Signal Private Messenger 4.32.8

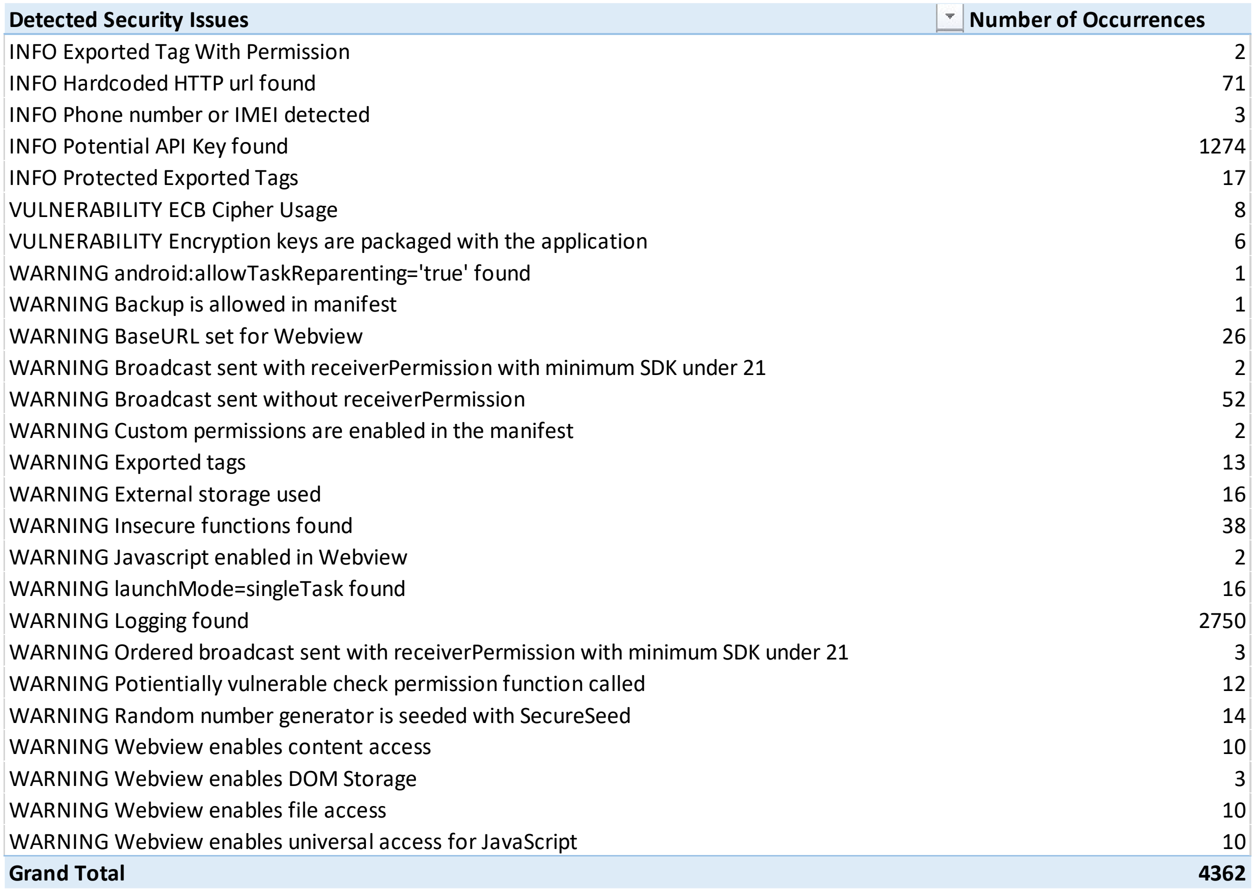
February 10th, 2018.

Description: This is a review of the Signal Private Messenger 4.32.8. After performing a static analysis of the app, vulnerabilities were identified related to device identification, file permissions, code execution, hash algorithm, and log activities. Additional results shows the use of 29 permissions without an android reference definition.

These analyses are aligned with the Common Weakness Enumerations (CWE) such as:

