CVE-2023-4357-Exploitation - Report

Name	Chromium XXE
ID	CVE-2023-4357
Severity	Medium
Type	XXE
Exploit Difficulty	Low

Introduction

In versions of Google Chrome prior to 116.0.5845.96, a critical vulnerability was discovered, arising from the insufficient validation of untrusted input during XML processing.

This issue is further compounded when considering the use of Libxslt, the default XSL transformation library employed in WebKit-based browsers like Google Chrome, Safari, and others. Libxslt allows external entities within documents that are loaded by the XSL document() method. This characteristic of Libxslt facilitates an exploitation technique whereby an attacker can bypass security restrictions to access file:// URLs from http(s):// URLs, thus gaining unauthorized file access.

The default sandbox settings in these browsers do not entirely mitigate this risk; for instance, it permits reading of the /etc/hosts file on various operating systems and platforms including iOS (Safari/Chrome), macOS (Safari/Chrome), Android (Chrome), and even on Samsung TV's default browser.

The situation becomes even more precarious when the -no-sandbox attribute is used, a scenario often seen in applications built on frameworks like Electron or PhantomJS. In such cases, the attacker's capacity to read files is not limited to certain locations but extends to any file on the operating system, posing a grave security threat across different platforms and devices.

Reproduced the Environment

In order to experiment with this exploitation on Ubuntu, I am planning to set up a **Virtual Machine running Ubuntu**.

Additionally, it will be necessary to download and install **Google Chrome** on the Ubuntu VM because the exploitation technique specifically targets a vulnerability present in versions of Chrome prior to 116.0.5845.96.

Reproduce the Exploitation

To reproduce the exploitation of the vulnerability identified in certain versions of Google Chrome, involving insufficient validation of untrusted XML input, a detailed approach is required. The following steps outline the process to be followed to effectively demonstrate the exploitation under a controlled environment:

1. **Execute Google Chrome on the Ubuntu Virtual Machine:**

   This step ensures that we are working within a realistic scenario where the browser's vulnerability can be exploited.

2. **Start a Web Service Designed to Exploit the Vulnerability:**

   This involves deploying a server-side application that serves a specially crafted page which will contain malicious XML content designed to test the vulnerability by attempting to bypass Chrome's file access restrictions.

3. **Access the Browser to Interact with the Malicious Web Service:**

   If the exploitation is successful, it will demonstrate the vulnerability by bypassing the intended file access restrictions, thereby confirming the potential for unauthorized access or manipulation of system files and data.