# 浏览器专题一期报告

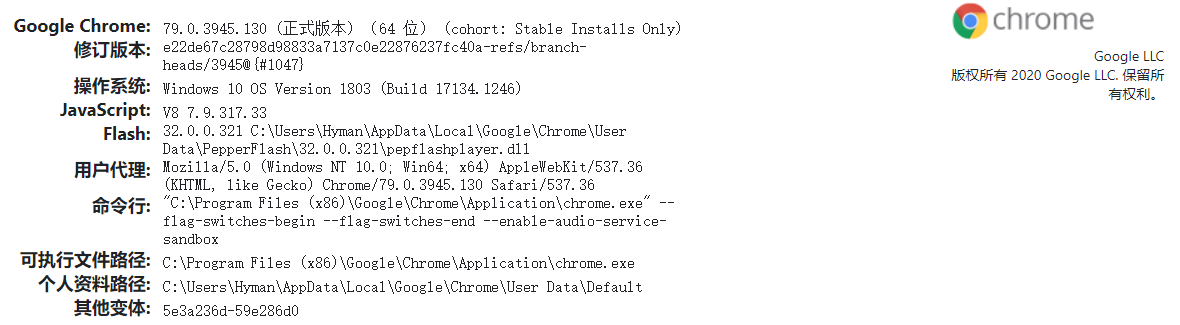
1. 锦帛浏览器内核详细信息

锦帛浏览器是基于chromium极简内核进行开发。Chromium项目为chrome浏览器的开源版本，由于目前助手厂商未提供详细内核版本号，故可以参考Chrome43版本的内核信息，对目前锦帛浏览器内核信息进行追溯，后续联系厂商寻求更准确的相关信息。

本地安装Chrome 43.0.2357.132版本进行调试：在chrome浏览器的地址栏输入<chrome://version/>即可查看浏览器的底层信息



对比目前正式稳定版79:



1. 锦帛浏览器与Chrome43浏览器以及Chrome最新版浏览器的差异
2. 功能性差异
   1. 对于HTML5、CSS3、ES6特性支持差异：

主要使用以下两个站点进行对比，总结异同点结果：

[**http://html5test.com/**](http://html5test.com/)

**https://www.caniuse.com/#compare=chrome+43,chrome+82**

**html5test对比结果:**

|  |  |  |
| --- | --- | --- |
| BROWSERS | **Chrome 43** | **My browser79** |

| **parsing rules** |  |  |
| --- | --- | --- |
| <!DOCTYPE html> triggers standards mode | Yes ✔ | Yes ✔ |
| HTML5 tokenizer | Yes ✔ | Yes ✔ |
| HTML5 tree building | Yes ✔ | Yes ✔ |
|  |  |  |
| Parsing inline SVG | Yes ✔ | Yes ✔ |
| Parsing inline MathML | Yes ✔ | Yes ✔ |

| **elements** |  |  |
| --- | --- | --- |
| Embedding custom non-visible data | Yes ✔ | Yes ✔ |
| Section elements | Yes ✔ | Yes ✔ |
| Grouping content elements | Yes ✔ | Yes ✔ |
| Text-level semantic elements | Partial ○ | Yes ✔ |
| Interactive elements | Partial ○ | Partial ○ |
| hidden attribute | Yes ✔ | Yes ✔ |
| Dynamic markup insertion | Yes ✔ | Yes ✔ |

