HP's VAV testing focuses mostly on security and uses as its foundation the same recommendations for Android apps, which are described at https://developer.android.com/topic/security/#security-essentials-checklist. Please make sure these guidelines have been followed before submitting your app for VAV testing.

In addition, please complete the following HP checklist before submitting your app for VAV. The HP checklist is updated as HP's recommendations change. Since tests are added continuously, completing the HP checklist does not guarantee passing VAV.

- ☐ Use HTTPS for all network communication
- ☐ Use strong Crypto (See https://blog.devknox.io/best-practices-aes-encryption-in-android)
- ☐ Don't store Credentials in the App (See https://developers.hp.com/jetadvantage-link-device/managing-client-credentials)
- ☐ Use singleTask/singleInstance/taskAffinity/allowTaskReparenting very carefully (See https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-ren-chuangang.pdf)
- ☐ Use FLAG_SECURE on all sensitive Activities
- ☐ Set android:allowBackup to false
- ☐ Set android:debuggable to false
- ☐ Don't log any sensitive data and keep the log information to the minimum at the INFO level
- ☐ For the exported CP always verify the URI root. Better avoid returning fd
- ☐ Avoid exporting PreferenceActivity. If required, implement isValidFragment
- ☐ Set android:usesCleartextTraffic=false in Android manifest file and Networksecurityconfig.xml file
- ☐ Sign the application with the v2 signature scheme
- ☐ Avoid using insecure permissions
- ☐ Turn off long-press in all WebViews displayed
- ☐ Use androidx.webkit.WebViewAssetLoader to load file content securely
- ☐ Set android:useEmbeddedDex to true to avoid data tampering
- ☐ Be sure to target a supported Platform Version (See: https://developers.hp.com/workpath-sdk/platform-versions). Note: Targeting a Platform Version nearing end of support will be reported as a Medium issue.
- ☐ Do not make web service calls directly to printer-hosted web services (e.g. OXPd web services).

See the following pages for an example VAV report.

Vulnerability	Description	Severity	Recommended Actions
Clear text communication	*It was observed that the application uses clear text communication in the following files:* *XXXX*	*High*	*It is recommended to force TLS when transmitting sensitive information. Users should be prevented from accessing the page using http (port 80) using re-direction or by disabling access via port 80. Both NIST 800-52 and PCI DSS v3.1 strongly recommend upgrade to the latest version of TLS available, TLS 1.3. Or, at a minimum an upgrade to TLS 1.2.*
Clear text traffic is Enabled For App [android:usesCleartextTraffic=true]	*The app intends to use cleartext network traffic, such as cleartext HTTP, FTP stacks, DownloadManager, and MediaPlayer. The key reason for avoiding cleartext traffic is the lack of confidentiality, authenticity, and protections against tampering; a network attacker can eavesdrop on transmitted data and also modify it without being detected.*	*High*	*It is recommended to set android:usesCleartextTraffic="false" in AndroidManifest.xml.*
Clear text traffic is Enabled For App [cleartextTrafficPermitted="true"]	*The app intends to use cleartext network traffic, such as cleartext HTTP, FTP stacks, DownloadManager, and MediaPlayer. The key reason for avoiding cleartext traffic is the lack of confidentiality, authenticity, and protections against tampering; a network attacker can eavesdrop on transmitted data and also modify it without being detected.*	*High*	*It is recommended to set cleartextTrafficPermitted="false" in res\xml\network_security_config.xml*
Logging and Monitoring	*It was observed that the application logs sensitive information.* *File :* *XXXX*	*High*	*Applications must not log sensitive information, e.g., passwords, secret keys, tokens, etc. All sensitive logs must be encrypted.*
Janus Vulnerability	*It was observed that the application is signed only with v1 signature scheme. If an application is signed only with v1 signature scheme then app is vulnerable to Janus. Janus allows an attacker to modify the code in the app without affecting their signatures.*	*High*	*It is recommended to sign the application with v2 signature scheme.*
ECB Mode in Encryption Algorithm	*It was observed that the app uses ECB mode in Cryptographic encryption algorithm. ECB mode is known to be weak as it results in the same ciphertext for identical blocks of plaintext.* *File:* *XXXX*	*High*	*It is recommended to use CBC or GCM mode of encryption.*
Exposure of Sensitive Data	*It was observed that sensitive information like RSA Private Key and Certificate was exposed in the following file:* *XXXX*	*High*	*It is recommended not to expose or hardcode any sensitive information in the code.*
Insecure Permissions	*It was observed that the following insecure permissions were enabled:* *1) android.permission.MANAGE_ACCOUNTS* *2) android.permission.AUTHENTICATE_ACCOUNTS* *3) android.permission.USE_CREDENTIALS*	*High*	*It is recommended to disable all the insecure permissions.* *These permissions were removed in Api Level 23* *Reference* *1) https://developer.android.com/reference/android/accounts/AccountManager.html* *2) https://developer.android.com/sdk/api_diff/23/changes/android.Manifest.permission.html*
