Panda Banker银行木马分析

angel010 / 2018-10-10 21:28:24 / 浏览数 1978 技术文章 技术文章 顶(0) 踩(0)

本文翻译自：https://threatvector.cylance.com/en_us/home/threat-spotlight-panda-banker-trojan-targets-the-us-canada-and-japan.html

---

Panda Banker是使用Zeus源代码变种的银行木马。该银行木马于2016年首次被发现，至今仍然活跃。

Panda Banker会用man-in-the-browser技术将恶意代码注入到受害者浏览器的web页面中。注入的代码会窃取银行账户、信用卡和个人信息。

Panda Banker最近也通过Emotet进行传播。Panda Banker采用许多步骤来隐藏其行为。复杂的代码混淆和多层加密使其C2通信和恶意脚本很难被分析。

Panda Banker主要的攻击目标位于美国、加拿大和日本。恶意软件主要关注银行账户、信用卡和web钱包信息。

技术分析

概览

Panda
Banker的攻击流程非常复杂（如图1）。首先检查受害者运行的环境是不是沙箱，然后创建一个包括扩展文件属性的副本，副本创建完成后，进程会在退出前加载新创建的恶

Panda Banker的C2
URL是从payload中嵌入的配置数据中获取的。也会与C2服务器通信来获取其他的配置信息。如果发现一个已知的web浏览器进程，就会注入插件dll到web浏览器中来拦截

然后Panda
Banker会等受感染的浏览器访问目标站点（比如银行或信用卡公司的网站）。当访问目标站点时，恶意软件会注入针对特定目标的恶意脚本来窃取银行账户、信用卡和个人

