CYBERSEC 2024
臺灣資安大會

Quark Script

Dig Vulnerabilities in the Black Box

KunYu Chen, YuShiang Dang, ShengFeng Lu
Telecom Technology Center

Quark Script

Creative & Innovative

- We dig vulnerabilities in the **black box**.
- We solve some **automation** problem.
- We provide useful APIs and expect **users** to use them in a creative and innovative way.

Dynamic & Static Analysis

- Quark script integrates both **static analysis** tools (e.g. Quark) and **dynamic analysis** tools (e.g. frida).

Re-Usable & Shareable

- Once the user creates a Quark script, it can be **used in different targets**.
- Once the user creates a Quark script, it can be **shared with other users**.

Quark Script Analysis Modules

String Module

Retrieve and analyze
strings
in an APK file.

Used in :
Quark Script CWE 22, 73, 798

Method Module

Detect, monitor and obtain
info about the specified methods
in an APK file.

Used in :
Quark Script CWE 22, 23, 73, 78, 79, 88, 295, 312, 328, 532, 601

Behavior Module

Define, detect and obtain
info about the defined behaviors
in an APK file.

Used in :
Quark Script CWE 20, 22, 23, 73, 78, 79, 88, 89, 94, 117, 319, 327, 338, 502, 749, 780, 798, 921, 940

Receiver Module

Retrieve and analyze
info about receivers (Android component)
in an APK file.

Used in :
Quark Script CWE 925

Activity Module

Retrieve and analyze
info about activities (Android component)
in an APK file.

Used in :
Quark Script CWE 926

Quark Script CWE-798

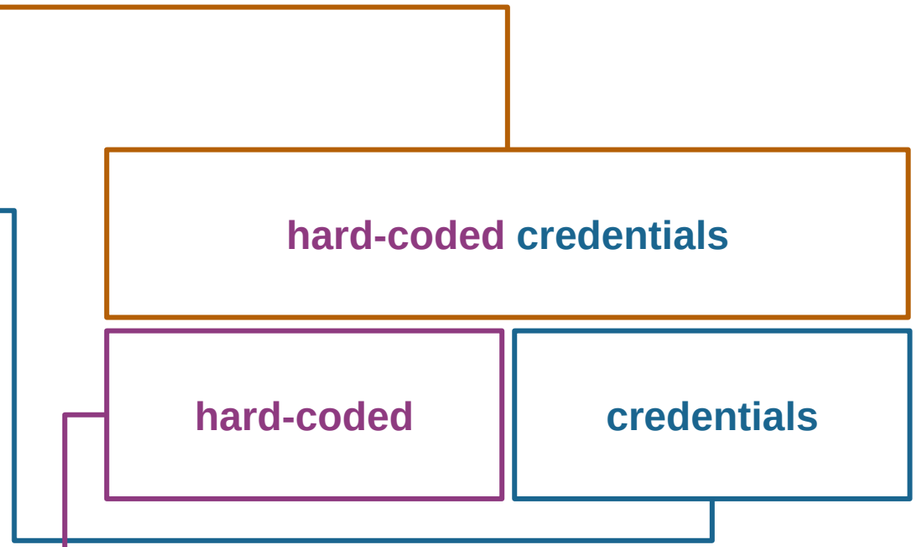
Use of Hard-coded Credentials

Definition of CWE-798

Title:
Use of Hard-coded Credentials

Description:
The product contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.

Extend Description:
Hard-coded credentials typically create a significant hole that allows an attacker to bypass the authentication that has been configured by the product administrator. This hole might be difficult for the system administrator to detect. Even if detected, it can be difficult to fix, so the administrator may be forced into disabling the product entirely.. (detailed discussion omitted)...



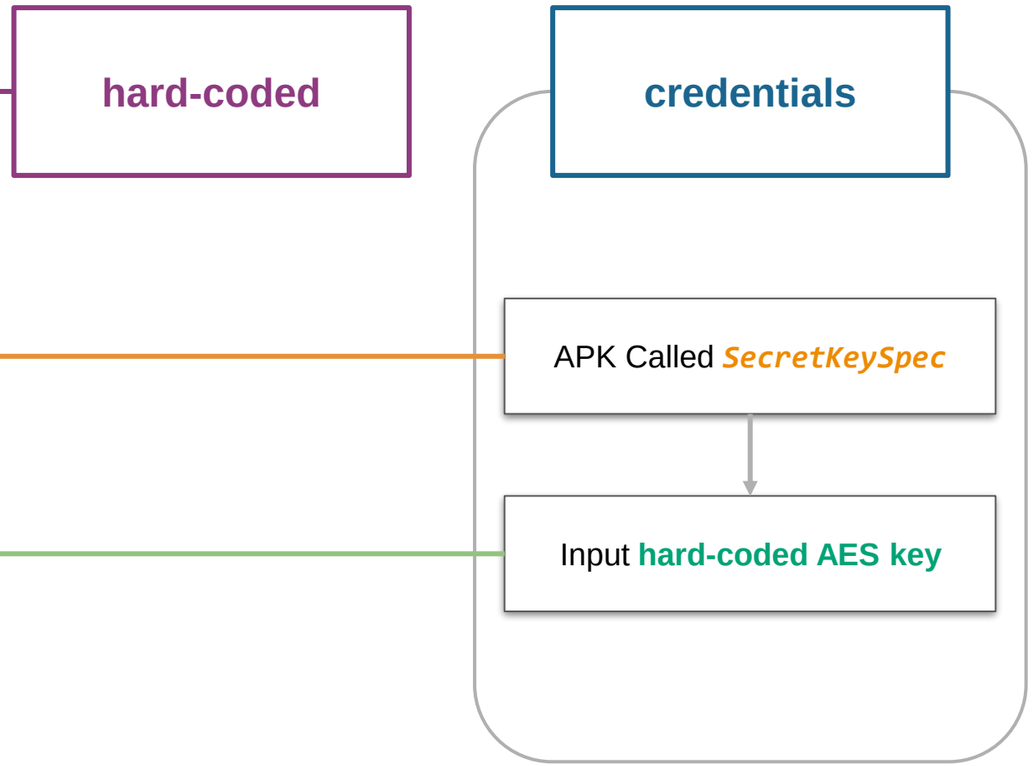
"Data or parameters in a program in such a way that they cannot be altered without modifying the program."