Weak Hash Algorithm	*It was observed that the application uses MD5 hash algorithm in the following files:* *XXXX* *MD5 is a weak hash known to have hash collisions.*	*High*	*It is recommended to implement strong hash algorithms like SHA512.*
ACRA Node Server detected	*It was observed that ACRA Node server is configured for crash reporting. If common username and password is used by all the printers to access the ACRA server, each printer can gain unauthorized access to the sensitive information in the crash reports of all the printers in the field.* *Also, only WRITE permission should be given on the ACRA server, so that only the crash reports are logged and cannot be read by unauthorized users.* *The server uses Basic authentication, which weakly encodes credentials using the Base64 algorithm, before transmitting them over the network. If access to traffic is gained, the traffic can be decoded and data stolen.*	*High*	*The following solution are recommended* *1) Use Firebase crashlytics from Google (https://firebase.google.com/docs/crashlytics/)* *2) Do not use common username and password by all the printers to access the ACRA server.* *3) Restrict access to the files on ACRA node server.* *4) Disable Basic authentication for the web server and do not use hardcoded credentials.*
Exposure of Sensitive Data	*It was observed that sensitive information like Client ID and Client Secret was exposed in the following file:* *XXXX*	*High*	*It is recommended not to expose or hardcode any sensitive information in the code.*
Insecure Implementation of SSL	*It was observed that hostname verification is disabled when making SSL connections. Trusting all the certificates or accepting self signed certificates is a critical Security Hole. This application is vulnerable to MITM attacks.*	*High*	*It is recommended not to use the setDefaultHostnameVerifier() function. The correct host name should be verified when making a SSL connection.*
Insecure Implementation of SSL (getUnsafeOkHttpClient)	*It was observed that the JA Link APIs use insecure communications protocols to interact with printer functionality. X509 certificate validation is disabled- getUnsafeOkHttpClient.*	*High*	*It is recommended to ensure all API to device communications are carried out using TLSv1.2 with strong certificate validation.*
Sensitive Information Hardcoded	*It was observed that sensitive information like API_KEY was hardcoded in the following file:* *XXXX*	*High*	*It is recommended not to hardcode any sensitive tokens and keys.*

Weak Hash Algorithm	*It was observed that the application is signed with SHA1withRSA. SHA1 hash algorithm is known to have collision issues.*	*High*	*It is recommended to sign the app with SHA256 hash algorithm.*
Long Press not disabled	*It was observed that a user could do a long press on a WebView to select text, which would show a web search button that allowed navigation to anywhere on the internet. Using this it was able to access everything on google and Play Store.*	*High*	*It is recommended to turn off long press in all WebViews displayed.*
Insecure SSL Implementation	*WebView ignores SSL Certificate errors and accept any SSL Certificate. This application is vulnerable to MITM attacks.*	*High*	*It is recommended to implement proper SSL certificate validation.*
Application Uses Basic Authenticatio	*It was observed that the application uses Basic authentication, which weakly encodes credentials using the Base64 algorithm. If access to traffic is gained, the traffic can be decoded and data stolen.* *File: XXXX*	*High*	*As the application uses Basic Authentication to share the static client secret, it can be easily extracted from the apps and allow others to impersonate the app.* *Following are the recommendations to fix this issue:* *1) Disable Basic Authentication* *2) Avoid using static client secrets in the application. Implement dynamic client authentication or native app authorization via External User-Agent.* *Reference:-* *https://tools.ietf.org/html/rfc8252* *https://tools.ietf.org/html/rfc7636* *https://tools.ietf.org/html/rfc7591*
Insecure WebView Implementation	*Execution of user controlled code in WebView is a critical Security Hole.* *File:* *XXXX*	*High*	*It is recommended not to set setJavaScriptEnabled(true) and not to use .addJavascriptInterface(' with webview.*
Broken authentication	*It was observed that the API method has username "guest" and blank password.*	*High*	*It is recommended to use X509 client certificates rather than HTTP authentication to* *authenticate requests. Or ensure that a secure method is used to establish a unique password for each user or device.*
App request root privileges	*It was observed that the app may request root (Super User) privileges 'eu.chainfire.supersu' in the following file:*	*High*	*It is recommended not to provide super user privileges to the application. If the file is not being used remove it from the code base. If the file is required please provide justification.*
