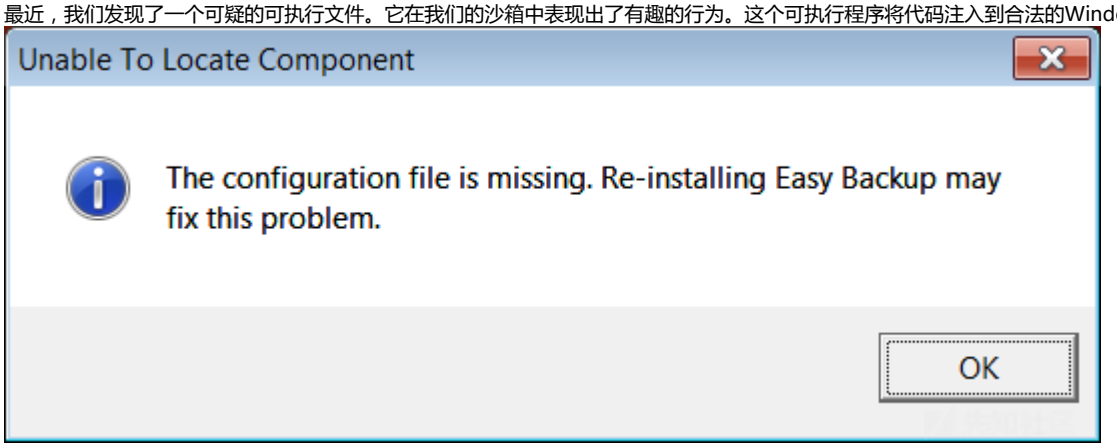
AdKoob数据窃贼将目标瞄准Facebook广告购买信息

Pinging / 2018-08-17 12:30:29 / 浏览数 3073 安全技术 技术讨论 顶(0) 踩(0)

■■■https://news.sophos.com/en-us/2018/07/29/adkoob-information-thief-targets-facebook-ad-purchase-info/

■■■■■■■■■■■■■■■■■■■■Facebook■■■■■■■■■■■■

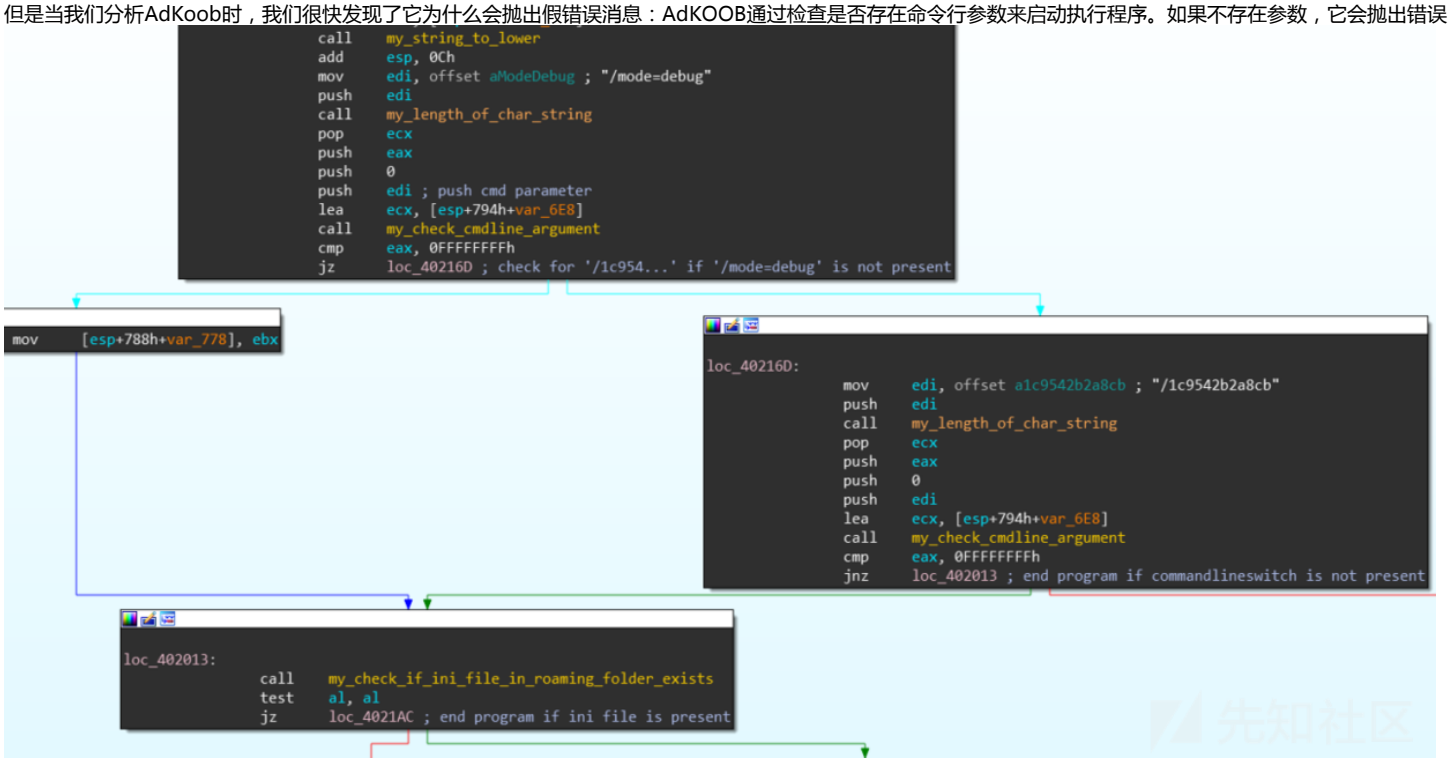
在Sophos公司中，我们不断寻找新的威胁。其中一个沙箱系统帮助我们筛选每日大量新的恶意软件，并给予我们分析运行态恶意软件行为的能力。
最近，我们发现了一个可疑的可执行文件。它在我们的沙箱中表现出了有趣的行为。这个可执行程序将代码注入到合法的Windows二进制文件中（svchost.exe），并且注入



