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. AirWatch MDM

A method to bypass multi-factor authentication (MFA) in VMware's AirWatch MDM services by exploiting the AirWatch configuration to bypass the Intelligence HUB application and directly register users, effectively circumventing MFA protections. MFA Implementation in AirWatch

MFA is only enforced during the initial registration process of the MDM client. This means that once a device is registered, subsequent authentication processes do not require MFA, leaving them vulnerable to Single-Factor Authentication (SFA) attacks.

A. Key Points on MFA Implementation

- **Initial Registration**: During the initial registration of a device with AirWatch, MFA is enforced. This typically involves the user providing a password and a second form of authentication, such as a one-time passcode (OTP) sent to their mobile device or email. This step ensures that the device is securely registered to the MDM system, reducing the risk of unauthorized devices being added.
- **Post-Registration Vulnerability**: After the device is registered, subsequent authentication processes revert to SFA. This means that users only need to provide a single form of authentication, usually a password, to access the MDM services. The lack of continuous MFA enforcement creates a significant security gap, as it allows attackers to exploit the system if they manage to obtain user credentials.
- **SFA Attack Surface**: the vulnerabilities associated with the SFA attack surface in AirWatch can be exploited both prior to and in parallel with the registration process, potentially compromising user credentials or registering malicious devices.

B. Initial Registration Process

- **MFA Enforcement**: During the initial registration of a device with AirWatch, MFA is enforced. This typically involves the user providing a password and a second form of authentication, such as a one-time passcode (OTP) sent to their mobile device or email. This step ensures that the device is securely registered to the MDM system, reducing the risk of unauthorized devices being added.
- **Secure Registration**: The initial registration process is designed to verify the identity of the user and the device. By requiring multiple forms of authentication, AirWatch aims to ensure that only legitimate users and devices can complete the registration process. This process includes submitting an email address or server endpoint, which triggers a discovery request to locate the appropriate MDM endpoint and obtain necessary configuration details.

C. Post-Registration Vulnerability

- **Reversion to SFA**: After the device is registered, subsequent authentication processes revert to SFA. This means that users only need to provide a single form of authentication, usually a password, to access the MDM services. The lack of continuous MFA enforcement creates a significant security gap, as it allows attackers to exploit the system if they manage to obtain user credentials.
- **Authentication Endpoints**: The specific API endpoints are vulnerable to SFA attacks. These endpoints allow for password attacks and limited user enumeration of valid domain accounts.
- **CAPTCHA and MFA Limitations**: Although CAPTCHA enforcement and MFA are applied during the user registration process, these protections are not extended to subsequent authentication attempts. Many of the API functions within AirWatch rely on SFA, and previous registration of a user is not required to establish an SFA attack surface. All the vulnerable API endpoints are publicly accessible to an unauthenticated attacker, allowing for password attacks.

D. SFA Attack Surface in AirWatch MDM

1) Key Points on SFA Attack Surface

- **Discovery Process Exploitation**: The AirWatch MDM client initiates a discovery process to locate the appropriate MDM endpoint. This involves sending a request to a discovery service, which returns the authentication endpoint and GroupID (ActivationCode) associated with the requested domain. The GroupID is considered public information and is not protected, making it easier for attackers to obtain this critical piece of information.
- **API Endpoints Vulnerability**: Several API endpoints are identified as vulnerable to SFA attacks. These endpoints allow for password attacks and limited user enumeration of valid domain accounts. For example, the /deviceservices/enrollment/airwatchenroll.aws/validatelogincredentials and /deviceservices/authenticationendpoint.aws endpoints can be abused to perform SFA attacks.
- **User Enumeration**: The ability to enumerate users is dependent on the authentication integrator and configuration of the endpoint. If functional, user enumeration would allow attackers to identify valid usernames, making it easier to target specific accounts.
- **Bypassing CAPTCHA and MFA:** CAPTCHA enforcement and MFA are only applied during the user registration process. Once registration is complete, SFA is solely leveraged, leaving the system vulnerable to brute-force attacks. Attackers can reset values such as the InternalIdentifier or Universal Device ID (UDID) and the active SessionID (SID) to bypass CAPTCHA protections and execute authentication attempts without identity protections.

