

Regin 模块 50251 和 Qwerty 击键记录器之对比

非官方中文译文·安天技术公益翻译组 译注

文档信息			
原文名称	Comparing the Regin module 50251 and the "Qwerty" keylogger		
原文作者	Costin Raiu, Igor Soumenkov	原文发布日期	2015 年 1 月 27 日
作者简介	Costin Raiu 是卡巴斯基实验室全球研究和分析团队的总监。也是一位自由思想家、软件开发员、建筑师、摄影师、编辑。 http://securelist.com/author/costin/		
原文发布单位	卡巴斯基实验室		
原文出处	http://securelist.com/blog/research/68525/comparing-the-regin-module-50251-and-the-Qwerty-keylogger/		
译者	安天技术公益翻译组	校对者	安天技术公益翻译组
免责声明	<p>本译文译者为安天实验室工程师，本文系出自个人兴趣在业余时间所译，本文原文来自互联网的公共方式，译者力图忠于所获得之电子版本进行翻译，但受翻译水平和技术水平所限，不能完全保证译文完全与原文含义一致，同时对所获得原文是否存在臆造、或者是否与其原始版本一致未进行可靠性验证和评价。</p> <p>本译文对应原文所有观点亦不受本译文中任何打字、排版、印刷或翻译错误的影响。译者与安天实验室不对译文及原文中包含或引用的信息的真实性、准确性、可靠性、或完整性提供任何明示或暗示的保证。译者与安天实验室亦对原文和译文的任何内容不承担任何责任。翻译本文的行为不代表译者和安天实验室对原文立场持有任何立场和态度。</p> <p>译者与安天实验室均与原作者与原始发布者没有联系，亦未获得相关的版权授权，鉴于译者及安天实验室出于学习参考之目的翻译本文，而无出版、发售译文等任何商业利益意图，因此亦不对任何可能因此导致的版权问题承担责任。</p> <p>本文为安天内部参考文献，主要用于安天实验室内部进行外语和技术学习使用，亦向中国大陆境内的网络安全领域的研究人士进行有限分享。望尊重译者的劳动和意愿，不得以任何方式修改本译文。译者和安天实验室并未授权任何人士和第三方二次分享本译文，因此第三方对本译文的全部或</p>		

	者部分所做的分享、传播、报道、张贴行为，及所带来的后果与译者和安天实验室无关。本译文亦不得用于任何商业目的，基于上述问题产生的法律责任，译者与安天实验室一律不予承担。
--	---

Regin 模块 50251 和 Qwerty 击键记录器之对比

Costin Raiu, Igor Soumenko

2015 年 1 月 27 日

2015 年 1 月 17 日,《明镜周刊》发布了一篇根据爱德华·斯诺登提供的文件所编写的详细的文章。与此同时,他们提供了代号为“QWERTY”的恶意程序的副本,据说好几个政府在其 CNE (计算机网络开发) 活动中都用了这一程序

(<http://www.spiegel.de/media/media-35668.pdf>)。

我们已经从《明镜周刊》获得了恶意文件的副本,当分析它们时,我们立刻联想到了 Regin。我们仔细地分析了代码,得出的结论是:Qwerty 恶意软件的功能与 Regin 50251 插件是完全一样的。