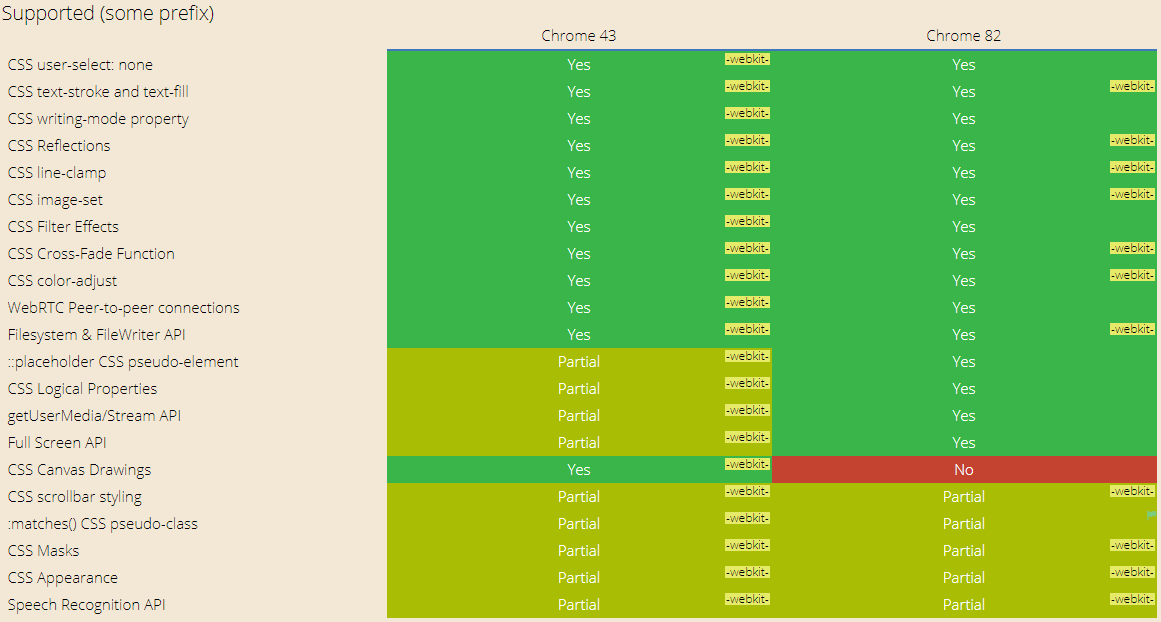
| **forms** |  |  |
| --- | --- | --- |
| input type=text | Yes ✔ | Yes ✔ |
| input type=search | Yes ✔ | Yes ✔ |
| input type=tel | Yes ✔ | Yes ✔ |
| input type=url | Yes ✔ | Yes ✔ |
| input type=email | Yes ✔ | Yes ✔ |
| input type=date | Yes ✔ | Yes ✔ |
| input type=month | Yes ✔ | Yes ✔ |
| input type=week | Yes ✔ | Yes ✔ |
| input type=time | Yes ✔ | Yes ✔ |
| input type=datetime-local | Yes ✔ | Yes ✔ |
| input type=number | Yes ✔ | Yes ✔ |
| input type=range | Yes ✔ | Yes ✔ |
| input type=color | Yes ✔ | Yes ✔ |
| input type=checkbox | Yes ✔ | Yes ✔ |
| input type=image | Yes ✔ | Yes ✔ |
| input type=file | Partial ○ | Partial ○ |
| textarea | Yes ✔ | Yes ✔ |
| select | Yes ✔ | Yes ✔ |
| fieldset | Yes ✔ | Yes ✔ |
| datalist | Yes ✔ | Yes ✔ |
| output | Yes ✔ | Yes ✔ |
| progress | Yes ✔ | Yes ✔ |
| meter | Yes ✔ | Yes ✔ |
| Field validation | Yes ✔ | Yes ✔ |
| Association of controls and forms | Yes ✔ | Yes ✔ |
| Other attributes | Yes ✔ | Yes ✔ |
| CSS selectors | Yes ✔ | Yes ✔ |
| Events | Yes ✔ | Yes ✔ |
| Form validation | Yes ✔ | Yes ✔ |

