate with remote endpoints. ”MASVS-AUTH-1” emphasizes that apps must follow best practices for secure protocol use, which aligns with the ENISA Guideline’s recommendation to verify the identity of the remote endpoint via trusted certificates to establish a secure TLS connection, thus reducing the risk of man-in-the-middle attacks. Both are concerned with ensuring the security of data in transit and preventing unauthorized access.
- MASVS-AUTH-3: Both MASVS-AUTH-3 and the ENISA guideline stress the importance of secure authentication methods to protect sensitive actions or data. MASVS-AUTH-3 suggests using additional authentication methods, like biometric or MFA, for sensitive app actions, which aligns with the ENISA guideline’s recommendation to verify the server’s identity before establishing a secure TLS connection for sensitive data, thus preventing man-in-the-middle attacks. Both guidelines prioritize security measures to authenticate and protect critical transactions and data exchanges.
- MASVS-CRYPTO-1: There is a correlation between ”MASVS-CRYPTO-1” and its description regarding the importance of cryptography in securing user data—especially in scenarios where physical access to a mobile device is plausible—and the ENISA Guideline on reducing the risk of man-in-the-middle attacks. Both emphasize the significance of robust cryptographic measures to protect sensitive information. MASVS-CRYPTO-1 underscores the role of cryptography in general security best practices, while the ENISA Guideline specifically addresses the establishment of secure connections using TLS with trusted certificates to verify the identity of the server as a defense against man-in-the-middle attacks. The essence of both statements converges on the principle that employing strong, trusted cryptographic protocols and practices is essential to safeguard data.
- MASVS-CRYPTO-2: The correlation exists in the context of ensuring the security of sensitive data. MASVS-CRYPTO-2 emphasizes the importance of managing cryptographic keys throughout their lifecycle to prevent compromise of even the strongest cryptography due to poor key management. The ENISA Guideline stresses the importance of establishing secure connections by verifying the identity of the remote endpoint, which involves the use of trusted certificates. Key management, as described by MASVS-CRYPTO-2, is integral to the trustworthiness of the certificates used in TLS connections, which aligns with the guidance provided by ENISA to avoid man-in-the-middle attacks. Both guidelines underscore the critical role that proper handling and verification of cryptographic elements play in maintaining the confidentiality and integrity of communications.
- MASVS-NETWORK-1: The description of ”MASVS-NETWORK-1” emphasizes the importance of ensuring data privacy and integrity for data in transit by encrypting data and

authenticating the remote endpoint, similar to the TLS protocol's functions. The ENISA Guideline similarly stresses the importance of verifying the identity of the remote endpoint using trusted certificates before establishing a secure connection to mitigate the risk of man-in-the-middle attacks. Both sources advocate for the use of secure connections and authentication to protect against interception and tampering, thus showing a direct correlation.

- MASVS-NETWORK-2: The correlation between "MASVS-NETWORK-2" and the described ENISA Guideline is clear. "MASVS-NETWORK-2" refers to the practice of certificate or public key pinning, which involves trusting only specific Certificate Authorities (CAs) rather than all default root CAs provided by the framework or device. This aligns with the ENISA Guideline's recommendation to verify the identity of the remote-endpoint for secure connections and to establish TLS only with end-points that have trusted certificates. Both aim to reduce the risk of man-in-the-middle attacks by ensuring that the communication is only established with trusted parties, verified through their certificates.
- MASVS-PLATFORM-1: The correlation exists in that both "MASVS-PLATFORM-1" and the ENISA Guideline are concerned with secure communication between different entities in a mobile environment. "MASVS-PLATFORM-1" touches on secure interactions involving IPC (Inter-Process Communication) mechanisms, indicating that there should be safe exposure of data and functionality between apps or between the user and the app. Similarly, the ENISA Guideline emphasizes the need for trusted communication by verifying the server's identity to mitigate man-in-the-middle attacks when handling sensitive data. Both consider the integrity and confidentiality of data being transferred in scenarios where interception by an unauthorized party is possible.
- MASVS-PLATFORM-2: Both "MASVS-PLATFORM-2" and the ENISA guideline emphasize the importance of secure configurations to protect sensitive data. While the MASVS-PLATFORM-2 focuses on securely configuring WebViews to prevent data leakage and exposure of sensitive functionality, the ENISA guideline highlights the importance of verifying the identity of the remote endpoint via trusted certificates before establishing a secure connection. Both aim to mitigate risks related to data interception and unauthorized access, albeit in slightly different contexts (WebView configuration versus secure TLS communication).
- MASVS-PRIVACY-1: Both "MASVS-PRIVACY-1" and the ENISA Guideline focus on ensuring the security and privacy of user data. "MASVS-PRIVACY-1" underlines the importance of data minimization and informed consent, particularly when it comes to third-party SDK data sharing. The ENISA Guideline emphasizes the mitigation of risks associated with sensitive data transmission by using secure connections and verifying server identities through trusted certificates. Both standards implicitly advocate for the protection of user data by controlling access to it and ensuring secure data transmission, which forms a cohesive approach to securing user data from unauthorized access and potential breaches.
- MASVS-PRIVACY-3: Although "MASVS-PRIVACY-3" and the ENISA Guideline address different aspects of mobile application security and privacy, there is a correlation in their underlying principles. "MASVS-PRIVACY-3" emphasizes the users' right to be informed about how their data is managed, including unexpected data collection practices, which contributes to data protection and user trustworthiness. The ENISA Guideline complements this by providing a technical requirement for protecting sensitive data during transmission—specifically, that apps should establish secure connections with servers verified by trusted certificates to prevent man-in-the-middle attacks. Together, both guidelines

work toward providing a safer and more transparent environment for the user's data, which includes both informing users and ensuring secure data handling practices.

- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the described ENISA Guideline is that both emphasize the importance of operating in a secure environment to protect sensitive data. MASVS-RESILIENCE-1 focuses on ensuring that the platform on which the app is running has not been tampered with, as a compromised OS can undermine the app's security controls, such as secure storage and sandboxing. If the integrity of the operating system is compromised, it could potentially allow unauthorized access to sensitive data or disable security measures that protect against various attacks, including man-in-the-middle (MITM). On the other hand, the ENISA Guideline underlines the necessity of verifying the identity of the remote end-point before establishing a secure connection for sensitive data transactions. This is to mitigate the risk of MITM attacks by ensuring that a TLS connection is only established with servers that present trusted certificates, thus protecting the data in transit. The correlation lies in the common goal of both: securing sensitive data by ensuring the environment (either the platform/OS in MASVS-RESILIENCE-1 or the secure connection end-point in the ENISA Guideline) is trustworthy and has not been compromised. Both are preventive measures to protect against MITM attacks and other security threats that could compromise sensitive data.

5.4 Implementation Guidance (ENISA 4.4):

ENISA Secure Smartphone Development Guidance (4.4): Leverage the platform specific support for enforcing additional security requirements for HTTP-based networking requests (e.g., ATS in iOS and clear text traffic opt-out in Android).

5.4.1 OWASP MASVS MAPPING

- MASVS-CODE-4: MASVS-CODE-4 and the ENISA Guideline both emphasize the importance of treating all incoming data as potentially untrusted and ensuring that appropriate security measures are applied to this data. MASVS-CODE-4 is broader in its scope, covering not only network data but all forms of data entry including UI, IPC, filesystem, etc., and focuses on verification and sanitization of incoming data to prevent security issues like SQL injection, XSS, and insecure deserialization. On the other hand, the ENISA Guideline specifically mentions leveraging platform-specific security features for HTTP networking requests, which is a subset of the data entry points mentioned in MASVS-CODE-4. The guideline's mention of ATS (App Transport Security) in iOS and the opt-out of clear text traffic in Android are platform-specific implementations to ensure the security of data in transit over HTTP, which complements the goal of MASVS-CODE-4 to handle all incoming data securely. Both propose ways to enforce security requirements and prevent exploitation of untrusted input.
- MASVS-NETWORK-1: "MASVS-NETWORK-1" concerns the importance of data privacy and integrity by encrypting data and authenticating the remote endpoint, which aligns with the ENISA guideline that suggests leveraging platform-specific support for enforcing additional security requirements for HTTP-based networking requests, like ATS in iOS and clear text traffic opt-out in Android—both of which are designed to ensure secure connections and prevent clear text data transmission.
- MASVS-NETWORK-2: The correlation exists because both MASVS-NETWORK-2, which refers to the practice of certificate pinning or public key pinning, and the ENISA Guideline, which speaks to enforcing additional security requirements for network requests, revolve around the concept of enhancing the security of network communication in mobile applications. Certificate pinning is a specific technique to trust only certain CAs, thereby preventing man-in-the-middle attacks, and it aligns with the broader objective of the ENISA Guideline to leverage platform-specific features to secure HTTP networking requests, such as ATS (App Transport Security) in iOS and clear text traffic opt-out in Android, which also aim to reduce the risk of insecure connections.
- MASVS-PLATFORM-1: "MASVS-PLATFORM-1" addresses the secure interaction between apps and the user through IPC (Inter-Process Communication) mechanisms provided by the platform. The ENISA Guideline pertains to platform-specific supports, such as ATS in iOS and clear text traffic opt-out in Android, which enhance security measures for networking requests. Both "MASVS-PLATFORM-1" and the ENISA Guideline emphasize the use of built-in platform features to elevate the security of different aspects (IPC for "MASVS-PLATFORM-1" and networking requests for ENISA) within an app. They correlate in the sense that they both focus on utilizing platform-specific features to improve overall app security, albeit in different contexts (IPC vs. networking).
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA Guideline exists as both relate to securing specific platform features against po-

tential vulnerabilities or sensitive data leakage. ”MASVS-PLATFORM-2” is focused on the secure configuration of WebViews, which can involve handling of HTTP requests and ensuring that such views do not expose sensitive data or functionalities. The mentioned ENISA Guideline also deals with security for HTTP-based networking requests, like Apple’s App Transport Security (ATS) or Android’s opt-out mechanism for clear text traffic. These are mechanisms provided by the platform to enhance the security of HTTP requests, similar to how MASVS-PLATFORM-2 aims to ensure WebViews are configured securely. Both guidelines emphasize the importance of leveraging available platform features to prevent sensitive data exposure through network communications.

- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the ENISA guideline is that both emphasize the importance of relying on the underlying platform’s security features. MASVS-RESILIENCE-1 discusses the risk of running on a tampered platform that can compromise security features like secure storage and sandboxing, which are essential for app security. Similarly, the ENISA guideline advises leveraging platform-specific support to enhance security, such as ATS (App Transport Security) in iOS and opting out of clear text traffic in Android, which are security features provided by the operating system. Both focus on the necessity of utilizing the security mechanisms provided by the platform to protect the app and its data.
- MASVS-RESILIENCE-2: Both ”MASVS-RESILIENCE-2” and the mentioned ENISA Guideline are focused on enhancing the security and integrity of mobile applications. ”MASVS-RESILIENCE-2” emphasizes the importance of safeguarding the app against modifications to its original code and resources. This can include protections against running modified versions of the app which can harm the app’s intended functionality or compromise its business logic (for example, enabling premium features without payment or cheating in games). The ENISA Guideline, on the other hand, refers to the use of platform-specific features to enforce additional security requirements for HTTP networking, like App Transport Security (ATS) in iOS and network security configurations in Android (typically to prevent clear text traffic). These features contribute to the overall resilience of the application by ensuring secure communication and protecting the integrity of data in transit. Although the ENISA Guideline specifically targets secure networking practices and ”MASVS-RESILIENCE-2” is more generalized in preventing code and resource modifications, both guidelines contribute to the resilience of the app against security threats. In the context of MASVS-RESILIENCE-2, utilizing platform-specific networking security features can be seen as one of the measures to prevent tampering and to protect the app’s integrity.

5.5 Implementation Guidance (ENISA 4.5):

ENISA Secure Smartphone Development Guidance (4.5): Use strong and standardized encryption algorithms (e.g., AES) and appropriate key lengths (check recommendations for the algorithm you use e.g. for the TLS configuration). Remove support for weak ciphers.

5.5.1 OWASP MASVS MAPPING

- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the ENISA Guideline is evident as both emphasize the importance of using strong, standardized encryption methods and best practices in securing the user’s data, especially in scenarios where physical access to the device by attackers is possible. While MASVS-CRYPTO-1 speaks to general best practices in cryptography, the ENISA Guideline provides a more specific recommendation to use robust encryption algorithms like AES with appropriate key lengths, aligning with the best practices mentioned in MASVS-CRYPTO-1. Both guidelines advocate for removing support for weak ciphers, which is also in line with general cryptographic best practices.
- MASVS-CRYPTO-2: Both ”MASVS-CRYPTO-2” and the ENISA Guideline emphasize the importance of strong cryptographic practices. While MASVS-CRYPTO-2 focuses on the entire lifecycle of cryptographic key management, including generation, storage, and protection, the ENISA Guideline highlights the use of strong and standardized encryption algorithms and appropriate key lengths, as well as the removal of support for weak ciphers. These two statements complement each other as strong key management is a critical part of using strong and standardized encryption algorithms effectively. Without proper key management, even strong algorithms can become ineffective or compromised.
- MASVS-NETWORK-1: The ”MASVS-NETWORK-1” requirement involves ensuring data privacy and integrity for app data in transit, specifically by encrypting data and authenticating endpoints, which is what protocols like TLS are designed to do. The ENISA Guideline similarly emphasizes the use of strong and standardized encryption algorithms and appropriate key lengths, as well as the removal of support for weak ciphers, which aligns with the goal of secure data transmission over networks. Both highlight the importance of preventing the misuse of encryption technologies and maintaining secure communication standards.
- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the mentioned ENISA Guideline exists because both are concerned with the protection of sensitive data. MASVS-STORAGE-1 refers to the need for proper protection of sensitive data stored by the application, regardless of the storage location. The ENISA guideline emphasizes the use of strong and standardized encryption algorithms and appropriate key lengths to secure data, including but not limited to storage. Using strong encryption is a method to ensure that sensitive data stored by the app is properly protected, which aligns with the objective of MASVS-STORAGE-1.
- MASVS-STORAGE-2: While the description of ”MASVS-STORAGE-2” specifically addresses the unintentional storage or exposure of sensitive data in publicly accessible locations, the ENISA Guideline emphasizes the use of strong encryption and removal of weak ciphers. The correlation lies in the overarching theme of protecting sensitive data. MASVS-STORAGE-2 implies that developers should be aware of and mitigate the risks

of inadvertent data leaks through proper handling of APIs and system functionalities. The ENISA Guideline complements this by specifying that when data is intentionally stored or transmitted, strong encryption should be used to secure it and that weak ciphers that could lead to data exposure should be removed. Both aim to prevent unauthorized access to sensitive data through different means: one through proper data handling and storage practices, and the other through the use of strong cryptographic measures.

5.6 Implementation Guidance (ENISA 4.6):

ENISA Secure Smartphone Development Guidance (4.6): Enforce secure TLS versions. Safely abort the connection, if this is not possible.

5.6.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between "MASVS-AUTH-1" and the ENISA guideline regarding the enforcement of secure TLS versions is related to the best practices that must be followed when apps connect to a remote endpoint. MASVS-AUTH-1 stresses the importance of following relevant best practices for secure protocol usage when authentication and authorization mechanisms are involved. The ENISA guideline specifically mentions enforcing secure TLS versions, which is an important part of maintaining secure communication channels between the app and remote services. Ensuring the use of secure TLS versions falls under the broader category of following security best practices in network communication, which is covered by MASVS-AUTH-1.
- MASVS-CRYPTO-1: The MASVS-CRYPTO-1 citation emphasizes the importance of cryptography for securing user data, especially on mobile devices that are prone to physical access by attackers. This correlates with the ENISA Guideline on enforcing secure TLS versions, as TLS (Transport Layer Security) is a cryptographic protocol designed to provide secure communication over a computer network. Both statements underline the significance of employing strong and reliable cryptographic measures to protect data. The MASVS-CRYPTO-1 control suggests adherence to general cryptography best practices, which would include the use of secure TLS versions as recommended by ENISA. Safely aborting connections when secure TLS cannot be enforced is a specific guideline that falls under such best practices to prevent compromise of sensitive data during transit.
- MASVS-CRYPTO-2: The MASVS-CRYPTO-2 requirement related to the management of cryptographic keys throughout their lifecycle aligns with the ENISA guideline to enforce secure TLS versions and safely abort the connection if this is not possible. Both controls are concerned with ensuring the security of communications through proper cryptographic practices. MASVS-CRYPTO-2's focus on key management ensures that the keys used in TLS, which is a cryptographic protocol, are properly generated, stored, and protected, fulfilling prerequisites for secure TLS deployment as advised by ENISA. If key management is compromised, TLS security cannot be guaranteed, making it essential to enforce secure versions and abort connections if security requirements are not met. Proper key management is a foundation for secure TLS.
- MASVS-NETWORK-1: The control "MASVS-NETWORK-1" and the ENISA guideline both emphasize the importance of securing data in transit. MASVS-NETWORK-1 underlines the need for data privacy and integrity by using encryption and endpoint authentication, similar to what is achieved with TLS. The ENISA guideline specifically focuses on enforcing secure TLS versions and safely aborting connections if secure TLS cannot be enforced. Both statements advocate for strong network security measures to prevent data interception or tampering, making them correlated in their aim to ensure secure communication for mobile apps.
- MASVS-NETWORK-2: The correlation is that both "MASVS-NETWORK-2" and the ENISA guideline refer to practices that enhance the security of the network communication of mobile applications. MASVS-NETWORK-2 specifically focuses on the concept of

certificate or public key pinning, which involves limiting the trusted Certificate Authorities (CAs) to a defined set that the application expects, therefore preventing man-in-the-middle attacks facilitated by compromised or untrusted CAs. The ENISA guideline, on the other hand, emphasizes enforcing secure TLS (Transport Layer Security) versions and safely aborting connections if secure TLS cannot be established. While it doesn't explicitly mention certificate pinning, enforcing secure TLS versions is part of ensuring that the application is using the latest and most secure protocols that have not been compromised. Both controls are aimed at preventing attackers from intercepting or manipulating the encrypted communication in a man-in-the-middle attack. Certificate pinning complements the use of secure TLS versions by adding an additional layer of verification that further restricts which servers and CAs the application will communicate with.

5.7 Implementation Guidance (ENISA 4.7):

ENISA Secure Smartphone Development Guidance (4.7): Use certificates signed by trusted CA providers. Do not allow self-signed certificates and do not disable or ignore certificate chain validation.

5.7.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA guideline about using certificates signed by trusted CA providers is evident in the emphasis on following best practices to ensure secure protocol use. The ENISA guideline specifically addresses a secure use of TLS/SSL certificates, which is a crucial part of secure authentication and authorization protocols implemented by apps as described in MASVS-AUTH-1. Proper certificate validation is part of the best practices for secure communications between the app and remote endpoints.
- MASVS-AUTH-3: While the specific MASVS-AUTH-3 guideline mentioned focuses on the implementation of additional forms of authentication for sensitive actions inside an app, it inherently considers the underlying assumption of communication security. Using certificates signed by trusted Certificate Authorities (CAs) is a foundational element of ensuring secure communication channels. Although the ENISA Guideline explicitly talks about using trusted CAs for certificates and not allowing self-signed certificates (which is more directly applicable to transport security), both guidelines aim to increase the security posture of the application. MASVS-AUTH-3’s implementation of secure additional authentication would rely on secure communication, which would be compromised if the application accepted self-signed certificates or disabled certificate chain validation. Therefore, MASVS-AUTH-3’s goal of securely implementing additional forms of authentication correlates with the ENISA Guideline’s emphasis on using certificates from trusted CAs to maintain a secure environment for such operations.
- MASVS-CRYPTO-1: MASVS-CRYPTO-1 emphasizes the importance of cryptography in mobile environments where physical access to devices is possible, highlighting the need for sound cryptographic best practices often outlined in external standards. The ENISA guideline mentioned aligns with this because using certificates signed by trusted CA providers ensures the integrity and authenticity of the cryptographic processes. This is a best practice and a component of a comprehensive approach to securing data through cryptography. Disallowing self-signed certificates and not ignoring certificate chain validation is consistent with the MASVS-CRYPTO-1’s emphasis on adhering to established cryptographic standards and protecting user data.
- MASVS-CRYPTO-2: Both ”MASVS-CRYPTO-2” and the ENISA guideline relate to the proper management and validation of cryptographic credentials. ”MASVS-CRYPTO-2” emphasizes the importance of managing cryptographic keys responsibly throughout their lifecycle, including secure generation, storage, and protection. This implicitly includes using keys and certificates from trusted sources. The ENISA guideline explicitly states the importance of using certificates signed by trusted certificate authorities (CAs) and warns against the risks associated with self-signed certificates as well as ignoring certificate validation. Both sets of guidelines are concerned with ensuring that cryptography is implemented securely and that trustworthiness of cryptographic mechanisms is maintained. The control and guideline complement each other by reinforcing the principle that

cryptographic processes should rely on trusted sources and secure practices for key and certificate management to prevent compromise.

- MASVS-NETWORK-1: Both ”MASVS-NETWORK-1” and the ENISA guideline stress the importance of ensuring secure communication and data integrity over the network. ”MASVS-NETWORK-1” emphasizes the importance of encrypting data and authenticating the remote endpoint, pointing out the risk of developers bypassing secure defaults or using insecure libraries, which could lead to insecure connections. The ENISA guideline specifically addresses one aspect of secure connections by requiring the use of certificates signed by trusted Certificate Authorities (CAs), and advising against the use of self-signed certificates or ignoring certificate chain validation. Both sources advocate for maintaining and not compromising the integrity and privacy of data in transit through proper security measures. Hence, there is a correlation in their objectives of protecting data in transit by endorsing the use of secure, validated cryptographic protocols and methods.
- MASVS-NETWORK-2: Both MASVS-NETWORK-2 and the ENISA Guideline stress the importance of trusting certificates from only specific and reliable Certificate Authority (CA) providers. MASVS-NETWORK-2 refers to the practice of certificate pinning, which is about establishing trust in specific CAs and public keys rather than accepting all default root CAs, thus ensuring that an app communicates only with the intended server. Similarly, the ENISA Guideline advises using certificates from trusted CA providers and cautions against the use of self-signed certificates, as well as the importance of not disabling or ignoring certificate chain validation. Both guidelines aim to enhance the security of the communication channel by ensuring certificates are trusted and validated properly.
- MASVS-PLATFORM-2: The correlation is indirect. The ENISA Guideline about using certificates signed by trusted CA providers and not allowing self-signed certificates is focused on ensuring the authenticity and integrity of secure connections, typically relevant to network security and the protection of data in transit. On the other hand, ”MASVS-PLATFORM-2” is concerned with the secure use of WebViews within mobile applications to prevent sensitive data leakage and exposure of sensitive functionality. Both guidelines aim to protect sensitive data, but they do so in different contexts: ”MASVS-PLATFORM-2” is specific to how WebViews are configured within the app, and the ENISA Guideline pertains to the security of network communications. However, in the broader context of mobile application security, both contribute to a defense-in-depth strategy where each layer or component of the app is secured to minimize overall risk. Thus, while they address different aspects, both are integral parts of securing mobile applications against common vulnerabilities and threats.
- MASVS-RESILIENCE-1: Both the MASVS-RESILIENCE-1 statement and the ENISA Guideline emphasize the importance of operating in a secure and trusted environment. MASVS-RESILIENCE-1 highlights the dangers of running an app on a tampered platform which may compromise security features, while the ENISA Guideline focuses on the trustworthiness of certificates provided by Certificate Authorities (CAs). Both are concerned with validation—be it the integrity of the operating system or the validity of certificates—and ensuring that an app operates in an environment where its security mechanisms, like secure storage and sandboxing, are dependable and not undermined by untrusted modifications or certificates.
- MASVS-RESILIENCE-2: The correlation between ”MASVS-RESILIENCE-2” and the ENISA guideline about the use of certificates signed by trusted CA providers is grounded in their shared focus on ensuring the integrity and authenticity of an application and its communications. ”MASVS-RESILIENCE-2” addresses the need to prevent modifications to the original code and resources, which is a part of maintaining the integrity of the

app's intended functionality. The use of certificates signed by trusted CA providers, as recommended by ENISA, likewise serves to ensure that the app communicates securely and that it is indeed the authentic, unmodified version that was intended for distribution. Both controls are measures against man-in-the-middle attacks, unauthorized modifications, and other attacks that could compromise the security of the app or its communications.

5.8 Implementation Guidance (ENISA 4.8):

ENISA Secure Smartphone Development Guidance (4.8): Introduce certificate pinning.Restrict an application's trusted certificates to a small set of known certificates that are used by the backend servers.

5.8.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between MASVS-AUTH-1 and the ENISA guideline regarding certificate pinning is that both are related to ensuring secure communication and proper authentication/authorization practices between the app and its remote endpoints. MASVS-AUTH-1 mentions enforcing best practices for secure use of protocols, which could include the use of SSL/TLS for secure transmissions. Certificate pinning, as per the ENISA guideline, directly enhances the security of SSL/TLS by restricting the set of trusted certificates, thus preventing man-in-the-middle attacks and ensuring that the app is connecting to the correct server. Both guidelines aim to protect against threats that compromise secure communication and the integrity of authentication/authorization mechanisms.
- MASVS-CODE-2: While MASVS-CODE-2 and the ENISA guideline on certificate pinning address different security aspects of mobile apps—MASVS-CODE-2 focuses on the mechanism to force app updates and the ENISA guideline focuses on limiting trusted certificates—they both ultimately aim to ensure the security of the app and its backend communication. MASVS-CODE-2's intent to rapidly respond to critical vulnerabilities by forcing updates correlates with the intention of the ENISA guideline, which is to prevent man-in-the-middle attacks by making sure the app only communicates with legitimate servers, which may be necessary especially after discovering such vulnerabilities. Both contribute to maintaining a secure and resilient app ecosystem.
- MASVS-CRYPTO-1: The ENISA guideline "Introduce certificate pinning. Restrict an application's trusted certificates to a small set of known certificates that are used by the backend servers." aligns with the description for "MASVS-CRYPTO-1" in the context of protecting user data through cryptographic best practices. Certificate pinning is a security measure that can prevent attackers from eavesdropping on encrypted connections by ensuring an application only trusts predefined certificates. This falls under the umbrella of general cryptography best practices that MASVS-CRYPTO-1 mentions, especially in scenarios where attackers might have physical device access, and thus enhancing data security in mobile environments.
- MASVS-CRYPTO-2: "MASVS-CRYPTO-2" deals with the management of cryptographic keys throughout their lifecycle, including how the keys are generated, stored, and protected. This emphasizes the importance of handling keys with care to maintain the security of cryptographic operations. The ENISA guideline's recommendation to "introduce certificate pinning" is a specific way to handle key management by restricting the set of trusted certificates to a well-defined and small collection that the application is meant to communicate with, therefore reducing the attack surface and making it more difficult for attackers to compromise communications through rogue certificates or poor key management practices. Both "MASVS-CRYPTO-2" and the ENISA guideline focus on ensuring that cryptographic keys and certificates are managed securely to prevent compromise.

- MASVS-NETWORK-1: The correlation between ”MASVS-NETWORK-1” and the ENISA Guideline regarding certificate pinning is evident. MASVS-NETWORK-1 emphasizes the importance of data privacy and integrity in transit by securing connections, which includes using encryption and authentication to validate the remote endpoint, akin to what TLS provides. The ENISA Guideline’s advice to introduce certificate pinning is a specific method to ensure that trust is anchored to a known set of certificates, thereby preventing interception or man-in-the-middle attacks. This guideline directly supports the goal of MASVS-NETWORK-1 by enhancing the security of connections between the app and backend servers. Certificate pinning is indeed a practice that can help fulfill the requirements of MASVS-NETWORK-1 by explicitly verifying the server’s certificate against a known set of certificates, thus ensuring that the app establishes secure connections under all circumstances.
- MASVS-NETWORK-2: The MASVS-NETWORK-2 guideline and the ENISA Guideline both emphasize the practice of restricting trust to a limited set of known certificates. This is typically known as certificate pinning or public key pinning. Both guidelines aim to enhance security by reducing the risk of man-in-the-middle (MITM) attacks that exploit the trust in potentially insecure root CAs pre-installed on the device or framework. By pinning to specific certificates, the trust is limited to the certificates known to be used by the backend servers, thus aligning with the principles stated in each guideline.
- MASVS-PLATFORM-2: The ”MASVS-PLATFORM-2” describes securing WebViews to prevent data leakage and exposure of sensitive functionality, which includes risks related to insecure communication channels. ENISA’s guideline to ”Introduce certificate pinning” is related because certificate pinning is a security measure that helps to prevent man-in-the-middle attacks by ensuring that the application only accepts a predefined set of trusted certificates. By doing so, it ensures that encrypted traffic between the app and backend servers is not susceptible to interception or tampering. Both statements are concerned with strengthening the security of network communication to protect sensitive data and functionality.
- MASVS-RESILIENCE-2: The correlation between ”MASVS-RESILIENCE-2” and the ENISA Guideline ”Introduce certificate pinning” can be seen in the underlying goal of both recommendations, which is to enhance the security of the application by ensuring its integrity and trustworthiness. MASVS-RESILIENCE-2 focuses on preserving the integrity of the app’s original code and resources to prevent unauthorized modifications, which aligns with certificate pinning’s goal of ensuring that the application only communicates with the intended, trusted servers, thereby also helping to prevent man-in-the-middle attacks and ensuring that backdoored versions of the app do not communicate with legitimate servers. Both controls contribute to the resilience of the application against tampering and malicious alterations.
- MASVS-RESILIENCE-3: While MASVS-RESILIENCE-3 focuses on impeding static analysis by making it difficult to understand the app internals, introducing certificate pinning as per the ENISA guideline supports this goal by limiting the ability to intercept and analyze secure communications. Certificate pinning ensures that the application only trusts specific certificates, which makes it harder for an attacker to perform man-in-the-middle attacks that are part of the analysis process. By doing so, it adds a layer of resistance against understanding and tampering with the application’s data exchange, thereby contributing indirectly to the resilience referenced in MASVS-RESILIENCE-3.
- MASVS-RESILIENCE-4: Both MASVS-RESILIENCE-4 and the ENISA Guideline regarding certificate pinning share the goal of increasing the difficulty for an attacker to perform analysis or manipulation of the app at runtime. MASVS-RESILIENCE-4 focuses

on making dynamic analysis and instrumentation difficult to protect against runtime modifications, while certificate pinning restricts trust to specific certificates, thus preventing man-in-the-middle attacks and making it harder to intercept or tamper with secure communication at runtime. Both controls contribute to resilience against runtime threats.

5.9 Implementation Guidance (ENISA 4.9):

ENISA Secure Smartphone Development Guidance (4.9): Design the user interface in a way that warns the user if the peer certificate does not match the expected certificate and provide the ability to abort any further interaction.

5.9.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The ENISA guideline about designing the user interface to warn users about potential certificate mismatches directly contributes to a secure use of involved protocols as mentioned in MASVS-AUTH-1. By warning users of certificate issues and allowing them to abort interactions, the app supports secure authentication and authorization mechanisms, adhering to best practices in app security.
- MASVS-NETWORK-2: Both ”MASVS-NETWORK-2” and the ENISA guideline emphasize the importance of establishing trust in communication channels by validating certificates. MASVS-NETWORK-2 describes certificate pinning, which is a specific method to ensure that an application communicates exclusively with the intended server by validating the server's certificate against a known set of identifiers. The ENISA guideline complements this by advocating for a design in which users are warned when there is a discrepancy in the peer certificate, which could indicate a man-in-the-middle attack. Both controls are aimed at preventing users from unknowingly interacting with a potentially compromised or malicious server, thereby enhancing the security of network communications for the application.
- MASVS-PLATFORM-2: Both ”MASVS-PLATFORM-2” and the ENISA guideline emphasize the importance of secure configuration to protect sensitive data and ensure the integrity of user interactions. ”MASVS-PLATFORM-2” focuses on securing WebViews, which involves a secure setup to prevent data leaks and unauthorized access to native code through JavaScript bridges. The ENISA guideline stresses the importance of a user interface design that alerts users of potential security risks when there is a certificate mismatch, allowing users to abort unsafe interactions. Both guidelines aim to protect against threats that could compromise user data or the security of the app's operations, thus showing a correlation in their intent to prevent security breaches through proper configuration and secure design practices.
- MASVS-PRIVACY-3: Both the MASVS-PRIVACY-3 control and the ENISA guideline emphasize the importance of transparency and informed user consent regarding data usage and security. MASVS-PRIVACY-3 focuses on the user's right to know how their data is being used, including unexpected data collection practices, while the ENISA guideline ensures that users are warned about potential security issues (such as a mismatched peer certificate) that could affect the confidentiality and integrity of their data during interaction. Both contribute to raising user awareness and giving them control over their personal data and security, albeit in different contexts.

5.10 Implementation Guidance (ENISA 4.10):

ENISA Secure Smartphone Development Guidance (4.10): SMS and MMS should not be used to send sensitive data (e.g., two-factor authentication tokens) to or from mobile end-points as SMS and MMS can be intercepted.

5.10.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA guideline regarding the use of SMS and MMS for sensitive data is that both emphasize the importance of secure transmission of authentication and authorization data. ”MASVS-AUTH-1” focuses on apps following best practices for secure communication with remote endpoints, which would include avoiding insecure transmission methods like SMS and MMS that can easily be intercepted, as highlighted by the ENISA guideline. Therefore, adhering to ”MASVS-AUTH-1” would imply not using SMS or MMS for transmitting sensitive tokens, in agreement with the ENISA guidance on information security.
- MASVS-AUTH-3: The MASVS-AUTH-3 requirement mentions the implementation of additional forms of authentication for sensitive actions within an app, considering different methods such as biometric, pin, MFA code generator, email, deep links, etc., which need to be implemented securely. The ENISA guideline advises against using SMS and MMS for sending sensitive information like two-factor authentication tokens since they can be intercepted. Both sources correlate in emphasizing the need for secure implementation of authentication mechanisms, and while MASVS-AUTH-3 does not explicitly mention SMS or MMS, the ENISA guideline’s specific caution against using them falls under the broader security considerations that MASVS-AUTH-3 would encompass. Therefore, there is a correlation between MASVS-AUTH-3’s call for secure implementation of authentication measures and ENISA’s specific advice against using certain insecure channels (SMS/MMS) for sensitive authentication data.
- MASVS-CRYPTO-1: Correlation exists between ”MASVS-CRYPTO-1” and the ENISA guideline. ”MASVS-CRYPTO-1” emphasizes the importance of cryptography, especially in a mobile environment, because physical access to user devices is a common risk. The guideline’s focus on cryptography best practices aligns with the ENISA recommendation against using interceptable communication methods like SMS and MMS for sensitive data. Both are concerned with ensuring secure handling of sensitive information against potential interception.
- MASVS-CRYPTO-2: The correlation between MASVS-CRYPTO-2 and the ENISA guideline regarding SMS and MMS is that both are concerned with the security and integrity of sensitive data. MASVS-CRYPTO-2 emphasizes the importance of managing cryptographic keys effectively to maintain the strength of encryption, while the ENISA guideline highlights a specific vulnerability with sending sensitive data such as two-factor authentication tokens over SMS and MMS, which is a form of poor key (or sensitive data) management if intercepted. Both are addressing the overarching theme of protecting sensitive data by managing its security through the lifecycle (for keys) or by choosing secure communication methods (for data transmission), thus acknowledging the risk of interception and the need for robust security practices.
- MASVS-NETWORK-1: The given description of ”MASVS-NETWORK-1” emphasizes the importance of data privacy and integrity for data in transit by using secure connections

and proper encryption, such as TLS. This correlates with the ENISA guideline advising against using SMS and MMS for sensitive data due to interception risks, as both statements address the necessity of protecting data in transit from being compromised.

- MASVS-PLATFORM-1: Both "MASVS-PLATFORM-1" and the ENISA guideline are addressing the security of inter-process communication (IPC) mechanisms and the secure handling of sensitive data. "MASVS-PLATFORM-1" is focused on ensuring that all IPC interactions are secure, while the ENISA guideline is specifically addressing the insecurity of transmitting sensitive data via SMS and MMS, which can be considered forms of IPC. The correlation lies in the shared concern for protecting sensitive data during transmission between processes or apps, and both recommend secure practices to mitigate interception risks.
- MASVS-PLATFORM-3: Both "MASVS-PLATFORM-3" and the ENISA Guideline emphasize the need to protect sensitive data on mobile platforms. "MASVS-PLATFORM-3" is about preventing unintentional leakage of sensitive data through platform mechanisms, which includes being careful about how such information is displayed to prevent issues like shoulder surfing or accidental sharing. The ENISA Guideline advises against using SMS and MMS for transmitting sensitive data because these can be intercepted, which aligns with the prevention of data leakage emphasized by "MASVS-PLATFORM-3". Both statements are concerned with the secure handling of sensitive information in the context of mobile security.
- MASVS-PRIVACY-1: Both MASVS-PRIVACY-1 and the ENISA Guideline focus on the protection of user data and preventing unauthorized or unintended data access. MASVS-PRIVACY-1 advocates for minimal data access aligned with app functionality and user consent, while the ENISA Guideline specifically addresses the risks associated with transmitting sensitive data via SMS and MMS, which can be intercepted. Both underscore the importance of security measures in mobile applications to safeguard privacy and personal data.
- MASVS-PRIVACY-3: The correlation between "MASVS-PRIVACY-3" and the ENISA guideline regarding SMS and MMS is that both are focused on the protection and proper handling of user data. MASVS-PRIVACY-3 emphasizes the users' right to be informed about how their data is used, including data collection, storage, and sharing practices, which should not include unexpected behaviors like background data collection. In line with this, the ENISA guideline specifically addresses the security of sensitive data by advising against the use of SMS and MMS for transmitting such data due to the possibility of interception. Both serve the broader goal of ensuring privacy and security of user data in mobile applications.
- MASVS-STORAGE-1: Both "MASVS-STORAGE-1" and the ENISA guideline emphasize the importance of protecting sensitive data on mobile devices. "MASVS-STORAGE-1" focuses on ensuring that sensitive data stored locally by an app is properly protected, regardless of whether it is stored in private or public locations. The ENISA guideline specifically mentions that SMS and MMS are not secure methods for transmitting sensitive data such as two-factor authentication tokens because they can be intercepted. Both statements correlate in the sense that they acknowledge the vulnerabilities associated with handling sensitive data on mobile devices and prescribe measures to safeguard it.

5.11 Implementation Guidance (ENISA 4.11):

ENISA Secure Smartphone Development Guidance (4.11): Always use platform supported or vetted frameworks for establishing secure communication channels. Avoid using custom solutions.

5.11.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The guideline from ENISA recommending the use of platform supported or vetted frameworks for establishing secure communication channels is in correlation with the MASVS-AUTH-1 description, which emphasizes the need for applications to adhere to best practices for secure protocol usage, particularly in the context of user authentication and authorization. Both stress the importance of utilizing secure and proven methods rather than custom, potentially untested solutions that may introduce security vulnerabilities. By following vetted frameworks, an app is more likely to implement the authentication and authorization mechanisms correctly, which aligns with the objective of MASVS-AUTH-1.
- MASVS-AUTH-2: The correlation between "MASVS-AUTH-2" and the ENISA Guideline is that both emphasize the use of secure and trusted methods for authentication and communication. "MASVS-AUTH-2" refers to the correct implementation of biometric or local PIN code authentication, which often involves using platform-supported features to ensure security. Similarly, the ENISA guideline advises against custom solutions and instead recommends using platform-supported or vetted frameworks for secure communication. The common theme is the reliance on trusted and vetted mechanisms provided by the platform to ensure security, which applies to both local authentication methods and establishing secure communication channels.
- MASVS-AUTH-3: The correlation between "MASVS-AUTH-3" and the ENISA guideline is that both emphasize the importance of secure implementations for authentication mechanisms. "MASVS-AUTH-3" suggests that sensitive actions within an app require additional forms of authentication, which should be implemented securely, while the ENISA guideline urges the use of platform-supported or vetted frameworks for secure communications, essentially advising against custom solutions that may not be as secure. Both references advocate for security in authentication processes, aligning with each other in promoting the use of established, secure methods over unverified custom approaches.
- MASVS-CODE-3: The correlation exists in the sense that both statements emphasize the importance of using trusted components over custom ones, in order to enhance security. MASVS-CODE-3 implies the use of secure, well-tested third-party components which can be scrutinized for known vulnerabilities, aligning with the ENISA guideline that suggests using platform-supported or vetted frameworks rather than custom solutions, as the latter may not have undergone thorough security assessments. The underlying principle in both is to rely on standard, community or industry-vetted solutions that are more likely to be secure.
- MASVS-CRYPTO-1: The correlation between "MASVS-CRYPTO-1" and the ENISA guideline is that both emphasize the importance of using established, secure cryptographic practices. MASVS-CRYPTO-1 highlights the significance of cryptography in protecting user data, especially on mobile devices where physical access to the device can be a risk, and suggests that best practices are often defined by external standards. The ENISA

guideline reinforces this by advising the use of platform-supported or vetted frameworks instead of custom solutions for secure communication, which aligns with following best practices and using established cryptographic methods. Both are concerned with adhering to proven security measures to ensure the confidentiality and integrity of user data during transmission.

- MASVS-CRYPTO-2: ”MASVS-CRYPTO-2” focuses on the lifecycle management of cryptographic keys, emphasizing the importance of proper generation, storage, and protection. The ENISA guideline advises using platform-supported or vetted frameworks for secure communication to avoid the potential weaknesses of custom solutions. This reflects a shared concern for using established, well-reviewed processes and tools to maintain security in cryptographic practices. Both the MASVS requirement and the ENISA guideline implicitly acknowledge that even strong cryptography can be undermined by poor implementation, including key management and insecure communication channels. Thus, there is a clear correlation between the two in promoting the use of trusted, rigorously-tested methods to ensure the effectiveness of cryptographic security measures.
- MASVS-NETWORK-1: Both the description of ”MASVS-NETWORK-1” and the ENISA Guideline emphasize the importance of using secure, platform-supported or vetted frameworks and methods to protect data in transit. ”MASVS-NETWORK-1” focuses on the critical nature of ensuring data privacy and integrity when an app communicates over a network by encouraging the use of encryption and endpoint authentication. This is in alignment with the ENISA Guideline that advises against custom solutions and instead recommends the use of platform-supported or vetted frameworks to establish secure communication channels. Both are advising on best practices to minimize the risks associated with network communications by adhering to secure and standard protocols like TLS, thereby showing a correlation in their core message.
- MASVS-NETWORK-2: The correlation between ”MASVS-NETWORK-2” which describes certificate pinning or public key pinning as a method to trust specific CAs (Certificate Authorities) and the ENISA guideline recommending the use of platform supported or vetted frameworks for secure communications is that both are focused on enhancing the security of network communications. Using certificate pinning is a way to enforce the trust in known, specific CAs, reducing the risk of trusting certificates from potentially untrustworthy CAs that are included by default in the device’s root certificate store. This aligns with the ENISA guideline because it suggests relying on security practices that are supported and vetted, rather than creating custom, potentially less secure or unveted, solutions. Certificate pinning is a commonly recommended security practice for protecting against certain types of man-in-the-middle attacks and is supported by various network security frameworks.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the ENISA Guideline is clear. ”MASVS-PLATFORM-1” focuses on ensuring the secure interaction with an app through its IPC (Inter-Process Communication) mechanisms, which are provided by the platform. This involves data exposure and functionality that may be available to both installed apps and the user. The ENISA Guideline emphasizes the use of platform-supported or vetted frameworks for secure communication channels, which aligns with the MASVS requirement by implying that these established and vetted mechanisms are more secure than custom solutions. Both stress the importance of relying on the security measures and frameworks that are provided by the platform itself to avoid the risks that come with custom or non-vetted solutions.
- MASVS-PLATFORM-3: While ”MASVS-PLATFORM-3” deals with ensuring sensitive data isn’t leaked through platform mechanisms such as screenshots or shoulder surfing and

the ENISA guideline focuses on using secure communication channels, both are concerned with protecting sensitive data on the platform. "MASVS-PLATFORM-3" addresses the accidental disclosure of sensitive data on the device, and using platform-supported secure communication frameworks, as recommended by ENISA, complements this by avoiding exposures through interception or poor cryptographic practices. Together, they represent comprehensive measures for safeguarding sensitive information within a mobile application's environment.

- MASVS-PRIVACY-3: Both "MASVS-PRIVACY-3" and the ENISA guideline described are concerned with protecting user data and ensuring transparency in its use. While MASVS-PRIVACY-3 focuses on informing users about data practices and adhering to platform guidelines, the ENISA guideline emphasizes the use of secure, platform-supported or vetted frameworks for communication to avoid vulnerabilities that might arise from custom solutions. Both are aligned in their approach to prioritize the security and privacy of user data within the app's design and operation.
- MASVS-RESILIENCE-1: The concept of "MASVS-RESILIENCE-1" which emphasizes the importance of the app running on an uncompromised platform for security features to function properly, correlates with the ENISA guideline which advises using platform-supported or vetted frameworks for secure communications. Both stress the reliance on the underlying platform's security mechanisms—whether it is to ensure the reliability of security features in the case of MASVS-RESILIENCE-1, or the use of secure communication channels per ENISA's guideline. Each acknowledges that custom solutions might bypass these inherent security measures, potentially introducing vulnerabilities.
- MASVS-RESILIENCE-2: "MASVS-RESILIENCE-2" focuses on preventing modifications to the app's original code and resources, ensuring the integrity of the app's intended functionality. Similarly, the ENISA guideline emphasizes using platform-supported or vetted frameworks for secure communication to avoid vulnerabilities that may arise from custom, potentially non-secure solutions. Both aim to maintain the security and integrity of the application by relying on standardized, proven methods rather than bespoke, potentially less secure approaches.
- MASVS-RESILIENCE-3: The correlation between MASVS-RESILIENCE-3 and the ENISA guideline lies in the underlying principle of reducing the attack surface and making it more difficult for an attacker to compromise the application. MASVS-RESILIENCE-3 emphasizes impeding the understanding of an app through obscurity to prevent tampering, while the ENISA guideline advises using platform-supported or vetted frameworks for secure communications instead of custom solutions to leverage the security vetting and updates provided by established frameworks. Both aim to protect the app from being easily analyzed and exploited by attackers.
- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" and the ENISA guideline on using platform-supported or vetted frameworks for secure communication channels is related to the principle of minimizing the risk of sensitive data exposure. "MASVS-STORAGE-2" addresses the concern of unintentionally storing or exposing sensitive data through improper API use or system capabilities, which can be mitigated by being aware of how data storage is handled by the app. The ENISA guideline recommends using vetted frameworks to establish secure communication channels, implicitly suggesting that this minimizes the chances of data being unintentionally exposed due to custom, potentially less secure, solutions. Both the MASVS control and the ENISA guideline aim to prevent the leak of sensitive information by advocating for the use of trusted mechanisms and scrutiny of data handling practices.

5.12 Implementation Guidance (ENISA 4.12):

ENISA Secure Smartphone Development Guidance (4.12): Ensure adequate logs on the server are retained about established connections. In the case of multiple intermediate proxies, make sure that HTTP headers are parsed correctly (e.g., X-Forwarded-For).

5.12.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: Both "MASVS-AUTH-1" and the ENISA Guideline focus on the security of remote connections and data transmission between the app and the server. "MASVS-AUTH-1" emphasizes the importance of secure authentication and authorization practices in apps when they connect to remote endpoints, which implies ensuring both data being transmitted is protected and access is correctly given to legitimate users. The ENISA Guideline suggests retaining adequate logs and correctly parsing HTTP headers like "X-Forwarded-For" in cases of multiple intermediate proxies to track and ensure the integrity of established connections. While "MASVS-AUTH-1" is more focused on the app's responsibilities in managing secure connections and protocols, the ENISA Guideline complements this by highlighting the server-side logging mechanisms to monitor these connections, which could involve tracking authentication and authorization attempts. Both guidelines together contribute to a holistic approach to secure remote connections and data transmission.
- MASVS-NETWORK-1: The correlation between "MASVS-NETWORK-1" and the ENISA Guideline is that both are concerned with securing data during transit across networks, though they address different aspects of this security. MASVS-NETWORK-1 focuses on the importance of data privacy and integrity by establishing secure connections, such as those using TLS encryption and endpoint authentication, to prevent snooping or tampering. Meanwhile, the ENISA Guideline emphasizes the importance of logging and correctly parsing HTTP headers in the context of established connections, particularly in environments with multiple intermediate proxies. These logs are crucial for monitoring, auditing, and investigating security incidents involving network communications. Both controls contribute to the broader goal of ensuring secure network communication within an application's ecosystem.
- MASVS-PRIVACY-3: The correlation exists in the sense that both MASVS-PRIVACY-3 and the ENISA guideline are focused on transparency and accountability regarding user data. MASVS-PRIVACY-3 concerns the user's right to understand how their data is used by the application, which includes data collection, storage, and sharing practices. The ENISA guideline relates to ensuring that logs on the server about established connections are adequately retained, which also implicates transparency in how user data (in this case, connection data) is handled. Proper logging, including parsing HTTP headers such as X-Forwarded-For, can help in tracking how user data flows through the system, which is a part of disclosing to users how their data is managed and ensuring that data handling complies with expected privacy standards. Both standards aim to provide protection and clarity to users about their personal data within the context of app usage and data management.

5.13 Implementation Guidance (ENISA 4.13):

ENISA Secure Smartphone Development Guidance (4.13): In the case of rooted or jailbroken devices, consider to integrate a custom or third party secure container for the transmission channel, since the platform security controls that establish the TLS connection cannot be trusted

5.13.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation exists because MASVS-AUTH-1 discusses the necessity for apps to follow best practices ensuring secure usage of protocols for user authentication and authorization, while the ENISA guideline reinforces this by suggesting additional measures like using a secure container for the transmission channel on compromised (rooted or jailbroken) devices. This is because on such devices, the default platform security controls (like those that establish TLS connections) may be unreliable, making it important for the app to adopt enhanced security practices as endorsed by MASVS-AUTH-1.
- MASVS-AUTH-2: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-AUTH-2” mentions the need for correct implementation of biometrics or a local PIN code for user authentication. This relates to ensuring that authentication mechanisms are secure and reliable even if the app does not communicate with a remote endpoint. In cases where the device is rooted or jailbroken, platform-level security controls (including those related to secure storage and TLS connections) may be compromised. The ENISA guideline suggests using a custom or third-party secure container specifically when devices are rooted or jailbroken, acknowledging that the built-in platform security can no longer be relied upon in such cases. The use of a secure container can help in maintaining the security of the transmission channel, thereby indirectly supporting the secure implementation of authentication mechanisms as stated in ”MASVS-AUTH-2.” In summary, there is a correlation because both are concerned with maintaining authentication security in scenarios where device integrity is in question and where typical platform-level security measures may not be effective. Therefore, custom or enhanced security measures are recommended to ensure that authentication and data transmission remain secure.
- MASVS-AUTH-3: MASVS-AUTH-3 addresses the need for additional forms of authentication for sensitive actions within an app, aiming to enhance security measures beyond basic username and password authentication. The ENISA guideline regarding the use of a secure container for the transmission channel on rooted or jailbroken devices is a specific instance of implementing additional security measures. It recognizes that the base security controls (such as TLS) cannot be trusted on compromised devices, which correlates with the MASVS-AUTH-3 emphasis on secure implementation of added authentication methods, albeit in a broader sense. Both MASVS-AUTH-3 and the ENISA guideline are concerned with strengthening the security of authentication and data transmission within an application.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the ENISA guideline is that both emphasize the importance of employing secure cryptographic practices in mobile environments where there is a higher risk of attackers gaining physical access to user devices. ”MASVS-CRYPTO-1” discusses the importance of cryptography for securing user data, especially in a mobile context. The ENISA guideline complements this by specifically mentioning that in scenarios where devices are rooted or jailbroken,

standard platform security controls for transmission (like TLS) cannot be trusted. Hence, it suggests using a secure container for the transmission channel as a part of cryptographic best practices to ensure data security. This is in line with the general cryptography best practices mentioned in "MASVS-CRYPTO-1".

- MASVS-CRYPTO-2: The correlation between "MASVS-CRYPTO-2" and the ENISA Guideline concerning rooted or jailbroken devices can be established through the concept of key management and environmental trust. "MASVS-CRYPTO-2" emphasizes the importance of cryptographic key management throughout their lifecycle, which includes ensuring keys are generated, stored, and protected securely. This control implicitly acknowledges that the security of cryptographic operations relies not only on the strength of the algorithms but also on the environment in which these cryptographic keys are managed. Poor key management practices can jeopardize the entire encryption system regardless of the algorithm's strength. The ENISA Guideline focuses on additional precautions in environments where the device may be compromised, such as rooted or jailbroken devices. These modifications can bypass platform security controls, potentially exposing cryptographic operations to risks such as key extraction or manipulation. Both controls advocate for robust security measures tailored to the operating environment's security posture. MASVS-CRYPTO-2 is focused on the general management of keys, while the ENISA Guideline is specific to compromised environments. Nevertheless, both are fundamentally concerned with maintaining the integrity and confidentiality of cryptographic operations in potentially insecure or untrusted environments.
- MASVS-NETWORK-1: Both the "MASVS-NETWORK-1" and the ENISA Guideline highlight the importance of ensuring secure data transmission. While the MASVS-NETWORK-1 focuses on establishing secure connections through encryption and endpoint authentication, typically with TLS, and warns against disabling secure defaults or bypassing them, the ENISA Guideline addresses the specific scenario of rooted or jailbroken devices. In such cases, the default platform security is considered unreliable, and the guideline recommends using a secure container for the transmission channel as an alternative. Both sources emphasize the need for maintaining data privacy and integrity in network communications, just from slightly different approaches or contexts.
- MASVS-NETWORK-2: The correlation between "MASVS-NETWORK-2" and the ENISA guideline is clear. Both the MASVS requirement and the ENISA guideline deal with enhancing the security of the network communication channel in mobile applications in the face of compromised device security or trust environments. "MASVS-NETWORK-2" seeks to reduce the risk of man-in-the-middle (MITM) attacks by restricting trust to specific Certificate Authorities (CAs) through techniques such as certificate pinning or public key pinning. This measure ensures the app does not blindly trust all default root CAs provided by the framework or device, which could be a vulnerability especially on rooted or jailbroken devices where the system's trust store might be tampered with. Similarly, the ENISA guideline advises the use of a custom or third-party secure container for the transmission channel on rooted or jailbroken devices. This is because rooted or jailbroken devices have compromised platform security controls, including those that establish TLS connections. By using a secure container, the application can enforce its own security measures and isolate the communication channel from the potentially compromised aspects of the device's OS. Both guidelines aim to protect data transmission against interception or tampering by establishing a trusted communication channel, thereby demonstrating a related approach to network security in mobile applications, especially on compromised devices.
- MASVS-PLATFORM-1: The correlation exists because "MASVS-PLATFORM-1" is concerned with ensuring that Inter-Process Communication (IPC) mechanisms provided by

the platform are used securely, which includes the secure interaction between the app and the platform or other installed apps, regardless of the device's integrity. The ENISA Guideline advises the use of secure containers, especially on rooted or jailbroken devices, because the default platform security controls (such as those involved in establishing TLS connections) may be compromised on such devices. Both are emphasizing the importance of maintaining secure data transmission and functional exposure even when the underlying platform security is not reliable, indicating a conceptually similar security consideration.

- MASVS-PLATFORM-3: MASVS-PLATFORM-3 and the ENISA guideline both address the concern of protecting sensitive data under conditions where platform security cannot be fully trusted. While MASVS-PLATFORM-3 emphasizes the need to prevent the unintentional leakage of sensitive data displayed in the UI due to platform mechanisms, the ENISA guideline specifically mentions the additional risks posed by rooted or jailbroken devices and recommends the use of secure containers to safeguard the transmission channel. Both controls are focused on reducing the risk of sensitive data exposure considering the platform's security limitations.
- MASVS-PRIVACY-1: Although there may not be a direct one-to-one correlation between the MASVS-PRIVACY-1 control that focuses on data minimization and informed consent and the ENISA guideline which deals with the transmission channel security on rooted or jailbroken devices, there exists an indirect correlation in the broader context of data security and privacy principles. Both aim to enhance user privacy and secure data handling: MASVS-PRIVACY-1 does this by limiting data access and ensuring user consent, thereby reducing the attack surface and potential data leaks, while the ENISA guideline addresses the risk of compromised platform security controls in the specific scenario of rooted or jailbroken devices, recommending secure containers to safeguard the data during transmission. Both are concerned with protecting sensitive data under their respective circumstances.
- MASVS-RESILIENCE-1: Both "MASVS-RESILIENCE-1" and the ENISA Guideline address the risks associated with running apps on a compromised platform. "MASVS-RESILIENCE-1" highlights the importance of validating the security integrity of the OS to ensure that security features such as secure storage and sandboxing can be trusted. Similarly, the ENISA Guideline advises the use of secure containers on rooted or jailbroken devices because the platform's default security controls, particularly those establishing TLS connections, are no longer reliable. Both emphasize the need for additional security measures when the platform's trustworthiness is compromised.
- MASVS-RESILIENCE-2: The Mobile Application Security Verification Standard (MASVS) Resilience requirement MASVS-RESILIENCE-2 relates to the protection of an app's integrity by preventing modifications to its original code and resources. This correlates with the ENISA guideline which suggests using a secure container for the transmission channel on rooted or jailbroken devices because the platform's default security controls (like those for establishing TLS connections) cannot be trusted in such an environment. Both MASVS-RESILIENCE-2 and the ENISA guideline address concerns over maintaining the security and integrity of an app and its communications on a compromised device. The focus in both instances is ensuring that the app's operation cannot be altered or intercepted by an attacker, which could include preventing code modification (as per MASVS-RESILIENCE-2) or securing the communication channel (as per the ENISA guideline) in the case of compromised OS-level controls.
- MASVS-RESILIENCE-3: The correlation between "MASVS-RESILIENCE-3" and the ENISA guideline mentioned is that both are concerned with protecting an app against threats that arise when a user has an elevated level of control over their device, such as

with rooted or jailbroken devices. MASVS-RESILIENCE-3 emphasizes making it difficult for attackers to understand and tamper with an app through static analysis. The ENISA guideline complements this by addressing the issue of transmission security on compromised devices, advocating for the use of secure containers to maintain trust in the TLS transmission even when the device's platform security controls are unreliable. Together, these measures enhance the overall resilience of the app against reverse engineering and tampering threats.

- MASVS-RESILIENCE-4: Both "MASVS-RESILIENCE-4" and the ENISA Guideline emphasize the importance of safeguarding against dynamic analysis and modifications at runtime, especially on compromised devices where the base platform security cannot be trusted. MASVS-RESILIENCE-4 focuses on making it difficult to perform dynamic analysis and to modify code at runtime, which aligns with the ENISA's recommendation to use secure containers on rooted or jailbroken devices to protect the transmission channel due to the loss of trust in the platform's security mechanisms.
- MASVS-STORAGE-1: The correlation between "MASVS-STORAGE-1" and the ENISA Guideline lies in the protection of sensitive data. MASVS-STORAGE-1 emphasizes that apps should securely handle and store sensitive data, regardless of whether it is kept in private or public storage areas. This aligns with the ENISA Guideline, which specifically mentions the use of secure containers on rooted or jailbroken devices because the usual platform security measures (like establishing TLS connections) cannot be entirely relied upon in such compromised environments. Both statements underscore the necessity of implementing robust security precautions when dealing with sensitive data storage and transmission to safeguard against vulnerabilities in the underlying platform.

