

Mobile Application Security Audit of Neva android app

Test APK
[release.apk](#)

Interim Report
08th June 2020



AAA Technologies P. Ltd

278-280, F Wing, Solaris-1,
Saki Vihar Road, Opp. L & T Gate No. 6,

Powai, Andheri (East),

Mumbai 400 072, INDIA

Tel: + 91 22 28573815 / 16

Fax: + 91 22 40152501

info@aaatechnologies.co.in

www.aaatechnologies.co.in



Mobile Application Security Test Report For Neva Android App



S.No.	Severity	Vulnerability Description	Level-1	Compliance Status
1.	High	Insecure Authorization.	Open	Not Complied
2.	High	Debug mode is enabled in the app.	Open	Complied
3.	High	Allow backup is enabled in the app.	Open	Complied
4.	High	External Storage is allowed in the app.	Open	Not Complied
5.	High	OTP Flooding attack is possible in the app.	Open	Complied
6.	High	OTP brute force is possible in the app.	Open	Complied
7.	High	User password is travel in the clear text in the app.	Open	Complied
8.	High	Brute force attack is possible in the app.	Open	Complied
9.	Medium	Old version of ASP.NET is used in the app.	Open	-
10.	Medium	OTP is not getting validated in the app.	Open	Complied
11.	Low	Web Server Information Disclosure.	Open	Complied
12.	Low	Max. Length for input fields is not defined called in the application.	Open	Complied
13.	Low	Input fields are not getting cleared after invalid login attempts.	Open	Not Complied
14.	Low	Change password module is not implemented.	Open	Not Complied

Mobile Application Security Test Report For Neva Android App



15.	Medium	Password is not getting validated.	-	Complied
16.	Medium	Sensitive information disclosure is possible in the app.	-	Complied
17.	Low	Forgot password is not implemented in the app.	-	Complied
18.	Informational	Functionality Issues.	-	Not Complied

S.No.	Severity	Vulnerability Description	Level-1	Compliance Status
1.	High	Dangerous permissions are used in the app.	-	NEW
2.	High	Aadhar number shows in the response.	-	NEW
3.	Medium	Java hash code is used in the app.	-	NEW
4.	Medium	Base64 encoding is used in the app.	-	NEW
5.	Low	PrintStackTrace method is used in the app.	-	NEW
6.	Low	Weak algorithm is used in the app.	-	NEW



High

Mobile Application Security Test Report For Neva Android App



1. Insecure Authentication.

1) Vulnerability Title: Insecure Authentication mechanism (Response Replay) is possible in the app.	
Risk	High
Abstract	It was observed that Authentication bypass is possible in the application.
Ease of Exploitation	Medium
Impact	Insecure Authentication mechanism is used in the application. Attacker can bypass the authentication to gain access to the application
Recommendations	Proper Authentication mechanism must be implemented to avoid authentication bypass.
Snapshot	

How Test was performed:

Step#1: Open release.apk and login with valid credentials (U: 9667892443, P: neva42) and intercept the request as shown below:



Mobile Application Security Test Report For Neva Android App



Step#2: Copy valid response into the notepad and forward the request as shown below:

The screenshot shows a Notepad window on the left containing the following text:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: application/json
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type, Accept, User-Agent
Access-Control-Allow-Methods: POST, GET, PUT, DELETE, OPTIONS
Access-Control-Max-Age: 1728000
Date: Thu, 04 Jun 2020 12:16:21 GMT
Content-Length: 48

{"Response_SMS": "OTP has been sent to the user"}
```

On the right, the Burp Suite interface shows a response from `http://164.100.68.194:80/AndroidService/AndroidService.svc/AuthenticateMemberWithMobileSV1New`. The response body is highlighted in orange and matches the text in the Notepad. A red arrow points from the Notepad to the Burp Suite response.

Step#3: Now enter the valid OTP and intercept the request then copy the valid response into the notepad and forward the request as shown below:

The screenshot shows a Notepad window on the left containing a large JSON response:

```
Date: Thu, 04 Jun 2020 12:17:20 GMT
Content-Length: 1274

{"Statecode": "LokSabha", "SwearingPath": "", "Discriptions": "", "TotalMembers": "", "PartyCode": "", "DOB": "", "Token": "", "MemberCode": "451", "AadhaarNo": "544444444444", "Prefix": "", "Name": "Sh. Radha Mohan Singh", "NameLocal": "Radha Mohan Singh", "ShimlaAddress": "6, Ashoka Road, \r\nNew Delhi - 110 001\r\nTels. : (011) 23072370, 23072380, 09013180251 (M)\r\nFax : (011) 23072390", "PermanentAddress": "Gandhi Complex, Station Road, Motihari, \r\nDistt. East Champaran-845401, Bihar\r\nTel : (06252) 241210, 09431815551 (M)", "Mobile": "9667892443", "Email": "minister-agri@nic.in", "PartyName": "Bhartiya Janta Party", "PartyNameLocal": "भारत-आधार-अभियान", "Designation": "Member of Parliament", "ConstituencyName": "Purvi Champaran", "ConstituencyNameLocal": "Purvi Champaran", "ProfilePicPath": "https://cms.neva.gov.in/FileStructure_LS/Member/07_17_2019_02_21_17.jpg", "IsActive": "True", "ConstituencyCode": "486", "Online": "", "StatusId": "", "UserName": "", "SubUserId": "17", "SubUserName": "", "Photo": "https://cms.neva.gov.in/FileStructure_LS/Member/07_17_2019_02_21_17.jpg", "departmentName": "", "OfficeName": "", "departmentId": "", "OfficeId": "", "OfficeLevel": "5", "UserId": "", "SubDivisionCode": "", "SubDivisionName": "", "DistrictCode": "", "DistrictName": "", "Result": ""}
```

On the right, the Burp Suite interface shows a response from `http://164.100.68.194:80/AndroidService/AndroidService.svc/VerifyOTPMemberSV1New`. The response body is highlighted in orange and matches the text in the Notepad. A red arrow points from the Notepad to the Burp Suite response.

Mobile Application Security Test Report For Neva Android App



Step#4: Now logout from the app and again login with invalid credentials, intercept the request as shown below:

5:52:20 0.00 K/s 31%

LOK SABHA लोक सभा
सत्यमेव जयते

National eVidhan Application
लॉग इन करें

2222222222

....|

ओटीपी भेजें

साइन इन करें

[Forgot password](#)

Home Video News Profile NEA

Mobile Application Security Test Report For Neva Android App



Step#5: Now replace invalid response with copied valid response and forward the request as shown below:

The screenshot shows a Notepad window on the left with the following text copied: `HTTP/1.1 200 OK`, `Cache-Control: private`, `Content-Type: application/json`, `Access-Control-Allow-Origin: *`, `Access-Control-Allow-Headers: Content-Type,Accept,User-Agent`, `Access-Control-Allow-Methods: POST,GET,PUT,DELETE,OPTIONS`, `Access-Control-Max-Age: 1728000`, `Date: Thu, 04 Jun 2020 12:16:21 GMT`, `Content-Length: 48`, and `{"Response_SMS":"OTP has been sent to the user"}`. On the right, the Burp Suite interface shows an intercepted response from `http://164.100.68.194:80/AndroidService/AndroidService.svc/AuthenticateUserByMobileNo`. The response headers are visible, and the body contains `"Unauthorize User"`. The 'Forward' button is highlighted, indicating the next step in the process.


The screenshot shows the same Notepad window on the left. On the right, the Burp Suite interface shows the same intercepted response. The 'Raw' tab is selected, and the body of the response is highlighted in red. The text `{"Response_SMS":"OTP has been sent to the user"}` has been pasted into the body, replacing the previous `"Unauthorize User"`. A red arrow points from the Notepad window to the Burp Suite interface, indicating the source of the copied text.


