

# Comparing the Regin module 50251 and the "Qwerty" keylogger

By [Costin Raiu](#), [Igor Soumenkov](#) on January 27, 2015. 11:00 am

RESEARCH

---

[APT](#) [KEYLOGGERS](#) [TARGETED ATTACKS](#)



**Costin Raiu**  
[@craiu](#)



**Igor Soumenkov**  
[@soumenkov](#)

On January 17 2015, Spiegel.de published an

extensive article based on documents obtained from Edward Snowden. At the same time, they provided a copy of a malicious program codenamed "QWERTY" (<http://www.spiegel.de/media/media-35668.pdf>), supposedly used by several governments in their CNE operations.

We've obtained a copy of the malicious files published by Der Spiegel and when we analyzed them, they immediately reminded us of [Regin](#). Looking at the code closely, we conclude that the "QWERTY" malware is identical in functionality to the Regin 50251 plugin.

## Analysis

The Qwerty module pack consists of three binaries and accompanying configuration files. One file from the package— 20123.sys – is particularly interesting.

The "20123.sys" is a kernel mode part of the keylogger. As it turns out, it was built from source code that can also be found one Regin module, the "50251" plugin.

Using a binary diff it is easy to spot a significant part of code that is shared between both files:

614	18A9		
62B	e0 74 18 80 7d e7 01 75 09 ff 75 e0 ff 15 14 02 01 00 ff 75 e0 ff 15	..t..J..U..U.....U..	
642	f8 02 01 00 33 c0 e8 13 08 00 00 c2 10 00 cc 53 68 c0 14 01 00 6a 0d	3.....Sh.....i	
659	6a 01 68 01 00 00 7f ff 35 08 15 01 00 32 db e8 f0 0b 00 00 85 c0 75	j.h.....5.....2.....U	
670	02 fe c3 8a c3 5b c3 5c 00 44 00 65 00 76 00 69 00 63 00 65 00 5c 00	.....I..V..D.e.v.i.c.e.\	
687	4b 00 65 00 79 00 62 00 6f 00 61 00 72 00 64 00 43 00 6c 00 61 00 73	K.e.y.b.o.a.r.d.C.l.a.s	
69E	00 73 00 30 00 00 00 00 00 6b 00 62 00 64 00 63 00 6c 00 61 00 73 00	.s.0.....k.b.d.c.l.a.s	
6B5	73 00 2e 00 73 00 79 00 73 00 00 55 8b ec 83 ec 14 53 68 60 06 01	s.....s.y.s.....U.....Sh	
6CC	00 8d 45 ec 50 32 db ff 15 b8 02 01 00 8d 45 f8 50 8d 45 f4 50 68 00	..E.P2.....E.P.E.Ph	
6E3	00 10 00 8d 45 ec 50 ff 15 c4 02 01 00 85 c0 0f 85 a0 01 00 00 39 45	....E.P.....9E	
6FA	f4 0f 84 a4 01 00 00 39 45 f8 0f 84 8e 01 00 00 ff 75 f4 e8 33 08 00	.....9E.....U..3..	
711	00 85 c0 89 45 f8 0f 84 7b 01 00 00 a1 08 15 01 00 8b 48 04 8b 49 0c	....E...C.....H..I	
728	8d 55 fc 52 50 ff 51 2c 84 c0 59 59 0f 84 5e 01 00 00 8b 45 fc 8b 55	..U.RP.Q...YY..^.....E..U	
73F	f8 8b 52 08 8b 48 04 ff 72 44 8b 49 04 8b 49 0c 50 ff 91 38 01 00 00	..R...H..rD.I...I.P..8...	