Source: Oxford Languages

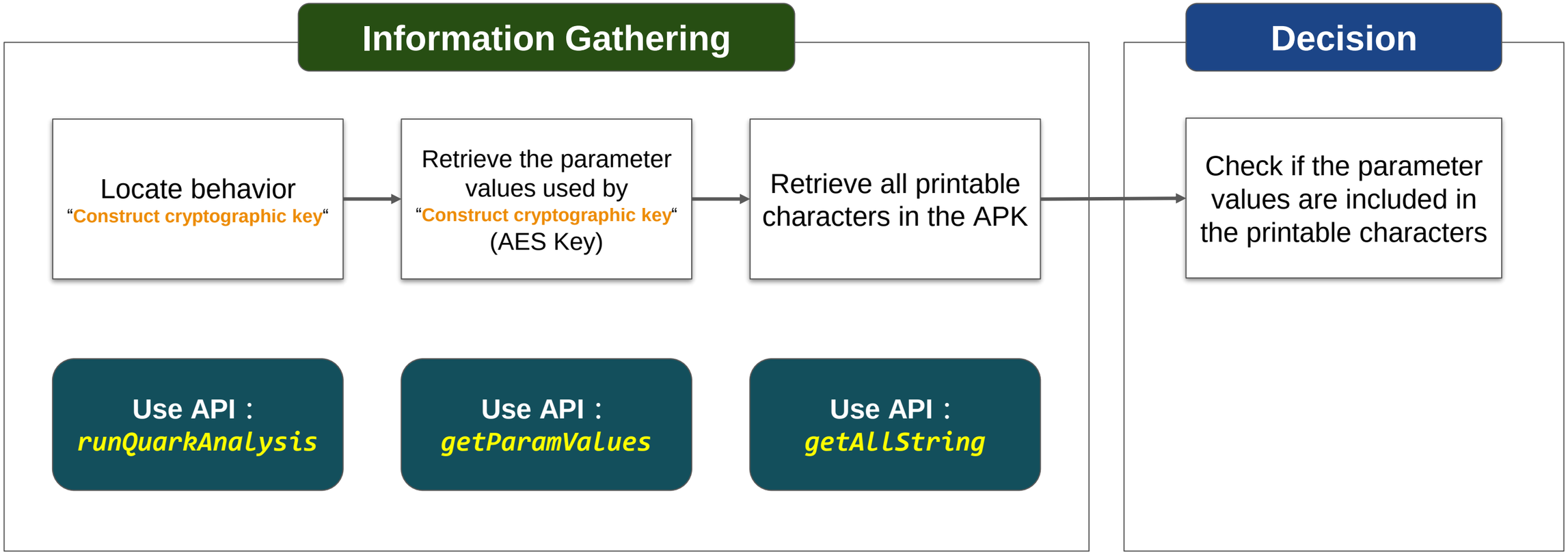
Source: cwe.mitre.org

ovaa.apk sample vulnerability code

```java
package oversecured.ovaa.utils;

import android.util.Base64;

import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

public class WeakCrypto {
    private static final String KEY = "49u5gh249gh24985ghf429gh4ch8f23f";

    private WeakCrypto() {
    }

    public static String encrypt(String data) {
        try {
            SecretKeySpec secretKeySpec = new SecretKeySpec(KEY.getBytes(), "AES");
            Cipher instance = Cipher.getInstance("AES");
            instance.init(Cipher.ENCRYPT_MODE, secretKeySpec);
            return Base64.encodeToString(instance.doFinal(data.getBytes()), 0);
        }
        catch (Exception e) {
            return "";
        }
    }
}
```

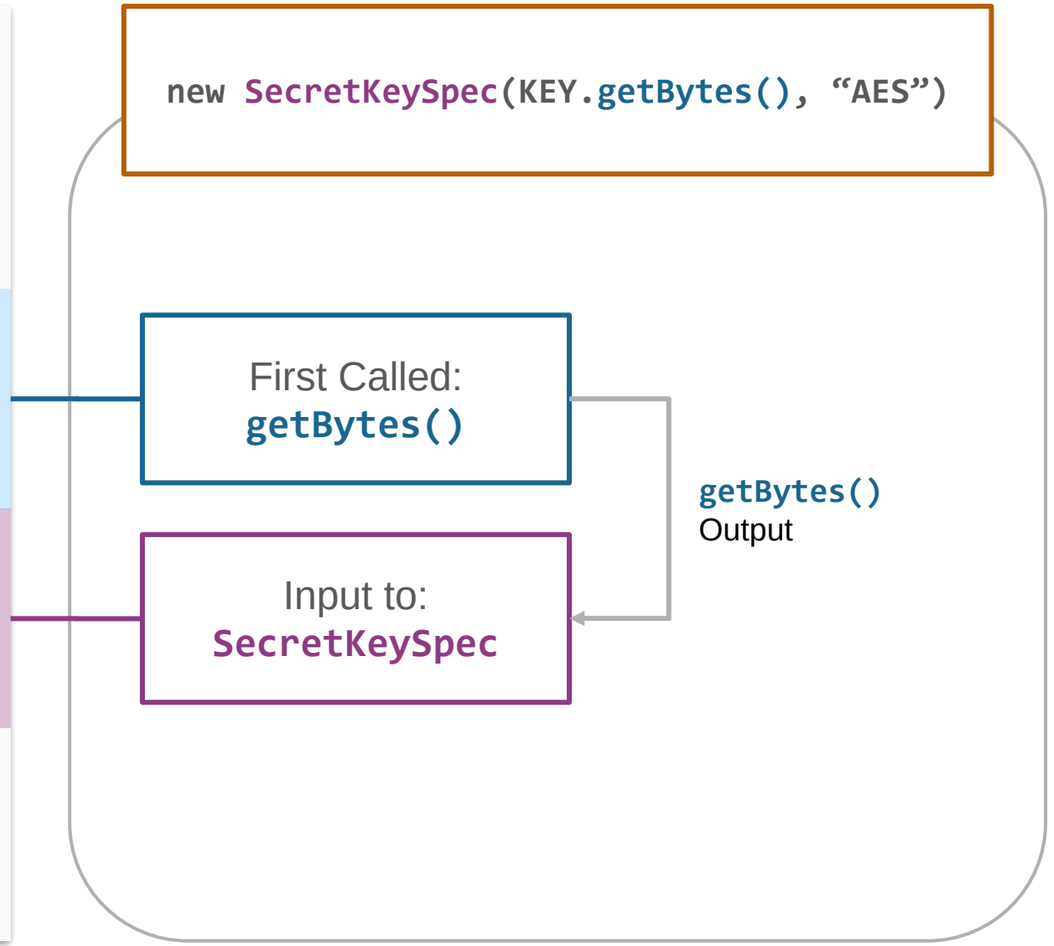


CWE-798 Detection Process using Quark Script API



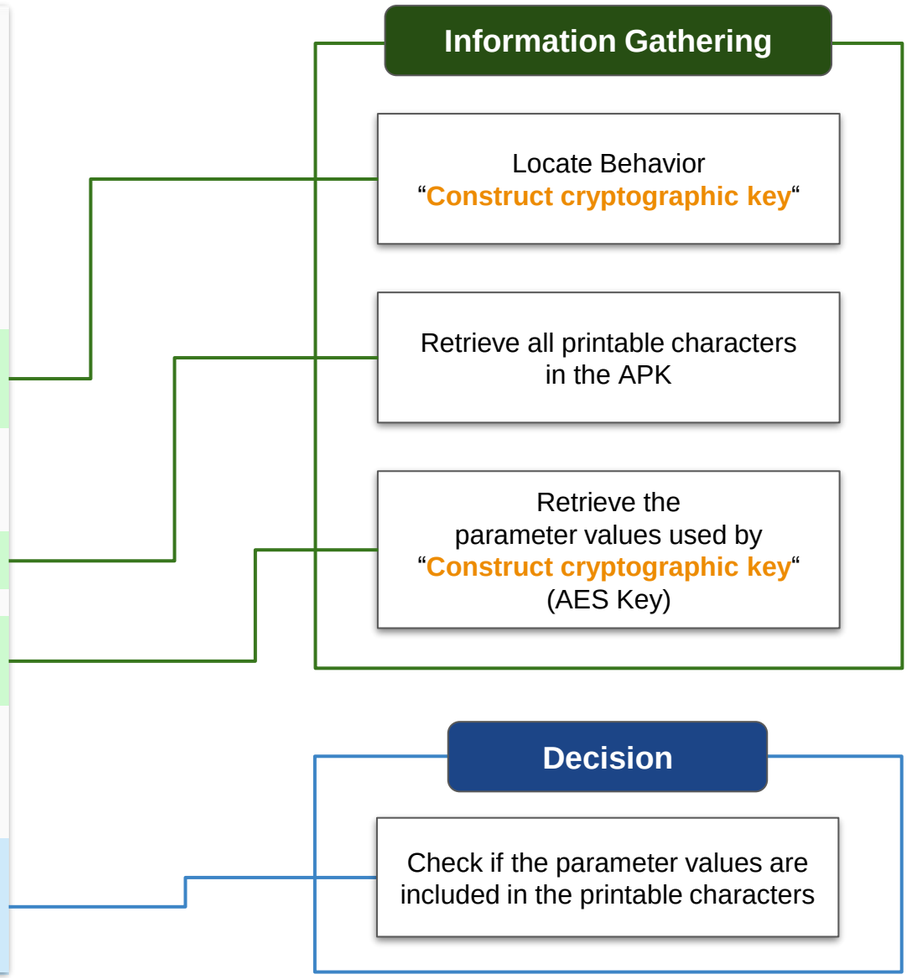
Behavior Define : Construct cryptographic key

```
{
    "behavior": "Construct cryptographic key",
    "permission": [],
    "api": [
        {
            "descriptor": "()[B",
            "class": "Ljava/lang/String;",
            "method": "getBytes"
        },
        {
            "descriptor": "([BLjava/lang/String;)V",
            "class": "Ljavax/crypto/spec/SecretKeySpec;",
            "method": "<init>"
        }
    ],
    "score": 1,
    "label": []
}
```