2) Detailed Attack Scenarios

- **Discovery Service Exploitation**: Attackers can exploit the discovery service to obtain the authentication endpoint and GroupID. With this information, they can craft requests to the MDM server, potentially bypassing initial security checks.
- **SessionID (SID) Manipulation**: The SID value is easily recoverable by following the standard MDM registration process. Submitting a POST request to /DeviceManagement/Enrollment/EmailDiscovery returns a validation SID that can be used in the validateuserCredentials request. This SID value can then be passed as part of the parametrized request to validateuserCredentials, allowing attackers to bypass certain security checks.
- **GroupID Disclosure**: The GroupID value is critical for performing any further attacks against the environment. All AirWatch authentication interfaces require the submission of a GroupID value; without this information, it would not be possible to carry out an authentication attack against the environment. Upon recovering the endpoint and GroupID values, AirWatch discloses the configuration settings of the MDM environment. This further allows an unauthenticated attacker to identify sub-groups, authentication integrations, and numerous additional configuration settings of the environment.
- **Boxer Application Vulnerability**: The Boxer application only supports SFA and is incapable of supporting MFA in its current form. This makes it a perpetual SFA attack interface that can be leveraged to bypass all MFA protections implemented within the AirWatch product suite. Both the Boxer registration and authentication API functions leverage the same API endpoint of authenticationendpoint.aws. The variable factor in this request is the Request Header Content-Type value. During the registration process, this value is populated as UTF-8, allowing for the submission of an XML formatted message body.

E. AirWatch Functionality: Containerized Access Management

AirWatch offers a robust framework for managing mobile devices across various operating systems and platforms. AirWatch's containerized access management functionality is a cornerstone of its MDM solution, offering secure, efficient, and flexible management of mobile devices in a corporate environment. By providing a secure workspace, supporting BYOD management, and integrating with enterprise systems, AirWatch ensures that organizations can leverage the benefits of mobility without compromising on security.

- **Secure Workspace**: AirWatch Container offers a secure workspace on personal devices, pushing a distinct boundary between corporate and personal data. This separation ensures that corporate resources are securely managed and accessed without intruding on the user's personal space.
- **BYOD Management**: The solution is particularly beneficial for Bring Your Own Device (BYOD) management. It allows businesses to distribute Workspace ONE UEM applications and internal applications to the AirWatch Container, enabling employees to use their mobile devices for work without compromising security.
- **Application-Level Security**: AirWatch utilizes a common Software Development Kit (SDK) framework to secure enterprise applications within the container. This framework ensures that applications are visible inside and outside the AirWatch Container but maintains strict security measures for enterprise applications through container passcodes and encryption.
- **Single Sign-On Authentication**: The platform supports seamless single sign-on authentication, allowing users to access corporate resources securely through an app tunnel VPN. This feature simplifies the login process for users while ensuring that access is securely managed.
- **Global Infrastructure Communication**: AirWatch's functionality extends through a localized or cloud-based appliance that communicates back to a global infrastructure maintained by AirWatch through the domain awmdm.com. This global reach ensures that

mobile users can establish secure communications with the corporate environment, regardless of their location.

- **Flexible Deployment**: AirWatch supports a hybrid device deployment model, adding functionality to the current deployment and allowing devices to adopt settings from the desired organization group. This flexibility is crucial for organizations with diverse device usage policies, including corporate-owned, employee-owned, and line-of-business models.
- **Security and Encryption**: The platform standardizes security and data loss prevention strategies across mobile devices. It enforces passcode/encryption within the AirWatch Container, preventing data leakage outside of the app. For iOS devices, it secures data with FIPS 140-2 encryption and supports biometric authentication methods like Touch ID or EyeVerify.
- **Integration with Enterprise Systems**: AirWatch integrates seamlessly with existing enterprise systems, maximizing current investments and extending those capabilities to mobile device management. This integration is vital for maintaining a cohesive security posture across all corporate IT assets.
- **Comprehensive Email and Content Management**: AirWatch provides comprehensive solutions for mobile email and content management, ensuring that corporate email infrastructures are secure and that sensitive content is protected in a corporate container. These functionalities are critical for maintaining the integrity of corporate data on mobile devices

F. Discovery Process in AirWatch MDM

The discovery process in AirWatch MDM solution is a critical initial step that allows the MDM client to locate the appropriate authentication endpoint and obtain necessary configuration details. This process is essential for the proper functioning of the MDM system, ensuring that devices can securely connect to the corporate network.

