Abstract –this document provides a comprehensive analysis of critical security vulnerabilities in Mobile Device Management (MDM) solutions. The analysis covers various aspects of these vulnerabilities, including their technical details, potential attack vectors, and implications for security professionals and organizations across different industries.

The analysis provides a high-quality summary of these vulnerabilities, offering valuable insights for security professionals, IT administrators, and other specialists. By understanding these vulnerabilities and their implications, organizations can better protect their MDM solutions, enhance their security posture, and mitigate the risks associated with these flaws. This document serves as a crucial resource for those looking to safeguard their mobile device management systems against sophisticated cyber threats.

I. MobileIron MDM

A security vulnerability in the MobileIron MDM solution exposes account enumeration and single-factor authentication (SFA) to unauthenticated attacks. The analysis reveals that a static key within the MobileIron MDM can be exploited to enumerate user accounts, potentially leading to unauthorized access.

A. Hardcoded Mobile@Work Encryption Key:

The Mobile@Work agent uses a hardcoded API key, which can be extracted by an unauthenticated attacker to discover an organization's MobileIron authentication endpoint. The risk level is low and suggested mitigation is disable MobileIron discovery services to reduce the attack vector.

- The Mobile@Work agent uses a hardcoded encryption key for the authentication process, which could allow an attacker to construct MobileIron authentication requests and potentially capture account credentials via man-in-the-middle (MitM) attacks.
- MobileIron acknowledges this issue but considers the attack vector minimal due to the multi-layered encryption strategy through TLS.

1) MobileIron's Response to Security Concerns:

- MobileIron recommends configuring their Core products using multi-factor authentication and configuring Mutual Certificate Authentication to secure device check-ins.
- They refute several findings in a report by Optiv, suggesting that the tested MobileIron Core server wasn't properly configured.

2) CVE-2021-3391:

- This vulnerability allows attackers to distinguish among valid, disabled, and nonexistent user accounts by observing the number of failed login attempts needed to produce a Lockout error message.

3) General Security Practices and Vulnerabilities:

- Hardcoding access keys in mobile apps/APIs is not considered secure, as anything embedded in a client is fully accessible to users, making it possible to reproduce any request the app makes.
- The use of hardcoded cryptographic keys is a common vulnerability, allowing attackers to decrypt encrypted configuration files and retrieve sensitive information.

4) MobileIron Platform Security:

- The MobileIron Platform provides security functions such as security audit, cryptographic support, identification and authentication, security management, and protection of the TSF.
- It uses TLS to secure communication channels between itself and mobile device users.

5) CVE-2020-35138:

- The MobileIron agents for Android and iOS contain a hardcoded encryption key used to encrypt the submission of username/password details during the authentication process.

6) MobileIron MDM Static Key Allowing Account Enumeration:

- A hardcoded API key in the Mobile@Work agent allows for account enumeration, demonstrating a security vulnerability.

7) Security Vulnerabilities in MobileIron:

- Various security vulnerabilities have been identified in MobileIron agents for Android and iOS, including the use of a hardcoded API key and encryption key.

8) MobileIron CVE2020-15505 Vulnerability:

- This vulnerability has been targeted by APT nation-state groups and cybercriminals to compromise organizations.

9) Norwegian Government Hacked with MobileIron Zero-Day:

- The Norwegian government was hacked using a MobileIron zero-day vulnerability, highlighting the critical nature of securing MobileIron environments against known vulnerabilities.

B. Hardcoded Mobile@Work API Key

The Mobile@Work agent uses a hardcoded encryption key, allowing an unauthenticated attacker to construct MobileIron authentication requests. This can also enable a well-positioned attacker to capture account credentials via man-in-the-middle (MitM) attacks. The risk level is medium and there is no known

mitigation exists. It would require MobileIron to remove the encryption functionality or eliminate the hardcoded nature of the encryption key.

1) *MobileIron MDM Registration Process:*
- Users launch the Mobile@Work app and provide their email address or MobileIron MDM environment endpoint.
- Submitting an email initiate a discovery process against a MobileIron hosted API to identify the authentication endpoint.

2) *API Discovery Request:*
- The API request requires two values:
  - key: The MobileIron API key to authorize requests.
  - domain: The registered FQDN of the user's email address.
- Requests without the proper API key receive an HTTP 403 error.