Quark Script for CWE-798 Detection

```python
import re
from quark.script import runQuarkAnalysis, Rule

SAMPLE_PATH = "ovaa.apk"
RULE_PATH = "constructCryptographicKey.json"

ruleInstance = Rule(RULE_PATH)
quarkResult = runQuarkAnalysis(SAMPLE_PATH, ruleInstance)

for secretKeySpec in quarkResult.behaviorOccurList:

    allStrings = quarkResult.getAllStrings()

    firstParam = secretKeySpec.getParamValues()[1]
    secondParam = secretKeySpec.getParamValues()[2]

    if secondParam == "AES":
        AESKey = re.findall(r'\((.*?)\)', firstParam)[1]

    if AESKey in allStrings:
        print(f"Found hard-coded {secondParam} key {AESKey}")
```



Quark Script CWE-312

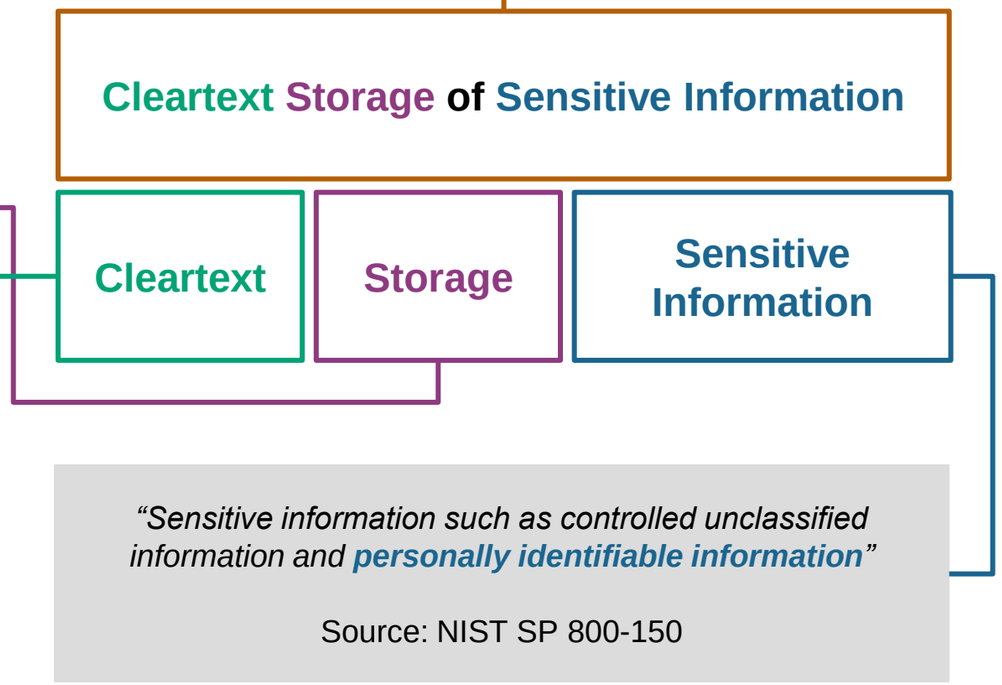
Cleartext Storage of Sensitive Information

Definition of CWE-312

Title:
Cleartext Storage of Sensitive Information

Description:
The application stores sensitive information in cleartext within a resource that might be accessible to another control sphere.

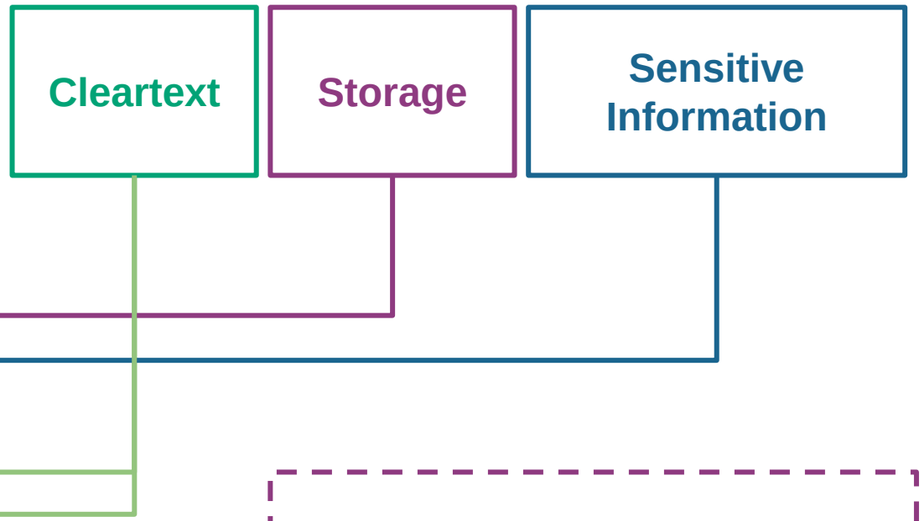
Extend Description:
Because the information is stored in cleartext (i.e., unencrypted), attackers could potentially read it. Even if the information is encoded in a way that is not human-readable, certain techniques could determine which encoding is being used, then decode the information.



Source: cwe.mitre.org

ovaa.apk sample vulnerability code

```
public class LoginUtils {
    private static final String EMAIL_KEY = "email";
    private static final String PASSWORD_KEY = "password";
    private static final String LOGIN_URL_KEY = "login_url";

    ...

    public void saveCredentials(LoginData loginData) {
        editor.putString(EMAIL_KEY, loginData.email)
                .putString(PASSWORD_KEY, loginData.password)
                .commit();
    }

    ...

}
```