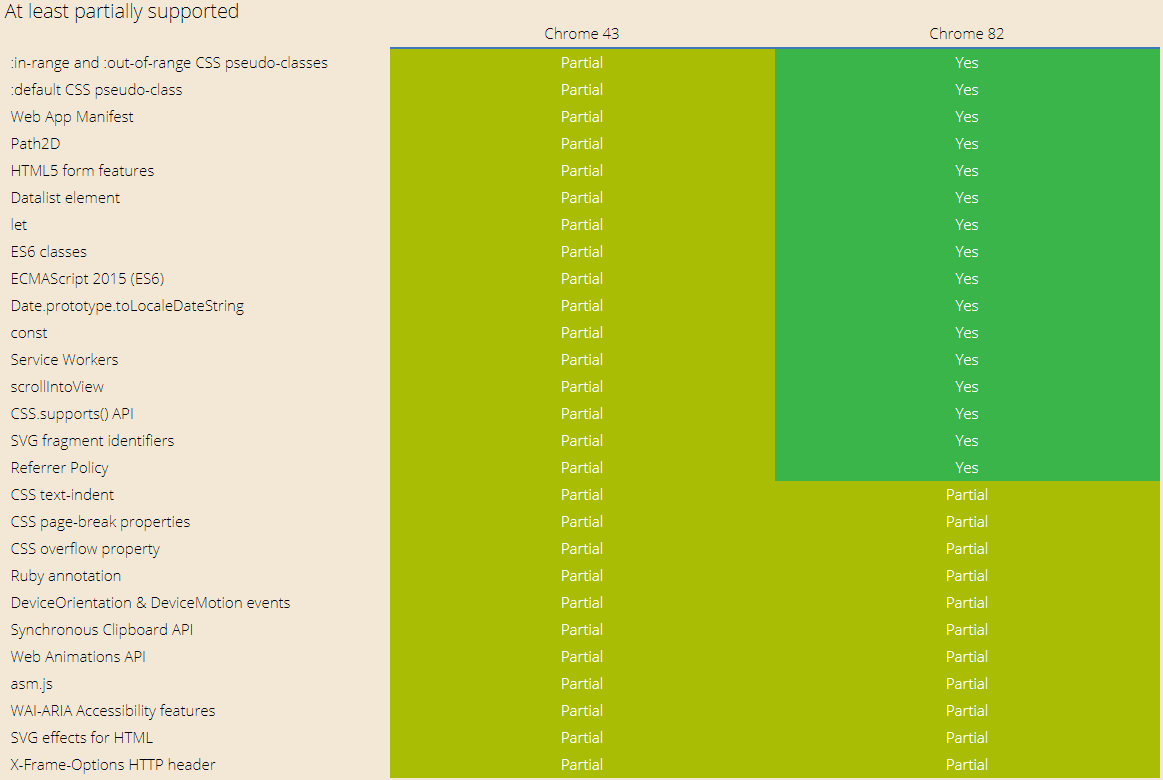
| **web components** |  |  |
| --- | --- | --- |
| Custom elements | Yes ✔ | Yes ✔ |
| Shadow DOM | Partial ○ | Yes ✔ |
| HTML templates | Yes ✔ | Yes ✔ |
| HTML imports | Yes ✔ | Yes ✔ |