这个奇怪的行为促使我们去深入分析恶意软件。我们一步跳入了这个研究洞穴，并发现了先前未被证实的威胁，我们称其为AdKoob。这是一个凭证窃取的恶意软件。它会将

躲避检测和沙箱

为了避免被检测出来，AdKoob主要进行了双层包装：首先进行开放源码UPX封装，然后使用自定义的代码注入工具。这个自定义的工具使用了称为"进程镂空"的技术将恶意
但是当我们分析AdKoob时，我们很快发现了它为什么会抛出假错误消息：AdKOOB通过检查是否存在命令行参数来启动执行程序。如果不存在参数，它会抛出错误，但在



如果恶意软件进程在开始的时候是通过其正确的方法执行的，那么AdKoob会通过检查%appdata%目录下'FC29FA0894FE.ini'文件的存在性来选择是否继续执行。此方法保
使用内置命令行参数检查是一种简单而狡猾的反沙箱技术，因为自动化沙箱不太可能提前知道它需要传递给可执行文件的参数。

窃取浏览器凭证

一旦能够逃脱了所有的检查，AdKoob就开始其第一个核心任务：窃取用户保存在浏览器中的用户名和密码。为了实现这个目的，AdKoob会直接访问各种浏览器存储凭证的
AdKoob使用一些技巧从不同浏览器中获取数据信息。对于Chrome和Firefox的旧版本（版本58之前），它对这些存储在磁盘中的浏览器SQLite数据库进行SQL查询（对于

```
SELECT encryptedUsername, encryptedPassword, formSubmitURL FROM moz_logins
SELECT origin_url, username_value, password_value FROM logins
```

AdKoob通过文件名访问SQLite数据库：

```
signons.sqlite (Firefox's SQL credential database)
Google\Chrome\User Data\Default\Login Data (path to Chrome's database)
```

在最近的版本中，AdKoob也可以去查询保存在JSON文件内部的火狐存储凭证。
此外，AdKoob看起来将浏览器的登陆表单凭证保存在以下注册表位置：

```
Software\Microsoft\Internet Explorer\IntelliForms\Storage2
```