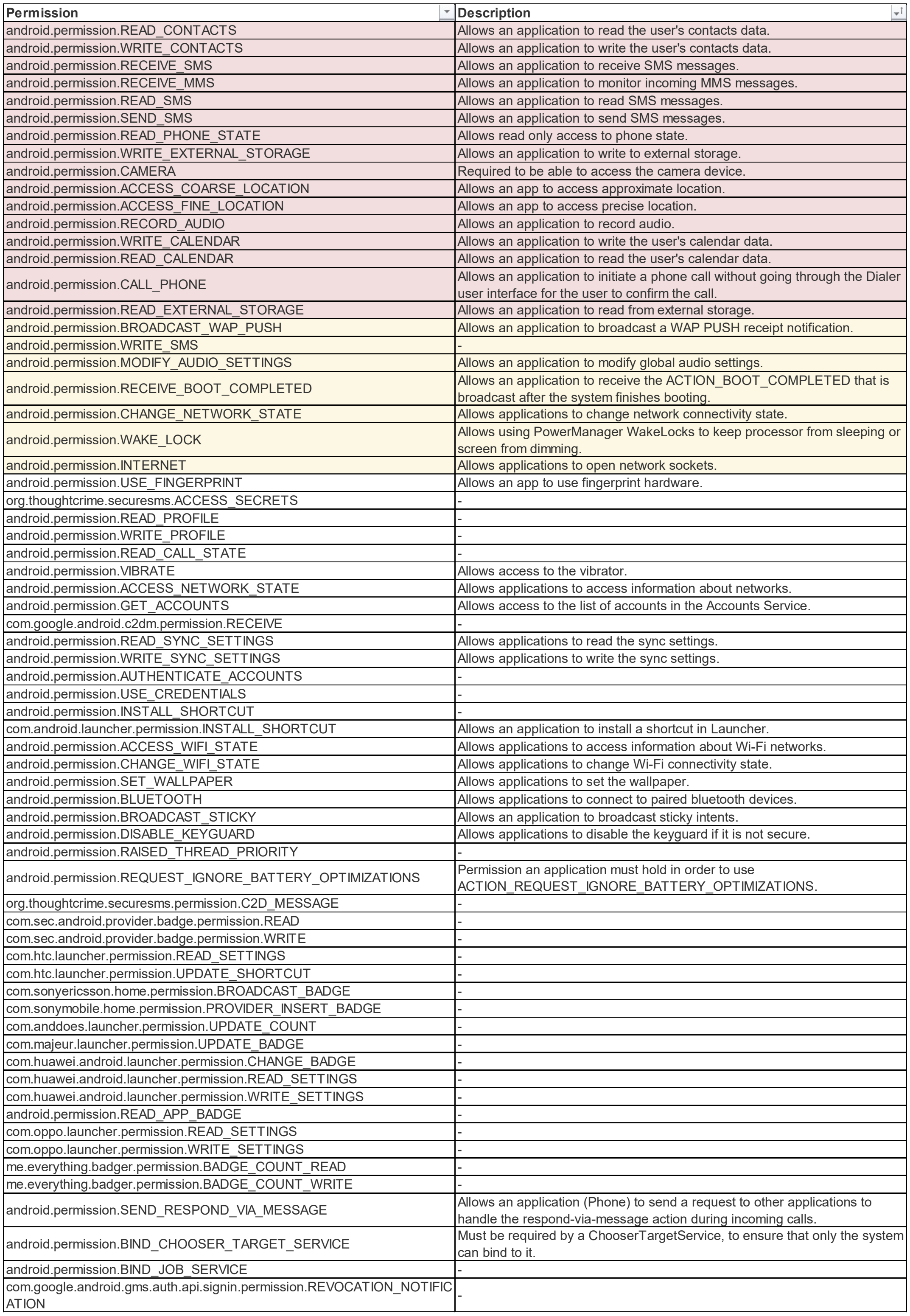
a) 312 (Cleartext Storage of Sensitive Information),
b) 327 (Use of a Broken or Risky Cryptographic Algorithm),
c) 532 (Information Exposure Through Log Files),
d) 330 (Use of Insufficiently Random Values),
e) 200 (Information Exposure),
f) 276 (Incorrect Default Permissions), and
g) 89 (Improper Neutralization of Special Elements used in an SQL Command - 'SQL Injection').

The summary of vulnerabilities include event such as:



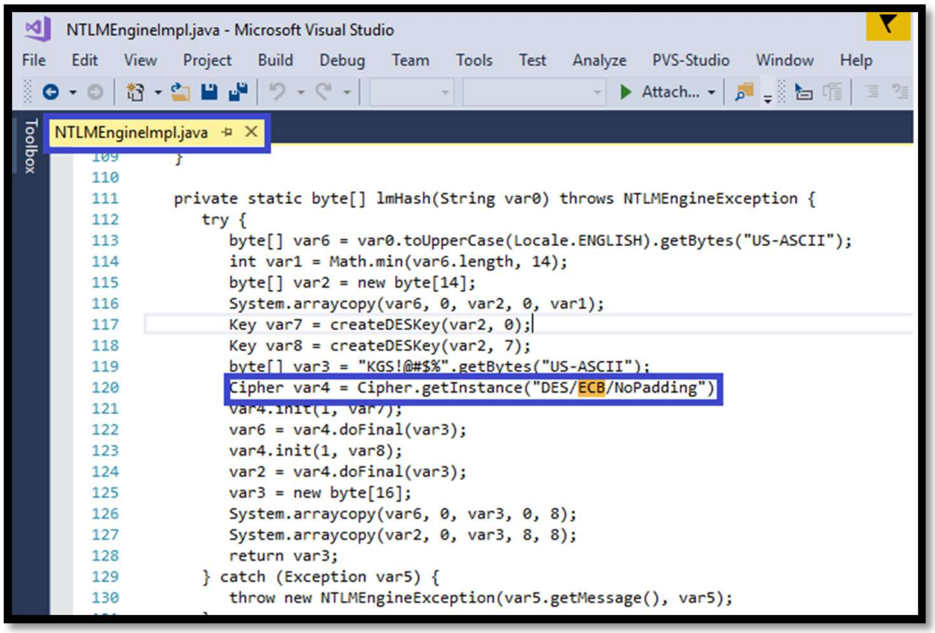
Detected Security Issues	Number of Occurrences
INFO Exported Tag With Permission	2
INFO Hardcoded HTTP url found	71
INFO Phone number or IMEI detected	3
INFO Potential API Key found	1274
INFO Protected Exported Tags	17
VULNERABILITY ECB Cipher Usage	8
VULNERABILITY Encryption keys are packaged with the application	6
WARNING android:allowTaskReparenting='true' found	1
WARNING Backup is allowed in manifest	1
WARNING BaseURL set for Webview	26
WARNING Broadcast sent with receiverPermission with minimum SDK under 21	2
WARNING Broadcast sent without receiverPermission	52
WARNING Custom permissions are enabled in the manifest	2
WARNING Exported tags	13
WARNING External storage used	16
WARNING Insecure functions found	38
WARNING Javascript enabled in Webview	2
WARNING launchMode=singleTask found	16
WARNING Logging found	2750
WARNING Ordered broadcast sent with receiverPermission with minimum SDK under 21	3
WARNING Potientially vulnerable check permission function called	12
WARNING Random number generator is seeded with SecureSeed	14
WARNING Webview enables content access	10
WARNING Webview enables DOM Storage	3
WARNING Webview enables file access	10
WARNING Webview enables universal access for JavaScript	10
Grand Total	**4362**