Mobile Application Security Test Report For Neva Android App



Step#6: It is been observed that the other is successfully bypass the login:

5:54:21 0.29 K/s 30%

 **LOK SABHA लोक सभा**
सत्यमेव जयते


सत्यमेव जयते

National eVidhan Application

लॉग इन करें

2222222222






....

ओटीपी भेजें

OTP दर्ज करें

साइन इन करें

[Forgot password](#)

Mobile Application Security Test Report For Neva Android App



Step#7: Now enter invalid OTP and intercept the request and then replace the response with copied valid response and forward the request as shown below:

The screenshot shows a Notepad window on the left containing a valid JSON response from the server. The response includes fields such as "Statecode", "TotalMembers", "PartyCode", "DOB", "Token", "MemberCode", "AadhaarNo", "Prefix", "Name", "ShimlaAddress", "Tels", "Fax", "PermanentAddress", "Station Road", "Distt", "Bihar", "Tel", "Mobile", "Email", "PartyName", "Party", "PartyNameLocal", "Designation", "ConstituencyName", "ConstituencyNameLocal", "ProfilePicPath", "ConstituencyCode", "Online", "StatusId", "UserName", "SubUserId", "SubUserName", "Photo", "departmentName", "OfficeName", "departmentId", "OfficeId", "OfficeLevel", "UserId", "SubDivisionCode", "SubDivisionName", "DistrictCode", and "DistrictName".

On the right, the Burp Suite interface shows the intercepted response. The "Response" tab is active, displaying the same JSON data. The "Intercept" button is highlighted, and the "Intercept is on" status is visible.


The screenshot shows a Notepad window on the left containing a modified JSON response. The response is identical to the one in the previous screenshot, but the "Mobile" field has been changed to "9667892443".

On the right, the Burp Suite interface shows the intercepted response. The "Response" tab is active, displaying the modified JSON data. A red arrow points from the Notepad window to the "Response" tab in Burp Suite, indicating that the modified response is being used to replace the original one.

Mobile Application Security Test Report For Neva Android App




Step#8: It is been observed that the other is successfully login in the app without valid credentials and able to access authenticated links:




LOK SABHA लोक सभा
सत्यमेव जयते

मेरा पेज



Sh. Radha Mohan Singh
Bhartiya Janta Party
Purvi Champaran , LokSabha



ऑनलाइन जमा करें



मेरे प्रश्न / उत्तर



अल्प अवधि सूचना

Mobile Application Security Test Report For Neva Android App



3. Dangerous permissions are allowed in the Android Application

3) Vulnerability Title: Dangerous permissions are allowed in the Android Application	
Risk	High
Abstract	It was observed that Dangerous permissions are allowed in the Android Application.
Ease of Exploitation	Easy
Impact	It is allow an app to view network status, create network socket, prevent phone from sleeping, display system level alerts.
Recommendations	It is recommended to restrict app permissions.