AdKoob能够提取浏览器中保存的"基本访问认证"凭证（这些凭证被某些Web应用程序使用，并在特定HTTP报头中发送）。它使用GUID字符串，而GUID字符串需要生成加

```
abe2869f-9b47-4cd9-a358-c22904dba7f7 (■■■■■■■■■■GUID■■■)
Microsoft_WinInet (■■■■■■■■■■■■■■■■■■)
```

在获取凭证之后，AdKoob通过将HTTP请求发送给'useragent.cc'
或者'mybrowserinfo.com'来确定受害者的公共IP地址。除了被害人的公共IP地址之外，AdKoob还收集了有关主机的基本信息。
这些被获取到的浏览器凭证被编码后通过HTTPS发送给攻击者，在此过程中使用了恶意程序中的两个硬编码URL。每个浏览器都有一个单独的POST请求。对于被分析的样本

```
hxxps://104[.]200[.]131[.]253:1989/stats1.asp
hxxps://45[.]32[.]91[.]128:1989/stats1.asp
```

一个加密过后的请求如下：

```
3ED4ACD2B09E4389E997B918A9A7ADB4B07A4611BF|Windows NT 6.1<<1.2.3.4|UTC+00:00 GMT Standard Time|chrome|Mozilla/5.0
(Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99
Safari/537.36{||}aHR0cHM6Ly9hY2NvdW50cy5nb29nbGUuY29tL3NpZ25pbi92Mi9zbC9wd2Q=|dGVzdGNyZWQ=|U2F2ZU1lIQ==|yangyangfb
```

解码请求中的字段描述如下：
字段	数据
唯一机器标示符（从C驱动器的主机名和卷序列号导出）	3ED4ACD2B09E4389E997B918A9A7ADB4B07A4611BF
□操作系统版本	Windows NT 6.1
公告IP（样本）	1.2.3.4
被攻击者的本地时间	UTC+00:00 GMT Standard Time
AdKoob浏览器标识符字符串	chrome
用户代理（从证书中窃取的浏览器凭证）	Mozilla/5.0 (Windows NT 6.1; Win64; x64)...
base64编码后的浏览器凭据（网站、用户名、密码）	aHR0cHM6Ly9...dGVzdGNyZWQ=U2F2ZU1lIQ==
Bot标示符	yangyangfb

对Facebook个人信息部分的间谍活动

一旦保存的浏览器凭据从受害者的机器中获取出来，AdKoob就开始它的第二个核心任务：建立一个Facebook会话并将数据从受害人的Facebook数据存储中删除。AdKoo
第一种方法依赖于通过Facebook认证cookies，使用存储在用户浏览器中的Facebook会话。AdKoob通过受害者安装的浏览器专门寻找与facebook.com域名相关的的Coo

```
.rdata:004C60C0 aSGoogleChromeU:                        ; DATA XREF: my_steal_facebook_credentials_in_browser+E2↑o
.rdata:004C60C0 text "UTF-16LE", '%s\Google\Chrome\User Data\Default',0
.rdata:004C6106 align 4
.rdata:004C6108 ; char aHostKey[]
.rdata:004C6108 aHostKey db 'host_key',0                 ; DATA XREF: sub_41C51C+A5↑o
.rdata:004C6111 align 4
.rdata:004C6114 ; char aEncryptedValue[]
.rdata:004C6114 aEncryptedValue db 'encrypted_value',0  ; DATA XREF: sub_41C51C+11A↑o
.rdata:004C6124 aS_2 db '%S',0                          ; DATA XREF: sub_41C51C+14A↑o
.rdata:004C6127 align 4
.rdata:004C6128 ; char aExpiresUtc[]
.rdata:004C6128 aExpiresUtc db 'expires_utc',0          ; DATA XREF: sub_41C51C+179↑o
.rdata:004C6134 align 8
.rdata:004C6138 aSelectFromCook db 'SELECT * FROM cookies WHERE cookies.host_key LIKE "%.facebook.com'
.rdata:004C6138                                         ; DATA XREF: my_steal_fb_cookie_chrome+7E↑o
.rdata:004C6138 db '";',0
.rdata:004C617C aCookies:                               ; DATA XREF: my_steal_fb_cookie_chrome:loc_41C7BD↑o
.rdata:004C617C text "UTF-16LE", 'Cookies',0
```