分析

Qwerty 模块由 3 个二进制文件和相应的配置文件组成。其中的一个文件 20123.sys 尤其有趣。

20123.sys 是 Qwerty 击键记录器的内核模式的一部分。事实证明,其源代码也存在于 Regin 的一个插件 50251 中。

使用二元差异法,可以很容易地发现两个文件所共享的大部分代码。

大部分共享代码用于访问系统键盘驱动程序。

00010756 DeviceKeyboardclass0: ; DATA XREF: sub_10660	00010660 DeviceKeyboardclass0: ; DATA XREF: sub_10660
00010756 unicode 0, <\Device\KeyboardClass0>,0	00010660 unicode 0, <\Device\KeyboardClass0>,0
00010784 db 2 dup(0)	0001068E align 10h
00010786 Kbdclass_sys: ; DATA XREF: sub_10690	00010690 Kbdclass_sys: ; DATA XREF: sub_10690
00010786 unicode 0, <kbdclass.sys>,0	00010690 unicode 0, <kbdclass.sys>,0
000107A0 ; ===== SUBROUTINE =====	000106AA ; ===== SUBROUTINE =====
000107A0 ; Attributes: bp-based frame	000106AA ; Attributes: bp-based frame
000107A0 sub_107A0 proc near ; CODE XREF: sub_106AA	000106AA sub_106AA proc near ; CODE XREF: sub_106AA
000107A0 DestinationString= UNICODE_STRING ptr -14h	000106AA DestinationString= UNICODE_STRING ptr -14h
000107A0 FileObject = dword ptr -0Ch	000106AA FileObject = dword ptr -0Ch
000107A0 DeviceObject = dword ptr -8	000106AA DeviceObject = dword ptr -8
000107A0 var_4 = dword ptr -4	000106AA var_4 = dword ptr -4
000107A0 push ebp	000106AA push ebp
000107A1 mov ebp, esp	000106AB mov ebp, esp
000107A3 sub esp, 14h	000106AD sub esp, 14h
000107A6 push ebx	000106B0 push ebx
000107A7 push offset DeviceKeyboardclass0 ; Sc	000106B1 push offset DeviceKeyboardclass0 ; \
000107AC lea eax, [ebp+DestinationString]	000106B6 lea eax, [ebp+DestinationString]
000107AF push eax ; DestinationStr	000106B9 push eax ; DestinationStr
000107B0 xor bl, bl	000106BA xor bl, bl
000107B2 call ds:RtlInitUnicodeString	000106BC call ds:RtlInitUnicodeString
000107B8 lea eax, [ebp+DeviceObject]	000106B8 lea eax, [ebp+DeviceObject]
000107BC push eax ; DeviceObject	000106BC push eax ; DeviceObject
000107B8 lea eax, [ebp+FileObject]	000106C5 lea eax, [ebp+FileObject]
000107BC push eax ; FileObject	000106C9 push eax ; FileObject
000107C0 push 100000h ; DesiredAccess	000106CA push 100000h ; DesiredAccess
000107C5 lea eax, [ebp+DestinationString]	000106CF lea eax, [ebp+DestinationString]
000107C8 push eax ; ObjectName	000106D0 push eax ; ObjectName
000107C9 call ds:IoGetDeviceObjectPointer	000106D3 call ds:IoGetDeviceObjectPointer
000107CF test eax, eax	000106D9 test eax, eax
000107D1 jnz loc_10977	000106DB jnz loc_10881
000107D7 cmp [ebp+FileObject], eax	000106E1 cmp [ebp+FileObject], eax
000107DA jz loc_10984	000106E4 jz loc_1088E
000107E0 cmp [ebp+DeviceObject], eax	000106EA cmp [ebp+DeviceObject], eax
000107E3 jz loc_10977	000106ED jz loc_10881
000107E9 push [ebp+FileObject]	000106F3 push [ebp+FileObject]
000107EC call IoGetBaseFileSystemDeviceObject	000106F6 call IoGetBaseFileSystemDeviceObject
000107F1 test eax, eax	000106FB test eax, eax
000107F3 mov [ebp+DeviceObject], eax	000106FD mov [ebp+DeviceObject], eax
000107F6 jz loc_10977	00010700 jz loc_10881
000107FC mov eax, dword_113C0	00010706 mov eax, dword_11508
00010801 mov ecx, [eax+4]	0001070B mov ecx, [eax+4]
00010804 mov ecx, [ecx+0Ch]	0001070E mov ecx, [ecx+0Ch]
00010807 lea edx, [ebp+var_4]	00010711 lea edx, [ebp+var_4]
0001080A push edx	00010714 push edx
0001080B push eax	00010715 push eax
0001080C call dword ptr [ecx+2Ch]	00010716 call dword ptr [ecx+2Ch]
0001080F test al, al	00010719 test al, al
00010811	0001071B
00010812	0001071C
00010813	0001071D
00010819	00010723
0001081C mov edx, [ebp+DeviceObject]	00010726 mov edx, [ebp+DeviceObject]
0001081F mov edx, [edx+8]	00010729 mov edx, [edx+8]
00010822 mov ecx, [eax+4]	0001072C mov ecx, [eax+4]
00010825 push dword ptr [edx+44h]	0001072F push dword ptr [edx+44h]
00010828 mov ecx, [ecx+4]	00010732 mov ecx, [ecx+4]
0001082B mov ecx, [ecx+0Ch]	00010735 mov ecx, [ecx+0Ch]
0001082E push eax	00010738 push eax
0001082F call dword ptr [ecx+138h]	00010739 call dword ptr [ecx+138h]
00010835 test al, al	0001073F test al, al
00010837 pop ecx	00010741 pop ecx
00010838 pop ecx	00010742 pop ecx
00010839 jz loc_10963	00010743 jz loc_1086D
0001083F mov edx, [ebp+DeviceObject]	00010749 mov edx, [ebp+DeviceObject]
00010842 mov eax, [ebp+var_4]	0001074C mov eax, [ebp+var_4]

