Penetration Re-Testing Report

(Mobile Banking E-Statments)

PREPARED FOR

FABMisr

PREPARED BY

Offensive Security Team

Table Of Content

Overview

Engagement Summary:

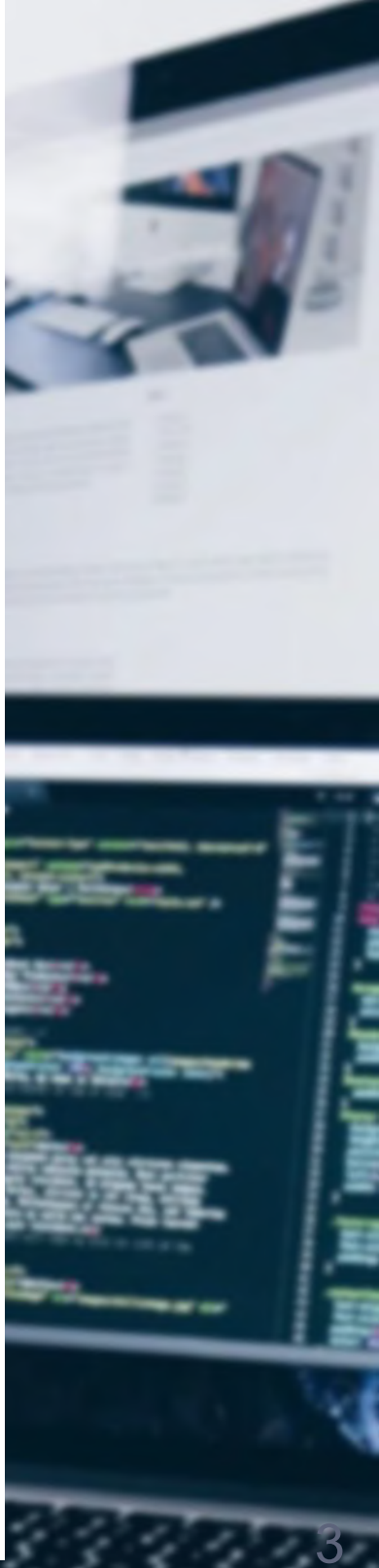
On Fabuary 5th, we conducted a penetration Re-testing engagement on FABMisr's Mobile Banking E-Statement function. This assessment included the mobile application (Android/iOS) and the web application. The objective was to identify and evaluate potential security vulnerabilities within both platforms, ensuring the protection of sensitive data and the integrity of the system.

Scope and Methodology:

The penetration test focused on the mobile banking application's e-statement functionality for Android and iOS. Following the OWASP Mobile Security Testing Guide (MSTG), we evaluated key areas such as user authentication, session management, data storage, and secure communication specific to the e-statement feature. We examined vulnerabilities in application logic, input validation, and data transmission, as well as searched for hardcoded secrets and insecure configurations through reverse engineering and code analysis.
This thorough assessment identified security weaknesses and areas for improvement to enhance the security of the e-statement functionality against potential threats.

Key Findings

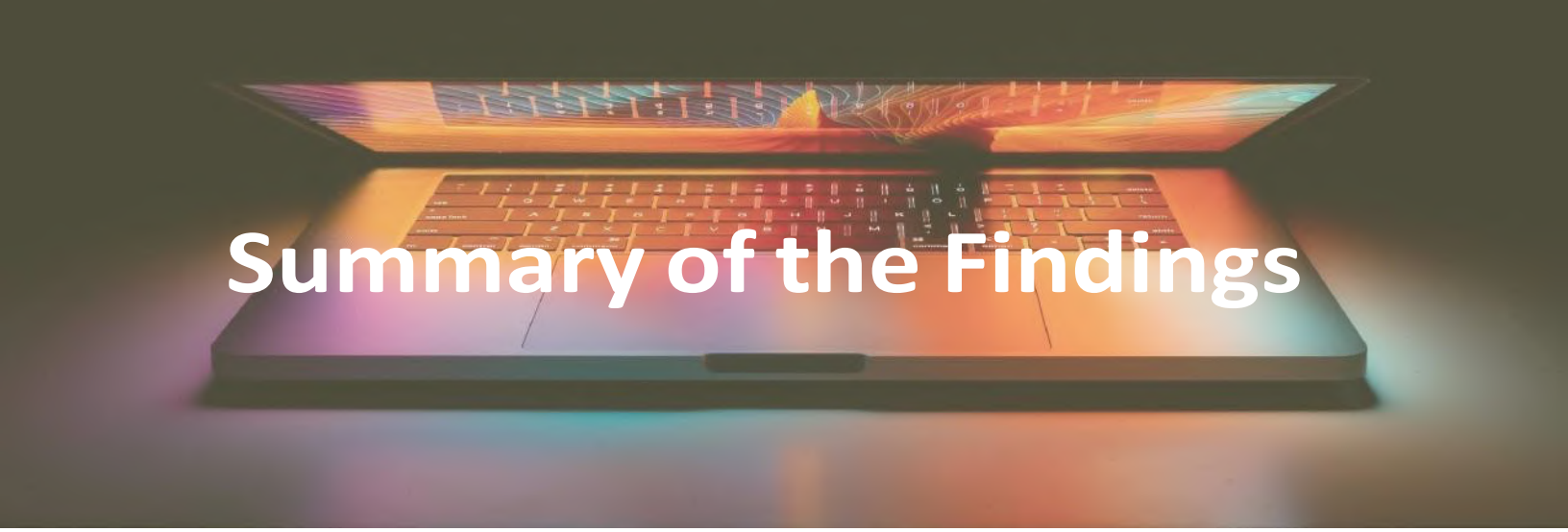
Our testing resulted in the identification of 3 vulnerabilities categorized by their severity:



Document Control – Penetration Testing

Item	Description
Requestor	**FABMisr**
Service Covered	Penetration Testing
File Name	FABMisr Mobile Banking E-Statements Penetration Re-Testing Report
Publish Date	February 6, 2025
Publish Number	1.2

Version	Date	Name	Description
1.0	6th/February/2025	Omar Ali	Document Creation
1.1	6th/February/2025	Mohammed Badawy	Document Review
1.2	6th/February /2025	Abdulrahman Khater	Final Review



Summary of the Findings

Finding	Severity	Status
No Rate Limiting on Resend OTP via Phone Number	Medium	Fixed
Bypassing OTP Verification via Default OTP Number	Medium	Fixed
No Rate Limiting on OTP Requests via Email	Medium	Fixed
Bypassing Rate Limit on Resend OTP Request via Email	LOW	Fixed

Risk Rating

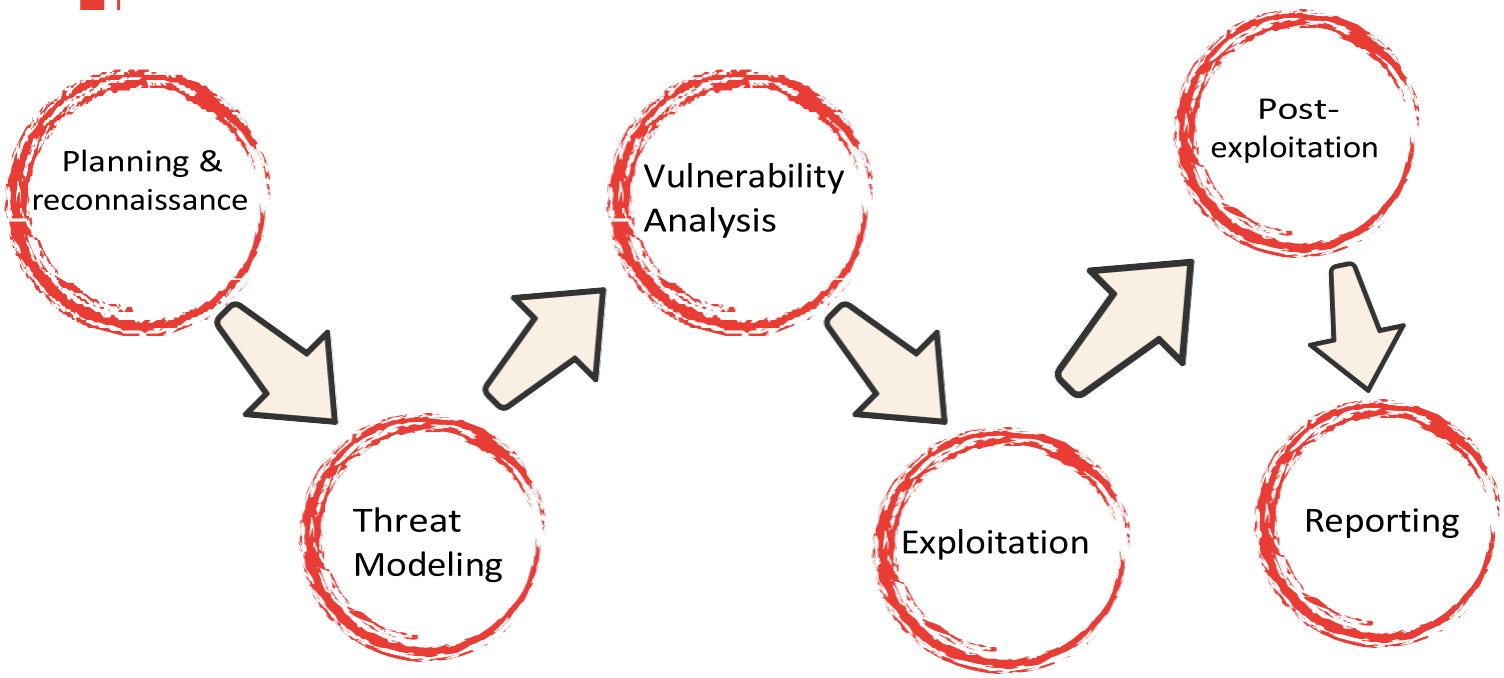
In this report, we have assigned each report finding an associated risk using the risk rating methodology outlined in the Open Web Application Security Project (OWASP). The underlying principle of this method is to calculate risk for each vulnerability using the standard risk model: Risk = Likelihood x Impact

Original Summery

Retest Summery

	Critical	HIGH	MEDIUM	LOW
Original Summery	**0**	**0**	**3**	**1**
Re-test Finding	**0**	**0**	**0**	**0**

Penetration Testing Methodology:



Penetration testing, also known as ethical hacking, is a proactive and strategic approach to fortifying an organization's cyber defenses. It involves simulated cyber-attacks conducted by skilled professionals to identify and rectify security weaknesses before malicious actors can exploit them. This crucial process is not just about identifying vulnerabilities but is an essential component of a comprehensive cybersecurity strategy.

The purpose of penetration testing extends beyond compliance requirements; it is a proactive measure to safeguard against the constantly changing landscape of cyber threats. By mimicking the tactics, techniques, and procedures of potential adversaries, penetration testing provides a realistic assessment of an organization's security posture. This proactive stance empowers businesses to stay ahead of emerging threats, anticipate vulnerabilities, and fortify their defenses effectively.

At our organization, our security team distinguishes itself by not adhering to a fixed methodology for penetration testing. Instead, we pride ourselves on crafting tailored penetration test methodologies that align precisely with each customer's unique requirements and specifications. This approach ensures that our security assessments are not only comprehensive but also specifically designed to address the individualized needs and concerns of our valued clients.

Our penetration testing methodology is designed not only to identify technical vulnerabilities but also to evaluate the effectiveness of security policies, incident response procedures, and employee awareness. This holistic approach ensures that your organization is resilient not just in the face of technological vulnerabilities but also in terms of its people and processes.

Penetration Testing Methodology:

This engagement has executed adhere to a dedicated methodology based on the 'OWASP Testing Guide' and the 'Open Source Security Testing Methodology Manual (OSSTMM)'. The procedures are outlined as follows:

1. Planning and Reconnaissance

1.1 Define Scope

The initial phase of any penetration testing engagement is to clearly define the scope. This involves collaboratively establishing the systems, networks, and applications that will be included in the testing process. Defining scope is critical to ensure that the assessment is targeted and aligned with your organization's specific security concerns. This step involves close consultation with key stakeholders to understand business priorities and areas of heightened sensitivity.

1.2 Authorization

Before initiating any penetration testing activities, it is imperative to obtain explicit authorization from relevant stakeholders. This authorization serves as a formal agreement that grants ethical hackers permission to probe and assess the security infrastructure. Obtaining clear and documented permission not only ensures compliance with legal and regulatory requirements but also prevents any potential disruption to business operations.

1.3 Information Gathering

In the information gathering phase, our team collects pertinent data about the target environment. This involves understanding the organizational structure, network architecture, and internet footprint. The goal is to gather as much open-source intelligence as possible to simulate the reconnaissance phase of a real-world cyber-attack. Information gathered in this stage forms the basis for subsequent testing activities.

This information gathering is conducted with a focus on minimizing the impact on your organization's operations, ensuring a non-disruptive and controlled process. The objective is to amass data that would be available to a potential adversary, enhancing the realism of the subsequent testing phases.

As part of our commitment to transparency and collaboration, regular communication is maintained with your organization's IT and security teams throughout the pre-engagement phase. This ensures that any specific concerns or considerations are addressed, and the penetration testing process aligns seamlessly with your business priorities.