1) Steps in the Discovery Process

- **Client Initialization**: When a user launches the AirWatch MDM client on their device, they are prompted to enter either an email address or a server endpoint. This input is used to initiate the discovery process.
- **Sending the Discovery Request**: The client constructs an HTTP GET request to the AirWatch discovery service. This request is sent to a predefined URL, typically discovery.awmdm.com, and includes the domain associated with the user's email address.

```
http
GET
/autodiscovery/DeviceRegistry.aws/v2/domainlookup/domain/vmware.com
HTTP/1.1
Host: discovery.awmdm.com
User-Agent:
Agent/20.08.0.23/Android/11
Accept-Encoding: gzip, deflate
Connection: close
```

- **Authorization Header**: The request includes an authorization header and additional validation checks. However, none of this information is server-side validated, meaning the request header can be simplified significantly without losing functionality.

```
http
GET
/autodiscovery/awcredentials.aws/v2/domainlookup/domain/vmware.com
HTTP/1.1
Host: discovery.awmdm.com
User-Agent:
Agent/20.08.0.23/Android/11
Accept-Encoding: gzip, deflate
Connection: close
```

- **Server Response**: The discovery service processes the request and responds with a JSON payload that includes the authentication endpoint and the GroupID (ActivationCode) associated with the requested domain.

```
json
{
  "authenticationEndpoint":
"https://auth.awmdm.com",
  "groupID": "VMWprod"
}
```

- **GroupID (ActivationCode)**: The GroupID is a required value for authenticating against the AirWatch solution. It is used to associate a public device enrollment with the customer's organization. Despite its importance, VMware does not consider the GroupID to be sensitive information, as it is publicly accessible and not protected.

2) Security Implications

- **Public Accessibility**: The GroupID and authentication endpoint are considered public information. This means that anyone with knowledge of the domain can obtain these values, which could potentially be used in further attacks against the MDM environment.
- **Potential for Exploitation**: Attackers can exploit the discovery service to obtain the GroupID and authentication endpoint. With this information, they can craft requests to the MDM server, potentially bypassing initial security checks and gaining unauthorized access.
- **API Endpoints**: There are several API endpoints that are involved in the discovery process:
  - /autodiscovery/awcredentials.aws/v1/domainlookup/domain/
  - /autodiscovery/awcredentials.aws/v2/domainlookup/domain/
  - /DeviceManagement/Enrollment/validate-userCredentials

These endpoints return critical information that can be used to further exploit the MDM system.

- **SessionID (SID) Manipulation**: The SID value is easily recoverable by following the standard MDM registration process. Submitting a POST request to /DeviceManagement/Enrollment/EmailDiscovery returns a validation SID that can be used in subsequent requests to validate user credentials.

G. Discovery Service Exploitation in AirWatch MDM

The discovery service in AirWatch MDM solution is a critical component that allows the MDM client to locate the appropriate authentication endpoint and obtain necessary configuration details, such as the GroupID. However, this service can be exploited by attackers to gain information that is essential for further attacks against the environment.

1) How the Discovery Service Works

- **Client Initialization**: When a user launches the AirWatch MDM client on their device, they are prompted to enter either an email address or a server endpoint. This input is used to initiate the discovery process.
- **Sending the Discovery Request**: The client constructs an HTTP GET request to the AirWatch discovery service. This request is sent to a predefined URL, typically discovery.awmdm.com, and includes the domain associated with the user's email address.

```
http
GET
/autodiscovery/DeviceRegistry.aws/v2/d
omainlookup/domain/vmware.com
HTTP/1.1
Host: discovery.awmdm.com
User-Agent:
Agent/20.08.0.23/Android/11
Accept-Encoding: gzip, deflate
Connection: close
```

- **Server Response**: The discovery service processes the request and responds with a JSON payload that includes the authentication endpoint and the GroupID (ActivationCode) associated with the requested domain.

```
json
{
  "authenticationEndpoint":
"https://auth.awmdm.com",
  "groupID": "VMWprod"
}
```