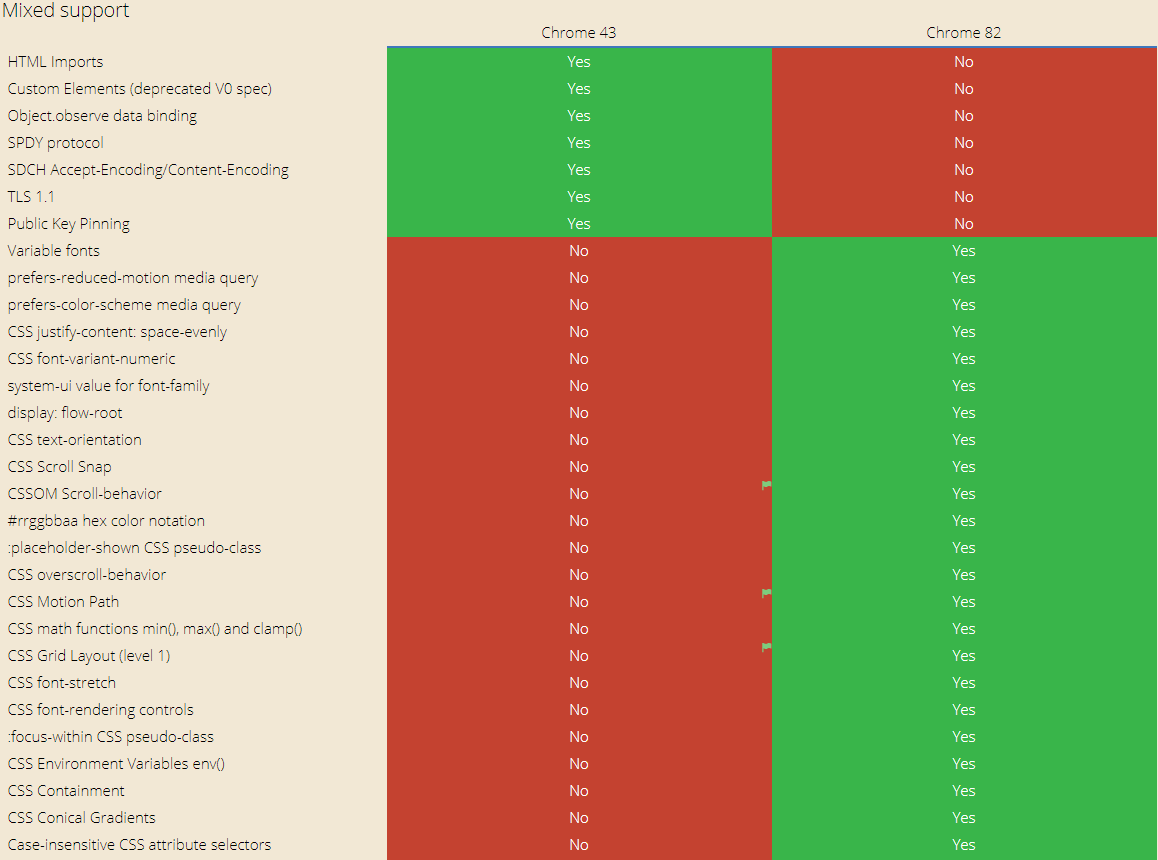
| **security** |  |  |
| --- | --- | --- |
| Web Cryptography API | Yes ✔ | No ✘ |
| Content Security Policy 1 | Yes ✔ | Yes ✔ |
| Content Security Policy 2 | Yes ✔ | Yes ✔ |
| Cross-Origin Resource Sharing | Yes ✔ | Yes ✔ |
| Subresource Integrity | No ✘ | Yes ✔ |
| Cross-document messaging | Yes ✔ | Yes ✔ |
| Web Authentication / FIDO 2 | No ✘ | No ✘ |
| Credential Management | No ✘ | No ✘ |
| Sandboxed iframe | Yes ✔ | Yes ✔ |
| iframe with inline contents | Yes ✔ | Yes ✔ |
| **ES6** |  |  |