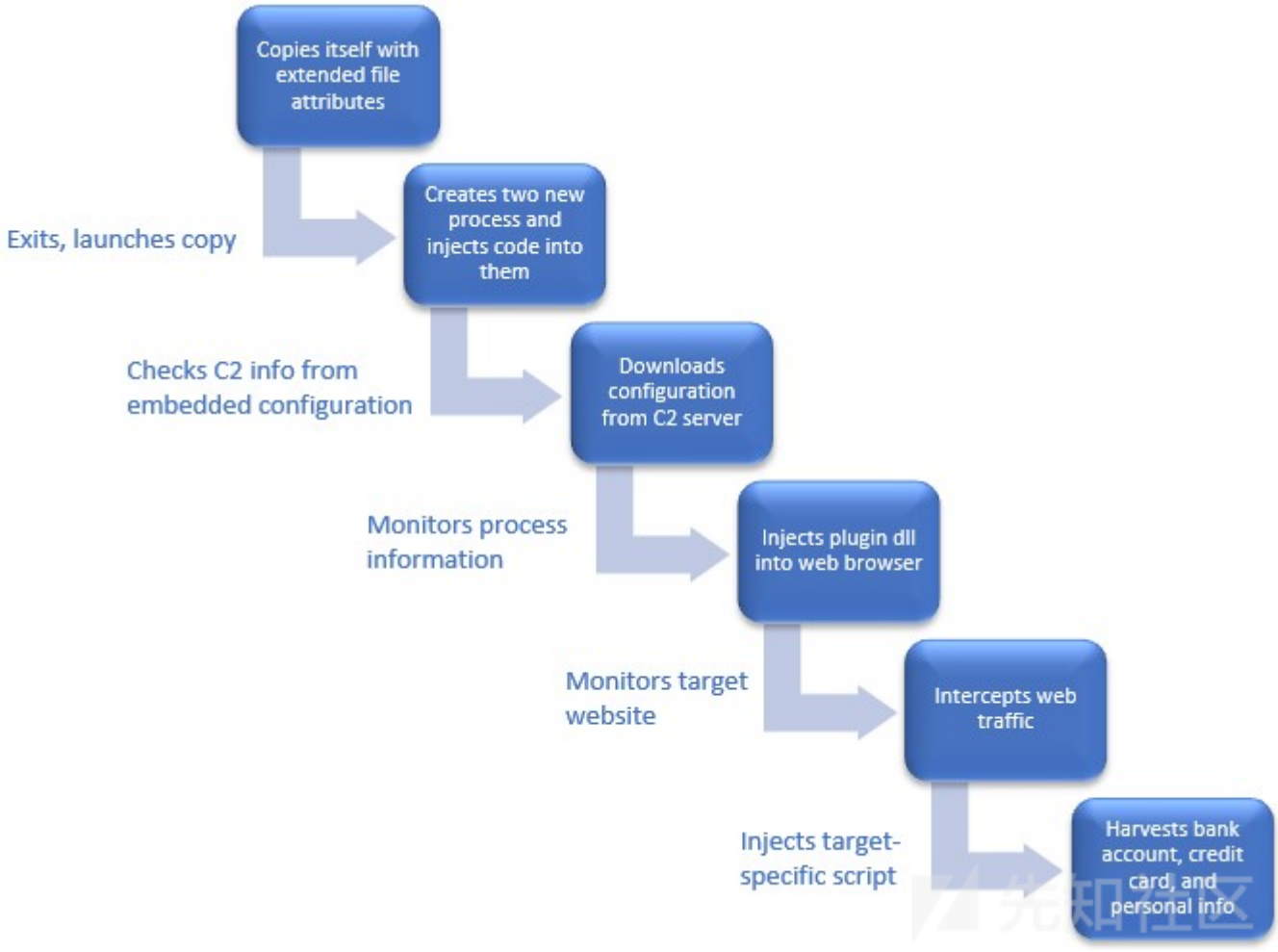

图1: Panda Banker攻击流程

规避检测

Panda
Banker会检查受害者的环境来避免沙箱和手动分析（表1），包括包抓取持续、调试器、反编译工具和其他恶意软件分析工具。如果在环境中找到这些工具，恶意软件就会退

Action	Target
Open File	C:\\popupkiller.exe
	C:\\stimulator.exe
	C:\\TOOLS\\execute.exe
	\\\\.\\NPF_NdisWanIp
	\\\\.\\REGVXG
	\\\\.\\FILEVXG
	\\\\.\\REGSYS
	\\\\.\\FILEM
	\\\\.\\TRW
Load Library	SbieDLL.dll
Create Mutex	Sandboxie_SingleInstanceMutex_Control
	Frz_State
Find Process Name	Wireshark
	Immunity
	Processhacker
	Procexp
	Procmon
	Idaq
	regshot
	aut2exe
	perl
	python
Open Registry	HKCU\\Software\\WINE
	HKLM\\Software\\WINE
Call GetProcAddress	wine_get_unix_file_name

表1: Panda Banker检查规避检测的字符串

一旦Panda Banker通过环境检查，就会创建4个新文件。其中一个文件是Panda Banker的副本。`blocklist.exe`就是payload（图2）：

Name	Date modified	Type	Size
3561288849sdhlie.uwi	6/4/2018 1:28 PM	UWI File	0 KB
blocklist.exe	6/11/2018 7:11 PM	Application	261 KB
data_0.cus	6/23/2018 8:28 PM	CUS File	0 KB
xulstore.hye	5/29/2018 5:30 PM	HYE File	0 KB

图2: Panda Banker创建的4个文件

Panda Banker会通过`Ntseteafiles API`给恶意软件副本分配一个扩展文件属性，在本例中是EaName is BEAR（图3）。加载了副本后原始payload就会退出。一旦Panda Banker在扩展文件属性中找到`BEAR`，就会创建两个svchost.exe进程并注入：