2) Exploitation of the Discovery Service

- **Public Accessibility**: The GroupID and authentication endpoint are considered public information. This means that anyone with knowledge of the domain can obtain these values, which could potentially be used in further attacks against the MDM environment.
- **Simplified Requests**: The request header can be simplified significantly without losing functionality, as none of the information is server-side validated. This makes it easier for attackers to craft requests to the discovery service.

```
http
GET
/autodiscovery/awcredentials.aws/v2/do
mainlookup/domain/vmware.com
HTTP/1.1
Host: discovery.awmdm.com
User-Agent:
Agent/20.08.0.23/Android/11
Accept-Encoding: gzip, deflate
Connection: close
```

- **API Endpoints**: Several API endpoints are involved in the discovery process:
  - /autodiscovery/awcredentials.aws/v1/domainlookup/domain/
  - /autodiscovery/awcredentials.aws/v2/domainlookup/domain/
  - /DeviceManagement/Enrollment/validate-userCredentials

These endpoints return critical information that can be used to further exploit the MDM system.

- **SessionID (SID) Manipulation**: The SID value is easily recoverable by following the standard MDM registration process. Submitting a POST request to /DeviceManagement/Enrollment/EmailDiscovery returns a validation SID that can be used in subsequent requests to validate user credentials.

3) Implications of Discovery Service Exploitation

By exploiting the discovery service, attackers can obtain the GroupID and authentication endpoint, which are essential for performing further attacks against the MDM environment. This information allows attackers to craft requests that can bypass initial security checks.

- **Configuration Disclosure**: Upon recovering the endpoint and GroupID values, AirWatch discloses the configuration settings of the MDM environment. This further allows an unauthenticated attacker to identify sub-groups, authentication integrations, and numerous additional configuration settings of the environment.
- **User Enumeration**: The ability to enumerate users is dependent on the authentication integrator and configuration of the endpoint. If functional, user enumeration would allow attackers to identify valid usernames, making it easier to target specific accounts.
- **Potential for Further Attacks**: With the GroupID and authentication endpoint, attackers can perform Single-Factor Authentication (SFA) attacks, potentially compromising user credentials or registering malicious devices. This can lead to unauthorized access to corporate resources and sensitive data.

H. Configuration Disclosure in AirWatch MDM

By exploiting the discovery service to recover endpoint and GroupID values, attackers can disclose detailed configuration settings of the MDM environment. This section provides a detailed explanation of how this exploitation works and its implications.

1) How Configuration Disclosure Works

- **Discovery Service Exploitation**: Attackers can exploit the discovery service to obtain the authentication endpoint and GroupID. These values are essential for performing further attacks against the MDM environment. The GroupID, or ActivationCode, is a required value when attempting to authenticate against the AirWatch solution. Although VMware does not consider this information sensitive, it is critical for further exploitation.
- **API Endpoints**: Several API endpoints are involved in the discovery process and can be exploited to obtain configuration details:
  - /autodiscovery/awcredentials.aws/v1/domainlookup/domain/
  - /autodiscovery/awcredentials.aws/v2/domainlookup/domain/
  - /DeviceManagement/Enrollment/validate-userCredentials

These endpoints return critical information that can be used to further exploit the MDM system.

- **SessionID (SID) Manipulation**: The SID value is easily recoverable by following the standard MDM registration process. Submitting a POST request to /DeviceManagement/Enrollment/EmailDiscovery returns a validation SID that can be used in subsequent requests to validate user credentials.
- **GroupID Disclosure**: Upon recovering the endpoint and GroupID values, attackers can use the /deviceservices/enrollment/airwatchenroll.aws/validategroupidentifier endpoint to disclose configuration settings of the MDM environment.

```
http
POST /deviceservices/enrollment/airwatchenroll.aws/validategroupidentifier HTTP/1.1
Host: vmware.awmdm.com
User-Agent: Agent/20.08.0.23/Android/11
Content-Length: 118
Content-Type: application/json
Accept-Encoding: gzip, deflate
Connection: close

{"Header":{"SessionId":"00000000-0000-0000-0000-000000000000"},"Device":{"InternalIdentifier":""},"GroupId":"VMWprod"}
```