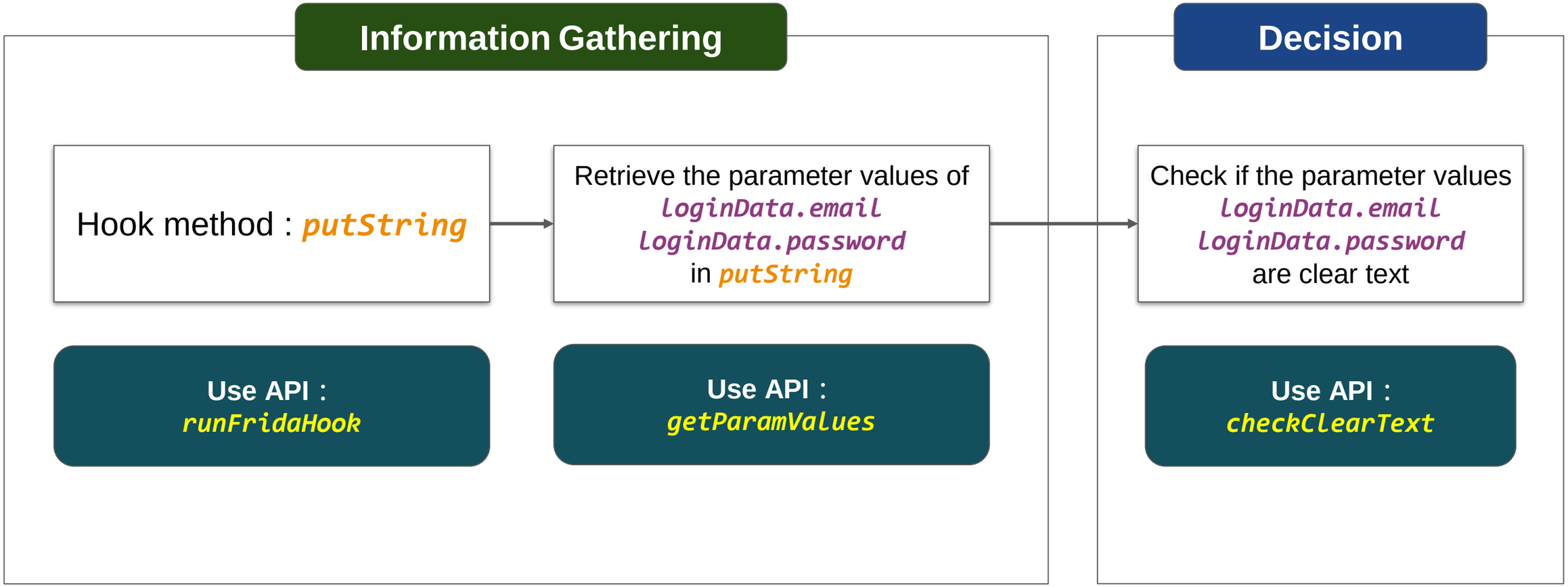


SharedPreferences.Editor.putString

SharedPreferences is best suited to **storing data** about how the user prefers to experience the app (detailed discussion omitted)...

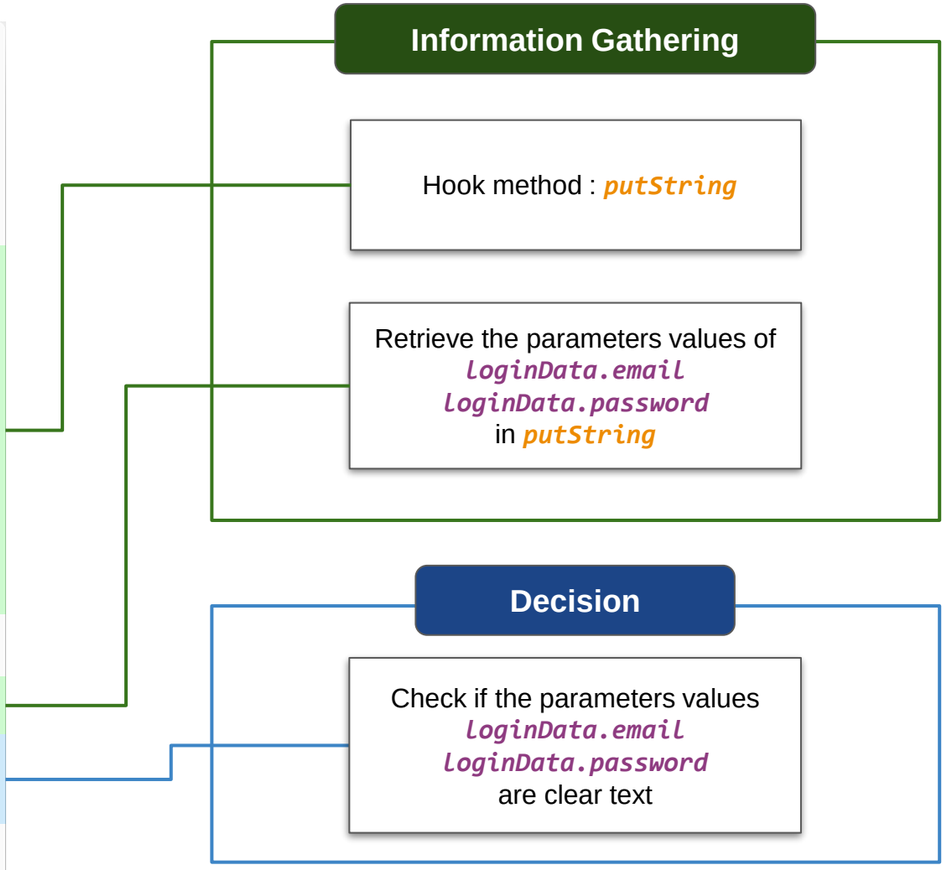
(Source: Android Developers)

CWE-312 Detection Process using Quark Script API



Quark Script for CWE-312 Detection

```
from quark.script.frida import runFridaHook
from quark.script.ciphey import checkClearText

APP_PACKAGE_NAME = "oversecured.ovaa"

TARGET_METHOD = "android.app." \
                "SharedPreferencesImpl$EditorImpl." \
                "putString"

METHOD_PARAM_TYPE = "java.lang.String," \
                    "java.lang.String"

fridaResult = runFridaHook(APP_PACKAGE_NAME,
                           TARGET_METHOD,
                           METHOD_PARAM_TYPE,
                           secondToWait = 10)

for putString in fridaResult.behaviorOccurList:

    firstParam, secondParam = putString.getParamValues()

    if firstParam in ["email", "password"] and \
        secondParam == checkClearText(secondParam):

        print(f'The CWE-312 vulnerability is found. The cleartext is "{secondParam}"')
```



Quark Script CWE-73

External Control of File Name or Path

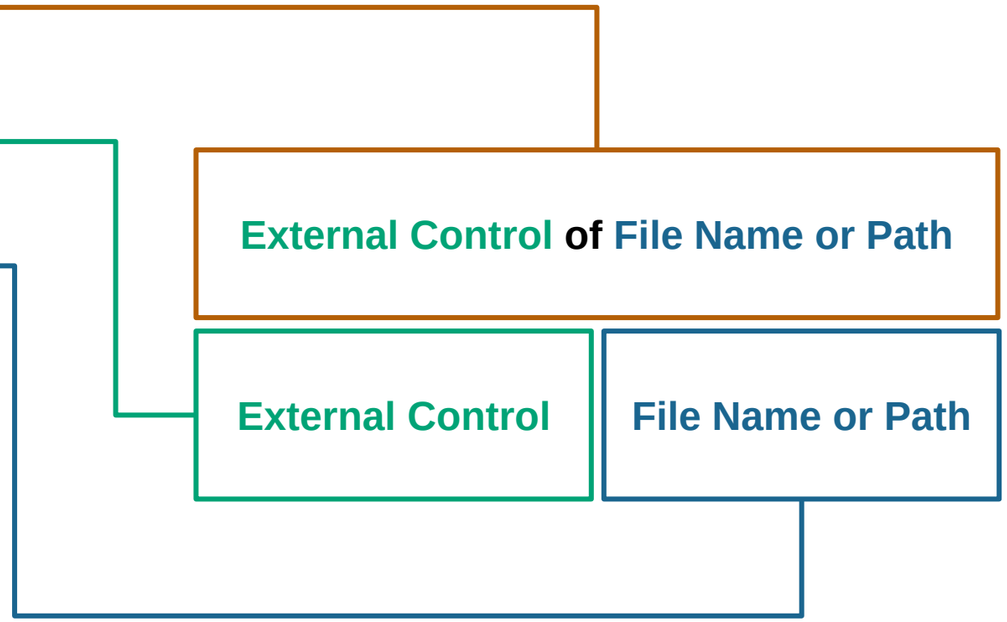
Definition of CWE-73

Title:
External Control of File Name or Path

Description:
The product allows user input to control or influence paths or file names that are used in filesystem operations.

Extend Description:
This could allow an attacker to access or modify system files or other files that are critical to the application.

For example, the program may give the attacker the ability to overwrite the specified file or run with a configuration controlled by the attacker.