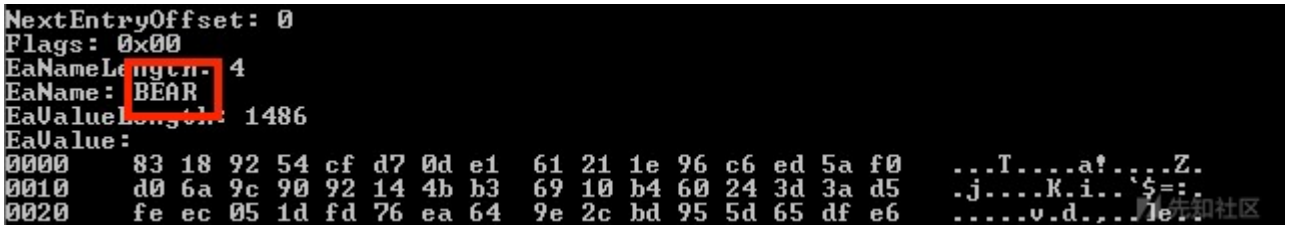

图3: Panda Banker分配给扩展文件属性EaName

Payload中的配置数据

Panda Banker的payload含有配置数据，含有到C2服务器的URL和公钥。配置数据是用AES算法加密的，加密数据结构如图4所示：

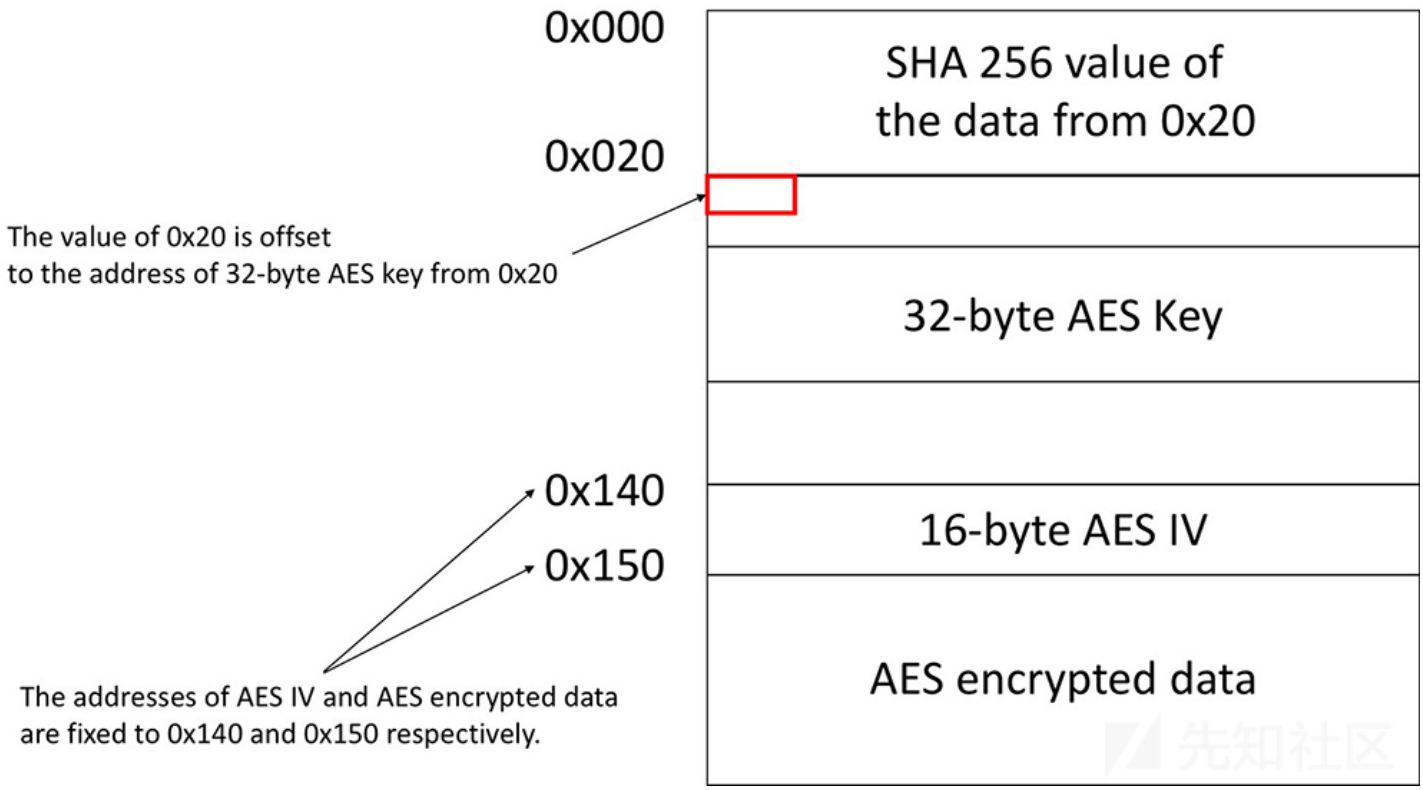

图4: Panda Banker payload中嵌入的加密的配置数据结构

对加密数据解密后，会看到用RC4加密的C2 URL和`X.509 subjectPublicKeyInfo DER SEQUENCE`格式的RSA公钥。

解密URLs需要用嵌入的RSA公钥进行RC4解密。这里，`66 c7 5b 69 f4 5a 4e 12`等于`https://`：

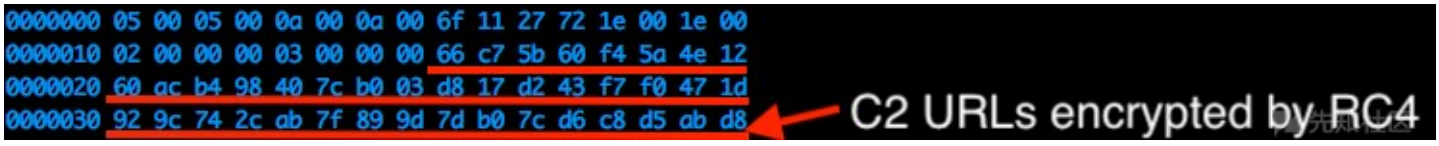

图5: RC4加密的C2 URLs

图6: RSA公钥

URL生成算法

Panda Banker在访问C2服务器时会生成URL，见图7。生成的URL看起来像随机字符串，但后面有个算法：

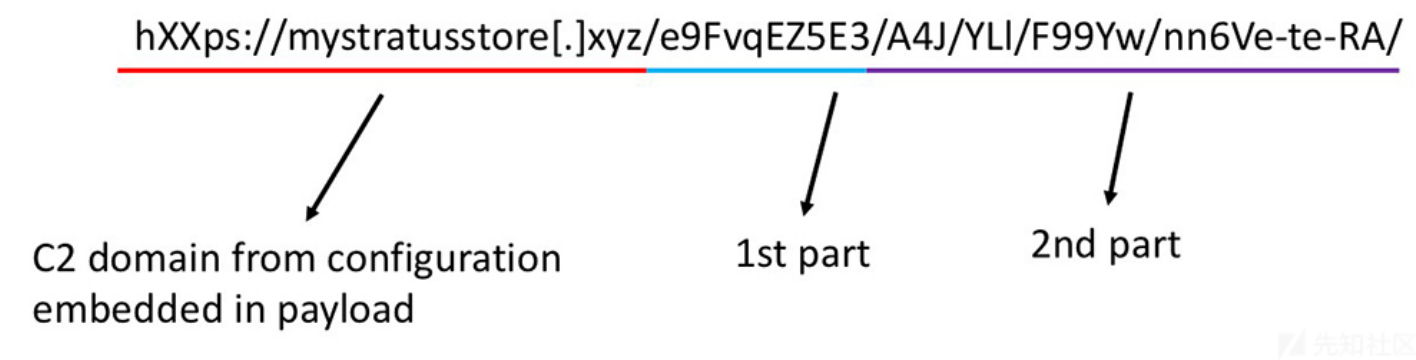

图7: 生成的URL示例

Panda Banker用`Mersenne Twister`算法生成随机值。整个URL算法描述如下，步骤1-5描述的是第一部分，6-10描述第二部分：

1. 根据公式`mod(A random value from Mersenne Twister, 9) + 2`决定part 1的长度，结果在2~10之间。
2. 根据公式`mod(A random value from Mersenne Twister, 62)`获取随机的index值，结果在0~61之间。
3. 从预定义的字符串中去除有个字母数字字符，步骤2的结果会成为index值。
   预定义的字符串：`qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM1234567890`,如果index是1，选择w。
4. 将步骤3的结果加到URL中，作为第一部分的一部分。
5. 重复步骤2到4，具体次数由步骤1的结果决定。

   得到4个值：

   1. 通过`GetComputerNameW API`得到计算机名；
   2. 通过`HKLM\software\microsoft\windows nt\currentversion`得到`InstallDate`；
   3. 通过`HKLM\software\microsoft\windows nt\currentversion`得到`DigitalProductId`值，并计算CRC值；
   4. 通过`GetVersionEx API`得到`OSVERSIONINFOEX`值，并计算CRC值；