Snapshot	<pre> <?xml version="1.0" encoding="UTF-8"?> <manifest platformBuildVersionName="10" platformBuildVersionCode="29" package="evmember.sbl.nationalevidhan" android:compileSdkVersionCodename="10" android:compileSdkVersion="29" xmlns:android="http://schemas.android.com/apk/res/android"> <uses-permission android:name="android.permission.ACCESS_DOWNLOAD_MANAGER"/> <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/> <uses-permission android:name="android.permission.INTERNET"/> <uses-permission android:name="android.permission.CALL_PHONE"/> <uses-permission android:name="android.permission.WAKE_LOCK"/> <uses-permission android:name="android.permission.ACCESS_DOWNLOAD_MANAGER"/> <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/> <uses-permission android:name="android.permission.INTERNET"/> <uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT"/> <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/> <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/> <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/> <supports-screens android:largeScreens="true" android:normalScreens="true" android:resizeable="true" android:smallScreens="true" android:xlargeScreens="true" android:anyDensity="true"/> <application android:name="evmember.sbl.nationalevidhan.application.MyNevaApplication" android:usesCleartextTraffic="true" android:supportsRtl="true" android:roundIcon="@drawable/app_launcher" android:requestLegacyExternalStorage="true" android:largeHeap="true" android:label="@string/app_name" android:icon="@drawable/app_launcher" android:hardwareAccelerated="true" android:fullBackupContent="false" android:debuggable="false" android:configChanges="locale" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:allowBackup="false"> <uses-library android:name="org.apache.http.legacy" android:required="false"/> <activity android:name="evmember.sbl.nationalevidhan.SplashActivity" android:theme="@style/Theme.NoTitle" android:screenOrientation="portrait" android:noHistory="true"> <intent-filter> <action android:name="android.intent.action.MAIN"/> <category android:name="android.intent.category.LAUNCHER"/> </intent-filter> </activity> <activity android:name="evmember.sbl.nationalevidhan.TodayBirthdayActivity" android:configChanges="orientation" android:theme="@style/AppTheme" android:screenOrientation="portrait"/> <activity android:name="evmember.sbl.nationalevidhan.views.ChangePasswordActivity" android:configChanges="orientation" android:theme="@style/AppTheme" android:screenOrientation="portrait"/> <activity android:name="evmember.sbl.nationalevidhan.DebateTypeSearchActivity" android:configChanges="orientation" android:theme="@style/AppTheme" android:screenOrientation="portrait" android:windowSoftInputMode="adjustResize stateHidden"/> <activity android:name="evmember.sbl.nationalevidhan.DebateGuidedSearchActivity" android:configChanges="orientation" android:theme="@style/AppTheme" android:screenOrientation="portrait" android:windowSoftInputMode="adjustResize stateHidden"/> <activity android:name="evmember.sbl.nationalevidhan.BusinessGovernmentBillActivity" android:configChanges="orientation" android:theme="@style/AppTheme" </pre>



Medium

Mobile Application Security Test Report For Neva Android App



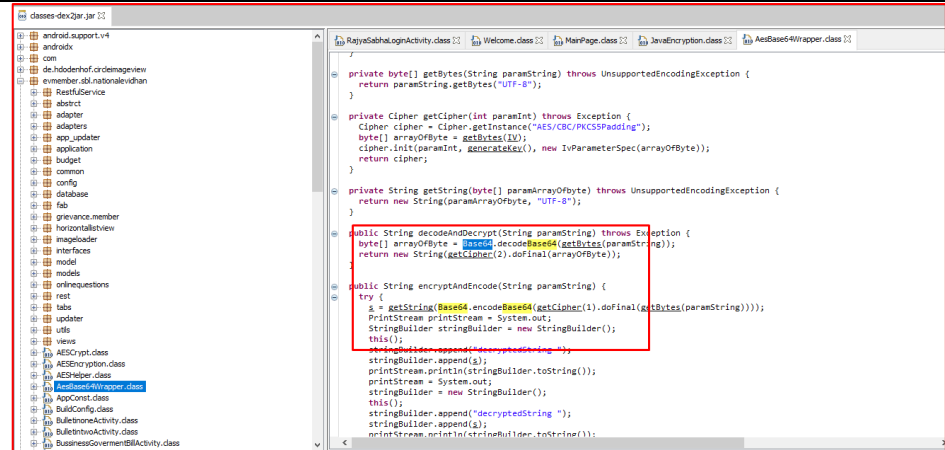
5. Java Hash code is used in the application

1) Vulnerability Title: Java hash code is used in the application	
Risk	Medium
Abstract	It was observed that the App uses Java Hash Code.
Ease of Exploitation	Easy
Impact	It's a weak hash function and should never be used in Secure Crypto Implementation
Recommendations	Need to serialize the object to a byte stream (which you need to do anyway if you're going to send it over the network). If you're using a serialization that always maps the same values to the same sequence of bytes, you can just hash that byte stream. A cryptographic hash such as MD5 or SHA-1 would be ok for many cases, but might be a bit heavyweight if you're dealing with a really high throughput service.