Source: cwe.mitre.org

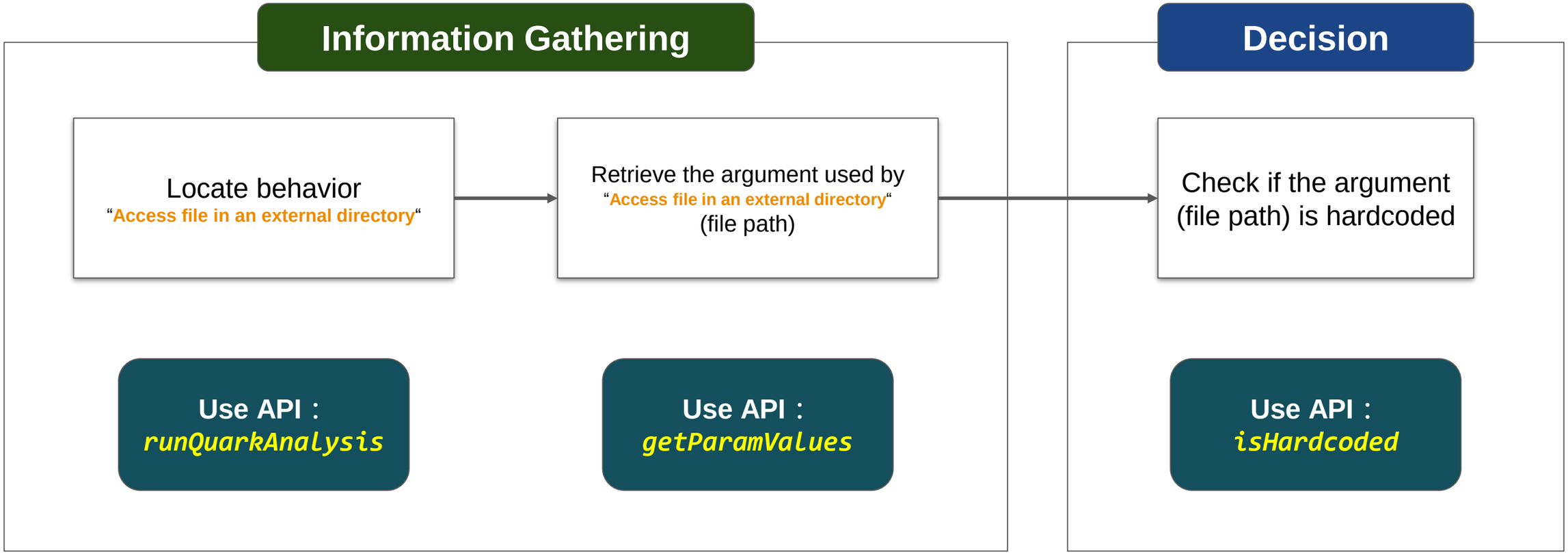
ovaa.apk sample vulnerability code

```
public class TheftOverwriteProvider extends ContentProvider {

    ...

    @Override
    public ParcelFileDescriptor openFile(
      @NonNull Uri uri,
      @NonNull String mode
    ) throws FileNotFoundException {

        File file = new File(Environment.getExternalStorageDirectory(), uri.getLastPathSegment());

        return ParcelFileDescriptor.open(file, ParcelFileDescriptor.MODE_READ_WRITE);

    }
}
```