先知社区

类似于保存的凭据，Cookie存储在Firefox和Chrome中的SQLite数据库中。我们可以很容易地通过以下SQL语句来查询cookies：

```
SELECT * FROM cookies WHERE cookies.host_key LIKE "%.facebook.com";
SELECT * FROM moz_cookies WHERE moz_cookies.host LIKE "%.facebook.com";
```

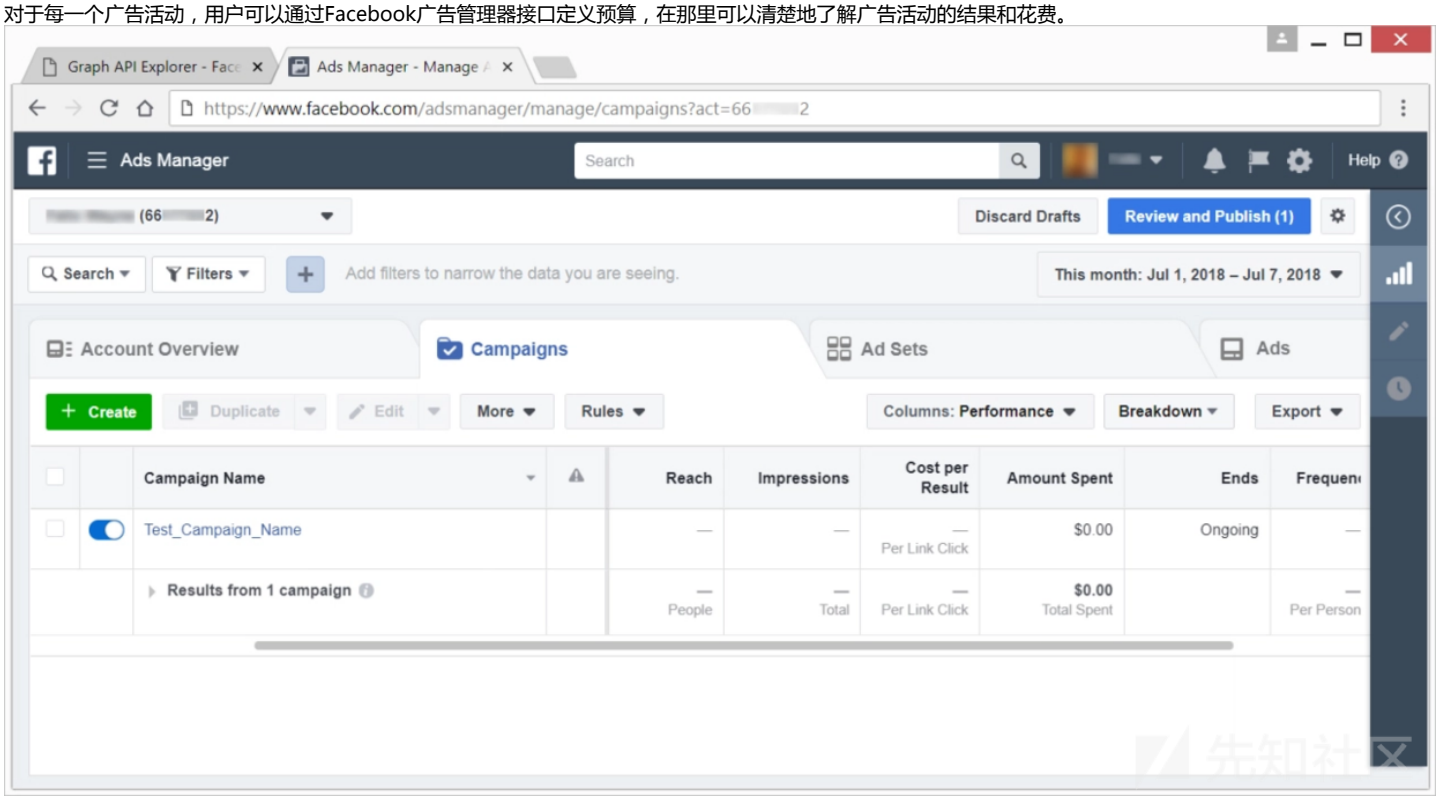
□除了Chrome和Firefox之外，AdKoob还查询了Internet Explorer和微软的Edgies浏览器中存在的Facebook的Cookie。
第二种方法利用了AdKoob所盗取的浏览器凭证。AdKoob试图通过发送适当的HTTP请求登录到Facebook，如果成功的话，它能够获得额外的Facebook会话cookies。
一旦认证成功，AdKoob会继续进行间谍活动。它会对Facebook的一系列url进行请求。恶意软件使用相应的正则表达式集检查来自每个请求的响应数据。这种技术能够使A
下列三个请求是Facebook的url：

```
https://www.facebook.com/
https://www.facebook.com/username/about
https://www.facebook.com/bookmarks/pages
```