3) *Hardcoded API Key in Mobile@Work Agent:*
- The MobileIron API key is hardcoded in the Mobile@Work agent application.
- The decompiled the Android APK file to obtain the original Java source code.
- The hardcoded API key was found in the sources/com/mobileiron/registration/RegisterActivity.java file.

4) *Impact of Hardcoded Key:*
- Recovering this API key allows any unauthenticated attacker to locate an organization's MobileIron authentication endpoint.
- The successful API discovery request using the hardcoded key.

5) *MobileIron's Response:*
- MobileIron acknowledged the issue and identified the functionality as critical to the Mobile@Work workflow.
- They are reviewing alternative solutions, but there is currently no timeline for remediation.
- Validation of this attack surface CVE-2020-35137can be performed using a tool called Dauthi

C. Account Enumeration in MobileIron

The account authentication process allows external entities to enumerate user accounts and perform authentication attacks without triggering account lockout conditions. The risk level is medium and there is no known mitigation exists. Organizations can monitor the MobileIron endpoint for excessive authentication requests to gain situational awareness of malicious activity.

1) *Integration with Active Directory (AD):*
- MobileIron typically integrates with an environment's user-identity source, such as Microsoft Active Directory (AD), using LDAP filters to view the user repository.
- Registration of devices is not allowed for all visible users by default; accounts must be enabled in MobileIron before device registration is permitted.

2) *Authentication Responses:*

MobileIron provides various responses based on the MobileIron Protocol (MIPR) content:

- **Successful Authentication**: Results in a zLib compressed payload containing the MobileIron MDM profile with details like username, SenderGUID, UUID, and cookie value.
- **Failed Authentication**: Identified by a specific 0x1D message.
- Account Lockout: Triggered after a threshold of failed attempts, with a lockout duration of about 30 seconds.

3) *Ancillary Responses:*
- **Null Response**: Indicates a format or conditional input problem.
- **Device Unregistered**: Indicates a revoked or unregistered PIN authorized session.
- **Unknown Client ID**: Indicates an invalid or unknown SenderGUID.

4) *Interesting Items in Provisioning Packet:*
- **cookie**: Represents the authenticated and registered MDM session.
- **easV3Signature**: A Base64 encoded certificate for mutual certificate authentication.
- **easi**: An HTTP client authorization header.
- **rsn**: Device UUID value used as a primary key for MDM registration.
- **senderGUID**: Numeric ID for the authenticated and registered MDM session.
- **userID / username**: Indicates the username associated with the authentication session.

5) *Failed Authentication and Lockout:*
- MobileIron has a lockout threshold of about five failed attempts, which is a local condition and does not affect the upstream AD.
- The lockout event is indicated by a specific 0x1D response.

6) *User Enumeration:*
- **Invalid Account**: No lockout condition occurs, indicating the username is invalid.
- **Disabled/Locked AD Account**: The first attempt fails, and the second attempt results in a lockout response.
- **Valid Account**: Five failed attempts before a lockout condition occurs.
- **Validation** of this attack surface CVE-2021–3391 и CWE-204 can be performed using a tool called Dauthi.

D. Mitigation Strategies

1) *PIN-Based Authentication*
- PIN authentication can use a single PIN value or require PIN + user credentials.
- PINs are 6-digit single-use values tied to a single account.
- However, PIN authentication requests are not throttled, allowing brute-forcing of valid PINs.
- PIN-based registration prevents the user enumeration technique

- Therefore, PIN-based authentication successfully reduces the MobileIron attack surface.

2) *Mutual Certificate Authentication*

- it has no effect on the client/server communication channel.
- It only enables TLS validation checks of the Mobile@Work agent.
- Based on this, mutual certificate authentication does not mitigate the attack surface.

3) *Additional Best Practices*

- **Strong Corporate Password Policy**:
  - Implement min 12-character password length policy.
  - Combine with blocklisting common password patterns like "SeasonYear".
- **Limit User Device Registration**:
  - Allow only a single UUID and/or validated UUID values for device registration.
  - Prevents attacker from registering malicious device with compromised account.
- **Monitor MDM Authentication Requests**:
  - Monitor MobileIron connector service and logs for malicious activity.
  - Look for excessive authentication attempts indicating brute-forcing.