The comprehensive pre-engagement activities lay the groundwork for a successful and targeted penetration testing engagement. By defining the scope, obtaining proper authorization, and gathering pertinent information, our team ensures that the subsequent testing phases are efficient, effective, and tailored to address your organization's unique security challenges.

Penetration Testing Methodology:

2. Threat Modeling

Threat modeling is a foundational step in our penetration testing methodology, aiming to provide a strategic and systematic approach to identifying potential threats and vulnerabilities within an organization's digital infrastructure. This phase is essential for prioritizing testing efforts, ensuring a targeted and effective assessment of the security landscape.

2.1 Identify Assets

In this initial step, our team collaborates with your organization to identify and categorize critical assets. These assets may include sensitive data, intellectual property, customer information, and key infrastructure components. The goal is to establish a comprehensive inventory of assets, allowing for a focused evaluation based on the criticality and importance of each element to your business operations.

2.2 Attack Surface Analysis

Understanding the attack surface is vital for anticipating potential points of compromise. Our experts analyze the digital landscape, mapping out entry points and potential weak links that could be exploited by malicious actors. This involves scrutinizing networks, systems, applications, and any interfaces that may expose vulnerabilities. By comprehensively assessing the attack surface, we ensure that our penetration testing efforts are aligned with the actual risks faced by your organization.

2.3 Risk Assessment

Once assets and attack surfaces are identified, our team conducts a thorough risk assessment. This involves evaluating the impact and likelihood of potential threats on your business operations. Risks are categorized based on severity, taking into account factors such as financial impact, reputational damage, and regulatory compliance. This risk-centric approach guides subsequent penetration testing activities, ensuring that efforts are concentrated on addressing the most critical and impactful vulnerabilities.

2.4 Mitigation Strategy

Building on the risk assessment, we collaborate with your team to develop a mitigation strategy. This strategy outlines proactive measures to reduce or eliminate identified risks. It includes recommendations for security controls, policy enhancements, and procedural changes aimed at fortifying the overall security posture. Our goal is not only to identify vulnerabilities but to empower your organization with actionable insights to strengthen its resilience against potential threats.

In the subsequent phases of our methodology, we leverage the insights gained during threat modeling to tailor our penetration testing efforts, ensuring a targeted and effective evaluation of your organization's cybersecurity defenses.

3. Vulnerability Analysis

3.1 Passive Scanning

3.1.1 Purpose:

Passive scanning serves as the initial phase of reconnaissance in our penetration testing methodology. Its primary goal is to gather crucial information about the target environment without actively engaging systems. This non-intrusive approach allows our team to understand the organization's infrastructure, potential vulnerabilities, and potential points of entry.

3.1.2 Stealthy Reconnaissance:

A. Low Footprint:

- **Objective:** Minimize the impact on the target environment by conducting reconnaissance activities without sending any active probes or queries directly to the systems.
- **Rationale:** This approach allows our team to gather crucial information discreetly, avoiding triggering any alerts from intrusion detection or prevention systems.

B. Foundation for Active Scanning:

Target Identification:

- **Objective:** Identify live hosts, services, and potential entry points within the target network.
- **Rationale:** Passive scanning sets the stage for subsequent active scanning phases by providing a clear picture of the network's topology and critical assets.

C. Informing Strategy:

- **Objective:** Gather intelligence that informs the penetration testing strategy, helping to prioritize targets and plan subsequent activities effectively.
- **Rationale:** The insights gained during passive scanning guide the team in determining the best approach for the active phases of the engagement.

D. Network Insight:

Topology Understanding:

- **Objective:** Map the network topology to understand the relationships between devices, subnets, and potential security zones.
- **Rationale:** This information aids in identifying potential chokepoints, critical infrastructure components, and areas of heightened security.

E. Domain Relationship Mapping:

- **Objective:** Analyze the relationships between different domains to gain insights into the organizational structure.
- **Rationale:** Understanding domain relationships helps in identifying dependencies and potential weak points in the domain hierarchy.

F. Data Collection:

DNS Record Enumeration:

- **Objective:** Query DNS records passively to collect information about the organization's domain names, subdomains, and associated IP addresses.
- **Rationale:** Extracting DNS information provides valuable details such as mail servers, name servers, and domain registrar information.

Penetration Testing Methodology:

i. Trafic Capture:

- **Objective:** Deploy Wireshark for passive traffic capture to analyze network communications.
- **Rationale:** Capturing network traffic passively allows for the extraction of user agents, protocols in use, and potential security risks from the data.

Activities:

- **Network Discovery:**
- Leverage tools like Nmap for passive network reconnaissance, identifying live hosts, services, and open ports. • Analyze network topology to understand the architecture and potential security boundaries.
- **Passive DNS Analysis:**
- Query DNS records passively to collect information about the target's domain names, subdomains, and associated IP addresses.
- Identify the relationships between different domains, providing insights into the organizational structure.
- **Network Snifing:**
- Deploy packet-sniffing tools like Wireshark to capture and analyze network traffic passively.
- Extract information such as user agents, protocols in use, and potential security risks from the intercepted data.

Beneits of Passive Scanning:

Informed Decision-Making:

- **Beneit:** Equip decision-makers with foundational insights to guide subsequent penetration testing activities.
- **Explanation:** The information gathered during passive scanning informs decisions on target prioritization, attack vectors, and overall testing strategy.

Reduced Alert Triggering:

- **Beneit:** Minimize the likelihood of triggering alerts or defensive responses from the target environment.
- **Explanation:** Operating stealthily ensures that the organization's security measures are not prematurely activated, allowing for a smoother progression through the testing phases.

Eficient Targeting:

- **Beneit:** Enable efficient targeting of critical assets and potential vulnerabilities.
- **Explanation:** The insights gained help the penetration testing team focus on areas of higher risk and importance, optimizing the use of resources.

Passive scanning is not merely an introductory step but a strategic and calculated approach to gather foundational intelligence. By balancing the need for information with the requirement for discretion, this phase ensures that subsequent testing activities are well-informed, targeted, and aligned with the overall goals of the penetration testing engagement.

3.2 Active Scanning

Purpose:
Active scanning is a pivotal phase in our penetration testing methodology, involving the use of automated tools to actively interrogate target systems. This proactive approach aims to identify vulnerabilities, potential entry points, and weaknesses within the target environment. The activities within this phase are carefully orchestrated to provide a comprehensive assessment of the security posture.

A. Vulnerability Scanning:

Utilization of Nessus/Acunetix :

- **Objective:** Employ advanced vulnerability scanners like Nessus, Acunetix to perform active scans across the network.
- **Rationale:** Identify known vulnerabilities in software, operating systems, and configurations, providing a comprehensive list of potential weaknesses.

CVE Database Correlation:

- **Objective:** Correlate identified vulnerabilities with the Common Vulnerabilities and Exposures (CVE) database for accurate classification and severity assessment.
- **Rationale:** Linking vulnerabilities to the CVE database ensures a standardized and universally recognized categorization.

B. Service Enumeration:

Active Querying with Nmap:

- **Objective:** Actively query services on the network using tools like Nmap to enumerate information about running applications, versions, and configurations.
- **Rationale:** Understand the services in use, including potential weak points that could be exploited during subsequent testing phases.

Banner Grabbing Techniques:

- **Objective:** Employ banner grabbing techniques to extract information from service banners, revealing details about the services and their versions.
- **Rationale:** Detailed service information aids in assessing potential vulnerabilities and identifying outdated or unpatched software.

C. Web Application Scanning:

Automated Scanning with Burp Suite/Acunetix/Tenable/Invicti:

- **Objective:** Utilize automated scanners such as Burp Suite pro scanner or to identify vulnerabilities specific to web applications.
- **Rationale:** Test for common web application vulnerabilities, including SQL injection, cross-site scripting (XSS), and security misconfigurations.

Authentication Testing:

- **Objective:** Simulate authentication processes to identify vulnerabilities related to user authentication and session management.
- **Rationale:** Evaluate the robustness of authentication mechanisms, ensuring secure access control.

Penetration Testing Methodology:

Benefits:

Comprehensive Vulnerability Identification:

- **Benefit:** Identify a wide range of vulnerabilities across different layers of the target environment.
- **Explanation:** Active scanning covers vulnerabilities in network services, applications, and web interfaces, ensuring a comprehensive assessment.

Prioritization of Remediation Efforts:

- **Benefit:** Provide a prioritized list of vulnerabilities based on severity and potential impact.
- **Explanation:** This helps organizations focus their remediation efforts on critical vulnerabilities that pose the highest risk.

Baseline for Security Posture:

- **Benefit:** Establish a baseline for the security posture of the target environment.
- **Explanation:** The findings from active scanning serve as a benchmark, enabling organizations to track improvements over time.

Automated Exploitation:

1. **Metasploit Framework:**

- **Objective:** Optionally, use tools like the Metasploit Framework to automate the exploitation of known vulnerabilities.
- **Rationale:** Demonstrate the potential impact of identified vulnerabilities, helping organizations understand the real-world consequences.

Safe and Controlled Exploitation:

- **Objective:** If automated exploitation is conducted, ensure it is done in a safe and controlled environment to prevent unintended consequences.
- **Rationale:** Automated exploitation serves as a proof-of-concept, showcasing the potential risk without causing harm to production systems.

Active scanning is a dynamic and proactive phase that not only identifies vulnerabilities but also assesses the potential impact and risks associated with them. By utilizing advanced scanning tools and methodologies, this phase plays a crucial role in fortifying an organization's security defenses.

Penetration Testing Methodology:

3.3 Manual Analysis

Purpose:
Manual analysis in penetration testing serves as a critical and intricate phase conducted by skilled security professionals. This phase extends beyond automated tools, aiming to uncover nuanced and context-specific security weaknesses. The primary objectives of manual analysis are multi-faceted and contribute significantly to the overall effectiveness of the penetration testing engagement.

A. Code Review:

1. **Identification of Vulnerabilities:**
   - **Objective:** Conduct a meticulous review of web application code to identify security flaws that may not be detectable by automated tools.
   - **Rationale:** Automated scanners might miss logic-based vulnerabilities or issues specific to the application's unique architecture, making manual code review crucial.
2. **Architecture and Logic Assessment:**
   - **Objective:** Analyze the application's architecture and logic for potential vulnerabilities that could be exploited by an attacker.
   - **Rationale:** Understanding the intricate details of the application's design allows for the identification of vulnerabilities arising from complex interactions and workflows.

B. Password Analysis:

1. **Policy Evaluation:**
   - **Objective:** Manually assess password policies, including complexity requirements, expiration periods, and storage mechanisms.
   - **Rationale:** Automated tools may not fully capture the nuances of password security policies, and manual analysis ensures a comprehensive evaluation.
2. **Weak Password Identification:**
   - **Objective:** Test for weak or default passwords that might pose a risk to the organization's security.
   - **Rationale:** Human intuition is valuable in recognizing patterns and common pitfalls related to password choices that automated tools might overlook.

C. Custom Exploitation:

1. **Exploration of Unique Attack Vectors:**
   - **Objective:** Explore unique attack vectors that automated tools may overlook, including business logic vulnerabilities and unconventional entry points.
   - **Rationale:** Custom exploitation allows for the identification of vulnerabilities specific to the organization's unique processes and applications.
2. **Zero-Day Vulnerability Testing:**
   - **Objective:** Conduct targeted testing to identify potential zero-day vulnerabilities that may not have been addressed by existing patches.
   - **Rationale:** Manual testing helps simulate the tactics of sophisticated attackers who leverage undisclosed vulnerabilities for exploitation.

Penetration Testing Methodology:

D. Security Architecture Review:

1. **Overall Assessment:**
   - **Objective:** Evaluate the overall security architecture of the target environment, identifying potential misconfigurations and weaknesses in the design.
   - **Rationale:** A holistic view of security architecture ensures that the organization's defenses are robust at both macro and micro levels.

1. **Effectiveness of Security Controls:**
   - **Objective:** Analyze the effectiveness of security controls in place and assess adherence to industry best practices.
   - **Rationale:** Understanding how well security controls are implemented provides insights into the organization's resilience against various cyber threats.

E. Social Engineering Assessment:

1. **Simulation of Social Engineering Attacks:**
   - **Objective:** Simulate social engineering attacks to assess the human factor in security.
   - **Rationale:** Assessing employee awareness, susceptibility to phishing, and the organization's ability to detect social engineering attempts is vital for a comprehensive security strategy.
2. **Response Evaluation:**
   - **Objective:** Evaluate the organization's response mechanisms to social engineering attempts.
   - **Rationale:** Understanding how well the organization can detect and respond to social engineering attacks helps strengthen the human layer of defense.

Benefits:

A. In-Depth Vulnerability Identification:

- **Benefit:** Identify subtle vulnerabilities that automated tools may overlook.
- **Explanation:** The human intuition of skilled professionals allows for a nuanced understanding of potential weaknesses, especially in complex systems.

B. Real-world Scenario Simulation:

- **Benefit:** Simulate real-world attack scenarios that automated tools might not replicate.
- **Explanation:** Manual analysis enables the exploration of unique attack vectors, mimicking the tactics of advanced adversaries.

