Kunal pandey Follow
Apr 29, 2020 · 4 min read · Listen

Indirect UXSS issue on a private Android target app

Hello Everyone,

It's been a while since I've done some writeups. So, here is the write-up about Indirect UXSS on a private integrated browser on a crypto app.

Around last year August, I got an invitation to a private program on H1. While looking at the scopes, I saw the Crypto android app.

While navigating to the app, there were different functionalities like sending ETH to any other user and also their integrated browser which users can normally use it as a normal browser to navigate.

Explanation

First, I reversed the app using *JADX-GUI* and gone through AndroidManifest.xml to check if any custom schema has been declared or not.

Well, I was in luck and saw there was custom schema implemented:

> *<data android:host="browse" android:pathPrefix="/" android:scheme="example"/>*

To get a better understanding of what this schema is used for, I searched for their Github Organization's Repo and look for commit files and saw it was used for website navigation into the integrated browser.

E.g *example://browse/example.com*

So, I tried to href link in chrome to test functionality

example://browse/https://google.com

[Note: At this point, I was first trying to check basic schema https or HTTP to trigger "browse" path on deeplink]

click here

click here

Fig 1 — Just an href link

Next, Clicked on the href link, and Url address has been triggered on the Integrated Browser.

[Note: You can also use an integrated browser normally like inserting URL address and navigate to any web page.]

Fig 2- Triggered google website using deeplink

So, now everything cleared up, I started to inject normally to check javascript schema XSS.

example://browse/javascript:alert(1) → Didn't triggered

Next, Added Double encoded CRLF character and triggered using href deeplink on chrome