2) Implications of Configuration Disclosure

- **Detailed Configuration Information**: The response from the validategroupidentifier endpoint includes detailed configuration settings of the MDM environment. This information can be used by attackers to understand the structure and configuration of the MDM system.

```
json
{
  "Header": {
    "ProtocolRevision": 0,
    "Language": null,
    "SessionId": "6debe689-709d-4f6a-b038-fe5f61fde336",
    "Mode": 2,
    "AgentToken": "978969b0-d2cf-4dd6-8e3c-eef546010b34",
    "ProtocolType": 0,
    "App": 0,
    "AppVersion": null
  },
  "Status": {
    "Code": 1,
    "Notification": ""
  },
  "NextStep": {
    "InstallUrl": "",
    "ServiceUrl": "https://vmware.awmdm.com/DeviceManagement/Enrollment/begin-samlAuthentication?sid=6debe689-709d-4f6a-b038-fe5f61fde336",
    "DeviceUserMode": 0,
    "StagingRequired": false,
    "DisplayStagingMessage": null,
    "UserIdentifier": null,
    "AfwProvisioningMode": 0,
    "RegistrationTypePo": 0,
    "RegistrationTypeDo": 0,
    "VidmForCico": false,
    "IsLbusEnabled": false,
    "ClosedNetworkEnrollment": false,
    "Type": 18,
    "SettingsPayload": "",
    "AgentSettings": null,
    "RequireServicesFromStore": false,
    "IsCaptchaRequired": false,
    "CaptchaValue": null,
    "AndroidEnrollmentTarget": 0,
    "KnoxPlayForWorkCapable": false,
    "AndroidWorkTempPassword": null,
    "UserEmailAddress": null,
    "showEnrollmentInfoMessages": false,
    "AFWUserAuthToken": null,
    "AFWAccountIdentifer": null,
    "IsDeviceAfwCertified": false,
    "GreenBoxUrl": null,
    "VidmServerUrl": null,
    "IsVidmConfigured": false,
    "IsGreenBoxCatalogEnabled": false,
    "IsContainerModeEnabled": false,
    "ScepPayload": null,
    "BeaconConsoleSettingsServer": null,
    "CollectImeiNumber": false,
```

```
"IsCustomOnboardingExperienceEnable
d": false,
    "CustomOnboardingMessage": null,
    "CustomOnboardingUserName": null,
    "CustomOnboardingWelcomeText":
null
  }
 }
```

- **Identification of Sub-Groups**: The disclosed configuration settings include information about sub-groups within the MDM environment. This allows attackers to identify different organizational units and their associated settings.
- **Authentication Integrations**: The response also reveals details about authentication integrations, such as whether the environment is configured with AirWatch identity services, third-party authenticators, or SAML-based authentication.
- The Type value in the response indicates the authentication configuration:
  - **Type 1**: Environment's evaluation license has expired or is not active.
  - **Type 2**: Environment is configured with AirWatch identity services and supports SFA.
  - **Type 4**: Environment is not configured with a third-party authenticator and supports SFA.
  - **Type 8**: Environment requires token registration prior to user authentication.
  - **Type 18**: Environment has a SAML integration and requires MFA.
- **Service URLs**: The response includes service URLs for identity validation through third-party authenticator integrations. These URLs can be used by attackers to attempt authentication within the context of the integrator.
- **Potential for Further Attacks**: With the disclosed configuration information, attackers can craft more targeted attacks against the MDM environment. This includes performing Single-Factor Authentication (SFA) attacks, registering malicious devices, and potentially compromising user credentials.

I. API Endpoints Vulnerability in AirWatch MDM

The several specific API endpoints within VMware's AirWatch Mobile Device Management (MDM) product suite are vulnerable to Single-Factor Authentication (SFA) attacks that allow attackers to perform password attacks and limited user enumeration of valid domain accounts.