C. Holistic Security Assessment:

- **Benefit:** Assess not only technical vulnerabilities but also broader security aspects.
- **Explanation:** Security architecture, password policies, and social engineering awareness contribute to a comprehensive evaluation of the organization's security posture.

Penetration Testing Methodology:

D. Custom-Tailored Recommendations:

- **Benefit:** Provide recommendations tailored to the organization's specific vulnerabilities and processes. **Explanation:**
- Customized recommendations ensure that remediation efforts align with the organization's unique security landscape.

E. Human-Centric Security Enhancement:

- **Benefit:** Strengthen the human layer of defense against social engineering attacks.
- **Explanation:** Assessing employee awareness and response capabilities enhances the organization's resilience to human-centric threats.

Manual analysis is a cornerstone of the penetration testing methodology, enabling a holistic assessment that goes beyond automated assessments. Skilled professionals apply their expertise to uncover vulnerabilities that may pose a risk in real-world scenarios, ensuring that the organization is well-prepared to defend against a diverse range of cyber threats.

Penetration Testing Methodology:

4. Exploitation

4.1 Exploit Identification

Purpose:

Exploit identification is a pivotal phase in penetration testing where potential vulnerabilities identified in the previous stages are actively tested for their exploit-ability. The primary goals are to determine the severity of each vulnerability, understand potential attack vectors, and assess the impact of successful exploitation on the target environment.

Activities:

A. Vulnerability Prioritization:

1. **Risk Ranking:**
   - **Objective:** Prioritize vulnerabilities based on their severity, potential impact on the organization, and the likelihood of successful exploitation.
   - **Rationale:** This prioritization guides the penetration testing team in focusing efforts on addressing the most critical security risks
2. **Business Context Consideration:**
   - **Objective:** Consider the business context to align exploit testing with the organization's priorities and potential financial or reputational impacts.
   - **Rationale:** Understanding the business context ensures that the testing aligns with the organization's risk tolerance and strategic goals.

B. Exploitation Attempts:

1. **Automated Exploitation:**
   - **Objective:** Use automated tools like Metasploit to conduct initial exploitation attempts on identified vulnerabilities.
   - **Rationale:** Automated tools can rapidly test known exploits, providing a quick assessment of vulnerability severity.
2. **Manual Exploitation:**
   - **Objective:** Conduct manual exploitation to verify and explore the intricacies of vulnerabilities that automated tools might miss.
   - **Rationale:** Manual testing allows for a more nuanced understanding of potential weaknesses and the identification of zero-day vulnerabilities.

Penetration Testing Methodology:

4.2 Privilege Escalation

Purpose:

Privilege escalation testing evaluates the target system's resistance to unauthorized access attempts, ensuring that even if an initial compromise occurs, the penetration tester cannot gain unauthorized privileges. This phase is crucial for identifying weaknesses that might allow an attacker to elevate their access level within the environment.

Activities:

A. User Account Testing:

1. **Credential Verification:**
   - **Objective:** Attempt to verify the validity of compromised credentials and assess their potential for unauthorized access.
   - **Rationale:** Verifying credentials helps understand the scope of a potential compromise and assess the need for further security measures.
2. **Brute-Force Attempts:**
   - **Objective:** Conduct brute-force attacks to test the strength of password-based authentication.
   - **Rationale:** Identifying weak passwords or misconflgured authentication settings is essential for mitigating the risk of unauthorized access.

B. Privilege Escalation Attempts:

1. **Exploitation of System Weaknesses:**
   - **Objective:** Identify and exploit weaknesses in the target system that might allow elevation of privileges.
   - **Rationale:** Understanding the pathways to privilege escalation provides insights into potential weaknesses in the organization's security controls.
2. **Lateral Movement Testing:**
   - **Objective:** Attempt to move laterally within the network to assess the effectiveness of network segmentation and access controls.
   - **Rationale:** Evaluating lateral movement capabilities simulates real-world scenarios where an attacker seeks to expand their influence within the network.

Penetration Testing Methodology:

Benefits:

A. Realistic Threat Simulation:

- **Benefit:** Mimic real-world attacker behaviors to provide a realistic assessment of potential risks.
- **Explanation:** By simulating the actions of an attacker, organizations gain insights into their actual defensive capabilities and potential points of failure.

B. Identification of Critical Weaknesses:

- **Benefit:** Identify critical weaknesses that might lead to unauthorized access, privilege escalation, or data breaches.
- **Explanation:** Highlighting critical weaknesses enables organizations to prioritize remediation efforts effectively.

C. Impact Assessment:

- **Benefit:** Assess the potential impact of successful exploitation on the target environment.
- **Explanation:** Understanding the impact helps organizations gauge the severity of potential security breaches and plan mitigating actions.

D. Data Protection Insights:

- **Benefit:** Gain insights into the effectiveness of data protection measures.
- **Explanation:** Assessing data exfiltration capabilities helps organizations enhance their data loss prevention strategies.

E. Post-Compromise Resilience:

- **Benefit:** Evaluate the resilience of the target environment after a successful compromise.
- **Explanation:** Identifying weaknesses in persistence and anti-forensic measures helps organizations enhance their post-compromise resilience.

The exploitation and post-exploitation phases provide a comprehensive evaluation of an organization's ability to resist and respond to cyber threats. By simulating real-world attack scenarios, penetration testing helps organizations identify and address vulnerabilities before malicious actors can exploit them.

Penetration Testing Methodology:

4.3 Post-Exploitation

Purpose:

Post-exploitation activities simulate the actions of an attacker who has successfully compromised a system. This phase evaluates the ability to maintain unauthorized access, exfiltrate sensitive information, and assesses the overall impact on the target environment.

Activities:

A. Data Exfiltration Testing:

1. **Sensitive Data Identification:**
   - **Objective:** Identify sensitive data within the compromised environment that might be of value to an attacker.
   - **Rationale:** Understanding what data is at risk helps organizations tailor their security measures to protect critical assets.
2. **Simulated Exfiltration:**
   - **Objective:** Simulate the extraction of sensitive information to assess the effectiveness of data loss prevention measures.
   - **Rationale:** Identifying potential weaknesses in data protection ensures a robust defense against unauthorized data access.

B. Persistence Testing:

1. **Registry and Configuration Modifications:**
   - **Objective:** Modify system configurations and registry settings to test the ability to maintain access even after the initial compromise.
   - **Rationale:** Assessing persistence capabilities helps understand the potential impact of a successful breach on long-term system integrity.
2. **Anti-Forensic Techniques:**
   - **Objective:** Employ anti-forensic techniques to erase or obfuscate traces of the penetration tester's activities.
   - **Rationale:** Mimicking real-world attackers who seek to cover their tracks provides insights into potential challenges faced by forensic investigators.

Benefits:

A. Realistic Threat Simulation:

- **Benefit:** Mimic real-world attacker behaviors to provide a realistic assessment of potential risks.
- **Explanation:** By simulating the actions of an attacker, organizations gain insights into their actual defensive capabilities and potential points of failure.

Penetration Testing Methodology:

B. Identification of Critical Weaknesses:

- **Benefit:** Identify critical weaknesses that might lead to unauthorized access, privilege escalation, or data breaches.
  Explanation: Highlighting critical weaknesses enables organizations to prioritize remediation efforts effectively.
-

C. Impact Assessment:

Benefit: Assess the potential impact of successful exploitation on the target environment.

- **Explanation:** Understanding the impact helps organizations gauge the severity of potential security breaches
- and plan mitigating actions.

D. Data Protection Insights:

- **Benefit:** Gain insights into the effectiveness of data protection measures.
- **Explanation:** Assessing data exfiltration capabilities helps organizations enhance their data loss prevention strategies.

E. Post-Compromise Resilience:

- **Benefit:** Evaluate the resilience of the target environment after a successful compromise.
- **Explanation:** Identifying weaknesses in persistence and anti-forensic measures helps organizations enhance their post-compromise resilience.

The purpose of the exploitation phase is to go beyond identifying vulnerabilities and actively test the security controls in place. By simulating realistic attack scenarios, penetration testing uncovers critical weaknesses that could have severe consequences if exploited by malicious actors. This phase is instrumental in strengthening an organization's overall cybersecurity posture.

Penetration Testing Methodology:

5. Reporting

The reporting phase is not merely a documentation process; it is a vital communication tool that empowers the organization to enhance its cybersecurity defenses. The comprehensive report, tailored for different stakeholders, ensures that the outcomes of the penetration testing engagement are understood, actionable, and aligned with the organization's strategic goals.

5.1 Executive Summary

Purpose:

The executive summary serves as a high-level overview of the penetration testing engagement, designed for senior leadership and key decision-makers. This section provides a snapshot of the assessment's outcomes, emphasizing critical findings, potential business impacts, and high-level recommendations.

Content:

1. **Overview of Findings:**
   - Summarize the key findings of the penetration testing engagement, focusing on critical vulnerabilities and their potential impact on the organization.
2. **Business Impact Assessment:**
   - Provide an assessment of the potential business impact associated with the identified vulnerabilities, considering factors such as financial implications and reputational risks.
3. **High-Level Recommendations:**
   - Outline high-level recommendations for addressing the most critical vulnerabilities and improving overall cybersecurity resilience.

5.2 Technical Details

Purpose:

The technical details section provides an in-depth analysis of the vulnerabilities discovered during the penetration testing engagement. It is tailored for the technical team, including system administrators, network engineers, and cybersecurity professionals responsible for implementing remediation measures.

Content:

1. **Detailed Vulnerability Report:**
   - Provide a comprehensive list of all identified vulnerabilities, including details such as severity, affected systems, and recommended remediation steps.
2. **Exploitation Details:**
   - Document the specifics of successful exploitation attempts, including the methods used, entry points, and potential impacts.
3. **Privilege Escalation Paths:**
   - Detail the paths taken to escalate privileges within the environment, emphasizing weaknesses in access controls and

Penetration Testing Methodology:

recommendations for improvement.

Penetration Testing Methodology:

5.3 Remediation Recommendations

Purpose:

The remediation recommendations section provides actionable guidance for addressing identified vulnerabilities and improving the overall security posture. It is designed to assist the organization in implementing effective and targeted remediation measures.

Content:

1. **Prioritized Remediation Plan:**
   - Present a prioritized plan for remediation, focusing on addressing the most critical vulnerabilities first.
   - Include a timeline for implementing remediation measures to guide the organization in managing risk effectively.
2. **Technical Mitigation Measures:**
   - Provide detailed technical measures for mitigating each identified vulnerability, including configuration changes, software updates, and patching instructions.
3. **Policy and Procedural Recommendations:**
   - Offer recommendations for enhancing security policies and procedures to prevent similar vulnerabilities in the future.
   - Emphasize the importance of employee training and awareness programs to strengthen the human layer of defense.

5.4 Lessons Learned and Best Practices

Purpose:

The lessons learned and best practices section focuses on extracting valuable insights from the penetration testing engagement. It aims to help the organization understand the root causes of vulnerabilities and adopt best practices to enhance its overall security posture.

Content:

1. **Root Cause Analysis:**
   - Analyze the root causes of identified vulnerabilities, exploring factors such as misconfigurations, lack of security controls, or gaps in security policies.
2. **Best Practices Recommendations:**
   - Provide recommendations based on industry best practices for cybersecurity, covering areas such as secure configuration, access controls, and incident response.
3. **Employee Training Suggestions:**
   - Offer suggestions for employee training and awareness programs, emphasizing the importance of a security- conscious workforce in preventing security incidents.

Penetration Testing Methodology:

5.5 Future Recommendations

Purpose:

The future recommendations section takes a forward-looking approach, providing guidance on sustaining and improving the organization's cybersecurity resilience beyond the current penetration testing engagement.

Content:

2. Continuous Improvement Strategies:

- Outline strategies for continuous improvement, including regular security assessments, threat intelligence integration, and ongoing employee training.

2. Emerging Threat Considerations:

- Highlight emerging cybersecurity threats and suggest proactive measures to stay ahead of evolving risks in the threat landscape.

3. Incident Response Planning:

- Recommend enhancements to the organization's incident response plan, ensuring a rapid and effective response to security incidents.

5.6 Legal and Ethical Considerations

Purpose:

The legal and ethical considerations section addresses the ethical aspects of the penetration testing engagement, ensuring that the testing was conducted within the bounds of legality and ethical standards.

Content:

1. **Scope Adherence Confirmation:**

- Confirm that the penetration testing activities were conducted within the agreed-upon scope and did not extend beyond the defined boundaries.

2. **Ethical Conduct Acknowledgment:**

- Acknowledge adherence to ethical conduct throughout the engagement, ensuring that testing activities were conducted responsibly and with respect for the organization's policies.

3. **Legal Compliance Verification:**

- Verify compliance with legal requirements, ensuring that the penetration testing activities were conducted in accordance with relevant laws and regulations.

Penetration Testing Methodology:

5.7 Stakeholder Briefing

Purpose:

The stakeholder briefing section involves presenting the key findings, recommendations, and action plans to stakeholders in a live or virtual briefing session. This interactive session allows for clarifications, questions, and direct communication between the penetration testing team and the organization's leadership.