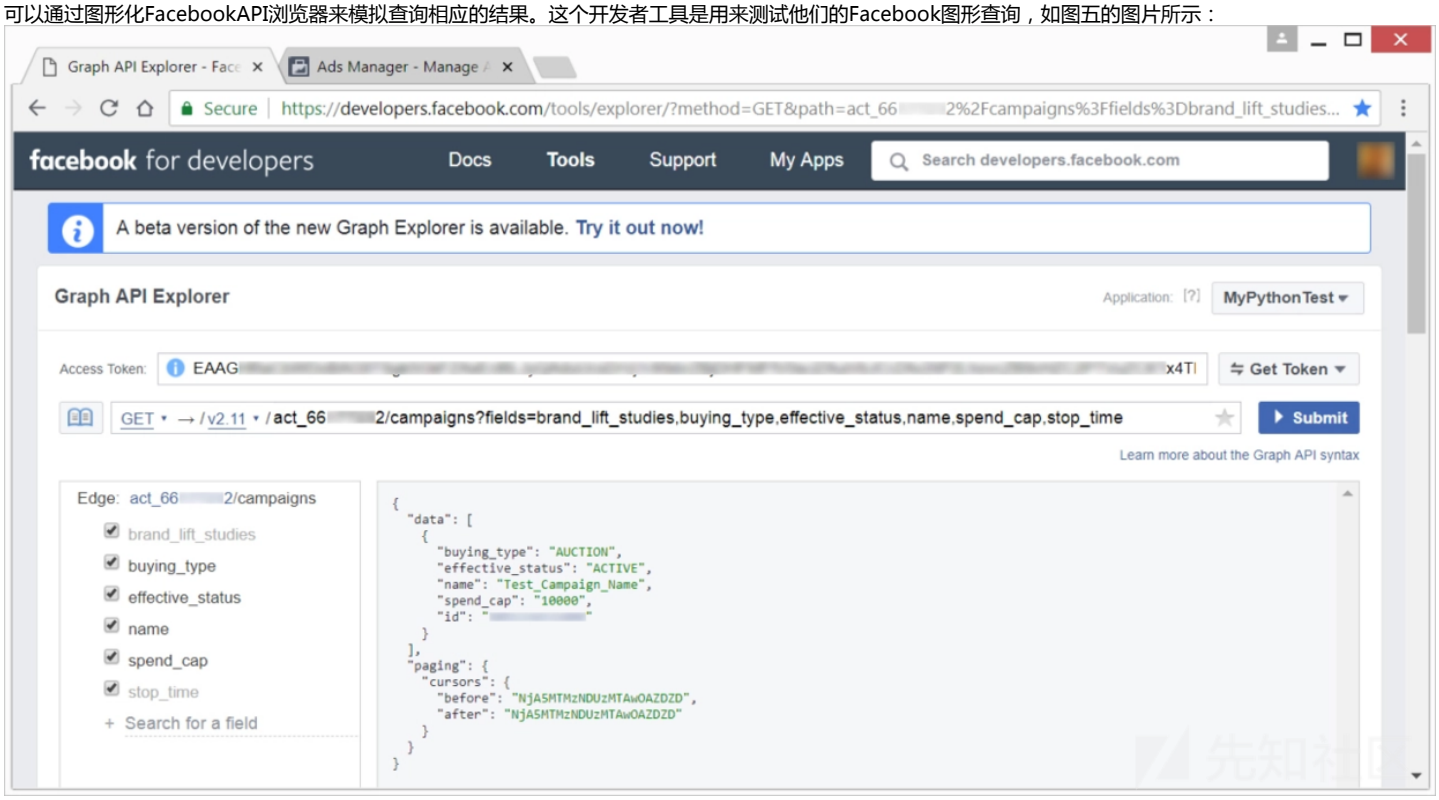
从这些请求中，AdKoob盗取了下列数据：

- Facebook所在地（定义用户首选接口语言、区域的参数）。
- Facebook用户ID（一个无法自识别Facebook用户但与Facebook用户个人信息相关联的字符串）。
- Facebook数据存储用户名（用户选择的用户名，其被用来与用户的个人信息相连接，例如www[.]facebook.com/janedoe3）。
- 从用户的Facebook数据存储（包括用户的全名和生日）的部分信息。

用户创建的Facebook网页页面名称。
###广告花费中的趣事
在盗取受害者的Facebook个人资料之后，AdKoob继续向一个不被日常脸谱网用户使用的界面发出一系列有趣的请求，这与付费的广告活动有关。企业有可能在Facebo
对于每一个广告活动，用户可以通过Facebook广告管理器接口定义预算，在那里可以清楚地了解广告活动的结果和花费。



AdKoob向提供服务的URL发起请求。之后，它检查响应并提取广告帐户ID（唯一标识广告帐户的字符串）和花费在广告活动上的总金额。
然后，提取的广告帐户ID用于对Facebook图形API进行查询，该程序允许程序员从Facebook帐户中查询信息。下面的查询由AdKoob发起：
`act_<adaccountid>/campaigns?date_preset=this_month&fields=["boosted_action_type","brand_lift_studies","buying_type","effecti`
可以通过图形化FacebookAPI浏览器来模拟查询相应的结果。这个开发者工具是用来测试他们的Facebook图形查询，如图五的图片所示：



AdKoob发起的查询窃取了来自广告活动的各种信息，例如广告名称、当前花费的金额以及广告活动的预算限制。
这可能是AdKoob最有趣的事情了。有效载荷表明这些攻击背后的罪犯瞄准的是商业用户。原因很简单，普通家庭用户不太可能使用Facebook进行广告活动。
所有的被提取的脸谱网数据都被发送到同一个硬编码URL，而这些URL被用来过滤浏览器的凭证。除了来自Facebook账户的数据被盗以外，Facebook的用户cookies和F

一旦所有的盗取目的达成，AdKoob会给不同的URL发送GET请求，这些请求包括基于用户的安全标识符的认证码（SID）：

```