   将步骤6的结果打包，并计算SHA256的值；

8. 用步骤7中得到的值的前16个字节；

   用下面的伪代码对步骤4和8中的结果进行XOR运算：

```
resultlist = []
       for i in range(0, 16):
           xoredvalue = the_begining_of_16bytes_from_result_of_8[i] ^ 1st_part[i % len(1st_part)]
           resultlist.append(xoredvalue)
```

   resultlist中的每个值都用base64编码，并将+, /,=替换掉。然后，根据步骤d中的公式计算mod，如果mod小于20，在第二部分末尾加上/。

   1. a. '+' -> '-' (hyphen)
   2. b. '/' -> '_' (underbar)
   3. c. '=' -> '' (Nothing)
   4. d. mod(A random value from Mersenne Twister, 100)

   将C2域名，第1部分，第2部分连接起来：`C2 domain/1st part/2nd part`。

C2通信

请求

Panda Baker的POST请求参数示例如图8所示，是用32字节key的AES-256 CBC模式和16字节的IV加密的，process行是Panda Banker将自己注入svchost.exe的过程。name值在Panda Banker payload中已预配置过了。如果有必要，恶意软件会从C2服务器接收配置数据：

```
{
    "BotInfo": {
        "systime": 1525924621,
        "process": "svchost.exe",
        "user": USERNAME of victim's computer,
        "id": "unique value per user",
        "botnet": "not set",
        "version":"2.6.10"
    },
    "File": {
        "name": "2itopfetoebenfeakoqas[.]dat"
    }
}
```


图8: POST参数示例

对每个POST请求，Panda Banker会常见32字节的key和16字节的IV用于AES加密。生成的AES key用RAS key加密，如图6。然后将用RSA加密的AES key、16字节的AES IV和AES加密的POST参数打包。然后Panda Banker会从生成的URL的第1和第2部分和打包的内容中计算SHA256值（如图9）。最后，用base64编码所有数据：

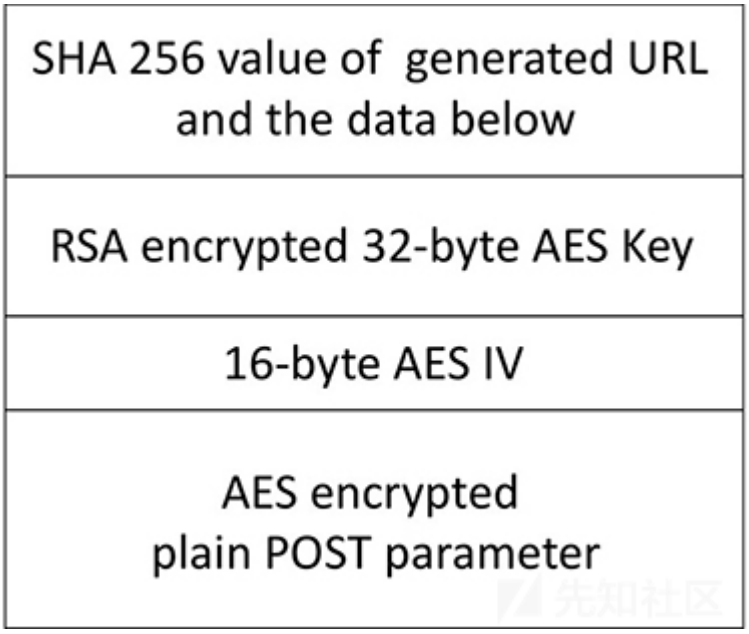

图9: POST主体的二进制数据

C2服务器的响应

Panda Banker的C2服务器会向受害者机器发送多层加密的二进制数据。解密步骤如下：

第一层

因为来自C2服务器的响应数据是base64方式编码的。解码后的二进制格式见图10。二进制数据中的SHA256值用于完整性检查。为了解密AES加密的数据，Panda Baker会复用POST请求中的AES key：