Content:

1. **Presentation of Key Findings:**
   - Provide a concise presentation of the key findings, emphasizing critical vulnerabilities and their potential impact.
2. **Interactive Q&A Session:**
   - Allow stakeholders to ask questions, seek clarifications, and engage in a discussion about the penetration testing outcomes.
3. **Actionable Insights for Decision-Makers:**
   - Summarize actionable insights for decision-makers, emphasizing the importance of investing in cybersecurity measures based on the penetration testing results.

Tools Used

The use of a diverse set of tools, both automated and manual, allows for a comprehensive assessment of the target environment. It is essential to leverage the right tools for each phase of the penetration testing engagement, ensuring thorough coverage and accurate identification of vulnerabilities.

6.1 Scanning and Enumeration Tools

Purpose:
Scanning and enumeration tools play a crucial role in the initial phases of penetration testing, helping identify live hosts, services, and potential entry points within the target environment.

Tools:

Nmap:

- Purpose: Nmap is a network scanning tool designed to discover hosts and services on a computer network, thus creating a map of the network's structure.

MassScan:

- Purpose: MassScan is a high-speed port scanning tool primarily used for quickly identifying open ports on a large number of hosts.

BurpSuite:

- Purpose: BurpSuite is an integrated platform for web application security, providing tools for scanning, crawling, and analyzing web applications for potential vulnerabilities.

Nessus:

- Purpose: Nessus is a comprehensive vulnerability scanning tool that identifies and assesses security vulnerabilities in networks, systems, and applications.

Maltego Pro:

- Purpose: Maltego Pro is a data visualization and analysis tool used for link analysis and gathering information about entities on the internet to aid in the investigation and reconnaissance processes.

Acunetix Vulnerability Scanner:

- Purpose*:* Acunetix is designed to identify and assess vulnerabilities in web applications, providing automated scanning for security weaknesses such as SQL injection, cross-site scripting (XSS), and other common web-related threats.

SonarQube:

- Purpose*:* SonarQube serves as a continuous code quality inspection tool, analyzing source code for bugs, code smells, and security vulnerabilities, promoting the development of maintainable and secure software.

MobSF (Mobile Security Framework):

- Purpose*:* MobSF is tailored for mobile application security testing, offering dynamic and static analysis to identify vulnerabilities and weaknesses in mobile apps, ensuring robust security measures in mobile development.

Tools Used

Tools:

whois:

- Purpose: Whois performs the registration record for the domain name or IP address that you specify.

Sublist3r:

- Purpose: Fast subdomains enumeration tool for penetration testers

dnscan:

- Purpose: Fast and lightweight dns bruteforcer with built-in wordlist and zone transfer checks.

Recon-ng:

- Purpose: Open Source Intelligence gathering tool aimed at reducing the time spent harvesting information from open sources.

Dirb

- Purpose: DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary based attack against a web server and analyzing the responses.

Nuclei:

- Purpose: Fast and customizable vulnerability scanner based on simple YAML based DSL.

WPScan:

- The purpose of WPScan is to identify and enumerate vulnerabilities in WordPress websites by scanning for outdated plugins, themes, and known security issues, providing insights for robust security measures.

PingCastle:

- *Purpose:* PingCastle is used for Active Directory (AD) security assessments, providing insights into AD configuration weaknesses, security risks, and overall hygiene.

Seatbelt:

- *Purpose:* Seatbelt is a PowerShell script that performs various security-related checks on Windows systems, offering a comprehensive view of potential vulnerabilities and misconfigurations.

BloodHound:

- *Purpose:* BloodHound is a tool designed for analyzing and visualizing Active Directory attack paths, assisting in the identification of security risks and potential paths for privilege escalation within a network.

Tools Used

6.2 Exploitation Tools

Purpose:

Exploitation tools are utilized to actively test identified vulnerabilities, verifying their exploitability and assessing the potential impact on the target environment.

Tools:

Metasploit Framework:

- **Purpose:** Penetration testing framework.
- **Functionality:** Exploitation of known vulnerabilities, post-exploitation actions, and payload delivery.

sqlmap:

- **Purpose:** Automated SQL injection and database takeover tool.
- **Functionality:** Detect and exploit SQL injection vulnerabilities in web applications.

XSStrike:

- Purpose: To detect and exploit cross-site scripting vulnerabilities in web applications.

Wifite2:

- Purpose: To automate the wireless network penetration testing process by targeting vulnerabilities in Wi- Fi networks.

Hydra:

- Purpose: To perform brute-force attacks by systematically attempting various username and password combinations on login interfaces.

Hashcat:

- Purpose: To crack hashed passwords using various attack methods, including dictionary attacks and brute-force attacks.

John:

- Purpose: To identify weak passwords by performing password cracking through dictionary attacks and various hash cracking techniques.

Psexec:

- Purpose: To execute processes on remote Windows systems, often used for penetration testing and exploitation in a network environment.

NoSQLMap:

- Purpose: To automate the detection and exploitation of NoSQL injection vulnerabilities in web applications using NoSQL databases.

Msfvenom:

- Purpose: To generate customized and malicious payloads for exploitation, commonly used with the Metasploit framework for creating exploit modules.

Tools Used

6.3 Privilege Escalation and Post-Exploitation Tools

Purpose:

Privilege escalation and post-exploitation tools are employed to simulate real-world scenarios where attackers attempt to elevate their access and maintain control over compromised systems.

Tools:

windows-exploit-suggester:

- *Purpose:* To identify potential Windows vulnerabilities and suggest relevant exploits based on the operating system's configuration and patch level.

Windows-kernel-exploits:

- *Purpose:* Provides a repository of Windows kernel exploits, aiding security professionals in testing and assessing the security posture of Windows systems.

linux-exploit-suggester-2:

- *Purpose:* Identifies potential Linux vulnerabilities and recommends relevant exploits by analyzing the Linux system's configuration and installed software.

Linux-kernel-exploits:

- *Purpose:* Offers a collection of Linux kernel exploits for security professionals to assess and test the security resilience of Linux-based systems.

BeRoot:

- *Purpose:* Designed to assist in privilege escalation on Windows systems by identifying and exploiting potential vulnerabilities that may grant elevated access.

LinPEAS:

- *Purpose:* Facilitates privilege escalation on Linux systems by automating the discovery of misconfigurations, vulnerabilities, and potential routes to elevate privileges.

WinPEAS:

- *Purpose:* Similar to LinPEAS but tailored for Windows systems, WinPEAS automates the identification of security weaknesses and potential privilege escalation paths on Windows platforms.

Tools Used

6.4 Custom Scripts and Manual Tools

Purpose:

Custom scripts and manual tools are often developed and employed to address specific requirements and scenarios encountered during the penetration testing engagement.

Tools:

Custom Python Scripts:

- **Purpose:** Automating specific tasks or tests based on the unique characteristics of the target environment.
- **Functionality:** Tailored automation to address specific testing scenarios.

Manual Testing Techniques:

- **Purpose:** Hands-on testing conducted by skilled security professionals.
- **Functionality:** Customized testing methodologies and manual analysis to uncover vulnerabilities not detected by automated tools.

Scope of Work

The Penetration Re-Test was conducted on February 3rd, on the following System

Gray Box:

Mobile Banking E-Statement

Scope
Mobile Banking Android Application – E-Statements
Mobile Banking IOS Application – E-Statements
Web Application – E-Statements

OWASP Security Top 10

Test Type	Status
Input Validation Testing	
Validate and sanitize all user inputs.	PASSED
Use parameterized queries or prepared statements for database queries.	PASSED
validating, filtering & sanitizing all incoming data.	PASSED
Validate user permissions on the server side.	PASSED
Check for Reflected Cross Site Scripting vulnerabilities	PASSED
Check for Stored Cross Site Scripting vulnerabilities	PASSED
Check for DOM based Cross Site Scripting vulnerabilities	PASSED
Check for Cross Site Flashing vulnerabilities	PASSED
Check for HTML Injection vulnerabilities	PASSED
Check for SQL Injection vulnerabilities	PASSED
Check for LDAP Injection vulnerabilities	PASSED
Check for XML Injection vulnerabilities	PASSED
Check for XXE Injection vulnerabilities	PASSED
Check for HTTP Splitting/Smuggling vulnerabilities	PASSED
Check for HTTP Verb Tampering vulnerabilities	PASSED
Check for Open Redirection vulnerabilities	PASSED
Check for File Inclusion vulnerabilities	PASSED

OWASP Security Top 10

Test Type	Status
Broken Authentication	
Use strong authentication mechanisms (multi-factor authentication, password policies).	PASSED
Implement secure session management practices.	PASSED
Regularly test and update session timeout configurations.	PASSED
Test for user enumeration	PASSED
check for authentication bypass vulnerabilities	PASSED
check for bruteforce protection	PASSED
check password reset and/or recovery	PASSED
check password change process	PASSED
Check any flaws in CAPTCHA	PASSED
check multi factor authentication vulnerabilities	PASSED
Test for logout functionality presence	PASSED
Test for cache management on HTTP (eg Pragma, Expires, Max-age)	PASSED
Test for default logins	PASSED
Insecure Direct Object References (IDOR)	
Implement proper access controls to restrict unauthorized access.	PASSED
Validate user permissions on the server side.	PASSED
Avoid exposing internal implementation details in URLs.	PASSED

OWASP Security Top 10

Test Type	Status
Session Management	
Check session tokens for cookie flags (httpOnly and secure)	PASSED
Check session cookie scope (path and domain)	PASSED
Check session cookie duration (expires and max-age)	PASSED
Check session termination after a maximum lifetime	PASSED
Check session termination after relative timeout	PASSED
Check session termination after logout	PASSED
Test to see if users can have multiple simultaneous sessions	PASSED
Test session cookies for randomness	PASSED
File Uploads	
Test that file size limits, upload frequency and total file counts are defined and are enforced	PASSED
Test that file contents match the defined file type	PASSED
Test that unsafe filenames are sanitised	PASSED
Test that uploaded files are not directly accessible within the web root	PASSED
Test that files and other media are integrated with the authentication and authorisation schemas	PASSED
Test malicious file upload by encoded file name	PASSED

OWASP Security Top 10

Test Type	Status
Security misconfiguration	
Check that platform restrict administrative access.	PASSED
A Cross-Origin Resource Sharing (CORS) policy is missing or improperly set	PASSED
Error messages include stack traces, or expose other sensitive information	PASSED
Transport Layer Security (TLS) is missing	PASSED
The latest security patches are missing, or the systems are out of date	PASSED
Cross-Site Request Forgery (CSRF)	
Implement anti-CSRF tokens in forms and requests.	PASSED
Validate and compare the origin and referer headers.	PASSED
Sensitive Data Exposure	
Use strong encryption algorithms for sensitive data (TLS for data in transit, bcrypt for passwords, etc.).	PASSED
Ensure proper key management practices.	PASSED
Regularly audit and monitor access to sensitive information.	PASSED
Unvalidated Redirects and Forwards	
Avoid using user input to construct redirect URLs.	PASSED

OWASP Security Top 10

Test Type	Status
Broken Function Level Authorization	
Can a user perform sensitive actions (e.g. creation, modification, or deletion) that they should not have access to by simply changing the HTTP method (e.g. from `GET` to `DELETE`)	**PASSED**
Can a user from group X access a function that should be exposed only to users from group Y, by simply guessing the endpoint URL and parameters (e.g. /api/v1/users/export_all)?	**PASSED**
Check for path traversal Vulnerabilities	**PASSED**
Check for bypassing authorization schema	**PASSED**
Check for vertical Access control problems (a.k.a. Privilege Escalation)	**PASSED**
Test for horizontal Access control problems (between two users at the same privilege level)	**PASSED**
Secure Transmission	
Check SSL Version, Algorithms, Key length	**PASSED**
Check credentials only delivered over HTTPS	**PASSED**
Check that the login form is delivered over HTTPS	**PASSED**
Check session tokens only delivered over HTTPS	**PASSED**
Check if HTTP Strict Transport Security (HSTS) in use	**PASSED**
No-Rate Limiting	
Ensure rate limiting is enabled	**PASSED**
Try to bypass rate limiting by changing the case of the endpoints	**PASSED**
Try to bypass rate limiting by adding / at the end of the URL	**PASSED**

OWASP Security Top 10

Try Bypassing Rate Limit Mechanism	**PASSED**

Details Of The Findings

No Rate Limiting on Resend OTP via Phone Number

Severity: Medium

Status: Fixed

Affected Assets:
Mobile Application – Android / IOS

Description:
Our team discovered that the application does not implement rate limiting for the "Resend OTP" feature via phone number. This vulnerability allows an attacker or user to repeatedly trigger OTP requests without restriction, which can result in the legitimate user being blocked after a certain number of failed attempts. Such a flaw can lead to Denial-of-Service (DoS) attacks, where the target user is locked out of their account and unable to complete essential actions, creating a disruption in service.

Impact:
Denial of Service (DoS): Users can be locked out of their accounts by repeated OTP requests, preventing legitimate access.

User Inconvenience: Legitimate users may face service disruption or be locked out after multiple failed OTP attempts.

Abuse of Resources: The application's resources, such as SMS gateways, can be unnecessarily taxed due to repeated requests.