Chapter 6
Secure the backend services and the platform server and APIs

The majority of mobile applications interact with a backend using web services or proprietary protocols. Insecure implementation of backend APIs, services, and not keeping the back-end platform hardened/patched will allow attackers to compromise data on the mobile device when transferred to the back-end, or to attack the backend through the mobile application. In this section we try to only provide specific measures to secure mobile application backends (appropriate literature for securing servers and web services exists and the reader should refer to those)

6.1 Implementation Guidance (ENISA 5.1):

ENISA Secure Smartphone Development Guidance (5.1): Carry out a specific check of your code for sensitive data unintentionally transferred between the mobile device and web-server back-ends and other external interfaces - (e.g., is location or other information transferred within file metadata).

6.1.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The MASVS-AUTH-1 requirement in mobile application security relates to ensuring that authentication and authorization mechanisms are correctly implemented and follow best practices. This includes securely handling sensitive data during transfer between the mobile app and remote endpoints. The ENISA guideline specifically calls for a check of the code to prevent sensitive data from being unintentionally transferred, which falls under the overall goal of MASVS-AUTH-1 to securely manage protocols involving user authentication and authorization. Ensuring that sensitive data like location or other information is not leaked within file metadata or through other means is part of following security best practices for user authentication and authorization mechanisms.
- MASVS-CODE-2: The correlation between MASVS-CODE-2 and the ENISA guideline is that both are concerned with reducing the exposure of sensitive data. MASVS-CODE-2 is about being able to react to critical vulnerabilities swiftly to prevent exploitation, which may include vulnerabilities that involve exposure of sensitive data. The ENISA guideline is specific about checking for unintentional data leakages, including through metadata. Both controls imply a need to actively manage the security of the app and protect sensitive data from being compromised.
- MASVS-CODE-4: Both ”MASVS-CODE-4” and the ENISA Guideline emphasize the importance of treating all incoming data as untrusted and the need for data validation and sanitization. While ”MASVS-CODE-4” is broader in scope, addressing multiple data entry points and their associated risks, the ENISA Guideline specifically points out the risk of sensitive data being unintentionally transferred and underscores the necessity of checking code for any such occurrences. Both guidelines are focusing on the security implications of how data is handled and the potential for sensitive information leakage if proper controls are not implemented.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the ENISA guideline is that both emphasize the careful handling and protection of sensitive data in mobile environments. MASVS-CRYPTO-1 underscores the importance of cryptography in securing user data, especially when attackers can gain physical access to the device, thereby implying that data at rest (stored on the device) and data in motion (transferred between the mobile device and other endpoints, like web-server back-ends) should be encrypted and handled according to best practices. The ENISA guideline focuses on conducting checks for sensitive data that might be unintentionally transferred—including within file metadata—which aligns with the MASVS-CRYPTO-1’s goal of ensuring general cryptography best practices to protect such data during transfer.
- MASVS-NETWORK-1: Both the MASVS-NETWORK-1 and the ENISA guideline emphasize the importance of protecting sensitive data during transfer between the mobile device and external systems. MASVS-NETWORK-1 focuses on encrypting data and authenticating the remote endpoint to ensure data privacy and integrity in transit, which

includes preventing unintentional data leaks. Similarly, the ENISA guideline specifically instructs to check for sensitive data that might be unintentionally transferred, possibly as metadata. Both are concerned with the secure handling of data being communicated over the network.

- MASVS-PLATFORM-1: The description of "MASVS-PLATFORM-1" pertains to ensuring secure interactions involving IPC (Inter-Process Communication) mechanisms. Since IPC is a way for various components of an app to communicate with each other, it might involve the transfer of data between the app and external interfaces, which could include sensitive information. The ENISA Guideline stresses the importance of checking for sensitive data that might be unintentionally transferred between the mobile device and web-server backend or other external interfaces, which can occur through IPC mechanisms. Both highlight the importance of securing the transfer of sensitive data, illustrating a correlation between the MASVS control and the ENISA recommendation.
- MASVS-PLATFORM-2: The MASVS-PLATFORM-2 description and the ENISA guideline both emphasize the importance of secure configuration to prevent sensitive data leakage. While MASVS-PLATFORM-2 specifically mentions the secure configuration of WebViews to prevent leaks, including JavaScript bridges, the ENISA guideline talks about checking for unintentional sensitive data transfer in the code, which is a complementary aspect of preventing data leakage through different interfaces, including WebViews in mobile applications. Both are concerned with safeguarding sensitive data during its transfer between mobile devices and external interfaces.
- MASVS-PLATFORM-3: Both the Mobile Application Security Verification Standard (MASVS) PLATFORM-3 and the ENISA Guideline mentioned focus on mitigating risks associated with unintended exposure of sensitive data. MASVS-PLATFORM-3 deals with on-device exposure due to platform mechanisms such as screenshots, notifications, and shoulder surfing, while the ENISA Guideline is more broadly concerned with preventing the unintended transfer of sensitive data between the mobile device and server back-ends or through external interfaces, including potential leaks through metadata. The common theme here is the careful handling and protection of sensitive information to prevent leaks and exposure.
- MASVS-PRIVACY-1: Both "MASVS-PRIVACY-1" and the ENISA Guideline emphasize the importance of data minimization and careful management of sensitive data. "MASVS-PRIVACY-1" talks about apps only requesting access to data they absolutely need and obtaining informed consent, which correlates with the ENISA Guideline's recommendation to check for unintentional transfer of sensitive data to ensure that no unnecessary information is conveyed to external interfaces. Both are concerned with the oversight of data sharing and protection against data leaks or breaches.
- MASVS-PRIVACY-2: There is a correlation between "MASVS-PRIVACY-2" and the ENISA guideline provided. Both emphasize the importance of protecting user identity and sensitive information. MASVS-PRIVACY-2 focuses on the use of unlinkability techniques to prevent user identification and tracking, which directly relates to the ENISA guideline that advises conducting a check for sensitive data (such as location information) that might unintentionally be transferred to external interfaces. Both are concerned with ensuring that user privacy is maintained and that sensitive data is not exposed or repurposed in a way that could compromise user identity.
- MASVS-PRIVACY-3: Both MASVS-PRIVACY-3 and the ENISA Guideline emphasize the importance of transparency and control over the user's personal data. MASVS-PRIVACY-3 requires clear information to be provided to the user about how their data is used, while the ENISA Guideline focuses on ensuring that sensitive data is not unintentionally transferred

to external interfaces. Both guidelines aim to protect user privacy by ensuring users are informed and that their data is not shared without their knowledge or consent.

- MASVS-PRIVACY-4: The correlation between ”MASVS-PRIVACY-4” and the ENISA Guideline is that both emphasize the protection and management of sensitive user data. MASVS-PRIVACY-4 advocates for user control over their data, including the ability to manage, delete, and modify their data, as well as change privacy settings and re-consent when needed. The ENISA Guideline complements this by suggesting a code review specifically to check for unintentional transfer of sensitive data, including metadata, which is a form of data management and protection. Both guidelines aim to minimize the risk of sensitive information exposure and enhance user privacy control.
- MASVS-RESILIENCE-3: Both ”MASVS-RESILIENCE-3” and the ENISA guideline you mentioned are concerned with protecting sensitive information within the context of a mobile app. ”MASVS-RESILIENCE-3” refers to making it difficult for attackers to understand app internals to prevent tampering and reverse engineering, while the ENISA guideline advises checking the code to ensure that sensitive data is not unintentionally transferred, which could be a consequence of tampering or reverse engineering. In essence, while ”MASVS-RESILIENCE-3” addresses obfuscation and anti-tampering measures to protect an app’s internals, the ENISA guideline deals with the practical outcome of ensuring sensitive data is not disclosed, possibly due to such tampering. Both contribute to the overall resilience and security of mobile applications against information disclosure threats.
- MASVS-STORAGE-1: The correlation exists in that both the MASVS-STORAGE-1 requirement and the ENISA guideline address concerns related to the handling and storage of sensitive data on mobile devices. MASVS-STORAGE-1 emphasizes the importance of protecting sensitive data regardless of the storage location used by the app, either private or public. On the other hand, the ENISA guideline is focused on ensuring that sensitive data is not unintentionally transferred between the mobile device and external interfaces, like web-server back-ends, which includes being cautious about metadata in files that may contain sensitive information. Though they focus on different aspects of data handling, both aim to prevent the exposure of sensitive data and ensure its security, making them related with respect to information security and data protection practices in mobile apps.
- MASVS-STORAGE-2: The MASVS-STORAGE-2 control and the ENISA guideline are correlated as both involve the scrutiny of mobile application code to prevent sensitive data from being unintentionally stored, exposed, or transferred. They both acknowledge the risks associated with APIs, system capabilities, and external interfaces that might lead to data leaks and recommend developers to implement measures to avoid such unintended disclosures. The control and the guideline are parallel in that they aim to protect sensitive data from being mishandled due to developer oversight or the misuse of mobile device capabilities.

6.2 Implementation Guidance (ENISA 5.2):

ENISA Secure Smartphone Development Guidance (5.2): All back-end services (web services) for mobile apps should be tested for vulnerabilities periodically, e.g., using static code analyser tools and fuzzing tools for testing and finding security flaws. Perform abuse case testing, in addition to use case testing.

6.2.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: Both ”MASVS-AUTH-1” and the ENISA Guideline emphasize the importance of securing remote endpoints and the mobile apps that interact with them. ”MASVS-AUTH-1” focuses on authentication and authorization best practices that apps should follow when connecting to remote endpoints. The ENISA Guideline complements this by recommending that back-end services undergo periodic security testing to identify vulnerabilities. The combination of enforcing best practices on the app side and testing the back-end services aims to create a comprehensive security strategy that covers both ends of the communication channel.
- MASVS-CODE-2: The correlation between ”MASVS-CODE-2” and the ENISA Guideline can be perceived in terms of the proactive measures to maintain the security of a mobile application. While ”MASVS-CODE-2” is focused on implementing a mechanism to ensure users update the app in case critical vulnerabilities are discovered after production, the ENISA guideline emphasizes the importance of periodic testing of backend services for vulnerabilities using tools like static code analyzers and fuzzing tools. Both the MASVS-CODE-2 control and the ENISA guideline pertain to the domain of vulnerability management and mitigation. MASVS-CODE-2 is about ensuring users are operating on the latest, most secure version of the app, likely responding to previously discovered vulnerabilities. In contrast, the ENISA guideline suggests a regular search for new vulnerabilities to anticipate potential security issues. Both security strategies are important and complementary, where MASVS-CODE-2 is about response and mitigation, and the ENISA guideline is about early detection and prevention.
- MASVS-CODE-3: Both statements emphasize the importance of security assessments in different components of a mobile application. ”MASVS-CODE-3” talks about performing a whitebox assessment and checking for vulnerabilities, even if it’s just the ”low-hanging fruit,” such as scanning libraries for known vulnerabilities. The ENISA Guideline suggests regular testing of back-end services for vulnerabilities using tools like static code analyzers and fuzzing tools, as well as conducting abuse case testing in addition to use case testing. Both acknowledge that while it may not always be possible to perform an exhaustive assessment, certain security evaluations can and should be routinely performed to ensure the detection and remediation of potential vulnerabilities.
- MASVS-CODE-4: The MASVS-CODE-4 requirement emphasizes treating all incoming data as untrusted and ensuring that it is properly verified and sanitized before use, which is intended to prevent security issues such as SQL injection, XSS, or insecure deserialization. The ENISA guideline complements this by recommending periodic testing of back-end services for vulnerabilities using tools like static code analyzers and fuzzers. Both address the concept of validating and testing data and services to protect against security flaws, which are likely to arise from untrusted input if not properly handled. Testing, whether through static analysis or fuzzing, is an implicit acknowledgment that data entry points

may be susceptible to modification by untrusted actors, and therefore, these points need to be rigorously verified, which is the essence of MASVS-CODE-4.

- MASVS-NETWORK-1: The correlation between ”MASVS-NETWORK-1” and the listed ENISA Guideline is that both are focused on ensuring the security and integrity of data in transit for mobile applications. ”MASVS-NETWORK-1” emphasizes the necessity for secure connections and the prevention of disabling or bypassing secure defaults, which directly impacts the data privacy and integrity of network communication. The ENISA Guideline complements this by recommending regular vulnerability assessments on back-end services, which are a critical part of the network communication chain for mobile apps. These assessments, using tools such as static code analyzers and fuzzing tools, help identify security flaws that could compromise data privacy and integrity. Therefore, following ”MASVS-NETWORK-1” contributes to better preparedness for the tests suggested by ENISA, and periodic testing as per ENISA’s advice helps ensure compliance with ”MASVS-NETWORK-1” over time.
- MASVS-PLATFORM-2: MASVS-PLATFORM-2 emphasizes the secure configuration of WebViews to prevent sensitive data leakage and functionality exposure, while the ENISA Guideline highlights the importance of periodic security testing for back-end services. Both focus on preventing security flaws through proper configuration and testing, indicating a correlation between the need for secure implementation and regular security assessments.
- MASVS-RESILIENCE-2: Both ”MASVS-RESILIENCE-2” and the ENISA Guideline are concerned with maintaining the integrity and security of the mobile application and its associated services. MASVS-RESILIENCE-2 focuses on preventing modifications to the app’s original code and resources, which can be a method of introducing backdoors or activating unauthorized features. The ENISA Guideline emphasizes the importance of periodic testing of back-end services for vulnerabilities using tools such as static code analyzers and fuzzing tools. Both are preventive measures aimed at mitigating the risk of unauthorized access and modification, hence there is a correlation as they both seek to protect the mobile application ecosystem from security breaches and maintain the intended functionality of the app.
- MASVS-STORAGE-2: The correlation between MASVS-STORAGE-2 and the ENISA Guideline is that both involve ensuring that sensitive data is not unintentionally exposed or stored in insecure locations. MASVS-STORAGE-2 explicitly mentions the prevention of unintentional data leaks that can occur due to the usage of certain APIs or system features like backups or logs. The ENISA Guideline complements this by recommending periodic vulnerability testing of back-end services, including the use of tools that can help detect security flaws that may result in data exposure. Both guidelines aim to protect sensitive data through proactive measures - one focuses on avoiding leaks in the app’s storage mechanisms, while the other emphasizes the importance of security testing to prevent vulnerabilities that could lead to data exposure.

6.3 Implementation Guidance (ENISA 5.3):

ENISA Secure Smartphone Development Guidance (5.3): Disable metadata publishing (e.g., metadata for WSDL documents and for WSDL derived objects), in order to prevent unintentional disclosure of potentially sensitive service metadata.

6.3.1 OWASP MASVS MAPPING

- MASVS-NETWORK-1: The correlation between "MASVS-NETWORK-1" and the ENISA Guideline regarding disabling metadata publishing lies in the interest of protecting data privacy and integrity. "MASVS-NETWORK-1" focuses on ensuring that data in transit is kept private and maintains integrity, which involves setting up secure network connections and preventing vulnerabilities in transmission by using encryption and proper authentication. Similarly, the ENISA Guideline suggests disabling metadata publishing to avoid the accidental leak of sensitive information that could be exploited by adversaries. Both guidelines are aimed at reducing the risk of exposing sensitive data to unauthorized parties, reinforcing the importance of securing communication channels and safeguarding information handled by the application.
- MASVS-PLATFORM-1: Both "MASVS-PLATFORM-1" and the ENISA guideline emphasis on secure interaction with the app components and prevention of unintentional data exposure. "MASVS-PLATFORM-1" pertains to secure IPC (Inter-Process Communication) mechanisms, which would include the secure management of how data and functionality are exposed to other apps and users. The ENISA guideline's directive to disable metadata publishing aligns with this by mitigating the risk of revealing sensitive information through service metadata, as both involve safeguarding the details that could be exploited if improperly disclosed.
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA guideline on disabling metadata publishing relates to the underlying principle of preventing sensitive data leakage. "MASVS-PLATFORM-2" addresses the need to configure WebViews securely to avoid exposing sensitive information, while the ENISA guideline aims to prevent the unintentional disclosure of sensitive service metadata. Both guidelines are concerned with enhancing the security of software components and protecting sensitive data.
- MASVS-PLATFORM-3: Both MASVS-PLATFORM-3 and the ENISA Guideline pertain to the prevention of unintentional disclosure of sensitive information. MASVS-PLATFORM-3 addresses the risk of sensitive data being leaked through mechanisms like auto-generated screenshots or physical observation (like shoulder surfing), while the ENISA Guideline advises against publishing metadata that could reveal sensitive service data. Both guidelines aim to protect data that might be exploited if it were exposed. Even though they deal with different types of data and mechanisms of exposure, the underlying principle of avoiding the inadvertent release of sensitive information is a point of correlation between the two.
- MASVS-PRIVACY-1: The correlation exists in the concept of limiting unnecessary data exposure. MASVS-PRIVACY-1 advocates for minimal data access, informed user consent, and careful sharing with third parties, emphasizing the importance of consent and reducing data breaches. Similarly, the ENISA guideline suggests disabling metadata publishing to

prevent unintentional disclosure of sensitive information. Both are concerned with the principle of data minimization and controlling access to private information.

- MASVS-PRIVACY-2: The ”MASVS-PRIVACY-2” control and the ENISA guideline both emphasize the protection of user privacy through the minimization of data that could lead to user identification. ”MASVS-PRIVACY-2” focuses on techniques such as data abstraction, anonymization, and pseudonymization to prevent user identification and tracking, and it also mentions isolating data streams for specific purposes to prevent repurposing that could compromise privacy. The ENISA guideline similarly suggests disabling metadata publishing to avoid accidental disclosure of sensitive information. Both controls are aligned in their goal to reduce the risk of exposing private user information by limiting data availability and controlling data usage.
- MASVS-PRIVACY-3: The correlation between ”MASVS-PRIVACY-3” and the ENISA guideline on disabling metadata publishing is that both are concerned with the protection of user’s privacy and the prevention of unintentional disclosure of sensitive information. ”MASVS-PRIVACY-3” emphasizes the user’s right to know how their data is used and mandates clear communication regarding data practices, while the ENISA guideline advises against the publishing of metadata that could unintentionally disclose sensitive information about services and, by extension, user data associated with those services. Both guidelines aim to enhance privacy controls and reduce the risk of data leakage.
- MASVS-RESILIENCE-3: Both ”MASVS-RESILIENCE-3” and the ENISA Guideline mentioned about disabling metadata publishing are aimed at protecting sensitive information and impeding an attacker’s understanding of the internal workings of a system. The MASVS-RESILIENCE-3 focuses on making static analysis of an app difficult, whereas the ENISA Guideline specifically addresses preventing disclosure through metadata. Both are measures to increase security by reducing the amount of information readily available to potential attackers, although they apply to different aspects of information security and different types of systems (apps for MASVS-RESILIENCE-3 and services for the ENISA Guideline).
- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the ENISA Guideline about disabling metadata publishing lies in the principle of protecting sensitive data. MASVS-STORAGE-1 emphasizes the need for appropriate protection of sensitive data stored locally by mobile apps, whether the storage is private or public. The ENISA Guideline recommends disabling metadata publishing to prevent unintentional disclosure of sensitive service metadata, which aligns with the MASVS-STORAGE-1’s intent to safeguard sensitive information from unintended access. Both advocate for measures to prevent the exposure of sensitive data, even though they pertain to different aspects of information security (storage and metadata).
- MASVS-STORAGE-2: Both ”MASVS-STORAGE-2” and the ENISA Guideline concerning the disabling of metadata publishing refer to preventing unintentional exposure of sensitive data. While MASVS-STORAGE-2 is concerned with preventing leaks through APIs, backups, logs, or other system capabilities, the ENISA Guideline specifically addresses the potential risks associated with metadata in web services. However, the core principle in both is the proactive avoidance of accidental data disclosure through the correct use of technologies and awareness of how sensitive information could be exposed.

6.4 Implementation Guidance (ENISA 5.4):

ENISA Secure Smartphone Development Guidance (5.4): Ensure that the back-end platform (server) is running with a hardened configuration with the latest security patches applied to the OS, web server and other application components.

6.4.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-AUTH-1” refers to the best practices for secure user authentication and authorization in mobile apps that connect to a remote endpoint. This is correlated to the ENISA Guideline which emphasizes ensuring the security of the server-side components, including the application of the latest security patches and a hardened configuration on the remote endpoint. Both MASVS-AUTH-1 and the ENISA guideline highlight the importance of security on the remote endpoint (server) to prevent unauthorized access and to ensure that the system as a whole, including mobile app interactions, is secure. The correlation exists because securely implementing authentication and authorization on the app directly relates to maintaining a secure and hardened configuration on the server, which are part of comprehensive security control measures.
- MASVS-CODE-1: Both ”MASVS-CODE-1” and the ENISA Guideline focus on maintaining an up-to-date operating system and applying the latest security patches to mitigate known vulnerabilities, though ”MASVS-CODE-1” refers specifically to the mobile OS on client-side, while the ENISA Guideline pertains to the server-side back-end platform. Nevertheless, the principle of keeping systems updated and patched for enhanced security underlies both guidelines.
- MASVS-CODE-3: MASVS-CODE-3 is concerned with performing a comprehensive security assessment on all app components, including both proprietary and third-party components. This includes looking for known vulnerabilities, particularly in third-party libraries, which is referred to as checking for ”low-hanging fruit.” The ENISA Guideline emphasizes the importance of a hardened security configuration for the back-end platform, including the latest patches for the OS, web server, and other application components. While MASVS-CODE-3 focuses on application components with an emphasis on scanning libraries for vulnerabilities, and ENISA mentions securing the back-end with patches and hardening, both are correlated in the broader context of maintaining security by ensuring all parts of the system are evaluated and kept up-to-date to protect against known vulnerabilities. MASVS-CODE-3 addresses part of what the ENISA guideline recommends, specifically for app components, which indirectly implies that up-to-date security patches should be applied, which is a direct requirement from ENISA for back-end components.
- MASVS-RESILIENCE-1: Both ”MASVS-RESILIENCE-1” and the ENISA Guideline emphasize the importance of relying on a secure and uncompromised platform for trustworthy operations. ”MASVS-RESILIENCE-1” is concerned with ensuring that mobile app data is safe by validating the integrity of the operating system and the platform’s security features. On the other hand, the ENISA Guideline advises hardening the back-end platform and keeping it up-to-date with security patches, which is also a practice aimed at maintaining the security and integrity of the platform. Both are focused on the platform’s trustworthiness and security as a basis for the overall security posture of the system.

6.5 Implementation Guidance (ENISA 5.5):

ENISA Secure Smartphone Development Guidance (5.5): Ensure adequate logs are retained on the back-end in order to detect and respond to incidents and perform forensics (within the limits of data protection law).

6.5.1 OWASP MASVS MAPPING

6.6 Implementation Guidance (ENISA 5.6):

ENISA Secure Smartphone Development Guidance (5.6): Protect the back-end from client initiated log injections that may corrupt or forge the history of events.

6.6.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: MASVS-AUTH-1 is concerned with ensuring that mobile applications follow best practices for secure authentication and authorization when connecting to remote endpoints. This implicitly includes protecting against security threats that could compromise these mechanisms. Log injection is a type of attack where malicious input is sent to the application in an attempt to interfere with its logging system, which can lead to forged or corrupted logs. This can compromise the integrity of the event history, making it difficult to trust logs for security auditing or forensics. Therefore, following best practices for authentication and authorization, as stipulated by MASVS-AUTH-1, would naturally involve implementing measures to protect against client-initiated log injections to maintain the integrity of the event history on the back-end. This aligns with the ENISA Guideline to protect the back-end from client-initiated log injections.
- MASVS-AUTH-2: The correlation between ”MASVS-AUTH-2,” which focuses on the correct implementation of biometrics or local PIN code authentication mechanisms, and the ENISA Guideline regarding the protection of the back-end from client-initiated log injections, lies in the broader context of ensuring the security and integrity of authentication processes. While MASVS-AUTH-2 emphasizes the importance of secure local authentication methods within the app, it implicitly suggests that the integrity and security of local authentication are crucial to ensuring the overall security of the system, including the back-end. If local authentication methods are compromised, manipulated, or incorrectly implemented, it could lead to the possibility of log injections or other attacks, which the ENISA guideline warns against. By ensuring that local authentication mechanisms are correctly implemented (as mentioned in MASVS-AUTH-2), the risk of client-initiated log injections and other attacks that could forge or corrupt the history of events on the back-end is reduced. Both MASVS-AUTH-2 and the ENISA guideline aim to protect different parts of the system, but they are correlated in their mutual goal to secure the authentication process and by extension, the entire application infrastructure.
- MASVS-CODE-4: The description of ”MASVS-CODE-4” and the ENISA guideline both focus on the importance of treating data received by the app as untrusted input and ensuring it is verified and sanitized to prevent security issues. ”MASVS-CODE-4” covers the concept that data entering the app from various entry points might be modified by untrusted actors and might bypass critical security checks, leading to injection attacks or other security vulnerabilities. Similarly, the ENISA guideline is concerned with protecting the back-end from client-initiated log injections, which is a type of attack where untrusted input might be used to corrupt or forge the history of events. Both statements emphasize the need to treat incoming data as potential security risks that must be validated and sanitized to maintain the integrity and security of the application and its back-end systems.
- MASVS-NETWORK-2: Certificate pinning, as described under ”MASVS-NETWORK-2,” enhances security by trusting only specific Certificate Authorities (CAs) rather than all default root CAs provided by the framework or device. This prevents connections to unauthorized or rogue servers, which could be used to inject or manipulate data. By en-

forcing this control, an app mitigates the risk of man-in-the-middle (MITM) attacks, which could be a vector for client-initiated log injections. Since certificate pinning ensures that the app communicates only with the genuine backend (protected by the pinned certificate), it makes it more difficult for an attacker to introduce forged or corrupt log entries into the backend system, thereby helping to protect the integrity of the event history. This correlates with the ENISA guideline to "Protect the back-end from client initiated log injections that may corrupt or forge the history of events," as both controls intend to safeguard the communication channel and data integrity between the client and the backend.

- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA guideline on protecting the back-end from client-initiated log injections can be seen in the emphasis on secure configuration and protection against data leakage and exposure of sensitive functionality. The MASVS-PLATFORM-2 description points out the need for secure configuration of WebViews to prevent sensitive data leakage, which implicitly includes avoiding situations where a client could inject logs or forge history. By ensuring that WebViews do not expose sensitive functionality, such as JavaScript bridges to native code which could be exploited to inject malicious logs, the security guideline by ENISA about protecting the back-end from such log injections is indeed related and supported.
- MASVS-RESILIENCE-2: The Mobile Application Security Verification Standard (MASVS) "RESILIENCE-2" and the ENISA guideline about protecting the back-end from client-initiated log injections both address the security concern of unauthorized modifications to the system, either through modified local versions of apps or corrupting/forging logs. While MASVS-RESILIENCE-2 focuses on ensuring the integrity of the application code and resources to prevent cheating or unauthorized premium access, the ENISA guideline emphasizes the need to safeguard the integrity of event histories on the back-end. Both controls aim to preserve the integrity of the system and protect against modifications that could otherwise compromise functionality or security.
- MASVS-RESILIENCE-3: Although MASVS-RESILIENCE-3 and the ENISA guideline seem to address different aspects at a first glance—with the former focusing on impeding static analysis to understand app internals and the latter on protecting the backend from client-initiated log injections—there is a correlation in the context of security principles. Both controls aim to increase security by reducing attack vectors. MASVS-RESILIENCE-3 does so by complicating the comprehension of the app's operation, which can mitigate reverse engineering and tampering efforts. Similarly, the ENISA guideline aims to protect the integrity of backend systems by preventing log forgery or corruption, which could be used to disguise malicious activities or aid in further attacks. Both measures contribute to a comprehensive security posture by protecting against different forms of exploitation of the app and its ecosystem.
- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" and the ENISA guideline about protecting the back-end from client-initiated log injections is that both are concerned with preventing sensitive data from being unintentionally exposed or stored in insecure locations. "MASVS-STORAGE-2" addresses the potential for sensitive data being unintentionally saved or leaked through mechanisms like APIs, backups, or logs, which could lead to unintentional exposure of sensitive data. The ENISA guideline highlights the importance of securing the back-end against client-initiated log injections that could alter or falsify the history of events, which is a form of log exposure that can also be considered under unintentional leaks when developers do not properly sanitize log inputs. Both controls emphasize the necessity for developers to actively manage and secure potentially exposed logs and data storage to avoid sensitive information leaks.

6.7 Implementation Guidance (ENISA 5.7):

ENISA Secure Smartphone Development Guidance (5.7): Employ rate limiting and throttling on a per-user/IP basis (if user identification is available) to reduce the risk from denial of service (DoS) attack.

6.7.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA guideline on employing rate limiting and throttling is that both are concerned with the security aspects of user authentication and the prevention of abuse in systems that are accessible remotely. MASVS-AUTH-1 emphasizes the importance of apps following best practices for secure protocol use in authentication and authorization, which would inherently include mechanisms to prevent abuse such as DoS attacks. Rate limiting and throttling, as suggested by the ENISA guideline, are specific best practices that protect against such abuse by limiting the number of requests a user or IP can make in a given timeframe, thus aligning with the intent of MASVS-AUTH-1 to promote secure app usage in authentication contexts.

6.8 Implementation Guidance (ENISA 5.8):

ENISA Secure Smartphone Development Guidance (5.8): Test for DoS vulnerabilities where the server may become overwhelmed by certain resource intensive application calls

6.8.1 OWASP MASVS MAPPING

- MASVS-PLATFORM-1: The correlation between "MASVS-PLATFORM-1," which deals with secure interactions involving Inter-Process Communication (IPC) mechanisms, and the ENISA guideline on testing for Denial of Service (DoS) vulnerabilities is that both involve ensuring the robustness and security of application operations that can affect or involve resources. IPC mechanisms, if not secured properly, can be exploited to perform unintended actions or to access sensitive data, which can lead to a DoS condition if an attacker overwhelms the IPC mechanisms with malicious requests. Thus, while MASVS-PLATFORM-1 is about securing IPC, it implicitly requires considering what would happen if these IPC mechanisms were abused, which includes testing for DoS vulnerabilities. Ensuring the security of IPC mechanisms is a proactive measure to mitigate potential resource-intensive calls that could lead to a DoS, fitting into the broader concept of maintaining application availability and resilience against such attacks as expressed in the ENISA guideline.

Chapter 7
Secure data integration with third party code

Third party code can represent both a security and a privacy liability. Third party code can use the application's access to user data and leak it on purpose or by accident. Similarly, third party code can introduce security vulnerabilities into an otherwise secure application. Application developers have to invest a minimum of time to vet any third party code they include in their application.

7.1 Implementation Guidance (ENISA 6.1):

ENISA Secure Smartphone Development Guidance (6.1): Vet the security/authenticity of third party code/libraries used in the mobile application. Ensure that third party code is only taken from a reliable source that maintains their code. (A) Audit code for security issues. (B) Audit the library and inspect any transmitted data to third-party services for privacy issues (e.g., analytics or ad libraries).

7.1.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA Guideline is that both emphasize the importance of security and proper vetting when it comes to code that handles user authentication, authorization, and third-party libraries. MASVS-AUTH-1 mentions the need to follow best practices for protocols related to authentication and authorization. In parallel, the ENISA guideline urges the vetting of third-party code and libraries for security and privacy issues, which are closely related to secure authentication and authorization practices. Both recognize the need to ensure that the app’s implementation does not introduce vulnerabilities through insecure use of protocols and third-party components.
- MASVS-CODE-3: The description of ”MASVS-CODE-3” correlates with the ENISA guideline. Both passages emphasize the importance of assessing the security of third-party components used in a mobile application. The MASVS-CODE-3 description discusses conducting a thorough assessment, acknowledging the practical limitations such as for third-party components, and focusing on easily identifiable (low-hanging fruit) vulnerabilities. The ENISA guideline recommends vetting the security and authenticity of third-party code, ensuring it comes from a reliable source, and auditing the code for security and privacy issues. Both sets of guidance aim to improve the security posture of an application by acknowledging the potential risks associated with third-party libraries.
- MASVS-CODE-4: The correlation between ”MASVS-CODE-4” and the ENISA Guideline is centered around the concept of treating incoming data and third-party code/libraries as potential security risks. ”MASVS-CODE-4” discusses the importance of verifying and sanitizing untrusted input from various data entry points to prevent security vulnerabilities such as injection attacks. Similarly, the ENISA Guideline emphasizes the need to vet the security and authenticity of third-party code/libraries to avoid security and privacy issues. Both statements concern the due diligence required to ensure that external inputs, whether they are data or code, do not compromise the security and privacy of the mobile application.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the ENISA Guideline is that both emphasize the importance of security in mobile applications, particularly related to the use of cryptography and third-party libraries. ”MASVS-CRYPTO-1” talks about general cryptography best practices, which includes ensuring that robust cryptographic measures are in place to protect user data, especially because physical access to the device is a risk factor in mobile environments. The ENISA Guideline focuses on vetting third-party code/libraries for security and authenticity, which often involves confirming that cryptographic protocols used by the libraries are secure and that data transmitted to third-party services do not compromise user privacy. In mobile app development, following cryptography best practices inherently includes auditing and ensuring the security of any external code or libraries integrated into the app.

- MASVS-NETWORK-1: The correlation is evident as both the ”MASVS-NETWORK-1” and the ENISA Guideline focus on ensuring the security and privacy of data in transit within mobile applications. ”MASVS-NETWORK-1” highlights the importance of encrypting data and authenticating remote endpoints to maintain data privacy and integrity over network communications, which may involve the use of third-party libraries. Similarly, the ENISA Guideline emphasizes the need to carefully vet third-party code/libraries for security and authenticity, ensuring that they come from reliable sources and do not compromise the app’s security or the users’ privacy. Both emphasize the need to audit and inspect code for security and privacy issues, indicating a clear correlation.
- MASVS-NETWORK-2: The correlation exists in the sense that both MASVS-NETWORK-2 and the ENISA Guideline emphasize the importance of ensuring the security and trustworthiness of third-party components involved in a mobile application’s operation. MASVS-NETWORK-2 refers specifically to the use of certificate pinning to ensure that communications are only trusted when they are signed with expected certificates, effectively reducing the risk of man-in-the-middle attacks due to compromised or rogue CAs. The ENISA Guideline addresses broader concerns with third-party code and libraries used in mobile applications, stressing that they should come from reliable sources and be audited for security and privacy issues. Both are aligned with the goal of minimizing the trust placed in external entities and maximizing the security of the application by closely managing and scrutinizing the sources of third-party components.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the described ENISA Guideline exists. Both are concerned with the security of interprocess communication (IPC) and the use of third-party code or libraries. ”MASVS-PLATFORM-1” is focused on ensuring secure interactions via IPC mechanisms provided by the platform, meaning applications must handle the data and functionality exposed through IPC securely. Similarly, the ENISA Guideline emphasizes the importance of vetting third-party code and libraries for security and privacy issues, bearing in mind that third-party components often interact with the main application through IPC mechanisms. Ensuring the security of third-party code and auditing for privacy issues contributes to the overall secure interaction with the app that ”MASVS-PLATFORM-1” aims to guarantee. Both highlight the importance of safeguarding the app from potential vulnerabilities that could arise from external interactions, whether from third-party code or IPC.
- MASVS-PLATFORM-2: The correlation exists because ”MASVS-PLATFORM-2” describes the need for secure configuration of WebViews, which can involve JavaScript bridges to native code, to prevent data leakage and exposure of sensitive functionality. This aligns well with the ENISA Guideline which emphasizes vetting the security and authenticity of third-party code/libraries and ensuring they come from a reliable source. Both guidelines focus on reviewing and securing code—either the app’s own code in the case of WebViews or third-party code/libraries—to prevent security and privacy issues.
- MASVS-PRIVACY-1: Both ”MASVS-PRIVACY-1” and the described ENISA Guideline emphasize the importance of vetting third-party code/libraries for security and privacy issues. Both suggest auditing and controlling the data accessed and transmitted by third-party services to ensure it aligns with necessary data minimization principles and informed user consent. This demonstrates a correlation in their objectives to enhance the privacy and security posture of mobile applications by controlling third-party code use and ensuring it comes from reputable sources with good maintenance practices.
- MASVS-PRIVACY-3: The correlation between ”MASVS-PRIVACY-3” and the ENISA Guideline is that both emphasize the importance of ensuring the privacy and security of user data. The MASVS-PRIVACY-3 highlights the need for clear information about data

practices and adherence to platform guidelines, which encompasses the concept of vetting third-party code/libraries for security and privacy as suggested by the ENISA Guideline. Both are concerned with how data is handled, especially with regards to unexpected or unauthorized access and usage, including by third-party services.

- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the ENISA guideline lies in the emphasis on ensuring the security of the environment in which the mobile application runs. While MASVS-RESILIENCE-1 specifically discusses the integrity of the platform (OS) and its trustworthiness to secure the app’s data through its built-in security features (like sandboxing), the ENISA guideline focuses on the reliability and security of third-party code/libraries utilized within the app. Both highlight the need for a secure and authentic execution environment and dependencies to protect the app from security vulnerabilities and preserve user privacy. Essentially, both are advocating for vigilance against tampering and ensuring the integrity of all components involved in the functioning of the mobile application.
- MASVS-RESILIENCE-2: Both MASVS-RESILIENCE-2 and the cited ENISA Guideline emphasize the importance of safeguarding the mobile application from unauthorized modifications and ensuring the integrity of the application’s code and resources. MASVS-RESILIENCE-2 focuses on preventing modifications to maintain the app’s intended functionality, while the ENISA Guideline underlines the need to vet third-party code/libraries for security and authenticity, which contributes to the same goal of preserving the original application’s integrity and preventing the inclusion of potentially harmful or backdoored code.
- MASVS-RESILIENCE-3: The correlation between ”MASVS-RESILIENCE-3” and the ENISA guideline is that both are concerned with protecting the mobile application from vulnerabilities that can be exploited through understanding or tampering with the app’s internals. MASVS-RESILIENCE-3 emphasizes making it difficult to analyze the app statically to hinder tampering attempts. In contrast, the ENISA guideline focuses on ensuring that any third-party code or libraries used in the app come from reliable sources and are audited for security and privacy issues. Both controls address the security of the app’s code, albeit from different perspectives—one from the angle of obfuscation to prevent understanding how the app works and the other from the angle of vetting and auditing third-party code to prevent the introduction of vulnerabilities and privacy risks.
- MASVS-STORAGE-1: The correlation exists in the context of protecting sensitive data. ”MASVS-STORAGE-1” emphasizes the need for apps to protect sensitive data regardless of where it is stored, ensuring that it remains secure whether it’s in private or public storage locations. The ENISA guideline complements this by recommending that security and privacy should be a primary consideration when using third-party code or libraries, which often handle or have access to sensitive data. Both standards highlight the importance of careful management and protection of sensitive information within mobile applications.

7.2 Implementation Guidance (ENISA 6.2):

ENISA Secure Smartphone Development Guidance (6.2): Track third party frameworks/APIs used in the mobile application for security patches. Integrate security updates for third party code/libraries/frameworks/APIs on a regular basis together with your own code. Ask the provider for a security report.

7.2.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation exists because both the MASVS-AUTH-1 requirement and the ENISA guideline emphasize the importance of secure interactions with remote endpoints. MASVS-AUTH-1 specifically mentions that apps must follow best practices for secure use of protocols in user authentication and authorization contexts, which would include being aware of security patches for any third-party frameworks or APIs that are used for such purposes. The ENISA guideline directly states that mobile applications should track and update third-party components for security, and asks for security reports from providers, which aligns with the intent of MASVS-AUTH-1 to ensure all components that participate in the authentication and authorization flow are secure and up-to-date. The overall aim is to strengthen the security posture of the mobile application by maintaining the integrity of both own code and third-party components involved in authentication and authorization mechanisms.
- MASVS-CODE-1: Both ”MASVS-CODE-1” and the ENISA Guideline emphasize the importance of using updated software to ensure security. ”MASVS-CODE-1” mentions that apps must run on up-to-date platform versions to have the latest security protections, which aligns with the ENISA guideline that stresses the need to track and regularly update third-party frameworks/APIs for security patches. Both are concerned with mitigating the risks associated with known vulnerabilities by maintaining current software versions.
- MASVS-CODE-2: Both ”MASVS-CODE-2” and the ENISA Guideline emphasize the importance of updating software to address security vulnerabilities. ”MASVS-CODE-2” is concerned with having a mechanism to force updates to the app when critical vulnerabilities are discovered, ensuring users are running the most secure version. The ENISA Guideline complements this by advising the tracking of third-party components for security patches and regular integration of these updates, along with requesting security reports from providers. Both controls aim to mitigate risks associated with outdated software, and thus, they correlate with each other in terms of maintaining the security of the application through updates.
- MASVS-CODE-3: The MASVS-CODE-3 description emphasizes the importance of a thorough security assessment even for third-party components, acknowledging that a full whitebox assessment may not always be possible, but at least known vulnerabilities should be checked for. The ENISA Guideline complements this by suggesting that developers track security patches for third-party frameworks/APIs and regularly integrate security updates, including requesting a security report from the provider. Both point towards a proactive approach in managing and securing third-party components included in the mobile application.
- MASVS-NETWORK-1: The MASVS-NETWORK-1 requirement about ensuring data privacy and integrity for data in transit aligns with the ENISA Guideline on tracking third-party frameworks/APIs for security patches. Both focus on maintaining secure communications

and the integrity of the app's network interactions. The MASVS-NETWORK-1 emphasizes securing connections, which can be undermined by insecure third-party components, while the ENISA Guideline suggests regular updates and security checks for third-party elements to mitigate such risks. Thus, they correlate as they both contribute to securing network communications within mobile applications.

- MASVS-NETWORK-2: The correlation between "MASVS-NETWORK-2" and the described ENISA Guideline is that both are concerned with enhancing the trust and security of the mobile application's networking environment. "MASVS-NETWORK-2," which focuses on certificate pinning, ensures that the app trusts only specific certificate authorities, thus reducing the risk of man-in-the-middle attacks and increasing network communication security. On the other hand, the ENISA guideline about tracking third-party frameworks and integrating security updates regularly is about maintaining the security of the application's dependencies. While certificate pinning is a more specific control, both the MASVS requirement and the ENISA guideline aim to protect against potential vulnerabilities introduced by external entities, thereby maintaining the integrity and trustworthiness of the app's network communication and dependencies.
- MASVS-PLATFORM-2: Both "MASVS-PLATFORM-2" and the ENISA Guideline emphasize on safeguarding the mobile application from potential security vulnerabilities associated with third-party components. While "MASVS-PLATFORM-2" addresses the need for secure configuration of WebViews, which could integrate third-party functionalities or frameworks, to prevent data leakage or exposure of sensitive functionalities, the ENISA Guideline also underlines the importance of tracking third-party frameworks/APIs and regularly updating them to patch security vulnerabilities. The underlying rationale in both cases is to maintain control over external code within the app to protect against security threats.
- MASVS-PRIVACY-1: The correlation exists in the emphasis on the responsible use of third-party SDKs or frameworks. "MASVS-PRIVACY-1" underlines the importance of judiciously using user data, requesting consent, and being aware of the data shared with third-party SDKs. It conveys the necessity for apps to not only restrict SDKs' access based on user consent but also to be mindful of their 'supply chain', assessing data passed to dependencies. The ENISA Guideline complements this by advocating that mobile applications keep track of third-party frameworks/APIs for security patches, integrate security updates regularly, and obtain security reports from the provider. Both stress the importance of managing third-party components to maintain security and privacy, thus supporting the end-to-end accountability of apps for their data practices.
- MASVS-PRIVACY-2: The Mobile Application Security Verification Standard (MASVS) privacy control "MASVS-PRIVACY-2" focuses on the importance of protecting user identity by employing data abstraction, anonymization, and pseudonymization to prevent user identification and tracking. It emphasizes the isolation of data streams to ensure they serve their intended purposes without compromising user privacy. This aligns with the ENISA guideline advising the tracking of third-party frameworks/APIs for security patches and regular integration of security updates. Both emphasize the importance of handling user data responsibly and maintaining privacy by preventing misuse of data through security measures, including regular updates and isolation of data usage for specified functions.
- MASVS-PRIVACY-3: Both MASVS-PRIVACY-3 and the specified ENISA guideline focus on the importance of transparency and security in the handling of user data. MASVS-PRIVACY-3 requires clear information to be provided to users about data handling practices, including unexpected data collection methods, which aligns with the intent of the ENISA guideline to track and update third-party frameworks/APIs for security purposes.

Continuous monitoring and integration of security updates from third-party providers are steps towards ensuring that user data is handled securely and in accordance with declared practices, which supports the principles of MASVS-PRIVACY-3.

- MASVS-RESILIENCE-3: The correlation between ”MASVS-RESILIENCE-3” and the ENISA guideline about tracking third-party frameworks/APIs for security patches arises from a shared focus on enhancing app security. MASVS-RESILIENCE-3 seeks to impede understanding of app internals to prevent tampering, while the ENISA guideline emphasizes the maintenance of third-party components for security. Both aim at protecting apps from being compromised—MASVS-RESILIENCE-3 by obfuscation to hinder reverse engineering, and the ENISA guideline by ensuring all components are up-to-date and secure. Both are proactive measures within a broader security strategy.

7.3 Implementation Guidance (ENISA 6.3):

ENISA Secure Smartphone Development Guidance (6.3): Pay attention to validate all data received from third parties before processing them within your application. This includes local applications, OS services as well as data received over the network.

7.3.1 OWASP MASVS MAPPING

- MASVS-AUTH-3: There is a correlation between "MASVS-AUTH-3" and the cited ENISA Guideline. The MASVS-AUTH-3 requirement talks about implementing additional forms of authentication securely, which implies careful consideration of security measures to protect sensitive actions within the app. This can involve validating and securing data received from biometric sensors, pin input, email, or deep links, which are third-party sources. The ENISA Guideline specifically mentions the importance of validating all data received from third parties before processing, which is an integral part of implementing those additional authentication methods securely. Both guidelines emphasize the security of the app in contexts that could involve data from external sources.
- MASVS-CODE-1: While MASVS-CODE-1 and the ENISA guideline are addressing different specific aspects of mobile app security, they share a correlation in the broader context of maintaining the security of the app and protecting it from known vulnerabilities. MASVS-CODE-1 emphasizes the importance of the app running on an up-to-date platform version to benefit from the latest security protections, whereas the ENISA guideline highlights the necessity of validating all data received from third parties to prevent untrusted inputs from compromising the app. Both contribute to the overall security posture by ensuring that the app is protected against known threats, either by leveraging the latest platform defenses or by actively validating external data to avoid exploitation.
- MASVS-CODE-3: Both the MASVS-CODE-3 reference and the ENISA Guideline emphasize the security considerations when dealing with third-party components or data received from third parties. While MASVS-CODE-3 discusses the importance of white-box assessments and acknowledges the practical limitations often present, leading to a focus on scanning for known vulnerabilities in libraries, the ENISA Guideline stresses the importance of validating all third-party data before processing. Both points converge on the theme of the potential risks associated with third-party code and the necessary precautions to mitigate these risks, highlighting the importance of due diligence against vulnerabilities in the context of app security.
- MASVS-CODE-4: The Mobile Application Security Verification Standard (MASVS) code MASVS-CODE-4 description and the ENISA guideline both emphasize the importance of treating data from various entry points as untrusted and ensuring that this data is validated, verified, and sanitized before processing. Both address the risks associated with the input of data that could be manipulated by external, untrusted sources. The MASVS-CODE-4 specifically mentions multiple data entry points and the potential security vulnerabilities they present, such as injection attacks, which is in line with the ENISA guideline's emphasis on validating all third-party data. Therefore, there is a clear correlation between MASVS-CODE-4 and the ENISA guideline, as they both aim to mitigate security risks through proper data validation practices.
- MASVS-NETWORK-1: Both "MASVS-NETWORK-1" and the ENISA Guideline emphasize the importance of securing data and validating it to protect against integrity and

privacy issues. ”MASVS-NETWORK-1” focuses on ensuring the secure transmission of data over the network by using encrypted connections and proper endpoint authentication, while the ENISA Guideline highlights the necessity of validating all third-party data before processing. Although the two are not identical, they both contribute to the broader goal of safeguarding data integrity and privacy within an application, which includes preventing unauthorized access and manipulation of data as it is communicated and received. The correlation is that both principles are part of a comprehensive security practice that involves protecting data at different stages - MASVS-NETWORK-1 focuses on data in transit, and the ENISA Guideline on data being processed.

- MASVS-NETWORK-2: The correlation between ”MASVS-NETWORK-2” and the ENISA guideline about validating data from third parties lies in the concept of trust and validation of external entities. ”MASVS-NETWORK-2” focuses on limiting trust to specific Certificate Authorities (CAs) through the practice of certificate pinning, which is a form of trust validation for TLS connections ensuring the application communicates only with the intended server(s). The ENISA guideline emphasizes the broader principle of validating all data received from third parties, which includes not just network data but any data that originates outside the application. Both are preventative measures against various security threats such as man-in-the-middle (MITM) attacks, where an attacker could present a false certificate or send malicious data. By pinning certificates or public keys, and validating third-party data, an app can enhance its security posture by ensuring it only accepts trusted connections and data, which is aligned with the principle of data validation described by both MASVS and ENISA.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the ENISA guideline is that both emphasize the importance of secure interactions and the validation of data received from external sources. ”MASVS-PLATFORM-1” focuses on secure interactions involving IPC (Inter-Process Communication) mechanisms, which can include communications with other local applications and OS services. The ENISA guideline advises on validating all data received from third parties, which encompasses both local and network sources. Both are concerned with ensuring that external data does not lead to security vulnerabilities when processed by the application.
- MASVS-PLATFORM-2: The correlation between ”MASVS-PLATFORM-2” and the ENISA Guideline is that both emphasize the importance of securing sensitive data and functionality. MASVS-PLATFORM-2 specifically mentions secure configuration of WebViews to prevent data leakage and exposure of sensitive functionality, which often includes validating and sanitizing data received, especially when integrating JavaScript bridges that allow communication between web-based and native components. The ENISA Guideline similarly stresses the need to validate all data received from third parties before processing, in order to prevent possible security issues such as data corruption, leakage, or malicious code execution. Both are concerned with ensuring the security and integrity of the data being handled by the application, which directly relates to preventing vulnerabilities that could be exploited through the misuse of WebViews or other third-party data.
- MASVS-PRIVACY-1: The correlation between ”MASVS-PRIVACY-1” and the ENISA Guideline is that both emphasize the care and consideration that should be given to data sourced from third parties. ”MASVS-PRIVACY-1” discusses the importance of user consent and regulating third-party SDKs’ data access, which involves ensuring that data is not collected without consent and that no unnecessary data is shared along the chain of dependencies. Similarly, the ENISA Guideline calls for the validation of all third-party data before it is processed by the application. Both guidelines aim to protect user data and ensure that applications handle it responsibly.

- MASVS-RESILIENCE-1: Both ”MASVS-RESILIENCE-1” and the ENISA Guideline focus on the principle of not trusting the operating environment by default. ”MASVS-RESILIENCE-1” emphasizes the importance of running on a secure platform and warns against tampering that may compromise security features, which would put the app’s data at risk. Similarly, the ENISA guideline advises on validating all data from third parties before processing to avoid being compromised. Although MASVS-RESILIENCE-1 deals more with the state of the platform itself while ENISA’s guidance pertains to third-party data validation, both stress the importance of ensuring that the operating environment (which includes the platform and data from third-parties) is secure and not tampered with. This overlaps in the broader context of security and trust in one’s operating environment.
- MASVS-RESILIENCE-2: Both the Mobile Application Security Verification Standard (MASVS) ”RESILIENCE-2” control and the ENISA guideline highlight the importance of maintaining the integrity and security of the app’s functionality in the face of potential tampering and third-party data sources. MASVS-RESILIENCE-2 emphasizes protecting the application against modifications, ensuring that the original code and resources are not altered improperly. This is directly related to the ENISA guideline’s recommendation to validate all third-party data before it is processed by the app, as such validation can prevent malformed or malicious data from causing harm or being used to exploit the app, which is a form of modification or tampering. Both are addressing the threats posed by running on user-controlled devices and the need for protective measures against code and data manipulation.
- MASVS-RESILIENCE-3: The correlation between ”MASVS-RESILIENCE-3” and the ENISA guideline regarding validating all data received from third parties is centered around the goal of increasing an app’s security and resilience against tampering and malicious exploitation. While ”MASVS-RESILIENCE-3” focuses on making it difficult to comprehend and manipulate an app through static analysis, the ENISA guideline emphasizes the importance of validating external data to prevent attacks that could exploit data handling vulnerabilities within the app. Both controls aim to protect the integrity and security of the app from different angles. ”MASVS-RESILIENCE-3” targets protection against reverse engineering and code tampering, while the ENISA guideline targets protection against data-based attacks. Together, they contribute to a comprehensive security posture.
- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the ENISA guideline is that both emphasize the importance of properly handling and protecting sensitive data. MASVS-STORAGE-1 focuses on ensuring that sensitive data, whether it comes from various sources and is stored locally by the app, is protected in both private and public storage locations. The ENISA guideline underlines the necessity to validate all data received from third parties before processing, which includes both local and external sources. Validation is a part of protecting sensitive data by ensuring its integrity and security before it is stored or processed by the application. Therefore, these concepts are related as they both deal with safeguarding sensitive data from possible vulnerabilities or exposures.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA Guideline about validating data from third parties is that both are focused on the protection of sensitive data within an application. ”MASVS-STORAGE-2” addresses the risk of unintentional data leaks through public storage or logs, which can occur if a developer does not properly handle APIs or system capabilities. Similarly, the ENISA Guideline emphasizes the need to validate data from external sources to prevent malicious content from entering the application and causing harm, which includes the risk of sensitive data exposure. Both controls are concerned with implementing appropriate measures to ensure that sensitive data is not inadvertently compromised.

7.4 Implementation Guidance (ENISA 6.4):

ENISA Secure Smartphone Development Guidance (6.4): Avoid using third-party libraries that contain main processor-only cryptographic implementations. Prefer using cryptographic framework provided by a platform supported secure hardware (e.g. TEE, SE).

7.4.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between "MASVS-AUTH-1" and the ENISA Guideline about avoiding third-party libraries with main processor-only cryptographic implementations lies in the emphasis on security best practices. MASVS-AUTH-1 highlights the necessity for applications to follow security best practices for authentication and authorization protocols. Part of these best practices would include the use of secure and vetted cryptographic libraries. The ENISA Guideline recommends using cryptographic frameworks supported by secure hardware to enhance security further. Both guidelines aim to ensure the secure handling of authentication, authorization, and encryption mechanisms within the app to protect against vulnerabilities and attacks. Using platform-supported secure hardware aligns with following relevant best practices for security, as prescribed by MASVS-AUTH-1.
- MASVS-AUTH-2: The correlation between "MASVS-AUTH-2" and the ENISA Guideline is present as both are concerned with the implementation of secure authentication mechanisms. MASVS-AUTH-2 emphasizes the need for correct implementation of biometric and local PIN code authentication, which could leverage secure hardware such as TEE (Trusted Execution Environment) or SE (Secure Element), as suggested by the ENISA Guideline. The guideline's recommendation to avoid using main processor-only cryptographic implementations and to prefer cryptographic frameworks supported by secure hardware aligns with ensuring the robustness of local authentication methods mentioned in MASVS-AUTH-2. Secure hardware components can offer a more secure environment for storing and processing sensitive authentication data, which is an integral part of correctly implementing such features in mobile apps.
- MASVS-AUTH-3: Both "MASVS-AUTH-3" and the ENISA guideline emphasize the importance of securely implementing authentication mechanisms, particularly for sensitive actions. "MASVS-AUTH-3" suggests that additional forms of authentication are often desirable, and these should be implemented securely, which implies they should be robust against various attack vectors, including those that might exploit weaknesses in third-party libraries. The ENISA guideline advises against using third-party libraries for cryptographic implementations that only work on the main processor and suggests using platform-supported secure hardware, such as Trusted Execution Environments (TEE) or Secure Elements (SE), which provide a more secure environment for sensitive operations like authentication. Both are concerned with enhancing the security of authentication by using secure and trustworthy components.
- MASVS-CODE-3: The correlation between "MASVS-CODE-3" and the ENISA guideline focuses on the attention to the use of third-party components within the mobile application. While MASVS-CODE-3 addresses the security considerations by suggesting a whitebox assessment to find "low-hanging fruit" cases, such as scanning for known vulnerabilities in libraries, the ENISA guideline specifically advises against using third-party libraries with

certain cryptographic implementations. Both pieces of guidance converge on the principle that third-party components can introduce security risks, and both suggest favoring more secure or vetted options—MASVS-CODE-3 through security testing and the ENISA guideline through the preference of secure hardware-supported cryptographic frameworks over third-party libraries.

- MASVS-CRYPTO-1: The correlation between "MASVS-CRYPTO-1" and the ENISA Guideline is that both emphasize the importance of robust and secure cryptographic practices, particularly in mobile environments where physical access to a device is a realistic threat. MASVS-CRYPTO-1 refers to the use of general cryptography best practices, and the ENISA Guideline advises against using third-party libraries with main processor-only implementations in favor of frameworks that leverage secure hardware elements like TEE or SE. Both directives aim to enhance the security of user data by recommending the use of the most secure and reliable cryptographic techniques available, which includes taking advantage of hardware-based security features where possible.
- MASVS-CRYPTO-2: Both MASVS-CRYPTO-2 and the ENISA Guideline emphasize the importance of secure handling of cryptographic practices and components. MASVS-CRYPTO-2 specifically addresses the need for proper management of cryptographic keys throughout their lifecycle, which would include considerations of how keys are stored and protected. The ENISA guideline dovetails with this by recommending the avoidance of third-party libraries that are limited to main processor-only implementations for cryptography, and instead, suggesting the use of platform-supported secure hardware like Trusted Execution Environments (TEE) or Secure Elements (SE). The correlation lies in the principle that both guidelines seek to enhance security by using hardware-based solutions to protect cryptographic keys, rather than relying solely on software-based solutions which might be more vulnerable to attack. Secure hardware environments can provide additional layers of security for storing and handling cryptographic keys, thus aligning with the key management objectives outlined in MASVS-CRYPTO-2.
- MASVS-NETWORK-1: Both the MASVS-NETWORK-1 requirement and the ENISA guideline emphasize the importance of using secure and trusted methods to protect data during transit over the network. MASVS-NETWORK-1 underlines the significance of ensuring data privacy and integrity by encrypting data and properly authenticating endpoints, cautioning against disabling secure defaults or bypassing them with low-level APIs or untrusted third-party libraries. Similarly, the ENISA guideline advises against using third-party libraries with cryptographic implementations that only operate on the main processor and recommends relying on platform-supported cryptographic frameworks that leverage secure hardware, such as Trusted Execution Environments (TEEs), which offer a higher level of security. Both stress the risks involved with untrusted third-party libraries, therefore showing a clear correlation between the two security advices.
- MASVS-PLATFORM-1: The correlation exists in that "MASVS-PLATFORM-1" refers to securely using platform-provided IPC (Inter-Process Communication) mechanisms, which implies relying on secure platform features, while the ENISA Guideline recommends using cryptographic frameworks supported by platform secure hardware. Both highlight the importance of leveraging the security features provided by the platform—IPC mechanisms in MASVS-PLATFORM-1 for secure interactions, and cryptographic frameworks in hardware like TEE or SE for secure cryptographic implementations as per the ENISA Guideline. They are correlated in their advocacy for using platform-provided security features to ensure better security in app development.
- MASVS-PRIVACY-1: There is a correlation between "MASVS-PRIVACY-1" and the ENISA Guideline as both emphasize the importance of apps being conscientious about

the third-party code they incorporate. MASVS-PRIVACY-1 advises that third-party SDKs should operate based on user consent, and apps must be vigilant about the data these SDKs could access or share. Similarly, the ENISA Guideline warns against using third-party libraries for main processor-only cryptographic implementations, encouraging the use of secure hardware-supported frameworks instead. Both guidelines aim to minimize the privacy and security risks associated with third-party code by advocating for more secure and user-consent-based practices.