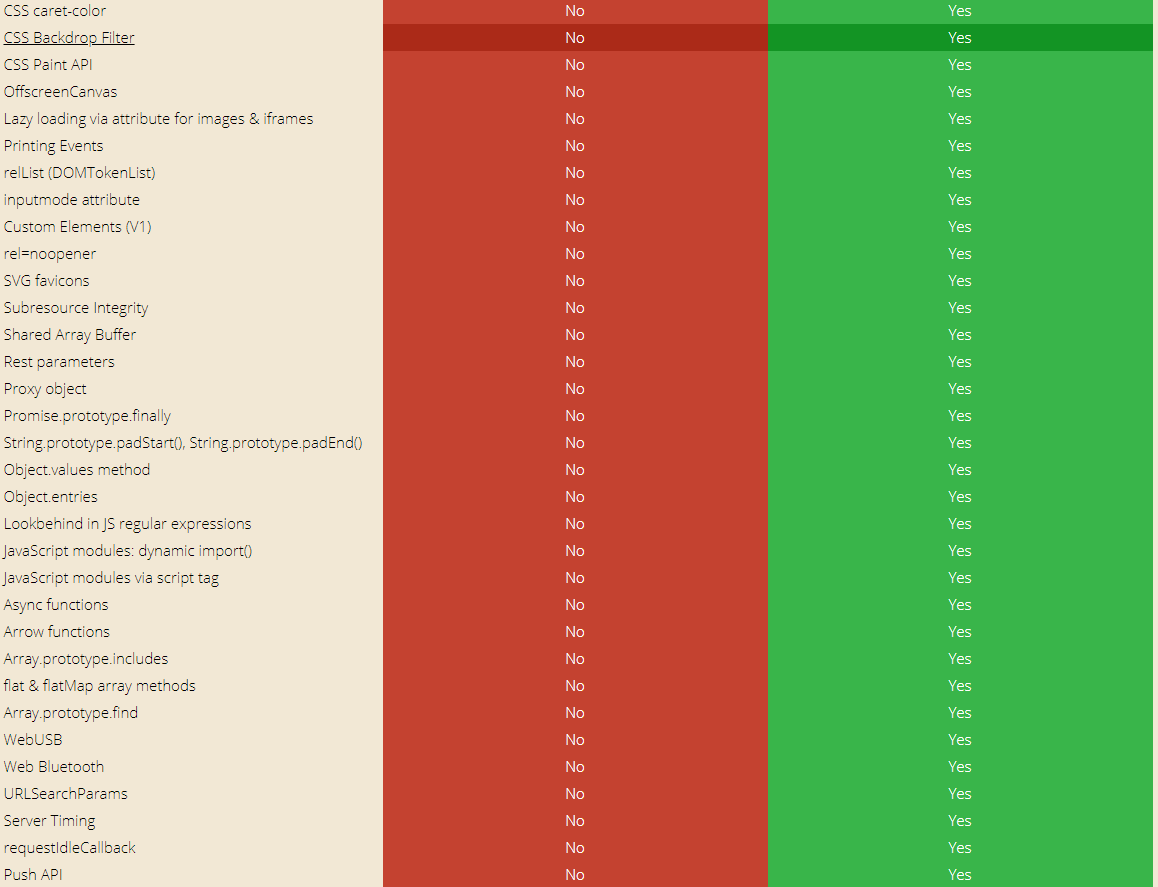
|  |  |  |
| --- | --- | --- |
| Modules | No ✘ | Yes ✔ |
| Classes | No ✘ | Yes ✔ |
| Arrow functions | No ✘ | Yes ✔ |
| Promises | Yes ✔ | Yes ✔ |
| Template strings | Yes ✔ | Yes ✔ |
| Typed arrays | Yes ✔ | Yes ✔ |
| Internationalization | Yes ✔ | Yes ✔ |