Accessing content from local resources	*Application allows WebView to load local resources from the app data directory or external storage. It is possible for an attacker to perform File based XSS attacks.*	*High*	*It is recommended to use androidx.webkit.WebViewAssetLoader to load file content securely.*
Insecure Storage Mechanism	*It was observed that the application stores password in database in the clear text.* *Files:*	*High*	*It is recommended to use Android Keystore to store any sensitive information like passwords, tokens, etc.* *Or use EncryptedSharedPreferences. As EncryptedSharedPreferences uses AES 256 for encryption and the master key must be stored in Android Keystore.*
Bypass Certificate Pinning	*Base config is configured to bypass certificate pinning.*	*High*	*An app trusts all pre-installed CAs. If any of these CAs were to issue a fraudulent certificate, the app would be at risk from a man-in-the-middle attack. Some apps choose to limit the set of certificates they accept by either limiting the set of CAs they trust or by certificate pinning. Certificate pinning is done by providing a set of certificates by hash of the public key. A certificate chain is then valid only if the certificate chain contains at least one of the pinned public keys.* *Reference:* *https://developer.android.com/training/articles/security-config*
Custom trust anchors - user	*Base config is configured to trust user installed certificates.*	*High*	*It is recommended to limit the set of trusted CAs. An app that does not want to trust all CAs trusted by user can instead specify its own reduced set of CAs to trust. This protects the app from fraudulent certificates issued by any of the other CAs.This reduces the possibility of MITM attacks.* *Reference:* *https://developer.android.com/training/articles/security-config*
Weak RSA Algorithm	*Application is using PKCS1Padding which is vulnerable to padding oracle attack.* *File:*	*High*	*For RSA encryption algorithm, the recommended padding scheme is Optimal Asymmetric Encryption Padding (OAEP).*
Unauthorized OXPd APIs	*It was observed that the application is using OXPd APIs which is considered to be unauthorized.* *File:*	*High*	*It is against HP Policy to use OXPd APIs in Workpath apps.*
Validating Content from Third Parties	*shouldOverrideUrlLoading - Give the host application a chance to take control when a URL is about to be loaded in the current WebView.* *shouldInterceptRequest - Notify the host application of a resource request and allow the application to return the data.* *It is possible for an attacker to load malicious URLs.*	*Medium*	*shouldOverrideUrlLoading (WebView view, String url) method is deprecated so use shouldOverrideUrlLoading(WebView, WebResourceRequest) instead.* *Do not call WebView#loadUrl(String) with the same URL and then return true. The correct way to continue loading a given URL is to simply return false, without calling WebView#loadUrl(String).* *Reference:* *1)https://developer.android.com/reference/android/webkit/WebViewClient#shouldOverrideUrlLoading(android.webkit.WebView,%20android.webkit.WebResourceRequest)* *2)https://developer.android.com/reference/android/webkit/WebViewClient#shouldInterceptRequest(android.webkit.WebView,%20android.webkit.WebResourceRequest)*
Remote WebView debugging is enabled.	*It was observed that remote webview debugging is enabled in the following file:* *WebView debugging enabled allows anyone to read all the files inside the private data directory.*	*Medium*	*It is not recommended to leave webiew debugging enabled.*

Screen Capture via 3rd party Apps	*It was observed that the app does not protect sensitive screens from being displayed in screencasts initiated by 3rd party apps.* *Files:* *XXXX*	*Medium*	*It is recommended that to protect your apps from being recorded by other apps, FLAG_SECURE should be used on any views containing sensitive data. Additionally, using of virtual keyboards should be avoided.*
CBC Mode in Encryption Algorithm	*The App uses the encryption mode CBC with PKCS5/PKCS7 padding. This configuration is vulnerable to padding oracle attacks.*	*Medium*	*It is recommended to use GCM mode of encryption.*
Application Data can be Backed up	*It was observed that [android:allowBackup] flag is missing.* *By default it is set to true and allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data off of the device.*	*Medium*	*It is recommended to set the flag [android:allowBackup] to false.*
Elf built without protection	*It was observed that there exists an elf built without Stack Protection. Stack canaries can greatly increase the difficulty of exploiting a stack buffer overflow because it forces the attacker to gain control of the instruction pointer by some non-traditional means such as corrupting other important variables on the stack. Built with option -fstack-protector.* *Files :* *XXXX*	*Medium*	*It is recommended to implement proper Stack Protection while compiling elf files.*
Missing Intent Protection	*It was observed that an intent-filter exists but no protection was detected for the following broadcast Receiver:* *XXXX* *A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Broadcast Receiver is explicitly exported.*	*Medium*	*It is recommended to protect the Broadcast Receiver to prevent it from being accessed by any other applications on the device. Use only explicit intents*