- MASVS-RESILIENCE-1: There is a correlation between MASVS-RESILIENCE-1 and the ENISA Guideline mentioned. MASVS-RESILIENCE-1 concerns the importance of running on a secure, uncompromised platform for the proper functioning of various security features such as secure storage and sandboxing. ENISA's advice to avoid using third-party libraries for cryptographic implementations in favor of platform-supported secure hardware like TEE (Trusted Execution Environment) or SE (Secure Element) aligns with this principle, as relying on platform hardware features generally suggests that the platform's security features are trusted and intact. Both highlight the importance of ensuring the secure state of the platform to maintain the security posture of the application.
- MASVS-RESILIENCE-2: Both the MASVS-RESILIENCE-2 guideline and the ENISA Guideline pertain to ensuring the security and integrity of the mobile application. While MASVS-RESILIENCE-2 focuses on the general concept of protecting the app's original code and resources to prevent unauthorized modifications and maintaining the integrity of the app's functionality, the ENISA Guideline specifically addresses the need to use cryptographic frameworks that are supported by secure hardware to enhance the security of cryptographic operations. Both guidelines aim to defend against potential security threats that may arise from user-controlled devices or incorporation of insecure third-party components. By avoiding main processor-only cryptographic implementations and opting for platform-supported secure hardware, the integrity of the app's functionality, as outlined in MASVS-RESILIENCE-2, is further enhanced by making cryptographic breaches more difficult, thus providing a form of resilience against code and resource modifications.
- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" and the ENISA Guideline regarding the use of third-party libraries for cryptography lies in the focus on preventing unintentional exposure of sensitive data. MASVS-STORAGE-2 emphasizes the importance of ensuring that sensitive data is not accidentally stored or made publicly accessible, often due to misuse of APIs or system features. The ENISA Guideline advises against using third-party libraries that lack secure hardware support to prevent the main processor from being the only point of cryptographic operations, which may lead to potential vulnerabilities and exposure of sensitive data. Both emphasize the careful handling of sensitive data and the use of secure methods and hardware to prevent data leaks, aligning with the goal of protecting sensitive information from unintentional disclosure.

7.5 Implementation Guidance (ENISA 6.5):

ENISA Secure Smartphone Development Guidance (6.5): Software components that are no longer supported by the vendor or developer must not be used

7.5.1 OWASP MASVS MAPPING

- MASVS-CODE-1: Both ”MASVS-CODE-1” and the ENISA Guideline emphasize the importance of keeping software up to date to leverage the most recent security patches and features. ”MASVS-CODE-1” addresses the necessity for mobile apps to run on updated operating system versions to ensure the latest security protections are in place. The ENISA Guideline advises against using software components that are no longer supported by the vendor or developer, which also implies that using unsupported, and thus potentially outdated, software can expose users to known vulnerabilities. The correlation is that both guidelines aim to mitigate security risks by avoiding the use of outdated software that may contain unpatched security issues.
- MASVS-CODE-2: The correlation between ”MASVS-CODE-2” and the ENISA guideline regarding the use of unsupported software components is based on the principle of ensuring that applications remain secure throughout their lifecycle. ”MASVS-CODE-2” focuses on the importance of a mechanism that enforces app updates, particularly when critical vulnerabilities are identified. This includes scenarios where a component may no longer be supported and thus requires an update to a newer, supported version to resolve vulnerabilities. The ENISA guideline explicitly states that unsupported software components must not be used, directly complementing the MASVS-CODE-2 requirement by implying that continuous updates may be necessary to avoid the use of unsupported, vulnerable components. Both aim to minimize security risks by ensuring the software is up-to-date and supported.
- MASVS-CODE-3: The description of ”MASVS-CODE-3” implies that although a full whitebox assessment for all app components (including third-party ones) is ideal for security, it is not always feasible. This correlates with the ENISA Guideline, which states that unsupported software components must not be used. The ”low-hanging fruit” mentioned in MASVS-CODE-3 refers to easily detectable vulnerabilities, which would include those found in unsupported software components since they no longer receive updates or patches for known issues. Thus, there is a correlation as both advocate for the avoidance or extra scrutiny of components that could pose security risks due to lack of vendor support.
- MASVS-NETWORK-1: Both the MASVS-NETWORK-1 requirement and the ENISA guideline emphasize the importance of security best practices in maintaining the integrity and privacy of data. MASVS-NETWORK-1 specifically focuses on the importance of secure network connections, which include using up-to-date encryption methods and properly authenticating remote endpoints. The ENISA guideline complements this by stating that software components that are no longer supported—or outdated—should not be used, as this could introduce vulnerabilities and potentially compromise secure network connections. Therefore, the guideline indirectly supports the MASVS-NETWORK-1 requirement by ensuring that all network-related software components are up-to-date and supported, thus helping to ensure that secure defaults are maintained and cannot be bypassed using such outdated or unsupported software.

- MASVS-PRIVACY-1: Both ”MASVS-PRIVACY-1” and the ENISA guideline regarding unsupported software components share a common goal of protecting user data and enhancing security through good data management and secure software practices. ”MASVS-PRIVACY-1” emphasizes the need for apps to request only necessary data with informed consent, which implicitly means relying on secure, trustworthy components—including SDKs—that respect user consent. Utilizing outdated or unsupported software components, as cautioned against by ENISA, can lead to vulnerabilities that would undermine the data minimization and secure access principles advocated by ”MASVS-PRIVACY-1.” Unsupported components may not respect user consent mechanisms or could introduce risks by failing to patch known issues, thereby creating potential for unauthorized data access or breaches. Both guidelines advocate for using up-to-date and secure components to maintain user privacy and app security.

Chapter 8
Consent and privacy protection

Mobile applications often store and operate on personal information, therefore, they need to be designed to prevent unintentional disclosure of personal or private information. Developers have to pay specific attention to obtain consent to, any data collection, sharing and usage that takes place on the application.

8.1 Implementation Guidance (ENISA 7.1):

ENISA Secure Smartphone Development Guidance (7.1): Check whether your application is collecting personal data. It may not always be obvious - for example, do you use persistent unique identifiers linked to central data stores containing personal information?

8.1.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: MASVS-AUTH-1 is concerned with user authentication and the enforcement of authorization mechanisms in mobile applications, which often involves the handling of personal data, particularly in the context of identifying and authorizing users. The ENISA guideline about checking whether an application is collecting personal data, including persistent unique identifiers linked to central data stores containing personal information, relates directly to the mechanisms mentioned in MASVS-AUTH-1. Both the MASVS requirement and the ENISA guideline are focused on the secure management of user identities and the personal data associated with them, and thus show a clear correlation.
- MASVS-AUTH-2: MASVS-AUTH-2 emphasizes the importance of correctly implementing biometrics or a local PIN code for user authentication. If the authentication data, such as biometric data or PINs, are stored locally or transmitted, they could be considered personal data. The ENISA guideline is concerned with whether an application collects personal data, which includes unique identifiers that could be linked to personal information. Since the implementation of biometric and PIN authentication would typically involve handling sensitive, unique data linked to an individual, there is a correlation as both the MASVS standard and the ENISA guideline address the management, protection, and potential collection of personal data by the application.
- MASVS-CODE-4: Both the Mobile Application Security Verification Standard (MASVS) CODE-4 and the ENISA Guideline about personal data collection are concerned with the management of incoming data and its security implications in mobile applications. MASVS-CODE-4 emphasizes the importance of treating all incoming data as untrusted and ensuring it is properly verified and sanitized to prevent injection attacks and other security vulnerabilities. Similarly, the ENISA Guideline advises checking if the application collects personal data, which also relates to the handling of incoming data, particularly how it is collected, stored, and linked, especially considering privacy concerns. Both guidelines highlight the need for strict data handling procedures to maintain the security and privacy of user data.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and its description, and the ENISA Guideline about checking for personal data collection is present in the emphasis both place on data protection, especially in mobile environments where devices can be more easily compromised due to physical access. ”MASVS-CRYPTO-1” highlights the importance of implementing strong cryptography practices as a means to protect user data which could include personal information. This is related to the ENISA guideline that underlines the importance of understanding whether an application collects personal data, implicitly suggesting that wherever personal data is collected, measures should be in place to protect that data, potentially through the use of cryptography. Both suggest a proactive approach to data protection, whereby recognizing the types of data being handled (like personal data) is crucial for its secure management.

- MASVS-PLATFORM-1: MASVS-PLATFORM-1 involves ensuring secure interactions with inter-process communication (IPC) mechanisms provided by the platform, which includes the secure handling of intentionally exposed data and functionality. This correlates with the ENISA Guideline's emphasis on being vigilant about personal data collection through the app, including the use of persistent unique identifiers. Since IPC mechanisms could potentially involve the exchange or accessibility of personal data, both MASVS-PLATFORM-1 and the ENISA Guideline are concerned with the secure management and oversight of data handling within the app to prevent unintended access or leaks of personal information.
- MASVS-PLATFORM-3: Both MASVS-PLATFORM-3 and the ENISA guideline emphasize the importance of handling sensitive data with care. MASVS-PLATFORM-3 is specifically concerned with the unintentional leakage of sensitive data that may be displayed in the UI due to platform mechanisms, while the ENISA guideline brings attention to the collection of personal data and its association with persistent unique identifiers, which could lead to privacy breaches. The correlation lies in the broader context of data privacy and security—both are guiding towards practices to prevent disclosure of personal information. MASVS-PLATFORM-3's concerns about sensitive data leaks through screenshots or shoulder surfing are related to the ENISA's caution about not always obvious ways of collecting personal data. Both advocate for proactive measures to safeguard users' personal and sensitive information.
- MASVS-PRIVACY-1: The correlation between "MASVS-PRIVACY-1" and the ENISA guideline is evident in their mutual emphasis on data minimization and informed consent regarding personal data collection. Both stress the importance of limiting data access to only what is necessary for app functionality, and they highlight the responsibility of app developers to ensure consent is obtained before collecting any personal data. The ENISA guideline specifically mentions checking for the collection of personal data, even in non-obvious forms such as persistent unique identifiers, which directly aligns with the MASVS-PRIVACY-1 mandate for apps to restrict access control and share data with third parties only with necessary user consent. Additionally, both advocate for transparency and accountability in the use of third-party SDKs and data practices, aligning with broader regulatory requirements.
- MASVS-PRIVACY-2: The Mobile Application Security Verification Standard's (MASVS) control "MASVS-PRIVACY-2" and the ENISA Guideline regarding the collection of personal data are correlated. The MASVS control emphasizes on the importance of using methods like data abstraction, anonymization, and pseudonymization to prevent user identification and tracking, focusing on separating out user identity, especially with complex identifiers like device IDs or IP addresses. This aligns with ENISA's advice to be aware of whether personal data is being collected, subtly or overtly, and to be particularly cautious with the use of persistent unique identifiers that could be linked to personal information. Both guidelines prioritize user privacy and the responsible handling of data that could potentially identify individual users.
- MASVS-PRIVACY-3: Both "MASVS-PRIVACY-3" and the ENISA Guideline emphasize the importance of transparency in data collection practices by applications. "MASVS-PRIVACY-3" specifically states that users should be informed about data collection, storage, and sharing practices, which resonates with the ENISA Guideline's requirement to check if the application collects personal data, including not-so-obvious methods like using persistent unique identifiers. Both guidelines advocate for user awareness and adherence to data collection policies, indicating a correlation between them.

- MASVS-PRIVACY-4: The correlation between ”MASVS-PRIVACY-4” which emphasizes user control over their data, providing mechanisms for users to manage, delete, and modify their data, along with changing privacy settings and re-prompting for consent when more data is needed, aligns with the ENISA Guideline which focuses on checking if the application is collecting personal data through persistent unique identifiers linked to personal information. Both address the significance of understanding data collection practices, user consent, and management of personal data, highlighting the importance of privacy and control in the context of user data.
- MASVS-STORAGE-1: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-STORAGE-1” and the ENISA guideline both emphasize the importance of properly handling and protecting sensitive data that may be collected and stored by a mobile application. MASVS-STORAGE-1 focuses on ensuring that sensitive data stored locally by the app is protected, regardless of whether the storage location is private to the app or public. The ENISA guideline advises checking whether the application collects personal data and mentions the use of persistent unique identifiers linked to central data stores containing personal information, which is a form of sensitive data. Both sources are concerned with the security implications of collecting and storing sensitive information and advocate for protective measures.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA Guideline about collecting personal data is that both address the concern of unintentionally exposing or storing sensitive data where it can be accessed publicly or by unauthorized parties. MASVS-STORAGE-2 specifically mentions the unintentional leaks that can occur through the use of certain APIs and system capabilities, which aligns with the ENISA Guideline’s point about being aware of personal data collection, even through indirect means such as persistent unique identifiers. Both guidelines emphasize the importance of developer awareness and proactive measures to prevent the exposure of sensitive information.

8.2 Implementation Guidance (ENISA 7.2):

ENISA Secure Smartphone Development Guidance (7.2): Create a privacy policy covering the usage of personal data and make it available to the user especially prior to making consent choices.

8.2.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: The correlation between ”MASVS-AUTH-2” and the ENISA Guideline can be understood by considering the concerns of both guidelines. MASVS-AUTH-2 focuses on the correct implementation of authentication mechanisms, which may include biometric data or PIN codes. Biometric data is considered a form of personal data, as it uniquely identifies an individual. Proper implementation of such mechanisms would entail ensuring confidentiality, integrity, and appropriate handling of this sensitive data. The ENISA Guideline emphasizes creating a privacy policy that covers the usage of personal data. In the context of mobile apps with authentication via biometrics or a local PIN code, a privacy policy is crucial to inform users about how their biometric data or information is utilized and protected. Such a privacy policy should be made available especially before users make consent decisions, implying that users should be aware of how their personal data will be handled prior to utilizing the app’s authentication features. While MASVS-AUTH-2 is more focused on the security aspect of authentication, and the ENISA Guideline is focused on privacy and user consent, they correlate in the sense that both advocate for the responsible handling of personal data, which includes informing users about how their data is used and ensuring secure implementation of data processing mechanisms.
- MASVS-AUTH-3: Both ”MASVS-AUTH-3” and the ENISA guideline address the aspect of protecting sensitive user activities and data. ”MASVS-AUTH-3” suggests using additional forms of authentication to secure sensitive actions in the app, while the ENISA guideline emphasizes creating a privacy policy detailing the usage of personal data, which should be available to the user prior to giving consent. Both relate to enhancing user security and privacy within the app’s context.
- MASVS-CRYPTO-1: While ”MASVS-CRYPTO-1” directly refers to the implementation of cryptographic practices to ensure data security, which is a technical measure, the ENISA guideline emphasizes the importance of having a privacy policy that covers the usage of personal data, which is more about informing the user and providing transparency. The correlation exists in the broader context of data protection and user privacy. Cryptography is a method to protect user data, which aligns with the spirit of the ENISA guideline to ensure that users are aware of and consent to how their personal data is being used. Implementing proper cryptography helps to protect the privacy of user data, which is in line with the objectives of a privacy policy aimed at safeguarding personal information.
- MASVS-NETWORK-1: The correlation between ”MASVS-NETWORK-1” and the ENISA Guideline about creating a privacy policy is that both are concerned with the privacy and protection of user data. ”MASVS-NETWORK-1” focuses on ensuring data privacy and integrity for data in transit, primarily through encryption and secure connections to protect the data from interception or tampering. On the other hand, the ENISA Guideline emphasizes transparency and user awareness regarding the use of personal data through a privacy policy. While MASVS-NETWORK-1 deals with the technical aspects of data protection,

the ENISA Guideline addresses the legal and informative aspects. Both contribute to an overall framework of user data protection and privacy.

- MASVS-PLATFORM-1: While the MASVS-PLATFORM-1 requirement from the Mobile Application Security Verification Standard (MASVS) does not explicitly mention privacy policies or the usage of personal data, there is an indirect correlation with the ENISA guideline on creating a privacy policy for personal data usage. This is because secure interaction with Inter-Process Communication (IPC) mechanisms, as stated in the MASVS-PLATFORM-1, is crucial for protecting sensitive data, including personal data, from being exposed to unauthorized parties. The ENISA guideline highlights the importance of informing users about the privacy practices concerning their personal data, which is an aspect of creating a secure and trustworthy app environment. Secure IPC mechanisms contribute to the overarching goal of data protection and privacy, which includes transparency about data usage as suggested by ENISA. Together, they both aim to enhance user trust and security in mobile applications.
- MASVS-PLATFORM-2: While "MASVS-PLATFORM-2" specifically addresses the secure configuration of WebViews to prevent data leakage and exposure of sensitive functionality, it implicitly relates to the overarching goal of protecting user privacy, which is the focus of the ENISA guideline. By ensuring WebViews are configured securely, an app can help safeguard personal data from unauthorized access or leaks, which is a vital aspect of any privacy policy that the ENISA guideline mandates should be available to users. Hence, there is a correlation in that both aim to enhance the protection of users' personal data, albeit through different means - technical measures for WebViews in the case of MASVS-PLATFORM-2 and transparency and user consent in the case of the ENISA guideline.
- MASVS-PRIVACY-1: The "MASVS-PRIVACY-1" requirement is closely correlated with the ENISA guideline regarding the creation of a privacy policy covering the usage of personal data. Both stipulate the importance of informed user consent and transparency in the handling of user data. The MASVS-PRIVACY-1 emphasizes on data minimization, consent-based data access, responsible third-party SDK data handling, and awareness of data practices throughout the app's SDK supply chain. Similarly, the ENISA guideline insists on having a privacy policy that is available to users, especially before they make consent decisions, ensuring they are informed about how their personal data will be used. Both are targeted at protecting user privacy and align on the basis of requiring apps to be clear and conscientious about how they access and use personal data.
- MASVS-PRIVACY-2: MASVS-PRIVACY-2 and the ENISA guideline both focus on the protection of user privacy and identity, promoting transparency and user control over personal data. MASVS-PRIVACY-2 emphasizes techniques like anonymization and pseudonymization to enhance privacy and prevent user tracking, which aligns with the creation of a privacy policy as mandated by the ENISA guideline. Both aim to ensure users are informed and have a choice regarding their personal data, reinforcing the principle of user consent.
- MASVS-PRIVACY-3: The correlation exists because both MASVS-PRIVACY-3 and the ENISA guideline highlight the importance of informing users about data usage practices. MASVS-PRIVACY-3 stipulates that apps should provide clear details on how the data is used, which aligns with the ENISA principle that there should be a privacy policy detailing the use of personal data available to users, especially before they need to make consent choices. Both emphasize transparency and user awareness regarding data handling by the application.

- MASVS-PRIVACY-4: Both ”MASVS-PRIVACY-4” and the ENISA guideline emphasize the importance of user control and consent over their personal data. MASVS-PRIVACY-4 states the importance for users to manage their data and privacy settings, including the ability to revoke consent and to be prompted again for consent if the app’s data requirements change. Similarly, the ENISA guideline insists on the availability of a privacy policy that covers data usage, which should be accessible to users prior to consenting. Both guidelines focus on ensuring that users are informed and have agency over their personal information within the app.
- MASVS-STORAGE-1: The ”MASVS-STORAGE-1” requirement addresses the need for secure handling and storage of sensitive data within a mobile application. The correlation with the ENISA guideline about creating a privacy policy is that both are focused on protecting user privacy and personal data. The MASVS-STORAGE-1 control ensures technical measures are in place to protect data at rest, while the ENISA guideline ensures users are informed about how their personal data is used, which can include details about data storage practices. Both aim to enhance the privacy and security of user data.
- MASVS-STORAGE-2: The ”MASVS-STORAGE-2” control is related to the prevention of unintentional storage or exposure of sensitive data in publicly accessible locations. This directly correlates with the ENISA guideline that advices the creation of a privacy policy covering the usage of personal data, as both aim to protect sensitive information. Implementing ”MASVS-STORAGE-2” would involve ensuring that personal data is not unintentionally stored or exposed, thereby upholding the privacy standards mentioned in the ENISA guideline, which focuses on transparently communicating how personal data is used to the users.

8.3 Implementation Guidance (ENISA 7.3):

ENISA Secure Smartphone Development Guidance (7.3): Prior to using personal data consent should be obtained. When obtaining consent, explicitly notify the user with specific information such as: (A) what exactly personal data will be used; (B) what is the purpose of the processing; (C) who are the recipients of the data; (D) where is the data stored and for how long. In case that the user does not grant consent to all requested data, he/she should be informed about possible limitations of the app's functionality.

8.3.1 OWASP MASVS MAPPING

- MASVS-PLATFORM-1: "MASVS-PLATFORM-1" is concerned with ensuring that all Inter-Process Communication (IPC) happens securely, which directly relates to how personal data is managed and protected within an app, especially when shared with other apps or systems. The ENISA Guideline emphasizes the necessity of obtaining explicit consent before using personal data, providing specific information about the use, purpose, recipients, storage location, and duration of the data. By ensuring secure IPC mechanisms as per "MASVS-PLATFORM-1", an app can help uphold the principles set out in the ENISA Guideline by preventing unauthorized access to personal data and mitigating the risks of users' personal information being mishandled or accessed without consent. Secure IPC is part of creating a secure environment where user consent can be meaningfully obtained and respected.
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA Guideline is that both are focused on the protection of sensitive user data and ensuring informed consent. "MASVS-PLATFORM-2" emphasizes the secure configuration of WebViews to prevent data leakage, which aligns with the ENISA Guideline's emphasis on obtaining consent prior to using personal data and providing users with specific information regarding their data. Both guidelines aim to protect user privacy and maintain transparency regarding data usage.
- MASVS-PRIVACY-1: Both the "MASVS-PRIVACY-1" and the ENISA guideline emphasize the importance of obtaining informed consent from users before accessing or processing their personal data. They also highlight the need for transparency about what data is being used, why it is needed, who will receive it, and its retention period. Additionally, both stress the necessity of data minimization and the consequences for the app's functionality if consent is not granted for all requested data. This shows a clear correlation between the MASVS-PRIVACY-1 and the ENISA guideline regarding user consent and data privacy practices.
- MASVS-PRIVACY-2: Both the MASVS-PRIVACY-2 control and the ENISA Guideline emphasize the importance of user consent and making the user aware of privacy concerns related to personal data. MASVS-PRIVACY-2 focuses on methods like data abstraction, anonymization, and pseudonymization to prevent user tracking and identification, while the ENISA Guideline insists on clear consent and specific information related to the usage of personal data. Both sets of guidelines aim to enhance user privacy by ensuring that the user is informed and that their data is only used for its intended purpose.
- MASVS-PRIVACY-3: The MASVS-PRIVACY-3 guideline emphasizes informing users about data collection, storage, and sharing practices, which aligns with the ENISA guideline's requirement for consent and explicit notification about the use, purpose, recipients,

storage location, and duration of personal data processing. Both guidelines focus on transparency and user awareness regarding the handling of their personal data.

- MASVS-PRIVACY-4: Both the MASVS-PRIVACY-4 guideline and the ENISA guideline emphasize the importance of user control and informed consent regarding personal data. They both require clear disclosure of what personal data will be used for, the purpose of processing, and they necessitate that users are informed about the implications of not granting consent, which may include limitations on app functionality. Both guidelines promote transparency and user empowerment in managing their privacy settings and personal information.
- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the described ENISA Guideline is that both are focused on the proper handling of sensitive data. The MASVS-STORAGE-1 requirement ensures that sensitive data stored by the app is protected, which is in line with the ENISA Guideline’s requirement for informed user consent prior to using personal data. The ENISA Guideline emphasizes transparency and user control over personal data, which complements the storage security aspect by ensuring users are aware of, and consent to, how their data is handled and stored. Together, they support the overall objective of protecting users’ sensitive data within mobile applications.
- MASVS-STORAGE-2: The MASVS-STORAGE-2 description addresses the concern of unintentional storage or exposure of sensitive data in locations that might be publicly accessible. It speaks to the need for developers to be aware of and mitigate risks associated with storing sensitive data, particularly through system capabilities that can unintentionally expose this data. The ENISA Guideline emphasizes obtaining consent prior to using personal data, along with providing clear information about what data will be used, the purpose, recipients, storage location, and duration. Both MASVS-STORAGE-2 and the ENISA Guideline are correlated in the context that they highlight the importance of handling sensitive or personal data responsibly. MASVS-STORAGE-2 focuses on avoiding unintentional leaks, which supports the intention behind ENISA’s requirements for transparency and consent—ensuring that users’ data is not exposed or used without their clear and informed consent.

8.4 Implementation Guidance (ENISA 7.4):

ENISA Secure Smartphone Development Guidance (7.4): Consent may be collected in 3 main ways: (A) At install time. (B) At run-time when data is sent. (C) Via “opt-in” mechanisms where a user has to explicitly turn on a setting.

8.4.1 OWASP MASVS MAPPING

- MASVS-PRIVACY-1: The description for ”MASVS-PRIVACY-1” emphasizes the importance of apps requesting access to data strictly necessary for their functionality and obtaining informed consent from users, which directly correlates with the ENISA Guideline that outlines the three main ways consent may be collected (at install time, at run-time when data is sent, and via ”opt-in” mechanisms). Both stress the necessity of user consent for data access and sharing, thereby ensuring data minimization and user control over their personal information.
- MASVS-PRIVACY-3: The MASVS-PRIVACY-3 requirement for apps to provide clear information about data collection, storage, and sharing practices aligns with the ENISA Guideline’s mention of consent collection methods. Both emphasize the importance of user awareness and consent regarding data use. These guidelines aim to protect user privacy by ensuring users are duly informed and can consent prior to and during the use of the app, which correlates with the principle of MASVS-PRIVACY-3 ensuring users know how their data is being used.
- MASVS-PRIVACY-4: The correlation between ”MASVS-PRIVACY-4” and the ENISA guideline is that both emphasize the importance of user consent and control over personal data. ”MASVS-PRIVACY-4” addresses the need for users to manage, delete, modify their data, change privacy settings, and be re-prompted for consent if more data is required than initially specified. The ENISA guideline describes methods of collecting consent, which complement the principles outlined in ”MASVS-PRIVACY-4” by ensuring that consent is obtained through clear mechanisms either at install time, run-time, or via opt-in settings, all ensuring user control over their data.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA guideline on consent collection methods is that both of them are focused on preventing or managing the inappropriate exposure of sensitive data. ”MASVS-STORAGE-2” addresses the issue of unintentional storage or exposure of sensitive data in publicly accessible locations, which is a risk that can be mitigated by ensuring proper consent is obtained before any sensitive data is stored or logged. The ENISA guideline outlines how consent may be collected, which is crucial for ensuring that users are aware of and agree to how their sensitive data is being handled. Specifically, collecting consent at install time (A), at run-time (B), or via opt-in mechanisms (C) can help prevent such unintentional leaks by making sure that sensitive data is only collected and stored when the user has explicitly agreed to it.

8.5 Implementation Guidance (ENISA 7.5):

ENISA Secure Smartphone Development Guidance (7.5): It should be possible for the user to withdraw consent at any time in the application. Notify the user how the application behaviour might change in case that consent is withdrawn.

8.5.1 OWASP MASVS MAPPING

- MASVS-AUTH-3: Both ”MASVS-AUTH-3” and the ENISA guideline emphasize the importance of secure user interaction within mobile applications. ”MASVS-AUTH-3” focuses on the implementation of secure additional forms of authentication for sensitive actions, which inherently includes the consideration of user consent and control over authentication methods. Meanwhile, the ENISA guideline addresses the user's ability to withdraw consent, which could impact the behavior of these authentication mechanisms. Both are concerned with providing the user secure control over how their authentication and corresponding consent are managed within the app.
- MASVS-PRIVACY-1: The description of ”MASVS-PRIVACY-1” stresses the importance of only requesting access to data that is essential for app functionality and doing so with informed user consent. It aligns with the ENISA guideline, which emphasizes users' ability to withdraw consent at any time. Both the MASVS requirement and ENISA guideline share a common aim to empower the user with control over their personal data and its usage within applications, thus establishing a clear correlation between them.
- MASVS-PRIVACY-2: Both the MASVS-PRIVACY-2 and the ENISA Guideline focus on the importance of protecting user privacy and provide measures to enhance it. MASVS-PRIVACY-2 highlights using techniques like anonymization to prevent user identification, while the ENISA Guideline ensures users have control over their consent and are informed about changes in application behavior upon consent withdrawal. Both emphasize user autonomy and the minimization of data used for identification.
- MASVS-PRIVACY-3: Both ”MASVS-PRIVACY-3” from the Mobile Application Security Verification Standard (MASVS) and the ENISA Guideline focus on informing users about how their data is used and allowing them control over their own data. MASVS-PRIVACY-3 emphasizes the importance of user awareness regarding data collection, storage, and sharing, and the ENISA Guideline complements this by asserting that users should be able to withdraw their consent for data usage within the application at any time, ultimately providing users with control and transparency over their personal information.
- MASVS-PRIVACY-4: Both ”MASVS-PRIVACY-4” and the mentioned ENISA Guideline emphasize the importance of user control over their personal data within applications. They both advocate for mechanisms that allow users to manage, delete, modify their data, and change privacy settings. Additionally, they share the common requirement that users should be able to withdraw consent at any time and be informed about the consequences of such withdrawal. This reflects a correlation between the principles outlined in MASVS-PRIVACY-4 and the ENISA Guideline.

8.6 Implementation Guidance (ENISA 7.6):

ENISA Secure Smartphone Development Guidance (7.6): Audit communication mechanisms to check for unintended leaks (e.g., image metadata).

8.6.1 OWASP MASVS MAPPING

- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the ENISA Guideline about auditing communication mechanisms to check for unintended leaks is that they both address the security of inter-process communication (IPC). MASVS-PLATFORM-1 emphasizes the secure use of IPC mechanisms provided by the platform, ensuring that both installed apps and the user can interact with the app safely. Similarly, the ENISA guideline recommends auditing communication mechanisms, which includes IPC, to prevent unintended information leaks, such as through image metadata. Both are concerned with ensuring that data exchange processes, which could be exploited for leaks or other security issues, are secure and do not expose sensitive information.
- MASVS-PLATFORM-3: While ”MASVS-PLATFORM-3” specifically mentions the prevention of unintentional leaks via platform mechanisms like auto-generated screenshots that might display sensitive data in the UI, the ENISA guideline on auditing communication mechanisms similarly aims to identify and mitigate potential unintended data leaks, which may include artifacts like image metadata that can disclose sensitive information. Both are concerned with the secure handling of sensitive information to prevent exposure through various channels on the platform.
- MASVS-PRIVACY-1: The correlation between ”MASVS-PRIVACY-1” and the ENISA guideline to ”Audit communication mechanisms to check for unintended leaks (e.g., image metadata)” is that both emphasize the importance of minimizing the risk of data leakage and ensuring that only necessary data is accessed, shared, and transmitted with proper user consent. MASVS-PRIVACY-1 focuses on practicing data minimization, controlling access, and obtaining informed consent, which includes being cautious about sharing data with third parties and the data that third-party SDKs collect. The ENISA guideline stresses auditing communication mechanisms to prevent unintended data leaks, which is an integral part of data minimization and secure data handling as advocated by MASVS-PRIVACY-1. Both guidelines aim to enhance privacy and security by reducing unnecessary data exposure and maintaining control over data flows.
- MASVS-PRIVACY-2: MASVS-PRIVACY-2 and the ENISA guideline both focus on protecting user privacy through technical measures. MASVS-PRIVACY-2 highlights the importance of using unlinkability techniques such as data abstraction, anonymization, and pseudonymization to prevent user tracking and identification, as well as ensuring that data used for one purpose (like fraud detection) isn't repurposed for another (like analytics), which can lead to privacy breaches. The ENISA guideline complements this by specifically directing the auditing of communication mechanisms for unintended data leaks, such as image metadata which can contain identifying information, ensuring that privacy controls are effectively implemented and no accidental disclosures occur. Both emphasize the proactive management of data and privacy controls to protect user identity.
- MASVS-PRIVACY-3: MASVS-PRIVACY-3 which relates to informing users about how their data is used aligns with the ENISA guideline to audit communication mechanisms for

unintended leaks, such as image metadata. Both emphasize the prevention of unexpected data sharing and enforcing transparency in data handling practices.

- MASVS-PRIVACY-4: The correlation exists as both MASVS-PRIVACY-4 and the ENISA guideline emphasize the importance of user control and privacy protection in mobile applications. MASVS-PRIVACY-4 underlines the need for apps to provide mechanisms for users to manage their data and adjust privacy settings, which includes the ability to prevent unintended data leaks. The ENISA guideline specifically calls attention to auditing communication mechanisms to prevent unintended data leaks, such as through image metadata, which is one aspect of ensuring that users have control over their data and that their privacy settings are respected. Both are concerned with safeguarding user data from being shared or used in ways that the user has not consented to.
- MASVS-STORAGE-1: The correlation exists where both are focused on preventing unintended data leaks. ”MASVS-STORAGE-1” emphasizes on sensitive data handling and stresses that data stored locally by the app, regardless of its location, should be adequately protected to avoid unauthorized access. The ENISA Guideline similarly calls for auditing communication mechanisms—which would include data storage and transfer—to ensure there are no unintended leaks, such as through image metadata. Both are centered around the security of sensitive data when stored or transmitted, and involve processes to safeguard against leaks.
- MASVS-STORAGE-2: Both ”MASVS-STORAGE-2” and the ENISA Guideline refer to the need for auditing and controlling the storage and communication mechanisms to prevent unintended leaks of sensitive data. ”MASVS-STORAGE-2” specifically mentions the unintentional storage or exposure of sensitive data in publicly accessible locations due to side effects of using certain APIs and system capabilities. The ENISA Guideline similarly addresses the need to audit communication mechanisms, including details such as image metadata, to avoid accidental leaks. Both are focused on the oversight and mitigation of inadvertent data exposure risks that developers can manage through careful design and implementation practices.

8.7 Implementation Guidance (ENISA 7.7):

ENISA Secure Smartphone Development Guidance (7.7): Keep a record of user consent for the processing of different types of personal data.

8.7.1 OWASP MASVS MAPPING

- MASVS-PRIVACY-1: The Mobile Application Security Verification Standard (MASVS) Privacy Requirement PRV8 (”MASVS-PRIVACY-1” in the given text) emphasizes that apps should only request data strictly necessary for functionality, with informed consent from the user. It also concerns data minimization, restricted access, third-party data sharing based on consent, and awareness of the SDK ’supply chain’ to prevent unnecessary data flow. This is correlated with the European Union Agency for Cybersecurity (ENISA) guideline that stresses keeping a record of user consent for processing personal data. Both guidelines aim to ensure user privacy and data protection, with a focus on informed consent and responsible data handling. MASVS-PRIVACY-1 further takes into account the lifecycle and propagation of data through third-party services, while the ENISA guideline establishes the importance of documenting consent, thus enhancing accountability in data practices.
- MASVS-PRIVACY-3: The ”MASVS-PRIVACY-3” focuses on informing users about data usage, which includes data collection, storage, and sharing practices. ENISA’s guideline to ”Keep a record of user consent for the processing of different types of personal data” complements this by ensuring that there is a verifiable record of user’s agreement to these practices. Both address the importance of transparency and consent in the handling of user data, correlating with each other in the broader context of user privacy and data protection.
- MASVS-PRIVACY-4: There is a correlation between ”MASVS-PRIVACY-4” and the ENISA Guideline provided. ”MASVS-PRIVACY-4” emphasizes the importance of user control over their own data by providing mechanisms for managing, deleting, and modifying their data, as well as changing privacy settings and re-prompting for consent if more data is required than initially agreed upon. This aligns with the ENISA Guideline, which recommends keeping a record of user consent for the processing of different types of personal data. Both stress on transparency and the user’s right to control their personal information, implying a compliance-focused approach to enhancing user privacy and aligning app functionalities with legal requirements on data protection and privacy.
- MASVS-STORAGE-2: Both ”MASVS-STORAGE-2” and the ENISA guideline about keeping a record of user consent for processing different types of personal data are related to the management of sensitive data and ensuring that it is handled in a way that respects user privacy and consent. While ”MASVS-STORAGE-2” focuses on preventing unintentional leaks of sensitive data due to developer oversight or misuse of APIs and system capabilities, the ENISA guideline emphasizes the importance of documenting user consent for data processing activities. Although they address different aspects of data handling, both underscore the importance of meticulous management of sensitive data and adherence to practices that protect user privacy. ”MASVS-STORAGE-2” would be directly related to ensuring that sensitive data, which might include records of user consent, is not exposed unintentionally. Thus, compliance with ”MASVS-STORAGE-2” would support the principles outlined in the ENISA guideline by ensuring that such records are not leaked or stored in an insecure manner.

8.8 Implementation Guidance (ENISA 7.8):

ENISA Secure Smartphone Development Guidance (7.8): Check whether data collection (from the user's device) is not excessive with regard to the consent that has been granted by the user (e.g. collecting more types of data than needed - APP-native + WebKit HTML)

8.8.1 OWASP MASVS MAPPING

- MASVS-PLATFORM-2: Both the MASVS-PLATFORM-2 statement and the ENISA guideline highlight the importance of data protection and secure configuration to prevent sensitive data leakage. MASVS-PLATFORM-2 emphasizes the need for secure configuration of WebViews to avoid data leakage and unwanted exposure of functionality, while the ENISA guideline urges caution against excessive data collection beyond user consent, which is consistent with preventing data leakage as well. Both are concerned with safeguarding user data and privacy.
- MASVS-PRIVACY-1: The "MASVS-PRIVACY-1" description explicitly states that apps should only request access to data they need for their functionality, which aligns with the ENISA guideline's emphasis on not collecting excessive data beyond what the user has consented to. Both guidelines highlight the importance of data minimization and informed user consent.
- MASVS-PRIVACY-2: The MASVS-PRIVACY-2 control and the ENISA Guideline both focus on the principle of data minimization and ensuring that data collection aligns with what users consent to. MASVS-PRIVACY-2 emphasizes the use of techniques to protect user identity and prevent user tracking, which aligns with the ENISA Guideline's objective to verify that data collection is not excessive and is within the scope of user consent. Both are concerned with collecting only the necessary data for the intended purpose, without infringing on user privacy.
- MASVS-PRIVACY-3: "MASVS-PRIVACY-3" and the ENISA guideline both emphasize the principle of data minimization and the importance of transparency regarding data collection and user consent. MASVS-PRIVACY-3 requires clear information to be provided to users about data handling practices, which parallels the ENISA guideline's focus on ensuring data collection is not excessive in relation to the consent given by the user. Both share the underlying concern of protecting user privacy by avoiding unnecessary data collection and being clear about the data that is collected.
- MASVS-PRIVACY-4: Both the MASVS-PRIVACY-4 standard and the ENISA guideline emphasize the importance of user consent and the principle of data minimization. MASVS-PRIVACY-4 highlights the need for user control over their data, including the mechanisms to manage, delete, and modify data, as well as to revoke consent. It also specifies that apps should not collect more data than initially specified without re-prompting for consent. Similarly, the ENISA guideline checks for excessive data collection beyond what the user has consented to, which aligns with the MASVS-PRIVACY-4 standard's emphasis on not exceeding the initially specified data requirements without updated user consent. Both stress transparency and the user's right to control their personal data.
- MASVS-STORAGE-1: Both "MASVS-STORAGE-1" and the ENISA guideline pertain to the handling and storage of sensitive data collected from users or other sources on mobile devices. "MASVS-STORAGE-1" focuses on ensuring proper protection of sensitive data

that is stored locally by the app, while the ENISA guideline emphasizes the importance of not collecting excessive data beyond what the user has consented to. While the MASVS speaks to the secure handling of data that is stored, the ENISA guideline addresses the initial collection of data. There is a correlation in the sense that secure storage should be proportionate to the data collection practices, meaning that only the data that is necessary and has been consented to by the user should be collected and subsequently stored securely. The secure storage is a follow-up to compliant data collection practices.

- MASVS-STORAGE-2: Both MASVS-STORAGE-2 and the ENISA guideline emphasize on data management practices whereby the former focuses on preventing unintentional storage or exposure of sensitive data through proper use of APIs and system capabilities, while the latter is concerned with ensuring that the data collection from user's device is not excessive and aligns with the consent granted by the user. Essentially, they both correlate with the principle of minimizing the amount and scope of data handling to what is necessary and within the user's expectations and permissions.

8.9 Implementation Guidance (ENISA 7.9):

ENISA Secure Smartphone Development Guidance (7.9): Consider taking advantage of built-in features to require access to device sensors and data (e.g., access to gps, camera, etc.). Provide clear explanation on why the access is needed.

8.9.1 OWASP MASVS MAPPING

- MASVS-AUTH-3: Both "MASVS-AUTH-3" and the ENISA guideline emphasize the implementation of additional security measures for sensitive actions within an application. While "MASVS-AUTH-3" suggests the use of multiple forms of authentication such as biometric, pin, MFA code generator, etc., the ENISA guideline recommends leveraging built-in device features for secure access to sensors and data, providing a rationale for their use. Both advocate for enhanced security but approach it from slightly different angles—MASVS-AUTH-3 focuses on user authentication methods, while ENISA addresses secure access to device features. Nonetheless, there is a correlation in their mutual goal to reinforce app security.
- MASVS-PLATFORM-1: The Mobile Application Security Verification Standard (MASVS) requirement "MASVS-PLATFORM-1" and the ENISA Guideline both relate to secure interaction with platform-level features and the protection of sensitive device resources. "MASVS-PLATFORM-1" focuses on ensuring that Inter-Process Communication (IPC) mechanisms are securely managed, which includes the proper handling of access to device sensors and data by both installed apps and users. Meanwhile, the ENISA Guideline emphasizes the importance of utilizing built-in features to control access to device sensors and data (such as GPS, camera, etc.) and stresses providing a clear explanation to the user about why the access is necessary. Both can be seen as complementary as they deal with the secure usage and access management of critical device functionalities and user data.
- MASVS-PRIVACY-3: Both "MASVS-PRIVACY-3" and the ENISA Guideline emphasize the importance of transparency in how mobile applications access and use user data. "MASVS-PRIVACY-3" focuses on ensuring apps provide clear information about their data practices, while the ENISA Guideline advises using built-in features to require data access and providing a clear explanation for it. Both stress the user's right to understand the data usage practices of the app.
- MASVS-PRIVACY-4: Both "MASVS-PRIVACY-4" and the ENISA guideline emphasize the importance of user control and transparency concerning their data. "MASVS-PRIVACY-4" specifies that users should have the ability to manage, delete, and modify their data and consent, along with the necessity for apps to update disclosures and re-prompt for consent if more data is required. Similarly, the ENISA guideline advises leveraging built-in features to obtain consent for data access and stresses providing a clear explanation as to why access is needed. Thus, there is a clear correlation between the two as they both advocate for user empowerment and clarity of data use within apps.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the mentioned ENISA Guideline is that both are concerned with the security and integrity of the platform on which mobile applications run. "MASVS-RESILIENCE-1" emphasizes the importance of ensuring that the operating system (platform) has not been tampered with in order to maintain trust in various security features such as secure storage, biometrics, and sandboxing. This implies that the app relies on the integrity of the platform's security

mechanisms to protect sensitive data and functionality. The ENISA Guideline, on the other hand, suggests leveraging built-in platform features for accessing device sensors and data, while also advising to provide clear explanations to users on why such access is necessary. This guideline implicitly relies on the premise that the platform's built-in features are secure and trustworthy. If the operating system were compromised, as cautioned by "MASVS-RESILIENCE-1," the assurances provided by these built-in features could be undermined, potentially leading to unauthorized access to device sensors and data. Both guidelines demonstrate the importance of ensuring a secure and uncompromised operating system to safeguard the app's data and the privacy of its users, hence showing a correlation.

8.10 Implementation Guidance (ENISA 7.10):

ENISA Secure Smartphone Development Guidance (7.10): Minimize access to sensor data whenever possible (e.g., do not automatically collect geolocation data).

8.10.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation exists in the context of implementing best practices for secure use of protocols in mobile apps. MASVS-AUTH-1, as outlined in the Mobile Application Security Verification Standard (MASVS), emphasizes the need for apps to use secure authentication and authorization protocols while adhering to relevant best practices. The ENISA guideline to "Minimize access to sensor data whenever possible" is consistent with these best practices since minimizing unnecessary sensor data collection, like geolocation data, is an essential part of ensuring data privacy and security, which falls under the overall security protocol compliance for mobile apps.
- MASVS-CRYPTO-1: The correlation between "MASVS-CRYPTO-1" and the ENISA guideline that suggests minimizing access to sensor data lies in the underlying principle of data protection and minimizing the attack surface. While "MASVS-CRYPTO-1" deals with securing user data through best practices in cryptography, which is crucial when considering that physical access to a device could compromise data, the ENISA guideline addresses the principle of data minimization. Both are aimed at protecting sensitive user information - "MASVS-CRYPTO-1" through encryption and secure data handling practices, and the ENISA guideline through limiting the unnecessary collection of potentially sensitive data such as geolocation, which could also benefit from cryptographic protections. Hence, both guidelines are related in their goal of enhancing user data security in a mobile context.
- MASVS-PRIVACY-1: Both the MASVS-PRIVACY-1 description and the ENISA guideline emphasize the principle of data minimization. The MASVS-PRIVACY-1 description asserts that apps should only request access to data that is strictly necessary for their functionality and with user consent, which correlates with the ENISA guideline's advice to minimize access to sensor data and not to collect it automatically. Both stress the importance of limiting data access and collection to enhance user privacy and security.
- MASVS-PRIVACY-2: Both the MASVS-PRIVACY-2 and ENISA guideline emphasize the importance of protecting user privacy by minimizing the data collected and processed. The MASVS-PRIVACY-2 focuses on using techniques like data abstraction, anonymization, and pseudonymization to prevent user identification and tracking, while the ENISA guideline suggests minimizing access to sensor data, which is in line with the concept of collecting the least amount of data necessary to serve the purpose of the application, thereby upholding user privacy.
- MASVS-PRIVACY-3: Both "MASVS-PRIVACY-3" and the ENISA Guideline emphasize the principle of data minimization and transparency in the context of user privacy. "MASVS-PRIVACY-3" focuses on the right of users to be informed about how their data is used, including unexpected data collection practices, while the ENISA Guideline specifically mentions minimizing access to sensitive sensor data such as geolocation. Both guidelines aim to protect user privacy by ensuring that data is only collected when necessary and that users are aware of such practices.
- MASVS-PRIVACY-4: The MASVS-PRIVACY-4 guideline focuses on providing users with control over their data, including the ability to manage, delete, and modify their data,

and to change privacy settings when needed. This concept inherently supports minimizing unnecessary access to data, aligning with the ENISA guideline that advises minimizing access to sensor data. Both guidelines promote the principle of data minimization and user control over personal information, though they address different aspects of user privacy and data management.

- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA guideline regarding minimizing access to sensor data is that both are concerned with the protection of sensitive data. MASVS-STORAGE-2 addresses the risk of unintentional storage or exposure of sensitive data due to the use of certain APIs or system features, and emphasizes the role of the developer in preventing these leaks. Similarly, the ENISA guideline advises minimizing access to sensor data, such as geolocation, to prevent unnecessary collection or exposure. Both directives are aimed at reducing the risk of sensitive information being compromised by limiting exposure and access as a part of secure development practices.

8.11 Implementation Guidance (ENISA 7.11):

ENISA Secure Smartphone Development Guidance (7.11): Reduce data granularity and anonymize data on the device instead of remotely. (e.g., strip image metadata).

8.11.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: Both ”MASVS-AUTH-2” and the ENISA Guideline address security concerns related to data processing on the device. MASVS-AUTH-2 focuses on the correct implementation of biometrics or local PIN code authentication, which are local device security mechanisms. The ENISA Guideline suggests reducing data granularity and anonymizing data locally rather than remotely, which also pertains to enhancing security through local measures. Although they focus on different aspects—MASVS-AUTH-2 on authentication and ENISA on data privacy—both are concerned with ensuring that sensitive operations are handled in a secure manner on the local device, reducing reliance on remote processing where security might not be as easily enforced or verified.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” which emphasizes the importance of cryptography in securing user data on mobile devices, specifically against attackers with physical access, and the ENISA guideline to ”Reduce data granularity and anonymize data on the device instead of remotely” is rooted in the focus on data protection. Following MASVS-CRYPTO-1’s guidance to adhere to general cryptography best practices will inherently support minimizing the amount of sensitive data stored and processed on the device. By reducing data granularity and anonymizing data as per ENISA’s guideline, the potential impact of a cryptographic failure or breach is minimized. Both aim to enhance the security and privacy of user data on mobile devices.
- MASVS-PLATFORM-2: There is a correlation between ”MASVS-PLATFORM-2” and the ENISA guideline ”Reduce data granularity and anonymize data on the device instead of remotely. (e.g., strip image metadata).” Reasoning: Both statements are concerned with preventing sensitive data leakage from mobile applications. MASVS-PLATFORM-2 focuses on secure configuration of WebViews to prevent such leakage, which includes ensuring that sensitive data is not exposed through WebView interfaces, such as JavaScript bridges. The ENISA guideline advises reducing data granularity and anonymizing data on the device to prevent sensitive information from being transmitted or stored in a form that could lead to data leakage. Although these two guidelines approach the problem from different angles—one from the perspective of UI components and the other from data handling—they both aim to protect sensitive data within mobile applications.
- MASVS-PLATFORM-3: The correlation between ”MASVS-PLATFORM-3” and the ENISA guideline about reducing data granularity and anonymizing data on the device is that both are focused on protecting sensitive user data from being inadvertently exposed. ”MASVS-PLATFORM-3” aims to prevent sensitive data from being captured by platform mechanisms like screenshots, which could lead to unintentional data leaks. Similarly, the ENISA guideline suggests minimizing the amount of sensitive data (reducing data granularity) and anonymizing it on the device to prevent leaks and exposure, especially in the case where data might be shared or transmitted. Both are concerned with the in-device handling of sensitive information to enhance privacy and security.
- MASVS-PRIVACY-1: Both ”MASVS-PRIVACY-1” and the ENISA guideline emphasize data minimization and the protection of user privacy. MASVS-PRIVACY-1 strongly advo-

cates for apps to only access data that is absolutely necessary for functionality and with user consent, which correlates with the ENISA guideline's advice to reduce data granularity and anonymize data on the device to prevent unnecessary or excessive data from being remotely accessible or processed. Both promote principles that are intended to minimize the amount of data an app collects, processes, and shares, thereby enhancing user privacy and reducing the potential impact of data breaches or leaks.

- MASVS-PRIVACY-2: The correlation is present in the emphasis on techniques that mitigate the risk of user identification and tracking. MASVS-PRIVACY-2 focuses on the use of unlinkability techniques such as data abstraction, anonymization, and pseudonymization to protect user identity. Similarly, the ENISA Guideline recommends reducing data granularity and anonymizing data on the device as a means to avoid remote processing of detailed personal information, which aligns with the principle of transforming or withholding identity information to maintain privacy. Both guidelines aim to ensure that user data is handled in a way that minimizes the risk of personal identification, whether it is by limiting the use of detailed data or by employing anonymization techniques.
- MASVS-PRIVACY-3: Both MASVS-PRIVACY-3 and the ENISA Guideline concern the handling of user data by mobile applications. MASVS-PRIVACY-3 emphasizes the importance of transparency in informing users about data usage, which is complementary to the ENISA Guideline's focus on reducing data granularity and anonymizing data. By adhering to both directives, app developers support privacy by being explicit about data practices and implementing methods to minimize risks to user data.
- MASVS-PRIVACY-4: There is a correlation between "MASVS-PRIVACY-4" and the ENISA guideline mentioned. Both relate to the privacy and control users have over their data. MASVS-PRIVACY-4 focuses on giving users the ability to manage their data and control privacy settings, which aligns with the principle of reducing data granularity and anonymizing data on the device as indicated by ENISA. Both guidelines aim to enhance user privacy by limiting the amount of personal data that can be exploited.
- MASVS-STORAGE-1: The correlation exists because both the MASVS-STORAGE-1 requirement and the ENISA Guideline focus on protecting sensitive data on the device. MASVS-STORAGE-1 addresses the need for proper protection of sensitive data stored locally in various possible locations within a mobile application's environment, whether private or public. The ENISA guideline recommends reducing data granularity and anonymizing data on the device to enhance privacy and data protection. Both points aim to minimize the risks associated with sensitive data exposure by ensuring it is adequately secured and privacy is maintained, albeit through different methods (storage protection in the case of MASVS-STORAGE-1 and data anonymization in the case of the ENISA Guideline).
- MASVS-STORAGE-2: The "MASVS-STORAGE-2" description emphasizes the prevention of unintentional storage or exposure of sensitive data in publicly accessible locations, which could be caused by not appropriately handling APIs or system capabilities. The ENISA Guideline recommends reducing data granularity and anonymizing data on the device, which aligns with the MASVS-STORAGE-2 control as both aim to enhance data protection on the device. By stripping potentially sensitive metadata from images (as per the ENISA Guideline), a developer may prevent such unintentional leaks of sensitive information. Thus, there is a correlation in that both are concerned with mitigating risks related to unnecessary or insecure storage of sensitive data on the device.

8.12 Implementation Guidance (ENISA 7.12):

ENISA Secure Smartphone Development Guidance (7.12): Require consent prior to providing user data to third parties. Provide clear notice of data shared cross-application with third-parties. Never provide precise location data to third-party applications nor data stored in the secure containers of the application.

8.12.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA Guideline lies in the emphasis on ensuring the secure use of protocols and protecting user data. MASVS-AUTH-1 stresses the importance of apps following best practices for secure authentication and authorization to safeguard user access and data transmission to remote endpoints. Similarly, the ENISA Guideline enforces the principle of consent and transparency in data sharing with third parties, especially sensitive information like precise location data or data in secure containers. Both aim to protect user data from unauthorized access and misuse, highlighting the need for proper security measures in app development.
- MASVS-NETWORK-1: While ”MASVS-NETWORK-1” emphasizes on the importance of data privacy and integrity during transit by setting up secure connections, the cited ENISA guideline is concerned with user consent and transparency when sharing data with third parties. Both highlight the criticality of protecting user data, albeit MASVS-NETWORK-1 focuses on the technical means of protection during data transit, and the ENISA guideline concentrates on user rights and data sharing practices. The correlation lies in their shared goal to preserve user data confidentiality and integrity.
- MASVS-PLATFORM-1: Both MASVS-PLATFORM-1 and the ENISA guideline emphasize the secure management and sharing of user data. MASVS-PLATFORM-1 addresses secure interactions involving IPC mechanisms, which includes how data is exposed and shared between apps and parties on the platform. The ENISA guideline similarly calls attention to the need for user consent prior to sharing data with third parties, clear notification of data sharing practices, and restrictions on sharing sensitive data like precise location or data from secure containers. Although they describe the concept with different focuses, they both correlate in their aim to ensure controlled and secure data exchange to protect user privacy and data integrity.
- MASVS-PLATFORM-2: Both the MASVS-PLATFORM-2 description and the ENISA Guideline emphasize the importance of securing sensitive user data. The MASVS-PLATFORM-2 focuses on securely configuring WebViews to prevent data leakage and exposure of sensitive functionality. The ENISA Guideline addresses the need for user consent before sharing data with third parties, provides clarity on data sharing, and protects precise location data and secure container contents. Both are concerned with the secure handling of user information and preventing unauthorized or unintended data sharing.
- MASVS-PLATFORM-3: Both MASVS-PLATFORM-3 and the ENISA Guideline are concerned with the protection of sensitive user data. MASVS-PLATFORM-3 focuses on preventing unintentional leaks of sensitive data through UI elements and platform features, while the ENISA Guideline emphasizes the need for user consent before sharing data with third parties, clear notification of cross-application data sharing, and restricting access to precise location data and secure containers. Both aim to enhance user privacy and limit the

exposure of sensitive data, establishing a correlation in their objectives to safeguard user information from unauthorized access and disclosure.

- MASVS-PRIVACY-1: The MASVS-PRIVACY-1 description and the ENISA Guideline both emphasize the importance of obtaining user consent before sharing their data with third parties, ensuring transparency about data sharing, and protecting sensitive data such as precise location information. Both guidelines advocate for data minimization and heightened control over data access, aligning in their approach to user privacy and data protection.
- MASVS-PRIVACY-2: The description of "MASVS-PRIVACY-2" focuses on protecting user identity by employing techniques such as data abstraction, anonymization, and pseudonymization, which aligns with the ENISA guideline that emphasizes requiring consent before sharing user data with third parties. Moreover, the MASVS control discusses establishing technical barriers to prevent the repurposing of user data for different functions, which is consistent with the ENISA's guideline on providing clear notice of cross-application data sharing and restrictions on sharing precise location data and the contents of secure containers with third parties. Both sources advocate for user privacy and controlled data sharing to third-party applications.
- MASVS-PRIVACY-3: The correlation between "MASVS-PRIVACY-3" and the ENISA Guideline is clear as both focus on providing users with information about how their data is being used and shared, particularly with third parties. MASVS-PRIVACY-3 emphasizes the necessity for apps to clearly inform users about data collection, storage, and sharing practices, while the ENISA Guideline specifically requires user consent before sharing data with third parties, clear notice of data sharing, and restrictions on sharing sensitive location data and data in secure containers. Both guidelines prioritize user awareness and consent regarding data privacy practices.
- MASVS-PRIVACY-4: Both the MASVS-PRIVACY-4 description and the ENISA Guideline emphasize the importance of user control and consent over their data. MASVS-PRIVACY-4 focuses on users' ability to manage, delete, and modify their data, as well as the need for apps to re-prompt for consent when more data is required. The ENISA Guideline requires user consent before sharing data with third parties, clear notice of such sharing, and restrictions on sharing precise location data and data from secure containers. Both guidelines are aligned in prioritizing user consent and transparency regarding data management and sharing.
- MASVS-STORAGE-1: The correlation between "MASVS-STORAGE-1" and the ENISA Guideline lies in the focus on the protection and handling of sensitive data. "MASVS-STORAGE-1" talks about the importance of protecting sensitive data, regardless of where it is stored, whether in private or public storage locations. The ENISA Guideline stresses the need for user consent before sharing data with third parties and emphasizes the protection of data shared across applications and stored in secure containers. Both are concerned with ensuring that sensitive user data is not mishandled or inadvertently exposed to unauthorized parties, hence there is a clear alignment in their purpose of enhancing data security and privacy.
- MASVS-STORAGE-2: The correlation exists because both "MASVS-STORAGE-2" and the ENISA Guideline are concerned with the protection of sensitive user data. "MASVS-STORAGE-2" addresses the risk of unintentional data leaks that can occur due to improper use of APIs or system features. Similarly, the ENISA Guideline emphasizes the importance of user consent prior to sharing their data with third parties and preventing unintended disclosure of sensitive information, including precise location data and data within secure containers. Both guidelines aim to ensure that sensitive data is not inadvertently or inappropriately exposed or shared, highlighting a focus on user privacy and data security.

8.13 Implementation Guidance (ENISA 7.13):

ENISA Secure Smartphone Development Guidance (7.13): Reduce retention period on the mobile or remotely to the minimum amount of time needed to provide the service. Delete data immediately after the retention period has expired. Delete data from all locations (especially remote servers) where data might be stored.

