Chrome和Firefox浏览器被商业间谍软件Heliconia恶意攻击

12 月 2
日消息，谷歌威胁分析小组（TAG）本周三发布警告，**商业间谍软件 Heliconia 正利用谷歌 Chrome 浏览器、火狐浏览器 Firefox 和 Microsoft Defender 安全软件中存在的漏洞窃取用户数据**。

```python
def GetStrings(path):
    # as strings of interest can be in any flavour, it's better to use `strings` Linux tool to find them all
    # even on windows githooks are run via tiny bash so we don't have to cary this tool with repo
    strings = b''
    stringTypes = ['s', 'S', 'b', 'l', 'B', 'L']

    for stringType in stringTypes:
        strings += check_output(['tools\strings.exe', '-e', stringType, Path(path).as_posix()])

    return strings


def IsBinaryClean(path, doNotCheckShlccdeSection=False, doNotCheckDebugDir=False):
    print('Checking %s' % path)

    # check for sensitive strings

    data = None
    try:
        # we'll need it a bit later
        data = open(path, 'rb').read()
    except:
        print('%s is not a valid path')
        return False

    strings = GetStrings(path)

    badStrings = ['heliconia', 'heliconia-charlotte',      # project name
                  'freezer', 'majinbuu', 'janemba',        # developer names
                  'variston'                               # company name
                  ]
    for badString in badStrings:
        badStringVariants = [badString, badString.upper(), badString[0].upper() + badString[1:]]
        if any(badStringVariant.encode('utf-8') in strings for badStringVariant in badStringVariants):
            print('Found string %s (might be in different case)' % badString)
            return False

    # check for presense of debug directory
    pe = pefile.PE(data=data)

    if not doNotCheckDebugDir:
        for data_directory in pe.OPTIONAL_HEADER.DATA_DIRECTORY:
            if data_directory.name == 'IMAGE_DIRECTORY_ENTRY_DEBUG':
                if data_directory.Size != 0:
                    print('Found debug directory. Get rid of it')
                    return False

    if not doNotCheckShlccdeSection:
        for section in pe.sections:
            if b'.shllcde' in section.Name:
                print('Found .shllcde section. Get rid of it')
                return False

    return True
```

谷歌 TAG 团队表示在一份匿名的 Chrome 错误报告中发现了这个框架，其指令和源代码叫做“ Heliconia Noise ”、“ Heliconia Soft ”和“ Files ”。

TAG 团队通过对错误报告的分析之后，锁定了背后的开发者可能是西班牙巴塞罗那的安全公司 Variston IT，不过后者目前并未作出回应。

这三个组件执行以下功能：Heliconia Noise 是一个网络框架，用于部署 Chrome
浏览器渲染器漏洞（现已修复），逃离浏览器的沙盒封锁；Heliconia Soft 是一个网络框架，用于部署 Windows Defender 的 PDF 漏洞；Files 是一套用于 Linux 和 Windows 的 Firefox 漏洞。

TAG 团队表示，谷歌、微软和 Mozilla 在 2021 年和 2022 年初修复了这个漏洞，如果用户已经升级到最新版本应该不会受到影响。TAG
公司还在谷歌的安全浏览服务中增加了一个 Heliconia
检测机制，并敦促互联网用户保持他们的浏览器和软件的更新。

本文链接：https://dqcm.net/zixun/16699925991250.html