Details:
Our team found that the "Resend OTP" feature does not implement any rate-limiting controls. By testing the feature, we observed that repeatedly triggering OTP requests on the same phone number resulted in the l'egitimate user being blocked after several attempts. This lack of controls leaves the system vulnerable to both user inconvenience and potential abuse, as attackers could exploit the feature to lock out users.

Reproduction Steps:
1- Navigate to the OTP resend page of the application.
2- Enter a valid phone number and trigger the OTP resend function.
3- Note that there is no rate limit applied, and the system allows multiple OTP requests in quick succession.
4- Trigger multiple resend requests in a short period using a manual or automated process.
5- Observe that after several failed OTP attempts, the user is blocked or locked out of the system.

Details Of The Findings

Remediation

1- Implement rate limiting for the OTP resend functionality to restrict the number of resend attempts per phone number within a defined time period (e.g., 3 requests per minute).
2- Introduce CAPTCHA or other verification methods to prevent automated abuse of the resend functionality.
3- Notify users when they have reached the resend request limit and prevent further attempts until a cooldown period has passed.
4- Consider adding progressive delays between OTP resend attempts to prevent abuse and reduce the chance of brute-force attacks.
5- Log and monitor OTP requests to detect unusual patterns of activity that could indicate abuse

Proof Of Concept:

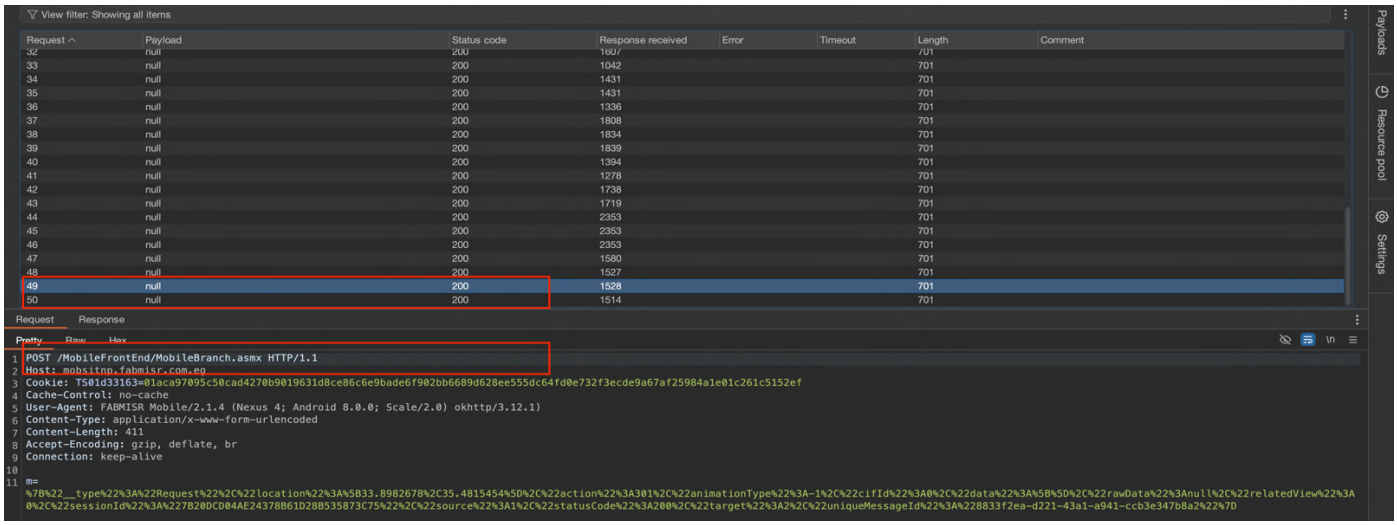


Figure 1 : Representation Request of the mobile OTP request.

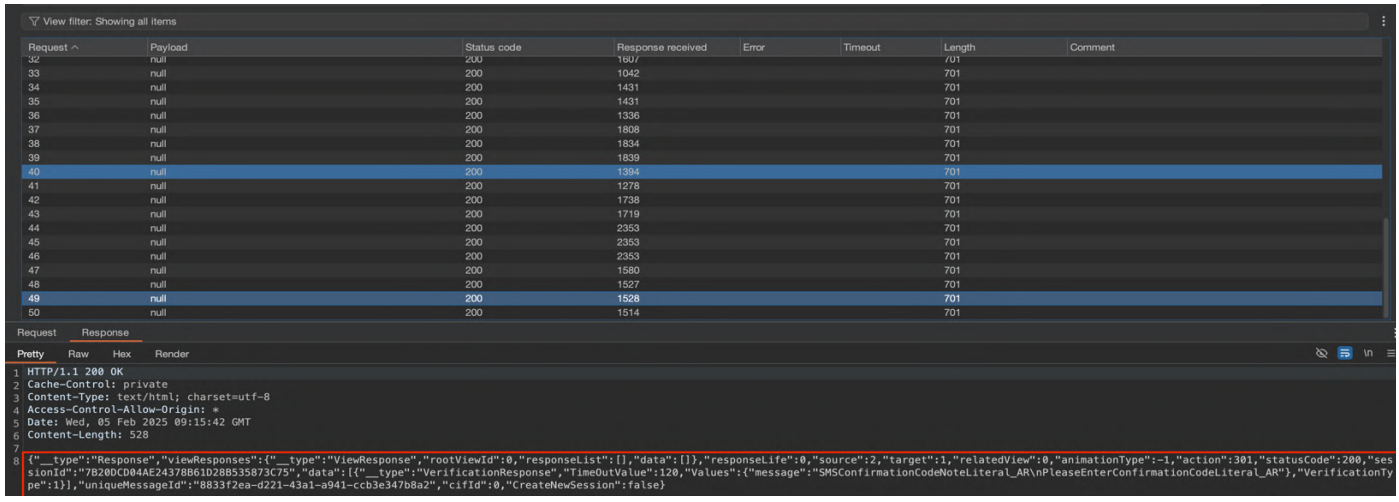


Figure 1 : Representation Response of the mobile OTP request.

Details Of The Findings

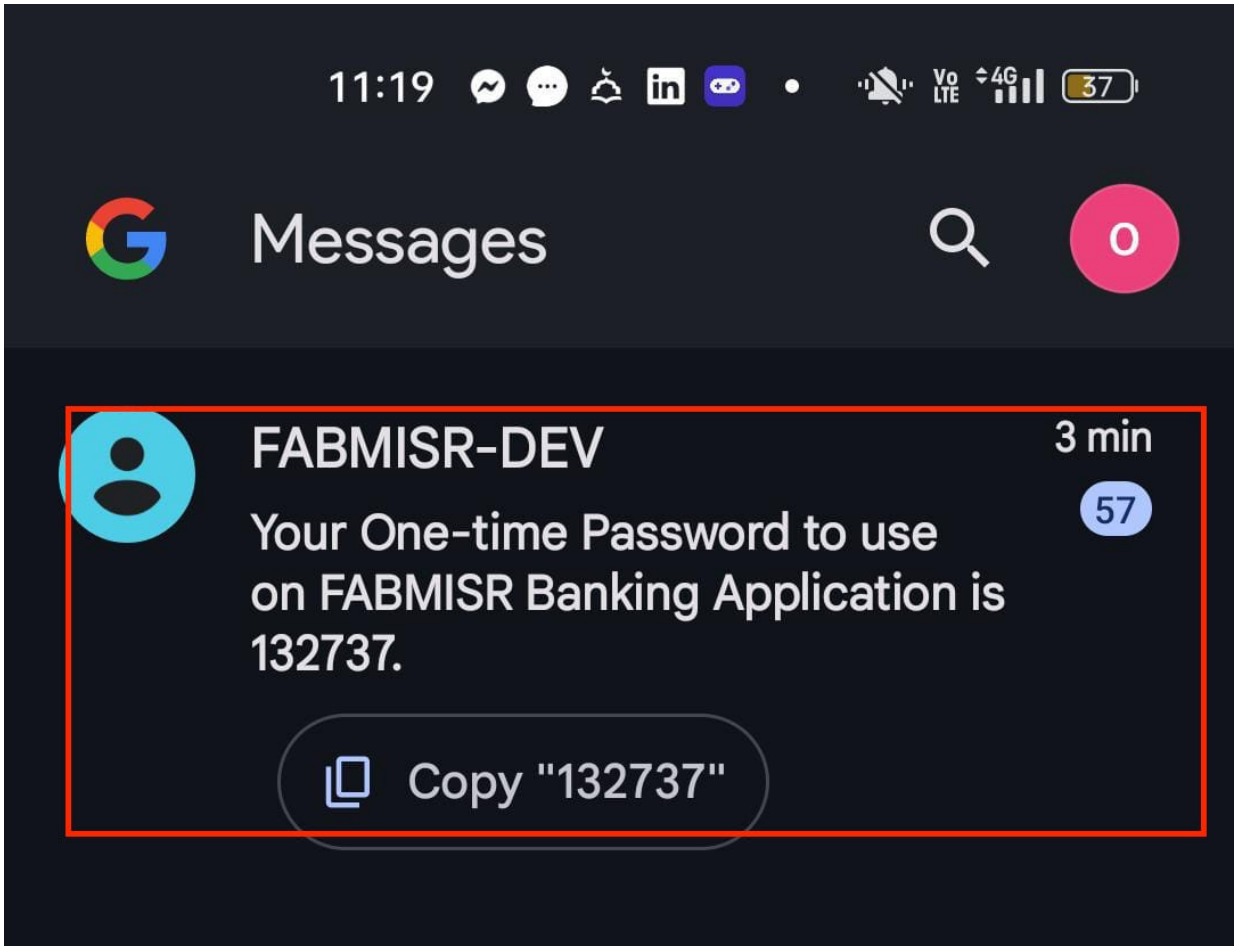


Figure 3: Representation of the mobile OTP received

Details Of The Findings

Proof Of Mitigation:

The vulnerability was mitigated by implementing rate limiting for the "Resend OTP" feature. After 10 OTP requests, the application blocks further requests and stops sending additional OTPs, preventing abuse and protecting users from Denial-of-Service (DoS) attacks or account lockouts.

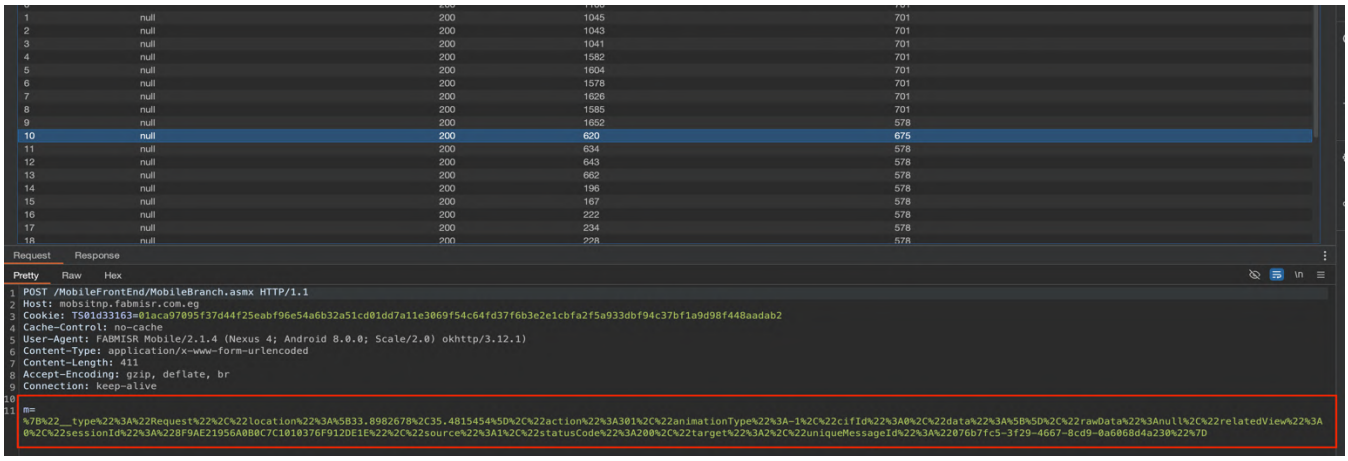


Figure 1: Representation of the mobile Resend OTP Request

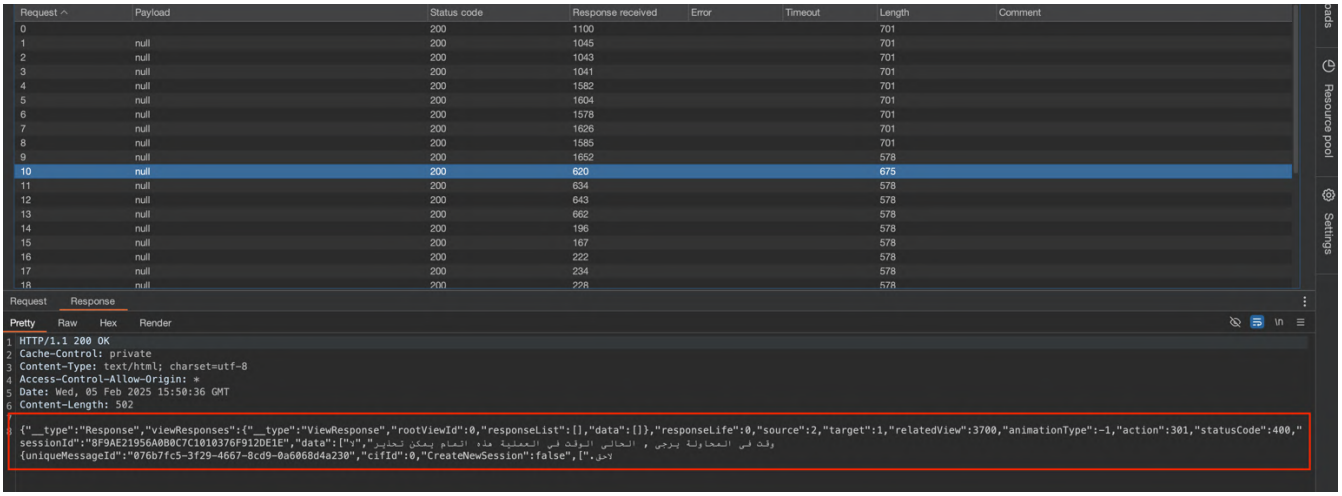


Figure 2: Representation of the mobile Resend OTP Request is failed sending OTP after 10 times

Details Of The Findings

Bypassing OTP Verification via Default OTP Number

Severity: Medium

Status: Fixed

Affected Assets:
Mobile Application – Wide Application

Description:
Our team discovered a vulnerability in the OTP verification mechanism where an attacker can bypass OTP authentication by using a hardcoded default OTP value. Specifically, if an attacker attempts to request an OTP via phone number and inputs "123456" as the OTP, the system will incorrectly accept it as valid. This occurs because the developer hardcoded "123456" as the default OTP value in the application. This flaw allows attackers to skip the OTP verification process and gain unauthorized access to the system or sensitive areas.

