MOBILE PENTERATION

Vulnerability Analyst / Penetration Tester

Mobile Penteration Team

Mobile
Penteration
Team

AGENDA

01 INTRODUCTION & OBJECTIVE

02 METHODOLOGY

03 TESTING PHASES

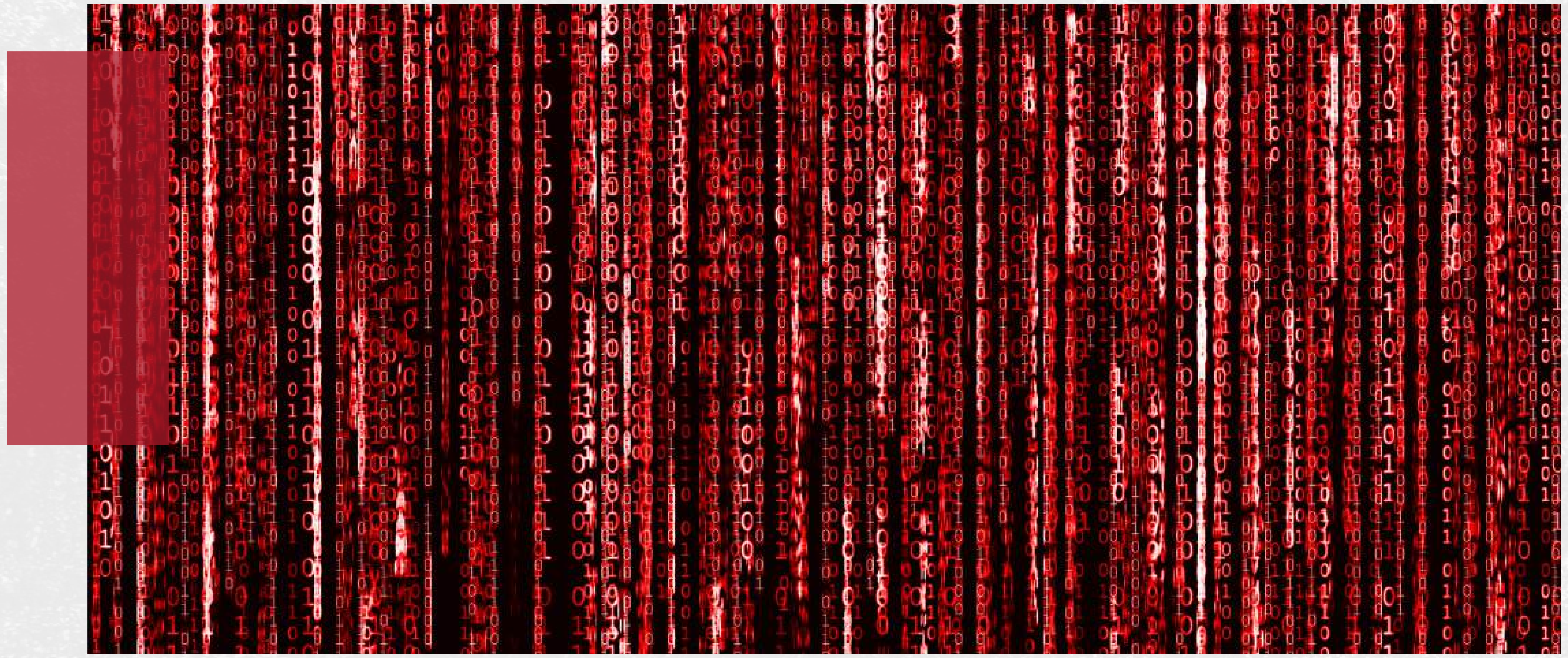
04 WHY WE SKIPPED RECON PHASE

05 KEY FINDINGS

06 RECOMMENDATIONS

07 CONCLUSIONS

08 OUR TEAM

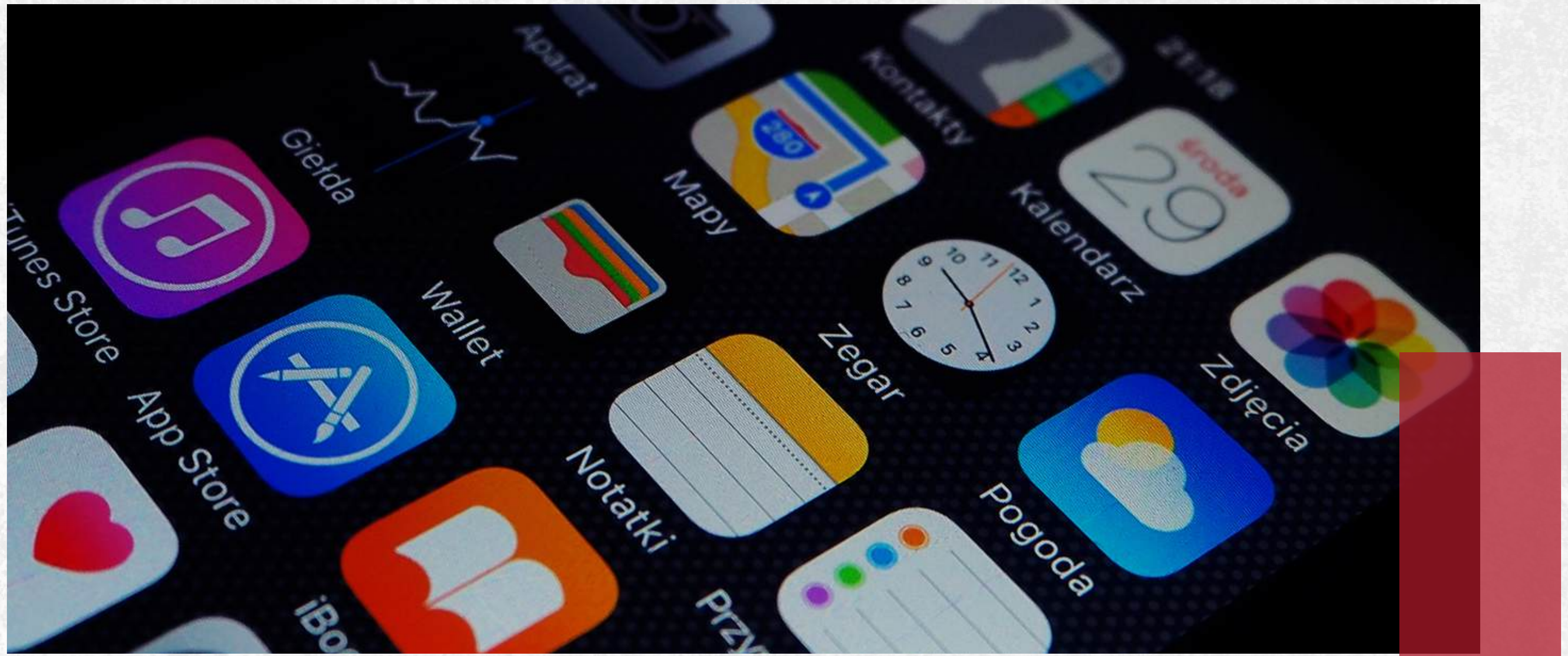


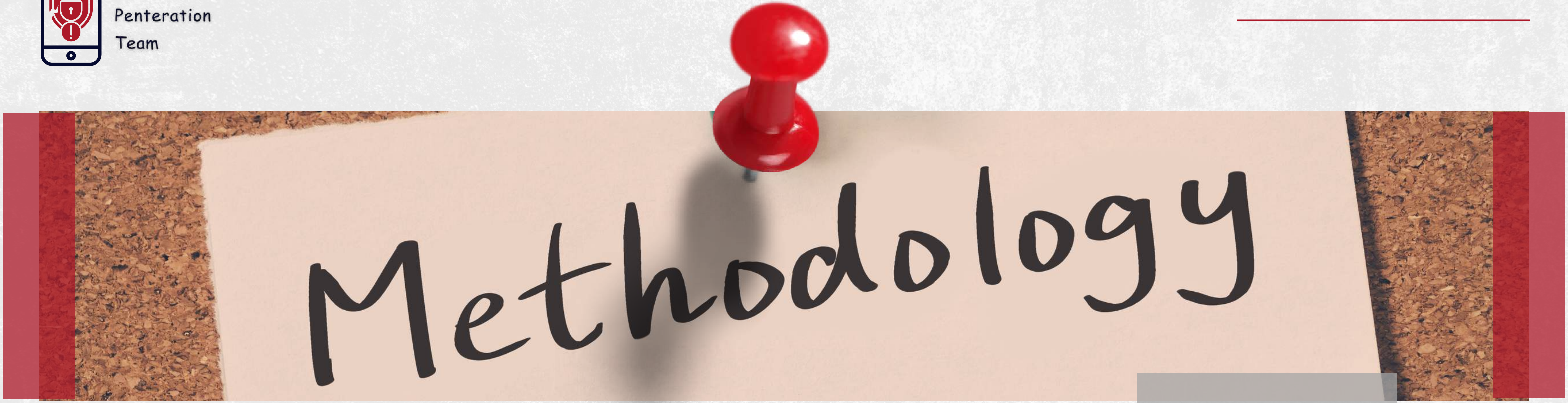
01. INTRODUCTION