1) Key Vulnerable API Endpoints

- **Discovery Service Endpoints:** /autodiscovery/awcredentials.aws/v1/domainlookup/domain/ ,/autodiscovery/awcredentials.aws/v2/domainlookup/domain/. These endpoints are used during the discovery process to locate the appropriate MDM authentication endpoint and obtain the GroupID (ActivationCode). The responses from these endpoints include critical information such as the authentication endpoint and GroupID, which are essential for further attacks.
- **User Credentials Validation Endpoint:** /DeviceManagement/Enrollment/validate-userCredentials. This endpoint requires a SessionID (SID) for communication. The SID can be easily recovered by following the standard MDM registration process. Submitting a POST request to /DeviceManagement/Enrollment/EmailDiscovery returns a validation SID that can be used in the validate-userCredentials request.
- **Login Credentials Validation Endpoint:** /deviceservices/enrollment/airwatchenroll.aws/validatelogincredentials. This endpoint is used during the user registration process by the Intelligence HUB application to validate user credentials. Even if a third-party authentication is configured within the environment, this API is still provided as a communication interface.
- **Authentication Endpoint**: /deviceservices/authenticationendpoint.aws. This endpoint is leveraged within two functions for the Boxer email agent: registration and authentication. The Boxer application only supports SFA and is incapable of supporting MFA in its current form, making it a perpetual SFA attack interface.

2) Exploitation of Vulnerable API Endpoints

- **Discovery Service Exploitation**: Attackers can exploit the discovery service endpoints to obtain the authentication endpoint and GroupID. With this information, they can craft requests to the MDM server, potentially bypassing initial security checks and gaining unauthorized access.
- **SessionID (SID) Manipulation**: The SID value is easily recoverable by following the standard MDM registration process. Submitting a POST request to /DeviceManagement/Enrollment/EmailDiscovery returns a validation SID that can be used in subsequent requests to validate user credentials.
- **User Enumeration**: The ability to enumerate users is dependent on the authentication integrator and configuration of the endpoint. If functional, user enumeration would allow attackers to identify valid usernames, making it easier to target specific accounts.
- **Password Attacks**: The vulnerable API endpoints allow for password attacks, where attackers can systematically try different password combinations to gain access. The lack of continuous MFA enforcement makes it easier for attackers to succeed with these attacks.

3) Example API Requests and Responses

Discovery Request:

```
http
```

```
GET
/autodiscovery/awcredentials.aws/v2/do
mainlookup/domain/vmware.com
HTTP/1.1
Host: discovery.awmdm.com
User-Agent:
Agent/20.08.0.23/Android/11
Accept-Encoding: gzip, deflate
Connection: close
```

Discovery Response:
Example of a discovery response:

```
json
{
  "authenticationEndpoint":
"https://auth.awmdm.com",
  "groupID": "VMWprod"
}
```

Validate User Credentials Request:
Example of a validate-userCredentials request:

```
http
GET
/DeviceManagement/Enrollment/validate
-
userCredentials?groupid=True&welcom
e=False&id=1e69ee15-4749-44fe-8d91-
a67bf7fd971e HTTP/1.1
Host: vmware.awmdm.com
User-Agent:
Agent/20.08.0.23/Android/11
Content-Length: 0
Accept: gzip, deflate
Content-Type: application/x-www-form-
urlencoded
Accept-Encoding: gzip, deflate
Connection: close
```

Validate Login Credentials Request:
Example of a validatelogincredentials request:

```
json
{
  "Username": "test",
  "Password": "test",
  "Header": {
    "SessionId":
"f4e74df0-f22f-48f5-
9496-1d5b66526ed3"
  },
  "SamlCompleteUrl": "aw://",
  "Device": {
    "InternalIdentifier":
"3c411751c74c4f6cbceac8e39dd053d4c
226d78d"
  }
}
```

Authentication Endpoint Request:
Example of a Boxer authentication request:

```
json
{
  "ActivationCode": "aCode",
  "BundleId": "com.box.email",
  "Udid":
"409853f111044398a463119d878f34665
e23271f",
  "Username": "test",
  "AuthenticationType": "2",
  "RequestingApp": "com.boxer.email",
  "DeviceType": "2",
  "Password": "test",
  "AuthenticationGroup":
"com.air-watch.boxer"
}
```

J. *Mitigation "Integrating SAML/IDP/MFA"*

While the integration of SAML/IDP/MFA services during the user registration process in AirWatch MDM enhances security, the limitations of the Boxer application and the lack of continuous MFA enforcement pose significant vulnerabilities. Implementing continuous MFA enforcement and additional security measures can help mitigate these risks and protect the MDM environment from potential attacks.