Snapshot	<p>The top screenshot shows the FileCache class in the package evmember.sdi.nationalevidhan.imageLoader. It contains a File cacheDir and a hashCode() method that returns the hash code of the cacheDir string.</p> <pre> package evmember.sdi.nationalevidhan.imageLoader; import android.content.Context; import android.os.Environment; import java.io.File; public class FileCache { private File cacheDir; public FileCache(Context paramContext) { if (Environment.getExternalStorageState().equals("mounted")) { this.cacheDir = new File(Environment.getExternalStorageDirectory(), "LazyList"); } else { this.cacheDir = paramContext.getCacheDir(); } if (!this.cacheDir.exists()) { this.cacheDir.mkdirs(); } } public void clear() { File[] arrayOfFile = this.cacheDir.listFiles(); if (arrayOfFile == null) return; int i = arrayOfFile.length; for (byte b = 0; b < i; b++) arrayOfFile[b].delete(); } public File getFile(String paramString) { int i = paramString.hashCode(); return new File(this.cacheDir, String.valueOf(i)); } } </pre> <p>The bottom screenshot shows the Rotate class in the package com.bumpteck.glide.load.resource.bitmap. It contains a hashCode() method that returns the hash code of the degreesToRotate integer.</p> <pre> package com.bumpteck.glide.load.resource.bitmap; import java.util.HashMap; import java.util.Map; private static final String ID = "com.bumpteck.glide.load.resource.bitmap.Rotate"; private static final byte[] ID_BYTES = "com.bumpteck.glide.load.resource.bitmap.Rotate".getBytes(CHARSET); private final int degreesToRotate; public Rotate(int paramInt) { this.degreesToRotate = paramInt; } public boolean equals(Object paramObject) { boolean bool = paramObject instanceof Rotate; boolean bool1 = false; if (bool) { paramObject = paramObject; if (this.degreesToRotate == ((Rotate)paramObject).degreesToRotate) bool1 = true; return bool1; } return false; } public int hashCode() { return Util.hashCode("com.bumpteck.glide.load.resource.bitmap.Rotate", hashCode(), Util.hashCode(this.degreesToRotate)); } protected Bitmap transform(@NonNull BitmapPool paramBitmapPool, @NonNull Bitmap paramBitmap, int paramInt1, int paramInt2) { return TransformationUtil.rotateImage(paramBitmap, this.degreesToRotate); } public void updateDiskCacheKey(@NonNull MessageDigest paramMessageDigest) { paramMessageDigest.update(ID_BYTES); paramMessageDigest.update(ByteBuffer.allocate(4).putInt(this.degreesToRotate).array()); } } </pre>

Mobile Application Security Test Report For Neva Android App



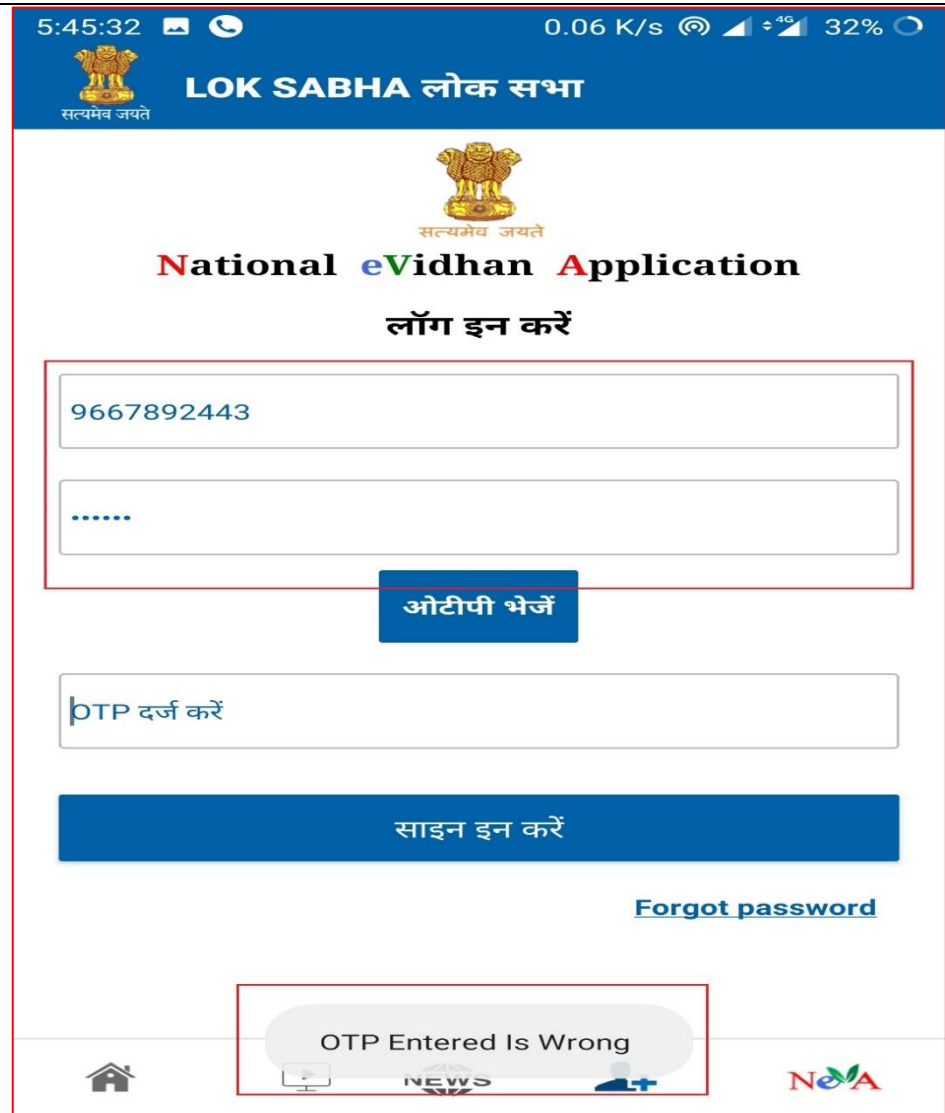
6. Base64 encoding is used in the application

2) Vulnerability Title: Base64 encoding is used in the application	
Risk	Medium
Abstract	It was observed that the App uses base64 encoding.
Ease of Exploitation	Easy
Impact	Base64 Encoded values can be decode easily.
Recommendations	It is recommended to use strong encryption along with salt value instead of Base64 encryption.
Snapshot	



Low

7. Input field is not getting cleared after invalid login attempts

1) Vulnerability Title: Input field is not getting cleared after invalid login attempts	
Risk	Low
Abstract	It was observed that form fields values are not getting cleared after an invalid attempt.