https://107[.]151[.]152[.]220:5658/down.asp?action=newinstall&u=<identifier>
```

这个URL不同于用于数据盗取相关的URL，这个URL大概是用于感染数量的统计。
最后AdKoob以创建文件'%appdata%\FC29FA0894FE.ini'来结束它的执行过程。这确保了恶意软件不会运行两次。然后它会进行自我销毁，试图不留下系统被破坏的
###IOCs
打包AdKoob样品（SHA256）
e383582413cc53ec6a630e537eedfeee35d6b5f3495266f2530770f4dd3193a6
Unpacked, analyzed AdKoob sample (SHA256): 6a6260cb5e1e0ad22af2bc8bb73bc8423df6315a88e39c2264f6def5798b6550
YARA规则

```
rule adkoob_information_stealer
{
 meta:
     author = "Felix Weyne, Sophos"
 strings:
     $facebook_cookie_firefox = "SELECT * FROM moz_cookies WHERE moz_cookies.host LIKE \"%.facebook.com\""  nocase ascii
     $facebook_cookie_chrome = "SELECT * FROM cookies WHERE cookies.host_key LIKE \"%.facebook.com\""  nocase ascii
     $facebook_regex_ad_account_id = "<td [^>]*?data-testid=\"all_accounts_table_account_id_cell\">([^<>]*?)</td>"  nocase w
     $self_destruction = "/C ping localhost -n 4 > nul & del" nocase wide
 condition:
     all of them
}
```

点击收藏 | 0 关注 | 1
上一篇：Hack the ch4inrul... 下一篇：通过内存转储破解 Linux 全盘...
1. 0 条回复
- 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板