Missing Intent Protection	*It was observed that an intent-filter exists but no protection was detected for the following Activity:* *XXXX* *The Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the services are explicitly exported.*	*Medium*	*It is recommended to protect the Activity to prevent it from being accessed by any other applications on the device. Use only explicit intents*
Debug Enabled	*It was observed that debugging was enabled on the app:* *[android:debuggable=true]* *This makes it easier for reverse engineers to hook a debugger to it. This allows dumping a stack trace and accessing debugging helper classes.*	*Medium*	*It is recommended to disable debugging.* *[android:debuggable=false]*
Elf built without Position Independent Executable Flag	*Found elf built without Position Independent Executable (PIE) flag* *File:*	*Medium*	*In order to prevent an attacker from reliably jumping to, for example, a particular exploited function in memory, Address space layout randomization (ASLR) randomly arranges the address space positions of key data areas of a process, including the base of the executable and the positions of the stack, heap and libraries. Built with option -pie.*
Raw SQL Query Executed	*It was observed that the app uses SQLite Database and execute raw SQL queries. Untrusted user input in raw SQL queries can cause SQL Injection. Also sensitive information should be encrypted and written to the database.* *File:* *XXXX*	*Medium*	*It is recommended to use parameterized queries and implement proper input validation.*
Shared functions not protected	*It was observed that the following shared activities, services and broadcast receivers were not protected [android:exported=true]:* *These are found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. Also, if the permission is set to normal or dangerous, a malicious application can request and obtain the permission and interact with the component.*	*Medium*	*It is recommended to set proper permissions on the shared activities, services and broadcast receivers . Set it to signature, so that only applications signed with the same certificate can obtain the permission.*
Application supports Older version of Android	*It was observed that Application can be installed on older version of Android (KitKat Android SDK 19) which is not supported by Google anymore. This may help an attacker to exploit if any open vulnerabilities in the older version.*	*Medium*	*It is recommended to disallow any application to be installed on non-supported and deprecated version of Android.*
Application Data can be tampered [android:useEmbeddedDex] flag is missing.	*The application does not enable using the embedded DEX file for app launching. This means the app is not taking all possible protections from tampering. By default [android:useEmbeddedDex] flag is set to false.*	*Medium*	*Enabling this feature can protect your app from tampering while stored on-device, but can slow down app launch times. To enable it, set the [android:useEmbeddedDex=true] in AndroidManifest.xml.*
Logging Enabled	*It was observed that the application logs sensitive information.* *File :* *XXXX*	*Low*	*It is recommended that the application should never log sensitive data.*
Launch Mode of Activity is not standard	*It was observed that the launch mode of the following activities is not standard:* *XXXX* *An Activity should not be having the launch mode attribute set to "singleTask/singleInstance" as it becomes root Activity and it is possible for other applications to read the contents of the calling Intent.*	*Low*	*It is recommended to use the "standard" launch mode attribute when sensitive information is included in an Intent.*
WAKE_LOCK permission enabled	*It was observed that android.permission.WAKE_LOCK was enabled. This would prevent user session from timing out.*	*Low*	*It is recommended to disable android.permission.WAKE_LOCK*
The remote server exposes the internal IP address	*It was observed that it was possible to obtain the internal IP address or internal network name due to a vulnerability in the installed server. The server exposes internal IP addresses that are usually hidden or masked behind a Network Address Translation (NAT) firewall or a proxy server.* *Files:* *XXXX*	*Low*	*It is recommended not to expose internal IP addresses.*
Insecure Random Number Generator	*It was observed that the app uses an insecure Random Number Generator in the following file:* *XXXX*	*Low*	*It is recommended to implement a cryptographic pseudo random number generators which can generate an output that is more difficult to predict*
TaskAffinity is set	*It was observed that the TaskAffinity is set for the following activities:*	*Low*	*It is recommended to always use the default setting keeping the affinity as the package name in order to prevent sensitive information inside sent or received Intents from being read by another application.*
App can write to App Directory	*It was observed that App can write to App Directory (Context.MODE_PRIVATE) in the following files:* *xxxx*	*Low*	*It is recommended to encrypt the sensitive information.*