8.13.1 OWASP MASVS MAPPING

- MASVS-CRYPTO-2: The correlation between ”MASVS-CRYPTO-2” and the ENISA guideline lies in the aspect of managing cryptographic keys and data securely. Both reflect on security best practices related to handling sensitive information. ”MASVS-CRYPTO-2” emphasizes the importance of proper key management throughout the lifecycle of cryptographic keys, which includes secure storage and timely disposal, aligning with the ENISA guideline that recommends minimizing data retention and deleting it after the retention period, including from remote servers where cryptographic keys might be stored. This reduces the risk of unauthorized access or misuse of keys and data after they are no longer needed, enhancing overall security.
- MASVS-PLATFORM-3: The correlation exists in the emphasis on protecting sensitive data. MASVS-PLATFORM-3 focuses on preventing accidental data leaks by controlling how sensitive data is displayed and handled on the platform, including considerations for screenshots and shoulder surfing. The ENISA guideline emphasizes minimizing the retention period of sensitive data and ensuring its deletion after the retention period, thereby reducing the risk of data being exposed or accessed without authorization. Both are concerned with safeguarding sensitive information, albeit through different approaches—one through user interface precautions and the other through data lifecycle management.
- MASVS-PRIVACY-1: The description of ”MASVS-PRIVACY-1” emphasizes the importance of data minimization and accessing only the data that is necessary for the app’s functionality, along with obtaining informed consent from the user. This principle directly correlates with the ENISA guideline advising to reduce the data retention period to the minimum necessary for providing the service and deleting data post that period. Both stress on minimizing the amount of data collected and stored to reduce the risk of data breaches or leaks, aligning with the principles of privacy by design and ensuring data protection compliance.
- MASVS-PRIVACY-2: The ENISA Guideline emphasizing the reduction of the retention period and deletion of data after it expired aligns with the philosophy of MASVS-PRIVACY-2, which focuses on protecting user identity through techniques such as anonymization and handling of unique identifiers. Both advocate for minimizing the risk to user privacy by limiting data usage and lifespan, and ensuring data is only used for its intended purpose without unnecessary retention or repurposing.
- MASVS-PRIVACY-3: The correlation between ”MASVS-PRIVACY-3” and the ENISA guideline on retention period centers around the management and transparency of user data. MASVS-PRIVACY-3 emphasizes users’ right to know how their data is used, including collection, storage, and sharing practices, which implies that users should be informed if data is retained only for the necessary period of time as per the service requirement. The ENISA guideline’s instruction to reduce the retention period to the minimum necessary and delete data after its expiration aligns with the principles of transparency and minimal

retention stated in MASVS-PRIVACY-3, reinforcing the idea that data should not be kept longer than needed and users should be aware of this practice. Both sets of guidelines highlight the importance of user privacy and the responsible handling of user data.

- MASVS-PRIVACY-4: Both ”MASVS-PRIVACY-4” and the ENISA Guideline emphasize user control over their personal data by ensuring that data is only kept for as long as necessary to provide the service and is deleted after that period. MASVS-PRIVACY-4 focuses on providing mechanisms for users to manage their data and updating consent when more data is needed, while the ENISA guideline specifically talks about reducing retention periods to the minimum necessary and deleting data after its retention period expires. Both statements aim to enhance user privacy by minimizing the amount of time personal data is retained and ensuring users have the power to control their data.
- MASVS-STORAGE-1: The description of ”MASVS-STORAGE-1” indicates that sensitive data must be properly protected regardless of where it is stored, implying a need for secure handling and retention practices. The ENISA Guideline emphasizes minimizing the retention period and ensuring data is deleted after this period, also from all storage locations. Both statements correlate as they are concerned with the secure and minimal retention of sensitive data, implying that measures should be in place to handle data storage and deletion properly, whether it is on the mobile device itself or remotely.
- MASVS-STORAGE-2: The MASVS-STORAGE-2 requirement addresses the need to avoid unintentional storage or exposure of sensitive data in publicly accessible locations, while the ENISA guideline emphasizes reducing retention periods to the minimum necessary and ensuring proper deletion of data after its retention period. Both concepts are correlated as they focus on the proper handling and limitation of sensitive data storage to prevent leaks and protect user privacy. By preventing unintentional leaks (MASVS-STORAGE-2) and ensuring data is deleted after its intended use (ENISA guideline), both aim to strengthen data protection and minimize the risk of sensitive data exposure.

8.14 Implementation Guidance (ENISA 7.14):

ENISA Secure Smartphone Development Guidance (7.14): Use privacy enhancing technologies, that support data minimization, anonymization and security of personal data.

8.14.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA Guideline ”Use privacy enhancing technologies, that support data minimization, anonymization and security of personal data” exists in the context of emphasizing security and privacy in applications, particularly in relation to user authentication and authorization. MASVS-AUTH-1’s mention of following best practices for secure use of authentication and authorization protocols inherently supports the principle of security of personal data as recommended by ENISA. Moreover, while not explicitly mentioned in MASVS-AUTH-1, the practice of following security best practices in authentication tends to go hand in hand with privacy-enhancing technologies that include data minimization and anonymization, as these technologies also aim to protect user credentials and personal information from being exposed or misused.
- MASVS-AUTH-2: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-AUTH-2” pertains to the proper implementation of biometric and local PIN code authentication methods. This relates to the ENISA guideline which emphasizes the use of privacy-enhancing technologies to support data minimization, anonymization, and security of personal data. Correct implementation of authentication mechanisms is a critical aspect of maintaining the security and privacy of personal data within an application. By ensuring that local authentication methods are properly implemented, an application will inherently be supporting data minimization (by not unnecessarily sharing data) and the security of personal data, because local authentication reduces the risk of personal data being intercepted or compromised during remote authentication processes. Additionally, the utilization of biometric data for authentication should also be done in such a way that it is anonymized and secure, aligning with the ENISA guideline on privacy enhancement.
- MASVS-AUTH-3: MASVS-AUTH-3 and the ENISA guideline on using privacy-enhancing technologies share the common goal of enhancing security and privacy, but they address it from slightly different perspectives. MASVS-AUTH-3 specifically focuses on implementing additional authentication methods to safeguard sensitive actions within an app. This can indirectly support data minimization and security of personal data by ensuring that only authorized users can access or perform sensitive operations, thus reducing the risk of unauthorized access to personal data. The ENISA guideline promotes technologies that enhance privacy through data minimization, anonymization, and securing personal data, which includes implementing robust authentication mechanisms as a component of protecting personal data. Both aim to increase the overall security and privacy posture of an application.
- MASVS-CODE-1: The MASVS-CODE-1 guideline emphasizes the importance of keeping the mobile application up-to-date with the latest security patches and features of the mobile operating system. This practice is crucial to mitigate known vulnerabilities and ensure the security of the application, which aligns with the ENISA guideline to use privacy-enhancing

technologies. By maintaining a secure platform, it inherently supports the security of personal data—one of the focuses of the ENISA guideline—because a secure platform is less likely to be compromised and leak personal data. Therefore, although the MASVS-CODE-1 guideline is about keeping the platform updated, indirectly it also supports the data minimization, anonymization, and security of personal data by providing a more secure environment against well-known threats.

- MASVS-CODE-4: The correlation between "MASVS-CODE-4" and the ENISA guideline about using privacy enhancing technologies is that both stress on the importance of treating data securely. MASVS-CODE-4 emphasizes the need to verify and sanitize incoming data to prevent security vulnerabilities such as injection attacks, which could lead to unauthorized access to or disclosure of personal data. The ENISA guideline points out the necessity of using technologies that support data minimization, anonymization, and security of personal data. Both are concerned with safeguarding data from untrusted sources and preventing misuse of personal information, albeit from slightly different angles: MASVS-CODE-4 is about the integrity and security of input data processing, and the ENISA guideline is about maintaining privacy and security throughout data handling practices.
- MASVS-CRYPTO-1: The Mobile Application Security Verification Standard (MASVS) guideline "MASVS-CRYPTO-1" correlates with the ENISA Guideline stating "Use privacy enhancing technologies, that support data minimization, anonymization and security of personal data." The reasoning for this correlation is that MASVS-CRYPTO-1 emphasizes the importance of cryptography in protecting user data in mobile environments, especially considering the likelihood of physical device access by attackers. The focus on general cryptography best practices is in line with using privacy-enhancing technologies as advocated by ENISA. These practices aim to secure personal data, which includes methods like data minimization, anonymization, and ensuring the security of personal data. Both guidelines prioritize the protection and privacy of user data within their respective contexts.
- MASVS-CRYPTO-2: The correlation between "MASVS-CRYPTO-2," which discusses the management of cryptographic keys throughout their lifecycle, including key generation, storage, and protection, and the ENISA Guideline on using privacy-enhancing technologies to support the minimization, anonymization, and security of personal data, can be understood as follows: Good key management as outlined in "MASVS-CRYPTO-2" is essential for maintaining the security and integrity of encryption systems that protect personal data. Without proper key management, even the strongest cryptography can be rendered ineffective. Privacy-enhancing technologies that the ENISA Guideline refers to often rely on strong cryptographic practices to anonymize and secure personal data. Ensuring that cryptographic keys are handled securely throughout their lifecycle directly supports the objective of keeping personal data secure and enhancing privacy. This shows a clear correlation where effective key management is an enabling factor for privacy-enhancing technologies to achieve data minimization, anonymization, and security.
- MASVS-NETWORK-1: The MASVS-NETWORK-1 control that emphasizes ensuring data privacy and integrity during data transit correlates with the ENISA guideline that advocates for the use of privacy-enhancing technologies. Both focus on the protection of personal data and maintaining privacy and security in the context of network communications. MASVS-NETWORK-1 specifically refers to secure connections, which is a key aspect of privacy-enhancing technologies as mentioned in the ENISA guideline.
- MASVS-NETWORK-2: The correlation between "MASVS-NETWORK-2" and the ENISA guideline is that both controls aim to enhance security and privacy. MASVS-NETWORK-2 focuses on reducing the attack surface by trusting only specific Certificate Authorities (CAs) through certificate pinning, which prevents man-in-the-middle attacks and ensures the in-

tegrity of the encrypted data being transmitted over the network. By doing so, it inherently supports the security of personal data. The ENISA guideline emphasizes the use of privacy-enhancing technologies that include measures for data minimization, anonymization, and the security of personal data. Secure and trusted communication facilitated by certificate pinning contributes to the overall security framework recommended by ENISA, as it guards against data interception and unauthorized access to sensitive personal information.

- MASVS-PLATFORM-1: The correlation between "MASVS-PLATFORM-1" and the ENISA guideline regarding the use of privacy enhancing technologies is present. The MASVS control emphasizes the security of inter-process communication (IPC) mechanisms, which are ways in which apps interact with one another and with users. By securing IPC mechanisms, an app can help protect personal data from being accessed or manipulated by unauthorized parties, aligning with the ENISA guideline's recommendation to use technologies that support the security of personal data. Securing IPC mechanisms is a part of implementing privacy enhancing technologies that contribute to data minimization and the protection of personal data, as secure IPC ensures that only intended interactions occur and that sensitive information is not inadvertently exposed. This reflects the intention of the ENISA guideline to encourage technologies that prevent data breaches and unauthorized access to personal data.
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA Guideline about using privacy enhancing technologies is established through the emphasis on secure configuration to prevent sensitive data leakage. MASVS-PLATFORM-2 highlights the need for securely configuring WebViews to protect sensitive data and functionality, which aligns with the aim of privacy enhancing technologies advocated by ENISA to minimize, anonymize, and secure personal data. Both sets of guidance focus on safeguarding user information and reducing the risk of exposing sensitive data.
- MASVS-PLATFORM-3: There is a correlation between "MASVS-PLATFORM-3" and the ENISA guideline mentioned. The MASVS-PLATFORM-3 control is about protecting sensitive data from being unintentionally leaked through platform mechanisms or accidentally disclosed, which aligns with the ENISA guideline's emphasis on using privacy-enhancing technologies to support data minimization, anonymization, and security of personal data. Both focus on safeguarding user data and privacy, ensuring that personal information is not exposed or compromised. Additionally, implementing measures as suggested by MASVS-PLATFORM-3 can be considered part of using privacy-enhancing technologies as advised by ENISA, since they would help minimize the amount of sensitive data exposed and strengthen the security and privacy of personal data.
- MASVS-PRIVACY-1: The MASVS-PRIVACY-1 description emphasizes that apps should request access to data strictly necessary for their functionality and with informed user consent, promote data minimization, and restrict unnecessary data sharing, particularly with third parties or through SDKs without consent. It also highlights the importance of being responsible for data throughout the SDK supply chain. This correlates with the ENISA guideline of using privacy-enhancing technologies that support data minimization, anonymization, and security of personal data, as both prioritize reducing data collection, limiting access, and ensuring security and privacy of user data.
- MASVS-PRIVACY-2: The description of "MASVS-PRIVACY-2" is closely aligned with the ENISA guideline to "Use privacy enhancing technologies, that support data minimization, anonymization and security of personal data." Both emphasize the importance of protecting user identity through techniques that make it difficult to link data to specific users. These include data abstraction, anonymization, pseudonymization, and the use of technical barriers to prevent misuse of user data. The goal is to ensure that personal data is

handled in a way that preserves privacy while still allowing for legitimate uses like fraud detection.

- MASVS-PRIVACY-3: Both MASVS-PRIVACY-3 and the ENISA Guideline articulate the importance of respecting user privacy by providing transparency around data usage, advocating for the minimization and protection of personal data. MASVS-PRIVACY-3 focuses on the right of users to understand how their data is used, including unexpected data collection practices, and adherence to platform guidelines. The ENISA Guideline emphasizes the use of privacy-enhancing technologies that support data minimization, anonymization, and security, which aligns with the principle of giving users clear information and protecting their data as outlined in MASVS-PRIVACY-3. Both guidelines are concerned with ensuring users' personal data is collected, stored, and shared responsibly and securely, and that users are informed about these practices.
- MASVS-PRIVACY-4: The connection between "MASVS-PRIVACY-4" and the ENISA guideline on using privacy-enhancing technologies is evident. Both emphasize the importance of user control over personal data. MASVS-PRIVACY-4 speaks to providing users with mechanisms to manage their data and to consent to how it's used, which aligns with ENISA's guideline to enhance privacy, focus on data minimization, and anonymization for protecting personal data. Both advocate for approaches that give users power over their information and enhance privacy.
- MASVS-RESILIENCE-3: The correlation between "MASVS-RESILIENCE-3" which is about implementing anti-static analysis mechanisms to impede comprehension and make it difficult to figure out how an app works using static analysis and the ENISA guideline to "Use privacy enhancing technologies (PETs)" that support data minimization, anonymization, and security of personal data lies in the shared goal of protecting software from unwanted analysis and misuse, which includes the abuse of personal data. While MASVS-RESILIENCE-3 is more focused on protecting the app's integrity by hindering reverse engineering attempts, the ENISA guideline promotes the use of PETs to minimize and secure data, which can include techniques to obscure data from being easily interpreted or misused, thus also potentially deterring reverse engineering or tampering. Both aim to enhance the security and privacy features of mobile applications.
- MASVS-STORAGE-1: The Mobile Application Security Verification Standard (MASVS) requirement "MASVS-STORAGE-1" is correlated with the ENISA guideline to "Use privacy enhancing technologies, that support data minimization, anonymization and security of personal data." The MASVS-STORAGE-1 focuses on ensuring that sensitive data handled by mobile apps, whether stored privately or publicly, is properly protected. This aligns with the ENISA's emphasis on utilizing technologies that enhance privacy by minimizing, anonymizing, and securing personal data. Both stress the importance of safeguarding sensitive information by implementing appropriate measures, which suggests a conceptual correlation between the two guidelines.
- MASVS-STORAGE-2: Both the MASVS-STORAGE-2 control and the ENISA Guideline emphasize the importance of handling sensitive data with care to prevent unintentional exposure and enhance data privacy. MASVS-STORAGE-2 specifically addresses the issue of unintentional leaks due to mishandling APIs or system capabilities, while the ENISA guideline focuses more broadly on employing technologies that minimize, anonymize, and secure personal data. The underlying principle in both cases is the prevention of sensitive data exposure and improving data privacy and security, showing a correlation between the two.

8.15 Implementation Guidance (ENISA 7.15):

ENISA Secure Smartphone Development Guidance (7.15): The default settings of the application should provide maximum privacy and security protection for the user.

8.15.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA guideline on default settings is that both are concerned with ensuring security and privacy for the user. ”MASVS-AUTH-1” specifically deals with authenticated and authorized access to remote endpoints and enforces best practices for secure communication protocols. This is in alignment with the ENISA guideline, which emphasizes that applications should have maximum privacy and security protection by default. A robust authentication and authorization mechanism is a fundamental aspect of securing an app and protecting user privacy, hence they are correlated.
- MASVS-AUTH-2: MASVS-AUTH-2 addresses the implementation of local authentication mechanisms, such as biometrics or a local PIN code, which are directly related to the privacy and security protection of the user. Ensuring these authentication methods are correctly implemented is crucial to prevent unauthorized access and protect user data, aligning with the ENISA guideline of providing maximum privacy and security by default. Proper implementation of local authentication contributes to the overall security posture of an app, preventing potential privacy breaches and securing user data, which is the essence of the ENISA guideline.
- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3” and the ENISA guideline on default privacy and security settings reflects a shared emphasis on enhancing security measures, particularly for sensitive actions within an application. MASVS-AUTH-3 addresses the need for additional authentication methods which reinforce security by verifying the user’s identity in more robust ways, such as biometric, pin, or multifactor authentication. These methods align with the ENISA guideline’s principle of maximizing user protection by default, as they constitute stronger security practices that should, ideally, be enabled by default in the app to better safeguard sensitive transactions and user data.
- MASVS-CODE-1: The MASVS-CODE-1 requirement for apps to support only up-to-date mobile OS versions aligns with the ENISA guideline for default settings to provide maximum privacy and security. By ensuring the app runs on a current platform with the latest security patches, the app leverages the default security and privacy improvements provided by the operating system, thereby offering better protection for the user, which is in line with the intention of the ENISA guideline.
- MASVS-CODE-2: The potential correlation between ”MASVS-CODE-2” and the ENISA guideline about default settings providing maximum privacy and security protection is that both are concerned with the security and protection of the user. The MASVS-CODE-2 control addresses the need for a mechanism to force updates when critical vulnerabilities are discovered, which ensures that users are not able to continue using a compromised version of the app that might put their privacy and security at risk. This aligns with the ENISA guideline which emphasizes that the default settings of an application should prioritize user privacy and security, implicitly supporting the idea that mandatory updates are a method of maintaining a secure environment by default.

- MASVS-CODE-3: While ”MASVS-CODE-3” does not explicitly mention default application settings, it emphasizes the importance of a thorough security assessment including third-party components to identify and mitigate known vulnerabilities. This aligns with the ENISA guideline’s principle that applications should be designed to offer maximum privacy and security by default, which includes ensuring that any included components do not introduce known security risks. Ensuring that libraries and third-party elements don’t harbor known vulnerabilities is a step towards providing secure defaults as recommended by ENISA.
- MASVS-CODE-4: The correlation between ”MASVS-CODE-4” and the ENISA guideline about default settings providing maximum privacy and security protection is that both points are concerned with ensuring the security and integrity of the application and user data. ”MASVS-CODE-4” emphasizes treating all incoming data as untrusted and taking steps to verify and sanitize it to prevent security vulnerabilities like injection attacks and bypasses, which directly contributes to an application’s security posture. The ENISA guideline suggests that default settings should prioritize user privacy and security, which complements the goal of ”MASVS-CODE-4” by ensuring that, from the outset, the application is configured to minimize risks and protect the user data against threats, including those that arise from insecure data handling. Together, they both aim to enhance the overall security and privacy of the application from different angles.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the ENISA Guideline regarding default settings for privacy and security is evident in their mutual emphasis on securing user data, particularly through the use of cryptography and secure default settings. ”MASVS-CRYPTO-1” explicitly mentions the importance of cryptography in securing user data on mobile devices, where physical access is a more common threat vector. The ENISA Guideline dictates that default settings should offer maximum privacy and security, which implicitly includes the use of strong cryptography to protect user data as a part of the application’s default configuration. Both the standard and the guideline aim to safeguard the user’s information from unauthorized access and align in their approach to ensure user security and privacy.
- MASVS-CRYPTO-2: Both ”MASVS-CRYPTO-2,” which emphasizes the importance of managing cryptographic keys throughout their entire lifecycle, and the ENISA guideline on default application settings providing maximum privacy and security protection, are aligned in their focus on ensuring strong security practices to protect user data. Proper key management is an essential aspect of maintaining privacy and security, as even strong cryptographic algorithms can be rendered ineffective if keys are mishandled. The ENISA guideline underlines the importance of safety by default, which encompasses secure key management as a fundamental part of providing robust security settings out-of-the-box.
- MASVS-NETWORK-1: The ”MASVS-NETWORK-1” requirement aligns with the ENISA guideline because both emphasize the importance of default settings that prioritize privacy and security for the user. ”MASVS-NETWORK-1” specifically mentions ensuring data privacy and integrity through secure network communications, which is a critical aspect of providing privacy and security protection by default. The effort to prevent developers from disabling secure defaults is also directly in support of the ENISA guideline to provide maximum protection through the application’s default settings.
- MASVS-NETWORK-2: The MASVS-NETWORK-2 control aligns with the ENISA guideline on default settings providing maximum privacy and security because it advocates for limiting trust to specific Certificate Authorities (CAs) rather than accepting all default root CAs. This practice of certificate pinning enhances security by reducing the attack

surface and preventing attacks related to rogue or compromised CAs, thus aligning with the principle of privacy and security by default.

- MASVS-PLATFORM-1: The correlation between "MASVS-PLATFORM-1" and the stated ENISA Guideline is that both are focused on ensuring secure interactions and protection for the user. "MASVS-PLATFORM-1" is concerned with secure Inter-Process Communication (IPC) mechanisms, which refers to how different parts of an application or different applications on the same device communicate with each other. Ensuring that these interactions are secure protects the user's data from unauthorized access, which aligns with the ENISA guideline advocating for maximum privacy and security protection by default. Both emphasize the principle of secure by design, which implies that security and privacy should be integrated into the system from the ground up.
- MASVS-PLATFORM-2: The correlation exists because "MASVS-PLATFORM-2" discusses the need for secure configuration of WebViews to prevent sensitive data leakage and exposure of sensitive functionalities, which aligns with the ENISA guideline that emphasizes default settings should maximize privacy and security protection. Securely configuring WebViews is part of providing secure default settings for the application.
- MASVS-PLATFORM-3: The correlation between "MASVS-PLATFORM-3" and the ENISA Guideline about default settings providing maximum privacy and security protection is that both are concerned with the protection of sensitive user data. "MASVS-PLATFORM-3" addresses the scenarios in which sensitive data might be unintentionally exposed due to platform behaviors like auto-generated screenshots or visual disclosures. It implies that there should be mechanisms in place to prevent such leaks, aligning with the ENISA principle that the application's default settings should prioritize the user's privacy and security, hence ensuring that sensitive information is safeguarded by default and reducing the risk of data exposure.
- MASVS-PRIVACY-1: There is a correlation between "MASVS-PRIVACY-1" and the ENISA Guideline regarding default privacy settings. Both concepts advocate for the principle of data minimization and the protection of user privacy. MASVS-PRIVACY-1 emphasizes that apps should only request access to data they absolutely need and stresses the importance of informed user consent. This is in line with the ENISA guideline that suggests default application settings should offer maximum privacy and security protection, which includes avoiding unnecessary data access permissions by default. Both guidelines aim to ensure that user data is handled responsibly and with regard to user privacy preferences, thereby reducing the potential risks associated with data breaches or improper data handling.
- MASVS-PRIVACY-2: Both MASVS-PRIVACY-2 and the ENISA Guideline emphasize protecting user privacy. MASVS-PRIVACY-2 focuses on techniques such as data abstraction, anonymization, and pseudonymization to prevent identification and tracking, while the ENISA Guideline advises that default app settings should prioritize maximum privacy and security for the user. Both address the aim of designing applications in a manner that inherently respects and protects user privacy, implying a correlation between the principles advocated by each.
- MASVS-PRIVACY-3: MASVS-PRIVACY-3 and the ENISA Guideline both emphasize the protection of user privacy. MASVS-PRIVACY-3 requires clear information about data practices and adherence to platform data declarations, which aligns with the ENISA Guideline's principle that default settings should offer maximum privacy and security for the user. Both standards aim to enhance user awareness and control over their data and to ensure that users are not subjected to unexpected or non-transparent data handling practices.

- MASVS-PRIVACY-4: The correlation exists because both statements emphasize the user's control over their personal data and their privacy. "MASVS-PRIVACY-4" details specific mechanisms and behaviors that allow users to manage their data and privacy settings, while the ENISA guideline advocates for applications to have default settings that offer maximum privacy and security. The alignment is in the underlying principle that both guidelines are designed to protect user privacy proactively.
- MASVS-RESILIENCE-1: The description of "MASVS-RESILIENCE-1" emphasizes the importance of an application running on a secure, unmodified platform in order to maintain the integrity and security of the app's data and leverage platform-specific security features such as secure storage, biometrics, and sandboxing. This closely correlates with the ENISA Guideline that stipulates the application's default settings should offer maximum privacy and security protection. Ensuring that the operating system has not been compromised directly supports the notion of providing strong security settings by default, as any tamper with the platform could undermine the app's security and privacy measures.
- MASVS-RESILIENCE-2: The correlation exists in that MASVS-RESILIENCE-2 aims to protect the integrity and functionality of an app on a user-controlled device, focusing on security from modifications that could undermine it. Such protections inherently align with the ENISA guideline that default settings should prioritize user privacy and security. By ensuring the app is resilient to unauthorized changes, it directly contributes to security protection, which is a core aspect of the ENISA guideline's focus on default settings providing maximum privacy and security. The measures implied by MASVS-RESILIENCE-2 would generally enhance the security posture by default, reinforcing the ENISA principle.
- MASVS-RESILIENCE-3: MASVS-RESILIENCE-3 which deals with impeding comprehension of an app's internals to prevent tampering aligns with the ENISA guideline that recommends default settings to provide maximum privacy and security. By making static analysis difficult through various obfuscation techniques, the app's default posture is more resilient to unauthorized access and modifications, which contributes to both privacy and security by protecting the app from being reverse-engineered and potentially exploited.
- MASVS-RESILIENCE-4: Both MASVS-RESILIENCE-4 and the ENISA guideline emphasize the importance of security by default. MASVS-RESILIENCE-4 focuses on making dynamic analysis and code runtime modification difficult, which indirectly ensures that the application provides better security protection against runtime attacks and tampering. The ENISA guideline insists on having maximum privacy and security protection by default, which aligns with the goal of MASVS-RESILIENCE-4 to prevent vulnerability exploits and enhance app resilience. Both require app developers to consider security and privacy measures from the onset, making them difficult to bypass.
- MASVS-STORAGE-1: The correlation between "MASVS-STORAGE-1" and the described ENISA Guideline regarding default settings providing maximum privacy and security is that both are emphasizing the importance of protecting sensitive data within the application environment. MASVS-STORAGE-1 focuses on ensuring that sensitive data isn't just protected in private locations but also when stored in locations that may be accessible by the user or other apps. This aligns with the ENISA guideline's principle that default settings should be geared towards maximum protection, ensuring privacy and security by default which would include secure handling and storage of sensitive data.
- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" and the ENISA Guideline about default settings providing maximum privacy and security is evident. MASVS-STORAGE-2 addresses the issue of unintentional storage or exposure of sensitive data in publicly accessible locations, which is a direct risk to user privacy and security. Ensuring that sensitive data is not unintentionally leaked aligns with the ENISA principle

of having default settings that protect user privacy and security to the fullest extent possible. Following MASVS-STORAGE-2 would help mitigate risks and ensure that the app's default behavior does not compromise sensitive information, adhering to the ENISA guideline's intention.

Chapter 9
Protect paid resources

Smartphone applications give programmatic access to paid resources on mobile phones such as phone calls, SMS, phone calls and SMS to premium numbers, roaming data, NFC payments, and third party payment systems. Applications that integrate those services must take particular care to prevent abuse. Developers have to consider the financial impact of vulnerabilities in their application. Furthermore, applications that implement In-Application payment for selling services to the user must protect their payment code against abuse.

9.1 Implementation Guidance (ENISA 8.1):

ENISA Secure Smartphone Development Guidance (8.1): Maintain logs of access to paid-for resources in non-repudiable format (e.g., a signed receipt sent to a trusted server backend - with user consent) and make them available to the end-user for monitoring. Logs should be protected from unauthorized parties.

9.1.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: MASVS-AUTH-1 discusses the necessity for apps to follow best practices in authentication and authorization when connecting to remote endpoints, which inherently includes securely managing user access to resources. The ENISA Guideline on maintaining logs of access to paid-for resources in a non-repudiable format aligns with this requirement as it is an important aspect of securely enforcing authorization and providing accountability. Both standards emphasize the importance of secure and verifiable access controls, with MASVS focusing on broader best practices while ENISA provides a specific method to ensure such access is logged and verifiable.
- MASVS-CRYPTO-1: The MASVS-CRYPTO-1 requirement emphasizes the importance of cryptography for securing user data, especially in scenarios where physical access to a device is possible. The ENISA guideline recommends maintaining logs of access to paid resources in a non-repudiable format, which implies the use of cryptographic mechanisms such as digital signatures to ensure the authenticity and integrity of the logs. Both guidelines highlight the need for cryptographic controls to protect sensitive information and provide security assurances, correlating in their emphasis on the proper use of cryptography for securing data and confirming events/actions.
- MASVS-CRYPTO-2: The correlation between ”MASVS-CRYPTO-2” and the ENISA guideline on maintaining logs in a non-repudiable format hinges on the principle of key management and ensuring cryptographic operations are secure. While MASVS-CRYPTO-2 focuses on the lifecycle management of cryptographic keys, which includes their generation, storage, and protection, the ENISA guideline addresses the integrity and non-repudiation of transaction logs, which would inherently involve the use of cryptographic keys to sign and securely transmit data. Proper key management is essential to maintain the security of log entries and to satisfy the requirement for logs being in a non-repudiable format. Without secure key management as described by MASVS-CRYPTO-2, the cryptographic measures mentioned in the ENISA guideline could not be effectively implemented, thereby compromising the integrity and non-repudiation of the logs.
- MASVS-PRIVACY-1: Both ”MASVS-PRIVACY-1” and the ENISA guideline emphasize the importance of informed user consent for data access and sharing. ”MASVS-PRIVACY-1” focuses on data minimization, restricting access, and avoiding third-party SDKs from collecting data without consent, whereas the ENISA guideline specifies logging access to paid resources with user consent and protecting logs from unauthorized access. Both stress accountability and the protection of user data, showing a clear correlation in their underlying principles related to privacy and security.
- MASVS-PRIVACY-2: Both ”MASVS-PRIVACY-2” and the ENISA Guideline emphasize the importance of protecting user privacy and ensuring that user data is managed in a secure and privacy-preserving manner. ”MASVS-PRIVACY-2” focuses on using techniques that prevent user identification and tracking, thereby protecting user identity. It includes mea-

sures like data abstraction, anonymization, and pseudonymization. Similarly, the ENISA Guideline underlines the need to maintain logs in a secure, non-repudiable format with user consent and to protect these logs from unauthorized access. Both sets of recommendations are concerned with safeguarding user information while allowing for its legitimate usage, such as fraud detection or access monitoring, without compromising privacy.

- MASVS-PRIVACY-3: Both MASVS-PRIVACY-3 and the ENISA guideline highlighted focus on ensuring user rights regarding their data. MASVS-PRIVACY-3 emphasizes that users must be informed about how their data is utilized, highlighting transparency in data collection, storage, and sharing. Similarly, the ENISA guideline insists on maintaining logs of access to paid-for resources in a secure manner and making those logs available to the user, which is also an aspect of transparency and user control over their data. Both guidelines aim to enhance user trust and security by keeping users informed and involved in the data management process.
- MASVS-PRIVACY-4: MASVS-PRIVACY-4 and the described ENISA guideline align in that both emphasize user control over their data. MASVS-PRIVACY-4 focuses on providing mechanisms for data management, consent revocation, and transparency, while the ENISA guideline suggests logging access to resources with user consent and ensuring logs are available for user monitoring, hence reinforcing the control and transparency aspects.
- MASVS-RESILIENCE-2: The correlation is that both ”MASVS-RESILIENCE-2” and the ENISA guideline revolve around maintaining the integrity and security of the app and its resources. MASVS-RESILIENCE-2 focuses on the integrity of the app’s code and resources to prevent unlawful modifications or misuse, while the ENISA guideline suggests maintaining non-repudiable logs to track access to paid resources, also as a security measure to prevent unauthorized use. Both aim to deter and detect tampering or unauthorized access, albeit through different methods.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA Guideline is apparent in the shared focus on preventing unintentional disclosure of sensitive data. While MASVS-STORAGE-2 addresses general prevention of sensitive data leaks due to API usage or system features like backups or logs, the ENISA Guideline specifically mentions maintaining access logs in a secure, non-repudiable format and protecting them from unauthorized access—which is a specific instance of preventing data leaks as prescribed by MASVS-STORAGE-2. Both aim to ensure that sensitive data, be it from unintentional storage or access logs, is properly handled and secured.

9.2 Implementation Guidance (ENISA 8.2):

ENISA Secure Smartphone Development Guidance (8.2): Check for anomalous usage patterns in paid-for resources usage and trigger re-authentication (e.g., when significant change in location, user-language changes, significant higher paid-for service usage).

9.2.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The MASVS-AUTH-1 guideline on user authentication and enforcement of authorization best practices is correlated with the ENISA Guideline on checking for anomalous usage patterns and triggering re-authentication. Both guidelines pertain to monitoring and ensuring secure authentication and are aimed at detecting and responding to unusual or potentially unauthorized user behaviors that could indicate security risks. The MASVS-AUTH-1's emphasis on following best practices for secure protocol use includes mechanisms that could help in detecting abnormalities in usage patterns, aligning with the ENISA guideline's recommendation for re-authentication in such cases.
- MASVS-AUTH-3: Both "MASVS-AUTH-3" and the ENISA Guideline emphasize the importance of additional authentication measures for sensitive actions or anomalies in application usage. "MASVS-AUTH-3" advocates for extra authentication types like biometrics or MFA for critical operations within an application, while the ENISA Guideline recommends re-authentication in the event of unusual usage patterns, such as a significant change in the user's location or language or a spike in paid resource consumption. Both guidelines are concerned with enhancing security by verifying that users performing sensitive or unusual actions are indeed authorized.
- MASVS-CODE-2: Both "MASVS-CODE-2" and the ENISA guideline pertain to the concept of responding to events or conditions that may suggest a security risk for the app and its users. "MASVS-CODE-2" focuses on a mechanism to ensure users update the app when critical vulnerabilities are identified, thus mitigating the risk of such vulnerabilities being exploited. The ENISA guideline suggests monitoring for unusual usage patterns that could indicate a security compromise, such as a change in location or language, and then responding by triggering a re-authentication. Both imply an operational procedure designed to protect the app's security posture through active monitoring and response, by either forcing an update or re-authentication, to address potential security threats.
- MASVS-PLATFORM-2: The correlation exists in the context of security considerations for apps with increased control over the UI, such as WebViews, and the prevention of sensitive data leakage. MASVS-PLATFORM-2 addresses the need to configure WebViews securely, which includes being vigilant against unexpected user behaviors that could indicate a security breach or misuse. The ENISA guideline's recommendation to check for anomalous usage patterns fits within this broader security context, as it is another layer of defensive measures against the compromise of user data or resources. Both MASVS-PLATFORM-2 and the ENISA guideline concern themselves with maintaining a secure environment within the application to protect against potential threats and data breaches.
- MASVS-PRIVACY-2: Both the MASVS-PRIVACY-2 control and the ENISA Guideline are aimed at enhancing user privacy and security. The MASVS-PRIVACY-2 control does this by employing techniques such as data abstraction, anonymization, and pseudonymization to prevent user identification and tracking, ensuring that signals like device IDs, IP addresses, and behavioral patterns are used only for their intended purposes. On the other hand, the

ENISA Guideline suggests monitoring for anomalous usage patterns as a means to protect the user's account from unauthorized access or fraud. Both share the theme of protecting user identity and privacy, albeit through different mechanisms—one through data handling and the other through usage monitoring and re-authentication.

- MASVS-PRIVACY-3: Both "MASVS-PRIVACY-3" and the ENISA guideline mentioned focus on protecting user data and ensuring transparency and security regarding how an app uses data. "MASVS-PRIVACY-3" emphasizes users' rights to understand their data's usage, which aligns with the ENISA guideline's intent to monitor for unusual patterns that could indicate unauthorized usage of user data, prompting further verification.
- MASVS-RESILIENCE-1: While MASVS-RESILIENCE-1 from the Mobile Application Security Verification Standard (MASVS) focuses on verifying that the operating system on which a mobile app runs has not been compromised to ensure the reliance on platform security features (such as secure storage, biometrics, and sandboxing), it can be correlated with the ENISA guideline advocating for checking anomalous usage patterns in paid-for resource usage to trigger re-authentication. Both controls aim to enhance security by addressing changes in the expected environment or usage patterns that might indicate potential security breaches. The MASVS control seeks to ensure the platform's integrity, while the ENISA guideline aims to detect anomalous behavior possibly caused by a compromised platform or user account, which are situations that could be facilitated by a lack of platform integrity. Detecting and reacting to anomalies is thus related to trusting the security of the underlying platform.

9.3 Implementation Guidance (ENISA 8.3):

ENISA Secure Smartphone Development Guidance (8.3): Consider a white-list model by default for paid-for resources addressing e.g., address book contacts only unless specifically authorized for phone calls.

9.3.1 OWASP MASVS MAPPING

- MASVS-AUTH-3: The Mobile Application Security Verification Standard (MASVS) guideline MASVS-AUTH-3, which emphasizes secure implementation of additional forms of authentication for sensitive actions inside an app, correlates with the ENISA Guideline suggesting a whitelist model for protecting paid-for resources and limiting access, such as to the address book or phone calls, to specifically authorized entities. The underlying principle in both guidelines is to enhance security by ensuring that sensitive actions and resources are protected through additional verification mechanisms and explicit user permissions, thereby preventing unauthorized access or misuse.
- MASVS-CODE-2: The MASVS-CODE-2 control which discusses the mechanism for forcing users to update the app to mitigate critical vulnerabilities correlates with the ENISA guideline about considering a white-list model for sensitive resources. The underlying principle in both is to protect the users and the system by ensuring that only safe, authorized operations are performed – in MASVS-CODE-2 through mandatory updates to fix vulnerabilities, and in the ENISA guideline through strict access control to sensitive resources. Both aim to minimize security risks.
- MASVS-CODE-4: Both ”MASVS-CODE-4” and the ENISA Guideline mentioned focus on the principle of treating user inputs as untrusted and therefore emphasize the need for input validation and sanitization. ”MASVS-CODE-4” underlines the importance of verifying and sanitizing all data entry points to prevent security vulnerabilities like injection attacks. Similarly, the ENISA guideline suggests using a white-list model for access to paid-for resources, which is a form of input validation that allows only specifically authorized actions. Both advocate for a default stance that considers inputs as untrusted unless verified, to enhance the security of the application.
- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1”, which emphasizes the importance of cryptography in securing user data on mobile devices, and the ENISA Guideline advising a white-list model for accessing paid-for resources, is that they both advocate for protective measures to ensure data security and privacy. The MASVS-CRYPTO-1 focuses on using cryptography to protect data, especially from physical access threats, while the ENISA guideline suggests limiting access to sensitive resources unless explicitly authorized, which indirectly supports the goal of data protection. Both guidelines are designed to minimize unauthorized access and ensure that user data is only accessed under strict security controls.
- MASVS-NETWORK-1: The correlation between ”MASVS-NETWORK-1” and the ENISA Guideline is that both advocate for measures to protect user data and ensure security in mobile applications. ”MASVS-NETWORK-1” focuses on the importance of securing data in transit through encryption and endpoint authentication, aiming to prevent developers from unintentionally creating vulnerabilities by disabling secure defaults or using potentially unsecure third-party libraries. The ENISA Guideline suggests employing a white-list model for accessing paid-for resources, implying that strict access controls should be in

place to protect sensitive resources like the address book from unauthorized access or use, such as making phone calls without explicit permission. Both guidelines emphasize the need for strict security measures to prevent unauthorized data access and ensure privacy and integrity.

- MASVS-NETWORK-2: The correlation between "MASVS-NETWORK-2" and the ENISA Guideline is that both controls are about restricting trust to a narrow scope for security purposes. "MASVS-NETWORK-2" suggests trusting only specific Certificate Authorities (CAs) rather than all default root CAs, which is a form of white-listing trusted entities—in this case, for securing network communications via certificate pinning. The ENISA Guideline recommends a white-list model for paid-for resources, where access is granted only to specifically authorized functions, much like how certificate pinning grants trust only to specific CAs. Both are proactive security measures to minimize risk by limiting access or trust to predefined entities.
- MASVS-PLATFORM-1: The correlation between "MASVS-PLATFORM-1" and the ENISA guideline is that both are emphasizing secure interaction with potentially sensitive resources or features on a mobile platform. "MASVS-PLATFORM-1" is about ensuring secure interactions involving Inter-Process Communication (IPC) mechanisms which can be used to expose data or functionality. Meanwhile, the ENISA guideline suggests using a white-list model for accessing paid-for resources such as address book contacts and making phone calls, only granting access when specifically authorized. Both stress the importance of securing and restricting access to app features to prevent unauthorized or malicious use, which aligns with the principle of least privilege and proactive security measures.
- MASVS-PLATFORM-2: The correlation exists because both MASVS-PLATFORM-2 and the ENISA guideline emphasize the importance of secure configuration to prevent unauthorized access to sensitive data or functionalities. MASVS-PLATFORM-2 focuses on the secure configuration of WebViews and limiting exposure through JavaScript bridges, which parallels the ENISA guideline's recommendation to use a whitelist model for controlling access to paid resources, like the address book, and limiting functionalities, such as making phone calls, to those explicitly authorized. Both aim to enhance security by granting permissions and access selectively.
- MASVS-PRIVACY-1: The correlation exists as both MASVS-PRIVACY-1 and the ENISA Guideline emphasize the principle of least privilege and user consent in the context of data access. MASVS-PRIVACY-1 focuses on data minimization and informed consent for app functionality, including the handling of third-party SDKs and supply chain transparency. Similarly, the ENISA Guideline suggests a white-list model, which implies granting access only to resources that are explicitly authorized, such as contacts, and no more than necessary (e.g., not authorizing phone calls unless specifically allowed). Both highlight the idea that apps should not access more data than needed and must respect user permissions and consent.
- MASVS-PRIVACY-3: The correlation between "MASVS-PRIVACY-3" and the described ENISA Guideline is that both emphasize the importance of transparency and user control over their data. MASVS-PRIVACY-3 mandates clear information to users about data practices, which aligns with the ENISA Guidelines' call for a white-list model where user data, such as address book contacts, is not accessed unless explicitly authorized. Both guidelines aim to protect users' privacy and ensure that apps do not access or utilize user data without consent.
- MASVS-PRIVACY-4: The statement "MASVS-PRIVACY-4" and the ENISA guideline both promote user control over their data. MASVS-PRIVACY-4 advocates for users to have mechanisms to manage their data and privacy settings, while the ENISA guideline

recommends a white-list model for resources, meaning users must explicitly grant permission. Both stress the importance of consent and user authority over data access.

- MASVS-RESILIENCE-2: Both concepts deal with protecting the integrity and intended functionality of an app. The statement regarding ”MASVS-RESILIENCE-2” is focused on ensuring the integrity and unmodified state of an app’s original code and resources. This is to prevent cheating, unauthorized enabling of premium features, or uploading a backdoored app to third-party stores. The ENISA guideline about considering a white-list model for paid-for resources is also about protecting the app’s resources from unauthorized access or use. A white-listing approach means only allowing access to certain functions or resources, like address book contacts, when the user gives explicit permission. This can prevent unauthorized actions, such as making phone calls without permission which can also be seen as ensuring the app works as intended and is not modified to perform functions that the user and the app developers have not authorized. Both statements reflect a concern for maintaining the app’s intended functionality and user security, which means there is a correlation between ”MASVS-RESILIENCE-2” and the described ENISA guideline.

9.4 Implementation Guidance (ENISA 8.4):

ENISA Secure Smartphone Development Guidance (8.4): Warn user and obtain consent for any cost implications for app behaviour.

9.4.1 OWASP MASVS MAPPING

- MASVS-PRIVACY-1: Both MASVS-PRIVACY-1 and the ENISA Guideline focus on the importance of obtaining informed consent from the user before accessing their data or performing actions that may have cost implications. MASVS-PRIVACY-1 emphasizes the app's responsibility to request only the necessary data for its functionality and to ensure that any third-party SDKs operate based on user consent. Similarly, the ENISA Guideline stresses the need to warn users and obtain consent for actions that could have cost implications, which also aligns with the concept of informed consent and user awareness. Both guidelines highlight the ethical and legal necessity of transparency and user control over their data and potential costs.
- MASVS-PRIVACY-3: Both the MASVS-PRIVACY-3 requirement and the ENISA guideline emphasize the importance of clear user communication and consent regarding app behavior that could impact the user, particularly in ways they may not expect. MASVS-PRIVACY-3 focuses on ensuring users are informed about data use, while the ENISA guideline specifies warning and consent for costs, which can be considered a subset of the data use information users should be aware of according to MASVS-PRIVACY-3. Both aim to give users control and transparency over how the app interacts with them or their resources.
- MASVS-PRIVACY-4: There is a correlation between "MASVS-PRIVACY-4" and the ENISA Guideline related to warning users and obtaining consent for any cost implications for app behavior. While MASVS-PRIVACY-4 emphasizes users' control over their data, allowing them to manage, delete, and modify their data and change privacy settings, it implicitly includes the need for user consent when the data usage changes—especially if more data than initially specified is required. Similarly, the ENISA Guideline insists on warning users and obtaining consent specifically regarding cost implications of app behavior. Both guidelines are concerned with ensuring that users are informed and have consented to particular aspects of how apps interact with their personal data or incur costs on their behalf. Thus, they are correlated in their underlying principle that users must be actively informed and give consent before certain app behaviors that could impact them, whether it's about personal data or financial costs.

9.5 Implementation Guidance (ENISA 8.5):

ENISA Secure Smartphone Development Guidance (8.5): Applications have to take into account that different operating system versions provide different levels of access control (e.g., Android permissions) for various system resources. Specifically paid resources have different levels of access control depending on the version of the OS. Applications have to implement access control for those resources to prevent abuse of the application's access to those resources due to missing access control in older/newer versions of the OS and and/or application framework.

9.5.1 OWASP MASVS MAPPING

- MASVS-AUTH-3: Both statements address the need for secure implementation of additional authentication mechanisms and appropriate access control in mobile applications. The MASVS-AUTH-3 description highlights the importance of secure implementation of various forms of additional authentication for sensitive actions within an app, such as biometrics or multi-factor authentication (MFA). Concurrently, the ENISA Guideline emphasizes the importance of adapting and implementing access control mechanisms for different levels of operating system access, especially in the context of varying versions that may offer different degrees of access control. Considering that both statements are concerned with secure authentication practices and access control measures to prevent abuse or security breaches, there is a correlation between them. They both acknowledge the evolving nature of security in different OS versions and suggest that applications should enforce their own access restrictions to safeguard against potential security shortfalls in the operating system or application framework.
- MASVS-CODE-1: The correlation between "MASVS-CODE-1" and the ENISA Guideline is that both advocate for the consideration of the security implications of supporting multiple operating system versions within a mobile application. "MASVS-CODE-1" emphasizes the importance of running apps on up-to-date platform versions to ensure users benefit from the latest security protections. Similarly, the ENISA Guideline underlines the need for applications to be aware of the varying levels of access control provided by different operating system versions, especially regarding access to system resources. Both emphasize the potential vulnerabilities that may arise from supporting older OS versions due to lack of access control and the importance of updating to leverage enhanced security features.
- MASVS-PLATFORM-1: The correlation exists because both the MASVS (Mobile Application Security Verification Standard) requirement "MASVS-PLATFORM-1" and the ENISA (European Union Agency for Cybersecurity) guideline emphasize the importance of secure Inter-Process Communication (IPC) and access control within mobile applications across different operating system versions. "MASVS-PLATFORM-1" focuses on ensuring secure interactions with IPC mechanisms provided by the platform, which could involve accessing shared resources securely. The ENISA guideline underlines the requirement for applications to implement access control for system resources, taking into consideration varying levels of access control available across different OS versions, especially to prevent abuse in versions with less stringent access controls. Both stress the need to handle permissions and resource access securely within the application code to mitigate the risks associated with different access control mechanisms across operating system versions.

- MASVS-PLATFORM-2: Both the MASVS-PLATFORM-2 statement and the ENISA Guideline highlight the importance of proper configuration and access control in mobile applications to prevent sensitive data leakage and misuse of app functionalities or system resources. The MASVS-PLATFORM-2 focuses on the secure configuration of WebViews and the prevention of data leakage and exposure through them, while the ENISA Guideline emphasizes the need for apps to take into account varying levels of OS access control across different versions, particularly for paid resources, to prevent abuse due to insufficient access controls. Both statements underline the necessity of implementing robust access controls to protect against potential security issues arising from the application's interaction with the operating system.
- MASVS-PRIVACY-1: Both the MASVS-PRIVACY-1 guideline and the ENISA guideline emphasize the importance of appropriate access control to system resources and data minimization based on the principle of least privilege. MASVS-PRIVACY-1 mentions that apps should only request access to the data they absolutely need for their functionality, which aligns with the ENISA guideline that stresses the need for applications to implement access control to prevent abuse due to missing access control in different versions of operating systems. Both guidelines aim to ensure that applications do not overstep their necessary data access, thereby protecting users' privacy and information.
- MASVS-PRIVACY-2: The concept of implementing access control for system resources as outlined in the ENISA Guideline correlates with "MASVS-PRIVACY-2" which is about protecting user identity through unlinkability techniques, and ensuring that data is only used for its intended purpose. Access control to resources is a means to ensure that user identity related data (like device IDs, IP addresses) is properly safeguarded across different operating system versions, in line with the data protection goals of "MASVS-PRIVACY-2".
- MASVS-PRIVACY-3: Both the Mobile Application Security Verification Standard (MASVS) PRIVACY-3 guideline and the ENISA Guideline address the concept of privacy and the appropriate use of access controls and permissions within mobile applications. The MASVS-PRIVACY-3 emphasizes the user's right to understand how their data is utilized by the application, which includes being transparent about data collection, storage, and sharing practices. It also references adhering to platform guidelines regarding data declarations, which implies a consideration for how the operating system manages data permissions. The ENISA Guideline discusses the importance of applications taking into account the varying levels of access control provided by different operating system versions, particularly regarding paid resources. It points to the necessity of an application implementing its own access controls to prevent the misuse of resources due to potential limitations in the operating system's access control, whether the limitations are due to older or newer OS versions. Both guidelines correlate in that they highlight the need for applications to responsibly manage data and system resources access to protect user privacy, although they approach the topic from slightly different perspectives. MASVS-PRIVACY-3 focuses on user awareness and platform adherence, while the ENISA Guideline focuses on compensating for variations in OS-level control to prevent abuse.
- MASVS-PRIVACY-4: Both the MASVS-PRIVACY-4 and the referenced ENISA Guideline emphasize the importance of user control over their data and the necessary mechanisms an application must implement to ensure proper access control. MASVS-PRIVACY-4 focuses on users being able to manage their data and consent within the app, while ENISA touches on the responsibility of the app to consider different OS versions' access control capabilities, which directly relates to how an app manages user permissions and access to resources, especially around paid resources. Both guidelines point towards designing apps

that respect user privacy and consent, and consider the varying levels of control provided by the platform to enforce such privacy.

- MASVS-RESILIENCE-1: Both the MASVS-RESILIENCE-1 control and the ENISA Guideline emphasize the importance of the operating system's integrity and security features in the context of mobile application security. The MASVS-RESILIENCE-1 control focuses on ensuring that the platform has not been compromised, which is crucial for relying on platform security features like secure storage, biometrics, and sandboxing. Meanwhile, the ENISA Guideline addresses the need for applications to implement access control for system resources, especially considering the varying levels of access control provided by different operating system versions. Both statements acknowledge that the security of the app is highly dependent on the integrity and security mechanisms of the underlying platform, and the need for applications to manage and handle these aspects appropriately across different OS versions.
- MASVS-RESILIENCE-2: The correlation between "MASVS-RESILIENCE-2" and the described ENISA Guideline is that both are concerned with securing the application against unauthorized changes or misuse, particularly around operating system level controls and access to resources. MASVS-RESILIENCE-2 focuses on preserving the integrity of the application's original code and resources to prevent cheating or enabling premium features illicitly. Similarly, the ENISA Guideline emphasizes the need for applications to implement their own access controls for system resources, especially considering the variance in operating system versions and their respective permission models, to prevent abuse through exploitation of the differences in access control provided by the OS and application frameworks. Both are measures to maintain an app's intended functionality and protect against modifications or access that could lead to security breaches or fraud.
- MASVS-STORAGE-1: The correlation between "MASVS-STORAGE-1" and the described ENISA Guideline exists in that both emphasize the importance of controlling access to sensitive data on a mobile device. The "MASVS-STORAGE-1" guideline deals with ensuring that sensitive data stored by the app is protected irrespective of where it is stored, while the ENISA Guideline highlights the need for apps to implement access control for system resources, which may vary across different OS versions. Both guidelines recognize the risks associated with improper handling and storage of sensitive data and the need for access control mechanisms to prevent abuse or unauthorized access, which is especially important in environments where operating system controls may differ or be insufficient.
- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" and the mentioned ENISA Guideline is that both are concerned with preventing unauthorized access to sensitive data due to the use of APIs or operating system capabilities. "MASVS-STORAGE-2" specifically addresses unintentional storage or exposure of sensitive data, which includes considerations for how different system capabilities (like backups or logs) might inadvertently leak data if not properly managed by the developer. Similarly, the ENISA Guideline emphasizes the need for applications to consider the varying levels of access control provided by different OS versions, especially regarding paid resources. It suggests that developers should implement access controls within their applications to mitigate potential abuse due to varying access control levels in the operating system and/or application framework, which may change across versions. Both guidelines highlight the developer's responsibility to safeguard against data access issues that could arise due to inconsistencies or changes in system behavior.

9.6 Implementation Guidance (ENISA 8.6):

ENISA Secure Smartphone Development Guidance (8.6): Follow the OS/device vendor guidelines for implementing In-App payment: (A) Implement validation of payment receipts on the backend server not on the device. (B) Pay specific attention when integrating payment acceptance from a third party wallet (wallet not integrated into the mobile OS).

9.6.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between "MASVS-AUTH-1" and the described ENISA Guideline is that both emphasize the importance of server-side validation and the implementation of secure communication protocols between the app and remote endpoints. MASVS-AUTH-1 pertains to user authentication and authorization processes, which should follow best practices and are typically enforced server-side for security reasons. Similarly, the ENISA Guideline insists on backend server validation of payment receipts rather than on-device validation, which aligns with the idea that sensitive operations such as payments and authorization mechanisms should not exclusively rely on the app (client-side) but should be securely managed and enforced server-side. This reduces the risk of tampering and exploitation by malicious actors.
- MASVS-AUTH-3: The MASVS-AUTH-3 requirement emphasizes the importance of implementing additional forms of authentication securely for sensitive actions within the app, including various methods like biometric, pin, or MFA, among others. The described ENISA guideline advises following the operating system or device vendor's guidelines for in-app payment implementation, including backend validation of payment receipts and careful integration of third-party wallets. The correlation is that both MASVS-AUTH-3 and the ENISA guideline stress on the security measures that need to be taken when dealing with sensitive operations within the app. While MASVS-AUTH-3 is broader and suggests secure implementation of supplementary authentication methods for sensitive actions, the ENISA guideline is more specific and concentrates on secure practices for in-app payment processes. They both converge on the theme of enhancing security to protect sensitive transactions and user data against potential threats.
- MASVS-CRYPTO-1: The Mobile Application Security Verification Standard (MASVS) "MASVS-CRYPTO-1" focuses on the general best practices of cryptography in the context of securing user data within a mobile environment, especially considering the likelihood of attackers gaining physical access. This aligns with the ENISA guideline which advises on best practices for secure implementation of in-app payments, such as performing validation of payment receipts on the backend server rather than on the device, to prevent manipulation or bypassing the validation if it were done client-side. Both MASVS-CRYPTO-1 and the ENISA guideline are concerned with implementing security measures to shield sensitive operations and data from potential attackers, acknowledging the higher risk due to the mobile context. MASVS-CRYPTO-1 provides a broad mandate for secure cryptographic practices while the ENISA guideline gives a specific application of such practices within the realm of in-app payments security.
- MASVS-NETWORK-1: The correlation between "MASVS-NETWORK-1" and the ENISA guideline about implementing In-App payments securely is based on the principle of protecting data privacy and ensuring the integrity of data in transit. "MASVS-NETWORK-1"

emphasizes the importance of setting up secure network connections to prevent leaks of sensitive information and unauthorized access, which aligns with the ENISA guideline's advocation for backend validation of payment receipts and careful integration of third-party wallets to protect financial transactions and sensitive payment data from being compromised. Both stress the need for following best practices in secure data transmission.

- MASVS-PLATFORM-1: The correlation exists because "MASVS-PLATFORM-1" focuses on ensuring secure interactions with IPC (Inter-Process Communication) mechanisms, which is a broader principle that would include secure handling of in-app payments, as referenced by the ENISA guideline. Payment processes often involve IPC mechanisms as they require communication between the app and external services, such as backend servers or third-party wallet services. Ensuring that IPC interactions are secure aligns with the ENISA guideline to implement receipt validation on the backend and to be cautious when integrating third-party wallets, both of which are measures to prevent security issues that could arise from insecure IPC.
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA guideline on implementing In-App payment can be observed in the focus on security and data protection. "MASVS-PLATFORM-2" emphasizes configuring WebViews securely to prevent sensitive data leakage and exposure of functionality, which is directly related to securing payment processes and sensitive financial transactions within an app. The ENISA guideline complements this by specifying that validation of payment receipts should be done on the backend server rather than the device, and by emphasizing caution when integrating third-party wallets. Both the MASVS requirement and the ENISA guideline aim to prevent exposure and misuse of sensitive functionalities, such as payment processes, through secure configurations and adherence to best practices.
- MASVS-PLATFORM-3: The described MASVS-PLATFORM-3 focuses on preventing involuntary leaks of sensitive data due to platform features such as screenshots. This relates to the ENISA Guideline about following OS/device vendor guidelines for secure in-app payments, which implicitly includes avoiding leaks of sensitive payment information, such as credit card details, during the payment process. While MASVS-PLATFORM-3 is broader and applies to all sensitive data displayed in the UI, the spirit of protecting sensitive payment information aligns with ENISA's recommendations to implement secure payment processing following vendor guidelines and to do sensitive operations like payment validation on the backend. Implementing MASVS-PLATFORM-3 would support ENISA's guideline by ensuring sensitive payment data displayed during transactions are not unintentionally exposed through platform features.
- MASVS-RESILIENCE-2: The Mobile Application Security Verification Standard (MASVS) "RESILIENCE-2" control and the ENISA Guideline regarding the implementation of in-app payments are correlated. Both are focused on promoting security measures to protect the integrity of the app and its payment functionalities from unauthorized modifications. MASVS-RESILIENCE-2 aims to prevent changes to an app by ensuring the integrity of its code and resources, which aligns with the ENISA Guideline's recommendations to implement payment validation on the backend server rather than on the device, minimizing the risk of tampering on user-controlled devices. By following both the MASVS control and ENISA's recommendations, developers can secure payment transactions and maintain the integrity of the in-app functionalities against modifications that could bypass payment requirements or implement fraudulent transactions.
- MASVS-RESILIENCE-3: The correlation between "MASVS-RESILIENCE-3" and the ENISA Guideline regarding In-App payments lies in the principle of obfuscation and the securing of sensitive operations. MASVS-RESILIENCE-3 emphasizes making it difficult

to understand an app's internals to impede tampering via static analysis. Following this control makes it harder for an attacker to reverse engineer the app to manipulate or bypass payment processes. Similarly, the ENISA guideline for In-App payment insists on server-side validation of payment receipts and careful integration of third-party wallets. This prevents attackers from manipulating or faking payment validations on the client-side, which is susceptible to tampering. Both guidelines are designed to enhance the resilience of the app against reverse engineering and tampering attacks, which could compromise payment transactions and lead to financial losses or unauthorized access to paid content or features.