External Control

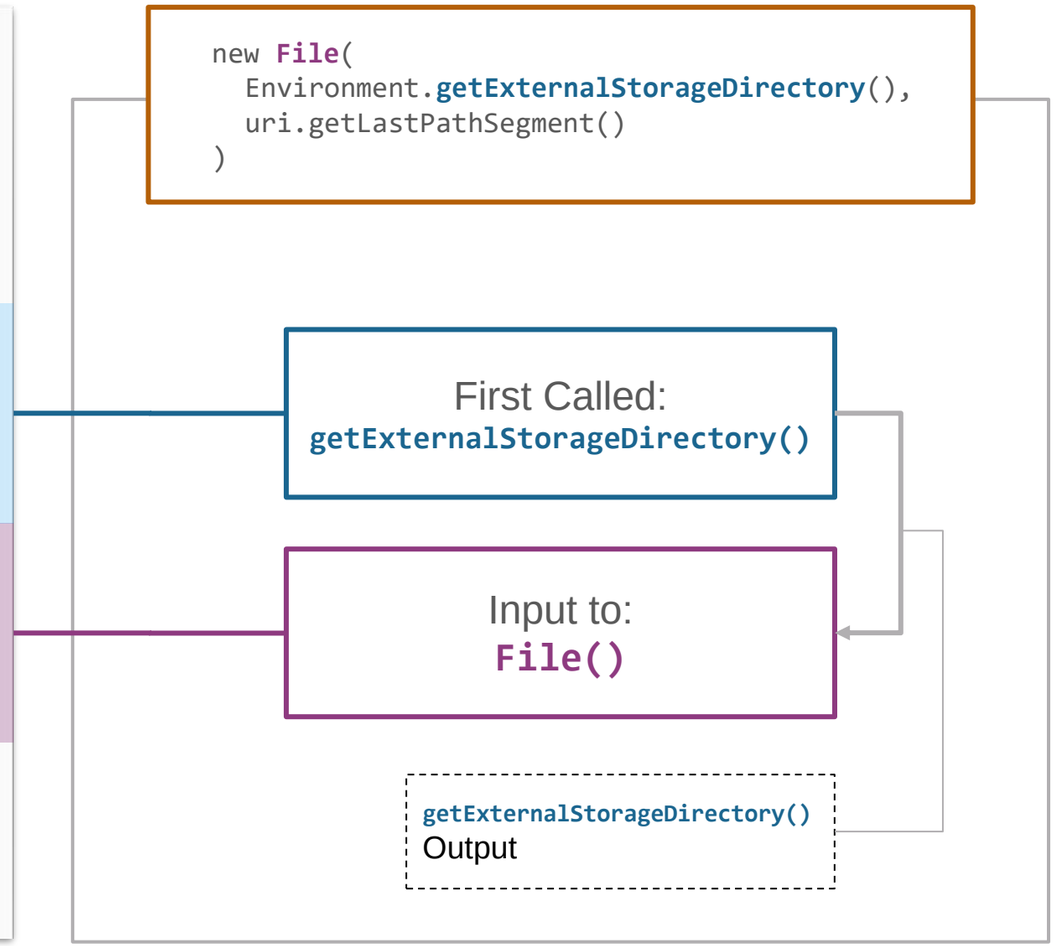
File Name or Path

CWE-73 Detection Process using Quark Script API



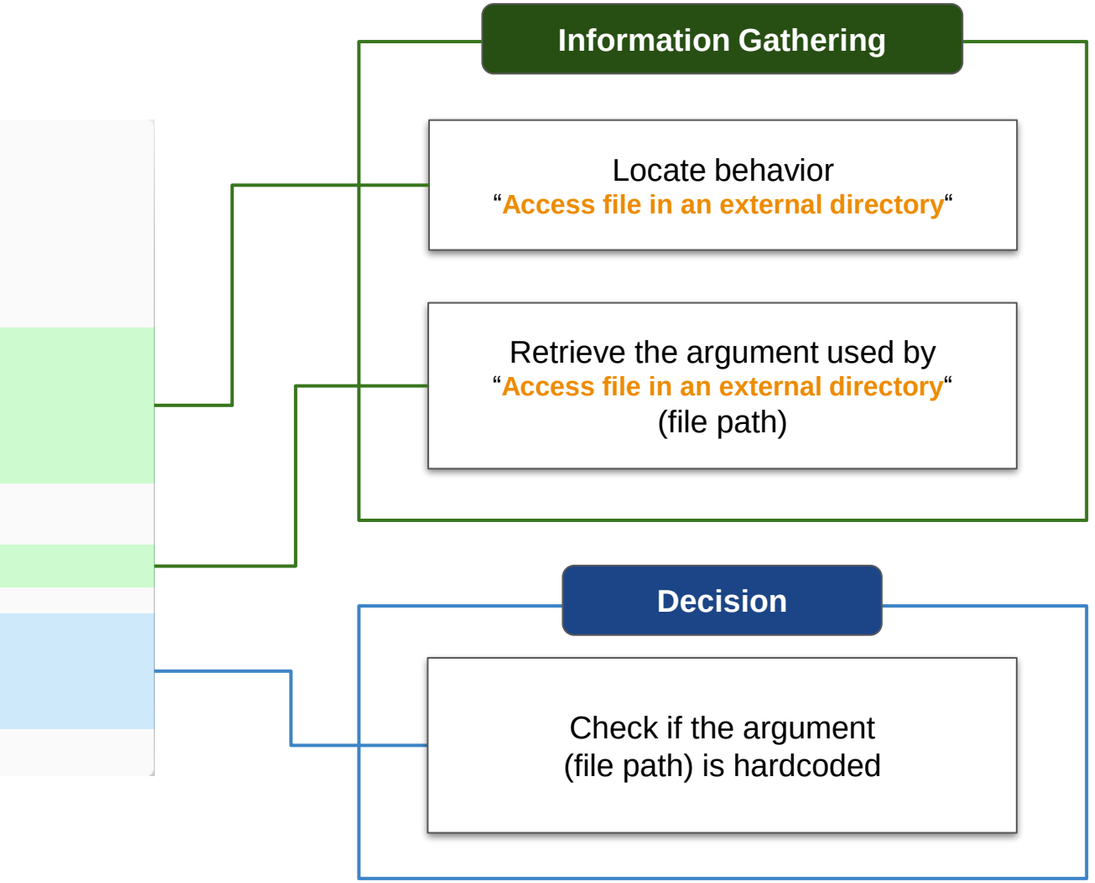
Behavior Define : Access file in an external directory

```
{
    "crime": "Access file in an external directory",
    "permission": [],
    "api": [
        {
            "class": "Landroid/os/Environment;",
            "method": "getExternalStorageDirectory",
            "descriptor": "()Ljava/io/File;"
        },
        {
            "class": "Ljava/io/File;",
            "method": "<init>",
            "descriptor": "(Ljava/io/File;Ljava/lang/String;)V"
        }
    ],
    "score": 1,
    "label": []
}
```



Quark Script for CWE-73 Detection

```
from quark.script import runQuarkAnalysis, Rule

SAMPLE_PATH = "ovaa.apk"
RULE_PATH = "accessFileInExternalDir.json"

ruleInstance = Rule(RULE_PATH)
quarkResult = runQuarkAnalysis(SAMPLE_PATH, ruleInstance)

for accessExternalDir in quarkResult.behaviorOccurList:
    filePath = accessExternalDir.secondAPI.getArguments()[2]

    if not quarkResult.isHardcoded(filePath):
        caller = accessExternalDir.methodCaller
        print("CWE-73 is detected in method, ", caller.fullName)
```



Quark Script CWE-89

Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Definition of CWE-89

Improper Neutralization of Special Elements used in an SQL Command

Title:

Improper Neutralization of Special Elements used in an SQL Command

Description:

The product constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component.

Extend Description:

Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the back-end database, possibly including execution of system commands.

Source: cwe.mitre.org

Improper Neutralization | **Special Elements** | **SQL Command**

Source: learn.microsoft.com

`;`	Query delimiter.
`'`	Character data string delimiter.
`--`	Single-line comment delimiter. Text following `--` until the end of that line isn't evaluated by the server.
`/*** ... ***/`	Comment delimiters. Text between `/*` and `*/` isn't evaluated by the server.
`xp_`	Used at the start of the name of catalog-extended stored procedures, such as `xp_cmdshell`.

*"The Database Language SQL (SQL) is a standard interface for accessing and **manipulating relational databases**."*

Source: NIST SP 800-8

AndroGoat.apk sample vulnerability code

```
override fun onCreate(savedInstanceState: Bundle?) {

        ...

        SQLibutton.setOnClickListener{
            var qry:String="SELECT * FROM users WHERE username='"+username.text.toString()+"'";

            try {
                  this.mDB = openOrCreateDatabase("aGoat", 0, null)
                  val QryResult = this.mDB!!.rawQuery(qry, null)

                  ...

              }
        }
    }
```

Improper Neutralization

Special Elements

SQL command

Source: learn.microsoft.com

`;`	Query delimiter.
`'`	Character data string delimiter.
`--`	Single-line comment delimiter. Text following `--` until the end of that line isn't evaluated by the server.
`/*** ... ***/`	Comment delimiters. Text between `/*` and `*/` isn't evaluated by the server.
`xp_`	Used at the start of the name of catalog-extended stored procedures, such as `xp_cmdshell`.

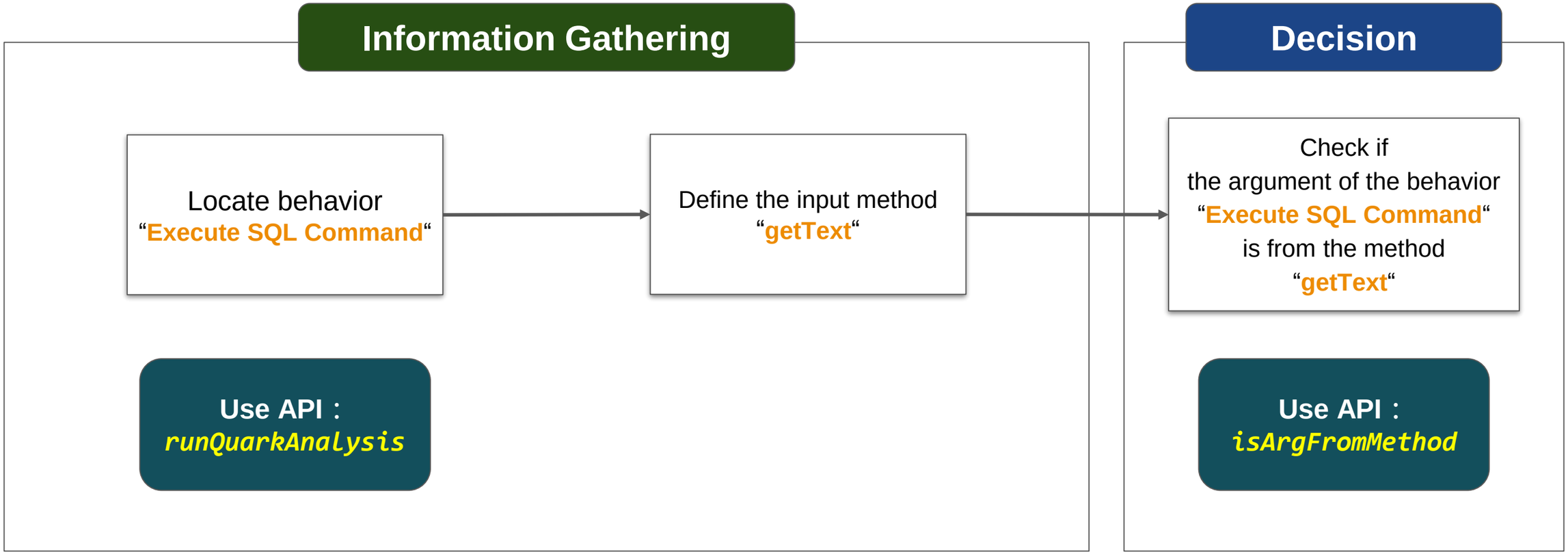
AndroGoat.apk sample vulnerability code

```
override fun onCreate(savedInstanceState: Bundle?) {

        ...

        SQLibutton.setOnClickListener{
            var qry:String="SELECT * FROM users WHERE username='"+username.text.toString()+"'";

            try {
                  this.mDB = openOrCreateDatabase("aGoat", 0, null)
                  val QryResult = this.mDB!!.rawQuery(qry, null)

                  ...

              }
        }
    }
```