This android app also requires the following list of permissions to enable messaging communication among users:

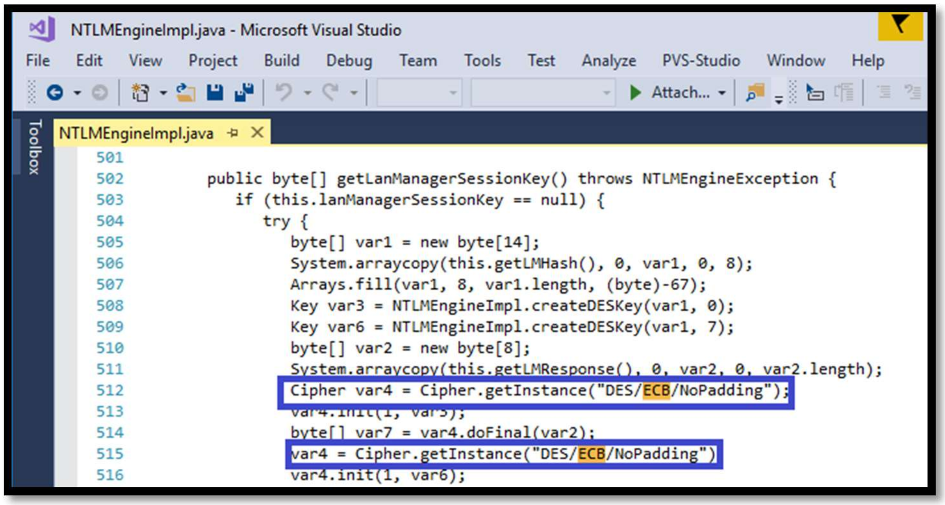


Permission	Description
android.permission.READ_CONTACTS	Allows an application to read the user's contacts data.
android.permission.WRITE_CONTACTS	Allows an application to write the user's contacts data.
android.permission.RECEIVE_SMS	Allows an application to receive SMS messages.
android.permission.RECEIVE_MMS	Allows an application to monitor incoming MMS messages.
android.permission.READ_SMS	Allows an application to read SMS messages.
android.permission.SEND_SMS	Allows an application to send SMS messages.
android.permission.READ_PHONE_STATE	Allows read only access to phone state.
android.permission.WRITE_EXTERNAL_STORAGE	Allows an application to write to external storage.
android.permission.CAMERA	Required to be able to access the camera device.
android.permission.ACCESS_COARSE_LOCATION	Allows an app to access approximate location.
android.permission.ACCESS_FINE_LOCATION	Allows an app to access precise location.
android.permission.RECORD_AUDIO	Allows an application to record audio.
android.permission.WRITE_CALENDAR	Allows an application to write the user's calendar data.
android.permission.READ_CALENDAR	Allows an application to read the user's calendar data.
android.permission.CALL_PHONE	Allows an application to initiate a phone call without going through the Dialer user interface for the user to confirm the call.
android.permission.READ_EXTERNAL_STORAGE	Allows an application to read from external storage.
android.permission.BROADCAST_WAP_PUSH	Allows an application to broadcast a WAP PUSH receipt notification.
android.permission.WRITE_SMS	-
android.permission.MODIFY_AUDIO_SETTINGS	Allows an application to modify global audio settings.
android.permission.RECEIVE_BOOT_COMPLETED	Allows an application to receive the ACTION_BOOT_COMPLETED that is broadcast after the system finishes booting.
android.permission.CHANGE_NETWORK_STATE	Allows applications to change network connectivity state.
android.permission.WAKE_LOCK	Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.
android.permission.INTERNET	Allows applications to open network sockets.
android.permission.USE_FINGERPRINT	Allows an app to use fingerprint hardware.
org.thoughtcrime.securesms.ACCESS_SECRETS	-
android.permission.READ_PROFILE	-
android.permission.WRITE_PROFILE	-
android.permission.READ_CALL_STATE	-
android.permission.VIBRATE	Allows access to the vibrator.
android.permission.ACCESS_NETWORK_STATE	Allows applications to access information about networks.
android.permission.GET_ACCOUNTS	Allows access to the list of accounts in the Accounts Service.
com.google.android.c2dm.permission.RECEIVE	-
android.permission.READ_SYNC_SETTINGS	Allows applications to read the sync settings.
android.permission.WRITE_SYNC_SETTINGS	Allows applications to write the sync settings.
android.permission.AUTHENTICATE_ACCOUNTS	-
android.permission.USE_CREDENTIALS	-
android.permission.INSTALL_SHORTCUT	-
com.android.launcher.permission.INSTALL_SHORTCUT	Allows an application to install a shortcut in Launcher.
android.permission.ACCESS_WIFI_STATE	Allows applications to access information about Wi-Fi networks.
android.permission.CHANGE_WIFI_STATE	Allows applications to change Wi-Fi connectivity state.
android.permission.SET_WALLPAPER	Allows applications to set the wallpaper.
android.permission.BLUETOOTH	Allows applications to connect to paired bluetooth devices.
android.permission.BROADCAST_STICKY	Allows an application to broadcast sticky intents.
android.permission.DISABLE_KEYGUARD	Allows applications to disable the keyguard if it is not secure.
android.permission.RAISED_THREAD_PRIORITY	-
android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS	Permission an application must hold in order to use ACTION_REQUEST_IGNORE_BATTERY_OPTIMIZATIONS.
org.thoughtcrime.securesms.permission.C2D_MESSAGE	-
com.sec.android.provider.badge.permission.READ	-
com.sec.android.provider.badge.permission.WRITE	-
com.htc.launcher.permission.READ_SETTINGS	-
com.htc.launcher.permission.UPDATE_SHORTCUT	-
com.sonyericsson.home.permission.BROADCAST_BADGE	-
com.sonymobile.home.permission.PROVIDER_INSERT_BADGE	-
com.anddoes.launcher.permission.UPDATE_COUNT	-
com.majeur.launcher.permission.UPDATE_BADGE	-
com.huawei.android.launcher.permission.CHANGE_BADGE	-
com.huawei.android.launcher.permission.READ_SETTINGS	-
com.huawei.android.launcher.permission.WRITE_SETTINGS	-
android.permission.READ_APP_BADGE	-
com.oppo.launcher.permission.READ_SETTINGS	-
com.oppo.launcher.permission.WRITE_SETTINGS	-
me.everything.badger.permission.BADGE_COUNT_READ	-
me.everything.badger.permission.BADGE_COUNT_WRITE	-
android.permission.SEND_RESPOND_VIA_MESSAGE	Allows an application (Phone) to send a request to other applications to handle the respond-via-message action during incoming calls.
android.permission.BIND_CHOOSER_TARGET_SERVICE	Must be required by a ChooserTargetService, to ensure that only the system can bind to it.
android.permission.BIND_JOB_SERVICE	-
com.google.android.gms.auth.api.signin.permission.REVOCATION_NOTIFICATION	-

A further review of the app identified the use of unsecure encryption tools such as Data Encryption Standard (DES)[1], Secure Hash Algorithm 1 (SHA-1)[2] and Electronic Code Book (ECB)[3] with no padding[4]. A reliable encryption also relies in a secure random number generator[5]. It is possible that a combination of these aspects could impair the confidentiality and privacy during users' communication.