- MASVS-RESILIENCE-4: The correlation between "MASVS-RESILIENCE-4" and the ENISA Guideline about implementing In-App payments is related to the objective of making it difficult for attackers to manipulate an application during runtime. The MASVS-RESILIENCE-4 control focuses on hardening the app against dynamic analysis and instrumentation, which could be used by attackers to modify code or behavior at runtime, potentially to bypass payment processing or receipt validation mechanisms. By following the ENISA Guideline, which recommends validating payment receipts on the backend server and not on the device, this directly aligns with the intent of MASVS-RESILIENCE-4 as it removes the validation process from the more vulnerable client-side environment. This makes it harder for an attacker to manipulate the payment validation process through dynamic analysis or runtime modification of the app, as the critical validation step occurs on the server, which is typically more secure and less susceptible to such attacks. Therefore, keeping the validation server-side is consistent with the goal of MASVS-RESILIENCE-4 to protect the app from dynamic manipulation, enhancing overall resilience against attacks targeting in-app payment processes.

Chapter 10
Secure software distribution

Overall software security on mobile devices is enforced by code signing and fast security updates. The use of secure practice for software distribution is paramount to the overall security of the application and it is fundamental to mitigate all risks described in these guidelines.

10.1 Implementation Guidance (ENISA 9.1):

ENISA Secure Smartphone Development Guidance (9.1): Applications must be designed and provisioned to allow updates for security patches, taking into account the requirements for approval by app-stores and the extra delay this may imply.

10.1.1 OWASP MASVS MAPPING

- MASVS-CODE-1: The MASVS-CODE-1 description emphasizes the importance of running apps on up-to-date platform versions to ensure users have the latest security protections, which correlates with the ENISA guideline that states applications must be designed to allow updates for security patches. Both stress the necessity of keeping the app updated to protect against known threats and vulnerabilities.
- MASVS-CODE-2: The ”MASVS-CODE-2” requirement correlates with the ENISA Guideline provided. Both emphasize the importance of having a mechanism in place to enable updates, particularly for the purpose of addressing security vulnerabilities. While MASVS-CODE-2 focuses on ensuring there's a mechanism to force users to update the app for continued use, the ENISA guideline stresses the need for the app design to accommodate security patch updates. Both requirements recognize the necessity of timely updates to mitigate security risks and imply that the update mechanism should preempt or quickly react to the discovery of critical vulnerabilities.
- MASVS-CODE-3: The correlation between ”MASVS-CODE-3” and the ENISA guideline about designing and provisioning applications to allow updates for security patches is based on their shared concern for application security, particularly in relation to third-party components and libraries. ”MASVS-CODE-3” addresses the importance of comprehensive assessments, including scanning libraries for known vulnerabilities which could be considered ”low-hanging fruit.” This aligns with the ENISA guideline's emphasis on the necessity for applications to be designed with the capability to receive security patches, which includes updating third-party libraries to resolve such known vulnerabilities. Both recognize the complexities and constraints involved in the update process, such as app store approvals and related delays.
- MASVS-PLATFORM-2: The correlation between ”MASVS-PLATFORM-2” and the ENISA Guideline lies in the theme of security within mobile applications. MASVS-PLATFORM-2 emphasizes the secure configuration of WebViews to prevent data leakage and exposure of sensitive functionalities. This secure configuration would naturally include keeping the WebView component updated with the latest security patches. The ENISA Guideline underlines the importance of designing applications in a way that allows for updates, specifically security patches, which is essential to maintaining the security measures that MASVS-PLATFORM-2 aims to enforce. Both statements highlight the critical role of regular updates and security patches in maintaining app security, which is why they are correlated.
- MASVS-PRIVACY-4: Although MASVS-PRIVACY-4 and the ENISA guideline mentioned are not directly addressing the same aspect of mobile application security, they share a correlation in emphasizing the importance of respecting user autonomy over personal data and ensuring app maintainability. MASVS-PRIVACY-4 focuses on user data control and privacy settings, which includes the ability to manage, delete, and modify data, and requires consent for additional data. The ENISA guideline concentrates on the app

design and update process, with a focus on security and the consideration of app store approval processes. Both statements reflect underlying principles related to protecting user privacy and ensuring the secure and ongoing operation of mobile applications, thereby presenting a connection between user consent, data management, and application maintenance for security purposes.

- MASVS-RESILIENCE-1: The ENISA Guideline that mentions ”Applications must be designed and provisioned to allow updates for security patches” correlates with ”MASVS-RESILIENCE-1” as both relate to maintaining the security integrity of the application and the platform it runs on. MASVS-RESILIENCE-1 emphasizes the importance of running an app on a secure, un-tampered platform, which includes the capability to receive and install security patches promptly. The guideline from ENISA complements this by underscoring the necessity for apps to be updateable, especially for security patches which may address vulnerabilities and prevent tampering or compromise of the operating system and platform security features. Both concepts are integral to ensuring the continued resilience and trustworthiness of the mobile application environment.

10.2 Implementation Guidance (ENISA 9.2):

ENISA Secure Smartphone Development Guidance (9.2): Official apps stores monitor apps for insecure code and are able to remotely remove apps at short notice in case of an incident. Distributing apps through official app-stores therefore provides a safety-net in case of serious vulnerabilities in your app.

10.2.1 OWASP MASVS MAPPING

- MASVS-CODE-2: The ENISA Guideline mentions that official app stores have the capability to monitor apps for insecure code and can take swift action, such as remotely removing apps if a serious vulnerability is found. This provides a safety net for addressing critical vulnerabilities in production apps. MASVS-CODE-2 also addresses the risk of vulnerabilities in production apps by ensuring there is a mechanism in place to force users to update the app, which indirectly relies on the app store's infrastructure to distribute the necessary updates for security issues. This requirement for a forced update mechanism correlates with the ENISA Guideline's emphasis on utilizing app store's capabilities for managing vulnerabilities in apps. Both are concerned with addressing and mitigating the risks associated with app vulnerabilities post-deployment.
- MASVS-CODE-3: Both the description of "MASVS-CODE-3" and the ENISA guideline emphasize the importance of security assessments and the handling of vulnerabilities. "MASVS-CODE-3" underlines the necessity of whitebox assessments and acknowledges that not all components (like third-party libraries) might be fully assessed, but stresses the importance of checking for known vulnerabilities, which are the "low-hanging fruit." The ENISA guideline touches on the safety measures of official app stores, which monitor for insecure code and can act quickly to mitigate incidents through app removal. There is a correlation in that both sources recognize the limitations of security assessments and the need for mechanisms to deal with vulnerabilities that slip through initial security checks.
- MASVS-CODE-4: The correlation between "MASVS-CODE-4" and the ENISA Guideline is that both address the importance of handling security vulnerabilities in mobile applications. "MASVS-CODE-4" focuses on the security of incoming data by ensuring that it is treated as untrusted input and is properly verified and sanitized. This is relevant since insecure handling of input can lead to classic injection attacks and other security issues. Meanwhile, the ENISA Guideline refers to how official app stores play a role in monitoring apps for such insecure code and can take quick action to mitigate incidents by removing vulnerable apps. This "safety-net" is complementary to the security practices outlined in "MASVS-CODE-4". While "MASVS-CODE-4" is about prevention, the ENISA guideline is about response—both are important in the security lifecycle of an app.
- MASVS-CRYPTO-1: The correlation between "MASVS-CRYPTO-1" and its description and the ENISA Guideline is that both emphasize the importance of security measures in mobile applications to protect user data. While "MASVS-CRYPTO-1" focuses on the implementation of cryptography best practices to secure data, especially against threats such as physical access to the device, the ENISA Guideline touches upon the role of app stores in monitoring and potentially removing apps that contain insecure code, which could include apps that fail to implement proper cryptography. In essence, both guidelines are geared towards preventing security vulnerabilities in apps that could compromise user data.

- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and its description, which emphasizes the importance of running apps on a secure and uncompromised platform, aligns with the ENISA Guideline that states official app stores monitor apps for security issues. Both highlight the reliance on platform-related security measures (MASVS) and the benefits of distributing apps through official channels (ENISA), which can enforce security by monitoring and removing insecure apps, thus contributing to maintaining platform integrity and user trust.
- MASVS-RESILIENCE-2: There is a correlation between ”MASVS-RESILIENCE-2” and the mentioned ENISA guideline. The MASVS-RESILIENCE-2 focuses on the integrity and resilience of the app against unauthorized modifications to its original code and resources, indicating the need for mechanisms to protect against such alterations which can be malicious. The ENISA guideline complements this by stating that official app stores monitor apps for insecure code and can act promptly, providing a layer of security that can reduce the risk of vulnerable or modified versions of applications reaching users. Both address the security framework and practices which help in ensuring the safe distribution and operation of mobile applications, preserving their integrity and protecting users.

10.3 Implementation Guidance (ENISA 9.3):

ENISA Secure Smartphone Development Guidance (9.3): Provide feedback channels for users to report security problems with apps such as a security@ email address.

10.3.1 OWASP MASVS MAPPING

10.4 Implementation Guidance (ENISA 9.4):

ENISA Secure Smartphone Development Guidance (9.4): If an enterprise app store is used, protect the application signing key with the utmost care (e.g., use an HSM, air-gapped machine, etc.).

10.4.1 OWASP MASVS MAPPING

- MASVS-CRYPTO-1: The MASVS-CRYPTO-1 description emphasizes the importance of cryptography in securing user data, especially in mobile environments where physical access to devices is possible. The focus on general cryptography best practices aligns with the ENISA Guideline advising the protection of application signing keys with high security measures such as HSMs (Hardware Security Modules) or air-gapped machines. Both stress the significance of robust cryptographic measures to safeguard sensitive information and resources, indicative of a correlation between the two.
- MASVS-CRYPTO-2: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-CRYPTO-2” pertains to the management of cryptographic keys throughout their lifecycle, including key generation, storage, and protection. The ENISA guideline specifically advises on protecting the application signing key with extreme care using methods such as Hardware Security Modules (HSMs) or air-gapped machines, which aligns with the key management and protection aspect of MASVS-CRYPTO-2. Both directives emphasize the importance of robust cryptographic key management to maintain the security integrity of the application and its data.
- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the ENISA guideline about protecting the app signing key exists in the context of ensuring the integrity and trustworthiness of the platform. MASVS-RESILIENCE-1 emphasizes the importance of running apps on a secure and uncompromised platform, which includes having reliable security features like secure storage and sandboxing. The ENISA guideline complements this by ensuring that the application signing key, which is used to guarantee the authenticity of the app and its updates, is protected against tampering or unauthorized access. Both guidelines are focused on maintaining a secure environment and protecting against compromises that could put app data at risk. Protecting the signing key with measures like using a Hardware Security Module (HSM) or keeping it on an air-gapped machine, supports the goal of MASVS-RESILIENCE-1 by ensuring that the security features provided by the platform can be trusted, as they are not undermined by compromised app integrity.
- MASVS-RESILIENCE-2: The correlation between ”MASVS-RESILIENCE-2” and the ENISA guideline about protecting the application signing key is based on the focus of both statements on preserving the integrity of mobile applications. MASVS-RESILIENCE-2 emphasizes the need to prevent modifications to the original code and resources to maintain the app’s intended functionality and secure user experience. Similarly, the ENISA guideline advises that if an enterprise app store is used, the application signing key should be highly protected using secure methods like Hardware Security Modules (HSMs) or an air-gapped machine. Safeguarding the signing key is essential to prevent unauthorized distribution of tampered or malicious versions of an app, as the key is used to sign applications, proving their authenticity and integrity. When the signing key is compromised, the trustworthiness of the application and the app store is at risk, making it possible for modified or malicious

versions to be mistaken as genuine. Both statements are concerned with ensuring that the app's code and functionalities remain unaltered from what the developers intended.

10.5 Implementation Guidance (ENISA 9.5):

ENISA Secure Smartphone Development Guidance (9.5): Out-of-appstore security updates should be shipped using an encrypted connection and their content should be verified before applying the update.

10.5.1 OWASP MASVS MAPPING

- MASVS-CODE-1: Both ”MASVS-CODE-1” and the ENISA Guideline emphasize the importance of maintaining up-to-date security measures. MASVS-CODE-1 focuses on ensuring that the app runs on an updated platform version to leverage the latest security patches and features provided by the mobile operating system. This practice inherently minimizes vulnerabilities known to be present in older versions. The ENISA guideline complements this by stating that security updates made available outside of the app store should be delivered securely (via encrypted connections) and verified prior to installation to ensure their integrity and authenticity. Both controls are aimed at protecting against well-known threats by enforcing the usage of the latest security updates, albeit through different scopes—one through OS updates and the other through direct app updates.
- MASVS-CODE-2: MASVS-CODE-2, which mentions the necessity for a mechanism to enforce updates for critical vulnerabilities in production apps, correlates with the ENISA Guideline that recommends secure shipment of out-of-appstore security updates via an encrypted connection and verification of the update content before application. Both are addressing the importance of ensuring that a mobile app can be updated in a secure manner when critical vulnerabilities are identified, to protect the user and the app’s ecosystem from potential threats.
- MASVS-CODE-3: There is a correlation between ”MASVS-CODE-3” and the ENISA guideline mentioned. ”MASVS-CODE-3” emphasizes the importance of comprehensive security assessment, including whitebox testing, which may not always be feasible for third-party components. However, it addresses scanning these components for known vulnerabilities as a necessary security measure. Meanwhile, the ENISA guideline focuses on ensuring the security of updates shipped outside of the app store by recommending the use of encrypted connections and verification of the update’s content before application. Both highlight the need to manage third-party components securely, albeit from different angles. ”MASVS-CODE-3” is concerned with scanning and identifying known vulnerabilities as part of a broader security assessment strategy. At the same time, the ENISA guideline is geared towards secure distribution and application of updates, which might include such third-party components. Both are part of broader measures to ensure the application’s security regarding its components and the update process.
- MASVS-CRYPTO-1: ”MASVS-CRYPTO-1” is about general cryptography best practices which cover securing user data, and the ENISA Guideline on using an encrypted connection and verification before applying updates falls within the realm of these best practices. Ensuring secure transmission and integrity of updates is aligned with cryptographic standards meant to protect user data, particularly in situations where physical access to a device could compromise security.
- MASVS-CRYPTO-2: The correlation exists because the management of cryptographic keys, as highlighted in ”MASVS-CRYPTO-2,” is intrinsic to the protection mechanism described in the ENISA Guideline regarding ”out-of-appstore security updates.” To ensure

security updates are shipped using an encrypted connection, proper key management is necessary for establishing and maintaining a secure channel. Additionally, verifying the content of the update before applying it involves the use of cryptographic techniques, which again, relies on well-managed keys to verify the integrity and authenticity of the update. Therefore, the practice of sound key management is a foundational aspect of fulfilling the ENISA Guideline for secure and trusted updates.

- MASVS-NETWORK-1: The correlation between "MASVS-NETWORK-1" and the ENISA Guideline regarding out-of-appstore security updates is that both are concerned with the secure transmission of data over the network. "MASVS-NETWORK-1" emphasizes the importance of data privacy and integrity in transit, requiring apps to establish secure connections. Similarly, the ENISA Guideline dictates that security updates shipped outside of app stores should use encrypted connections for transmission and that the content needs to be verified before updates are applied. Both guidelines aim to prevent interference and ensure that data cannot be intercepted or manipulated during transmission, thereby maintaining the security and trustworthiness of the app and its updates.
- MASVS-NETWORK-2: The correlation between "MASVS-NETWORK-2" and the ENISA guideline regarding out-of-appstore security updates is based on the principle of securing the communication channel. MASVS-NETWORK-2 emphasizes the need to trust only specific Certificate Authorities (CAs), which is an indication of implementing certificate or public key pinning. This pinning ensures that the app is connecting to the legitimate server it is intended to communicate with, thus securing the communication by preventing Man-in-The-Middle (MitM) attacks. The ENISA guideline suggests using an encrypted connection for shipping out-of-appstore security updates and verifying the content before applying the update. An encrypted connection typically relies on trusted CAs to establish a secure channel. By narrowing down the list of trusted CAs and possibly using pinning as suggested in MASVS-NETWORK-2, an app can ensure that the encrypted connection used for such updates is more secure and that it is indeed connecting to a trusted source. This improves the integrity and trustworthiness of the security updates being received and applied, aligning with the ENISA guideline's requirement for verification before updates are applied. Both controls are aimed at protecting the integrity and security of data in transit to prevent unauthorized access or tampering. Thus, they are correlated in their fundamental goal of ensuring secure communication for app updates.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the ENISA Guideline about out-of-appstore security updates exists in the mutual focus on ensuring that the platform remains secure and uncompromised. "MASVS-RESILIENCE-1" addresses the risk of running apps on a tampered platform and highlights the importance of verifying the integrity of the platform's security features. This aligns with the ENISA Guideline which suggests that security updates should be shipped securely and verified before application to maintain platform integrity. Both emphasize the trust in platform security to preserve the operating environment's integrity where the apps operate.
- MASVS-RESILIENCE-2: Both "MASVS-RESILIENCE-2" and the ENISA Guideline regarding out-of-appstore security updates emphasize the importance of maintaining the integrity of an app to ensure its intended functionality and prevent unauthorized modifications. "MASVS-RESILIENCE-2" is focused on making sure that the original code and resources of an app are not modified, which aligns with the ENISA guideline's aim to verify the content of out-of-appstore security updates before they are applied. Using an encrypted connection for shipping updates, as suggested by ENISA, also contributes to the goal of preventing backdoored or modified versions of an app. Thus, the principle of

protecting the integrity and functionality of an app is a common thread between these two guidelines.

10.6 Implementation Guidance (ENISA 9.6):

ENISA Secure Smartphone Development Guidance (9.6): Resources used by apps that are updated outside of the app-store normal mechanism must be signed. Apps must verify the signature before accepting the updated resource.

10.6.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The ENISA Guideline stating that ”Resources used by apps that are updated outside of the app-store normal mechanism must be signed. Apps must verify the signature before accepting the updated resource” correlates with the MASVS-AUTH-1 description. Both guidelines are concerned with the security of the communication between the app and remote endpoints. MASVS-AUTH-1 emphasizes the need for apps to effectively manage user authentication and authorization when connecting to remote services, which includes following best practices for secure protocol use. The ENISA Guideline complements this by enforcing the integrity of resources updated outside normal app-store mechanisms through signature verification. This guideline falls within the broader scope of ensuring that the application securely handles interaction with remote endpoints by only accepting resources that have a valid signature, which is a part of the overall authorization and authentication controls that secure the usage of protocols as described in MASVS-AUTH-1.
- MASVS-CODE-2: Both ”MASVS-CODE-2” and the ENISA Guideline pertain to the security measures involved with updates to mobile applications. ”MASVS-CODE-2” addresses the need for a mechanism to force users to update the app to address critical vulnerabilities, while the ENISA Guideline establishes that any resources updated outside of the standard app-store mechanisms must be signed and verified by the app. Both are focused on ensuring the app is updated in a secure manner, and while the MASVS control is about prompt updates in case of vulnerabilities, the ENISA Guideline is about the integrity of updates received outside of the official store, they are correlated in aiming to protect end users by ensuring updates are not only mandatory in some cases but also secure.
- MASVS-CODE-3: The correlation between ”MASVS-CODE-3” and the ENISA guideline on resource updates and signature verification can be seen in their shared focus on security for third-party components and external resources. While MASVS-CODE-3 highlights the importance of assessing all app components, including the check for known vulnerabilities in libraries (a type of ”low-hanging fruit”), the ENISA guideline emphasizes the need for securing updates for resources used by apps that occur outside of the normal app-store mechanisms, specifically through signature verification. Both are concerned with mitigating risks associated with externally sourced components or updates that could compromise application security.
- MASVS-CODE-4: The correlation between ”MASVS-CODE-4” and the mentioned ENISA guideline is that both are concerned with treating external inputs or updates as untrusted. MASVS-CODE-4 focuses on the need for data verification and sanitation for all input entry points to prevent injection attacks or bypassing security checks. Similarly, the ENISA guideline addresses the risk of resources being updated from outside the official app-store mechanism by mandating that such resources must be signed, and that the app should verify the signature before accepting the update. The common theme is ensuring the integrity and trustworthiness of external data or resources coming into the app.

- MASVS-CRYPTO-1: There is a correlation between ”MASVS-CRYPTO-1” and the described ENISA Guideline. The MASVS-CRYPTO-1 mention relates to the importance of cryptography in protecting user data, especially against attackers with physical access. This underlying principle directly supports the ENISA guideline for apps to use signatures on resources updated outside of the normal app store mechanisms. The signature acts as a cryptographic check ensuring the integrity and origin of the updated resource, which aligns with the general cryptographic best practices mentioned in MASVS-CRYPTO-1.
- MASVS-CRYPTO-2: Both MASVS-CRYPTO-2 and the ENISA Guideline address the security practices related to the proper management and verification of cryptographic elements. MASVS-CRYPTO-2 deals with cryptographic key management throughout their lifecycle, emphasizing the importance of securing keys at all stages—including generation, storage, and protection—because even strong cryptography can be undermined by poor key management practices. The ENISA Guideline stresses the importance of both the integrity of resources updated outside the normal app store mechanisms and ensuring their authenticity by requiring that they be signed and the signature be verified prior to acceptance. The correlation lies in the focus on maintaining the trustworthiness of cryptographic operations and the components involved in those operations, thus ensuring the security of the app’s data and functionality.
- MASVS-NETWORK-1: Both MASVS-NETWORK-1 and the ENISA Guideline emphasize the importance of maintaining data integrity and authenticity. MASVS-NETWORK-1 focuses on ensuring that data in transit is protected using secure connections, which often involve encryption and endpoint authentication to prevent interception or tampering. The ENISA guideline complements this by requiring that resources updated outside of the normal app-store mechanisms be signed and verified for integrity before use, which is another form of safeguarding against tampering and ensuring that any data or code introduced into the app is authentic. Both controls are about preventing unauthorized modification and ensuring the legitimacy of communication or updates, thus they are correlated with securing the app’s data and behavior.
- MASVS-NETWORK-2: The correlation exists because both the MASVS-NETWORK-2 guideline and the ENISA Guideline focus on ensuring the security and integrity of communications in mobile applications. MASVS-NETWORK-2 discusses implementing certificate pinning to trust only specific Certificate Authorities (CAs), which significantly enhances the security of TLS connections by preventing man-in-the-middle attacks that might occur if a malicious CA were trusted. Similarly, the ENISA guideline emphasizes the importance of using signatures to authenticate resources that are updated outside of the normal app store mechanisms. Both guidelines aim to prevent the acceptance of untrusted or tampered content, although in slightly different contexts (network communications vs. resource updates). However, their underlying principle of validating the trustworthiness of the content being received aligns well, reflecting a common goal of securing the app’s operation against certain attack vectors.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the ENISA Guideline about verifying the signature of externally updated resources lies in the concept of securing app interactions and resource integrity. ”MASVS-PLATFORM-1” focuses on ensuring secure interactions with IPC mechanisms, which can include updates or data exchanges between components. The ENISA Guideline extends this security principle by requiring that any resources updated outside of the regular app-store mechanism be signed and verified by the app. Both are concerned with preventing unauthorized or malicious interactions and ensuring that only safe and authorized communications and updates

occur within the app's ecosystem. They are components of a comprehensive approach to app security focusing on the integrity and safety of app interactions and updates.

- MASVS-PLATFORM-2: Both "MASVS-PLATFORM-2" and the ENISA Guideline address aspects of mobile application security related to the protection of sensitive data and integrity of app resources. "MASVS-PLATFORM-2" is focused on ensuring that WebViews, which can be used for rendering web content within mobile applications, are configured securely to prevent data leakage or exposure of sensitive functionalities, which might include updates or dynamic content loading. The ENISA Guideline complements this by stating that resources updated outside of the normal app-store mechanism must be signed, and apps must verify the signature before accepting updates. This is a measure to ensure the integrity and authenticity of updates or resources loaded by the app, which can help to prevent malicious content from being accepted by the app, thus aligning with the intent of preventing sensitive functionality exposure and data leakage as suggested by "MASVS-PLATFORM-2". The principles of secure configuration, integrity checking, and validation are common in both, aiming to safeguard the app and its data.
- MASVS-RESILIENCE-1: The MASVS-RESILIENCE-1 control relates to ensuring that the app runs on a secure, untampered platform, because a compromised OS can disable security features critical for app data protection. This aligns with the ENISA guideline that requires apps to verify signatures of updated resources to ensure they have not been tampered with outside the official update mechanisms, preserving the integrity of the platform and the app's trust in its security measures. Both share a common goal of maintaining the security integrity of the app's operating environment.
- MASVS-RESILIENCE-2: The correlation between "MASVS-RESILIENCE-2" and the ENISA guideline about signed resources being verified by apps is that both are focused on ensuring the integrity of the application and its resources. MASVS-RESILIENCE-2 addresses the broader issue of preventing unauthorized modifications to an app's code and resources, which can include methods such as code obfuscation, checksums, and digital signatures. The ENISA guideline specifically calls out the need for apps to use signatures for resources updated outside of the normal app store mechanisms, ensuring that any updates to the app come from a trusted source and have not been tampered with. Both sets of guidance are intended to protect against the risks associated with running modified versions of applications, such as cheating in games or enabling premium features without payment, and they align in their goal to preserve the intended functionality and security of mobile applications.
- MASVS-RESILIENCE-4: The correlation between "MASVS-RESILIENCE-4" and the ENISA Guideline regarding the signing and verification of updated resources outside of the app-store mechanism is found in their common goal of protecting the runtime environment and ensuring the integrity of the application. "MASVS-RESILIENCE-4" focuses on making it difficult to perform dynamic analysis and preventing dynamic instrumentation, which could potentially allow an attacker to modify the application code at runtime. By hindering these activities, the control contributes to maintaining the integrity of the application while it is running. Similarly, the ENISA Guideline mandates that resources used by apps, which are updated outside of the app-store's normal mechanism, must be signed, and that apps must verify the signature before accepting the updated resources. This requirement ensures that only legitimate and authenticated updates are applied to the application, again protecting the integrity of the app during runtime and thereby reducing the risk of dynamic manipulation. Both measures target the enhancement of the app's resilience to runtime attacks and manipulation, although they approach the problem from slightly different angles: "MASVS-RESILIENCE-4" with a broad perspective on prevent-

ing various forms of dynamic analysis, and the ENISA Guideline with a specific focus on the security of updated resources.

10.7 Implementation Guidance (ENISA 9.7):

ENISA Secure Smartphone Development Guidance (9.7): Do not deploy apps with ad-hoc signing certificates used for development and testing.

10.7.1 OWASP MASVS MAPPING

- MASVS-CRYPTO-1: The correlation between ”MASVS-CRYPTO-1” and the ENISA guideline about not deploying apps with ad-hoc signing certificates used for development and testing lies in the realm of ensuring the security of user data through the use of proper cryptographic practices. The MASVS-CRYPTO-1 requirement emphasizes the importance of cryptography in securing user data, especially given the potential for physical access to the device in a mobile environment. It advocates for adherence to general cryptography best practices. The ENISA guideline warns against using ad-hoc signing certificates because such certificates are typically less secure and are intended for development and testing, not for production. Deploying apps with these certificates can lead to security vulnerabilities that could be exploited to compromise user data. Both principles are directed towards the same goal: ensuring that cryptographic measures are appropriately implemented to protect user data. The MASVS-CRYPTO-1 requirement sets a broad expectation for cryptographic security, while the ENISA guideline provides a specific example of a practice that should be avoided to uphold cryptographic standards. Together, they underscore the importance of using secure and properly managed cryptographic mechanisms within mobile apps.
- MASVS-CRYPTO-2: The correlation between ”MASVS-CRYPTO-2” and the ENISA Guideline about not deploying apps with ad-hoc signing certificates exists. MASVS-CRYPTO-2 emphasizes the importance of managing cryptographic keys throughout their lifecycle, including their generation, storage, and protection. The ENISA guideline advises against using ad-hoc signing certificates that are meant for development and testing because this would represent poor key management. Using such certificates for production apps can lead to the risk of exposing the app to potential misuse or compromise, as these keys are typically less secure and not meant for a production environment. Deploying apps without properly managing the cryptographic keys, including the signing keys, would indeed compromise the strength of the app’s cryptography, directly relating to the MASVS-CRYPTO-2’s focus on key management.
- MASVS-NETWORK-1: The MASVS-NETWORK-1 requirement and the ENISA guideline both emphasize the importance of secure communication and integrity in mobile applications. MASVS-NETWORK-1 specifies that data privacy and integrity should be maintained in data transit by setting up secure connections, which commonly involves using TLS/SSL certificates. It implies that all aspects of ensuring secure connections, including proper use of certificates, are handled correctly. The ENISA guideline addresses a specific case where ad-hoc signing certificates, which are used for development and testing, should not be used in deployed apps. This is a security measure to ensure that the certificates in production are secure and validated. Using development certificates could potentially bypass secure defaults and introduce security weaknesses, which MASVS-NETWORK-1 control seeks to prevent. Both standards aim to prevent misconfigurations and insecure practices regarding secure data transmission and app authenticity.
- MASVS-NETWORK-2: Both the Mobile Application Security Verification Standard (MASVS) NETWORK-2 control and the ENISA guideline on not deploying apps with

ad-hoc signing certificates used for development and testing are focused on ensuring the security and integrity of the certificate trust chain. MASVS-NETWORK-2 suggests implementing certificate pinning to trust only specific certificate authorities (CAs), strengthening the system against man-in-the-middle (MITM) attacks that could exploit the trust in the default root CAs. The ENISA guideline advises against using ad-hoc signing certificates for deployment because they are intended for development and testing, not for production use. Using them in a production environment could compromise the app's security, potentially allowing unauthorized code or MITM attacks. Both controls aim to protect against threats that could undermine the security of the connection between the app and its backend services.

- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the mentioned ENISA Guideline exists in that both are concerned with the integrity and security of the platform on which the app is running. MASVS-RESILIENCE-1 emphasizes that apps should run on a secure, uncompromised operating system because many security measures depend on the OS's integrity, such as secure storage and sandboxing. Meanwhile, the ENISA Guideline advises against deploying apps with ad-hoc signing certificates used for development and testing because they could lead to apps being installed on compromised or untrusted environments which circumvent the platform's security features, similarly putting app data at risk. Both relate to ensuring that the platform's security is not undermined, which is critical for maintaining the app's overall security posture.
- MASVS-RESILIENCE-2: The correlation between "MASVS-RESILIENCE-2" and the ENISA Guideline about not deploying apps with ad-hoc signing certificates used for development and testing is that both are concerned with maintaining the integrity and security of the application. The MASVS-RESILIENCE-2 control focuses on ensuring the app functions as intended and is free from unauthorized modifications, which might include running a modified version of the app or backdooring it. Similarly, the ENISA Guideline advises against using ad-hoc signing certificates for deployment, because these certificates could allow the installation of non-final, potentially insecure, or modified versions of the app. Both the MASVS control and the ENISA Guideline aim to prevent distribution and execution of altered or unofficial app versions, thereby safeguarding the app's integrity and security.
- MASVS-RESILIENCE-4: The correlation between "MASVS-RESILIENCE-4" and the ENISA guideline about not deploying apps with ad-hoc signing certificates lies in the goal of increasing the app's resilience against manipulation and analysis. MASVS-RESILIENCE-4 focuses on making dynamic analysis and runtime manipulation more difficult for attackers. Ad-hoc signing certificates, which are intended for development and testing, are less secure than those meant for production. If an app is deployed with such certificates, it can make the app more susceptible to dynamic instrumentation and code modification at runtime, which is exactly what MASVS-RESILIENCE-4 aims to prevent. Hence, adhering to the ENISA guideline supports the objective of MASVS-RESILIENCE-4 by ensuring the app is deployed with proper, more secure signing processes, thereby reducing the risk of runtime attacks.

10.8 Implementation Guidance (ENISA 9.8):

ENISA Secure Smartphone Development Guidance (9.8): Do not generate one application for multiple environments. The production app must not contain log calls, developer URLs, test methods, and settings of the development or the testing environment.

10.8.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” which deals with user authentication and authorization best practices, and the ENISA Guideline regarding not generating one application for multiple environments is based on the common goal of ensuring the security of the application in a production environment. ”MASVS-AUTH-1” is about securing the authentication and authorization mechanisms within an app, ensuring that these processes are robust against attacks and follow security best practices. This includes avoiding common mistakes such as mishandling of credentials or inadequate session management, which could be exacerbated by having development-specific code present in a production app. The ENISA Guideline advises against including logs, developer URLs, test methods, and settings of development or testing environments in the production app. This is because such information could aid an attacker in understanding the app’s internal workings, identifying potential vulnerabilities, or gaining unauthorized access to the system. By excluding these elements from production apps, the app’s security posture is improved, which aligns with the secure use of protocols and best practices referenced in ”MASVS-AUTH-1”. Both guidelines are complementary in that they seek to eliminate potential security weaknesses that could be leveraged in an attack against the application’s authentication and authorization mechanisms.
- MASVS-CODE-2: The correlation between ”MASVS-CODE-2” and the ENISA Guideline relates to the overarching goal of maintaining a secure production environment for mobile applications. While MASVS-CODE-2 focuses on having a mechanism to enforce app updates, especially when critical vulnerabilities are identified, the ENISA Guideline emphasizes the necessity of differentiating between production and development/test environments to prevent dev-related security weaknesses such as log calls, developer URLs, and test methods from being present in the production app. Both are concerned with minimizing the risk of introducing vulnerabilities to production apps and ensuring that users are not exposed to potential security risks stemming from development practices. Enforcing updates (MASVS-CODE-2) can be seen as a response mechanism to quickly mitigate vulnerabilities that may arise from not following the ENISA guideline of strict separation between environments.
- MASVS-CODE-3: The MASVS-CODE-3 description emphasizes the importance of thorough security assessments, including checking for known vulnerabilities in libraries, which is a form of ensuring the app components are secure and free from easily detectable issues. This correlates with the ENISA Guideline, which focuses on maintaining a clean production environment without unnecessary development artifacts like log calls and test methods. Both aim to eliminate common vulnerabilities and potential security weaknesses that could be exploited if present in the production environment.
- MASVS-CODE-4: The correlation between ”MASVS-CODE-4” and the ENISA Guideline is centered around the concept of ensuring that the app has a strong security posture.

”MASVS-CODE-4” emphasizes treating all incoming data as untrusted and implementing proper verification and sanitization to prevent attacks like SQL injection, XSS, and insecure deserialization. Meanwhile, the ENISA Guideline focuses on differentiating and safeguarding production environments by avoiding developer tools, test methods, and settings that may lead to insecure practices or information leakage. Both principles underscore the importance of security considerations throughout the data handling and app configuration processes. ”MASVS-CODE-4” is concerned with data security at runtime and during app interactions, while the ENISA Guideline is concerned with making sure that the production version of the app does not include elements that can compromise the security of the application, which can be part of the app’s various data entry points. Therefore, there is a correlation as both guidelines aim to protect the app from potential vulnerabilities introduced through data or environmental misconfigurations.

- MASVS-NETWORK-1: The correlation here is that both MASVS-NETWORK-1 and the ENISA Guideline emphasize the importance of maintaining a secure and appropriate environment for the app’s operation. MASVS-NETWORK-1 deals with the security of data in transit, advocating for robust encryption and authenticated connections to protect data privacy and integrity. It cautions against bypassing secure defaults or using insecure methods, which can compromise security. Similarly, the ENISA Guideline advises against using a single application across multiple environments, warning that features or settings from a development or testing environment (like log calls, developer URLs, test methods) should not be present in the production app, as they can be potential vulnerabilities or entry points for attackers. Both guidelines are focused on minimizing security risks and ensuring that the app’s operational environment—whether in transit (MASVS-NETWORK-1) or in the different stages of development to production (ENISA)—is secure and that sensitive data is protected from exposure or compromise.
- MASVS-PRIVACY-1: Both MASVS-PRIVACY-1 and the ENISA Guideline describe practices aimed at minimizing the exposure of sensitive information and unnecessary data access. MASVS-PRIVACY-1 emphasizes the necessity for apps to access only the data they need and to obtain informed user consent, reflecting a principle of data minimization and reinforced user control over their private information. Similarly, the ENISA Guideline’s instruction to avoid including development artifacts like log calls and test methods in production apps, and to have distinct environments for development and production, also serves to minimize unnecessary data exposures and potential security vulnerabilities that could affect user privacy. While approaching the subject from different angles, both ultimately contribute to protecting user data and privacy in mobile applications.
- MASVS-PRIVACY-3: The correlation between ”MASVS-PRIVACY-3” and the ENISA Guideline lies in the emphasis on transparency and user privacy. MASVS-PRIVACY-3 requires apps to provide clear information about data practices and to avoid unexpected data collection, which can align with the ENISA guideline’s aim to prevent developer tools and settings (such as log calls and test methods that could potentially leak or misuse data) from being present in the production environment. Both seek to protect user data and ensure privacy by enforcing appropriate development practices.
- MASVS-RESILIENCE-1: Both ”MASVS-RESILIENCE-1” and the ENISA guideline emphasize the importance of a secure and uncompromised platform or environment when deploying mobile applications. ”MASVS-RESILIENCE-1” highlights the risks associated with running applications on a tampered platform, which could undermine security features such as secure storage and sandboxing. Likewise, the ENISA guideline advises against using the same application across multiple environments and stresses that the production app should be free of any elements that are specific to development or testing environ-

ments, such as log calls and developer URLs. Both are concerned with ensuring that the application's security is not compromised by the underlying environment or platform.

- MASVS-RESILIENCE-2: Both "MASVS-RESILIENCE-2" and the ENISA Guideline relate to ensuring the integrity and security of the mobile application in different environments. "MASVS-RESILIENCE-2" talks about preventing modifications to the original app code and resources to maintain its intended functionality, which can be compromised if developer URLs, log calls, or test methods are present in the production version—as indicated by the ENISA Guideline. These developer artifacts can aid in reverse-engineering or modifying the app, which directly goes against the MASVS-RESILIENCE-2 control. Therefore, there is a correlation between the two guidelines as they both aim to prevent unauthorized modifications and protect the app in user-controlled environments.
- MASVS-RESILIENCE-3: "MASVS-RESILIENCE-3" aims to make static analysis of an app as difficult as possible to prevent comprehension and subsequent tampering. The ENISA Guideline advises against including developer-specific elements like log calls and test methods in production apps, which aligns with the concept of obfuscating the app's internal workings to hinder understanding and protect it from tampering. Both recommendations aim to reduce the attack surface by minimizing the visibility of the app's internal logic and structure.
- MASVS-RESILIENCE-4: The correlation between "MASVS-RESILIENCE-4" and the ENISA guideline lies in the aim to protect the application against tampering and dynamic analysis which could expose sensitive data or logic. "MASVS-RESILIENCE-4" mentions making dynamic analysis and dynamic instrumentation difficult, which argues for obfuscation and hardening techniques that would help in preventing runtime manipulation. The ENISA guideline focuses on ensuring that the production application does not contain elements like log calls, developer URLs, or test methods/settings that are significant during development or testing but could be used as attack vectors or informational leaks if present in the production environment. Both stress upon reducing the attack surface and securing the application by restricting debugging and testing features to non-production environments. This prevention aids in making the application harder to analyze or manipulate at runtime, thus aligning with the MASVS-RESILIENCE-4 control objectives.
- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" and the ENISA guideline is that both are concerned with the prevention of unintended exposure of sensitive data in mobile applications. "MASVS-STORAGE-2" specifically addresses the risk of sensitive data being stored or exposed in publicly accessible locations due to the misuse of APIs or system features. Similarly, the ENISA guideline advises against including elements in the production application that are meant for development or testing environments, such as log calls and developer URLs, which could lead to sensitive information being unintentionally exposed. Both guidelines emphasize the need to ensure that sensitive data is not leaked due to developer oversight or misconfiguration.

Chapter 11
Handle runtime code interpretation correctly

Runtime interpretation of code and careless treatment of information flow may give an opportunity for untrusted parties to provide unverified input which is interpreted as code or to leak sensitive information. This gives an opportunity for malware to circumvent walled garden controls provided by app-stores. It can lead to injection attacks leading to data leakage, surveillance and, spyware. Lack of control on the information flow can lead to data leakage in the presence of a physical attacker. Note that it is not always obvious that your code contains an interpreter. Look for any capabilities accessible via user-input data and use of third party API's which may interpret user-input - such as JavaScript interpreters.

11.1 Implementation Guidance (ENISA 10.1):

ENISA Secure Smartphone Development Guidance (10.1): Filter user data passed to interpreters.

11.1.1 OWASP MASVS MAPPING

- MASVS-CODE-3: MASVS-CODE-3 is focused on using software components without known vulnerabilities, which is a preventative measure against the introduction of vulnerabilities into the app. The ENISA guideline on ”Filter user data passed to interpreters” deals with ensuring that untrusted inputs are properly handled to prevent them from being interpreted as code, which can prevent certain types of vulnerabilities, such as injection attacks, from being exploited. Both MASVS-CODE-3 and the ENISA guideline are related to the broader theme of securing the application by mitigating the risks associated with software components and untrusted inputs, respectively.
- MASVS-CODE-4: Both the description for ”MASVS-CODE-4” and the ENISA guideline ”Filter user data passed to interpreters” refer to the principle of treating data coming from untrusted sources—such as user input, IPC, network, or file system—as potentially malicious. The MASVS-CODE-4 control emphasizes the need to verify and sanitize such incoming data to prevent security issues like SQL injection, XSS, or insecure deserialization, which are types of attacks that occur when an interpreter executes unintended commands due to unfiltered user input. The ENISA guideline directly mentions filtering user data passed to interpreters as a measure to mitigate against such vulnerabilities. Both statements underscore the importance of input validation and data sanitation to protect applications from being compromised through their data entry points.
- MASVS-NETWORK-1: Although ”MASVS-NETWORK-1” and the ENISA Guideline ”Filter user data passed to interpreters.” are focusing on different aspects of security, there is a correlation in the broader context of ensuring security and integrity of data. ”MASVS-NETWORK-1” deals with the secure transmission of data over the network, emphasizing the importance of encryption and endpoint authentication to maintain data privacy and integrity in transit. It aims to prevent interception and alteration of data. The ENISA guideline to ”Filter user data passed to interpreters” addresses the need to sanitize inputs to prevent attacks such as SQL injection, which can compromise data integrity and privacy. Both controls are related to protecting data integrity and preventing unauthorized access to or manipulation of data, but they operate at different levels of the application stack. ”MASVS-NETWORK-1” is concerned with the transport layer and network interactions, while the ENISA guideline pertains to application-level input handling. Nonetheless, they both contribute to the overall security posture of an application by mitigating different types of threats that could lead to data breaches.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” which focuses on ensuring secure interactions involving IPC (Inter-Process Communication) mechanisms and the ENISA guideline ”Filter user data passed to interpreters” can be seen as follows: Both are concerned with securing the interactions and data flow within an application. The MASVS-PLATFORM-1 control is about securing IPC mechanisms. IPC is a way for different processes to communicate with each other, which could potentially involve user data. Ensuring that these communications are secure implies that any user data passed through these channels needs to be considered and handled securely. The ENISA guide-

line ”Filter user data passed to interpreters” emphasizes the importance of sanitizing user input, especially when it is passed to interpreters that could execute this data as code, to prevent issues such as injection attacks. The commonality between these two guidelines is the emphasis on validating, sanitizing, and securely managing user data within the context of app functionalities that could process or execute that data. Both aim to protect against security vulnerabilities that could arise from maliciously crafted user input.

- MASVS-PLATFORM-2: The correlation between ”MASVS-PLATFORM-2” and the ENISA guideline ”Filter user data passed to interpreters” is that both are related to ensuring secure handling of data within applications. ”MASVS-PLATFORM-2” focuses on the secure configuration of WebViews to prevent data leakage and exposure of sensitive functionality, which may involve controlling how user data is processed and displayed. The ENISA guideline specifically mentions filtering user data passed to interpreters, which is relevant to WebViews because they often interpret and render web content. Filtering user data in this context is a security measure to prevent malicious input from causing harm, which aligns with the purpose of securely configuring WebViews per the MASVS standard.
- MASVS-PRIVACY-1: Both ”MASVS-PRIVACY-1” and the ENISA Guideline: ”Filter user data passed to interpreters” emphasize the importance of handling user data securely and responsibly. Specifically, MASVS-PRIVACY-1 advocates for data minimization, consent-based data access, and careful management of third-party SDKs with regard to user data. Similarly, the ENISA guideline suggests applying a filter on user data before it is processed by interpreters, as a measure to prevent security risks such as unauthorized data access or injection attacks. Therefore, while each guideline focuses on different aspects—MASVS-PRIVACY-1 covering broader app data practices and ENISA focusing on a specific technical security control—both promote the principle of safeguarding user data and ensuring that it is handled with care and consent, demonstrating a correlation between the two.
- MASVS-PRIVACY-3: The correlation between ”MASVS-PRIVACY-3” and the ENISA Guideline ”Filter user data passed to interpreters” lies in the broader context of data protection and user privacy. While MASVS-PRIVACY-3 emphasizes the user’s right to know how their data is used, which includes data collection, storage, and sharing practices, the ENISA guideline focuses on a specific aspect of data handling, namely the proper filtering of user data that is passed to interpreters. Both guidelines are concerned with the secure and transparent handling of user data. Ensuring that user data is filtered before being passed to interpreters is a part of protecting user data from unexpected usage or security vulnerabilities, which aligns with the intention behind MASVS-PRIVACY-3, where users should be informed of data practices that they would not reasonably expect, like background data collection.
- MASVS-STORAGE-2: The ”MASVS-STORAGE-2” control relates to the prevention of unintentional storage or exposure of sensitive data in publicly accessible locations due to the use of certain APIs or system features. The ENISA Guideline ”Filter user data passed to interpreters” can be correlated with this control, as user data that is not adequately filtered before being passed to interpreters (such as log interpreters, backup systems, etc.) could lead to unintentional leaks of sensitive data. Both the MASVS control and the ENISA Guideline are concerned with safeguarding sensitive data by ensuring appropriate handling to prevent accidental disclosure.

11.2 Implementation Guidance (ENISA 10.2):

ENISA Secure Smartphone Development Guidance (10.2): Define comprehensive escape syntax as appropriate.

11.2.1 OWASP MASVS MAPPING

- MASVS-CODE-4: The ”MASVS-CODE-4” description talks about handling different data entry points in an application and treats the incoming data as untrusted. It emphasizes the importance of proper verification and sanitation of this data before use, to prevent security vulnerabilities like SQL injection, XSS (Cross-site scripting), or insecure deserialization. The ENISA guideline ”Define comprehensive escape syntax as appropriate” aligns with this by implying that a defined escape syntax (which is a part of the data sanitation process) is necessary to avoid such vulnerabilities by escaping special characters that might otherwise be interpreted in a harmful way. Both aim at improving application security by treating data as untrusted and enforcing validation and sanitation mechanisms to prevent injection attacks and other exploits.
- MASVS-PLATFORM-2: ”MASVS-PLATFORM-2” addresses the secure configuration of WebViews to prevent data leakage and exposure of sensitive functionality, such as through JavaScript bridges. The ENISA guideline ”Define comprehensive escape syntax as appropriate” is likely to be related to secure coding practices that guard against injection vulnerabilities and improper data handling, which can occur in WebViews. Escape syntax is part of ensuring that input is properly sanitized, which is a key to protecting against the issues outlined in ”MASVS-PLATFORM-2”. Both guidelines aim to enhance security in the context of application data handling and UI components, particularly where there is a mix of web content and native code.
- MASVS-RESILIENCE-4: The MASVS-RESILIENCE-4 control is about making it difficult to perform dynamic analysis and to prevent dynamic instrumentation of an app at runtime. This is related to the concept of defining comprehensive escape syntax as suggested by ENISA guidelines. Escape syntax is part of input validation strategies that help to mitigate against injection attacks and can be related to efforts that obstruct unauthorized analysis and modifications at runtime. By establishing strict and comprehensive escape syntax, an application can reduce the effectiveness of dynamic analysis tools that rely on predictable output or behavior to perform their analysis and code modifications. This supports the goal of MASVS-RESILIENCE-4, which is to protect the app against dynamic manipulation and analysis, enhancing its resilience against attacks.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA guideline ”Define comprehensive escape syntax as appropriate” exists in the context that both pertain to the prevention of unintentional exposure of sensitive data. ”MASVS-STORAGE-2” speaks to the need for careful handling and storage of sensitive data to avoid leaks due to improper usage of APIs or system capabilities, while ”defining comprehensive escape syntax” can be a method to ensure that data is sanitized and encoded properly before storage to prevent such leaks. Good escaping routines ensure that data written to storage does not inadvertently expose sensitive information or create security vulnerabilities, which aligns with the intention behind the control ”MASVS-STORAGE-2”.

11.3 Implementation Guidance (ENISA 10.3):

ENISA Secure Smartphone Development Guidance (10.3): Do not reveal sensitive information such as usernames, personal data and others through error messages.

11.3.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: MASVS-AUTH-1 stresses the importance of secure user authentication and authorization practices in apps that connect to a remote endpoint. Part of this security involves avoiding the leakage of sensitive information. The ENISA guideline of not revealing sensitive information through error messages aligns with this principle as error messages are a common vector for information leakage, including usernames and personal data. Both the MASVS requirement and the ENISA guideline aim to protect sensitive user information and maintain the confidentiality and integrity of the authentication and authorization processes.
- MASVS-AUTH-2: While the MASVS-AUTH-2 requirement focuses on the proper implementation of local authentication mechanisms such as biometrics or PIN codes, it inherently relates to protecting sensitive user information. A crucial aspect of correct implementation is ensuring that these authentication processes do not inadvertently expose sensitive data, such as usernames or personal details, which aligns with the ENISA guideline's emphasis on preventing sensitive information disclosure through error messages. Thus, there is a correlation as both concern safeguarding sensitive user information, albeit in different contexts within the application security landscape.
- MASVS-CODE-4: MASVS-CODE-4 emphasizes the treatment of all incoming data as untrusted and the importance of proper verification and sanitization before use. This approach directly correlates with the ENISA guideline of not revealing sensitive information through error messages, as proper data handling and sanitization would include preventing the leakage of sensitive information in error messages. Both concepts aim to protect sensitive data from being exposed due to improper handling of untrusted input.
- MASVS-CRYPTO-1: While MASVS-CRYPTO-1 is about general cryptography best practices and the ENISA Guideline refers to not revealing sensitive information through error messages, both are concerned with the protection of sensitive user data. MASVS-CRYPTO-1 acknowledges the importance of cryptography in securing user data, especially in mobile environments where physical access by attackers to a device is likely. The ENISA Guideline advises against exposing sensitive information through error messages as part of ensuring data is not leaked or made vulnerable. Both guidelines aim to prevent unauthorized access to or disclosure of sensitive information, and implementing robust cryptography is a key component of achieving this objective as it helps to safeguard data integrity and confidentiality, which includes how data is handled during errors. Therefore, there is a correlation between the two in the broad context of protecting sensitive information.
- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA Guideline about not revealing sensitive information through error messages is that both are concerned with preventing sensitive data leakage. "MASVS-PLATFORM-2" focuses on securing WebViews to prevent data leakage and exposure of sensitive functionality, which aligns with the ENISA Guideline that specifically warns against revealing sensitive information, such as usernames and personal data, through error messages. Both

aim to protect the user's privacy and security by ensuring sensitive data is not unintentionally exposed.

- MASVS-PLATFORM-3: Both MASVS-PLATFORM-3 and the ENISA guideline focus on preventing the unintentional disclosure of sensitive information. MASVS-PLATFORM-3 addresses the issue by ensuring sensitive data is not leaked through platform mechanisms like screenshots or shoulder surfing, whereas the ENISA guideline specifically addresses not revealing sensitive information through error messages. While the contexts are slightly different, the core objective of protecting sensitive data from being exposed aligns between the two, indicating a correlation.
- MASVS-PRIVACY-3: Both the Mobile Application Security Verification Standard (MASVS) requirement "MASVS-PRIVACY-3" and the ENISA Guideline regarding sensitive information in error messages are focused on protecting the privacy and sensitive data of users. MASVS-PRIVACY-3 emphasizes transparency in how user data is collected, stored, and shared, including unexpected ways such as background data collection. The ENISA Guideline similarly seeks to prevent accidental disclosure of sensitive information, such as usernames and personal data, through error messages. Both guidelines aim to prevent unauthorized access to or misuse of personal data, contributing to the overall privacy and security of the user's information.
- MASVS-RESILIENCE-3: The correlation exists in the context of preventing information disclosure that could aid an attacker. MASVS-RESILIENCE-3 focuses on making static analysis of an app difficult to prevent understanding its internals, which would include the potential discovery of sensitive information embedded in the code or resources. By making the app harder to analyze, it indirectly helps prevent an attacker from discovering sensitive information such as usernames or personal data, as mentioned in the ENISA guideline. Both address protecting sensitive information, MASVS-RESILIENCE-3 through obfuscation to protect against reverse engineering, and the ENISA guideline by advising against revealing such information through error messages directly. Both aim to reduce the risk of information leakage that could be exploited.
- MASVS-STORAGE-1: The correlation exists in the domain of protecting sensitive information within the application. "MASVS-STORAGE-1" addresses the security of sensitive data storage on mobile devices, ensuring that any intentional storage of sensitive information by the app is properly protected, regardless of where it is stored. The ENISA guideline about not revealing sensitive information through error messages is aligned with this idea, as both guidelines aim to prevent the exposure of sensitive data to unauthorized parties. While one talks about the storage aspect, the other addresses potential leaks through user interfaces; both are focused on protecting sensitive data from being compromised.
- MASVS-STORAGE-2: The description of "MASVS-STORAGE-2" and the ENISA guideline both address concerns related to the protection of sensitive data. While "MASVS-STORAGE-2" focuses specifically on preventing unintentional storage or exposure of sensitive data in publicly accessible locations, the ENISA guideline advises against revealing sensitive information through error messages. Both controls are concerned with preventing the leak of sensitive information, whether through storage mechanisms or through application error messages.

11.4 Implementation Guidance (ENISA 10.4):

ENISA Secure Smartphone Development Guidance (10.4): Deny interpreted code direct access to user data and encrypted storage.

11.4.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: Both ”MASVS-AUTH-1” and the ENISA guideline on denying interpreted code direct access to user data and encrypted storage deal with protecting user data and ensuring that the app interacts with remote endpoints securely. ”MASVS-AUTH-1” focuses on the necessity of enforcing authentication and authorization best practices within the app to securely use protocols when connecting to remote services. On the other hand, the ENISA guideline aims to prevent interpreted code (which could be more easily manipulated at runtime) from having direct access to sensitive user data and encrypted storage, as a measure to protect against unauthorized access or tampering. Together, they emphasize the importance of implementing layers of security to safeguard user information during authentication and while stored or processed by the app.
- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3,” which refers to implementing additional forms of authentication for sensitive actions in the app, and the ENISA guideline ”Deny interpreted code direct access to user data and encrypted storage” exists in the aspect of enhancing security measures to protect sensitive user data. The ENISA guideline aims to prevent untrusted code from accessing user data and encrypted storage directly, which aligns with the MASVS-AUTH-3 emphasis on secure implementation of additional authentication methods, including biometric, PIN, MFA, etc. Both emphasize on adding security layers to protect sensitive user actions and data within an application.
- MASVS-CODE-3: The correlation between ”MASVS-CODE-3” and the ENISA Guideline stems from the emphasis on securing app components against vulnerabilities. MASVS-CODE-3 suggests that while a full assessment of all components may not be feasible, especially for third-party ones, attention should still be paid to easily detectable issues, such as known vulnerabilities in libraries. The ENISA Guideline complements this by advising that interpreted code (which could include libraries or components in question) should not have direct access to sensitive user data or encrypted storage, aligning with the goal of mitigating potential security risks that could arise from exploitable vulnerabilities. Both guidelines highlight the importance of limiting exposure to and impact of common security issues within an application’s ecosystem.
- MASVS-CODE-4: The correlation exists between ”MASVS-CODE-4” which discusses treating all incoming data from various sources as untrusted and ensuring its proper verification and sanitization, and the ENISA Guideline that advises against giving interpreted code direct access to user data and encrypted storage—both emphasize the principle of treating data carefully to prevent security issues such as injection attacks and direct manipulation of sensitive data. The MASVS control addresses the broader need for data validation to protect against common vulnerabilities, while the ENISA guideline more specifically recommends a defensive programming strategy to limit the potential damage from code that could be manipulated by untrusted input.
- MASVS-CRYPTO-1: The Mobile Application Security Verification Standard (MASVS) guideline ”MASVS-CRYPTO-1” and the ENISA Guideline referred to both emphasize the protection of user data through the use of proper cryptographic practices. The MASVS-

CRYPTO-1 describes the need for good cryptographic practices because of the high risk of physical access to mobile devices by attackers, which could lead to direct attacks on data integrity and confidentiality. On the other hand, the ENISA Guideline's advice to deny interpreted code direct access to user data and encrypted storage is closely related because one of the key precepts of cryptography is to restrict access to sensitive data. Ensuring that only well-defined, secure pathways can interact with encrypted data is indeed a cryptographic best practice. Interpreted code often represents a larger attack surface — one where missteps can inadvertently grant malicious actors the ability to compromise otherwise well-protected user data. By restricting interpreted code from accessing user data and encrypted storage directly, the guideline indirectly enforces that any access to sensitive data happens through secure, intended cryptographic mechanisms, adhering to the best practices outlined in MASVS-CRYPTO-1.

- MASVS-CRYPTO-2: There is a correlation between "MASVS-CRYPTO-2" and the ENISA guideline "Deny interpreted code direct access to user data and encrypted storage." Reasoning: The MASVS-CRYPTO-2 emphasizes the importance of proper management of cryptographic keys throughout their lifecycle, including aspects such as generation, storage, and protection of keys. Improper key management could lead to compromised cryptography even if the algorithms themselves are strong. The ENISA guideline recommends denying interpreted code (such as scripts executed by an interpreter at runtime) direct access to user data and encrypted storage, which relates to the secure handling and access control of sensitive information. The correlation is that both are concerned with protecting sensitive data by ensuring secure cryptographic practices. MASVS-CRYPTO-2 focuses on the security of cryptographic keys themselves, while the ENISA guideline addresses the way interpreted code interacts with user data and encrypted content. By denying interpreted code direct access to encrypted storage, one reduces the risk of cryptographic keys being compromised through attack vectors that might stem from the execution of such code. Proper key management as per MASVS-CRYPTO-2 would be an essential part of an overall strategy to protect encrypted user data, aligning with the ENISA guideline's aim to safeguard sensitive information from unauthorized access.
- MASVS-NETWORK-1: The correlation between "MASVS-NETWORK-1" and the ENISA guideline "Deny interpreted code direct access to user data and encrypted storage." is based on the underlying principle of protecting the confidentiality and integrity of user data. MASVS-NETWORK-1 emphasizes the importance of securing data in transit by encrypting communications and properly authenticating the remote endpoint, effectively ensuring that data remains private and unaltered during transmission. This is aligned with the ENISA guideline, which advises against allowing interpreted code (which could be more easily tampered with or compromised) to have direct access to user data and encrypted storage, as such access could pose a risk to the data's privacy and integrity—whether stored or in transit. Both recommendations contribute to a secure data handling strategy within mobile applications.
- MASVS-PLATFORM-1: Both "MASVS-PLATFORM-1" and the ENISA guideline regarding the denial of interpreted code direct access to user data and encrypted storage involve considerations about secure data handling and minimizing security risks in the interactions between applications (or code) and sensitive data storage. "MASVS-PLATFORM-1" focuses on ensuring that all interactions involving Inter-Process Communication (IPC) mechanisms are secure. IPC mechanisms are a way for different processes to communicate with each other and potentially share data or functionality. This control is about making sure that any data exposed through IPC is handled securely and that the IPC mechanisms themselves do not introduce vulnerabilities. The ENISA guideline advises against allowing

interpreted code (like scripts that are executed at runtime) from having direct access to user data and encrypted storage because interpreted code can often be modified or injected with malicious intent, thus presenting a security risk. Both of these emphasize the importance of secure interaction patterns when it comes to dealing with user data and sensitive information storage, albeit from slightly different perspectives — "MASVS-PLATFORM-1" is more about secure communication between applications, while the ENISA guideline is specifically about preventing potentially unsafe code from accessing sensitive data directly. Both aim to protect user data from exposure due to insecure application behaviors.