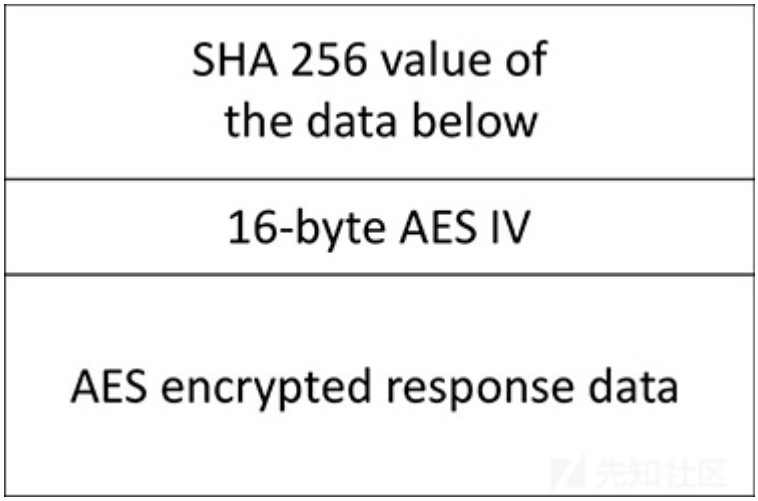

图10: 来自C2服务器的二进制响应数据

解密后的JSON数据如图11：

```
{
    "File": {
        "data": Base64 encoded and AES encrypted binary data,
        "id": unique value per user,
        "name": "2itopfetoebenfeakoqas[.]dat"
    }
}
```

图11: 解密的第一层

第二层

图11中的数据解码后，会出现另一个二进制格式，如图4所示。解密后，会出现更多的JSON数据，如图12。

```
{
        "data": Base64 encoded binary,
        "sign": Base64 encoded signature
}
```

图12: 解密的第二层

解码的`sign`值用于完整性检查。Panda Banker的RSA公钥值会用于检查解码的`data`值的完整性。如果计算的签名和JSON数据的签名不匹配，解码的`data`值就会被忽略。

解密的`data`也是用base64编码的，并含有：

- 配置文件或web注入数据
- PE32（PE32++）可执行文件

第一个例子中，解码的二进制格式如图4所示。一旦解密就会发现配置文件或web注入数据。第二个例子中，解码的数据是有个PE可执行文件（动态链接库）。

来自C2服务器的配置

真实的C2配置数据如图13所示，其中含有传播许多插件的URL，比如`url_plugin_webinject32, url_plugin_webinject64, url_plugin_vnc32, url_plugin_vnc64, url_plugin_backsocks, url_plugin_grabber, url_plugin_keylogger`。

也会显示VNC注入(`inject_vnc`), 窃取的数据(`grab_pass, grab_cookie`), 登陆进程名 (`keylog_process`和`screen_process`)的当前设定。本例中，执行键盘记录和屏幕监控的进程名为putty.exe：

图13: 来自C2服务器的配置数据

web注入方法

Panda Banker会通过`API hooking`拦截浏览器的web流量，将恶意脚本注入到受害者web浏览器的目标web页。也会通过移除`Content Security Policy header`来影响web浏览器的安全。

`url_plugin_webinject32`插件就是用于web注入的。根据分析，它会hook`iexplore.exe, microsoftedge.exe, microsoftedgecp.exe, firefox.exe, chrome.exe, opera.exe`等使用的API。一旦浏览器从访问的URL在`url_webinjects`的配置数据中，插件就会将对应的脚本注入到web浏览器的web页面中。
一些API hooks示例如下：

影响微软浏览器的API Hooks
HttpSendRequestsW
HttpSendRequestsA
HttpSendRequestsExW
HttpSendRequestsExA
InternetReadFile
InternetReadFileExA
InternetReadFileExW
InternetQueryDataAvailable
InternetCloseHandle
HttpOpenRequestsA
HttpOpenRequestsW
HttpQueryInfoA
InternetConnectA
InternetConnectW
InternetWriteFile

影响Firefox的API Hooks

```
PR_Close
PR_Read
PR_Write
PR_Poll
```

影响Chrome / Opera的API Hooks

```
closesocket
WSASend
WSARecv
recv
```

Web注入目标

`url_webinjects`的数据主要攻击银行和信用卡公司。图14是针对银行网站的web注入数据示例。例子中，恶意代码被注入到`<head>`tag后，代码含有下载特定目标窃取脚本的URL，这些命令都是混淆过的，以此隐藏Panda Banker的恶意行为：