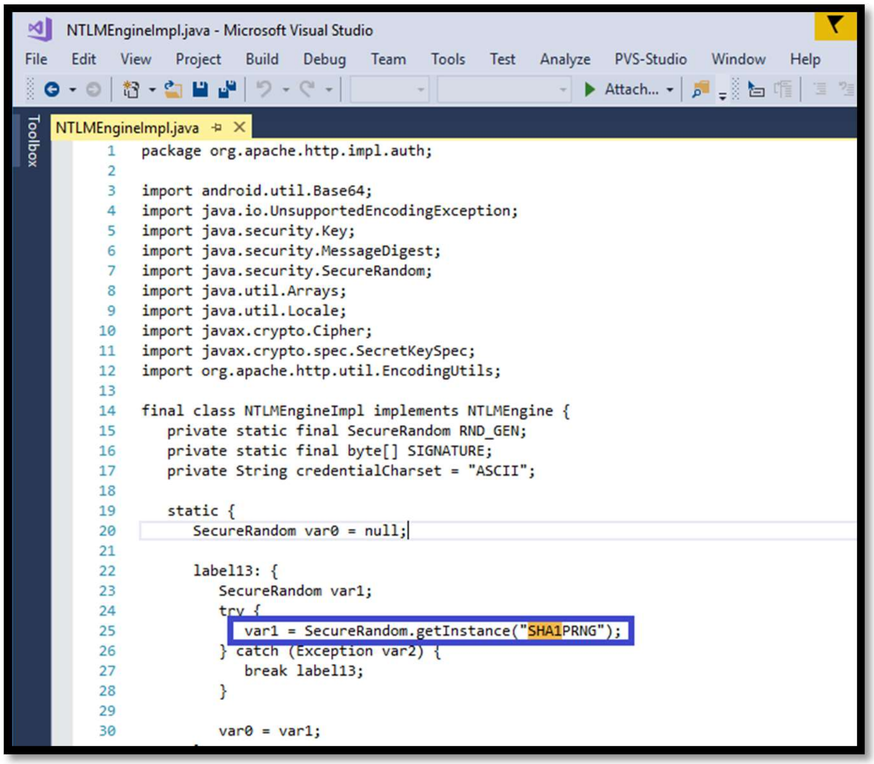

NTLMEngineImpl.java - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze PVS-Studio Window Help
Attach...
Toolbox
NTLMEngineImpl.java

```
   }

   private static byte[] lmHash(String var0) throws NTLMEngineException {
      try {
         byte[] var6 = var0.toUpperCase(Locale.ENGLISH).getBytes("US-ASCII");
         int var1 = Math.min(var6.length, 14);
         byte[] var2 = new byte[14];
         System.arraycopy(var6, 0, var2, 0, var1);
         Key var7 = createDESKey(var2, 0);
         Key var8 = createDESKey(var2, 7);
         byte[] var3 = "KGS!@#$%".getBytes("US-ASCII");
         Cipher var4 = Cipher.getInstance("DES/ECB/NoPadding")
         var4.init(1, var7);
         var6 = var4.doFinal(var3);
         var4.init(1, var8);
         var2 = var4.doFinal(var3);
         var3 = new byte[16];
         System.arraycopy(var6, 0, var3, 0, 8);
         System.arraycopy(var2, 0, var3, 8, 8);
         return var3;
      } catch (Exception var5) {
         throw new NTLMEngineException(var5.getMessage(), var5);
```

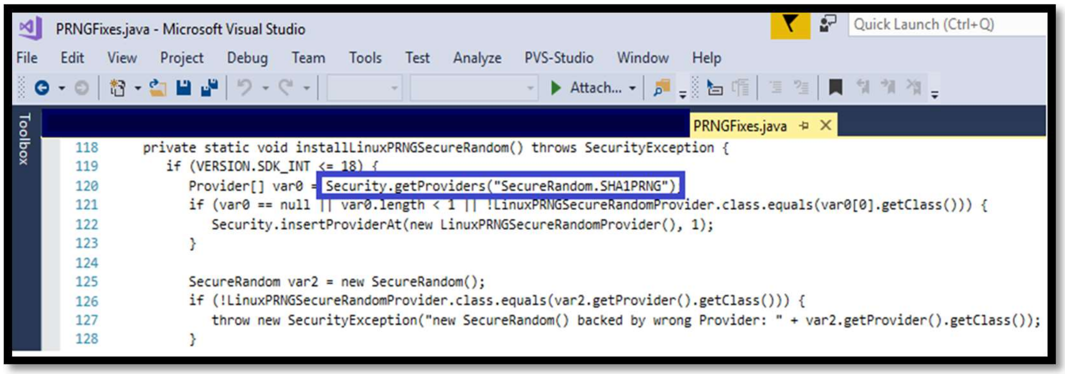

NTLMEngineImpl.java - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze PVS-Studio Window Help
Attach...
Toolbox
NTLMEngineImpl.java

```

   public byte[] getLanManagerSessionKey() throws NTLMEngineException {
      if (this.lanManagerSessionKey == null) {
         try {
            byte[] var1 = new byte[14];
            System.arraycopy(this.getLMHash(), 0, var1, 0, 8);
            Arrays.fill(var1, 8, var1.length, (byte)-67);
            Key var3 = NTLMEngineImpl.createDESKey(var1, 0);
            Key var6 = NTLMEngineImpl.createDESKey(var1, 7);
            byte[] var2 = new byte[8];
            System.arraycopy(this.getLMResponse(), 0, var2, 0, var2.length);
            Cipher var4 = Cipher.getInstance("DES/ECB/NoPadding");
            var4.init(1, var3);
            byte[] var7 = var4.doFinal(var2);
            var4 = Cipher.getInstance("DES/ECB/NoPadding")
            var4.init(1, var6);
```