- MASVS-PLATFORM-2: Both "MASVS-PLATFORM-2" and the ENISA guideline you mentioned are concerned with safeguarding sensitive information within mobile applications. "MASVS-PLATFORM-2" emphasizes secure configuration of WebViews to prevent data leakage and exposure of sensitive functionality, which is often achieved through JavaScript bridges. Enforcing strict security measures in WebView configurations is a way to implement the ENISA guideline's principle of denying interpreted code, such as JavaScript running within a WebView, direct access to user data and encrypted storage. The correlation lies in the common objective of protecting sensitive data against unauthorized access in the context of using features like WebViews that interpret code within mobile applications.
- MASVS-PLATFORM-3: Both "MASVS-PLATFORM-3" and the mentioned ENISA guideline address the protection of sensitive user data against unintended access or leaks. "MASVS-PLATFORM-3" focuses on preventing leaks of sensitive data displayed in the UI through platform mechanisms like screenshots, while the ENISA guideline aims to prevent direct access to user data and encrypted storage by interpreted code, which could be another potential vector for data leakage. Both are concerned with securing user data from exposure through different vulnerabilities in the system.
- MASVS-PRIVACY-3: Both "MASVS-PRIVACY-3" and the ENISA guideline address concerns regarding the use and protection of user data. "MASVS-PRIVACY-3" emphasizes the importance of transparency in data usage practices, namely how the data is collected, stored, and shared, and ensuring that any unexpected behaviors, like background data collection, are clearly communicated to the user. The ENISA guideline complements this by recommending restrictions on interpreted code's access to user data and encrypted storage, which is crucial for protecting the data from unauthorized or unexpected use. Together, they aim to safeguard user privacy and maintain the integrity of personal data by establishing boundaries on what an application can do with user data and enforcing transparency in these processes.
- MASVS-RESILIENCE-1: The MASVS (Mobile Application Security Verification Standard) RESILIENCE-1 control, which focuses on ensuring that the app runs on a secure, untempered platform, correlates with the ENISA (European Union Agency for Cybersecurity) guideline that states "Deny interpreted code direct access to user data and encrypted storage." The correlation exists because both are concerned with maintaining the integrity and security of the user's data. When an operating system is compromised, as RESILIENCE-1 warns, the security features like secure storage and sandboxing are at risk. The ENISA guideline aligns with this by recommending that interpreted code, which could potentially be manipulated or originate from untrusted sources, should not have direct access to sensitive data or encrypted storage, thus preserving the security of the platform and its data protection mechanisms.
- MASVS-RESILIENCE-2: The MASVS-RESILIENCE-2 guideline addresses concerns about integrity and modification of app code and resources, which aligns with the ENISA guideline's aim to limit interpreted code from direct access to user data and encrypted

storage. Both guidelines are focused on limiting unauthorized access or modification, preserving the integrity and security of the app and user data.

- MASVS-RESILIENCE-3: Both ”MASVS-RESILIENCE-3” and the ENISA Guideline ”Deny interpreted code direct access to user data and encrypted storage” share a core principle of enhancing an application’s security posture. The MASVS-RESILIENCE-3 requirement focuses on obstructing an adversary’s ability to understand and manipulate the application through static analysis by making the internal workings of the app less transparent. Meanwhile, the ENISA guideline proposes restricting interpreted code from directly accessing sensitive user data and encrypted storage to mitigate the risk of unauthorized access or manipulation. Both measures are aimed at averting tampering, and although they apply different methods, they are aligned in the overarching goal of securing the app against exploitation. The correlation is in their mutual aim to harden the application against reverse engineering and potential security breaches.
- MASVS-RESILIENCE-4: The correlation between ”MASVS-RESILIENCE-4” and the ENISA guideline ”Deny interpreted code direct access to user data and encrypted storage” lies in the emphasis on hindering unauthorized access and manipulation of an application at runtime. ”MASVS-RESILIENCE-4” focuses on making dynamic analysis and instrumentation difficult, which would potentially allow attackers to alter or inspect the code during execution. The ENISA guideline similarly aims to protect sensitive data by preventing interpreted code from having direct access to user data and encrypted storage. Both are concerned with preventing runtime attacks that could compromise data security.
- MASVS-STORAGE-1: The control ”MASVS-STORAGE-1” and the ENISA Guideline both emphasize the importance of protecting sensitive data, with MASVS-STORAGE-1 focusing on the proper protection of sensitive data stored by the app in various locations and the ENISA Guideline mandating the prevention of direct access to user data and encrypted storage by interpreted code. Both aim to ensure that sensitive information is safeguarded against unauthorized access or disclosure, regardless of storage location or the nature of the code trying to access it. These directives correlate in their commitment to data protection within mobile applications.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA guideline ”Deny interpreted code direct access to user data and encrypted storage” is that both are concerned with the protection of sensitive data. ”MASVS-STORAGE-2” addresses the unintended storage or exposure of sensitive data in publicly accessible locations, suggesting that developers should take measures to prevent such leaks. The ENISA guideline directs that interpreted code (such as scripts executed by an interpreter at runtime) should not have direct access to user data and encrypted storage, implying that additional security measures or restrictions should be in place to prevent potential breaches. Both statements aim to provide guidance to prevent unauthorized access to sensitive data, aligning with the principle of limiting exposure and access to such data to mitigate risks of data leakage and enhance security.

11.5 Implementation Guidance (ENISA 10.5):

ENISA Secure Smartphone Development Guidance (10.5): Strip unused functionalities from interpreters.

11.5.1 OWASP MASVS MAPPING

- MASVS-CODE-4: The correlation between ”MASVS-CODE-4” and the ENISA guideline to ”Strip unused functionalities from interpreters” lies in the concept of minimizing the attack surface by handling untrusted input carefully. MASVS-CODE-4 emphasizes treating incoming data from various entry points as untrusted input that needs to be verified and sanitized. This measure is to prevent attacks such as SQL injection and XSS, which exploit unsanitized input. The ENISA guideline complements this by advising the removal of unused functionalities in interpreters, which also reduces the likelihood of injection attacks through possible interpreter exploits. By stripping down interpreters, you eliminate potential vectors for an attacker to provide harmful input that would be incorrectly processed or executed. Both address the best practice of limiting the opportunity for untrusted data to be used in a harmful manner.
- MASVS-PLATFORM-1: ”MASVS-PLATFORM-1,” which pertains to secure interactions involving Inter-Process Communication (IPC) mechanisms, has a correlation with the ENISA guideline ”Strip unused functionalities from interpreters.” Both recommendations aim to minimize the attack surface and potential vulnerabilities. MASVS-PLATFORM-1 suggests ensuring IPC mechanisms are secure, which could involve removing any unnecessary functionalities that could be exposed through IPC. Similarly, the ENISA guideline advises stripping unused functionalities to prevent unintended access or misuse, which would include IPC exposures. Both address the principle of least privilege and reducing the possibility for unauthorized access or interaction with the app components.
- MASVS-PLATFORM-2: The Mobile Application Security Verification Standard (MASVS) guideline ”MASVS-PLATFORM-2” which refers to securely configuring WebViews to prevent data leakage and exposure of sensitive functionalities has a correlation with the ENISA guideline ”Strip unused functionalities from interpreters.” Both guidelines aim to reduce the attack surface and potential vulnerabilities within the application by controlling and limiting the capabilities accessible to potentially malicious users or scripts. In the context of MASVS-PLATFORM-2, securing WebViews is analogous to removing unnecessary or risky functionalities from interpreters as recommended by ENISA, especially considering that WebViews can serve as an interpreter for web content within native applications.
- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the ENISA Guideline ”Strip unused functionalities from interpreters” is related to the general principle of minimizing the attack surface and ensuring a secure runtime environment for the app. ”MASVS-RESILIENCE-1” addresses the importance of running an app on a secure and uncompromised platform, which involves trusting that the platform’s security features are intact and effective. By stripping unused functionalities from interpreters as recommended by ENISA, you reduce the possibility of an attacker exploiting those unnecessary or unused features to compromise the platform, thus helping to maintain the integrity of the platform’s security features that ”MASVS-RESILIENCE-1” depends

on. This practice aligns with the goal of a secure execution environment by minimizing potential vulnerabilities.

- MASVS-RESILIENCE-2: The Mobile Application Security Verification Standard (MASVS) Resilience requirement MASVS-RESILIENCE-2 emphasizes the need for protections against the modification of an app's code and resources to maintain the integrity of its intended functionality. This relates to the ENISA guideline "Strip unused functionalities from interpreters," as both aim to reduce the attack surface. Stripping unused functionalities from interpreters can help prevent attackers from exploiting those functionalities to modify the app or execute malicious code, which aligns with the goal of MASVS-RESILIENCE-2 to prevent app modifications. Removing such functionalities makes it harder for attackers to create a modified or backdoored version of the app, thus contributing to the app's resilience against integrity violations.
- MASVS-RESILIENCE-3: The description of "MASVS-RESILIENCE-3," which relates to impeding the understanding of how an app works to prevent tampering, is correlated with the ENISA guideline "Strip unused functionalities from interpreters." Both aim to reduce the attack surface and make it more difficult for an attacker to analyze or reverse-engineer the app by reducing unnecessary or unused code and features that could provide insights into the app's operations or be exploited. Stripping unused functionalities can be seen as a specific example or technique within the broader strategy of obfuscating and hardening an app against static analysis as described in MASVS-RESILIENCE-3.
- MASVS-RESILIENCE-4: The correlation between "MASVS-RESILIENCE-4," which talks about making it difficult to perform dynamic analysis and to prevent dynamic instrumentation (which could allow code modification at runtime), and the ENISA Guideline "Strip unused functionalities from interpreters" is that both aim to reduce the attack surface and make it harder for an attacker to exploit the application. By stripping unused functionalities, you eliminate potential entry points that could be used for dynamic analysis or instrumentation, which aligns with the intent of MASVS-RESILIENCE-4 to protect against such runtime manipulations.

11.6 Implementation Guidance (ENISA 10.6):

ENISA Secure Smartphone Development Guidance (10.6): Limit size of input data passed to interpreters.

11.6.1 OWASP MASVS MAPPING

- MASVS-CODE-4: Both ”MASVS-CODE-4” and the ENISA guideline ”Limit size of input data passed to interpreters” are addressing the need to treat data input into mobile applications or systems as potentially untrusted and to apply appropriate security measures to validate and sanitize this data to prevent security issues such as injection attacks. While ”MASVS-CODE-4” broadly highlights the importance of verification and sanitization of input data from various sources to avoid bypasses and classical injection attacks, the ENISA guideline specifically suggests limiting the size of the input data as a precautionary measure to reduce the risk of such attacks, which can be part of the verification and sanitization process described in ”MASVS-CODE-4”. However, it is important to note that limiting the size is just one aspect of the input validation process and should be integrated within a broader security strategy that handles all aspects of untrusted input handling.
- MASVS-PLATFORM-2: The correlation between ”MASVS-PLATFORM-2” and the ENISA Guideline ”Limit size of input data passed to interpreters” lies in the emphasis on security configurations to prevent data leakage and exposure of sensitive functionality. When WebViews are used by apps, they often include features such as interpreters to process HTML, CSS, and JavaScript. By limiting the size of input data passed to these interpreters, the risk of buffer overflow attacks, which can lead to unauthorized access and data leakage, is reduced. Therefore, both guidelines are concerned with securing input data processing to protect sensitive information and maintain app integrity.
- MASVS-PRIVACY-1: The correlation between ”MASVS-PRIVACY-1” and the ENISA Guideline ”Limit size of input data passed to interpreters” revolves around the concept of data minimization. MASVS-PRIVACY-1 emphasizes that apps should request only the necessary data they need for their functionality with informed user consent. Similarly, the ENISA guideline advises limiting the size of input data passed to interpreters. Both are measures to reduce the attack surface and the potential impact of data breaches or leaks, thereby enforcing the principle of using the least amount of data necessary for a given purpose.

Chapter 12
Check device and application integrity

Modified devices and/or applications undermine the security and privacy controls implemented in the mobile application. Device modification can be done through rooting/jailbreaking or by installing a custom OS image. Modified applications cannot be trusted to behave in the way the developer intended it. The same counts for modified devices. Current smartphone platforms support device and/or application integrity checking features those should be leveraged to check the integrity of the device and application.

12.1 Implementation Guidance (ENISA 11.1):

ENISA Secure Smartphone Development Guidance (11.1): Check the device/platform integrity to ensure that the device is not modified. Prefer using Platform services if available (e.g., Android SafetyNet attestation). Only implement custom or use third party root/jailbreak detection, if platform does not offer a built-in solution.

12.1.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: The correlation between MASVS-AUTH-2 and the ENISA Guideline exists in the context of ensuring the integrity and security of the authentication mechanisms provided by the application. MASVS-AUTH-2 emphasizes the need for correct implementation of local authentication mechanisms such as biometrics or PIN codes, and notes that some apps may completely rely on such local authentication without a remote endpoint. Similarly, the ENISA Guideline stresses the importance of checking device/platform integrity to ensure the device has not been compromised (e.g., through rooting or jailbreaking). Both guidelines aim to prevent unauthorized access and to maintain the security of the authentication process. The ENISA Guideline further advises using platform services (such as Android SafetyNet attestation) when available to guarantee this integrity, and suggests the use of custom or third-party detection methods only in the absence of built-in platform solutions. This alignment shows that both standards prioritize robust security checks for authentication, highlighting the importance of using secure and trustworthy platform features and services whenever possible.
- MASVS-CODE-1: The correlation is that both the Mobile Application Security Verification Standard (MASVS) control ”MASVS-CODE-1” and the ENISA guideline are focused on maintaining the security of the mobile device by ensuring that the devices run on secure, non-compromised platforms. The MASVS-CODE-1 emphasizes the importance of supporting only up-to-date platform versions that include the latest security patches, while the ENISA guideline recommends checking device/platform integrity to confirm that the device has not been modified through rooting or jailbreaking. Both suggest leveraging platform services—like Android SafetyNet attestation—to provide a layer of security, and they highlight the importance of avoiding vulnerabilities that can be exploited in older, unpatched, or compromised versions of the operating system.
- MASVS-CODE-2: The correlation between ”MASVS-CODE-2” and the ENISA Guideline is that both are concerned with maintaining the security integrity of the application and the environment it operates in. ”MASVS-CODE-2” focuses on ensuring that users have the latest version of the app, which presumably includes the most recent security patches and vulnerability fixes, by providing a mechanism to force updates. The ENISA Guideline also focuses on security integrity but from the perspective of the device or platform, suggesting checks for modifications such as root or jailbreak that could compromise security. Both controls are proactive measures to guard against potential threats and maintain the security posture of the app and device.
- MASVS-PLATFORM-1: The correlation exists in the context of platform security and integrity. ”MASVS-PLATFORM-1” focuses on secure interactions through platform-provided inter-process communication (IPC) mechanisms, which assumes that the platform is trusted and its services are reliable. The ENISA guideline emphasizes checking the platform integrity to ensure that the device has not been compromised (e.g., via rooting or

jailbreaking) and suggests using platform services, such as Android SafetyNet, which are designed to provide such integrity checks. Both highlight reliance on the platform's built-in mechanisms for security purposes and the importance of ensuring that these mechanisms are not undermined by platform modification.

- MASVS-RESILIENCE-1: The "MASVS-RESILIENCE-1" control from the Mobile Application Security Verification Standard (MASVS) and the described ENISA Guideline both emphasize the importance of verifying the integrity of the device and operating system. They both recognize the risks associated with running applications on compromised platforms, which can lead to the circumvention of security features like secure storage, biometrics, and sandboxing. Both suggest using platform services where available, such as Android's SafetyNet attestation, to ensure the device has not been tampered with. If such platform services are not available, they suggest implementing custom solutions or using third-party tools for root or jailbreak detection. The correlation lies in the shared goal of validating platform integrity to trust the security features provided by the OS.
- MASVS-RESILIENCE-2: Both MASVS-RESILIENCE-2 and the ENISA Guideline emphasize the importance of maintaining the integrity of the application and the device it runs on. MASVS-RESILIENCE-2 speaks to preventing modifications to the app's code and resources, while the ENISA Guideline suggests checking device/platform integrity to ensure it has not been modified and recommends using platform services like Android SafetyNet attestation to do so. Both suggest that protecting against unauthorized changes is key to maintaining the security of an app and its functionality.
- MASVS-RESILIENCE-3: The correlation between "MASVS-RESILIENCE-3" and the referenced ENISA Guideline is that both aim to protect the integrity and security of an application against tampering and unauthorized modifications. "MASVS-RESILIENCE-3" focuses on making static analysis difficult to prevent an understanding of the app's internals, which could lead to tampering. The ENISA Guideline suggests platform integrity checks to ensure a device has not been compromised through rooting or jailbreaking that could also enable tampering or bypassing security controls. Using platform services like Android SafetyNet or similar mechanisms aligns with the goal of MASVS-RESILIENCE-3 to protect the app through system-level integrity checks. Both approaches are concerned with preserving the security of the application by making it resistant to modifications, whether by obfuscating the app's code or by validating the integrity of the environment in which the app operates.
- MASVS-RESILIENCE-4: The description of "MASVS-RESILIENCE-4" aligns with the ENISA Guideline provided. Both focus on preventing the modification of an app or device at runtime, which could pose security risks. "MASVS-RESILIENCE-4" talks about making it difficult to perform dynamic analysis or instrumentation, which could lead to runtime modifications. The ENISA Guideline advises checking device/platform integrity to ensure it has not been modified and using platform services like Android SafetyNet to attest to this. Both guidelines aim to safeguard against unauthorized changes that could be leveraged by attackers.

12.2 Implementation Guidance (ENISA 11.2):

ENISA Secure Smartphone Development Guidance (11.2): Check the application integrity, check that the application and its resources are not modified: (A) Use platform service (e.g., Android SafetyNet attestation, iOS App Store receipt). (B) Perform in-memory code integrity checks to protect against code modification and/or runtime hooking.

12.2.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: Both the MASVS-AUTH-2 description and the ENISA guideline address the security mechanisms related to application integrity and authentication. MASVS-AUTH-2 focuses on the proper implementation of local authentication methods such as biometrics and PIN codes, which are part of accessing the application securely. The ENISA guideline complements this by suggesting to verify the application's integrity and ensuring that it hasn't been modified, including its in-memory code, which is fundamental for maintaining the security of the authentication mechanisms stated by MASVS-AUTH-2. Both are concerned with protecting the app from unauthorized access and ensuring that the app's security measures have not been compromised.
- MASVS-AUTH-3: The correlation exists because both MASVS-AUTH-3 and the ENISA guideline emphasize the importance of ensuring additional security measures for sensitive actions within an application. MASVS-AUTH-3 suggests enhancing authentication with methods such as biometrics, PINs, MFA, etc., while the ENISA guideline is concerned with verifying the integrity of the app and preventing unauthorized modifications. Implementing additional forms of authentication as per MASVS-AUTH-3 could involve verifying the integrity of the application to ensure that the additional authentication methods have not been tampered with, thus aligning with the ENISA guideline's recommendation for integrity checks. Both guidelines aim to prevent unauthorized access and modifications, ensuring that sensitive actions are securely authenticated and the app's integrity is maintained.
- MASVS-CODE-2: The correlation between "MASVS-CODE-2" and the ENISA Guideline is clear as they both emphasize the importance of maintaining the integrity of the mobile application. "MASVS-CODE-2" discusses having a mechanism to ensure users update the app to mitigate critical vulnerabilities, while the ENISA Guideline advises checking the application integrity to ensure that it and its resources have not been modified. Both guidelines aim to protect the app from unauthorized changes that could introduce vulnerabilities or compromise the app's security posture. Using platform services like Android SafetyNet attestation or iOS App Store receipt, and performing in-memory code integrity checks are methods to enforce and verify that the app remains unchanged and secure, which align with the concept of forcing updates when vulnerabilities are found. The necessity of maintaining a secure and unmodified application state is a common theme between the MASVS-CODE-2 description and the ENISA Guideline.
- MASVS-CRYPTO-1: The correlation between "MASVS-CRYPTO-1" and the ENISA Guideline exists because both are concerned with ensuring the security and integrity of user data and protecting the mobile application from unauthorized access or modification. "MASVS-CRYPTO-1" emphasizes the importance of using cryptography properly to secure data, especially from physical attackers. The ENISA Guideline's recommendation to check application integrity and prevent modifications relates to this by specifying how to

ensure that the application and its resources remain untampered with, employing techniques such as platform services for attestation and runtime integrity checks, which are in line with the best practices in cryptography and application security. Both are methods geared towards safeguarding the application environment and user data, which is the essence of "MASVS-CRYPTO-1".

- MASVS-CRYPTO-2: The correlation here is indirect but relevant. "MASVS-CRYPTO-2" is focused on the lifecycle management of cryptographic keys, which includes ensuring that the keys are generated, stored, and protected securely. Good key management is essential for maintaining the integrity of cryptographic operations. On the other hand, the ENISA Guideline on checking application integrity involves verifying that the application and its resources have not been tampered with. This can be done using platform services or performing in-memory code integrity checks. While MASVS-CRYPTO-2 is about maintaining the security of cryptographic keys, and the ENISA Guideline ensures the overall integrity of the application, both controls are ultimately about preserving the security and trustworthiness of the application. Strong key management as recommended by MASVS-CRYPTO-2 is a foundational aspect of ensuring that cryptographic verification processes, which may be used as a part of the integrity checks described by the ENISA Guideline, remain reliable and tamper-proof.
- MASVS-PLATFORM-1: The correlation exists because both "MASVS-PLATFORM-1" and the described ENISA guideline emphasize the importance of secure interactions involving IPC (Inter-Process Communication) mechanisms and platform services. "MASVS-PLATFORM-1" addresses the need for security in data and functionality exposure when using platform-provided IPC mechanisms, suggesting that the app must handle IPC securely which is part of maintaining application integrity. The ENISA guideline specifically mentions checking application integrity by using platform services (e.g., Android SafetyNet attestation, iOS App Store receipt) and performing in-memory code integrity checks, which directly relates to protecting the app from unauthorized modifications that could impact the security of IPC. Both are focused on ensuring that the application's data and code are securely managed and not subject to tampering, which is a key aspect of secure IPC.
- MASVS-RESILIENCE-1: The "MASVS-RESILIENCE-1" control is directly correlated with the ENISA Guideline to check the application integrity and ensure that the application and its resources are not modified. The MASVS control emphasizes the importance of running on a secure platform and ensuring that the operating system has not been compromised to trust its security features, such as secure storage and sandboxing. This aligns with the ENISA Guideline's recommendation to use platform services like Android SafetyNet attestation or iOS App Store receipt, as well as perform in-memory code integrity checks, to validate the integrity of the application and prevent code modification or runtime hooking. Both the MASVS control and the ENISA guideline are concerned with maintaining the integrity of the platform and the application to safeguard against risks posed by tampered platforms.
- MASVS-RESILIENCE-2: The ENISA guideline mentions checking the application integrity to ensure that the application and its resources are not modified. It aligns well with MASVS-RESILIENCE-2, which also emphasizes the importance of preventing modifications to the original code and resources to maintain the integrity of the app's intended functionality. Both the guideline and the MASVS control are focused on implementing measures to detect and prevent unauthorized changes, whether for the purpose of cheating in a game, enabling premium features, or introducing malicious code into the app. The methods mentioned, such as using platform services like Android SafetyNet attestation

or iOS App Store receipt, and performing in-memory code integrity checks, are practical ways of achieving this objective, which indicates a correlation between them.

- MASVS-RESILIENCE-3: The MASVS-RESILIENCE-3 control and the ENISA Guideline both aim to protect the integrity of mobile applications and prevent unauthorized modifications or understanding of the app's inner workings. MASVS-RESILIENCE-3 suggests hindering the static analysis to make it difficult to comprehend the app's functionality, which can be a step in tampering with the app. The ENISA Guideline focuses on verifying the integrity of the app and its resources through platform services and in-memory code checks, which can detect modifications or runtime alterations. Both guidelines are concerned with ensuring the app's original code and operation are unaltered, thus showing a clear correlation in their objectives.
- MASVS-RESILIENCE-4: There is a correlation between "MASVS-RESILIENCE-4" and the ENISA Guideline regarding checking application integrity and the prevention of code modification or runtime hooking. The description of "MASVS-RESILIENCE-4" suggests measures to make dynamic analysis and instrumentations difficult, which aligns with the ENISA Guideline's recommendation for performing code integrity checks and using platform services to ensure that the application and its resources have not been modified. Both aim to protect against code modification and enhance the app's resilience to attacks that involve altering runtime behavior.

12.3 Implementation Guidance (ENISA 11.3):

ENISA Secure Smartphone Development Guidance (11.3): Disable developer features: (A) Disable debugging in the application settings. (B) Check if the device is in developer mode if supported by platform (e.g., Android). (C) Check if debugger is attached and/or if the process is being traced. On platforms with managed code check for managed and native code debuggers.

12.3.1 OWASP MASVS MAPPING

- MASVS-CODE-1: The correlation between ”MASVS-CODE-1” and the ENISA guideline about ”Disable developer features” can be established based on the underlying objective of both controls, which is enhancing the security posture of the mobile application by leveraging platform-level security mechanisms. MASVS-CODE-1 emphasizes the need to ensure that the application runs on up-to-date platform versions to benefit from the latest security patches and features. By doing so, the application reduces its exposure to well-known threats that are addressed in newer OS updates. The ENISA guideline advises disabling developer features such as debugging to prevent an attacker from exploiting these features to compromise the application. Disabling debugging and checking for developer mode are developer-focused controls aligned with ensuring that the app is running in a secure environment, much like keeping the OS up-to-date, as mentioned in MASVS-CODE-1. Both sets of recommendations are serving to protect against not only known vulnerabilities but also to prevent exploitation contexts where an attacker might take advantage of debugging or developer modes to bypass security controls or analyze the application for vulnerabilities. Although they address slightly different technical specifics, the main theme remains consistent: security enhancements through platform security and reducing the attack surface.
- MASVS-PLATFORM-1: The MASVS-PLATFORM-1 guideline on ensuring secure interactions via IPC mechanisms can be correlated with the ENISA Guideline on disabling developer features. The necessity to disable debugging and check for developer mode or attached debuggers is related to securing IPC mechanisms. Debuggers and developer mode can be used to intercept or interact with IPC mechanisms insecurely, so ensuring these developer features are disabled aligns with the goal of MASVS-PLATFORM-1 to maintain secure IPC interactions.
- MASVS-PLATFORM-2: The correlation exists because ”MASVS-PLATFORM-2” emphasizes configuring WebViews securely to prevent data leakage and exposure of sensitive functionality, which can include preventing unauthorized access through debuggers or developer features. The ENISA Guideline specifically mentions disabling developer features such as debugging in the application settings, which directly relates to securing the application from potential leakage or exposure as targeted by ”MASVS-PLATFORM-2”. Both imply a focus on security in terms of the application’s runtime environment and protection against debugging and developer-related exposures.
- MASVS-RESILIENCE-1: The Mobile Application Security Verification Standard (MASVS) Resilience requirement ’MASVS-RESILIENCE-1’ focuses on ensuring that an app is running on a secure, uncompromised platform. This is clearly related to the ENISA Guideline on disabling developer features, as both emphasize the importance of a secure runtime environment. Debugging features and developer modes can undermine platform security

measures, hence checking for these as per the ENISA recommendation aligns with the intent of MASVS-RESILIENCE-1, which is to validate the integrity of the operating system and its security features.

- MASVS-RESILIENCE-2: The Mobile Application Security Verification Standard (MASVS) Resilience requirement 2 (MASVS-RESILIENCE-2) is focused on preventing the modification of an app's code and resources to maintain its intended functionality and integrity. This correlates with the ENISA guidelines on disabling developer features such as debugging to prevent reverse engineering or tampering with the application. Both guidelines aim to ensure app integrity by preventing unauthorized app modifications through developer tools.
- MASVS-RESILIENCE-3: The Mobile Application Security Verification Standard (MASVS) Resilience requirement MASVS-RESILIENCE-3's description mentions impeding comprehension through difficulties in static analysis of the app, which relates to the ENISA guideline of disabling developer features. Developer features such as debugging can be used for static code analysis to understand app internals and potentially modify its behavior. Disabling these features is a step toward making it harder for malicious users to analyze and tamper with the app. Both MASVS-RESILIENCE-3 and the ENISA guideline aim to increase app resilience against understanding and modifying the app's code.
- MASVS-RESILIENCE-4: The correlation between MASVS-RESILIENCE-4 and the ENISA Guideline is evident as both pertain to measures that should be taken to hinder dynamic analysis and manipulation of applications during runtime. MASVS-RESILIENCE-4 talks about making it difficult to perform dynamic analysis and prevent dynamic instrumentation where an attacker could modify code at runtime. The ENISA Guideline's recommendations to disable debugging, check for developer mode, and detect attached debuggers or code tracing are concrete practices that fulfill the intent of MASVS-RESILIENCE-4 by adding obstacles to dynamic analysis and preventing runtime manipulation of the app, thus enhancing its resilience against attacks.
- MASVS-STORAGE-2: The correlation between "MASVS-STORAGE-2" which addresses unintentional exposure of sensitive data due to misuse of APIs or system capabilities, and the ENISA Guideline about disabling developer features such as debugging, relates to the broader category of secure coding practices that prevent sensitive information leaks. Disabling debugging features is a specific recommendation to avoid sensitive information exposure during development or in production, complementing the MASVS control's goal to prevent data leaks due to developer oversight or misconfiguration. Both are concerned with securing the app against potential vulnerabilities that could be exploited to access or leak sensitive data.

12.4 Implementation Guidance (ENISA 11.4):

ENISA Secure Smartphone Development Guidance (11.4): Make reverse engineering harder: (A) Obfuscate code, (B) Encrypt data (e.g., strings) to further obfuscate application logic.

12.4.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: While MASVS-AUTH-2 focuses on the implementation of local authentication mechanisms such as biometrics or PIN codes, it implicitly necessitates the proper security of authentication data and logic on the device. The ENISA guideline, which advises to make reverse engineering harder through code obfuscation and data encryption, complements this by providing strategies to protect the integrity of the local authentication implementation. Obfuscating code and encrypting data make it more difficult for attackers to understand and manipulate the app's authentication logic, thereby supporting the secure implementation called for in MASVS-AUTH-2. Both measures contribute to the overall security of the application, especially when the app does not rely on remote endpoints and solely uses local authentication methods.
- MASVS-CODE-3: MASVS-CODE-3 focuses on utilizing only software components without known vulnerabilities. While this doesn't explicitly mention reverse engineering, it is a measure aimed at reducing the attack surface of an application by not including components that are prone to exploitation, which could otherwise lead to reverse engineering or other types of attacks. The ENISA Guideline recommends making reverse engineering harder through obfuscation and data encryption, which aligns with the aims of MASVS-RESILIENCE that deals with resilience against reverse engineering and tampering. The correlation exists because both MASVS-CODE-3 and the ENISA Guideline aim to improve the security of mobile applications either by preventing the use of vulnerable components or by making it more difficult for an attacker to understand and tamper with an app, respectively. The focus of MASVS-CODE-3 on using secure components indirectly contributes to making reverse engineering harder by reducing the likelihood that an attacker can find and exploit vulnerabilities within the app.
- MASVS-CRYPTO-1: The correlation exists because "MASVS-CRYPTO-1" pertains to the use of cryptography for securing user's data, which is crucial in mobile environments where physical access to the device is possible. It emphasizes on best practices in cryptography to protect data. This aligns with the ENISA guideline which recommends making reverse engineering harder by obscuring code and encrypting data, which are methods to protect application logic and sensitive information. Both advocate for enhancing security through obfuscation and encryption to protect against unauthorized access and reverse engineering efforts.
- MASVS-RESILIENCE-1: The MASVS-RESILIENCE-1 control emphasizes the importance of running on a secure and uncompromised operating system to ensure the trustworthiness of platform-dependent security features. Similarly, the ENISA guideline about making reverse engineering harder through code obfuscation and data encryption is also aimed at protecting the app from tampering and compromise. Both the MASVS control and the ENISA guideline address enhancing the resilience of the app against modifications that could weaken its security posture, indicating a correlation between the two.

- MASVS-RESILIENCE-2: The description of ”MASVS-RESILIENCE-2” directly correlates with the ENISA guideline of making reverse engineering harder. The MASVS control aims to prevent modifications to the app’s code and resources, which aligns with the guideline’s recommendations to obfuscate code to make it more difficult for an attacker to understand and manipulate it, as well as to encrypt data to further protect the application logic. Both measures are intended to increase the resilience of the app against reverse engineering and tampering.
- MASVS-RESILIENCE-3: The correlation exists between ”MASVS-RESILIENCE-3” and the ENISA guideline on making reverse engineering harder. Both are focused on implementing measures to protect the internals of an app from being easily understood and tampered with. MASVS-RESILIENCE-3 emphasizes impeding the comprehension of an app through complexities against static analysis, while the ENISA guideline suggests specific methods like code obfuscation and data encryption to obscure application logic, which are common techniques to achieve the goal described in MASVS-RESILIENCE-3.
- MASVS-RESILIENCE-4: The description of ”MASVS-RESILIENCE-4” directly correlates with the ENISA Guideline to ”Make reverse engineering harder”. Both statements emphasize the importance of making the application resistant against both static and dynamic analysis. MASVS-RESILIENCE-4 is concerned with combating dynamic analysis methods that can be used to understand or alter an app’s behavior at runtime, whilst the ENISA Guideline advocates for obfuscation and encryption as methods to obfuscate code and data, which are common tactics to increase the difficulty of reverse engineering. Both are essentially promoting measures to prevent easy understanding and modification of the app’s operation by unauthorized parties.
- MASVS-STORAGE-2: The correlation exists because ”MASVS-STORAGE-2” emphasizes the importance of preventing unintentional leaks of sensitive data, which could occur through improper use of APIs, backups, or logs. The ENISA guideline on making reverse engineering harder complements this by suggesting two methods: obfuscating code and encrypting data. Both obfuscation and encryption are measures that can help in preventing easy access to sensitive information by making it harder for attackers to understand the application logic or extract sensitive data, thus providing a layer of protection against unintentional data exposure as mentioned in MASVS-STORAGE-2.

Chapter 13
Protect the application from client side injections

Mobile apps present increased opportunities for client side injections, since they constantly interact with sensors, other installed apps and third party services. Existing mobile application flaws can be exploited in a similar way to vulnerabilities in traditional software applications. Attackers may force the application to use specially crafted data that will modify the application logic flow and lead to access control bypass or information disclosure attacks.

13.1 Implementation Guidance (ENISA 12.1):

ENISA Secure Smartphone Development Guidance (12.1): In the case that the application includes embedded web browsing capabilities (e.g., WebViews), restrict access to third party domains that do not comply with the required security standards, disable any unused platform supported functionalities, such as the plugins, local file accessibility, local content provider (content URL) accessibility and the dynamic code (e.g., JavaScript) execution support. Furthermore, avoid using full screen web interfaces since these can be abused from attackers to create fake application screens.

13.1.1 OWASP MASVS MAPPING

- MASVS-CODE-3: Both ”MASVS-CODE-3” and the ENISA Guideline emphasize the importance of security measures in the context of third-party components and embedded web browsing capabilities within an app. MASVS-CODE-3 mentions the need to scan libraries for known vulnerabilities, especially when full whitebox assessment isn't feasible. Similarly, the ENISA guideline advises restricting access to third-party domains that don't meet security standards and disabling unused functionalities, indicating a focus on minimizing the attack surface by controlling third-party content and ensuring that third-party domains meet certain security criteria. Both highlight proactive security precautions with third-party components to reduce the likelihood of exploitation.
- MASVS-CODE-4: The MASVS-CODE-4 guideline is about treating all incoming data as untrusted and ensuring it is properly verified and sanitized before use to prevent injection attacks and bypass of security checks. The ENISA Guideline complements this principle by specifically addressing embedded web browsing functionalities within apps, emphasizing restrictions on access to third-party domains, disabling unused functionalities, and avoiding full-screen web interfaces to prevent fake application screens. Both guidelines aim to mitigate risks associated with the handling of untrusted input in different contexts.
- MASVS-NETWORK-1: Both ”MASVS-NETWORK-1” and the ENISA guideline emphasize the importance of maintaining the integrity and privacy of data that the app handles, especially in transit. MASVS-NETWORK-1 focuses on setting up secure connections and ensuring data is encrypted and the remote endpoint is authenticated to prevent breaches in privacy and data integrity. Similarly, the ENISA guideline stresses the importance of restricting access to third-party domains that do not meet security standards, disabling unused functionalities that could be exploited, and being cautious with embedded web browsing capabilities, such as WebViews. Both guidelines highlight the risks involved with network communications and the need to adhere to best practices to safeguard data in transit. They are correlated in their goal to ensure that the mobile applications utilize secure communication channels and protect data from unauthorized access or manipulation.
- MASVS-PLATFORM-1: The correlation exists because both MASVS-PLATFORM-1 and the ENISA Guideline address the secure use of inter-process communication (IPC) mechanisms and the restriction of functionalities to prevent security risks. MASVS-PLATFORM-1 focuses on ensuring that all interactions involving IPC mechanisms are conducted securely, which encompasses restricting access to certain functionalities and data exposure. Similarly, the ENISA Guideline recommends restricting access to third-party domains and disabling unused platform functionalities within embedded web browsing environments

to maintain security standards. Both guidelines are concerned with minimizing the attack surface by securing how apps interact with user data, other apps, and system functionalities.

- MASVS-PLATFORM-2: Both the MASVS-PLATFORM-2 description and the ENISA Guideline are focused on the secure configuration and restricted use of WebViews within mobile applications. They both emphasize preventing sensitive data leakage and the exposure of sensitive functionality, as well as restricting certain features like plugins, local file accessibility, and dynamic code execution to enhance security. They align in advocating for the control and limitation of features that could potentially be leveraged by attackers if not properly managed.
- MASVS-PRIVACY-1: The "MASVS-PRIVACY-1" description emphasizes the importance of apps only requesting access to the data they need, ensuring informed user consent, and carefully managing third-party SDKs to respect user consent and data privacy. The ENISA Guideline deals with the restrictions on web browsing components within apps, such as WebViews, which align with the principles of restricting access to only necessary functionalities and ensuring security compliance. Both guidelines aim to minimize the unnecessary exposure of user data and enhance security by implementing strict access controls and ensuring adherence to privacy by design principles.
- MASVS-PRIVACY-3: The correlation between "MASVS-PRIVACY-3" and the ENISA Guideline exists in the emphasis on secure practices concerning user data and transparency about data handling. MASVS-PRIVACY-3 focuses on the user's right to be informed about data use, which aligns with the ENISA Guideline's directive on restricting access to third-party domains that do not meet security standards within WebViews. Both aim to protect unexpected use of data and require adherence to security measures, thereby reducing the risk of misuse of user data and ensuring user privacy. The aspect of disabling unused functionalities and avoiding full screen about web interfaces in ENISA Guideline also relates to preventing behavior a user wouldn't reasonably expect, reinforcing the principles stated in MASVS-PRIVACY-3.
- MASVS-STORAGE-1: The correlation between "MASVS-STORAGE-1" and the ENISA Guideline can be summarized as follows: Both "MASVS-STORAGE-1" and the ENISA Guideline are concerned with the security of sensitive data within the context of a mobile application. "MASVS-STORAGE-1" emphasizes the proper protection of sensitive data stored by the app, regardless of the storage location. It implies that the data comes from various sources and that there are public and private places where the data can be stored. The aim is to ensure that any sensitive data intentionally stored by the app is protected. The ENISA Guideline, on the other hand, specifically addresses the security aspects when an application includes embedded web browsing capabilities such as WebViews. It recommends restricting access to third-party domains that do not meet security standards and disabling unnecessary functionalities that could expose the app to exploitation, which includes the treatment of local files and content. The correlation lies in the overarching goal of protecting sensitive data within an application, albeit the guidelines approach this protection from different angles—one focusing on the secure storage of the data itself, and the other on the secure handling of content and functionalities within embedded web components that could potentially lead to exposure or misuse of sensitive data. Both contribute to enhancing the app's data security posture.

13.2 Implementation Guidance (ENISA 12.2):

ENISA Secure Smartphone Development Guidance (12.2): Avoid using API calls that provide bridging of dynamic code (e.g., JavaScript) with native code (e.g., Objective-C) since an injection in the dynamic code will lead to native code execution.

13.2.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between MASVS-AUTH-1 and the ENISA guideline is that MASVS-AUTH-1 mentions the importance of following best practices for secure protocol usage, which would include avoiding unsafe API calls. The ENISA guideline specifically cautions against using APIs that bridge dynamic and native code due to the risk of code injection leading to native code execution. Following this guideline is indeed part of adhering to security best practices as required by MASVS-AUTH-1. By avoiding such API calls, an app can better protect against unauthorized access or modification, which aligns with the intent of MASVS-AUTH-1 for secure authentication and authorization practices.
- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3” and the ENISA guideline is that both emphasize the security of sensitive actions within an app. ”MASVS-AUTH-3” mentions the use of additional forms of authentication to protect sensitive actions, whereas the ENISA guideline warns against the use of API calls that bridge dynamic and native code, which could be exploited through code injection to perform unauthorized actions. Both are concerned with preventing unauthorized access or modifications within an application, albeit from different angles—MASVS-AUTH-3 focuses on enhancing authentication, while the ENISA guideline focuses on preventing a potential vulnerability that could bypass authentication controls.
- MASVS-CODE-3: The correlation between ”MASVS-CODE-3” and the ENISA guideline described is the emphasis on security assessments to identify vulnerabilities. While MASVS-CODE-3 suggests a thorough whitebox examination of all components might not always be possible, particularly with third-party components, and thus recommends focusing on detectable known vulnerabilities, the ENISA guideline provides specific advice on avoiding certain API calls that bridge dynamic and native code. Both are concerned with the proactive identification and mitigation of potential security issues. MASVS-CODE-3’s approach to scanning libraries for known vulnerabilities can indirectly help prevent the kind of native code execution risks that the ENISA guideline warns against, but it may not directly address the avoidance of dynamic code bridging API calls unless such usage is part of the known vulnerabilities being scanned for.
- MASVS-CODE-4: Both ”MASVS-CODE-4” and the ENISA Guideline address concerns about untrusted inputs that can potentially compromise the security of mobile applications. ”MASVS-CODE-4” focuses on the verification and sanitation of all incoming data to prevent injection attacks and other security vulnerabilities stemming from untrusted sources. Similarly, the ENISA Guideline warns against using API calls that allow for dynamic code to interface with native code, as any injection flaws in the dynamic code could lead to the execution of malicious native code. Both highlight the importance of treating data inputs as untrusted and the necessity of implementing safeguards to sanitize and validate inputs to maintain application security.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the ENISA guideline is that both are addressing the security concerns related to inter-process

communication (IPC) and the use of APIs that could potentially expose the application to security risks. ”MASVS-PLATFORM-1” focuses on ensuring secure interactions that involve IPC mechanisms, while the ENISA guideline advises against using API calls that bridge dynamic code with native code to prevent code injection risks. Both statements aim to prevent unauthorized access or execution of code that could compromise the security of the app and the underlying platform.

- MASVS-PLATFORM-2: The ”MASVS-PLATFORM-2” description discusses ensuring secure configurations of WebViews to prevent sensitive data leakage and exposure of sensitive functionality, which might include JavaScript bridges to native code. The ENISA guideline specifically advises against using API calls that provide bridging between dynamic code, such as JavaScript, and native code, because an injection in the dynamic code could lead to native code execution. Both statements are concerned with the potential security risks that arise when bridging between web code in WebViews and native mobile code, so there is a clear correlation.
- MASVS-RESILIENCE-2: The correlation between ”MASVS-RESILIENCE-2” and the ENISA guideline regarding the bridging of dynamic and native code lies in the shared objective of preserving the integrity and security of the mobile application. ”MASVS-RESILIENCE-2” focuses on protecting the app against modifications to ensure its intended functionality remains intact. Similarly, the ENISA guideline advises against using API calls that bridge dynamic code with native code to prevent native code execution through dynamic code injection. Both are concerned with preventing unauthorized changes or exploitation of the app’s code, which could compromise its functionality and security.
- MASVS-RESILIENCE-3: Both the MASVS-RESILIENCE-3 description and the ENISA guideline aim to hinder the ability of an attacker to understand and manipulate the working of an app. While MASVS-RESILIENCE-3 focuses on making static analysis difficult, potentially including the scrutiny of bridging API calls, the ENISA guideline specifically addresses avoiding such APIs to prevent execution of tampered dynamic code. Both speak to the broader objective of protecting the integrity of the app’s code against unauthorized modifications and comprehension.
- MASVS-RESILIENCE-4: Both the MASVS-RESILIENCE-4 description and the ENISA Guideline emphasize the importance of reducing the risk associated with dynamic code execution and analysis. MASVS-RESILIENCE-4 aims to make dynamic analysis and instrumentation difficult, which aligns with the ENISA Guideline’s recommendation to avoid API calls that bridge dynamic code with native code to prevent code injection attacks that could lead to unauthorized native code execution. Both are concerned with making the app more resilient against runtime manipulation and code injection vulnerabilities.

13.3 Implementation Guidance (ENISA 12.3):

ENISA Secure Smartphone Development Guidance (12.3): In the case that the application uses JavaScript code running in the context of a file scheme URL, it is recommended to disable any unused platform supported attributes, such as accessing content from other file scheme URL and content from any origin.

13.3.1 OWASP MASVS MAPPING

- MASVS-CODE-1: While MASVS-CODE-1 and the ENISA Guideline address different specific aspects of mobile app security, they are both ultimately concerned with reducing the attack surface of the app. MASVS-CODE-1 emphasizes the need for an app to run on a platform version that includes the latest security protections, which implicitly includes mitigation for known vulnerabilities in JavaScript execution environments, as well as other components of the OS. The ENISA Guideline specifically recommends limiting the potential for exploitation by disabling unused platform-supported attributes when JavaScript code is running in the context of a file scheme URL, which aligns with the intent of MASVS-CODE-1 to ensure that the application is not vulnerable to well-known threats by leveraging up-to-date security features of the platform. Both aim to keep the app secure by advocating for best practices in reducing vulnerabilities.
- MASVS-CODE-4: The correlation between ”MASVS-CODE-4” and the ENISA Guideline is that both involve security practices related to handling untrusted input. MASVS-CODE-4 discusses the importance of treating all incoming data from various sources, such as the UI, IPC, network, and file system, as untrusted and ensuring its proper verification and sanitization to prevent injection attacks and other vulnerabilities. The ENISA Guideline complements this by specifically addressing the risks associated with JavaScript code executed within the context of a file scheme URL. It recommends disabling platform-supported attributes that are not being used, such as content access from other file scheme URLs and any origin, to restrict the attack surface that could arise from processing untrusted input. Both are aligned in the principle that applications should minimize their attack surface and handle input data in a secure manner to maintain the integrity and security of the application.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the mentioned ENISA guideline exists in the context of secure inter-process communication (IPC) mechanisms. ”MASVS-PLATFORM-1” talks about ensuring secure interactions with the IPC mechanisms, which implicitly includes mitigating risks related to the exposure of app data or functionality. The ENISA guideline focuses on the context of JavaScript running in file scheme URLs, advising to disable unused platform-supported attributes that could expose content from file scheme URLs or any origin, which could be considered a specific instance or component of secure IPC practices. Both texts aim to improve security by limiting unnecessary and potentially harmful interaction capabilities of apps with their environment or other apps.
- MASVS-PLATFORM-2: The correlation between ”MASVS-PLATFORM-2” and the ENISA Guideline is that both are concerned with the secure configuration and use of WebViews within mobile applications to prevent sensitive data exposure and unauthorized access. ”MASVS-PLATFORM-2” discusses the need for securely configuring WebViews to prevent data leakage and exposure of sensitive functionalities, which can include restrict-

ing JavaScript bridge access to native code. Similarly, the ENISA Guideline recommends disabling unused platform supported attributes when JavaScript is running inside a WebView using a file scheme URL, aiming to prevent potential security risks that could arise from accessing content across different URLs or schemes. Both guidelines are focused on limiting the attack surface within WebViews and ensuring that any interaction within the WebView is securely managed.

- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the described ENISA Guideline is that both are concerned with the integrity and trust of the platform on which an application is running. MASVS-RESILIENCE-1 emphasizes the importance of running an app on a safe, uncompromised platform, which is essential for the app to rely on built-in security features like secure storage and sandboxing. The ENISA guideline is specific to JavaScript running in the context of a file scheme URL but still pertains to disabling unused platform attributes to prevent unauthorized access and ensure security, which is in line with maintaining platform integrity. Both guidelines aim to protect the application from being exploited due to platform vulnerabilities or tampering.
- MASVS-RESILIENCE-2: Both ”MASVS-RESILIENCE-2” and the mentioned ENISA Guideline share the objective of preserving the integrity of the application and preventing unauthorized modifications or behavior. MASVS-RESILIENCE-2 is concerned with ensuring that the app’s original code and resources remain intact and function as intended, thereby preventing tampering, cheating, or enabling unauthorized features. Similarly, the ENISA Guideline focuses on the security of mobile applications that use JavaScript code running in file scheme URLs, recommending the disabling of unused attributes to prevent exploitation. Both are security measures aimed at maintaining the intended functionality and security posture of mobile applications by restricting unauthorized access and modification.
- MASVS-STORAGE-2: Both the ”MASVS-STORAGE-2” control and the ENISA guideline focus on the aspect of preventing unintentional data leaks due to the inappropriate or insecure use of APIs or platform capabilities. While MASVS-STORAGE-2 talks about sensitive data that could be unintentionally stored or exposed due to the misuse of APIs and system capabilities such as backups or logs, the ENISA guideline specifically refers to disabling unused platform-supported attributes to prevent exposure, which is a particular instance of the broader principle described in MASVS-STORAGE-2. Thus, they are correlated as both aim to minimize the risk of sensitive data exposure by enforcing secure programming practices and proper configuration of system characteristics.

13.4 Implementation Guidance (ENISA 12.4):

ENISA Secure Smartphone Development Guidance (12.4): Prevent interaction events when the application is obscured by another interface in the presentation layer in order to mitigate tapjacking attacks. By disabling the application interaction events, the possibility of a user interacting with a hidden view is eliminated.

13.4.1 OWASP MASVS MAPPING

- MASVS-PLATFORM-1: Both ”MASVS-PLATFORM-1” and the ENISA guideline address the security of inter-process communication (IPC) and user interaction with the application. ”MASVS-PLATFORM-1” focuses on ensuring that all interactions involving IPC mechanisms are secure, which includes the scenario where data or functionality is exposed to other apps or the user. The ENISA guideline specifically addresses the risk of tapjacking attacks, which can occur when an interface is obscured by another layer, allowing for malicious interaction. By preventing interaction events when the app is obscured, the guideline is essentially calling for secure IPC mechanisms as part of the app’s design, which correlates with the intent of ”MASVS-PLATFORM-1” to secure all forms of app interactions, including with the underlying platform.
- MASVS-PLATFORM-2: The correlation between ”MASVS-PLATFORM-2” and the ENISA Guideline regarding tapjacking attacks is that both focus on securing the user interface against potential threats that can result from improper handling of interface elements. ”MASVS-PLATFORM-2” emphasizes secure configuration of WebViews to prevent data leakage and exposure of sensitive functionality, which aligns with the ENISA Guideline’s principle of preventing interaction events when the application UI is obscured to mitigate tapjacking attacks. Both are concerned with ensuring that user interactions with the app’s interface do not lead to security vulnerabilities.
- MASVS-RESILIENCE-3: The correlation between ”MASVS-RESILIENCE-3” and the ENISA guideline regarding tapjacking attacks is that both are focused on enhancing the app’s resilience against security threats. ”MASVS-RESILIENCE-3” aims to impede understanding of the app through static analysis, which is a proactive measure to prevent tampering. The ENISA guideline suggests preventing interaction events when the app is obscured, which is a specific countermeasure against tapjacking attacks, a form of tampering where a transparent overlay could trick users into interacting with a malicious view. Both controls are designed to protect against tampering, albeit in different ways: one through obscurity to protect the app’s internals and the other through disabling events to protect the user interface layer.

13.5 Implementation Guidance (ENISA 12.5):

ENISA Secure Smartphone Development Guidance (12.5): In the case that the application requests custom permissions, and older platforms are supported (e.g., earlier than Android 5.0), always verify on the first run of the app that no other application has previously requested the same permissions.

13.5.1 OWASP MASVS MAPPING

- MASVS-AUTH-3: The correlation between "MASVS-AUTH-3" and the ENISA Guideline is present in the sense that both statements involve considerations for enhancing the security of mobile applications. "MASVS-AUTH-3" emphasizes the importance of implementing an additional form of authentication securely for sensitive actions within the app, which may include measures such as biometric authentication, pins, or multi-factor authentication. On the other hand, the ENISA Guideline addresses the need for due diligence in permission handling, particularly on older platforms where the permission model may be less robust, to ensure that no other application has been granted the same permissions that could undermine the security of the app. Both statements underscore the need for careful security practices to safeguard user data and the integrity of the application. "MASVS-AUTH-3" focuses on the method of user authentication, while the ENISA Guideline deals with the integrity of app permissions, both of which are critical aspects of app security and hence show a correlation in their concerns for preventing unauthorized access or privileges.
- MASVS-CODE-1: Both "MASVS-CODE-1" and the ENISA guideline focus on the importance of the application's compatibility with up-to-date mobile operating systems, which include the latest security patches and features. MASVS-CODE-1 emphasizes not supporting outdated OS versions due to well-known vulnerabilities, while the ENISA guideline specifically mentions verifying custom permissions on the first run of the app in scenarios where older platforms are supported. Both statements correlate as they aim to minimize the security risks associated with older mobile OS versions.
- MASVS-CODE-2: Both MASVS-CODE-2 and the ENISA Guideline pertain to maintaining the security of the app in production, especially in light of emerging vulnerabilities or changes in the app's operating environment. MASVS-CODE-2 focuses on having a mechanism to force updates to address critical vulnerabilities, while the ENISA Guideline emphasizes the need to check for duplicate custom permissions that could be indicative of a security issue on older platforms. Both controls are meant to ensure the app remains secure during its lifecycle and through changes in the ecosystem it operates within.
- MASVS-PLATFORM-1: Both "MASVS-PLATFORM-1" and the described ENISA Guideline address the security considerations involving interprocess communication (IPC) mechanisms and permissions in mobile applications. "MASVS-PLATFORM-1" aims to ensure secure interactions when leveraging platform-provided IPC mechanisms, whereas the ENISA Guideline focuses on verifying the uniqueness of custom permissions especially on older platforms where permission model behavior may differ. Together, they emphasize the importance of handling IPCs and permissions securely to prevent unauthorized interactions and potential security breaches.
- MASVS-PRIVACY-1: Both "MASVS-PRIVACY-1" and the ENISA Guideline emphasize the need for deliberate and restrictive use of permissions in mobile applications. "MASVS-PRIVACY-1" recommends that apps only request access to the data they need for their

functionality, which should be based on informed user consent. This is aligned with the ENISA recommendation that if an application requests custom permissions and supports older platforms, it should verify on the first run that no other application has requested the same permissions. The verification step in older platforms could aim to prevent permission re-delegation issues, ensuring that permissions are used responsibly and in accordance with users' expectations and consent. Both stress on the importance of managing permissions carefully to safeguard user privacy and follow best practices in app development.

- MASVS-PRIVACY-2: The correlation between "MASVS-PRIVACY-2" and the ENISA Guideline is that both are concerned with protecting user privacy. "MASVS-PRIVACY-2" advocates for methods such as unlinkability techniques that aim to prevent user identification and tracking. Meanwhile, the ENISA Guideline underscores the importance of permission verification on older platforms to ensure that no other applications have requested the same permissions, which could potentially lead to unauthorized access and privacy breaches. Both controls are designed to safeguard user privacy in mobile applications by implementing preventive measures against data compromise and user tracking.
- MASVS-PRIVACY-3: There is a correlation between "MASVS-PRIVACY-3," which emphasizes user rights to understand their data usage, and the described ENISA guideline that aims at ensuring transparency and user awareness, particularly in the context of permission requests on older platforms. Both promote clear communication and protection of user data privacy.
- MASVS-RESILIENCE-1: There is a correlation between "MASVS-RESILIENCE-1" and its description regarding the importance of running an app on a secure platform and the ENISA guideline about verifying custom permissions on older platforms. Both emphasize the risks associated with the app operating on a tampered or compromised OS. While MASVS-RESILIENCE-1 focuses on the necessity for the app to validate the integrity of the OS ensuring its security features can be relied upon, the ENISA guideline highlights the need to check for unique permissions on first run, especially on older platforms that might not have as robust security measures. Both controls aim to protect the app and its data from the risks originating from an untrusted or compromised environment.
- MASVS-RESILIENCE-2: The correlation here lies in the focus on ensuring the security and integrity of the app's operation on the user's device. "MASVS-RESILIENCE-2" highlights the need for protections against the modification of original code and resources, to maintain the app's intended functionality. Similarly, the ENISA Guideline addresses the risk of permission abuse on older platforms, where it recommends verifying on the first run whether the permissions have not been pre-requested by another application. Both are concerned with maintaining the integrity of the app's operation and preventing unauthorized actions that could compromise security, such as enabling premium features without paying, cheating, or inserting malicious backdoors.
- MASVS-STORAGE-1: The correlation between "MASVS-STORAGE-1" and the cited ENISA Guideline is that both focus on the protection of sensitive data managed by mobile apps. "MASVS-STORAGE-1" advocates for proper protection of sensitive data regardless of its storage location, whether private or public. The ENISA Guideline similarly addresses a security concern related to permissions and storage, targeting the risk that another app could maliciously take advantage of custom permissions, especially on older platforms. Although the two statements address different specific issues (storage protection and permission handling), they align in the broader context of ensuring data protection and security in mobile applications.

13.6 Implementation Guidance (ENISA 12.6):

ENISA Secure Smartphone Development Guidance (12.6): Always follow the domain name registration infrastructure to declare a custom permission, in order to avoid any collisions with other apps.

13.6.1 OWASP MASVS MAPPING

13.7 Implementation Guidance (ENISA 12.7):

ENISA Secure Smartphone Development Guidance (12.7): Restrict what apps can cause an application component (e.g., Android Activity) to start or are able to interact with it (e.g., Android Service and Content Provider). This can be accomplished using strict permissions.