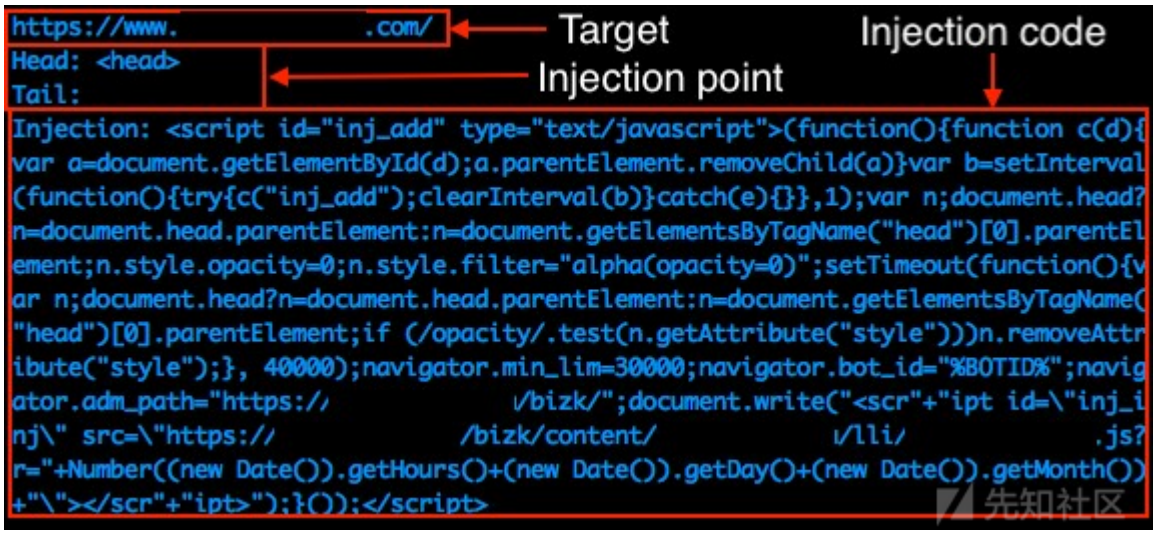

图14: 针对银行网站的注入代码

用于数据窃取的代码反混淆后，可以看出含有下面的功能：

- 注入伪造的消息（图15）
- 窃取卡号（图16）
- 收集昵称、支付限额、借记卡和贷记卡的取现限额

```
<div class="inj_message">
    We have detected a change in your online login pattern. That may be a change in hardware, IP address or installed software.
    In order to verify your identity please provide all of the following information.
</div>
```


图15:注入代码——伪造消息

```
<div class="card_num" "spec_container">
    <label for="inj_card_number_0">Card number 16 digits </label>
    <input tabindex="200" id="inj_card_number_0" autocomplete="off" type="text" class="card_input
    <input tabindex="201" id="inj_card_number_1" autocomplete="off" type="text" class="card_input
    <input tabindex="202" id="inj_card_number_2" autocomplete="off" type="text" class="card_input
    <input tabindex="203" id="inj_card_number_3" autocomplete="off" type="text" class="card_input
</div>
```


图16: 注入代码——窃取卡号

```
cardServiceWrapperList[inlineFile].cardService.nickName);

a.cardServiceWrapperList[inlineFile].cardService.card.identifier.cardNumber);

efault:data.cardServiceWrapperList[inlineFile].cardService.purchaseLimits[u9u.P4b].amount.displayAmount);

ax:data.cardServiceWrapperList[inlineFile].cardService.purchaseLimits[u9u.z4b].amount.displayAmount);

t:data.cardServiceWrapperList[inlineFile].cardService.withdrawalLimits[u9u.P4b].amount.displayAmount);
```


图17: 注入代码——收集昵称、支付限额、借记卡和贷记卡的取现限额

Web注入目标分析

表2是攻击目标的国家和行业分析：

Target Country	Industry
JP	1 video streaming services / E-commerce
	1 porn video streaming service
	11 credit card companies
US	8 banking companies
	2 payroll systems
	1 block chain company
CA	9 banking companies

表2: 攻击目标的国家和行业

美国、加拿大和日本是Panda
Banker的主要攻击国家。恶意软件主要窃取银行账号、信用卡信息和工资系统中的个人信息。Web钱包和区块信息也是工具的目标。

结论

Panda
Banker是一款高度混淆的、高度可配置的、活跃的恶意软件。威胁单元使用该恶意软件来窃取银行卡、信用卡信息，个人数据、web钱包和区块信息。主要的工具目标的美

上一篇：SQL 注入总结 下一篇：利用NodeJS SSRF漏洞获取...
1. 0 条回复
   - 动动手指，沙发就是你的了！

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板