Ease of Exploitation	Easy
Impact	Attacker may misuse the details or user enumeration.
Recommendations	All login form field values must get clear on page refresh or reload.
Snapshot	



8. Change Password module is not implemented in the application

2) Vulnerability Title: Change Password module is not implemented in the application	
Risk	Low
Abstract	It was observed that change password module is missing in the app.
Ease of Exploitation	Easy
Impact	Change password allows administrators to customize the password change experience for the users from the begging to the end.
Recommendations	Change password module should be implemented in the application.
Snapshot	-

Mobile Application Security Test Report For Neva Android App



9. PrintStackTrace method is used in the application

3) Vulnerability Title: PrintStackTrace method is used in the application	
Risk	Low
Abstract	It was observed that PrintStackTrace method is used in the app.
Ease of Exploitation	Easy
Impact	printStackTrace function is used for exception handling that may lead to data leakage.
Recommendations	The use of printStackTrace() exception method can reveal information about the application which may help an adversary in exploiting the application. The printStackTrace() method displays detailed information such as stack traces, database dumps, exception and application details. An adversary may gain sensitive information of application and can exploit the weakness. It is recommended not to use printStackTrace() method in production environment. Store such error messages on server side in error log file and display customized messages to the user
Snapshot	

Mobile Application Security Test Report For Neva Android App



10. Weak Hashing algorithm is used in the application

4) Vulnerability Title: Weak algorithm is used in the application	
Risk	Low
Abstract	It was observed that weak algorithm MD5 is used in the app.
Ease of Exploitation	Easy
Impact	Weak hashing algorithms [e.g. MD2, MD4, MD5, SHA-0 or SHA-1] can be vulnerable to hash collisions, so they should not be used when reliable data hashing is required
Recommendations	It is recommended to use strong hashing algorithm
Snapshot	<pre> public static final String md5(String paramString) { try { MessageDigest messageDigest = MessageDigest.getInstance("MD5"); messageDigest.update(paramString.getBytes()); byte[] arrayOfByte = messageDigest.digest(); StringBuilder stringBuilder = new StringBuilder(); this(); int i = arrayOfByte.length; for (byte b = 0; b < i; b++) { stringBuilder.append(Integer.toHexString(arrayOfByte[b] & 0xFF)); paramString = stringBuilder.append(" "); } return stringBuilder.toString(); } catch (NoSuchAlgorithmException noturhalozothafrention noturhalozothafrention) { } } </pre>


Mobile Application Security Test Report For Neva Android App



11. Functionality Issues

1. We are not able to upload file in Notice link.

1:24:18 0.00 K/s 4G 30%

 **LOK SABHA लोक सभा** 
सत्यमेव जयते

कार्य के प्रकार : Motion - Adjournment
Motion(Rule 67)

मंत्री : Rajnath Singh,DEFENCE

प्रश्न की भाषा : ENGLISH

तारीख चुनें : 03 April 2020

विषय :

upload

हस्ताक्षर : Loading...

कागज संलग्न करें : Attach a PDF document.

विवरण :

upload

सहेजें और बाद में भेजें भेजें

Mobile Application Security Test Report For Neva Android App



2. Home page – Committee- Composition, meetings, reports links are not working, as dropdown is not showing any data.