50251.dll (Regin module)

20123.sys ("qwerty")

大部分的 Qwerty 组件从同样的包（插件号码 20121 - 20123）中调用插件，但是有一段代码也存在于 Regin 平台的插件中。这段代码同时存在于 Qwerty 20123 模块和 Regin 50251 模块中，而且寻址 Regin 50225 插件（该插件在 Regin 的虚拟文件系统中）。插件 50225 负责内核模式挂钩。

00010905 mov ecx, [ecx+4]	0001080F mov ecx, [ecx+4]
00010908 mov ecx, [ecx+0Ch]	00010812 mov ecx, [ecx+0Ch]
0001090B push offset Kbdclass_sys ; "kbdclass.sys"	00010815 push offset Kbdclass_sys ; "kbdclass.sys"
00010910 call dword ptr [ecx+120h]	00010818 call dword ptr [ecx+120h]
00010917 test al, al	00010821 test al, al
00010919 pop ecx	00010823 pop ecx
0001091A pop ecx	00010824 pop ecx
0001091B jz short loc_10963	00010825 jz short loc_1086D
0001091D mov eax, [ebp+var_4]	00010827 mov eax, [ebp+var_4]
00010920 mov ecx, [eax+4]	0001082A mov ecx, [eax+4]
00010923 mov ecx, [ecx+4]	0001082D mov ecx, [ecx+4]
00010926 mov ecx, [ecx+0Ch]	00010830 mov ecx, [ecx+0Ch]
00010929 push 7	00010833 push 7
0001092B push 50225	00010835 push 50225
00010930 push 7	0001083A push 7
00010935 push eax	0001083F push eax
00010936 call dword ptr [ecx+0E0h]	00010840 call dword ptr [ecx+0E0h]
0001093C add esp, 10h	00010846 add esp, 10h
0001093F test eax, eax	00010849 test eax, eax
00010941 jnz short loc_10963	0001084B jnz short loc_1086D
00010943 mov eax, [ebp+var_4]	0001084D mov eax, [ebp+var_4]
00010946 mov ecx, [eax+4]	00010850 mov ecx, [eax+4]
00010949 mov ecx, [ecx+4]	00010853 mov ecx, [ecx+4]
0001094C mov ecx, [ecx+0Ch]	00010856 mov ecx, [ecx+0Ch]
0001094F	00010859
00010954	0001085E
00010955	0001085F

50251 (Regin)