Impact:
Unauthorized Access: Attackers can bypass OTP authentication by using the default OTP value, enabling unauthorized access to accounts or systems.

Security Risk: This vulnerability undermines the effectiveness of the OTP mechanism, exposing the system to potential misuse and exploitation.

Data Exposure: Attackers gaining unauthorized access may be able to view or alter sensitive data, depending on the access granted by bypassing the OTP.

Details:
Our team found that the OTP verification system accepted "123456" as a valid OTP, which was hardcoded by the developers as a default value. This allowed us to bypass the OTP process entirely. By submitting "123456" as the OTP during the authentication process, we were granted access without the need for the actual dynamically generated OTP, revealing a serious flaw in the OTP mechanism.

Reproduction Steps:
1- Navigate to the OTP request page and input a valid phone number.
2- Attempt to request an OTP.
3- When prompted to enter the OTP, input "123456" as the OTP.
4- Observe that the system accepts the "123456" OTP as valid and grants access, despite it being a default value.
5- Repeat the process to confirm that this default OTP bypass works consistently.

Details Of The Findings

Remediation

1- Remove Hardcoded OTP: Ensure that no default or hardcoded OTP values are used in the system. Each OTP should be dynamically generated and unique for each request.
2- Improve OTP Generation: Use a secure and randomized method for generating OTPs that are not predictable.
3- Implement OTP Expiration: Set a short expiration time for OTPs to ensure they cannot be reused or bypassed after a certain period.
4- Secure OTP Transmission: Ensure OTPs are transmitted over secure channels, such as HTTPS, to prevent interception by attackers.

Proof Of Concept:

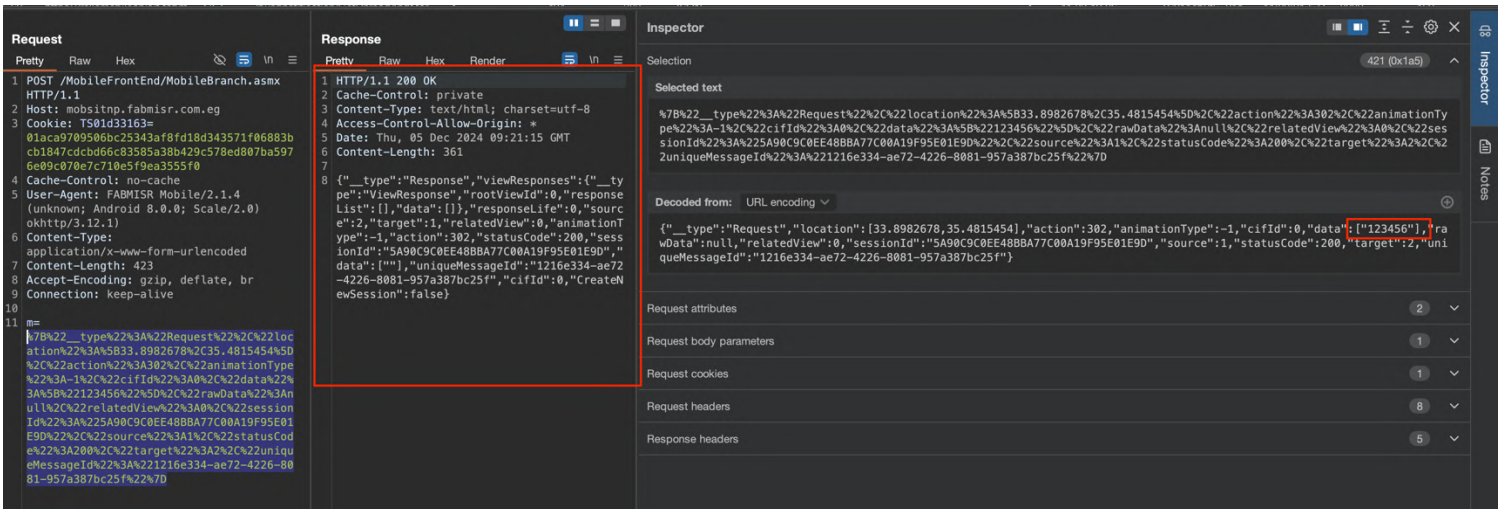


Figure1: Representation of the OTP number 123456.

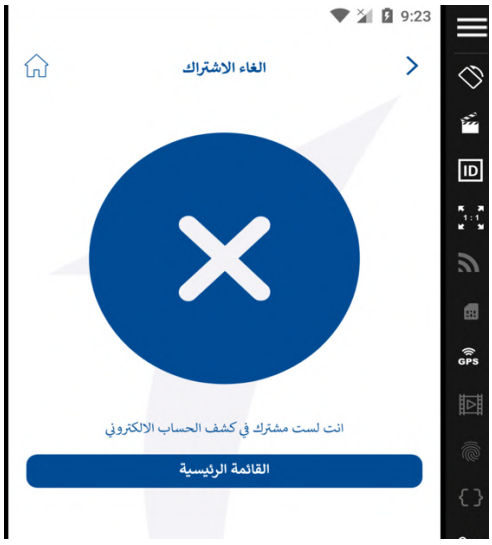


Figure1: Representation of bypassing the OTP number, and the un-subscription was successful.

Details Of The Findings

Proof Of Mitigation:

The vulnerability was mitigated by removing the hardcoded default OTP "123456" from the application and disabling the use of mock OTP values. This ensures that only dynamically generated and unique OTPs are accepted for verification, preventing bypass attacks.

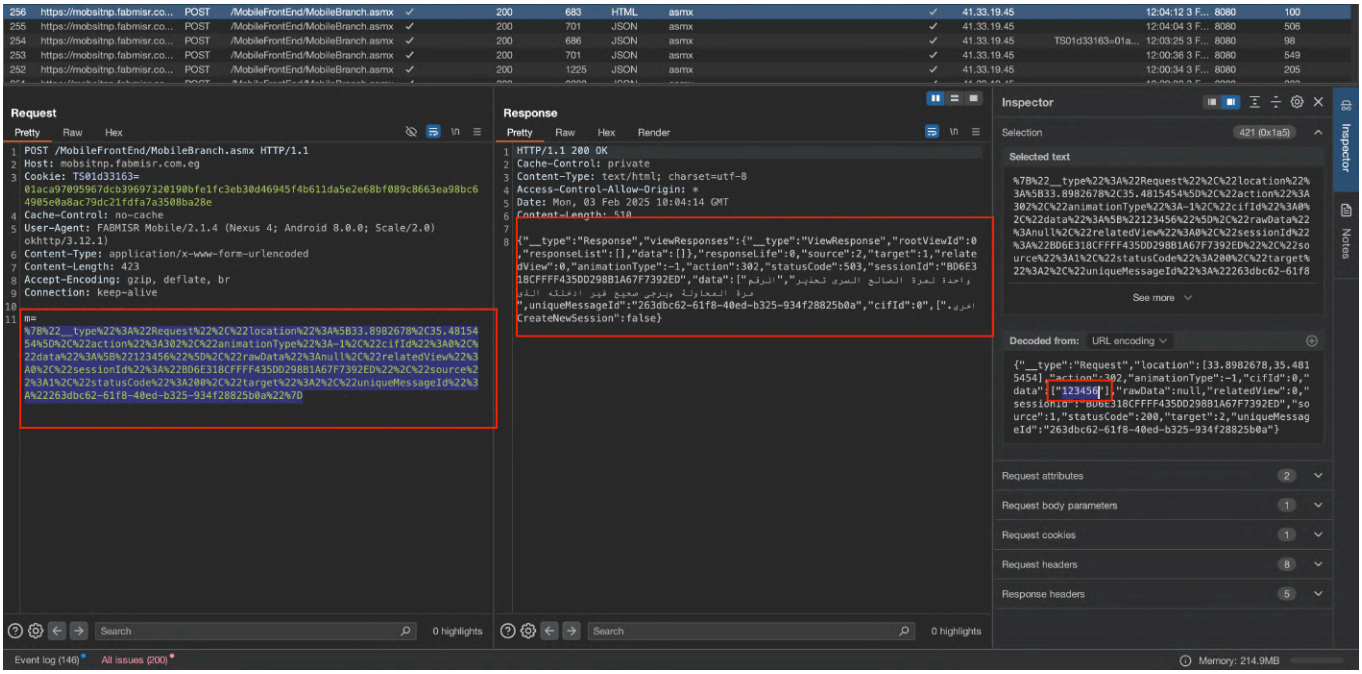


Figure1: Representation of the OTP number 123456 is not worked.

Details Of The Findings

No Rate Limiting on OTP Requests via Email

Severity: Medium

Status: Fixed

Affected Assets:
Web Application / Application Wide

Description:
Our team discovered that the application does not enforce rate limiting for OTP requests sent via email. This vulnerability allows an attacker or user to repeatedly request OTPs without restriction, potentially leading to abuse or denial of service. Without rate limits, attackers can overload the system or cause users to be spammed with OTPs, potentially leading to service disruption or increased operational costs.

Impact:
Denial of Service (DoS): Attackers can overload the system with excessive OTP requests, causing delays or disruptions for legitimate users.

Increased Costs: Excessive email OTP requests can result in unnecessary operational costs for email services, especially when using third-party providers.

User Frustration: Legitimate users may be inconvenienced or confused by receiving multiple OTP emails without being able to complete their actions.

Details:
Our team tested the OTP request functionality and discovered that there were no rate limiting controls on the number of requests sent via email. We were able to trigger multiple OTP requests in rapid succession, and the system continued to send OTP emails without any restriction or cooldown. This can lead to an unnecessary strain on email delivery systems and user inconvenience, and opens the door for potential abuse by attackers.

Reproduction Steps:
1- Navigate to the OTP request page via email on the application.
2- Enter a valid email address and request an OTP.
3- Trigger multiple OTP requests in a short period, either manually or via automation tools.
4- Observe that the application does not enforce any rate limit, allowing an unlimited number of OTP emails to be sent.
5- Confirm that after several OTP requests, there is no restriction or cooldown period applied.

Details Of The Findings

Remediation

1- Implement rate limiting for OTP requests to restrict the number of OTP emails a user can request within a defined time (e.g., no more than 3 requests per minute).
2- Use CAPTCHA or other challenge mechanisms to ensure that OTP requests are made by real users and not automated scripts.
3- Introduce a cooldown period after multiple failed OTP attempts, preventing further OTP requests until the period expires.