13.7.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the described ENISA Guideline exists in the context of secure user authentication and authorization practices within mobile applications. ”MASVS-AUTH-1” emphasizes the importance of implementing proper authentication and authorization mechanisms in apps that communicate with remote endpoints. Similarly, the ENISA Guideline focuses on restricting interaction with application components, thus enforcing security boundaries and permissions. Both highlight the need for secure practices to ensure that only authorized entities can access or perform actions within the app, which is a foundational aspect of authentication and authorization security.
- MASVS-AUTH-2: The correlation between ”MASVS-AUTH-2” and the mentioned ENISA guideline is that both are focused on securing the authentication mechanisms within apps. MASVS-AUTH-2 emphasizes the correct implementation of local authentication methods such as biometrics or PIN codes, which can be considered part of the app’s authentication components. The ENISA guideline advises on restricting the interaction with an app’s components, including authentication components, by using strict permissions. Both points aim to prevent unauthorized access and enhance security by controlling how components, particularly those involved in authentication, can be accessed or triggered within the app.
- MASVS-AUTH-3: Both ”MASVS-AUTH-3” and the ENISA guideline described are focused on enhancing authentication and authorization controls within a mobile app. ”MASVS-AUTH-3” is concerned specifically with adding additional layers of authentication for sensitive actions in the app to ensure that only the legitimate user can perform those actions. On the other hand, the ENISA guideline aims to restrict access to application components, ensuring that only authorized applications or components with the proper permissions can start or interact with them. Both standards aim to prevent unauthorized access and increase the security of the mobile application, which shows a correlation in their objectives of strengthening security measures related to authentication and restriction of access.
- MASVS-CODE-4: The ”MASVS-CODE-4” which talks about treating data from various input points as untrusted and the necessity of proper verification and sanitization before use, correlates with the ENISA guideline on restricting app components’ interactions. Both focus on mitigating the risk that comes from external entities interacting with the application, either through data entry points or app components. While MASVS-CODE-4 emphasizes handling data as untrusted input, the ENISA guideline aims to limit the ability of external applications to initiate or communicate with app components, thereby reducing the attack surface for unauthorized data input or interaction.
- MASVS-PLATFORM-1: The stated ”MASVS-PLATFORM-1” is referring to the secure usage of Inter-Process Communication (IPC) mechanisms provided by the platform. IPC

mechanisms are ways for different apps to interact with one another or with the system. The description implies that these interactions must be secure, and data or functionality exposed through IPC needs to be well controlled. The ENISA guideline emphasizes limiting the interaction capabilities of apps with application components to ensure security, which can be done through strict permissions. This recommendation correlates with the MASVS-PLATFORM-1's intent to control IPC interactions securely—both are concerned with the safe and regulated use of IPC in mobile applications.

- MASVS-PLATFORM-2: The correlation between "MASVS-PLATFORM-2" and the ENISA Guideline is the focus on preventing unauthorized access and ensuring secure configuration. MASVS-PLATFORM-2 emphasizes secure configuration of WebViews to prevent data leakage and exposure of sensitive functionality, which aligns with the ENISA Guideline's emphasis on restricting app capabilities to prevent unintended interactions with application components. Both are concerned with the secure use of features (WebViews in MASVS, application components in ENISA) and applying appropriate permissions or configurations to protect sensitive information and functionality.
- MASVS-PRIVACY-1: The description of "MASVS-PRIVACY-1" aligns closely with the ENISA guideline mentioned. Both emphasize strict control of access to data and functionality. MASVS-PRIVACY-1 speaks about apps requesting only necessary data with informed consent and managing third-party SDKs to respect user consent, which is a part of controlling which components can interact with an app, as per ENISA guidelines. The focus on data minimization, consent, and end-to-end responsibility correlates with the ENISA guideline's emphasis on restricting interaction with application components via permissions, ensuring that only authorized entities have access to sensitive functionalities and data.
- MASVS-RESILIENCE-1: The ENISA guideline on restricting what apps can cause an application component to start or interact with it is related to ensuring that the underlying platform remains secure and trusted, as indicated by MASVS-RESILIENCE-1. This control involves enforcing strict permissions which are part of the platform's security features that help prevent tampering with the OS and the data within the app, aligning with the goal of MASVS-RESILIENCE-1 to ensure the OS has not been compromised.
- MASVS-RESILIENCE-2: The correlation exists because both the MASVS-RESILIENCE-2 description and the ENISA Guideline mentioned are focused on maintaining the integrity and security of a mobile application. The MASVS-RESILIENCE-2 description addresses the importance of protecting the app from unauthorized modifications that could lead to cheating, piracy, or the introduction of malware when redistributed. The ENISA Guideline discussed restricting app interactions to prevent unwanted or hazardous behavior, which similarly contributes to guarding the app's integrity and functionality. Both sets of guidance aim to prevent unauthorized activities that could undermine an app's security posture.
- MASVS-RESILIENCE-3: Both the MASVS-RESILIENCE-3 description and the ENISA Guideline are focused on increasing the security of mobile applications by limiting the attack surface that could be exploited by an adversary. The MASVS-RESILIENCE-3 aims to prevent understanding the app's internals to impede tampering, while the ENISA Guideline's aim is to restrict app interaction to prevent unauthorized component starting or interaction, a form of static analysis prevention and tampering obstruction. Both guidelines are measures to enhance app resilience against reverse engineering and malicious interactions.
- MASVS-RESILIENCE-4: Both the MASVS-RESILIENCE-4 and the ENISA Guideline highlighted here are concerned with increasing the difficulty for attackers to analyze and manipulate an application's behavior at runtime. MASVS-RESILIENCE-4 focuses on

making dynamic analysis and instrumentation harder, which could prevent attackers from modifying code during execution. Similarly, the ENISA Guideline aims to restrict the interactions between application components, which can protect against unauthorized activities, such as starting activities or services without proper permission, ultimately contributing to preventing dynamic code manipulation and analysis. Both measures are methods of hardening the app against runtime attacks and unauthorized interactions, which aligns with the overall objective of app resilience.

- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the ENISA guideline is that both are focused on protecting sensitive data. ”MASVS-STORAGE-1” emphasizes on the secure handling and storage of sensitive data regardless of the storage location, while the ENISA guideline addresses security by restricting component interaction through permissions, which can include access to data storage. Both are concerned with ensuring that sensitive data is accessed and used in a secure and controlled manner.

13.8 Implementation Guidance (ENISA 12.8):

ENISA Secure Smartphone Development Guidance (12.8): Restrict the third party applications whose broadcast messages will be accepted by the application.

13.8.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between "MASVS-AUTH-1" and the ENISA guideline regarding restricting third-party applications from sending broadcast messages lies in the foundational principle of security concerning authentication and authorization. "MASVS-AUTH-1" discusses the importance of a secure implementation of authentication protocols and ensuring user credentials are managed properly when connecting to a remote endpoint. Part of managing secure authentication involves constraining how and which external entities (such as third-party apps) can interact with the application. The ENISA guideline focuses on limiting interactions with third-party applications to prevent unauthorized access or information leakage, which aligns with the authentication and authorization practices mentioned in "MASVS-AUTH-1". By restricting third-party applications whose broadcast messages are accepted, an app can mitigate risks of interception or manipulation that could compromise authentication mechanisms or lead to unauthorized actions within the app, hence supporting the enforcement of security as outlined in "MASVS-AUTH-1".
- MASVS-AUTH-3: Both "MASVS-AUTH-3" and the ENISA Guideline emphasize the importance of securing sensitive actions within an application. "MASVS-AUTH-3" suggests using additional forms of authentication like biometrics, PIN, or MFA to ensure that only authorized users can perform certain actions. The ENISA Guideline, on the other hand, focuses on restricting broadcast messages to prevent unauthorized third-party applications from triggering sensitive actions or gaining access within the application. Both of these point towards the principle of limiting and securing access to sensitive functionality to enhance security.
- MASVS-CODE-2: "MASVS-CODE-2", which refers to having a mechanism to force updates in production to mitigate critical vulnerabilities, correlates with the ENISA guideline regarding restriction on acceptance of broadcast messages from third-party applications. The reasoning is that updates to mitigate vulnerabilities may include improved restrictions on how the app interacts with third-party applications, including how it handles incoming broadcast messages, to prevent exploitation of vulnerabilities that could arise from less secure third-party application messages. In essence, ensuring apps are updated could also enforce stricter communication controls as suggested by ENISA.
- MASVS-CODE-4: The correlation between "MASVS-CODE-4" and the ENISA guideline lies in the principle of treating all external data sources as potentially dangerous and ensuring data validation and sanitization. "MASVS-CODE-4" focuses on the broad spectrum of data entry points that an app might have, arguing the necessity of treating all incoming data as untrusted, irrespective of its source, thus covering UI elements, IPC mechanisms, network interfaces, and file system interactions among others. The ENISA guideline's advice to "Restrict the third party applications whose broadcast messages will be accepted by the application" is a specific instance of what MASVS-CODE-4 is advocating. It delves into the realm of Inter-Process Communication (IPC), which is one of the data entry points mentioned in the MASVS description. By enforcing restrictions on which broadcasts are accepted from third-party applications, an app can prevent the acceptance and processing

of malicious or malformed data, which aligns with the sanitization and validation strategy that MASVS-CODE-4 recommends. In essence, both are geared towards ensuring that the application treats incoming data, particularly from less-trusted or untrusted third-party sources, with a high degree of caution to maintain security and integrity.

- MASVS-PLATFORM-1: The control "MASVS-PLATFORM-1" ensures that all Inter-Process Communication (IPC) mechanisms happen securely, which directly correlates to the ENISA guideline advising to restrict third-party applications whose broadcast messages will be accepted by the application. Both focus on securing IPC to prevent unintended or malicious interactions with the application's exposed functionalities or data by third-party apps. Restricting third-party applications aligns with the concept of secure interactions mentioned in "MASVS-PLATFORM-1".
- MASVS-PLATFORM-2: While there isn't a direct one-to-one correspondence between "MASVS-PLATFORM-2", which deals with the secure configuration of WebViews, and the ENISA guideline to restrict broadcast messages from third-party applications, there is a related concept of controlling the interaction between the app and external entities to prevent data leakage or exposure of sensitive functionalities. Configuring WebViews securely often involves managing how the app interacts with web content, including controlling JavaScript execution and preventing unauthorized access to native functions. This parallels the ENISA guideline's aim to restrict unwanted interactions with third-party applications, as both measures enhance security by defining clear boundaries for external communications and data exchange.
- MASVS-PRIVACY-1: Both "MASVS-PRIVACY-1" and the ENISA Guideline on restricting the third-party applications whose broadcast messages will be accepted by the application advocate for a controlled and minimal approach to data access and sharing in order to safeguard user privacy. The MASVS-PRIVACY-1 principle focuses on the importance of data minimization, informed user consent, and careful management of third-party SDKs to prevent unauthorized data access or sharing. Similarly, the ENISA Guideline aims to prevent unwanted data exposure and potential security breaches by limiting which third-party broadcasts are allowed. Both principles are aligned in their purpose of reducing the risk of data breaches and leaks by controlling third-party interaction within the app's ecosystem.
- MASVS-PRIVACY-2: The correlation between "MASVS-PRIVACY-2" and the ENISA guideline "Restrict the third party applications whose broadcast messages will be accepted by the application" exists in the context of protecting user privacy. MASVS-PRIVACY-2 focuses on using techniques such as data abstraction, anonymization, and pseudonymization to prevent user identification and tracking, and emphasizes that each data stream should serve its intended function without risking user privacy. The ENISA guideline complements this by proposing that an application restricts third-party applications from sending broadcast messages, which could potentially be used for tracking or identifying users, thereby ensuring that user privacy is maintained and that no unintended data streams are introduced that could compromise their privacy. Both are concerned with reducing the risk of user identity exposure and tracking.
- MASVS-PRIVACY-3: Both "MASVS-PRIVACY-3" and the ENISA guideline emphasize the importance of transparent data practices and the control over how data is handled within mobile applications. MASVS-PRIVACY-3 highlights the user's right to be informed about how their data is used, including unexpected data collection behaviors. Similarly, the ENISA guideline on restricting third-party applications from sending broadcast messages helps prevent unauthorized data sharing or access to sensitive information, aligning with the principle of providing clear information on data handling as required by MASVS-

PRIVACY-3. Both aim to protect user privacy by implementing controls over data handling within apps.

- MASVS-RESILIENCE-1: The correlation between MASVS-RESILIENCE-1 and the ENISA guideline regarding the restriction on third party applications whose broadcast messages will be accepted by the application exists because both are concerned with maintaining the security integrity of the mobile platform and the application. MASVS-RESILIENCE-1 emphasizes the importance of operating on a secure, unmodified platform to ensure the trustworthiness of the device's security features, such as secure storage and sandboxing. Meanwhile, the ENISA guideline's focus on restricting third party application broadcast messages aims to prevent potentially harmful communications that could exploit a compromised OS or bypass security controls. Both seek to protect the application and user data from risks associated with a tampered platform.
- MASVS-RESILIENCE-4: The correlation between "MASVS-RESILIENCE-4" and the ENISA Guideline about restricting broadcast messages from third-party applications is that both are security measures aimed at reducing the attack surface of an application. "MASVS-RESILIENCE-4" emphasizes making dynamic analysis and runtime instrumentation by an attacker as difficult as possible, which can include monitoring and interfering with broadcast messages. By restricting third-party applications' broadcast messages, an application reduces the chances of dynamic analysis and runtime manipulation since attackers often use broadcasts for malicious purposes such as injecting code or capturing sensitive information.
- MASVS-STORAGE-2: The correlation exists because the guideline from ENISA about restricting third-party applications from sending broadcast messages that will be accepted by the app is related to preventing unintentional leaks of sensitive data. The MASVS-STORAGE-2 control addresses the risk of sensitive data being unintentionally stored or exposed in publicly accessible locations due to the misuse of APIs or system capabilities. By restricting which third-party applications can interact with an app, developers can prevent these applications from inadvertently exposing or storing sensitive information in insecure ways, such as through broadcast messages that could be logged or otherwise exposed. Thus, both the MASVS-STORAGE-2 and the ENISA guideline aim to safeguard sensitive data by advising developers on best practices to avoid unintentional data exposure through system interactions.

13.9 Implementation Guidance (ENISA 12.9):

ENISA Secure Smartphone Development Guidance (12.9): In the case that the application utilizes a platform provided download manager, always verify that the received manager's notifications are related to application's initiated downloads.

13.9.1 OWASP MASVS MAPPING

- MASVS-CODE-2: The correlation between "MASVS-CODE-2" and the ENISA guideline is that both address the concern of app security in the context of updates and downloads. "MASVS-CODE-2" emphasizes the necessity for a mechanism to enforce app updates when critical vulnerabilities are discovered, ensuring that users are running a secure version. The ENISA guideline complements this by advising verification of download manager notifications to ensure that any updates or downloads initiated by the application are legitimate and not tampering or malicious exploits. Both controls aim to protect the integrity of app updates and maintain secure app operations in production.
- MASVS-CODE-4: The correlation exists because both MASVS-CODE-4 and the ENISA Guideline are focusing on the proper handling of untrusted input sources to prevent security vulnerabilities. MASVS-CODE-4 emphasizes that apps have multiple data entry points which should be treated as untrusted and thoroughly verified and sanitized. The ENISA Guideline specifically mentions the verification of downloads via a platform-provided download manager, which is an example of an untrusted data entry point as described in MASVS-CODE-4. Both guidelines aim to ensure that untrusted data is not processed without proper checks, reducing the risk of injection attacks and other security issues.
- MASVS-NETWORK-1: The correlation between "MASVS-NETWORK-1" and the ENISA guideline is that both are concerned with ensuring the security and integrity of network communications within mobile applications. "MASVS-NETWORK-1" emphasizes the importance of protecting data privacy and integrity during data transit, which involves encrypting data and authenticating endpoints to prevent interception or manipulation. Similarly, the ENISA guideline addresses the security of the download process, advising that applications should verify download manager notifications to ensure they are connected to intended app-initiated downloads, thus preventing unauthorized or malicious downloads. Both controls are designed to protect the integrity and trustworthiness of network interactions within mobile applications.
- MASVS-PLATFORM-1: The correlation exists because both the "MASVS-PLATFORM-1" and the ENISA Guideline are concerned with the secure use of platform-provided mechanisms. "MASVS-PLATFORM-1" focuses on ensuring secure interactions via IPC (Inter-Process Communication) mechanisms provided by the platform, implying that the interactions and data exchanges that occur should be intentional, well-defined, and secure. The ENISA Guideline specifically addresses the use of a platform-provided download manager, mentioning that an app should verify that notifications received are indeed related to its own initiated downloads. This is a particular instance of ensuring secure IPC, as it involves the app interacting with a platform component (the download manager) and verifying that this interaction is legitimate and related to the app's own operations. Both address the necessity for applications to handle platform interactions securely to prevent unintended or malicious access and the exposure of sensitive information.

- MASVS-RESILIENCE-2: While the Mobile Application Security Verification Standard (MASVS) resilience control ”MASVS-RESILIENCE-2” and the ENISA guideline both address different aspects of app security, they are correlated in their overarching goal to maintain the integrity of the application and protect against unauthorized alterations. The MASVS-RESILIENCE-2 focuses on ensuring the integrity of the app's functionality by preventing modifications to the original code and resources which could lead to running a modified or backdoored version of the app. This control aligns with the broader goal of preventing unauthorized changes that could compromise app behavior or security. The ENISA guideline pertains to secure handling of downloads through a platform-provided download manager. It emphasizes the need to validate that download notifications received from the manager are indeed related to downloads initiated by the application itself. This also serves to protect the integrity of the app by preventing the execution or installation of unauthorized or possibly malicious downloads. Both controls seek to prevent tampering and ensure that the app functions as intended without being compromised by external modifications or malicious content. While they address different mechanisms (code modification vs. download verification), they share a common goal related to app resilience and integrity, thus showing a correlation.
- MASVS-RESILIENCE-3: While MASVS-RESILIENCE-3 focuses on impeding the comprehension of an app through static analysis to prevent tampering, the ENISA Guideline emphasizes verifying manager's notifications to ensure they are related to the application's initiated downloads as a security measure. Both controls are concerned with different aspects of app security and resilience—the MASVS with protecting against understanding and tampering with app internals, and the ENISA with ensuring the integrity of application downloads. There is no direct correlation between these two specific controls, as one addresses understanding the app internals to prevent modification, and the other focuses on verifying the legitimacy of download notifications.
- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the cited ENISA Guideline is that both are concerned with the secure handling of sensitive data by mobile applications. ”MASVS-STORAGE-1” refers to the necessity for apps to appropriately protect any sensitive data they store, regardless of the storage location. The ENISA Guideline complements this by specifically addressing the security measures needed when an app uses a platform-provided download manager to handle data, ensuring that notifications received from the download manager pertain only to the app's intended downloads. Both guidelines aim to safeguard sensitive data from unauthorized access or leakage during storage and handling.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA Guideline stating ”In the case that the application utilizes a platform provided download manager, always verify that the received manager's notifications are related to application's initiated downloads.” is that both are concerned with preventing accidental exposure of sensitive data due to the misuse of system capabilities or APIs. MASVS-STORAGE-2 speaks to preventing unintentional leaks when developers could have taken measures to prevent such exposure, which aligns with the ENISA Guideline's recommendation to actively verify that download manager notifications pertain to the app's downloads, thereby avoiding potential leakage of sensitive information through system notifications that could be accessible to other entities.

13.10 Implementation Guidance (ENISA 12.10):

ENISA Secure Smartphone Development Guidance (12.10): Always verify dynamic code downloads and application updates at the client side. Any resource that is being retrieved from an external service (e.g., compressed files, APK files) should be validated for its integrity and its signing certificate.

13.10.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: Both MASVS-AUTH-1 and the ENISA Guideline emphasize the importance of secure communication and integrity checks when interacting with remote endpoints. MASVS-AUTH-1 focuses on ensuring adherence to best practices for secure authentication and authorization protocols when apps connect to remote services. On the other hand, the ENISA Guideline specifies the need for client-side verification of dynamic code downloads and application updates, which includes validating integrity and signing certificates. Both are concerned with preventing unauthorized access or manipulation of data transmitted between the client and remote services. They complement each other by addressing security from both the client-side (ENISA Guideline) and the service-side (MASVS-AUTH-1).
- MASVS-CODE-2: MASVS-CODE-2 describes the necessity for a mechanism to force users to update the app in case critical vulnerabilities are discovered after production. The ENISA Guideline emphasizes the importance of verifying the integrity and signing certificate of dynamic code downloads and application updates at the client side. Both statements correlate as they address the security of application updates. MASVS-CODE-2 focuses on the process of ensuring users apply updates, especially critical ones, while the ENISA Guideline provides specifics on how those updates, among other dynamic resources retrieved from external services, should be securely handled and verified on the client side. Together they support the principle that app updates should be securely managed and enforced to maintain application security.
- MASVS-CODE-3: Both ”MASVS-CODE-3” and the ENISA Guideline emphasize the importance of security verification for external components and updates. ”MASVS-CODE-3” suggests a whitebox assessment for all app components and acknowledges that checking third-party components can be limited to scanning for known vulnerabilities, which is similar to ENISA’s recommendation to verify the integrity and signing certificates of dynamically downloaded code and updates. Both guidelines aim to ensure that externally sourced code does not compromise the security of the application.
- MASVS-CODE-4: Both ”MASVS-CODE-4” and the ENISA Guideline emphasize the need for validation of incoming data to ensure its integrity. ”MASVS-CODE-4” calls for treating data from various entry points as untrusted and properly verifying and sanitizing it before use, while the ENISA Guideline specifically addresses the need to verify dynamic code downloads and application updates client-side for integrity and certificate authenticity. Both guidelines aim to prevent the use of malicious or modified data in the application, reducing the risk of bypassing security checks and injection attacks.
- MASVS-CRYPTO-1: The correlation between MASVS-CRYPTO-1 and the ENISA guideline is that both are concerned with ensuring the integrity and security of user data, particularly in the mobile environment where the risk of physical device compromise is higher. MASVS-CRYPTO-1’s mention of general cryptography best practices encompasses the

use of cryptographic measures to secure data, which includes verifying the integrity of dynamically downloaded code and application updates as described in the ENISA guideline. The act of verifying the integrity and signing certificates of external resources, as stipulated by ENISA, is an application of cryptographic best practices, ensuring that the resources have not been tampered with and are from a trusted source. This is aligned with the goal of MASVS-CRYPTO-1 to use cryptography to protect user data.

- MASVS-NETWORK-1: The ”MASVS-NETWORK-1” control and the ENISA guideline both emphasize the importance of maintaining data privacy and integrity when data is in transit. ”MASVS-NETWORK-1” refers to securing connections, which can imply verifying the integrity and authenticity of the data being communicated, as does the ENISA guideline which focuses on verifying the integrity and certificate of dynamically downloaded code and application updates. Both controls are designed to prevent tampering or unauthorized access during the data transfer process.
- MASVS-NETWORK-2: The correlation between ”MASVS-NETWORK-2” and the described ENISA Guideline is that both controls are aimed at increasing the security of network communications within a mobile application. ”MASVS-NETWORK-2” calls for the usage of certificate or public key pinning to ensure that the app is communicating with the intended server and that the server’s certificate is trusted explicitly, rather than relying on any certificate issued by a root CA on the device. The ENISA Guideline emphasizes verifying dynamic code and application updates, which involves ensuring the integrity and authenticity of downloaded resources by checking their signing certificate. Both controls are concerned with the validation of secure connections and the verification of resources to protect against man-in-the-middle attacks or malicious content delivery.
- MASVS-PLATFORM-1: The correlation exists because both ”MASVS-PLATFORM-1” and the ENISA Guideline emphasize the importance of secure interactions with external sources or mechanisms. ”MASVS-PLATFORM-1” focuses on ensuring secure interactions involving IPC (Inter-Process Communication) mechanisms, while the ENISA Guideline underscores the need to verify and validate resources retrieved from external services, including dynamic code downloads and application updates. Both involve assessing and securing the means through which data or code is exchanged or accessed to prevent unauthorized or harmful transactions, which may compromise app integrity or user data.
- MASVS-PLATFORM-2: Both ”MASVS-PLATFORM-2” and the ENISA guideline focus on the security aspects related to the use of external content within mobile applications. ”MASVS-PLATFORM-2” highlights the need for secure configuration of WebViews to prevent sensitive data leakage and exposure of sensitive functionality, which may include the safe handling and execution of dynamic and potentially untrusted content. The ENISA guideline stresses the importance of verifying dynamic code downloads and application updates for integrity and the authenticity of their signing certificates, which is in line with controlling the risks associated with importing and executing external resources. Both statements aim to guard against potential vulnerabilities that could be exploited via code or content fetched from external services, aligning in their emphasis on security checks for externally sourced data or code to ensure app security and data protection.
- MASVS-RESILIENCE-1: There is a correlation between ”MASVS-RESILIENCE-1” and the mentioned ENISA guideline. Both focus on ensuring the integrity and security of the application environment. MASVS-RESILIENCE-1 emphasizes the importance of the app running on a secure, uncompromised platform, which aligns with the ENISA guideline’s emphasis on verifying the integrity of dynamically downloaded code and application updates. In both cases, the underlying concern is to protect the app and its data from tampering or compromise that can occur if the platform or updates are not trustworthy.

- MASVS-RESILIENCE-2: Both ”MASVS-RESILIENCE-2” and the ENISA Guideline emphasize the importance of maintaining the integrity of an application to ensure that its intended functionality is preserved and that modifications are prevented. They both suggest that means of validation, such as integrity checks and certificate verification, are essential to prevent unauthorized code modifications that could lead to cheating, enabling of unauthorized premium features, or the introduction of malicious backdoors.
- MASVS-RESILIENCE-3: The correlation between ”MASVS-RESILIENCE-3” and the ENISA guideline exists in both advocating for measures to protect the app against tampering and ensuring the integrity of the app’s execution environment. MASVS-RESILIENCE-3 focuses on preventing easy understanding of an app’s internal workings through static analysis obstacles, which can make tampering more difficult. The ENISA guideline complements this by emphasizing the verification of dynamic code downloads and application updates for integrity and authenticity. Together, these controls constitute a defense-in-depth approach to maintaining application resilience against reverse-engineering and tampering attacks.
- MASVS-RESILIENCE-4: The description of ”MASVS-RESILIENCE-4” is correlated with the ENISA guideline mentioned. Both relate to ensuring app security by making it difficult for attackers to analyze or modify application behavior at runtime. While MASVS-RESILIENCE-4 focuses on hardening the app to prevent dynamic analysis and instrumentation, the ENISA guideline emphasizes the importance of verifying dynamic code downloads and updates for integrity and authenticity. Both aim to protect against similar threats, namely the modification of the app or its behavior by an attacker.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the mentioned ENISA Guideline is that both address issues related to the secure handling of sensitive data. MASVS-STORAGE-2 focuses on preventing unintentional leaks of sensitive data often due to the misuse of APIs or system capabilities, such as backups or logs that might inadvertently expose data to publicly accessible locations. The ENISA Guideline emphasizes the importance of verifying the integrity and authenticity of dynamic code downloads and application updates, which is a measure to ensure that no malicious or altered code compromises the app or exposes sensitive data. Both controls aim to protect sensitive information from being compromised or unintentionally leaked, ensuring data integrity and security within the application.

13.11 Implementation Guidance (ENISA 12.11):

ENISA Secure Smartphone Development Guidance (12.11): Always validate server responses when using backend APIs. Introduce a whitelist model for accepted responses.

13.11.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation exists because ”MASVS-AUTH-1” emphasizes the importance of applications following best practices for secure communication with remote endpoints, which involves both user authentication and authorization. The ENISA Guideline advises to always validate server responses when using backend APIs and to use a whitelist model for accepted responses. This directly relates to ensuring that an app communicates securely with a remote endpoint by validating the received data, which is part of enforcing best practices as mentioned in ”MASVS-AUTH-1”. The whitelist approach is a specific best practice that contributes to secure use of protocols and APIs, which falls under the scope of ”MASVS-AUTH-1”.
- MASVS-CODE-4: Both the Mobile Application Security Verification Standard (MASVS) described by ”MASVS-CODE-4” and the European Union Agency for Cybersecurity (ENISA) guideline focus on the concept of treating input data as potentially untrustworthy and enforcing strict validation. ”MASVS-CODE-4” emphasizes the importance of data verification and sanitization for various data entry points to prevent attacks such as injection vulnerabilities. Similarly, the ENISA guideline advocates for validating server responses when interacting with backend APIs and using a whitelist model—which is essentially a form of validation—to ensure that only accepted responses are processed. Both prescribe a defensive approach to handling input data to maintain application security.
- MASVS-NETWORK-1: The correlation between ”MASVS-NETWORK-1” and the ENISA Guideline about validating server responses is based on the concept of ensuring the security of data in transit. ”MASVS-NETWORK-1” emphasizes the importance of protecting data privacy and integrity by encrypting data and authenticating the remote endpoint, as is done with protocols such as TLS. It warns against bypassing secure defaults by misusing low-level APIs or third-party libraries. The ENISA Guideline complements this by focusing on the importance of backend communication security, specifically by validating server responses and using a whitelist approach to only accept known, safe responses. Both guidelines aim to mitigate the risks associated with network communications and reinforce the principle of defense in depth.
- MASVS-NETWORK-2: Both ”MASVS-NETWORK-2” and the ENISA guideline address the principle of minimizing trust and explicitly validating external inputs or connections. ”MASVS-NETWORK-2” focuses on limiting trust to specific Certificate Authorities (CAs), which is a way to ensure that the application only establishes secure connections with trusted servers, effectively a form of whitelisting at the certificate level known as certificate pinning or public key pinning. Similarly, the ENISA guideline’s recommendation to validate server responses and introduce a whitelist model for accepted responses is about establishing trust boundaries and ensuring that the application only accepts expected and verified data. Both controls are rooted in the principle of least privilege and are designed to prevent an app from interacting with untrusted or potentially malicious entities. They emphasize the importance of an application vetting what it trusts, whether it is the identity of a server (via certificate pinning) or the data it receives (via response validation and whitelists).

- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1,” which is about ensuring secure interactions involving Inter-Process Communication (IPC) mechanisms, and the ENISA Guideline about validating server responses and using a whitelist model for accepted responses is present in the security principle of validating and restricting inputs and outputs. MASVS-PLATFORM-1 emphasizes the security of the interactions with IPC mechanisms, which could involve data exchange between apps or with the user. The ENISA Guideline focuses on the security of interactions with backend APIs, specifying that server responses should be validated and controlled through a whitelist. Both guidelines are designed to prevent unauthorized or unexpected data from being processed, which could lead to vulnerabilities or exploits. They both aim to ensure secure communication, whether it's between local app components (MASVS-PLATFORM-1) or with remote servers (ENISA Guideline), by enforcing strict validation and control of the data exchanged.
- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the ENISA guideline about validating server responses when using backend APIs is thematically linked through the focus on protecting sensitive data. MASVS-STORAGE-1 deals with ensuring that sensitive data, regardless of its origin—including from backend sources—is protected when stored locally on a device. This implies that data from the backend must be handled securely. The ENISA guideline's recommendation to validate server responses when using backend APIs complements this by suggesting that all data received from the backend (i.e., server responses) should be validated and filtered based on a whitelist model. Both aim to protect sensitive data from being compromised, with MASVS-STORAGE-1 focusing on storage aspects and the ENISA guideline on the secure communication and processing of that data.

13.12 Implementation Guidance (ENISA 12.12):

ENISA Secure Smartphone Development Guidance (12.12): Mitigate SQL injections, local file inclusion, JavaScript injections, XML injections. When dealing with dynamic queries (e.g., SQL queries with untrusted inputs) or Content-Providers ensure you are using parameterized queries. Always validate user provided inputs that will be used for file accessing purposes or as part of a dynamic code execution. Use a vetted framework for XML operations.

13.12.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The MASVS-AUTH-1 description emphasizes the importance of apps following best practices to ensure secure protocol use when connecting to a remote endpoint with authentication and authorization mechanisms. The ENISA Guideline is outlining specific best practices to mitigate various types of code injection attacks, such as SQL injection, which are relevant to secure protocol use. Parameterized queries, input validation, and using vetted frameworks for operations like XML manipulation are all examples of secure practices that are necessary to prevent vulnerabilities in authentication and authorization processes. Both the MASVS-AUTH-1 and the ENISA Guideline advocate for the implementation of security measures to protect against improper use of dynamic queries and user inputs that can compromise security, therefore they are correlated.
- MASVS-CODE-3: MASVS-CODE-3 discusses conducting a whitebox assessment and addressing low-hanging fruit vulnerabilities, which commonly include misconfigurations or known vulnerabilities in libraries. The ENISA Guideline's recommendation to mitigate SQL injections, local file inclusion, JavaScript injections, XML injections, and ensure the use of parameterized queries and validation of user inputs align with the notion of addressing common vulnerabilities. Both sets of advice are concerned with proactively identifying and mitigating security risks commonly found in application components, which could include detecting issues during a whitebox assessment as referred to in MASVS-CODE-3.
- MASVS-CODE-4: Both the description of "MASVS-CODE-4" and the ENISA Guideline emphasize the importance of treating data from various entry points as untrusted and the necessity for proper validation and sanitization to prevent injection attacks such as SQL injection, XSS (Cross-Site Scripting), and insecure deserialization. They advocate for the use of parameterized queries when handling dynamic inputs and stress the need to validate user-provided inputs used in file access or dynamic code execution. Both guidelines recommend using vetted frameworks for operations like working with XML, which underscores a shared goal of mitigating common security vulnerabilities related to untrusted input.
- MASVS-PLATFORM-1: The correlation between MASVS-PLATFORM-1 and the ENISA Guideline is that both focus on secure interactions with an app. MASVS-PLATFORM-1 pertains to secure Inter-Process Communication (IPC) mechanisms, ensuring that data and functionality exposure is controlled. ENISA's guideline addresses secure handling of dynamic queries and user inputs to mitigate various injection vulnerabilities. While not mentioned directly, IPC mechanisms can be a part of the broader scope of secure interactions with dynamic queries and user inputs that the ENISA guideline discusses.

Additionally, both highlight the importance of utilizing safe methods to interact with untrusted inputs, which can be part of IPC in mobile applications.

- MASVS-PLATFORM-2: Both the MASVS-PLATFORM-2 description and the ENISA Guideline emphasize the importance of secure configuration and input validation to protect against various forms of code injection attacks. Although they address slightly different aspects of security—MASVS-PLATFORM-2 focuses on securing WebViews and their interaction with native code, while the ENISA Guideline addresses secure coding practices for preventing common injection attacks—both advocate for proactive measures to prevent sensitive data leakage and ensure the safe execution of dynamic code, showing a correlation in their underlying security principles.
- MASVS-STORAGE-1: The MASVS-STORAGE-1 description indicates the importance of proper handling and storage of sensitive data from various sources to protect it irrespective of the storage location. This correlates with the ENISA Guideline, which focuses on mitigating various injection attacks and ensuring security when dealing with dynamic queries, user inputs, and XML operations. Both emphasize the need for validating input and using secure methods to handle sensitive data, which can involve proper parameterization of queries and using vetted frameworks to protect against attacks that could compromise stored sensitive data.
- MASVS-STORAGE-2: Both the MASVS-STORAGE-2 description and the ENISA Guideline focus on preventing unintended leakage of sensitive data due to developer oversight or improper use of APIs and system capabilities. MASVS-STORAGE-2 highlights the unintentional exposure of sensitive data in publicly accessible locations, and the ENISA guideline emphasizes the importance of mitigating various injection vulnerabilities, validating user inputs, and using secure frameworks. Both stress the importance of secure coding practices to protect against data exposure and vulnerabilities that could lead to data leaks.

13.13 Implementation Guidance (ENISA 12.13):

ENISA Secure Smartphone Development Guidance (12.13): Protect from memory corruptions in applications that are developed using a programming language which supports explicit memory management (e.g., Objective-C, C, C++). Perform static analysis for memory management vulnerabilities in the development process.

13.13.1 OWASP MASVS MAPPING

- MASVS-RESILIENCE-2: While MASVS-RESILIENCE-2 and the ENISA Guideline mentioned are not directly talking about the same security controls, they both relate to the concept of ensuring the integrity and security of the application's code and runtime environment. MASVS-RESILIENCE-2 is about preventing modifications to an application to maintain its intended functionality, which includes protecting the app from being modified to exhibit unintended behaviors, such as cheating in games or enabling premium features without payment. This can be achieved through techniques such as code obfuscation, checksums, or anti-tampering protections. On the other hand, the ENISA Guideline focuses on protecting applications from memory corruption vulnerabilities which can occur due to improper memory management in languages like C, C++, and Objective-C. Memory corruptions can be exploited to perform unauthorized actions such as code execution or privilege escalation, ultimately compromising the integrity of the app. Both guidelines aim to protect the integrity of the application by reducing the risk of unauthorized modifications or exploitation, albeit through addressing different aspects of application security. The MASVS-RESILIENCE-2 focuses on protecting the application from being tampered with, while the ENISA Guideline emphasizes the need to perform static analysis to prevent vulnerabilities that could lead to memory corruptions in applications. Both contribute to the resilience and security of the application in a user-controlled environment.
- MASVS-RESILIENCE-3: The Mobile Application Security Verification Standard (MASVS) Resilience requirement - MASVS-RESILIENCE-3 - is conceptually correlated with the ENISA guideline you provided. While MASVS-RESILIENCE-3 focuses on impeding the understanding of an app's internals to prevent tampering via static analysis, the ENISA guideline focuses on protecting against memory corruption vulnerabilities in applications, particularly those developed with languages that support explicit memory management. Both measures are forms of static analysis aimed at fortifying the app's resilience against security threats; MASVS-RESILIENCE-3 does this by obfuscation to prevent understanding, while the ENISA guideline does so by identifying and correcting vulnerabilities that could otherwise be exploited. Static code analysis is a common thread between them as both recommend analyzing the code without executing it, to identify weaknesses that could lead to security breaches.

13.14 Implementation Guidance (ENISA 12.14):

ENISA Secure Smartphone Development Guidance (12.14): Do not use insecure cached data in HTTP connections and in embedded web browsing capabilities (e.g., WebViews). Caches are usually located on the device file system. Many platforms allow applications to place this cached data to insecure locations (e.g., sdcard on Android) in which they can be easily tampered.

13.14.1 OWASP MASVS MAPPING

- MASVS-CODE-4: The MASVS-CODE-4 control and the ENISA guideline both emphasize the importance of treating data from various entry points—including the file system, as potentially untrusted and ensuring it is properly handled to prevent security issues. The MASVS-CODE-4 description mentions the importance of verifying and sanitizing incoming data to avoid injection attacks and security check bypasses, while the ENISA guideline advises against using insecure cached data from HTTP connections and embedded web browsing due to the risk of tampering. Both are concerned with the integrity and security of data that an app processes, particularly data that could be modified by untrusted sources.
- MASVS-CRYPTO-1: There is a correlation between ”MASVS-CRYPTO-1” and the described ENISA Guideline. ”MASVS-CRYPTO-1” emphasizes the importance of using cryptography to protect user data, especially on mobile devices where physical access is a likely threat. The ENISA guideline addresses a specific scenario where cached data, if insecurely stored (e.g., on an SD card), could be tampered with. Cryptography best practices would mandate that any sensitive cached data should be encrypted to prevent unauthorized access or tampering, aligning with the principle outlined in ”MASVS-CRYPTO-1” of securing user data through cryptographic means.
- MASVS-NETWORK-1: Both the ”MASVS-NETWORK-1” description and the ENISA Guideline pertain to securing data in transit and ensuring that security measures are not bypassed through improper implementation practices. ”MASVS-NETWORK-1” focuses on the necessity of encrypting data and authenticating remote endpoints to prevent interception or tampering during transit, which can be compromised if developers disable secure defaults or use insecure APIs and libraries. Similarly, the ENISA Guideline warns against using insecurely cached data, particularly in HTTP connections and WebViews, since caches might reside in locations—like external storage on Android—that are susceptible to tampering. Both stress the importance of maintaining data integrity and privacy by following secure coding practices to avoid the risks posed by storing or transmitting data insecurely.
- MASVS-PLATFORM-1: The correlation between ”MASVS-PLATFORM-1” and the described ENISA Guideline is that both are concerned with secure interactions and data management within a mobile app environment. MASVS-PLATFORM-1 is focused on securing IPC (Inter-Process Communication) mechanisms, which can involve data sharing and interactions between different components or apps. The ENISA Guideline addresses the security of cached data within HTTP connections or embedded browsers (e.g., WebViews), emphasizing that insecure caching could be tampered with, especially if stored in accessible locations like an Android SD card. Both guidelines aim to prevent unauthorized access or manipulation of data within the app’s ecosystem, which could lead to security vulnerabilities.

- MASVS-PLATFORM-2: The ”MASVS-PLATFORM-2” statement refers to the secure configuration of WebViews, which includes concerns about sensitive data leakage. The ENISA guideline specifically addresses the insecurity that can arise from cached data in HTTP connections and embedded web browsing (WebViews). Both are concerned with preventing exposure of sensitive data on the device, making them correlated. The guideline's mention of caches being placed in insecure locations speaks to the broader issue of secure WebView configuration, which would include managing how and where cached data is stored to prevent tampering or leakage.
- MASVS-RESILIENCE-1: Both statements emphasize the importance of operating on a secure platform to ensure the integrity and safety of the app's data. ”MASVS-RESILIENCE-1” stresses the need for the app to validate that the operating system has not been compromised to maintain trust in the platform's security features. The ENISA Guideline also focuses on the risks associated with using insecure cached data, as compromised locations like an insecure sdcard on Android can lead to tampered data. Both are concerned with the potential of tampering and the subsequent risk to data that compromises the app's security.
- MASVS-STORAGE-1: The MASVS-STORAGE-1 requirement which emphasizes the proper protection of sensitive data stored by the app, regardless of the storage location, correlates with the ENISA guideline advising against the use of insecure cached data in HTTP connections and WebViews. Both highlight the risk of storing sensitive information in locations that could be accessed or tampered with by unauthorized users or applications, such as public folders or the device's file system. The essence of both statements is to ensure data protection and secure storage practices for sensitive data within mobile applications.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA Guideline is that both focus on preventing unintentional exposure or storage of sensitive data in locations that might be accessible to unauthorized parties. While MASVS-STORAGE-2 discusses the broader range of potential leaks due to the use of APIs or system features, the ENISA guideline specifically warns against the insecure handling of cached data in HTTP connections and WebViews. Both guidelines are emphasizing the importance of securing data storage to prevent leaks and tampering, with MASVS-STORAGE-2 providing a general principle and the ENISA guideline offering an example of where this can occur.

13.15 Implementation Guidance (ENISA 12.15):

ENISA Secure Smartphone Development Guidance (12.15): In platforms that support custom applications with accessibility permissions (e.g., Android), exclude sensitive user interface elements from being accessed by accessibility applications.

13.15.1 OWASP MASVS MAPPING

- MASVS-AUTH-2: Both ”MASVS-AUTH-2” and the described ENISA Guideline concern the proper implementation of security measures in mobile applications with a focus on authentication and protection of sensitive user information. ”MASVS-AUTH-2” emphasizes the need for correctly implementing biometric or local PIN code authentication, which could become a security risk if accessibility permissions are improperly managed. The ENISA Guideline advises on excluding sensitive user interface elements from accessibility applications to prevent them from being accessed by potentially malicious applications. Both guidelines aim to protect user credentials and sensitive data against unauthorized access or leaking through weaker points of the system, such as incorrectly implemented authentication or exploitation of accessibility features.
- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3”, which mentions the need for additional forms of authentication for sensitive actions inside an app, and the ENISA guideline, which suggests excluding sensitive user interface elements from being accessed by accessibility applications, is based on a common concern for the security of sensitive operations within mobile applications. MASVS-AUTH-3 emphasizes the implementation of secure authentication mechanisms to protect against unauthorized actions, while the ENISA guideline aims to prevent accessibility services, which could potentially be misused by malicious applications, from interacting with sensitive user interface elements. Both stress the importance of safeguarding sensitive features, albeit from different angles: MASVS-AUTH-3 through additional authentication and the ENISA guideline through restrictions on interface element accessibility.
- MASVS-CODE-3: The MASVS-CODE-3 description talks about performing security assessments on app components, including scanning libraries for known vulnerabilities. This correlates with the ENISA guideline, which advocates for protecting sensitive user interface elements from being accessed by accessibility applications that could potentially have custom permissions on platforms like Android. Both emphasize the need to safeguard app components from potential security risks, the first through assessing vulnerabilities and the second by restricting access to sensitive elements.
- MASVS-PLATFORM-1: ”MASVS-PLATFORM-1” talks about secure interactions involving IPC (Inter-Process Communication) mechanisms. IPC mechanisms are a subset of the functionality that platforms provide to enable apps to interact with other apps or the system. The ENISA guideline refers to accessibility applications, which often use IPC mechanisms to interact with other apps and the system, especially for accessing user interface elements. Both the MASVS control and the ENISA guideline emphasize the security of interactions that can potentially expose sensitive data, but they approach it from different angles. MASVS-PLATFORM-1 focuses on securing all IPC interactions, while the ENISA guideline specifically advises on limiting accessibility permissions that could expose sensitive UI elements to accessibility applications.

- MASVS-PLATFORM-2: The correlation between ”MASVS-PLATFORM-2” and the ENISA Guideline lies in the focus on preventing sensitive data leakage and limiting sensitive functionality exposure. ”MASVS-PLATFORM-2” addresses the secure configuration of WebViews, which can include content like JavaScript bridges to native code that, if misconfigured, could allow for sensitive data to be compromised. Similarly, the ENISA Guideline speaks to the need for excluding sensitive user interface elements from being accessed by accessibility applications that are granted extensive permissions on platforms like Android. Both stress the importance of securing user interface components to protect sensitive information, although they approach the problem from slightly different angles—one from the perspective of WebViews and the other from accessibility features.
- MASVS-PLATFORM-3: The correlation is present because both the MASVS-PLATFORM-3 and the ENISA guideline address the protection of sensitive data from unintentional leakage through mechanisms that could capture or access this data outside the intended context. MASVS-PLATFORM-3 is concerned with preventing sensitive data from being leaked through automatic screenshots by the platform or being seen by bystanders (shoulder surfing), while the ENISA guideline aims to prevent sensitive UI elements from being accessed by accessibility applications, which is another vector through which sensitive information could be unintentionally disclosed. Both are focused on securing sensitive user data within the application’s user interface.
- MASVS-PRIVACY-1: Both MASVS-PRIVACY-1 and the ENISA Guideline are concerned with restricting and controlling access to sensitive data and ensuring informed consent from the user. MASVS-PRIVACY-1 emphasizes that apps should only request access to data that is necessary for functionality, enforce third-party SDKs to operate based on user consent, and be aware of their SDK ’supply chain’. The ENISA Guideline underscores preventing accessibility applications from accessing sensitive user interface elements on systems where custom applications can be granted such permissions. Both aim to limit exposure of sensitive data and enhance privacy, and there’s clear correlation in their objective to minimize access to user data without informed consent.
- MASVS-PRIVACY-3: There is a correlation between ”MASVS-PRIVACY-3” and the ENISA Guideline mentioned. Both are concerned with the protection of user data and transparency regarding how apps handle this data. MASVS-PRIVACY-3 emphasizes the users’ right to know how their data is utilized, which includes being aware of any unexpected data usage such as background collection. In line with this, the ENISA Guideline points out the need to prevent accessibility applications from accessing sensitive user interface elements, which could be exploited to harvest user data unknowingly. Therefore, adhering to MASVS-PRIVACY-3 would implicitly support the principles behind the ENISA Guideline by ensuring that sensitive data is not unexpectedly accessed or shared, thus upholding user privacy and trust.
- MASVS-PRIVACY-4: While the specific wording might differ, MASVS-PRIVACY-4 and the referenced ENISA Guideline share the underlying principle of protecting user privacy and data control. MASVS-PRIVACY-4 emphasizes the importance of user control over their own data, including mechanisms for managing consent and privacy settings. The ENISA Guideline addresses the need for excluding sensitive user interface elements from accessibility applications, which also pertains to user privacy, as it prevents unauthorized access or control by third-party applications, especially those with extensive permissions such as accessibility permissions. Both guidance points aim to enhance user data protection and prevent misuse of personal information by applications, reflecting a correlation in their objectives to safeguard user privacy.

- MASVS-RESILIENCE-1: The correlation between MASVS-RESILIENCE-1 and the ENISA Guideline regarding platforms that support custom applications with accessibility permissions lies in the concept of ensuring the integrity and trustworthiness of the platform. MASVS-RESILIENCE-1 emphasizes the risk of running an app on a tampered platform, which can compromise security features such as secure storage and sandboxing. The ENISA Guideline focuses on a specific aspect of this—excluding sensitive user interface elements from being accessed by accessibility applications—to prevent misuse. Both are concerned with the resilience of the app in the face of potential security breaches that could stem from a compromised platform.
- MASVS-RESILIENCE-3: The correlation exists because both ”MASVS-RESILIENCE-3” and the ENISA Guideline mentioned focus on hindering potential attackers from gaining insights into an application’s internal workings. ”MASVS-RESILIENCE-3” specifically mentions making static analysis difficult, which aligns with preventing accessibility applications (that can potentially be malicious or exploited) from accessing sensitive user interface elements as suggested by ENISA. Both measures aim to block avenues through which an attacker might learn the app’s behavior or structure, thereby impeding tampering or exploitation efforts.
- MASVS-RESILIENCE-4: The MASVS-RESILIENCE-4 requirement and the ENISA guideline both address the concept of restricting unauthorized access to sensitive features of applications. The MASVS-RESILIENCE-4 requirement focuses on making it difficult for attackers to perform dynamic analysis and instrumentation that could potentially modify app behavior at runtime. Meanwhile, the ENISA guideline advises excluding sensitive user interface elements from accessibility applications on platforms that support such permissions. Both aim to protect against tampering or unauthorized interaction with the app, which shows a correlation in their intention to safeguard application integrity and user data security.
- MASVS-STORAGE-2: The correlation between ”MASVS-STORAGE-2” and the ENISA Guideline is that both are concerned with the protection of sensitive data from unintended exposure. ”MASVS-STORAGE-2” addresses the unintentional storage or exposure of sensitive data due to the use of APIs or system capabilities, which can include logs or backups that might be publicly accessible. Similarly, the ENISA Guideline aims to prevent sensitive user interface elements from being accessed by applications with accessibility permissions, which is another form of protection against unintentional exposure of sensitive information. Both guidelines emphasize the proactive steps developers should take to ensure that sensitive data is not exposed or accessible by unauthorized entities or through unintended means.

13.16 Implementation Guidance (ENISA 12.16):

ENISA Secure Smartphone Development Guidance (12.16): Avoid populating webviews loaded from the file URI scheme with user supplied DOM input.

13.16.1 OWASP MASVS MAPPING

- MASVS-CODE-2: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-CODE-2” indicates the need for a mechanism to force updates in case of critical vulnerabilities, which may include those found in webviews. The ENISA guideline to avoid populating webviews with user-supplied DOM input when loaded from the file URI scheme is related to preventing such vulnerabilities. Failure to update could leave vulnerable webviews exposed to exploitation, thus correlating with the need to enforce updates described in MASVS-CODE-2.
- MASVS-CODE-4: The correlation exists because ”MASVS-CODE-4” covers the principle of treating all data entering the application as untrusted and requiring validation and sanitization to prevent injection attacks and other security issues. The ENISA Guideline about avoiding populating webviews with user-supplied DOM input from the file URI scheme directly relates to this, as it is an example of a potential attack vector where untrusted user input could lead to attacks like cross-site scripting (XSS) if the webview were to improperly handle this input. By adhering to the ”MASVS-CODE-4” control, the risks described in the ENISA Guideline would be mitigated.
- MASVS-NETWORK-1: Both ”MASVS-NETWORK-1” and the ENISA guideline emphasize the importance of maintaining the security and integrity of data when an app communicates over the network. - MASVS-NETWORK-1 focuses on the necessity of preserving data privacy and integrity in transit by employing encryption and endpoint authentication, which are fundamental to avoid interception or tampering by malicious entities. - The ENISA guideline regarding webviews and the file URI scheme pertains to preventing potential security vulnerabilities, such as cross-site scripting (XSS) attacks, which can compromise data integrity and privacy. By advising against populating webviews with user-supplied input, it ensures that secure communication practices are upheld and that insecure inputs do not undermine the app’s defenses. Both points address the broader principle of safeguarding data in a networked environment as part of the app’s design and implementation.
- MASVS-NETWORK-2: The correlation between ”MASVS-NETWORK-2” with its description of trusting only specific CAs through certificate pinning, and the ENISA guideline advising against populating webviews loaded from the file URI scheme with user-supplied DOM input lies in the emphasis on security and the mitigation of potential risks due to trust in external entities or user input. Certificate pinning as described in ”MASVS-NETWORK-2” reduces the threat of a man-in-the-middle (MITM) attack by ensuring that the application communicates only with the designated server. This is directly related to network security and the trust chain, as it prevents the application from accepting certificates from CAs that might be compromised or malicious. The ENISA guideline aims to avoid security risks associated with mixing user content and local file access within webviews. By not allowing user-supplied DOM input in webviews that load content using the file URI scheme, the guideline prevents potential cross-site scripting (XSS) attacks or other forms of data injection that could compromise the app or the user’s data. Both controls focus on narrowing

down the trust boundaries within an application: MASVS-NETWORK-2 does so for the network connections and certificate authorities, while the ENISA guideline does so for the processing of local and user-supplied content. Each measure reduces the application's attack surface and enhances its resilience against certain types of security threats.

- MASVS-PLATFORM-1: The correlation between "MASVS-PLATFORM-1" and the ENISA guideline "Avoid populating webviews loaded from the file URI scheme with user-supplied DOM input" lies in the realm of secure interaction and communication between components within an app or between different apps. "MASVS-PLATFORM-1" is concerned with ensuring that IPC (Inter-Process Communication) is secured. IPC mechanisms are a way for different processes to communicate with each other. If not properly secured, an IPC endpoint may expose sensitive data or functionality to malicious apps or actors. The ENISA guideline advises against using user-supplied DOM (Document Object Model) input within webviews that are loaded from the file URI scheme. This is because the file URI scheme can access local files, and including user-supplied input without proper validation can lead to security issues like cross-site scripting (XSS). This guideline thus targets a specific instance of IPC, where web content (perhaps executing in a webview component) might interact insecurely with the local file system or other app components. Both directives aim to prevent unauthorized or insecure access to app resources. They are correlated through their common goal of protecting app data and functionality from being compromised through IPC, whether it's platform-provided mechanisms in general or specific cases such as the use of webviews and file URIs.
- MASVS-PLATFORM-2: Both "MASVS-PLATFORM-2" and the ENISA Guideline express concerns about the security implications of using WebViews within mobile apps. "MASVS-PLATFORM-2" emphasizes the need for secure configuration of WebViews to prevent data leakage and exposuring sensitive functionality, such as JavaScript bridges that may interact with native code. Similarly, the ENISA Guideline specifically warns against using the file URI scheme in WebViews with user-supplied DOM input, which also constitutes a potential security risk that could lead to, among other issues, data leakage and execution of malicious scripts. The correlation here is that both provide guidelines towards the secure use of WebViews as a measure to protect sensitive data and the integrity of the app.
- MASVS-PLATFORM-3: Both "MASVS-PLATFORM-3" and the ENISA guideline "Avoid populating webviews loaded from the file URI scheme with user supplied DOM input" are concerned with protecting sensitive user data from unintentional leaks. "MASVS-PLATFORM-3" addresses the need to prevent sensitive data captured in the UI from leaking through platform mechanisms like screenshots or shoulder surfing. In a similar vein, the ENISA guideline aims to prevent potential data leakage by recommending against populating webviews with user data that might be injected into the DOM when using the less secure file URI scheme. Both pieces of guidance are centered around minimizing the exposure of sensitive data to unauthorized parties.
- MASVS-STORAGE-1: The description of "MASVS-STORAGE-1" talks about the proper protection of sensitive data stored by the app in various locations, whether they are private or public. This implies that precautions must be taken to ensure that sensitive data is handled securely to prevent unauthorized access or leakage. The ENISA Guideline's advice to "Avoid populating webviews loaded from the file URI scheme with user supplied DOM input" is related because injecting user-supplied input into webviews can lead to a range of security issues, including the exposure of sensitive stored data. If an app uses webviews and populates them with content from file-based storage, it is crucial to properly sanitize and control the input to avoid security vulnerabilities such as cross-site scripting (XSS) that

could compromise stored sensitive data. Both guidelines are concerned with the protection of sensitive data in the context of storage and rendering within an application.

- MASVS-STORAGE-2: Both ”MASVS-STORAGE-2” and the ENISA guideline mentioned concern the proper handling of sensitive data to prevent unintentional leaks. While MASVS-STORAGE-2 focuses on preventing sensitive data from being stored or exposed in publicly accessible locations as a side-effect of using certain APIs or system capabilities, the ENISA guideline advises against using user-supplied DOM input in webviews loaded from the file URI scheme, which can also result in the exposure of sensitive data. Both pieces of advice aim to protect sensitive information from being leaked due to improper coding practices or inadequate security measures in mobile applications.

Chapter 14
Ensure correct usage of biometric sensors and secure hardware

Biometric sensors make authentication systems both easier and faster to use, however the authentication and accessibility policies must be enforced by the secure hardware in order to be protected against anything up to and including kernel compromise.

14.1 Implementation Guidance (ENISA 13.1):

ENISA Secure Smartphone Development Guidance (13.1): Always verify that there is a biometric sensor (e.g., Fingerprint reader) present and available on the device before using the API for authentication purposes. In the case that the sensor is not available, an alternative authentication control should be provided.

14.1.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between "MASVS-AUTH-1" and the ENISA guideline is that both emphasize the importance of secure authentication practices in mobile applications. While "MASVS-AUTH-1" broadly states that apps connecting to a remote endpoint require user authentication and must adhere to best practices for protocol use, the ENISA guideline sets a specific requirement for the use of biometric sensors when available, ensuring secure authentication by leveraging device hardware capabilities. Both guidelines aim to ensure that proper authentication mechanisms are in place to protect access to the application's data and functions, supplementing server-side authorization controls.
- MASVS-AUTH-2: Both "MASVS-AUTH-2" and the ENISA guideline are addressing the proper implementation and considerations around the use of biometric authentication in mobile applications. "MASVS-AUTH-2" stresses the importance of correctly implementing biometric or local PIN code authentication, which implies ensuring that such features should only be utilized if the appropriate hardware (biometric sensors) are present on the device, aligning with ENISA's guidance to verify the presence and availability of the biometric sensor before using it for authentication purposes. Furthermore, ENISA suggests providing an alternative authentication control if the sensor is not available, which is consistent with the MASVS-AUTH-2's underlying premise for robust and fallback authentication mechanisms, especially in apps without a remote endpoint.
- MASVS-AUTH-3: The correlation exists because both statements emphasize secure implementation of additional authentication mechanisms for sensitive actions within an app. "MASVS-AUTH-3" broadly suggests using secure additional forms of authentication like biometric, pin, MFA, etc., while the ENISA guideline specifically advises verifying the presence of a biometric sensor before utilizing its API for authentication and suggests providing an alternative control if the sensor is not available. They both advocate for enhancing security by ensuring robust authentication means are in place when needed.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the ENISA guideline can be reasoned as follows: The description of "MASVS-RESILIENCE-1" emphasizes the importance of the app running on a secure platform where the integrity of the operating system is not compromised. It specifically mentions that trust in the platform security features (such as secure storage, biometrics, sandboxing, etc.) is crucial, as a compromised OS can put app data at risk. The ENISA guideline advises developers to verify the presence and availability of a biometric sensor before using it for authentication purposes, suggesting a fallback to alternative authentication controls if the sensor is unavailable. The correlation exists because both stress on verifying security features (biometric sensor integrity in the ENISA guideline and platform integrity in MASVS-RESILIENCE-1) as a precondition for secure operations. Failure to ensure the proper functioning and availability of security features, including biometrics, could be indicative of a tampered or compromised platform, which "MASVS-RESILIENCE-1" is designed to protect against.