As mobile app usage grows, security is crucial. This penetration test follows OWASP Top 10 and MASVS to identify vulnerabilities in the app and backend, providing recommendations to enhance security

02. OBJECTIVE

Perform a thorough penetration test on the mobile app and its backend to identify vulnerabilities, assess resilience against attacks, ensure compliance with OWASP Top 10 and MASVS, and provide recommendations to improve security and protect sensitive data.



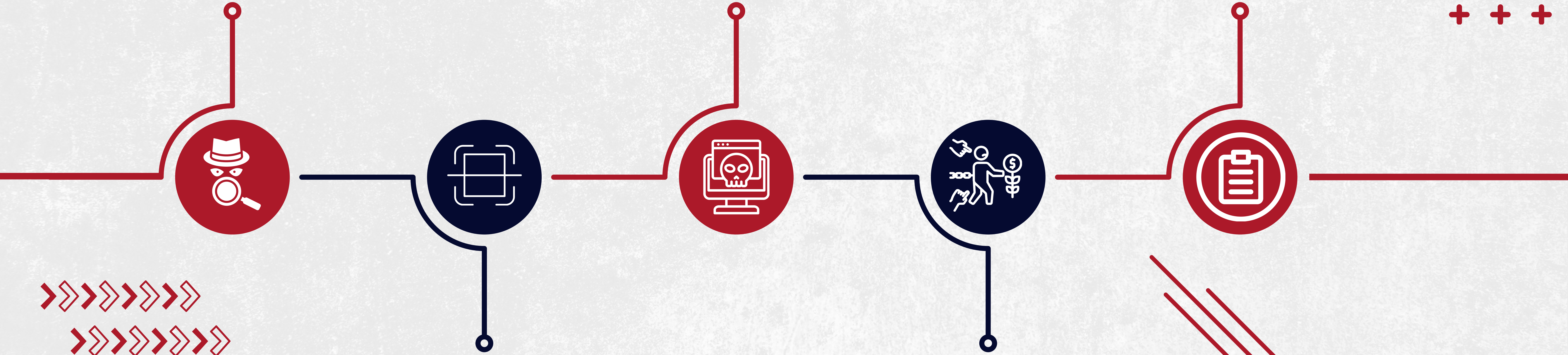


- the Testing followed industry-standard methodologies, inculding **OWASP** Top 10 and other commonly known attack vectors, to ensure a thorough analysis of the potential risks.

- Target: Smart Shipping Application
- Testing Type: Black Box

Methodology

TESTING PHASES



Reconnaissance

Gather information about the app, including its architecture, APIs, and potential attack vectors.

Scanning

Identify vulnerabilities by scanning the application and its backend services for weaknesses, misconfigurations, or outdated libraries.