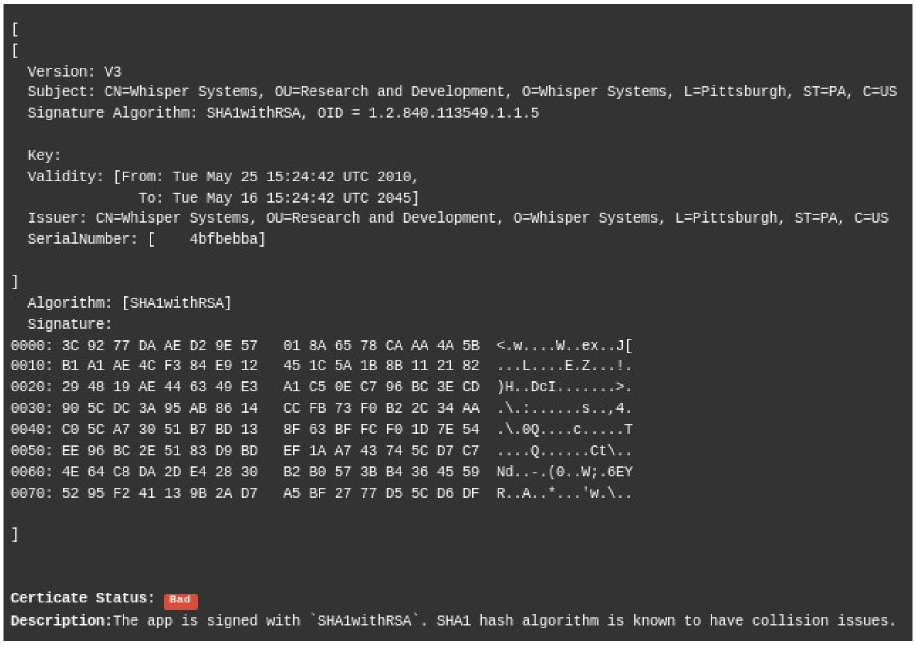
[1] https://searchsecurity.techtarget.com/definition/Data-Encryption-Standard
[2] https://en.wikipedia.org/wiki/SHA-1
[3] https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#ECB
[4] https://en.wikipedia.org/wiki/Padding_(cryptography)
[5] https://en.wikipedia.org/wiki/Random_number_generation



```
package org.apache.http.impl.auth;

import android.util.Base64;
import java.io.UnsupportedEncodingException;
import java.security.Key;
import java.security.MessageDigest;
import java.security.SecureRandom;
import java.util.Arrays;
import java.util.Locale;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import org.apache.http.util.EncodingUtils;

final class NTLMEngineImpl implements NTLMEngine {
   private static final SecureRandom RND_GEN;
   private static final byte[] SIGNATURE;
   private String credentialCharset = "ASCII";

   static {
      SecureRandom var0 = null;

      label13: {
         SecureRandom var1;
         try {
            var1 = SecureRandom.getInstance("SHA1PRNG");
         } catch (Exception var2) {
            break label13;
         }

         var0 = var1;
```

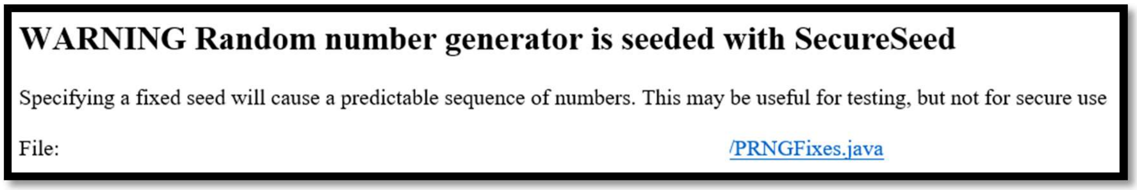


```
   private static void installLinuxPRNGSecureRandom() throws SecurityException {
      if (VERSION.SDK_INT <= 18) {
         Provider[] var0 = Security.getProviders("SecureRandom.SHA1PRNG")
         if (var0 == null || var0.length < 1 || !LinuxPRNGSecureRandomProvider.class.equals(var0[0].getClass())) {
            Security.insertProviderAt(new LinuxPRNGSecureRandomProvider(), 1);
         }

         SecureRandom var2 = new SecureRandom();
         if (!LinuxPRNGSecureRandomProvider.class.equals(var2.getProvider().getClass())) {
            throw new SecurityException("new SecureRandom() backed by wrong Provider: " + var2.getProvider().getClass());
         }
```



```
[
[
  Version: V3
  Subject: CN=Whisper Systems, OU=Research and Development, O=Whisper Systems, L=Pittsburgh, ST=PA, C=US
  Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

  Key:
  Validity: [From: Tue May 25 15:24:42 UTC 2010,
               To: Tue May 16 15:24:42 UTC 2045]
  Issuer: CN=Whisper Systems, OU=Research and Development, O=Whisper Systems, L=Pittsburgh, ST=PA, C=US
  SerialNumber: [    4bfbebba]

]
  Algorithm: [SHA1withRSA]
  Signature:
0000: 3C 92 77 DA AE D2 9E 57   01 8A 65 78 CA AA 4A 5B  <.w....W..ex..J[
0010: B1 A1 AE 4C F3 84 E9 12   45 1C 5A 1B 8B 11 21 82  ...L....E.Z...!.
0020: 29 48 19 AE 44 63 49 E3   A1 C5 0E C7 96 BC 3E CD  )H..DcI.......>.
0030: 90 5C DC 3A 95 AB 86 14   CC FB 73 F0 B2 2C 34 AA  .\.:......s..,4.
0040: C0 5C A7 30 51 B7 BD 13   8F 63 BF FC F0 1D 7E 54  .\.0Q....c.....T
0050: EE 96 BC 2E 51 83 D9 BD   EF 1A A7 43 74 5C D7 C7  ....Q......Ct\..
0060: 4E 64 C8 DA 2D E4 28 30   B2 B0 57 3B B4 36 45 59  Nd..-.(0..W;.6EY
0070: 52 95 F2 41 13 9B 2A D7   A5 BF 27 77 D5 5C D6 DF  R..A..*...'w.\..

]


Certicate Status: Bad
Description:The app is signed with `SHA1withRSA`. SHA1 hash algorithm is known to have collision issues.
```