Smali Code

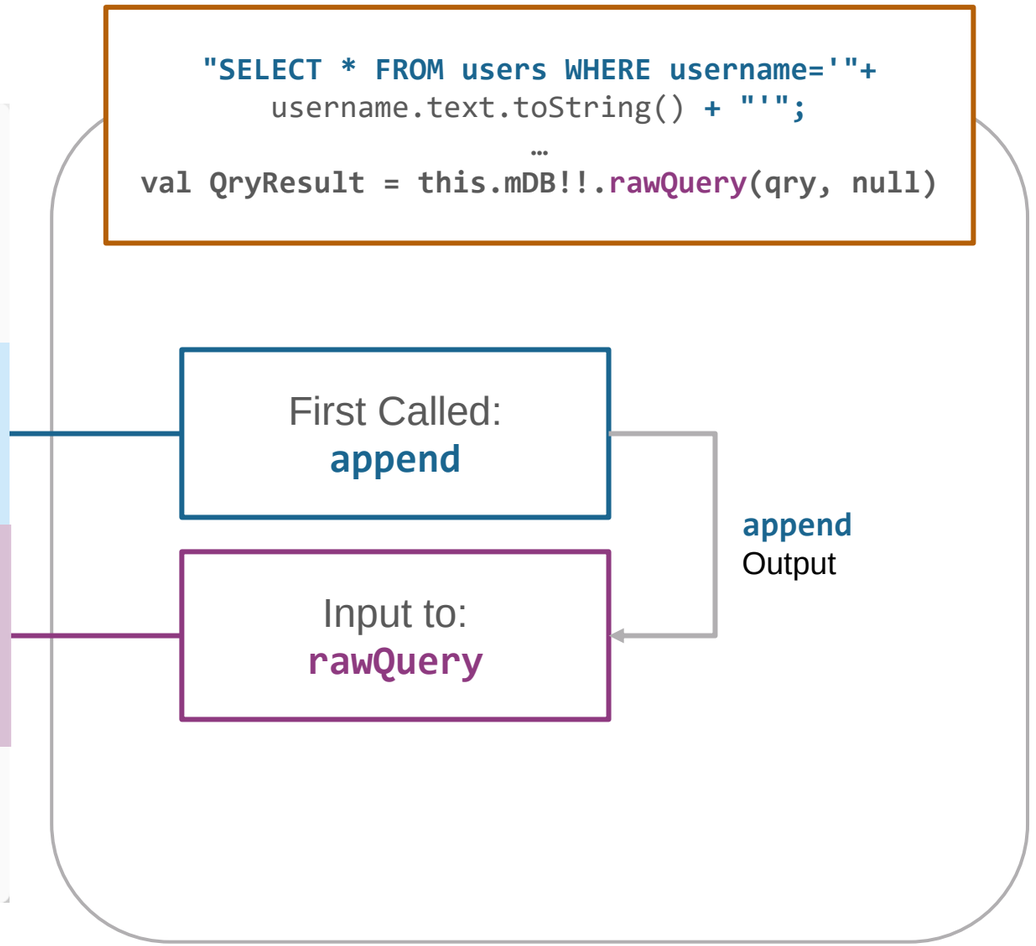
```
const-string v0, "SELECT * FROM users WHERE username=\'"

invoke-virtual {p2, v0}, Ljava/lang/StringBuilder;->append(Ljava/lang/String;)Ljava/lang/StringBuilder;

invoke-virtual {p0}, Landroid/widget/EditText;->getText()Landroid/text/Editable;
```

CWE-89 Detection Process using Quark Script API



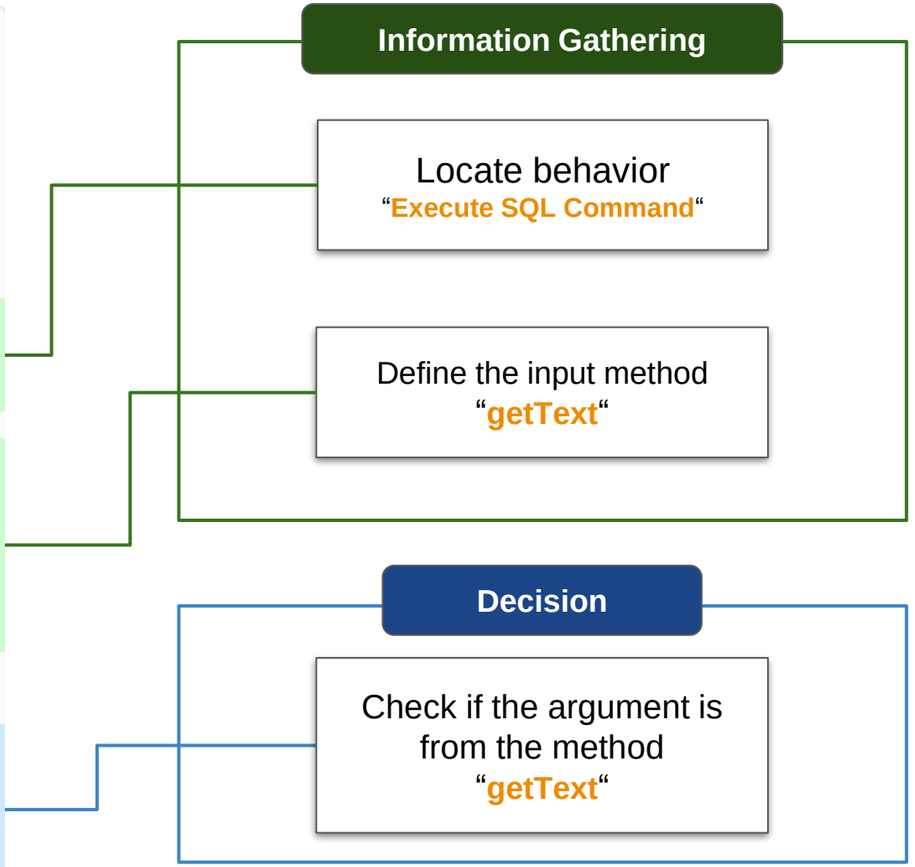
Behavior Define : Execute SQL Command

```
{
    "crime": "Execute SQL Command",
    "permission": [],
    "api": [
        {
            "class": "Ljava/lang/StringBuilder;",
            "method": "append",
            "descriptor": "(Ljava/lang/String;)Ljava/lang/StringBuilder;"
        },
        {
            "class": "Landroid/database/sqlite/SQLiteDatabase;",
            "method": "rawQuery",
            "descriptor": "(Ljava/lang/String;
                    [Ljava/lang/String;)Landroid/database/Cursor;"
        }
    ],
    "score": 1,
    "label": []
}
```



Quark Script for CWE-89 Detection

```python
from quark.script import runQuarkAnalysis, Rule

SAMPLE_PATH = "AndroGoat.apk"
RULE_PATH = "executeSQLCommand.json"

ruleInstance = Rule(RULE_PATH)
quarkResult = runQuarkAnalysis(SAMPLE_PATH, ruleInstance)

targetMethod = [
    "Landroid/widget/EditText;", # class name
    "getText",                   # method name
    "()Landroid/text/Editable;", # descriptor
]

for sqlCommandExecution in quarkResult.behaviorOccurList:
    if sqlCommandExecution.isArgFromMethod(
        targetMethod
    ):
        print(f"CWE-89 is detected in {SAMPLE_PATH}")
```