Exploitation

Actively exploit identified vulnerabilities to assess the extent of potential damage and unauthorized access.

Post-Exploitation

Evaluate what data can be accessed after exploitation, including sensitive user information and system controls.

Reporting

Document findings, including vulnerabilities discovered, exploitation methods used, and recommendations for remediation to enhance security.

Mobile
Penteration
Team

Why? WE SKIPPED RECON PHASE ?!

In penetration testing, the Reconnaissance (Recon) phase is typically crucial for gathering initial information about the target. However, in some cases, this phase can be skipped if the necessary information is already available or if the penetration tester has been provided with the essential details. In this scenario, the target was a well-known website, and the required information had already been provided, making the Recon phase unnecessary

KEY FINDINGS

MASVS-RESILIENCE-1:Emulator, Root, and Debugging Detection Bypassed

Insecure Authentication Mechanism

MASVS-RESILIENCE-3: Obfuscation

Insecure Data Storage leads to personal data exposure

MASVS-RESILIENCE-2: Vulnerability to Janus (CVE-2017-13156)

Key Findings

KEY FINDINGS

- Insecure OTP Handling in updateMe API
- Lack of Rate Limiting & Weak OTP Mechanism
- Weak Authentication Mechanism
- Firebase Misconfiguration – Unrestricted File Upload and Public Access
- Insecure Direct Object (IDOR) in Firebase Storage Access

Key Findings

RECOMMENDATIONS

Recommendations for closing vulnerabilities are suggested actions or best practices aimed at addressing identified security weaknesses in a system or application. These measures enhance security, protect sensitive data, and reduce the risk of exploitation by attackers.

MASVS-RESILIENCE-1:EMULATOR, ROOT, AND DEBUGGING DETECTION BYPASSED

- Implement robust detection methods that check for emulators, rooted devices, and debugging tools to prevent the app from running in insecure environments.

INSECURE AUTHENTICATION MECHANISM

- Strengthen authentication processes by implementing Multi-Factor Authentication (MFA) and using secure methods like OAuth for user login.

Recommendations

MASVS-RESILIENCE-3: OBFUSCATION

- Enhance code obfuscation techniques to make reverse engineering difficult, using tools that obfuscate your codebase effectively.

INSECURE DATA STORAGE LEADS TO PERSONAL DATA EXPOSURE

- Encrypt sensitive data before storage and utilize secure storage options, such as the Android Keystore or iOS Keychain, to protect user data.

Recommendations

MASVS-RESILIENCE-2: VULNERABILITY TO JANUS (CVE-2017-13156)

- Apply security patches provided by the developers to fix the Janus vulnerability and ensure that the app is protected against modified APK files.

INSECURE OTP HANDLING IN UPDATEME API

- Ensure that OTPs are securely generated, transmitted over encrypted channels (e.g., HTTPS), and validate them securely on the server side.

Recommendations

LACK OF RATE LIMITING & WEAK OTP MECHANISM

- Implement rate limiting to prevent abuse and use strong OTP generation methods, like TOTP (Time-based One-Time Password), to enhance security.

WEAK AUTHENTICATION MECHANISM

- Adopt secure authentication practices, including session management, token-based authentication, and implementing stricter password policies.

Recommendations

FIREBASE MISCONFIGURATION – UNRESTRICTED FILE UPLOAD AND PUBLIC ACCESS

- Review and update Firebase security rules to restrict file upload access and ensure that only authenticated users can access sensitive data.

INSECURE DIRECT OBJECT (IDOR) IN FIREBASE STORAGE ACCESS

- Implement proper access controls to ensure that users can only access their own data and not others' data within Firebase storage.