Proof Of Concept:

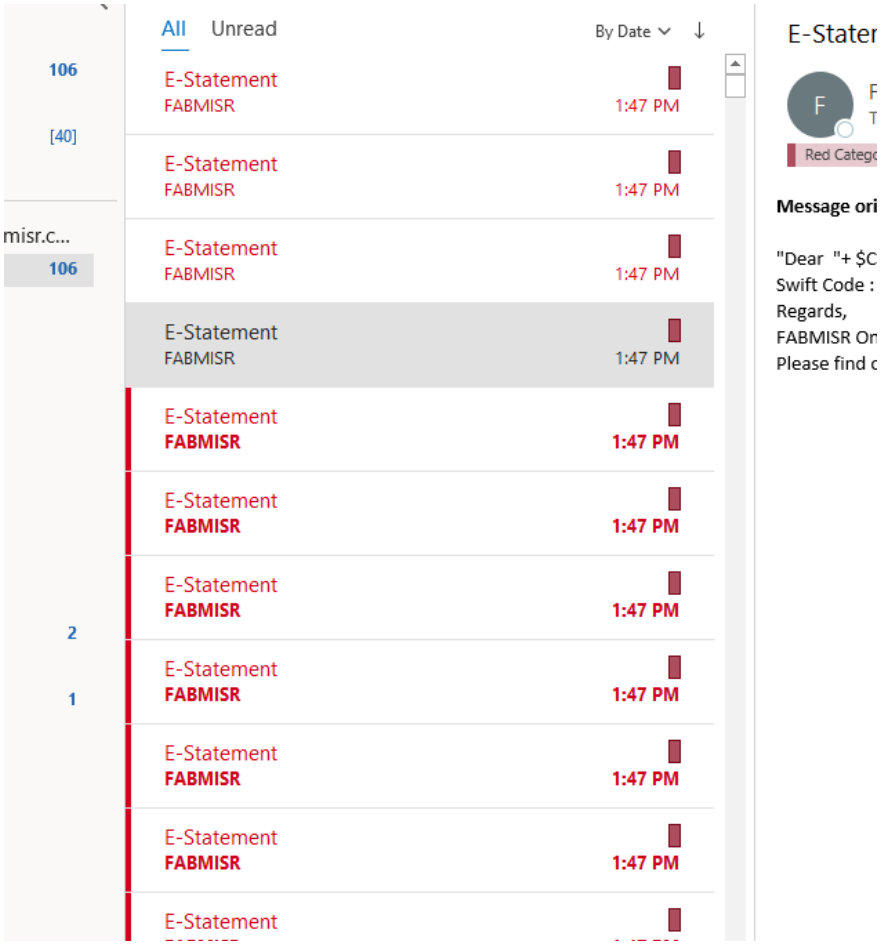


Figure 1: Representation of no rate limit on OTP requests via email.

Details Of The Findings

Proof Of Mitigation:

The vulnerability was mitigated by implementing rate limiting for OTP requests and redirecting users to the login page after three requests. Now, even if more than 50 requests are made, only 3 OTP emails are sent, and any further requests result in a redirection to the login page, preventing spam, abuse, and potential service disruption.

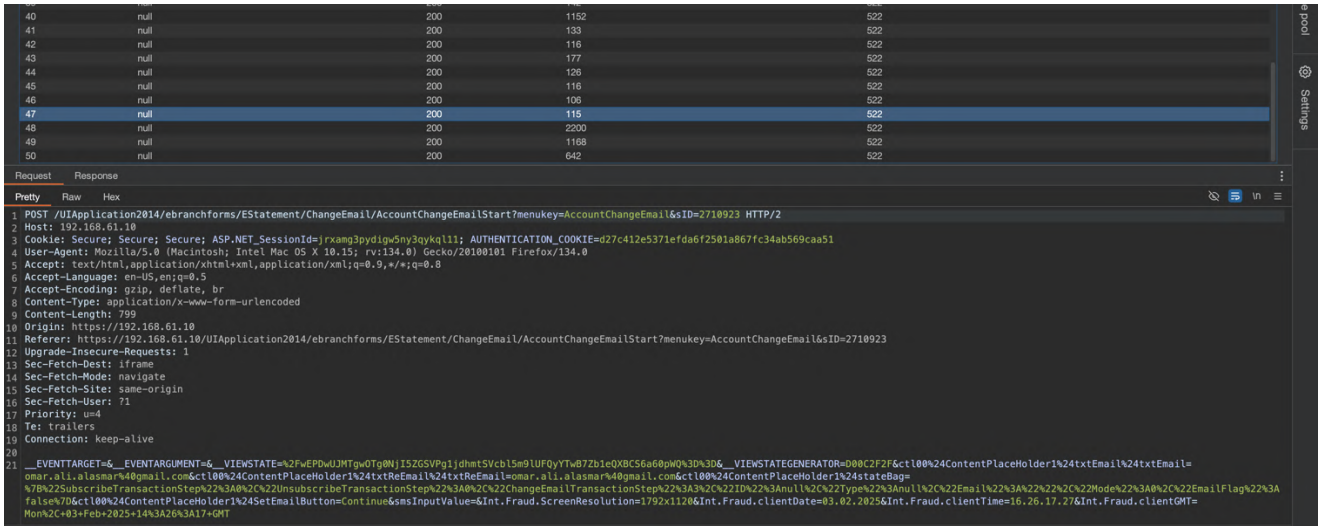


Figure 1: Representation requests for sending OTP via email.

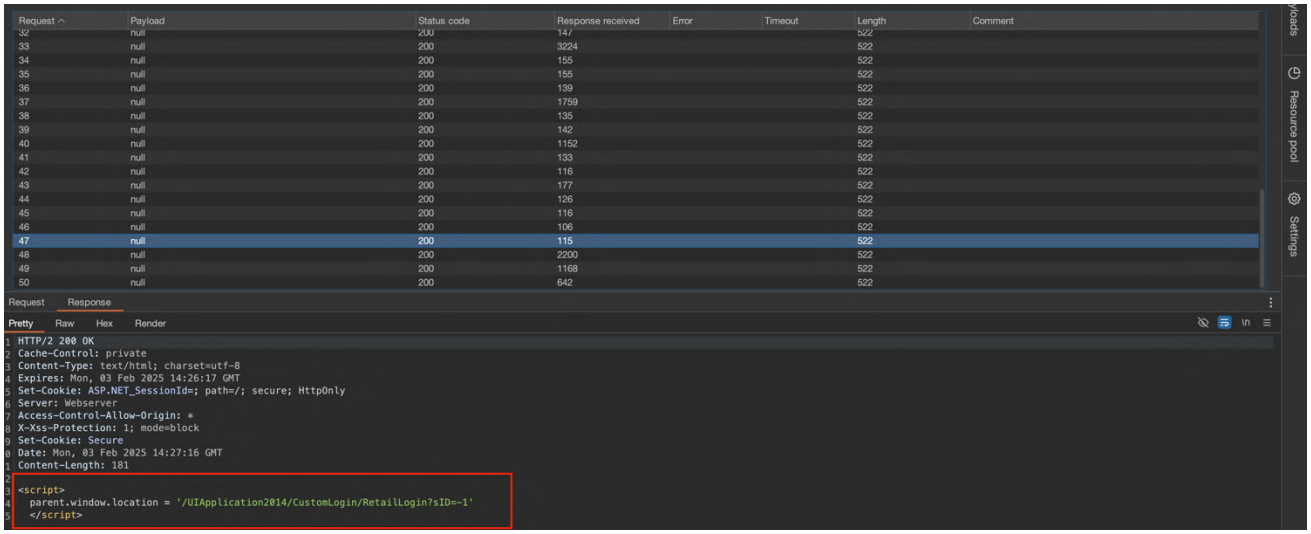


Figure 2: Representation redirecting to login page after sending multiple request.

Details Of The Findings

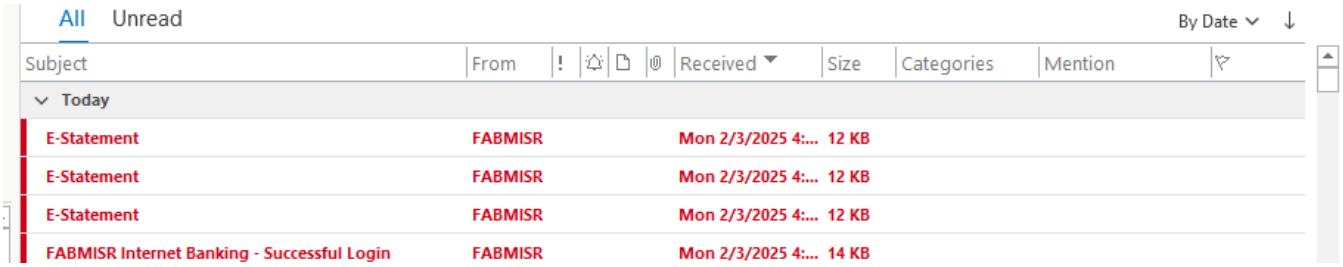


Figure 2: Representation of there is rate limit applying for OTP request via email.

Details Of The Findings

Bypassing Rate Limit on OTP Request via Email

Severity: Low

Status: Fixed

Affected Assets:
Web Application

Description:
Our team discovered a vulnerability in the OTP request functionality via email where the rate-limiting mechanism can be bypassed. After a user triggers the limit (e.g., three OTP requests), the user is blocked from making further OTP requests. However, an attacker can circumvent this block by saving the original OTP request in a tool like a repeater. Once the block is applied, the attacker can send the saved request once, and upon doing so, the block is removed, allowing them to resume making OTP requests. This cycle can be repeated multiple times to bypass the rate-limiting mechanism.

Impact:
Denial of Service (DoS): Attackers can bypass the rate limit and flood the system with OTP requests, causing disruptions for legitimate users.

Increased System Load: Overloading the system with excessive OTP requests can strain email services, leading to potential performance degradation.

Security Risk: The ability to bypass rate limits increases the attack surface, allowing attackers to target the OTP mechanism with brute-force or spam attacks.

Details:
Our team identified that after three OTP requests, the system blocks further requests for the user. However, we were able to bypass this rate limit by saving the request in a repeater tool. After the user is blocked, we sent the saved OTP request once, which removed the block, allowing us to continue requesting OTPs. By repeating this cycle, we could bypass the rate-limiting mechanism indefinitely. This vulnerability presents a significant risk as attackers could exploit it to flood the system with OTP requests, potentially leading to denial-of-service conditions or increased operational costs.

Reproduction Steps:
1- Navigate to the OTP request page via email and trigger an OTP request.
2- Perform this action three times to trigger the rate-limiting mechanism and block the user from making further requests.
3- Save the original OTP request in a repeater tool before the block is applied.

Details Of The Findings

4- Once the user is blocked, resend the saved request using the repeater tool.

5- Attempt to request an OTP again. You will find that the block is removed, and the user can request OTPs once again.

6- Repeat the process to confirm that the block can be bypassed multiple times.

Remediation:

1- Implement a more robust rate-limiting mechanism that accounts for multiple bypass methods, including repeated attempts using tools like a repeater.

2- Introduce IP-based rate limiting or CAPTCHA challenges for OTP requests after the rate limit is exceeded.

3- Add session-based or user-account-based rate limiting to prevent bypassing limits using different tools or sessions.

4- Monitor OTP request logs for abnormal patterns of repeated requests, and trigger alerts if suspicious activity is detected.

Proof Of Concept:

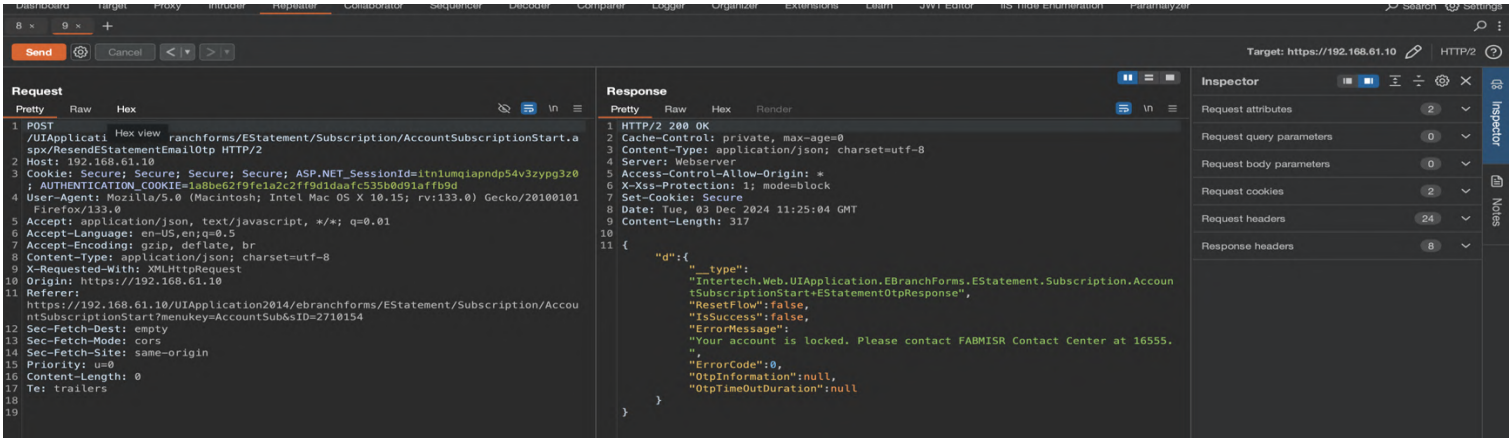


Figure 1: Representation of the OTP resend request being blocked after 3 attempts for the user.