20123 ("Qwerty")

有确实的证据显示 Qwerty 插件只能作为 Regin 平台的一部分运作 能够利用插件 50225

的内核挂钩功能。

两个模块使用相同软件平台的另一个证据是两个模块按照次序 1 导出的函数。它们包含可在 Regin 其他任何插件中找到的启动代码，并且包括平台中注册的实际插件号，以允许进一步的模块寻址。只有当模块与 Regin 平台 orchestrator 一起使用时，这才是有道理的。

<pre> 000103EE public _50251_1 000103EE proc near 000103EE = dword ptr 4 000103EE = dword ptr 8 000103EE mov eax, [esp+arg_0] 000103F2 mov ecx, [eax+0Ch] 000103F5 push ebx 000103F6 push 50251 000103F7 ; DATA XREF: off 000103FC xor bl, bl 00010401 call dword ptr [ecx+18h] 00010403 add esp, 0Ch 00010406 test al, al 00010409 jz short loc_1042E 0001040B push [esp+4+arg_4] 0001040D call sub_10E5E 00010411 test al, al 00010416 jz short loc_1041E 00010418 inc bl 0001041C jmp short loc_1042E 0001041E 0001041E loc_1041E: 0001041E mov eax, dword_113C0 00010423 mov ecx, [eax+4] 00010426 mov ecx, [ecx+0Ch] 00010429 push eax 0001042A call dword ptr [ecx+1Ch] 0001042D pop ecx 0001042E loc_1042E: 0001042E mov al, bl 00010430 pop ebx 00010431 retn 00010431 _50251_1 endp </pre>	<pre> 0001041A public _20123_1 0001041A proc near 0001041A = dword ptr 4 0001041A = dword ptr 8 0001041A mov eax, [esp+arg_0] 0001041E mov ecx, [eax+0Ch] 00010421 push ebx 00010422 push 20123 00010423 ; DATA XREF: off 00010428 xor bl, bl 0001042D call dword ptr [ecx+18h] 0001042F add esp, 0Ch 00010432 test al, al 00010435 jz short loc_1045A 00010437 push [esp+4+arg_4] 00010439 call sub_10C28 0001043D test al, al 00010442 jz short loc_1044A 00010444 inc bl 00010448 jmp short loc_1045A 0001044A 0001044A loc_1044A: 0001044A mov eax, dword_11508 0001044F mov ecx, [eax+4] 00010452 mov ecx, [ecx+0Ch] 00010455 push eax 00010456 call dword ptr [ecx+1Ch] 00010459 pop ecx 0001045A loc_1045A: 0001045A mov al, bl 0001045C pop ebx 0001045D retn 0001045D _20123_1 endp </pre>
---	---

为何这两个模块具有不同的插件 ID，这一点尚且未知。也许是因为它们由不同的攻击者使用，有各自分配的插件 ID 范围。

结论

通过分析《明镜周刊》发布的 Qwerty 恶意软件，我们发现它旨在等同于 Regin 平台的一部分。Qwerty 击键记录器不能作为一个独立的模块，它依赖于 Regin 模块 50225 提供的内核挂钩函数。考虑到 Regin 平台的极端复杂性，被不知道源代码的人复制的可能性非常低，所以我们认为 Qwerty 恶意软件开发者和 Regin 开发者是同一伙人或者共同合作。

另一个重要的发现是，Regin 插件存储在加密和压缩的 VFS（虚拟文件系统）之内，这意味着它们并不以“本机”格式直接存在于受害者的机器上。该平台调度器在启动时会加载并执行插件。找到击键记录器的唯一方法是扫描系统内存或解码虚拟文件系统。

附录（MD5）

QWERTY 20123.sys：

0ed11a73694999bc45d18b4189f41ac2

Regin 50251 插件：

c0de81512a08bdf2ec18cb93b43bdc2d

e9a43ea2882ac63b7bc036d954c79aa1