Recommendations

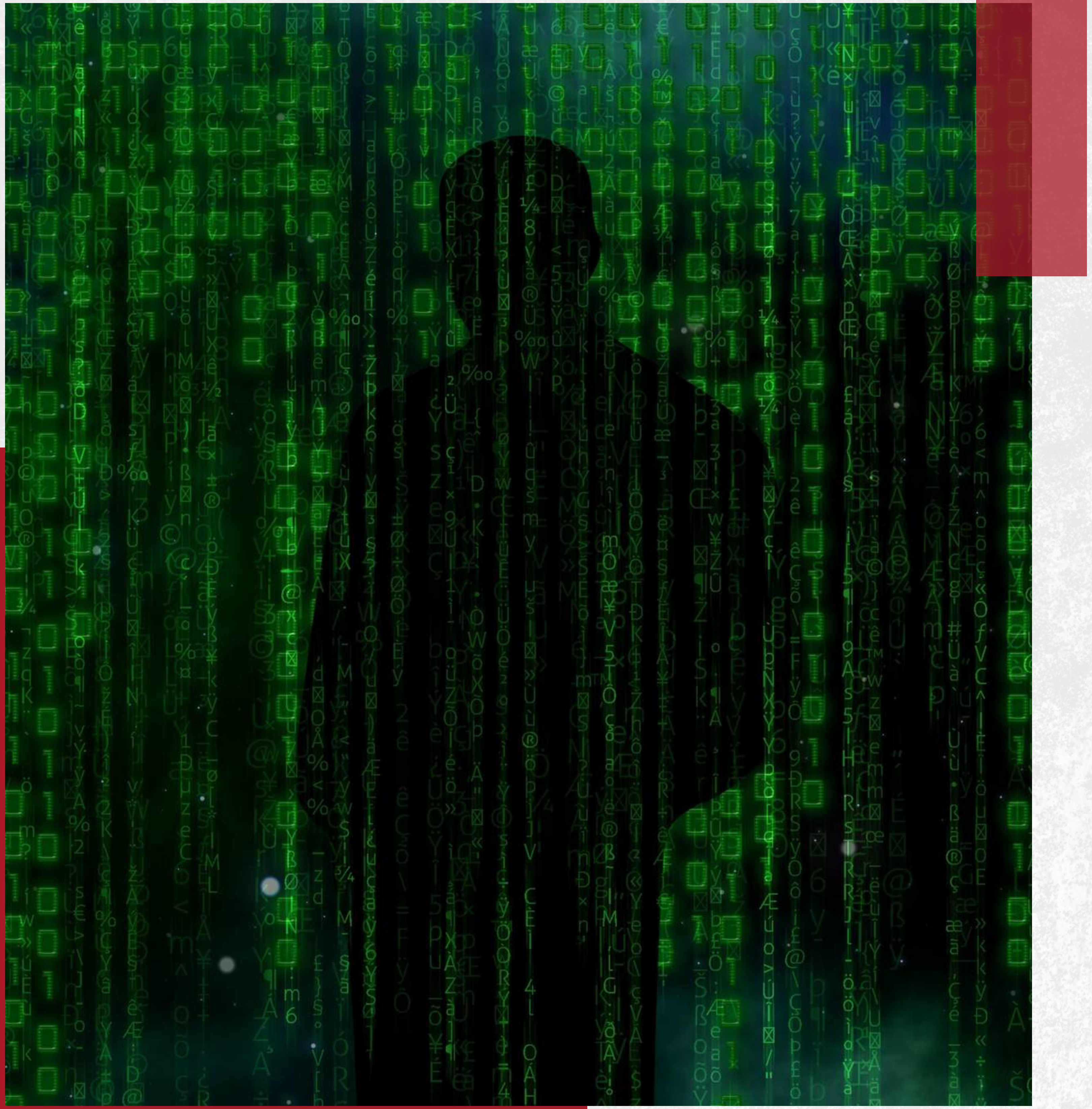
Ziad Mohamed

Abdelrahman Gomah

Mohamed Adel

Mohamed Makram

Noor Mahmoud



CONCLUSIONS

This testing was based on the technologies and known threats as of the date of this document. All the security issues discovered during that exercise were analyzed and described in this report. Please note that as technologies and risks change over time, the vulnerabilities associated with the operation of systems described in this report, as well as the actions necessary to reduce the exposure to such vulnerabilities, will also change

THANK YOU

for your attention