In addition, the app uses a static[6] Random Number Generator[7]



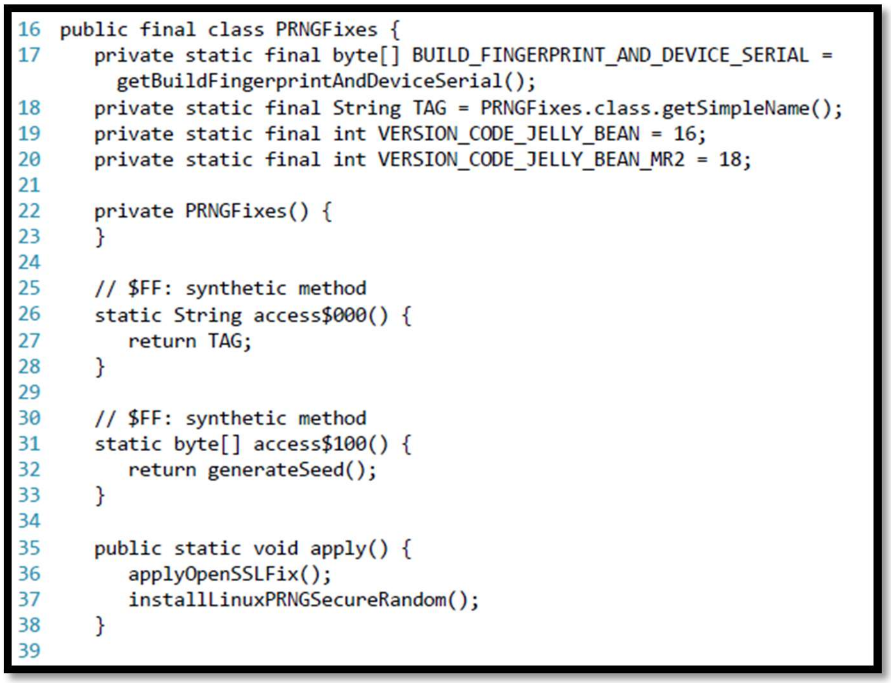
WARNING Random number generator is seeded with SecureSeed

Specifying a fixed seed will cause a predictable sequence of numbers. This may be useful for testing, but not for secure use

File: /PRNGFixes.java

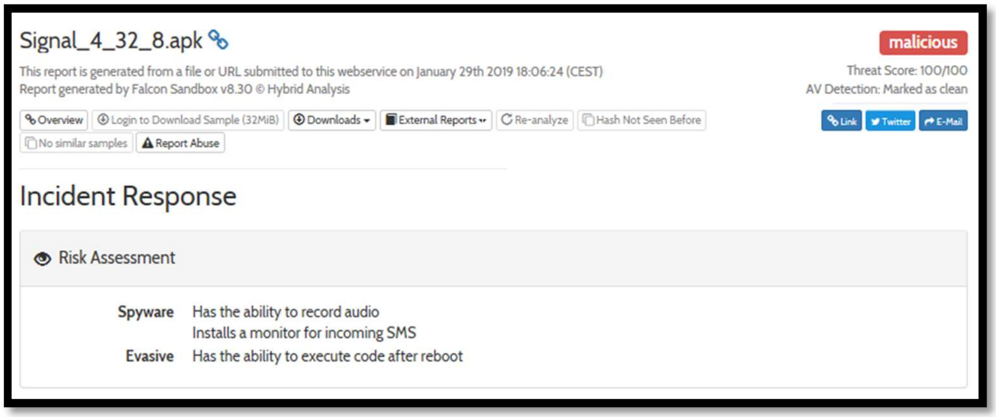
[6] https://docs.oracle.com/javase/7/docs/api/java/security/SecureRandom.html

[7] https://linux.die.net/man/4/urandom



```java
public final class PRNGFixes {
   private static final byte[] BUILD_FINGERPRINT_AND_DEVICE_SERIAL =
     getBuildFingerprintAndDeviceSerial();
   private static final String TAG = PRNGFixes.class.getSimpleName();
   private static final int VERSION_CODE_JELLY_BEAN = 16;
   private static final int VERSION_CODE_JELLY_BEAN_MR2 = 18;

   private PRNGFixes() {
   }

   // $FF: synthetic method
   static String access$000() {
      return TAG;
   }

   // $FF: synthetic method
   static byte[] access$100() {
      return generateSeed();
   }

   public static void apply() {
      applyOpenSSLFix();
      installLinuxPRNGSecureRandom();
   }

```

Additional vulnerabilities were found regarding audio, SMS and code execution[8]:



Signal_4_32_8.apk

This report is generated from a file or URL submitted to this webservice on January 29th 2019 18:06:24 (CEST)
Report generated by Falcon Sandbox v8.30 © Hybrid Analysis

malicious

Threat Score: 100/100
AV Detection: Marked as clean

Incident Response

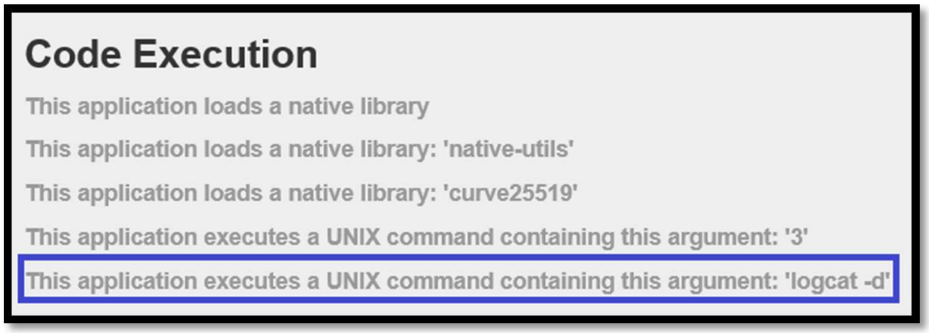
Risk Assessment

Spyware	Has the ability to record audio Installs a monitor for incoming SMS
Evasive	Has the ability to execute code after reboot

Under code execution, an example of activity that may be affected for a lower level of security encryption could be related to the log to the screen and each exist by the app using logcat (logcat -d)[9].

[8] https://www.hybrid-analysis.com/sample/ea486be30769593fbeb85b93f29033f68cf300cda17f125fbf44223a93554006/5c5075667ca3e1155004d018

[9] http://adbshell.com/commands/adb-logcat



Code Execution

This application loads a native library

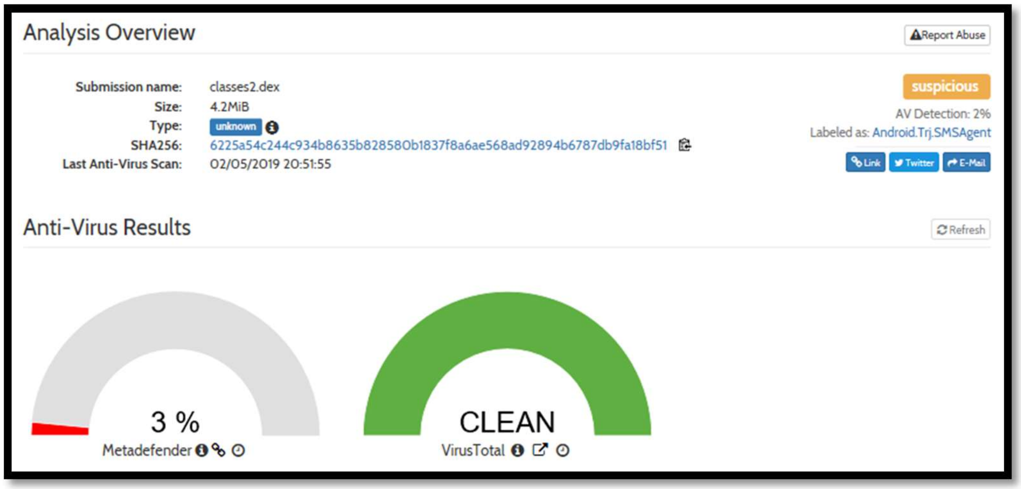
This application loads a native library: 'native-utils'

This application loads a native library: 'curve25519'

This application executes a UNIX command containing this argument: '3'

This application executes a UNIX command containing this argument: 'logcat -d'

With the component classes2.dex identified with a suspicious behavior[10].



Analysis Overview

Submission name: classes2.dex
Size: 4.2MiB
Type: unknown
SHA256: 6225a54c244c934b8635b828580b1837f8a6ae568ad92894b6787db9fa18bf51
Last Anti-Virus Scan: 02/05/2019 20:51:55

suspicious
AV Detection: 2%
Labeled as: Android.Trj.SMSAgent

Anti-Virus Results

3 % Metadefender

CLEAN VirusTotal

A possible explanation for these findings could be related to the work developed based on the Bouncy Castle[11] documentation since the codes[12] and vulnerabilities[13] are very similar.

[10] https://www.hybrid-analysis.com/sample/6225a54c244c934b8635b828580b1837f8a6ae568ad92894b6787db9fa18bf51/

[11] http://git.bouncycastle.org/mirrors.html

[12] https://github.com/bcgit/bc-java/tree/master/core/src/main/java/org/bouncycastle/crypto/prng

[13] https://www.cvedetails.com/cve/CVE-2016-1000352/

References:

1. Androwarn Report

https://github.com/GitHubAssessments/CVE_Assessment_02_2019/blob/master/Signal_Androwarn_Report.pdf

2. MOBSF Report

https://github.com/GitHubAssessments/CVE_Assessment_02_2019/blob/master/Signal_MOBSF_Report.pdf

3. QARK Report

https://github.com/GitHubAssessments/CVE_Assessment_02_2019/blob/master/Signal_QARK_Report.pdf