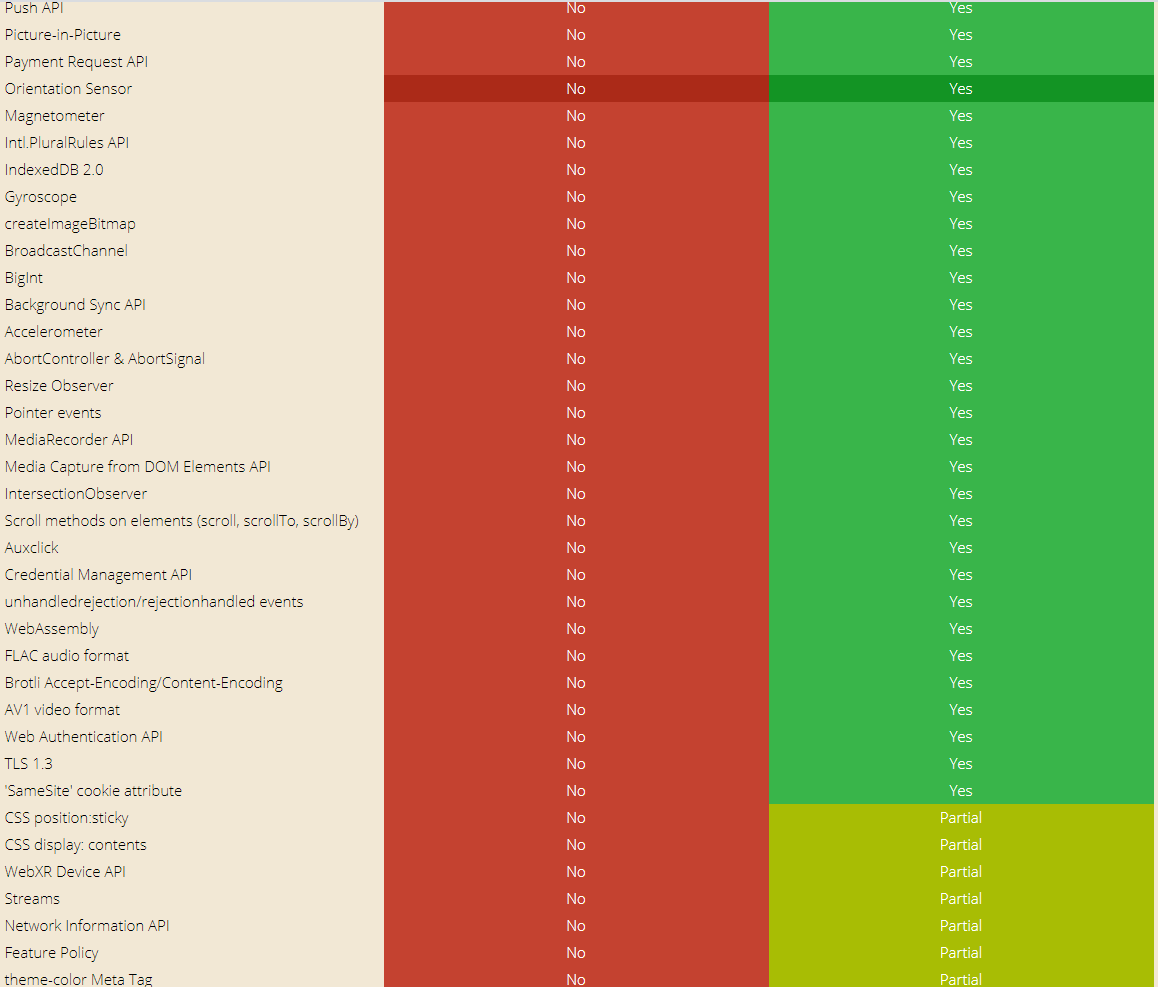
**caniuse对比结果:**

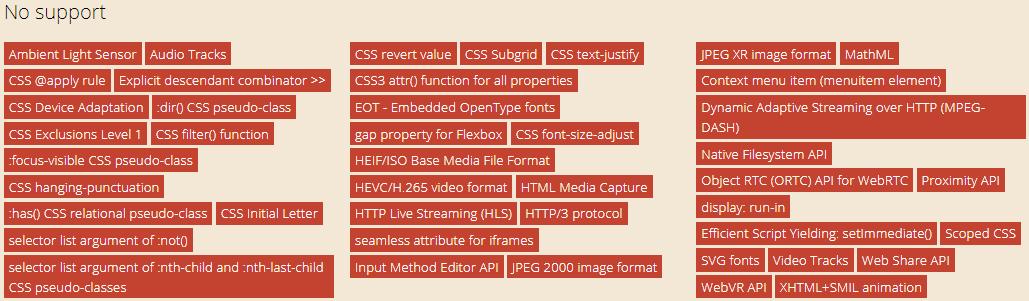












* *由两个站点对比结果可以看出，Chrome43与最新版本Chrome之间对HTML5、CSS3、ES6的支持情况还是有较多区别，这些支持的差异性也是造成兼容性的根本问题，在开发中以实时提醒的方式，将兼容性问题前移至开发阶段，就是“兼容性检测插件”的最大意义，也是开发该插件的最根本出发点。*
  1. 操作和界面级别的差异

1. 首页差异

锦帛：网银环境检测，快速登陆，常用服务等模块化界面，平铺展示供用户操作。

Chrome: 即为标准浏览器界面：首页TAB，导航栏，工具栏，Google搜索页等

1. 界面设计逻辑差异

锦帛：一切为“企业网银首页”服务，设计了环境异常检测，E盾管理，证书管理，信息收集等模块，企业网银模块强制定向至企业网银登陆页，该定向页不可更改。

Chrome: 是一个开放的，浏览互联网世界的“视口”，除初次定向至Google搜索页外，无逻辑导向行为，该定向页也可以客户定制，如设置为百度搜索页。