1) *SAML/IDP/MFA Integration*

- **Purpose of SAML/IDP/MFA**: The integration of Security Assertion Markup Language (SAML), Identity Provider (IDP), and Multi-Factor Authentication (MFA) services aims to enhance the security of the authentication process. These services provide additional layers of verification, making it more difficult for attackers to gain unauthorized access.
- **User Registration Process**: VMware suggests that SAML/IDP/MFA integrations are primarily supported during the user registration process through the Intelligence HUB application. This means that when a new device or user is being registered, MFA can be enforced to ensure that the registration is secure. The Intelligence HUB application communicates with the AirWatch MDM server to validate user credentials and enforce MFA during this initial registration phase.
- **Configuration Steps**: To implement SAML/IDP services, administrators need to configure the Workspace ONE Access (formerly VMware Identity Manager) to act as the identity provider. This involves setting up SAML metadata, configuring identity provider settings, and enabling SAML authentication for both the admin console and the self-service portal.
  - Downloading the Identity Provider (IdP) metadata from Workspace ONE Access.
  - Uploading the IdP metadata to the AirWatch console under Directory Services settings.
  - Enabling SAML authentication for the admin console and self-service portal.
  - Configuring access policies to enforce MFA during the registration process.
- **Integration with Workspace ONE Access**: Workspace ONE Access provides a unified platform for managing identity and access across various VMware services, including AirWatch MDM. It supports integration with third-party identity providers and SSO solutions like Okta, allowing for a seamless authentication experience. The integration ensures that users can leverage existing identity management solutions to authenticate against the AirWatch MDM environment.

2) Limitations and Challenges

- **Boxer Application Limitation**: One of the significant limitations is that the Boxer application, which is part of the AirWatch MDM suite, only supports Single-Factor Authentication (SFA). This means that even if SAML/IDP/MFA services are implemented during the registration process, the Boxer application does not enforce MFA for subsequent authentications. The inability of the Boxer application to support MFA creates a persistent vulnerability, as attackers can exploit the SFA attack surface to gain unauthorized access.
- **Post-Registration Vulnerability**: while MFA is enforced during the initial registration process, subsequent authentication processes revert to SFA. This means that once a device is registered, users only need to provide a single form of authentication, typically a password, to access the MDM services. The lack of continuous MFA enforcement leaves the system vulnerable to credential compromise and unauthorized access.

K. Mitigation "Disabling Boxer Enrollment Services"

While disabling Boxer registration services is suggested as a potential mitigation step for vulnerabilities associated with SFA attacks in AirWatch MDM, its effectiveness is uncertain. Clarification from VMware and the implementation of comprehensive security measures are essential for effectively reducing the attack surface and enhancing the overall security of the MDM environment.

1) Understanding Boxer Registration Services

Boxer is an email client application that is part of the AirWatch MDM suite. It is designed to provide secure email access on mobile devices managed by AirWatch. The application supports Single-Factor Authentication (SFA) but does not inherently support Multi-Factor Authentication (MFA).

Boxer registration services are responsible for the initial setup and registration of the Boxer application on user devices. This process involves authenticating user credentials and linking the device with the user's email account within the AirWatch environment.

2) Disabling Boxer Registration Services

- **Suggested Mitigation**: VMware suggests that disabling Boxer registration services could potentially mitigate vulnerabilities associated with SFA attacks. This step would prevent new devices from registering with the Boxer application, potentially reducing the attack surface.
- **Uncertainty Regarding Authentication API Endpoint**: The authentication API endpoint is critical for the registration process and subsequent authentications. If this endpoint remains active, attackers could still exploit it for SFA attacks, even if Boxer registration services are disabled.
- **Attack Surface Considerations**: The effectiveness of this mitigation step depends on the extent to which it impacts the overall attack surface. If disabling Boxer registration services does not remove the authentication API endpoint, the attack surface may not be significantly reduced. Attackers could potentially find alternative ways to exploit the system, leveraging the active authentication API endpoint.