CYBERSEC 2024
臺灣資安大會

Welcome to our Discord channel

QUARK
40th Quark Release
2022 Anniversary Recap

Mar 23 New quark rules which detect camera control.
Mar 30 Quark-Engine V22.3.1.
Apr 06 New version of Detection Rules Viewer.
Apr 13 New quark rules which detect getting SMS messages via URIs.
Apr 20 New feature Radiocontrast.
Apr 27 Quark-Engine v22.4.1.
May 04 New quark rules which detect audio recording.
May 11 New quark rules which detect contact info accessing.
May 18 New Quark web report.
May 25 Quark-Engine v22.5.1.
Jun 01 Rule Generation Editor.
Jun 08 New quark rules which detect SMS sending.
Jun 15 New Rule Generation Feature.
Jun 22 New quark rules which detect screen capture.
Jun 29 Quark-Engine v22.6.1.
Jul 06 The Docs Enhancement Project.
Jul 13 BladeHawk Web Report.
Jul 20 Release Quark Script project.
Jul 22 CWE-798 Quark Script.
Jul 26 CWE-94 Quark Script.
Jul 28 CWE-921 Quark Script.
Jul 29 Quark-Engine v22.7.1.
Jul 31 CWE-312 Quark Script.
Sep 07 CWE-89 Quark Script.
Sep 14 CWE-926 Quark Script.
Sep 22 CWE-749 Quark Script.
Sep 29 Quark-Engine v22.9.1.
Oct 05 Release Quark MIT program.
Oct 12 CWE-532 Quark Script.
Oct 19 CWE-780 Quark Script.
Oct 26 Quark-Engine v22.10.1.
Nov 02 CWE-319 Quark Script.
Nov 09 Show Quick Start with CWE-798 Quark Script.
Nov 16 Spotlight dig of vulnerabilities in the blackbox.
Nov 24 CWE-327 Quark Script.
Nov 30 Quark-Engine v22.11.1.
Dec 07 Release Quark Script repo.
Dec 14 CWE-20 Quark Script.
Dec 22 CWE-79 Quark Script.

Thanks to

@PippenWang @xspiritualx1 @YushianhD @haeter525 @zorro_wang @sasakikung1
Telecom Technology Center

QUARK
93th Quark Release
2023 Anniversary Recap

January
Jan 06 CWE 328 Quark Script
Jan 12 CWE 489 Quark Script

February
Feb 01 CWE 22 Quark Script
Feb 10 Document find_previouse_method
Feb 15 Document find_intersection
Feb 22 Quark Engine v23.2.1

March
Mar 01 CWE 23 Quark Script
Mar 08 CWE 338 Quark Script
Mar 16 Document method_recursive_search
Mar 23 Document find_api_usage
Mar 29 Quark Engine v23.3.1

April
Apr 12 Document evaluate_method
Apr 19 Document check_parameter_on_single_method
Apr 27 Quark Engine v23.4.1

May
May 03 CWE 88 Quark Script
May 12 CWE 925 Quark Script
May 18 Document check_parameter
May 27 Document check_parameter_values
May 31 CWE 73 Quark Script
May 31 Quark Engine v23.5.1

June
Jun 11 Document check_sequence
Jun 17 CWE 78 Quark Script
Jun 21 Document run
Jun 28 Quark Engine v23.6.1

July
Jul 05 Document get_json_report
Jul 12 Document generate_json_report
Jul 19 Document add_table_row
Jul 26 Quark Engine v23.7.1

August
Aug 02 Document show_summary_report
Aug 10 CWE 117 Quark Script
Aug 19 Release Visual Quark Script Program
Aug 25 CWE 940 Quark Script
Aug 31 Quark Engine v23.8.1

September
Sep 08 Document show_label_report
Sep 14 Document show_detail_report
Sep 23 Update Quark Script Visualization
Sep 28 Quark Engine v23.9.1

October
Oct 07 Document show_call_graph
Oct 14 Document show_rule_classification
Oct 20 Method overview quark.core.quark
Oct 27 Quark Engine v23.10.1

November
Nov 03 Document wrapper_lookup
Nov 09 Document show_comparison_graph
Nov 15 CWE 502 Quark Script
Nov 23 Document call_graph
Nov 29 Quark Engine v23.11.1

December
Dec 06 First Draft of Quark Script Visualization web layout
Dec 13 Document select_label _menu
Dec 23 CWE 601 Quark Script
Dec 27 Quark Engine v23.12.1

THANKS TO

@PippenWang @xspirtualx1 @YushianghD @haeter525 @zorro_wang
@sasakikung1 @PoyenLiang @NinaWeng_ @oraoraora947 @Kai_Shiang_605
Telecom Technology Center

資通安全檢測服務

具法令效力

關鍵電信基礎設施資通設備：路由器、交換器、防火牆

符合美國FDA、歐盟MDCG、國內TFDA之醫療器材網路安全規定

產業推動

無線寬頻分享器
機上盒

5G基地臺

智慧音箱

網路攝影機

行動應用APP
智慧型手機系統內建軟體

無人機

物聯網場域資安防護評估

政府委託

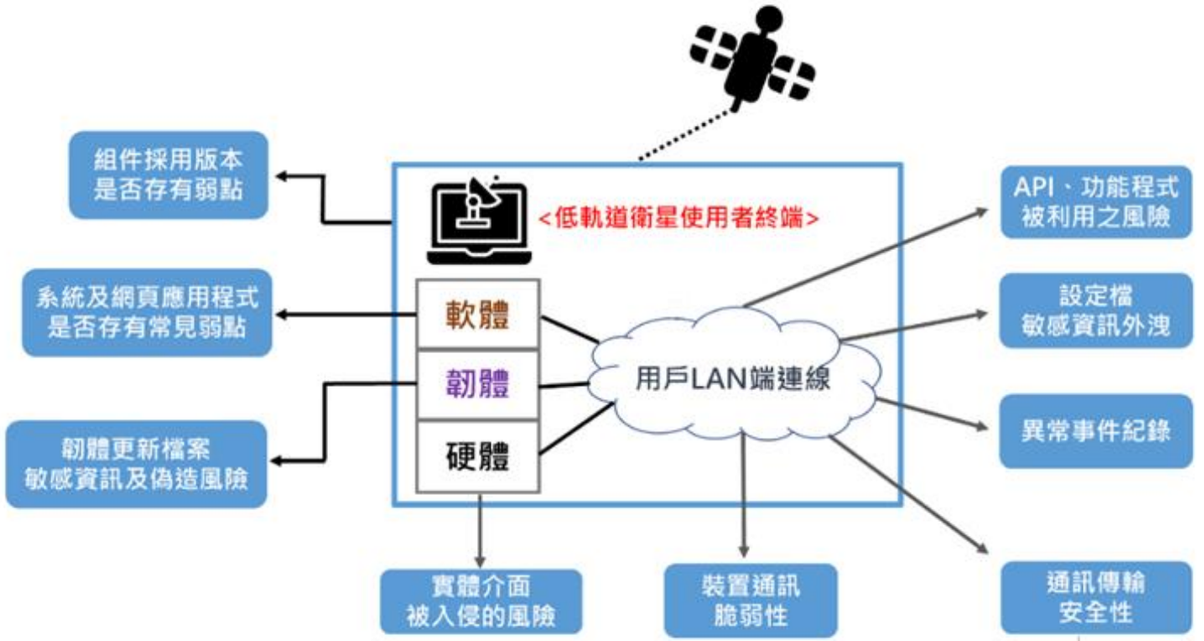
政府機關委託進階
資安測試分析與調查

低軌道衛星使用者終端資安標準及測試規範

臺灣低軌道衛星資安產業標準/指引首部曲

- 已透過TAICS於113年4月26日正式公告為產業標準文件*
- 適用範圍為低軌道衛星使用者終端主機本體，包括硬體、韌體、輸出入介面、傳輸協定以及終端內部通訊網路的區域網路Local Area Network (LAN) 端所提供之服務系統介面等。
- 參照衛星地球電臺設備技術規範，衛星地球電臺其設備包括射頻設備及天線，如非同步衛星之使用者終端設備，若架構雷同亦可適用本規範。

開放測試中
歡迎預約




太空系統網路安全研究暨測試實驗室
Space Systems cybersecurity research and testing laboratory

*台灣資通產業標準協會出版資訊：https://www.taics.org.tw/Publishing.aspx?PubCat_id=2