click here → triggered on integrated app

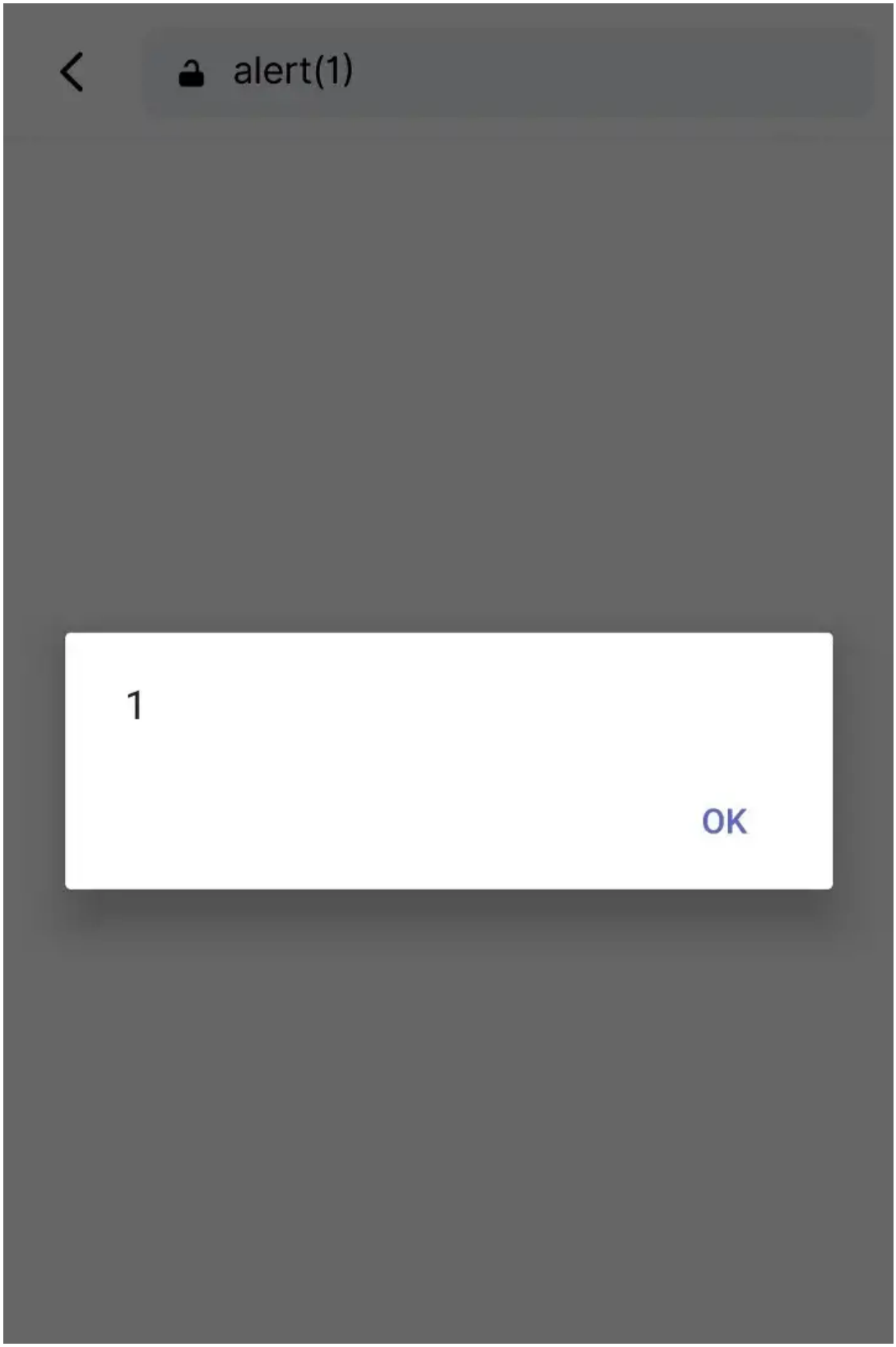

Fig 3- Testing javascript schema on deeplink

As XSS got triggered by custom deeplink, my first attempt was to trigger Direct UXSS on any site, in that case, I've tried to use the window.open() function to trigger UXSS including both protocols (https and deeplink schema) using setTimeout function, creating a web page, and navigate normally by inserting URL address on the integrated browser to test.

Attempt 1

```
<html>
<head>
<script>
function spoof() {
u = window.open("https://www.google.com");
setTimeout(() => { t = u.location.replace('example://browse/javascript://%250dalert(1)'); }, 1000)
}
</script>
</head>
<body><button onclick="spoof()"style="background-color:#4CAF50;color:white;font-family:Tahoma;font-size:40px;height:120px; width:400px">click here</button>
</body>
</html>
```

Turns out It didn't work and got failed because once setTimeout function was trying to implement deeplink protocol over the page of "google.com", the app got crashed.

Why I choose this technique?

I thought maybe if the integrated browser can try to override with the loaded web page using custom deeplink and can bypass SOP, But turns out it didn't work, my theory failed and the app got crashed.

Well, if it's not for direct UXSS, we can still perform indirect UXSS. :)

Second Theory — If there is already an opened website on the integrated browser (As a normal user, I have already opened the google.com web page on the integrated browser), and what if we trigger Javascript XSS using Custom deeplink and click on it, will there be indirect UXSS on an integrated browser?

- Well, it worked :)

I've already opened google.com website on the integrated browser and on chrome, clicked on href link as "*example://browse/javascript://%250dalert(document.domain)"* and the result was:

Fig 4-Indirect UXSS using deeplink

In the above image, we can see indirect UXSS has been triggered on google.com.

I've submitted this bug and got a reward of 1K USD.

Patch Work

Currently, they've strictly enforced only HTTP and HTTPS protocol on deeplink functionality, so anytime if you'll navigate using "example://browse/javascript://%250dalert(1)"

It'll automatically convert into http://javascript://%0dalert(1), so no more javascript XSS :)

Key points during this write-up

- Discussed failed attempts to get a better understand of the context.
- Mentioned integrated browser functionality as a normal browser so that anyone can get an idea that inserting URL address and navigation with the webpage is also possible and also, using deeplink, it's also possible to navigate.

Security Infosec