Therefore, adherence to ”MASVS-RESILIENCE-1” would implicitly fulfill the ENISA guideline’s requirement of verifying the availability and integrity of the biometric sensor as part of ensuring platform security.

14.2 Implementation Guidance (ENISA 13.2):

ENISA Secure Smartphone Development Guidance (13.2): Always verify that the biometric sensor/secure hardware authentication policy of the in-use platform complies with the application's authentication policy (passcode required after cold boot, biometric sensor authentication expiration, adding a fingerprint requires pin/passcode/biometric authentication, requirement for biometric sensor being individually paired with secure hardware).

14.2.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The ENISA Guideline's emphasis on ensuring that the biometric sensor/secure hardware authentication policy aligns with the application's authentication policy aligns with MASVS-AUTH-1's description which stresses the importance of apps following best practices for secure use of authentication and authorization protocols. Both highlight the need for rigorous verification mechanisms to maintain security in authentication processes.
- MASVS-AUTH-2: The Mobile Application Security Verification Standard (MASVS) requirement "AUTH-2" that mentions the need for proper implementation of biometric and local PIN code authentication mechanisms correlates with the ENISA Guideline that emphasizes the verification of the biometric sensor or secure hardware authentication policy aligning with the app's authentication policy. Both recommend ensuring that the application's local authentication mechanisms adhere to secure practices and that they are integrated in a manner consistent with the overall security policy of the platform. The guideline's specific mention of policies such as requiring a passcode after cold boot, biometric authentication expiration, ensuring secure addition of new fingerprints, and the pairing of the biometric sensor with secure hardware all pertain to the correct implementation of such features, which is what MASVS-AUTH-2 implies.
- MASVS-AUTH-3: Both MASVS-AUTH-3 and the ENISA Guideline emphasize the importance of implementing additional secure authentication mechanisms for sensitive actions within an application. MASVS-AUTH-3 refers to various methods such as biometrics, PIN, and MFA, while the ENISA Guideline specifically focuses on verifying that biometric sensors or secure hardware authentication comply with the application's authentication policy. The underlying correlation is the requirement of enhanced security measures for authentication processes in mobile applications.
- MASVS-RESILIENCE-1: The "MASVS-RESILIENCE-1" control and the ENISA Guideline both emphasize the importance of ensuring the integrity and security features of the underlying platform for the proper operation of security controls. Specifically, "MASVS-RESILIENCE-1" underlines the danger of running applications on a tampered platform, as it can compromise security features like secure storage, biometrics, and sandboxing. The ENISA Guideline reinforces this by advising the verification of the biometric sensor/secure hardware authentication policies to ensure they align with the application's authentication policy. Both advocate the reliability of platform security features, which are critical for trustworthy biometric authentication and other security functionalities that depend on the platform's security posture.

14.3 Implementation Guidance (ENISA 13.3):

ENISA Secure Smartphone Development Guidance (13.3): Ensure that there are enrolled data using the biometric sensor (e.g., user's fingerprints and/or user's iris are registered) before using the API for authentication purposes.

14.3.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between "MASVS-AUTH-1" and the ENISA Guideline exists because both are concerned with secure user authentication practices within mobile applications. MASVS-AUTH-1 emphasizes the importance of following best practices for secure use of authentication protocols, which would implicitly include ensuring proper enrollment of biometric data when biometric sensors are used for authentication purposes. Hence, the ENISA Guideline's focus on verifying that biometric data are enrolled before using the biometric authentication API aligns with the principle of secure authentication practices as mentioned in MASVS-AUTH-1.
- MASVS-AUTH-2: The correlation between "MASVS-AUTH-2" and the ENISA guideline lies in the focus on the proper implementation of local authentication mechanisms such as biometrics. "MASVS-AUTH-2" emphasizes the need for correct implementation of biometric or local PIN code authentication, while the ENISA guideline specifies that before using the biometric API for authentication, it is essential to ensure that biometric data, such as fingerprints or iris, are enrolled. Both address the importance of verifying the presence and proper setup of the user's biometric data for authentication, which fundamentally underpins the security effectiveness of biometric authentication within the app.
- MASVS-AUTH-3: The correlation exists because both MASVS-AUTH-3 and the ENISA Guideline emphasize the importance of additional authentication mechanisms for sensitive actions within an app. MASVS-AUTH-3 acknowledges the need for varied secure implementations of additional authentication, such as biometric or multi-factor authentication (MFA), which align with the ENISA Guideline that specifically calls for ensuring that biometric data, like fingerprints or iris information, is enrolled before utilizing biometric authentication APIs. Both guidelines are concerned with the secure use of enhanced authentication methods to protect sensitive functions and data.
- MASVS-RESILIENCE-1: The "MASVS-RESILIENCE-1" description emphasizes the importance of running apps on a secure, unmodified platform, as tampering can disable security features and put app data at risk. Ensuring the integrity of the operating system supports trust in security controls such as secure storage and biometrics. The ENISA guideline about ensuring that biometric data is enrolled before using the API for authentication purposes is related because it also concerns the trust in security features—specifically biometrics. If the operating system were compromised, the security of the biometric authentication process could be at risk, leading to potential misuse of biometric data or bypass of authentication. Therefore, "MASVS-RESILIENCE-1" and the described ENISA guideline are correlated as they both deal with ensuring the reliability and security of authentication mechanisms that rely on the platform's integrity.

14.4 Implementation Guidance (ENISA 13.4):

ENISA Secure Smartphone Development Guidance (13.4): Ensure that the enrolled biometric data has not been changed since the activation of the authentication control using the biometric sensor (e.g., another user added a new fingerprint/iris sample).

14.4.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the mentioned ENISA guideline is that they both emphasize the importance of maintaining the security and integrity of authentication mechanisms. ”MASVS-AUTH-1” specifies that the app should adhere to best practices in the secure use of protocols related to user authentication and authorization, while the ENISA guideline focuses on ensuring that enrolled biometric data, which is a form of user authentication, remains unchanged to prevent unauthorized access. Both address the reliability and trustworthiness of authentication measures within the context of securing mobile applications.
- MASVS-AUTH-2: ”MASVS-AUTH-2” and the ENISA guideline both emphasize the importance of correctly implementing authentication mechanisms that use biometric data. While MASVS-AUTH-2 does not directly mention the need to check for changes in enrolled biometric data since the activation of the biometric sensor, the recommendation from the ENISA guideline can be seen as a specific instance or requirement that falls under the broader category of proper implementation of biometric authentication as described by MASVS-AUTH-2. Ensuring that enrolled biometric data has not been tampered with is part of ”correctly implementing” these biometric authentication mechanisms.
- MASVS-AUTH-3: The correlation is that both MASVS-AUTH-3 and the ENISA guideline emphasize the security of authentication measures within an app. MASVS-AUTH-3 suggests the use of additional forms of authentication such as biometrics, which aligns with the ENISA guideline’s focus on ensuring the integrity of enrolled biometric data to prevent unauthorized changes or additions. Both are concerned with maintaining the security and reliability of sensitive actions that require authentication.
- MASVS-PLATFORM-1: MASVS-PLATFORM-1 is concerned with securing Inter-Process Communication (IPC) mechanisms provided by the platform to ensure that data or functionality exposed through IPC is accessed securely. The ENISA Guideline on ensuring that enrolled biometric data has not been changed aligns with the intent of MASVS-PLATFORM-1, because both guidelines aim to protect sensitive operations and data integrity within the app. MASVS-PLATFORM-1 does so by securing IPC mechanisms, which could be involved in the process of managing biometric data and enforcing authentication controls, while the ENISA Guideline specifically focuses on the integrity of biometric data, a type of sensitive data, ensuring it remains unchanged once authentication controls are set. They are correlated in their common agenda of securing the app’s sensitive data and interactions from unauthorized access and modification.
- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the mentioned ENISA Guideline is that both focus on maintaining the integrity and security of the platform on which the application runs. MASVS-RESILIENCE-1 emphasizes the importance of running the app on a secure platform that has not been compromised, as a tampered platform could disable security features that are crucial for protecting the app’s data and relying on other security controls. Similarly, the ENISA Guideline addresses the

need to ensure that biometric data hasn't been altered to maintain the security of biometric authentication controls. Both are concerned with preventing unauthorized modifications that could undermine the security of the application and its authentication mechanisms.

14.5 Implementation Guidance (ENISA 13.5):

ENISA Secure Smartphone Development Guidance (13.5): The application should not use the biometric sensor for just verifying user presence (e.g., iOS LocalAuthentication). This control can be easily circumvented using dynamic hooking/static patching. Instead, the application should use the biometric sensor to access keys stored using a hardware backed keystore/keychain and protected with keychain access control lists (ACL).

14.5.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA Guideline exists as both pieces of guidance emphasize the importance of secure user authentication mechanisms in mobile apps. ”MASVS-AUTH-1” suggests that applications must follow best practices for secure communication with remote endpoints, which includes secure user authentication. The ENISA Guideline complements this by providing a specific recommendation that biometric sensors should be used not just for verifying presence, but to facilitate access to keys in a secure hardware-backed keystore, thus enhancing the security of the user authentication process. This is in line with the “secure use of the involved protocols” as referenced in MASVS-AUTH-1, ensuring that the biometric data handling and user authentication processes are protected against common attacks like dynamic hooking and static patching.
- MASVS-AUTH-2: Both ”MASVS-AUTH-2” and the provided ENISA Guideline emphasize the need for correct implementation of biometric and other local authentication mechanisms in mobile applications. ”MASVS-AUTH-2” highlights the necessity of proper implementation without specifying methods, whereas the ENISA Guideline gives a more detailed security consideration by advising against mere user presence verification through biometric sensors (which can be tricked) and recommending the use of biometrics in conjunction with a hardware-backed keystore/keychain and ACLs to enhance security. Both aim to guard against simple bypasses of biometric checks and ensure that biometric authentication contributes to the application’s security posture.
- MASVS-AUTH-3: The statement in ”MASVS-AUTH-3” regarding the desirability of additional forms of authentication for sensitive actions within an app correlates with the ENISA guideline concerning the use of biometric sensors. The MASVS standard recognizes a range of secure methods for implementing extra authentication layers, while the ENISA guideline elaborates on one such method: using biometric sensors properly to access keys in a hardware-backed secure storage, as opposed to only using biometrics for user presence verification. Both emphasize the secure implementation of additional authentication measures.
- MASVS-CRYPTO-1: The MASVS-CRYPTO-1 guideline emphasizes the importance of cryptography in securing user data on mobile devices, which is an environment prone to physical access by attackers. The ENISA Guideline’s advice to use biometric sensors to access keys stored in a hardware-backed keystore relates to implementing a robust cryptographic mechanism to protect sensitive operations and user data. Using a biometric sensor for mere user presence verification is considered weak and bypassable, whereas leveraging it as part of a cryptographic operation aligns with best practices for ensuring

data security, as advocated by MASVS-CRYPTO-1. The correlation is in the underlying principle of employing strong cryptography to safeguard user data on mobile devices.

- MASVS-CRYPTO-2: The correlation between "MASVS-CRYPTO-2" and the ENISA Guideline is that both are concerned with the proper management and protection of cryptographic keys. While MASVS-CRYPTO-2 speaks broadly about managing cryptographic keys throughout their lifecycle, the ENISA Guideline details a specific use case about utilizing biometric sensors not just for verifying user presence but for accessing keys stored in a hardware-backed keystore/keychain. Both guidelines aim to ensure that cryptographic keys are managed securely to prevent them from being compromised, which is in line with the spirit of MASVS-CRYPTO-2's description of managing keys securely throughout their lifecycle. Using biometric sensors to access hardware-backed keys adds an additional layer of security as recommended by both guidelines.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the ENISA guideline is that both emphasize the importance of not solely relying on the platform's security features when they could be compromised. MASVS-RESILIENCE-1 warns about the dangers of running an app on a tampered platform as it may disable security features, thus endangering the data of the app. Accordingly, it suggests validating that the OS has not been compromised. Similarly, the ENISA guideline advises against using biometric sensors merely for user presence verification, as this can be bypassed by methods such as dynamic hooking or static patching. It recommends using biometric sensors in conjunction with a hardware-backed keystore/keychain and ACLs to ensure security even if the platform's other security features are compromised. Both are concerned with ensuring the integrity of the operating system and leveraging hardware-backed security mechanisms.
- MASVS-RESILIENCE-3: The MASVS-RESILIENCE-3 guideline and the ENISA Guideline both address the prevention of tampering and ensuring the security of an application's operations. MASVS-RESILIENCE-3 focuses on impeding comprehension of the app's internals through static analysis to prevent tampering, while the ENISA Guideline provides a specific control to prevent circumvention using dynamic hooking or static patching by recommending the use of biometric sensor in conjunction with hardware-backed keystore/keychain. Both guidelines are concerned with preventing unauthorized access and modifications, thereby they correlate in their aim to enhance app resilience against security threats.
- MASVS-RESILIENCE-4: The MASVS-RESILIENCE-4 requirement from the Mobile Application Security Verification Standard (MASVS) indicates that an app should be made resilient against dynamic analysis and runtime manipulation, making it harder for attackers to perform such actions. The ENISA Guideline suggests that an application should use hardware-backed keystore/keychain with keychain ACLs for biometric sensor authentication, rather than just verifying user presence, because the latter can be circumvented using dynamic hooking or static patching, which are methods of dynamic and static analysis. Both the MASVS requirement and the ENISA Guideline emphasize the importance of protecting against dynamic analysis or modification to ensure security, indicating a correlation between the two.

14.6 Implementation Guidance (ENISA 13.6):

ENISA Secure Smartphone Development Guidance (13.6): Ensure that the key material is bound to the secure hardware (e.g., TEE, SE) in platforms that this is optional (e.g., Android). When this feature is enabled for a key, its key material is never exposed outside of secure hardware.

14.6.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between ”MASVS-AUTH-1” and the ENISA guideline described is that they both emphasize security measures in the context of authentication and protection of sensitive operations. ”MASVS-AUTH-1” refers to the need for secure user authentication and the necessity for apps to follow best practices when communicating with remote endpoints, which implicitly includes proper handling and protection of authentication credentials and key material. The ENISA guideline specifically addresses the need to bind key material to secure hardware elements such as a Trusted Execution Environment (TEE) or Secure Element (SE) in order to enhance the security of authentication processes and protect against potential attacks that can expose key material. Both guidelines promote securing authentication mechanisms and sensitive operations to mitigate risks associated with compromised security on the client side.
- MASVS-AUTH-2: The Mobile Application Security Verification Standard (MASVS) requirement ”MASVS-AUTH-2” and the ENISA guideline both emphasize the importance of secure and correctly implemented authentication mechanisms. MASVS-AUTH-2 highlights the need for proper implementation of local authentication methods like biometrics or PIN codes, which may involve secure handling of authentication credentials within the device, possibly using secure hardware elements like a Trusted Execution Environment (TEE) or Secure Element (SE). The ENISA guideline specifically addresses the security of key material by ensuring it is bound to secure hardware. The correlation lies in the emphasis on using hardware-based security features to enhance the integrity and confidentiality of the authentication process. Binding key material to secure hardware is one possible implementation detail of MASVS-AUTH-2’s requirement for properly implemented authentication mechanisms.
- MASVS-AUTH-3: MASVS-AUTH-3 suggests implementing additional forms of authentication securely for sensitive actions, which is related to the ENISA Guideline’s emphasis on binding key material to secure hardware like TEE or SE. Both focus on enhancing security in operations that might expose sensitive information or key material. Binding keys to secure hardware is a technical strategy to implement secure additional authentications as suggested by MASVS-AUTH-3.
- MASVS-CRYPTO-1: The given description of ”MASVS-CRYPTO-1” highlights the importance of cryptography in securing user data on mobile devices, especially in scenarios where physical access by attackers is possible. The ENISA Guideline’s recommendation to bind key material to secure hardware like a Trusted Execution Environment (TEE) or Secure Element (SE) aims to ensure that the cryptographic keys are protected by hardware barriers, preventing them from being exposed outside of a secure environment. This is in line with the MASVS-CRYPTO-1’s emphasis on following best practices for cryptography to protect user data, as one of those best practices would indeed be the secure handling and storage of cryptographic keys, which the ENISA guideline specifies.

- MASVS-CRYPTO-2: The MASVS-CRYPTO-2 control references the importance of proper key management throughout the lifecycle, including protection during storage. The ENISA Guideline complements this by specifying a method for protecting key material through binding it to secure hardware, such as a Trusted Execution Environment (TEE) or Secure Element (SE) on platforms where this is not mandatory, thereby ensuring the key material remains protected and is not exposed outside of the secure hardware. Both are concerned with the secure management and protection of cryptographic keys.
- MASVS-NETWORK-1: The control MASVS-NETWORK-1 and the ENISA Guideline both focus on the security and integrity of data. MASVS-NETWORK-1 addresses the need to securely handle data in transit through encryption and proper endpoint authentication to prevent disabling or bypassing security defaults. The ENISA Guideline emphasizes the protection of key material by binding it to secure hardware, like a Trusted Execution Environment (TEE) or Secure Element (SE), particularly on platforms where this is not mandatory. Both requirements relate to ensuring that sensitive data remains confidential and unaltered during transmission and within the operational environment. While MASVS-NETWORK-1 is more general, covering secure connections, the ENISA Guideline is more specific, referring to the secure handling of cryptographic keys. Nonetheless, both contribute to the overall goal of maintaining data privacy and integrity in mobile apps, with MASVS-NETWORK-1 providing a broader framework and the ENISA Guideline offering a detailed method for securing cryptographic keys.
- MASVS-PLATFORM-3: While MASVS-PLATFORM-3 focuses on preventing unintentional leaks of sensitive data through platform mechanisms like auto-generated screenshots, the ENISA guideline emphasizes binding key material to secure hardware. The underlying correlation is the emphasis on protecting sensitive data—be it user credentials or encryption keys—from being exposed outside of a secure context. Both guidelines are concerned with security measures on mobile platforms to prevent sensitive information from unintended disclosure. MASVS-PLATFORM-3 is about UI elements and temporary data exposure, whereas the ENISA guideline is about permanent security measures for key material, but the purpose of preventing sensitive data leakage aligns them closely.
- MASVS-RESILIENCE-1: The mentioned MASVS-RESILIENCE-1 and its description highlight the importance of running an application on a secure, uncompromised platform, emphasizing that many security controls rely on the platform's trustworthiness to function correctly. This includes secure storage and sandboxing which are generally provided by the platform's secure hardware components such as Trusted Execution Environments (TEE) or Secure Elements (SE). The ENISA Guideline complements this by stating that key material should be bound to secure hardware, which is an action that depends on the platform's integrity. If a platform has been tampered with, the security features like TEE or SE may be compromised, and key material could be exposed, posing a risk similar to what is described in MASVS-RESILIENCE-1. Both the MASVS-RESILIENCE-1 and the ENISA Guideline highlight the critical role of platform security in protecting application data. They correlate in that they both emphasize the necessity of retaining the security features of the operating system and secure hardware to ensure that sensitive information, such as cryptographic keys, remains protected. Binding key material to secure hardware assumes and requires that the platform has not been compromised, aligning with the goals of MASVS-RESILIENCE-1.
- MASVS-RESILIENCE-2: The correlation between "MASVS-RESILIENCE-2" and the ENISA guideline about key material being bound to secure hardware is related to the objective of ensuring the integrity and security of the app and its data on a user-controlled device. The MASVS-RESILIENCE-2 focuses on preventing modifications to the app's code

and resources to maintain its integrity, while the ENISA guideline ensures that key material used for encryption or authentication is secured within a hardware element that is more resilient to tampering (such as a Trusted Execution Environment or Secure Element). Both aim to protect the app against unauthorized access, modification, or malicious exploitation, thereby contributing to the overall resilience of the app against security threats. Binding keys to secure hardware adds a layer of security that supports the goal of MASVS-RESILIENCE-2.

- MASVS-RESILIENCE-4: The correlation between "MASVS-RESILIENCE-4," which focuses on making dynamic analysis and instrumentation difficult for attackers, and the ENISA Guideline on binding key material to secure hardware is that both aim to enhance the security of the mobile app. By binding keys to a secure environment like a Trusted Execution Environment (TEE) or Secure Element (SE), the likelihood of key material being exposed during runtime is minimized. This approach complements MASVS-RESILIENCE-4's objective of protecting the app from runtime manipulation and analysis, thereby making it harder for attackers to compromise app security or reverse engineer the app. By employing both strategies, an app can strengthen its defense against attacks that attempt to modify or analyze its behavior during execution.
- MASVS-STORAGE-1: Both the MASVS-STORAGE-1 guideline and the ENISA Guideline focus on the secure handling of sensitive data, albeit from different perspectives. The MASVS-STORAGE-1 guideline emphasizes that sensitive data, regardless of its source or storage location, needs to be properly protected. This means that whether the data is in private app storage or in public folders, it should be secured appropriately. The ENISA Guideline reinforces this perspective by specifically addressing the secure storage of cryptographic key material, recommending that keys be bound to secure hardware like Trusted Execution Environments (TEE) or Secure Elements (SE) wherever possible, which is a way to ensure that sensitive key material is not exposed outside of protected hardware. Both guidelines aim to improve the security of sensitive data by advocating for measures that prevent unauthorized access and exposure.

14.7 Implementation Guidance (ENISA 13.7):

ENISA Secure Smartphone Development Guidance (13.7): For keys whose key material is inside a secure hardware (e.g., TEE, SE), ensure that cryptographic and user authentication authorizations are also enforced by secure hardware, in platforms that this is optional (e.g., Android). Authentication control (e.g., using Biometric checks) and key decryption should be performed atomically in a TEE or on a chip with a secure channel to the TEE.

14.7.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation here is that both MASVS-AUTH-1 and the ENISA Guideline emphasize the need for secure authentication and authorization practices, especially when they are related to remote endpoints. MASVS-AUTH-1 is concerned with following best practices for secure protocol use, and although it does not explicitly mention secure hardware, it does imply the need for robust security mechanisms which could include hardware solutions. The ENISA Guideline specifies the use of secure hardware like TEE (Trusted Execution Environment) or SE (Secure Element) to enforce cryptographic operations and user authentication authorizations, strengthening the security of user authentication and authorization processes which is in line with the goal of MASVS-AUTH-1 for secure app connectivity and remote access control.
- MASVS-AUTH-2: The correlation is clear between ”MASVS-AUTH-2” which discusses the correct implementation of biometrics or a local PIN code for app authentication and the ENISA Guideline that emphasizes the importance of secure hardware enforcement for cryptographic and user authentication operations. Both focus on ensuring secure authentication mechanisms, where MASVS-AUTH-2 mentions the need for correct implementation, and the ENISA Guideline specifies that these operations, which could include biometric checks as mentioned in MASVS-AUTH-2, should be done within a secure hardware environment like TEE (Trusted Execution Environment) or SE (Secure Element). This establishes a direct connection between the secure implementation of user authentication mechanisms and the ENISA guidance to leverage hardware-level protection to achieve a robust security posture.
- MASVS-AUTH-3: The correlation is present as both the ”MASVS-AUTH-3” guideline and the ENISA guideline emphasize the need for secure implementation of additional authentication mechanisms. ”MASVS-AUTH-3” suggests the use of different methods like biometric, pin, MFA code generators, etc., for sensitive actions inside an app. The ENISA guideline similarly stresses that for secure key materials housed in secure hardware (e.g., TEE, SE), it’s important to ensure that authentication is enforced by the secure hardware when possible (such as on Android platforms). It also specifies that authentication checks, like biometric verifications, should ideally be performed simultaneously (or atomically) with key decryption in a TEE or on a chip that has a secure channel to the TEE. This aligns with the objective of ”MASVS-AUTH-3” to securely implement additional authentication measures for sensitive in-app actions.
- MASVS-CRYPTO-1: Both ”MASVS-CRYPTO-1” and the ENISA Guideline focus on ensuring the security of user data through the implementation of robust cryptographic practices, especially considering the potential physical access attackers may have to a mobile device. ”MASVS-CRYPTO-1” underlines the importance of adhering to general

cryptography best practices as defined in external standards, which would include the use of secure hardware like TEE (Trusted Execution Environment) or SE (Secure Element) for key material, as recommended by ENISA. The ENISA guideline further specifies that cryptographic operations and user authentication should be enforced by secure hardware when available, a recommendation that aligns with ”MASVS-CRYPTO-1”’s emphasis on leveraging industry-standard best practices to secure data. The atomicity of authentication and key decryption within a secure environment like TEE is an example of applying such best practices in a mobile context.

- MASVS-CRYPTO-2: The correlation exists because both ”MASVS-CRYPTO-2” and the ENISA Guideline emphasize the importance of secure key management in the context of cryptographic operations. The MASVS-CRYPTO-2 requirement highlights the necessity of managing cryptographic keys throughout their entire lifecycle to maintain the strength of cryptography. This includes secure key generation, storage, and protection mechanisms. Similarly, the ENISA Guideline specifies that for keys stored within secure hardware (like a Trusted Execution Environment or Secure Element), the enforcement of cryptographic operations and user authentication must also be ensured by the secure hardware, particularly in platforms where this might be optional (e.g., Android). It further recommends that authentication and key decryption be performed securely and atomically within a TEE or similar secure environment, which aligns with the principles of key management and protection as described in MASVS-CRYPTO-2. Both statements underscore the necessity of safeguarding cryptographic keys within a secure and controlled environment to prevent compromise.
- MASVS-PLATFORM-3: The correlation exists in the aspect of securing sensitive data from unintended exposure. MASVS-PLATFORM-3 emphasizes preventing unintentional leaks of sensitive data displayed on the UI through mechanisms such as screenshots or physical observation. The ENISA guideline focuses on ensuring that cryptographic operations and user authentication involving secure hardware like TEE or SE are robust against unauthorized access, which includes protection mechanisms for sensitive data such as keys. Both standards advocate for heightened security measures to safeguard sensitive information within an application’s context.
- MASVS-RESILIENCE-1: The MASVS-RESILIENCE-1 control focuses on ensuring that an app is running on a secure platform that hasn’t been compromised, as a tampered platform can undermine security features such as secure storage, biometrics, and sandboxing. The ENISA Guideline emphasizes the importance of key material being protected in secure hardware, like TEE (Trusted Execution Environment) or SE (Secure Element), and that authentication and cryptographic operations should be enforced by this secure hardware. Both the MASVS-RESILIENCE-1 control and the ENISA Guideline stress the necessity of relying on the underlying platform’s integrity to maintain the security of the app and its data. They correlate by emphasizing the trust in the platform’s security features to protect sensitive operations and authentication mechanisms.
- MASVS-RESILIENCE-2: The ”MASVS-RESILIENCE-2” requirement and the ENISA guideline both emphasize the importance of protecting the integrity and security of the app and its operating environment on the user device. MASVS-RESILIENCE-2 focuses on preventing code and resource modifications to maintain app integrity while the ENISA guideline ensures that cryptographic operations and user authentication are securely enforced, potentially within a Trusted Execution Environment (TEE) or Secure Element (SE), to safeguard against threats like local modifications or backdoored versions of apps. Both stress the need for hardware-level security measures to defend against unauthorized

changes and secure sensitive operations, which align well with each other, addressing similar security concerns.

- MASVS-RESILIENCE-3: The correlation between ”MASVS-RESILIENCE-3” and the mentioned ENISA Guideline exists in their shared goal of enhancing the security of an application by preventing unauthorized access or tampering. MASVS-RESILIENCE-3 focuses on obfuscating the application internals to make static analysis difficult, while the ENISA Guideline recommends the use of secure hardware like TEE (Trusted Execution Environment) or SE (Secure Element) to enforce cryptographic and user authentication. Both controls aim to protect against understanding or modifying the app’s operation, preserving its integrity and ensuring that authentication and cryptographic processes are not bypassed or tampered with.
- MASVS-RESILIENCE-4: The correlation between ”MASVS-RESILIENCE-4” and the ENISA guideline exists in the context of enhancing the security of an app against dynamic analysis and runtime manipulation. ”MASVS-RESILIENCE-4” seeks to make it difficult for attackers to perform dynamic analysis or modify code at runtime, which aligns with the ENISA guideline’s emphasis on enforcing cryptographic and user authentication authorizations with secure hardware such as TEE (Trusted Execution Environment) or SE (Secure Element). By requiring that authentication and key decryption be performed atomically in hardware that is resilient to tampering, like TEE, the guideline supports the principle of making dynamic analysis and any subsequent manipulation more challenging. Both aim to increase the security posture by limiting the ability to interact with secure operations at runtime, albeit from different angles: ”MASVS-RESILIENCE-4” frames it broadly in terms of dynamic analysis, while ENISA focuses specifically on the secure handling of cryptographic keys and user authentication.
- MASVS-STORAGE-1: The correlation between ”MASVS-STORAGE-1” and the mentioned ENISA Guideline is that both emphasize the importance of secure handling and storage of sensitive data. MASVS-STORAGE-1 focuses on ensuring that sensitive data stored by apps is properly protected regardless of its location. The ENISA Guideline outlines that for keys protected by secure hardware such as a TEE (Trusted Execution Environment) or SE (Secure Element), there should be enforcement of cryptographic and user authentication controls by the secure hardware itself. Both guidelines are concerned with robust security measures for sensitive data and cryptographic keys, though MASVS-STORAGE-1 is broader, covering all sensitive data storage, while the ENISA Guideline is more specific to keys within secure hardware.
- MASVS-STORAGE-2: The MASVS-STORAGE-2 control is concerned with preventing unintentional leaks of sensitive data, which may include cryptographic keys and other security credentials that should be protected. The ENISA guideline highlights the importance of handling keys and cryptographic operations within secure hardware environments like TEE or SE, ensuring that user authentication and key management are tightly controlled. Both the control and the guideline aim to prevent sensitive data exposure through improved security practices around data storage and handling, especially for cryptographic keys and user authentication data. Hence, they both correlate in their intention to secure sensitive data against unintended access or exposure.

14.8 Implementation Guidance (ENISA 13.8):

ENISA Secure Smartphone Development Guidance (13.8): The application should avoid using temporal validity interval authorizations, since they are unlikely to be enforced by the secure hardware because it normally does not have an independent secure real-time clock.

14.8.1 OWASP MASVS MAPPING

- MASVS-AUTH-3: The correlation between ”MASVS-AUTH-3”, which discusses the implementation of additional forms of authentication (like biometric, pin, MFA code generator, email, deep links, etc.) for sensitive actions, and the ENISA guideline, which advises against using temporal validity interval authorizations, is that both are concerned with securing authentication mechanisms in the app. While MASVS-AUTH-3 focuses on the diverse methods that can be used to add extra layers of security, the ENISA guideline specifically cautions against relying on mechanisms that use temporal checks (like tokens valid for a certain time), since the secure hardware may not support such functionality reliably. Both stress the importance of considering the security implications of the authentication method selected.
- MASVS-CODE-2: The correlation between ”MASVS-CODE-2” and the ENISA guideline revolves around security controls and updates in response to vulnerabilities. MASVS-CODE-2 emphasizes the need for a mechanism to force app updates when critical vulnerabilities are found, ensuring users run the most secure version. The ENISA guideline warns against relying on temporal validity for authorizations due to hardware limitations in securely tracking real-time. Both stress the importance of active management to maintain security, where MASVS-CODE-2’s update enforcement complements ENISA’s stance on avoiding insecure time-based authorizations.
- MASVS-CRYPTO-2: The correlation between ”MASVS-CRYPTO-2” and the ENISA guideline mentioned is that both relate to the management and security aspects of cryptographic keys. MASVS-CRYPTO-2 emphasizes the entire lifecycle management of cryptographic keys, insisting on secure generation, storage, and protection. Meanwhile, the ENISA guideline points out a specific issue with key management, which is the avoidance of using temporal validity for authorizations, primarily because secure hardware often can’t enforce them due to the lack of an independent secure real-time clock. Lack of time-based enforcement could lead to poor key management by allowing keys to be valid outside their intended time frame, thus compromising security. Both MASVS-CRYPTO-2 and the ENISA guideline address the broader theme of ensuring robust cryptographic practices by managing keys effectively to prevent their compromise.
- MASVS-RESILIENCE-1: The correlation between ”MASVS-RESILIENCE-1” and the ENISA guideline about avoiding the use of temporal validity interval authorizations is that both are concerned with the integrity and security of the platform on which an application runs. MASVS-RESILIENCE-1 emphasizes the importance of ensuring that the Operating System (OS) has not been tampered with, as this would compromise security features that are critical to the app’s data protection. Trust in the platform’s integrity is crucial. Similarly, the ENISA guideline implies a concern for the trustworthiness of the underlying platform, specifically secure hardware, which may not enforce temporal restrictions due to the lack of

an independent secure real-time clock. Both are focused on the reliability of the platform's security features to protect the app's operation and data.

14.9 Implementation Guidance (ENISA 13.9):

ENISA Secure Smartphone Development Guidance (13.9): Verify that the application's authentication policy complies with the possibility that different people may enroll for biometric authentication in the same device. As a result, successful biometric authentication may be possible for different device users

14.9.1 OWASP MASVS MAPPING

- MASVS-AUTH-1: The correlation between "MASVS-AUTH-1" and the ENISA Guideline lies in the realm of authentication best practices and the recognition of the device's context. "MASVS-AUTH-1" emphasizes the importance of secure authentication protocols in apps, which involves considering different aspects of user authentication and authorization. The ENISA Guideline clearly states a specific scenario where multiple people could enroll for biometric authentication on the same device, indicating a need for the app to handle such situations securely. Both guidelines ultimately aim to ensure that the app securely manages user identity and access controls, even when faced with device-sharing scenarios. Hence, there is a clear correlation in their intent to secure user authentication mechanisms in applications.
- MASVS-AUTH-2: "MASVS-AUTH-2" mentions the need for correct implementation of biometric and local PIN code authentication mechanisms in apps. The ENISA guideline emphasizes the need to account for the fact that different people might register their biometric data on the same device, which aligns with the implementation aspect of MASVS-AUTH-2 as it implies enforcing an authentication policy that takes into consideration multiple users' biometric authentication on a single device. The correlation lies in the requirement for robust and secure authentication practices concerning the use of biometrics.
- MASVS-AUTH-3: The correlation exists in that both the MASVS-AUTH-3 and the ENISA Guideline are concerned with the secure implementation of additional authentication factors, particularly in relation to biometric authentication. MASVS-AUTH-3 suggests implementing additional forms of authentication such as biometric, and the ENISA Guideline specifies that the application's authentication policy must account for the possibility that multiple users could enroll for biometric authentication on the same device. Both guidelines emphasize the need for security measures when adding layers of authentication to protect sensitive in-app actions.
- MASVS-RESILIENCE-1: The correlation between "MASVS-RESILIENCE-1" and the ENISA Guideline is that both are concerned with the security of the platform on which the application is running and how it affects authentication mechanisms. "MASVS-RESILIENCE-1" is about validating that the operating system (OS) has not been compromised, which is important because a tampered OS can disable security features critical for authenticating users, such as secure storage and biometric sensors used for biometric authentication. The ENISA Guideline focuses on ensuring that the app's authentication policy takes into account the fact that multiple people may enroll for biometric authentication on the same device. Both emphasize the need for the app to operate securely in the context of device-level authenticity – "MASVS-RESILIENCE-1" by validating the trustworthiness of the platform itself, and the ENISA Guideline by ensuring authentication policies accommodate device sharing without compromising security.

Part III
REFERENCES

Part III of the book, titled ”References,” offers readers further reading materials and resources to expand their knowledge beyond the book’s content. This section is designed to guide readers towards additional information, studies, and guidelines that will assist them in staying abreast of the latest developments and best practices in the field of mobile app security. It’s an invaluable resource for those who wish to delve deeper into the subject and explore the broader landscape of mobile application security.

Chapter 15
OWASP MASVS - Mobile Application Security Verification Standard

The OWASP Mobile Application Security Verification Standard (MASVS)[1] is the industry standard for mobile application security. It provides a comprehensive set of security controls that can be used to assess the security of mobile apps across various platforms (e.g., Android, iOS) and deployment scenarios (e.g., con- sumer, enterprise). The standard covers the key components of the mobile app attack surface including storage, cryptography, authentication and authorization, network communication, interaction with the mobile platform, code quality and resilience against reverse engineering and tampering.

[1] OWASP Foundation, Mobile Application Security Verification Standard, Version 2.0, [Online]. Available: https://mas.owasp.org/MASVS/.

15.1 MASVS-STORAGE: Storage

Mobile applications handle a wide variety of sensitive data, such as personally identifiable information (PII), cryptographic material, secrets, and API keys, that often need to be stored locally. This sensitive data may be stored in private locations, such as the app's internal storage, or in public folders that are accessible by the user or other apps installed on the device. However, sensitive data can also be unintentionally stored or exposed to publicly accessible locations, typically as a side-effect of using certain APIs or system capabilities such as backups or logs. This category is designed to help developers ensure that any sensitive data intentionally stored by the app is properly protected, regardless of the target location. It also covers unintentional leaks that can occur due to improper use of APIs or system capabilities.

15.1.1 MASVS-STORAGE-1: The app securely stores sensitive data

Control

The app securely stores sensitive data.

Description

Apps handle sensitive data coming from many sources such as the user, the backend, system services or other apps on the device and usually need to store it locally. The storage locations may be private to the app (e.g. its internal storage) or be public and therefore accessible by the user or other installed apps (e.g. public folders such as Downloads). This control ensures that any sensitive data that is intentionally stored by the app is properly protected independently of the target location.

15.1.2 MASVS-STORAGE-2: The app prevents leakage of sensitive data

Control

The app prevents leakage of sensitive data.

Description

There are cases when sensitive data is unintentionally stored or exposed to publicly accessible locations; typically as a side-effect of using certain APIs, system capabilities such as backups or logs. This control covers this kind of unintentional leaks where the developer actually has a way to prevent it.

15.2 MASVS-CRYPTO: Cryptography

Cryptography is essential for mobile apps because mobile devices are highly portable and can be easily lost or stolen. This means that an attacker who gains physical access to a device can potentially access all the sensitive data stored on it, including passwords, financial information, and personally identifiable information. Cryptography provides a means of protecting this sensitive data by encrypting it so that it cannot be easily read or accessed by an unauthorized user. The purpose of the controls in this category is to ensure that the verified app uses cryptography according to industry best practices, which are typically defined in external standards such as and . This category also focuses on the management of cryptographic keys throughout their lifecycle, including key generation, storage, and protection. Poor key management can compromise even the strongest cryptography, so it is crucial for developers to follow the recommended best practices to ensure the security of their users' sensitive data.

15.2.1 MASVS-CRYPTO-1: The app employs current strong cryptography and uses it according to industry best practices

Control

The app employs current strong cryptography and uses it according to industry best practices.

Description

Cryptography plays an especially important role in securing the user's data - even more so in a mobile environment, where attackers having physical access to the user's device is a likely scenario. This control covers general cryptography best practices, which are typically defined in external standards.

15.2.2 MASVS-CRYPTO-2: The app performs key management according to industry best practices

Control

The app performs key management according to industry best practices.

Description

Even the strongest cryptography would be compromised by poor key management. This control covers the management of cryptographic keys throughout their lifecycle, including key generation, storage and protection.

MASVS-AUTH: Authentication and Authorization

Authentication and authorization are essential components of most mobile apps, especially those that con- nect to a remote service. These mechanisms provide an added layer of security and help prevent unautho- rized access to sensitive user data. Although the enforcement of these mechanisms must be on the remote endpoint, it is equally important for the app to follow relevant best practices to ensure the secure use of the involved protocols. Mobile apps often use different forms of authentication, such as biometrics, PIN, or multi-factor authenti- cation code generators, to validate user identity. These mechanisms must be implemented correctly to ensure their effectiveness in preventing unauthorized access. Additionally, some apps may rely solely on local app authentication and may not have a remote endpoint. In such cases, it is critical to ensure that local authentication mechanisms are secure and implemented following industry best practices. The controls in this category aim to ensure that the app implements authentication and authorization mech- anisms securely, protecting sensitive user information and preventing unauthorized access. It is important to note that the security of the remote endpoint should also be validated using industry standards such as the .

15.2.3 MASVS-AUTH-1: The app uses secure authentication and authorization protocols and follows the relevant best practices

Control

The app uses secure authentication and authorization protocols and follows the relevant best practices.

Description

Most apps connecting to a remote endpoint require user authentication and also enforce some kind of autho- rization. While the enforcement of these mechanisms must be on the remote endpoint, the apps also have to ensure that it follows all the relevant best practices to ensure a secure use of the involved protocols.

15.2.4 MASVS-AUTH-2: The app performs local authentication securely according to the platform best practices

Control

The app performs local authentication securely according to the platform best practices.

Description

Many apps allow users to authenticate via biometrics or a local PIN code. These authentication mechanisms need to be correctly implemented. Additionally, some apps might not have a remote endpoint, and rely fully on local app authentication.

15.2.5 MASVS-AUTH-2: The app secures sensitive operations with additional authentication

Control

The app secures sensitive operations with additional authentication.

Description

Some additional form of authentication is often desirable for sensitive actions inside the app. This can be done in different ways (biometric, pin, MFA code generator, email, deep links, etc) and they all need to be implemented securely.

15.3 MASVS-NETWORK: Network Communication

Secure networking is a critical aspect of mobile app security, particularly for apps that communicate over the network. In order to ensure the confidentiality and integrity of data in transit, developers typically rely on encryption and authentication of the remote endpoint, such as through the use of TLS. However, there are numerous ways in which a developer may accidentally disable the platform secure defaults or bypass them entirely by utilizing low-level APIs or third-party libraries. This category is designed to ensure that the mobile app sets up secure connections under any circumstances. Specifically, it focuses on verifying that the app establishes a secure, encrypted channel for network communication. Additionally, this category covers situations where a developer may choose to trust only specific Certificate Authorities (CAs), which is commonly referred to as certificate pinning or public key pinning.

15.3.1 MASVS-NETWORK-1: The app secures all network traffic according to the current best practices

Control

The app secures all network traffic according to the current best practices.

Description

Ensuring data privacy and integrity of any data in transit is critical for any app that communicates over the network. This is typically done by encrypting data and authenticating the remote endpoint, as TLS does. However, there are many ways for a developer to disable the platform secure defaults, or bypass them completely by using low-level APIs or third-party libraries. This control ensures that the app is in fact setting up secure connections in any situation.

15.3.2 MASVS-NETWORK-2: The app performs identity pinning for all remote endpoints under the developer's control

Control

The app performs identity pinning for all remote endpoints under the developer's control.

Description

Instead of trusting all the default root CAs of the framework or device, this control will make sure that only very specific CAs are trusted. This practice is typically called certificate pinning or public key pinning.

15.4 MASVS-PLATFORM: Platform Interaction

The security of mobile apps heavily depends on their interaction with the mobile platform, which often involves exposing data or functionality intentionally through the use of platform-provided inter-process com- munication (IPC) mechanisms and WebViews to enhance the user experience. However, these mechanisms can also be exploited by attackers or other installed apps, potentially compromising the app's security. Furthermore, sensitive data, such as passwords, credit card details, and one-time passwords in notifications, is often displayed in the app's user interface. It is essential to ensure that this data is not unintentionally leaked through platform mechanisms such as auto-generated screenshots or accidental disclosure through shoulder surfing or device sharing. This category comprises controls that ensure the app's interactions with the mobile platform occur securely. These controls cover the secure use of platform-provided IPC mechanisms, WebView configurations to prevent sensitive data leakage and functionality exposure, and secure display of sensitive data in the app's user interface. By implementing these controls, mobile app developers can safeguard sensitive user information and prevent unauthorized access by attackers.

15.4.1 MASVS-PLATFORM-1: The app uses IPC mechanisms securely

Control

The app uses IPC mechanisms securely.

Description

Apps typically use platform provided IPC mechanisms to intentionally expose data or functionality. Both installed apps and the user are able to interact with the app in many different ways. This control ensures that all interactions involving IPC mechanisms happen securely.

15.4.2 MASVS-PLATFORM-2: The app uses WebViews securely

Control

The app uses WebViews securely.

Description

WebViews are typically used by apps that have a need for increased control over the UI. This control ensures that WebViews are configured securely to prevent sensitive data leakage as well as sensitive functionality exposure (e.g. via JavaScript bridges to native code).

15.4.3 MASVS-PLATFORM-3: The app uses the user interface securely

Control

The app uses the user interface securely.

Description

Sensitive data has to be displayed in the UI in many situations (e.g. passwords, credit card details, OTP codes in notifications). This control ensures that this data doesn't end up being unintentionally leaked due to platform mechanisms such as auto-generated screenshots or accidentally disclosed via e.g. shoulder surfing or sharing the device with another person.

15.5 MASVS-CODE: Code Quality

Mobile apps have many data entry points, including the UI, IPC, network, and file system, which might receive data that has been inadvertently modified by untrusted actors. By treating this data as untrusted input and properly verifying and sanitizing it before use, developers can prevent classical injection attacks, such as SQL injection, XSS, or insecure deserialization. However, other common coding vulnerabilities, such as memory corruption flaws, are hard to detect in penetration testing but easy to prevent with secure architecture and coding practices. Developers should follow best practices such as the and to avoid introducing these flaws in the first place. This category covers coding vulnerabilities that arise from external sources such as app data entry points, the OS, and third-party software components. Developers should verify

and sanitize all incoming data to prevent injection attacks and bypass of security checks. They should also enforce app updates and ensure that the app runs up-to-date platforms to protect users from known vulnerabilities.

15.5.1 MASVS-CODE-1: The app requires an up-to-date platform version

Control

The app requires an up-to-date platform version.

Description

Every release of the mobile OS includes security patches and new security features. By supporting older versions, apps stay vulnerable to well-known threats. This control ensures that the app is running on an up-to-date platform version so that users have the latest security protections.

15.5.2 MASVS-CODE-2: The app has a mechanism for enforcing app updates

Control

The app has a mechanism for enforcing app updates.

Description

Sometimes critical vulnerabilities are discovered in the app when it is already in production. This control ensures that there is a mechanism to force the users to update the app before they can continue using it.

15.5.3 MASVS-CODE-3: The app only uses software components without known vulnerabilities

Control

The app only uses software components without known vulnerabilities.

Description

To be truly secure, a full whitebox assessment should have been performed on all app components. However, as it usually happens with e.g. for third-party components this is not always

feasible and not typically part of a penetration test. This control covers "low-hanging fruit" cases, such as those that can be detected just by scanning libraries for known vulnerabilities.

15.5.4 MASVS-CODE-4: The app validates and sanitizes all untrusted inputs

Control

The app validates and sanitizes all untrusted inputs.

Description

Apps have many data entry points including the UI, IPC, the network, the file system, etc. This incoming data might have been inadvertently modified by untrusted actors and may lead to bypass of critical security checks as well as classical injection attacks such as SQL injection, XSS or insecure deserialization. This control ensures that this data is treated as untrusted input and is properly verified and sanitized before it's used.

15.6 MASVS-RESILIENCE: Resilience Against Reverse Engineering and Tampering

Defense-in-depth measures such as code obfuscation, anti-debugging, anti-tampering, etc. are important to increase app resilience against reverse engineering and specific client-side attacks. They add multiple layers of security controls to the app, making it more difficult for attackers to successfully reverse engineer and extract valuable intellectual property or sensitive data from it, which could result in:

- The theft or compromise of valuable business assets such as proprietary algorithms, trade secrets, or customer data
- Significant financial losses due to loss of revenue or legal action
- Legal and reputational damage due to breach of contracts or regulations
- Damage to brand reputation due to negative publicity or customer dissatisfaction

The controls in this category aim to ensure that the app is running on a trusted platform, prevent tampering at runtime and ensure the integrity of the app's intended functionality. Additionally, the controls impede comprehension by making it difficult to figure out how the app works using static analysis and prevent dynamic analysis and instrumentation that could allow an attacker to modify the code at runtime. However, note that the lack of any of these measures does not necessarily cause vulnerabilities - instead, they add threat-specific additional protection to apps which must also fulfil the rest of the OWASP MASVS security controls according to their specific threat models.

15.6.1 MASVS-RESILIENCE-1: The app validates the integrity of the platform

Control

The app validates the integrity of the platform.

Description

Running on a platform that has been tampered with can be very dangerous for apps, as this may disable certain security features, putting the data of the app at risk. Trusting the platform is essential for many of the MASVS controls relying on the platform being secure (e.g. secure storage, biometrics, sandboxing, etc.). This control tries to validate that the OS has not been compromised and its security features can thus be trusted.

15.6.2 MASVS-RESILIENCE-2: The app implements anti-tampering mechanisms

Control

The app implements anti-tampering mechanisms.

Description

Apps run on a user-controlled device, and without proper protections it's relatively easy to run a modified version locally (e.g. to cheat in a game, or enable premium features without paying), or upload a back- doored version of it to third-party app stores. This control tries to ensure the integrity of the app's intended functionality by preventing modifications to the original code and resources.

15.6.3 MASVS-RESILIENCE-3: The app implements anti-static analysis mechanisms

Control

The app implements anti-static analysis mechanisms.

Description

Understanding the internals of an app is typically the first step towards tampering with it (either dynamically, or statically). This control tries to impede comprehension by making it as difficult as possible to figure out how an app works using static analysis.

15.6.4 MASVS-RESILIENCE-4: The app implements anti-dynamic analysis techniques

Control

The app implements anti-dynamic analysis techniques.

Description

Sometimes pure static analysis is very difficult and time consuming so it typically goes hand in hand with dynamic analysis. Observing and manipulating an app during runtime makes it much easier to decipher its behavior. This control aims to make it as difficult as possible to perform dynamic analysis, as well as prevent dynamic instrumentation which could allow an attacker to modify the code at runtime.

15.7 Correlation between OWASP MASVS 2.0 and ENISA SMARTPHONE SECURE DEVELOPMENT GUIDELINE

Please follow this Link to the Ressources Pages of this Book:
https://github.com/groonvandorp/publishings/wiki/SECURE-SMARTPHONE-DEVELOPMENT

Fig. 15.1 QR-Code to access Book Ressources

Chapter 16
NIST SP 800-163 Rev. 1 - Vetting the Security of Mobile Apps

16.1 Introduction

The publication[1] emphasizes the critical role mobile applications play in organizational and personal contexts, highlighting the need for stringent security measures to mitigate risks associated with vulnerabilities and defects. It defines a mobile application vetting process aimed at ensuring apps conform to an organization's security requirements.

16.2 App Security Requirements

Security requirements are delineated into general requirements, drawing from standards and practices by NIAP, OWASP, MITRE, and NIST, and organization-specific requirements, tailored to the entity's policies, regulations, and risk tolerance.

16.3 App Vetting Process

The vetting process is described as comprising four main sub-processes: app intake, app testing, app approval/rejection, and results submission. It details the steps involved in evaluating an app's conformity to security requirements and the decision-making process regarding its deployment within an organization.

16.4 App Testing and Vulnerability Classifiers

This section outlines the methodologies employed in app testing, including correctness testing, source and binary code testing, and static and dynamic testing. It also introduces vulnerability classifiers and quantifiers such as CWE, CVE, and CVSS, providing a framework for identifying and assessing security vulnerabilities.

---

[1] National Institute of Standards and Technology, "Vetting the Security of Mobile Applications," NIST Special Publication 800-163 Rev. 1, 2023. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-163r1.

16.5 App Vetting Considerations

Considerations include managing and unmanaging apps, the implications of app whitelisting and blacklisting, limitations of app vetting, the use of local and remote tools and services, automated approval/rejection processes, the concept of reciprocity, tool report analysis, and the distinction between compliance and certification. The section also underscores the importance of budgeting and staffing for effective app vetting operations.

16.6 App Vetting Systems

Lastly, the publication emphasizes the need for a systematic approach to app vetting, advocating for the integration of app vetting systems within organizational security frameworks to streamline the vetting process, enhance security postures, and facilitate the management of mobile applications in alignment with an organization's security and operational goals.

16.7 Appendix C—iOS App Vulnerability Types

This appendix identifies and defines the various types of vulnerabilities that are specific to apps running on mobile devices utilizing the Apple iOS operating system. The scope does not include vulnerabilities in the mobile platform hardware and communications networks. Although some of the vulnerabilities described below are common across mobile device environments, this appendix focuses on iOS-specific vulnerabilities. The vulnerabilities in this appendix are broken into three hierarchical levels, A, B, and C. The A level is referred to as the vulnerability class and is the broadest description for the vulnerabilities specified under that level. The B level is referred to as the sub-class and attempts to narrow down the scope of the vulnerability class into a smaller, common group of vulnerabilities. The C level specifies the individual vulnerabilities that have been identified. The purpose of this hierarchy is to guide the reader to finding the type of vulnerability they are looking for as quickly as possible.

16.8 General Categories of iOS App Vulnerabilities

Incorrect Permissions

Definition: Permissions allow accessing controlled functionality such as the camera or GPS and are requested in the program. Permissions can be implicitly granted to an app without the user's consent.

Negative Consequences: An app with too many permissions may perform unintended functions outside the scope of the app's intended functionality. Additionally, the permissions are vulnerable to hijacking by another app. If too few permissions are granted, the app will not be able to perform the functions required.

Exposed Communication - Internal and External

Definition: Internal communications protocols allow apps to process information and communicate with other apps. External communications allow information to leave the device.

Negative Consequences: Exposed internal communications allow apps to gather unintended information and inject new information. Exposed external communication (data network, WiFi, Bluetooth, etc.) leave information open to disclosure or man-in-the-middle attacks.

Potentially Dangerous Functionality

Definition: Controlled functionality that accesses system-critical resources or the user's personal information. This functionality can be invoked through API calls or hard coded into an app.

Negative Consequences: Unintended functions could be performed outside the scope of the app's functionality.

App Collusion

Definition: Two or more apps passing information to each other in order to increase the capabilities of one or both apps beyond their declared scope.

Negative Consequences: Collusion can allow apps to obtain data that was unintended such as a gaming app obtaining access to the user's contact list.

Obfuscation

Definition: Functionality or control flow that is hidden or obscured from the user. For the purposes of this appendix, obfuscation was defined as three criteria: external library calls, reflection, and packed code.
Negative Consequences:

1. External libraries can contain unexpected and/or malicious functionality.
2. Reflective calls can obscure the control flow of an app and/or subvert permissions within an app.
3. Packed code prevents code reverse engineering and can be used to hide malware.

Excessive Power Consumption

Definition: Excessive functions or unintended apps running on a device which intentionally or unintentionally drain the battery.

Negative Consequences: Shortened battery life could affect the ability to perform mission-critical functions.

Traditional Software Vulnerabilities

Definition: All vulnerabilities associated with Objective C and others. This includes: Authentication and Access Control, Buffer Handling, Control Flow Management, Encryption and Randomness, Error Handling, File Handling, Information Leaks, Initialization and Shutdown, Injection, Malicious Logic, Number Handling and Pointer and Reference Handling.

Negative Consequences: Common consequences include unexpected outputs, resource exhaustion, denial of service, etc.

Exposed Data Storage

Definition: All files and keychain items on iOS are assigned Data Protection classes. These dictate whether the item is 1) accessible while the device is locked, 2) accessible when the associated app is closed, and 3) if the item can be transferred to another device.

Negative Consequences: Sensitive data can be less protected on the file system while not being used, or unintentionally transferred to another system in a backup. However, restricting the use of this mechanism may impair an app's ability to perform desired functionality

16.9 iOS App Vulnerabilities by Level

Level A	Level B	Level C
Incorrect Permissions	Sensitive Information	Contacts
		Calendar Information
		Tasks
		Reminders
		Photos
		Bluetooth Access
Exposed Communications	External Communications	Telephony
		Bluetooth
		GPS
		SMS/MMS
		Network/Data Communications
	Internal Communications	Abusing Protocol Handlers
Potentially Dangerous Functionality	Direct Memory Mapping	Memory Access
		File System Access
	Potentially Dangerous API	Cost Sensitive APIs
		Device Management APIs
		Personal Information APIs
App Collusion	Data Change	Changes to Shared File Resources
		Changes to Shared Database Resources
		Changes to Shared Content Providers
	Data Creation / Deletion	Creation / Deletion to shared File Ressources
Obfuscation	Number of Services	Excessive Checks for Service State
	Native Code	Potentially Malicious Libraries Packaged but not Used
		Use of Potentially Dangerous Libraries
		Reflection Identification
		Class Introspection
	Library Calls	Constructor Introspection
		Field Introspection
		Method Introspection
	Packed Code	(-)
Excessive Power Consumption	CPU Usage	(-)
	I/O	(-)
Exposed Data Storage	Over Exposing Data	Over Granting File Data Protection Class
		Over Granting Keychain Data Protection Class

16.10 Correlation between NIST SP 800-163 Rev. 1 - VETTING THE SECURITY OF MOBILE APPS and ENISA SMARTPHONE SECURE DEVELOPMENT GUIDELINE

Please follow this Link to the Ressources Pages of this Book:
https://github.com/groonvandorp/publishings/wiki/SECURE-SMARTPHONE-DEVELOPMENT

Fig. 16.1 QR-Code to access Book Ressources

Chapter 17
Further Reading

Contemporary Trends in Mobile App Security

As mobile app development continues to evolve, staying updated with the latest security trends and best practices is crucial. Below are key resources and studies from 2024 that provide insights into current and emerging challenges in mobile app security:

- **Geo Compliance and Cyber Resilience**: Learn about the importance of geo-compliance and why cyber resilience is critical beyond mere compliance. Refer to *Appdome's 2024 Trends in Mobile App Security* for an in-depth analysis.
- **Generative AI in Cyber Threats**: Explore how generative AI is transforming the threat landscape. Detailed insights in *iTWire's Five Trends and Predictions in Mobile App Security for 2024*.
- **DevSecOps 2.0**: Understand the evolving role of cyber teams in mobile app security. Discussed in *iTWire's article on DevSecOps 2.0*.
- **Consumer Expectations on Mobile App Security**: Insights into consumer awareness and demand for security in mobile apps. See *Appdome's 2023 Global Consumer Expectation of Mobile Security Applications Survey*.

These resources provide a comprehensive overview of the dynamic field of mobile app security, offering valuable guidance for developers, security professionals, and stakeholders. **Note**: The above references are based on the latest research and discussions as of 2024. Continual research in this rapidly evolving field is recommended.

Tooling for Mobile App Security in 2024

In 2024, various tools have emerged as critical in ensuring the security of mobile applications. Below is a list of some of the top tools in this space:

1. **Burp Suite**: A frontrunner in web application security, offering both automated and manual testing options for mobile applications.
   `https://portswigger.net/burp`
2. **OWASP ZAP**: An open-source tool by OWASP for automated and manual penetration testing, known for its community-driven approach.
   `https://www.zaproxy.org/`

3. **Nessus**: Renowned for its robust vulnerability scanning capabilities, particularly effective in identifying vulnerabilities in mobile applications.
   https://www.tenable.com/products/nessus
4. **Metasploit**: A powerful tool and framework for cybersecurity professionals, invaluable for mobile app security testing.
   https://www.metasploit.com/
5. **Wireshark**: A network protocol analyzer, essential in identifying potential security flaws in mobile app network communications.
   https://www.wireshark.org/
6. **Acunetix**: Leads in automated web application security software, offering fast and accurate scanning for mobile apps.
   https://www.acunetix.com/
7. **Appium**: An open-source platform that supports automated testing of native, hybrid, and web applications.
   https://appium.io/
8. **MobSF (Mobile Security Framework)**: Performs static and dynamic analysis on Android, iOS, and Windows apps.
   https://mobsf.github.io/Mobile-Security-Framework-MobSF/
9. **Qualys**: A cloud-based platform known for its integrated suite of security and compliance solutions, effective in mobile app scanning.
   https://www.qualys.com/

These tools represent the forefront of mobile app security in 2024, providing comprehensive solutions for developers and security professional.