Details Of The Findings

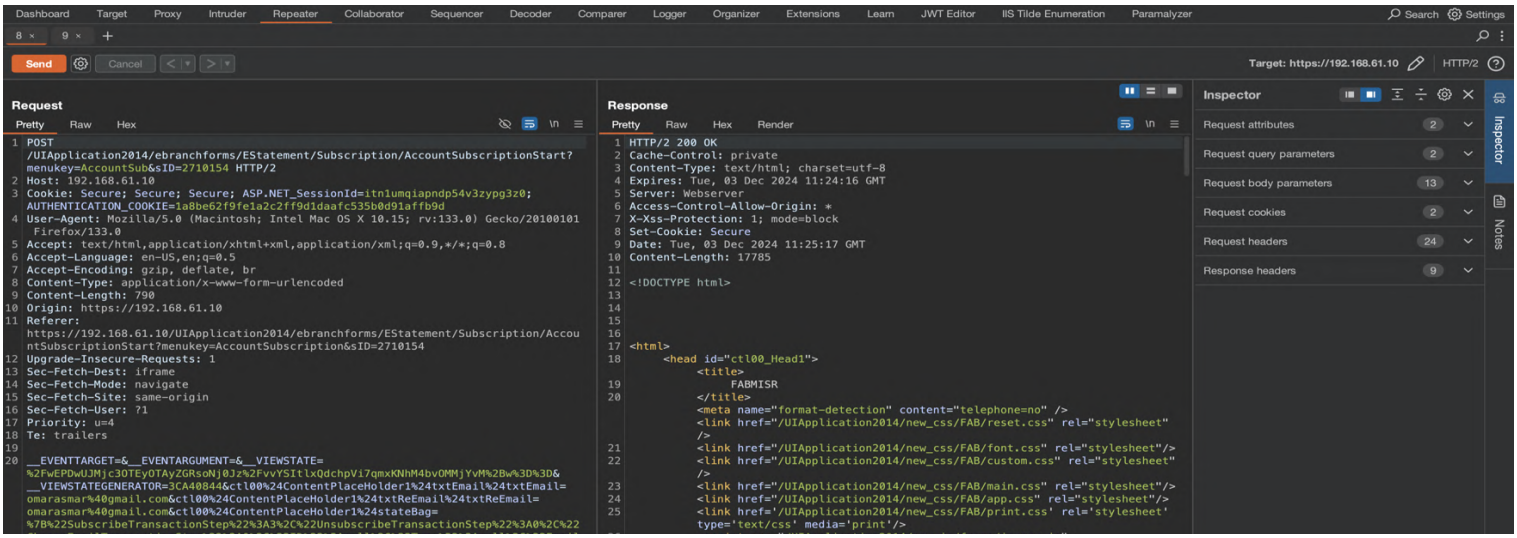


Figure 2: Representation of sending an OTP request to remove the block for resending OTP for the user.

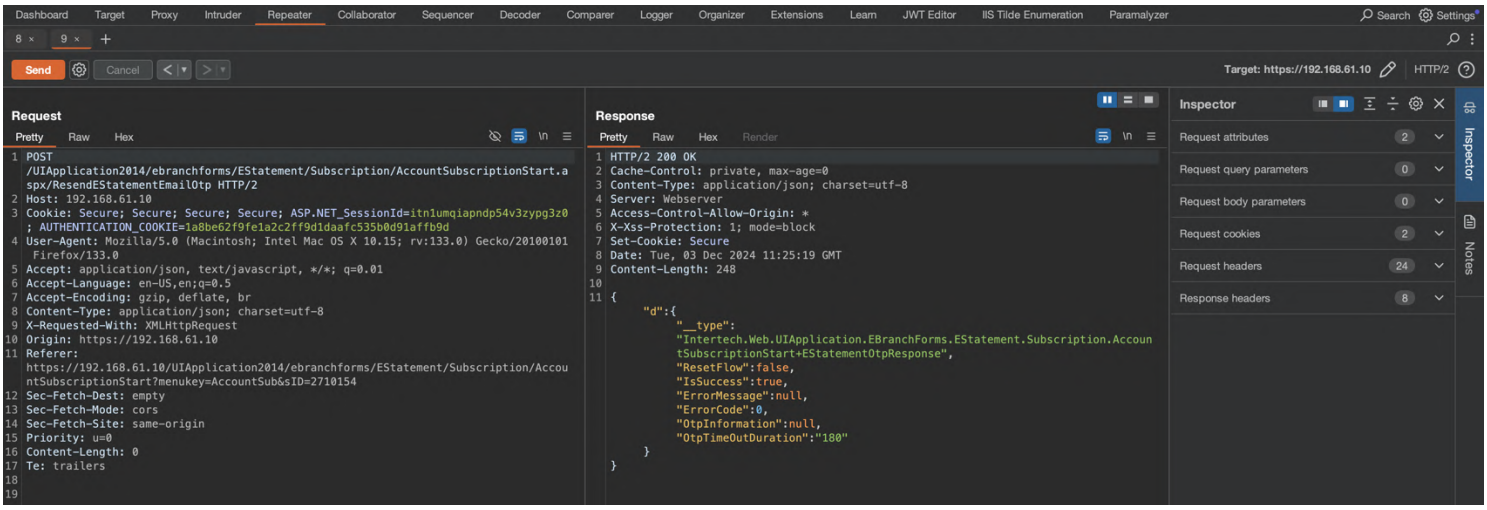


Figure 3: Representation of the user being able to make a resend request after the block was removed.

Details Of The Findings

Proof Of Mitigation:

The vulnerability was mitigated by redirecting users to the login page after three OTP requests. If an attacker attempts to bypass the rate limit by resending saved requests, they are consistently redirected to the login page, ensuring no additional OTP emails are sent beyond the three-request limit.

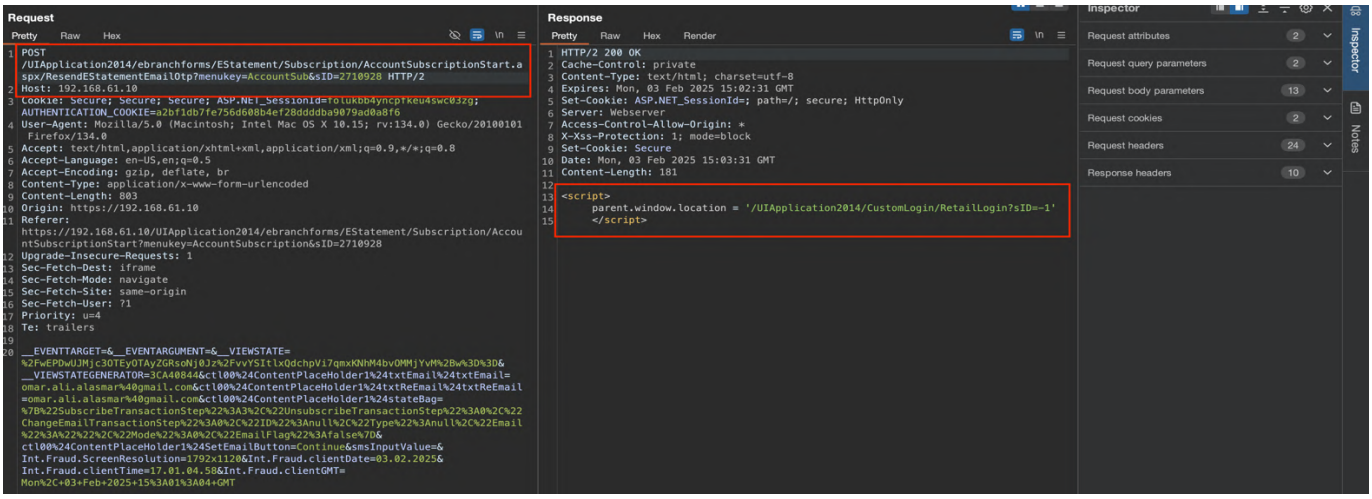


Figure 1: Representation of the OTP resend request being rediricet to login page after 3 times.

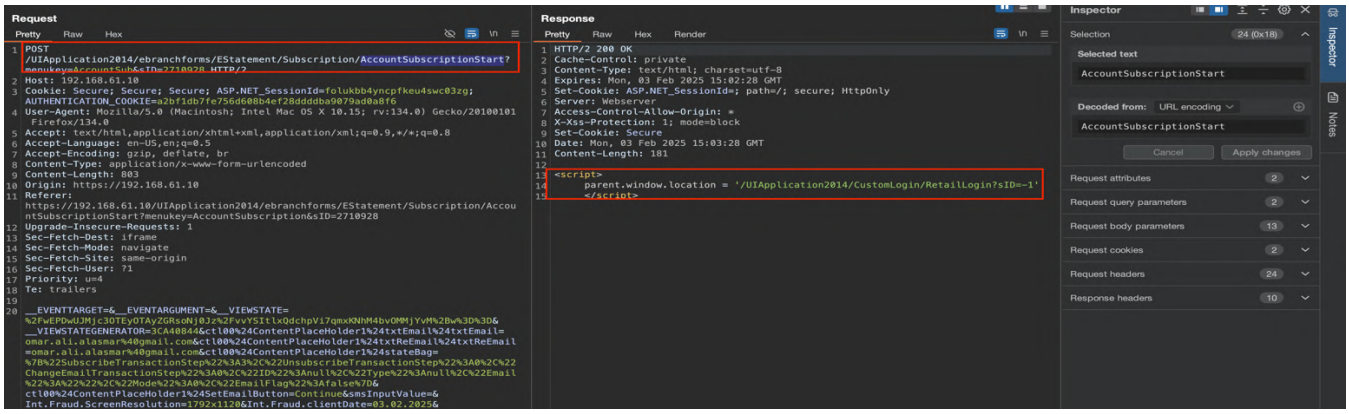


Figure 2: Representation of sending an OTP request to rediricet to login page also.

Details Of The Findings

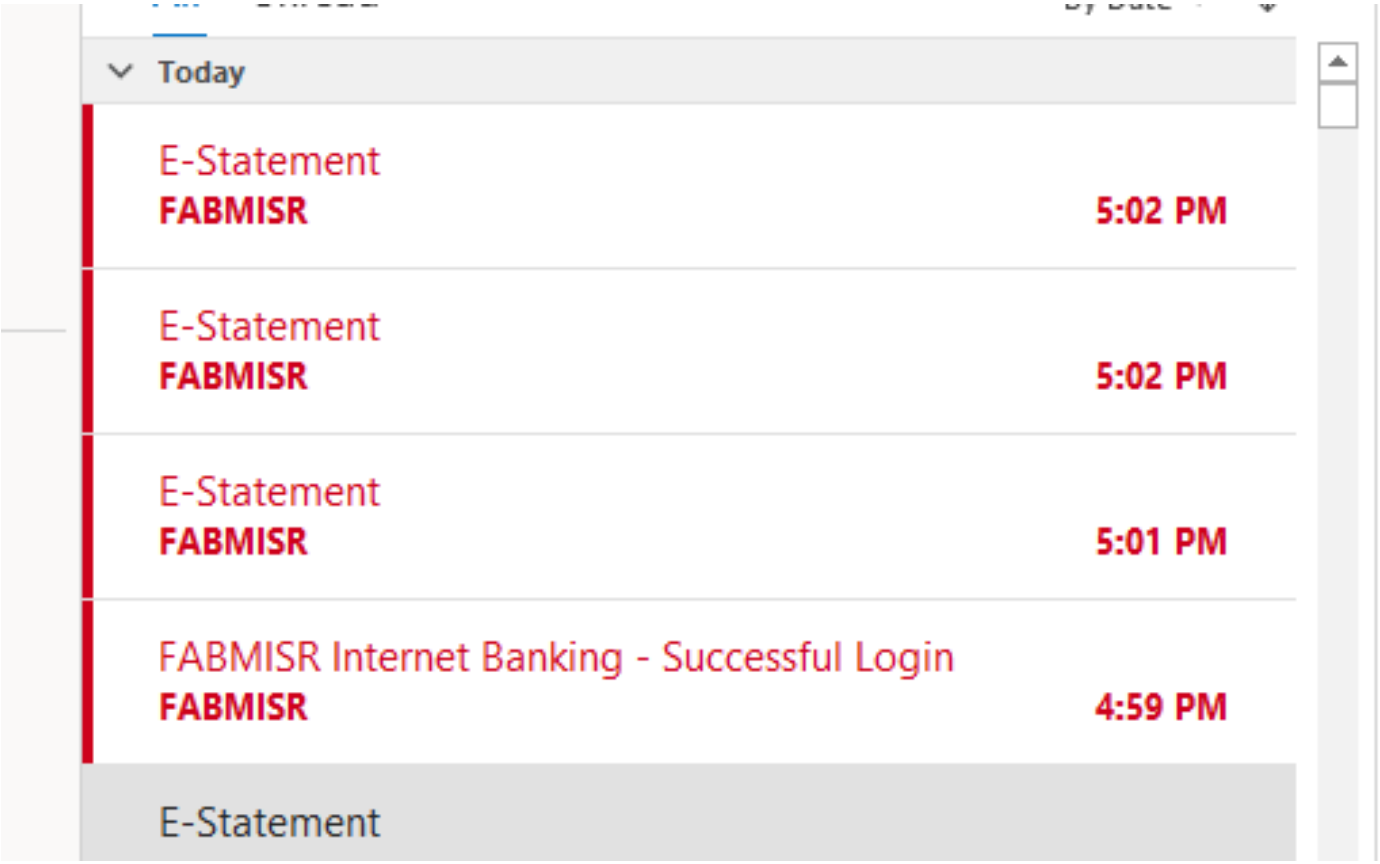


Figure 3: Representation of there is rate limit applying for OTP request via email.

Confidentiality

This document contains company confidential information of a proprietary and sensitive nature. As such this document should be afforded the security and handling precautions that a confidential document warrant.

This document should have a controlled distribution to relevant parties only, and should not be copied without written permission