3) Implications and Recommendations

- **Comprehensive Security Measures**: Organizations should consider implementing comprehensive security measures beyond disabling Boxer registration services. This includes enforcing Multi-Factor Authentication (MFA) across all access points, regularly auditing security configurations, and educating users about security best practices.
- **Continuous Monitoring**: Continuous monitoring of the AirWatch environment and the Boxer application is essential. Organizations should monitor for unusual activity that may indicate attempted exploitation and respond promptly to potential security incidents.
- **Alternative Solutions**: Exploring alternative solutions or configurations that inherently support MFA for email access on mobile devices may provide a more robust security posture. This could involve using different email clients that fully integrate with MFA solutions or enhancing the security configurations within the AirWatch MDM suite.

L. Mitigation "Disabling Discovery Services"

Disabling discovery services in AirWatch MDM is a suggested mitigation step to prevent the public disclosure of the AirWatch endpoint and GroupID. However, this measure has limited value in preventing attacks since the GroupID value can be brute-forced. To effectively secure the AirWatch MDM environment, organizations need to implement comprehensive security measures, including continuous MFA enforcement, regular security audits, and user education. These steps can help mitigate the risks associated with the public accessibility of critical information and enhance the overall security posture of the MDM environment.

1) Understanding Discovery Services

- **Discovery Service Functionality**: The discovery service in AirWatch MDM is responsible for locating the appropriate MDM authentication endpoint and obtaining the GroupID (ActivationCode). This process is initiated when a user provides an email address or server endpoint during the initial setup of the AirWatch MDM client. The discovery request is sent to a predefined URL, typically discovery.awmdm.com, and includes the domain associated with the user's email address. The response from the discovery service includes the authentication endpoint and GroupID.
- **Public Accessibility**: The GroupID and authentication endpoint are considered public information. VMware does not classify this information as sensitive, which means it is accessible to anyone with knowledge of the domain. This public accessibility is a key factor in the potential exploitation of the discovery service.

2) Disabling Discovery Services

- **Suggested Mitigation**: Disabling discovery services is suggested as a mitigation step to prevent the public

disclosure of the AirWatch endpoint and GroupID. By disabling these services, organizations aim to reduce the risk of attackers obtaining critical information that could be used in further attacks.

- **Implementation**: Disabling discovery services involves configuring the AirWatch MDM environment to restrict or eliminate the use of the discovery service endpoints. This could be achieved through administrative settings or network-level controls to block access to the discovery service URLs.

3) Limitations of Disabling Discovery Services

- **Brute-Forcing GroupID**: disabling discovery services has limited value in preventing attacks because the GroupID value can be brute-forced. Attackers can systematically try different GroupID values until they find a valid one, bypassing the need for the discovery service. The GroupID is a relatively short and predictable value, making it feasible for attackers to use automated tools to brute-force valid GroupID values.
- **Critical Information for Further Attacks**: Even though the GroupID is not considered sensitive, it is critical for performing further attacks against the AirWatch environment. All AirWatch authentication interfaces require the submission of a GroupID value. Without this information, it would not be possible to carry out an authentication attack against the environment.
- **Publicly Accessible API Endpoints**: Many of the API endpoints within AirWatch MDM are publicly accessible and do not require prior authentication. This means that even if discovery services are disabled, attackers can still interact with these endpoints using brute-forced GroupID values.

4) Practical Implications

- **Limited Effectiveness**: Disabling discovery services alone is not sufficient to secure the AirWatch MDM environment. While it may add a layer of obscurity, it does not address the fundamental issue of the GroupID being brute-forceable and the public accessibility of critical API endpoints.
- **Comprehensive Security Measures**: To effectively mitigate the risks, organizations need to implement comprehensive security measures that go beyond disabling discovery services. This includes enforcing Multi-Factor Authentication (MFA) for all authentication processes, regularly auditing security configurations, and monitoring for unusual activity.
- **Continuous MFA Enforcement**: Ensuring that MFA is enforced not only during the initial registration process but also for subsequent authentications can significantly enhance security. This would make it more difficult for attackers to exploit brute-forced GroupID values.
- **User Education and Training**: Educating users about the importance of security practices, such as not sharing their GroupID and being cautious about phishing attempts, can help reduce the risk of exploitation.