1. 操作模式差异

锦帛：功能全部平铺展示，客户操作层数较少。外观固定无法定制化。

Chrome: 主要展示区域为TAB页内容，关于环境安全方面设置进行了隐藏封装，需用户进入设置中更改。可定制化修改外观。

1. 环境安全差异

锦帛：首个模块即为“网银环境检测”，体现出对环境安全的重视，环境安全检查不可配置，检查项固定，可针对漏洞进行一键修复。

Chrome: 用户可在设置中设置密钥、证书等相关项。

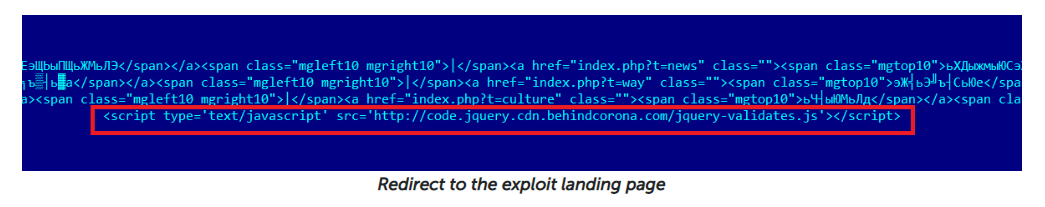
* 1. 协议支持情况差异

1. FTP: Chrome 78开始弃用, Chrome 82：删除与FTP相关的代码和资源
2. HTTP/3: Chrome Canary已整合支持实验性的QUIC与HTTP/3协议
3. HTTP/2:Chrome 51+ 不支持带 NPN 的 HTTP/2网站
4. TLS 1.0和TLS 1.1: Chrome 72不再支持过时的TLS 1.0和TLS 1.1标准, Chrome 81采取的取消支持这两个标准
5. 安全性差异

此章节重点关注0day漏洞：即零时差攻击，是指被发现后立即被恶意利用的安全漏洞。

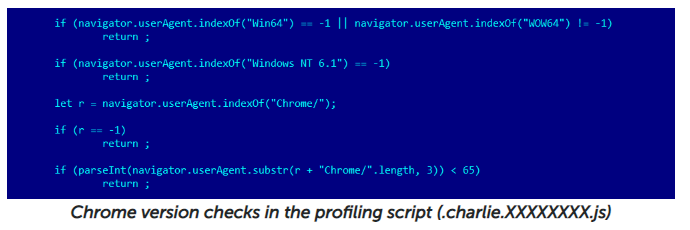
a) CVE-2019-13720漏洞技术细节：

该攻击利用韩语新闻门户网站进行水坑式注入。在主页面中插入了恶意的JavaScript代码，该代码从远程站点加载脚本。



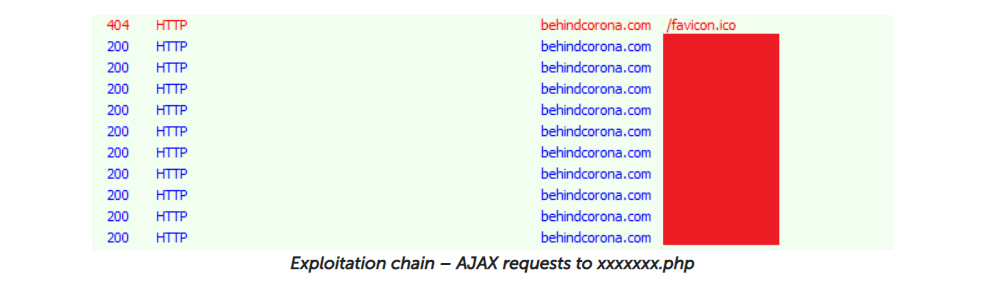
主页存在一个javascript脚本，并从hxxp://code.jquery.cdn.behindcona[.]com/加载了一个远程脚本。

然后，脚本加载另一个名为.charlie.xxxxxxx.js的脚本，通过与浏览器的用户代理进行比较来检查受害者的系统是否可能受到感染，该用户代理在64位版本的windows上运行；它还尝试获取浏览器的名称和版本。 该漏洞试图利用Google Chrome浏览器中的漏洞，脚本还会检查浏览器版本是否大于或等于65（当前Chrome版本为78）：

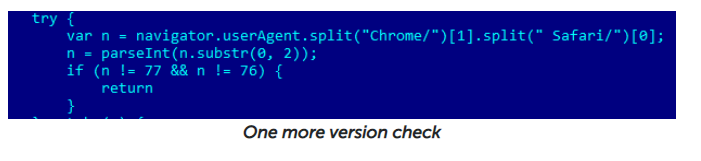


如果浏览器版本确认符合要求，脚本将开始向攻击者控制的服务器（behindcona[.]com）执行多个ajax请求，并向服务器中php脚本传递参数。

第一个请求是获取重要信息以供进一步分析。此信息包括几个十六进制编码的字符串，这些字符串告诉脚本应该从服务器下载多少个漏洞利用代码块，以及一个指向图像文件的URL，该图像文件嵌入了最终有效负载的密钥和rc4密钥，以解密这些漏洞利用代码块。



在下载所有块之后，rc4脚本解密并将所有代码块连接在一起，形成一个新的javascript代码，其中包含完整的浏览器攻击手段。



b) CVE-2019-5786漏洞技术细节：

漏洞源于Blink中的对象生存处理问题。远程攻击者可通过诱导用户点击精心制作的HTML页面来进行超出范围的内存访问，成功利用此漏洞的远程攻击者可执行任意代码。

验证方法：<https://www.jianshu.com/p/c2cd6c7e1976>

1. 下一步工作计划
2. 由于网上查询到的0day漏洞验证方法均比较复杂，有些还涉及到向服务器中注入写好的恶意脚本，实现难度较高，下一步继续查询验证可行性较高的安全漏洞，并尝试进行简化验证。
3. 依托Webkit内核，深入研究浏览器渲染原理，形成可汇报性组织资产，并思考由原理转化实践生产力的途径。
4. 继续研究根据浏览器内核进行二次开发的相关资料：
5. 使用什么语言
6. 依托哪些软件
7. 如何做简单原型实践🡪最好是简单套壳型实践，为接手助手产品，甚至管控，参与C端落地打基础。
8. 评估caniuse内移的思路、方案、可行性。

如果内移可行性较高，结合VSCode 插件开发步骤，探索如何开发出元素兼容性提示插件的开发路线。