756	84 c0 59 59 0f 84 24 01 00 00 8b 55 f8 8b 45 fc 8b 52 08 8b 48 04 8b	..YY..\$....U..E..R..H...	
76D	49 04 8b 49 0c 83 c2 44 52 50 ff 91 38 01 00 00 84 c0 59 59 0f 84 fd	I..I...DRP..8.....YY...	
784	00 8d 45 ec 50 32 db ff 84 8b 49 04 8b 49 0c 68 5a 05 01 00 50 ff 91	....E...H..I..I.hZ...P...	
79B	38 01 00 00 84 c0 59 59 0f 84 db 00 00 00 8b 45 fc 8b 48 04 8b 49 04	8.....YY.....E..H..I	
7B2	8b 49 0c 68 01 00 00 50 ff 91 2c 01 00 00 84 c0 59 59 0f 84 b9 00	..I.h...P.....YY...	
7C9	00 00 8b 45 fc 8b 48 04 8b 49 04 8b 49 0c 6a 00 50 ff 91 2c 01 00 00	....E..H..I..I..j.P.....	
7E0	84 c0 59 59 0f 84 9a 00 00 00 8b 45 fc 8b 48 04 8b 49 04 8b 49 0c 6a	..YY.....E..H..I..I...	
7F7	02 50 ff 91 24 01 00 00 84 c0 59 59 74 7f 8b 45 fc 8b 48 04 8b 49 04	..P..\$....YYt...E..H..I	
80E	8b 49 0c 6a 01 50 ff 91 24 01 00 00 84 c0 59 59 74 64 8b 45 fc 8b 48	..I..j.P..\$....YYtd.E..H	
825	04 8b 49 04 8b 49 0c 68 90 06 01 00 50 ff 91 20 01 00 00 84 c0 59 59	....I..I.h...P.....YY...	
83C	74 46 8b 45 fc 8b 48 04 8b 49 04 8b 49 0c 6a 07 68 31 c4 00 00 68 01	tF.E..H..I..I..j.h1...h	
853	00 00 7f 50 ff 91 e0 00 00 83 c4 10 85 c0 75 20 8b 45 fc 8b 48 04	...P.....U..E..H	
86A	8b 49 04 8b 49 0c 68 00 15 01 00 50 ff 91 bc 01 00 00 84 c0 59 59 74	..I..I.h...P.....YY...	
881	02 fe c3 8b 45 fc 8b 40 04 8b 40 04 8b 40 0c 8d 4d fc 51 ff 50 34 59	....E...@...@...M.Q.P4V	
898	8b 4d f4 85 c9 74 06 ff 15 d0 02 01 00 8a c3 5b c9 c3 cc 55 8b ec 83	..M...t.....I...U...	
8AF	ec 0c 53 8d 45 f4 50 8d 45 f8 50 6a 01 68 01 00 00 7f ff 35 08 15 01	..S.E.P.E.P.j.h...5...	
8C6	00 c6 45 ff 00 32 db e8 39 0a 00 00 85 c0 75 28 83 7d f8 0d 75 12 56	..E...2...9.....u&...u.V	
8DD	50 e8 01 0b 00 00	..U.W.....2...u.E.P.	
8F4	00 c7 05 c9 14 01	P.....U...%.....	
	20123 ("qwerty")		
702	1803		
719	5d e0 74 18 80 7d e7 01 75 09 ff 75 e0 ff 15 d0 02 01 00 ff 75 e0 ff	1..t..J..U..U.....U..	
730	15 d8 02 01 00 33 c0 e8 54 08 00 00 c2 10 00 53 68 60 13 01 00 6a 05	3.....T.....Sh.....i	
747	6a 01 68 01 00 00 7f ff 35 08 13 01 00 32 db e8 6a 0a 00 00 85 c0 75	j.h.....5.....2...j.....U	
75E	02 fe c3 8a c3 5b c3 5c 00 44 00 65 00 76 00 69 00 63 00 65 00 5c 00	.....I..V..D.e.v.i.c.e.\	
775	4b 00 65 00 79 00 62 00 6f 00 61 00 72 00 64 00 43 00 6c 00 61 00 73	K.e.y.b.o.a.r.d.C.l.a.s	
78C	00 73 00 30 00 00 00 00 00 6b 00 62 00 64 00 63 00 6c 00 61 00 73 00	.s.0.....k.b.d.c.l.a.s	
7A3	73 00 2e 00 73 00 79 00 73 00 00 55 8b ec 83 ec 14 53 68 4e 07 01	s.....s.y.s.....U.....ShN	
7BA	00 8d 45 ec 50 32 db ff 15 d4 02 01 00 8d 45 f8 50 8d 45 f4 50 68 00	..E.P2.....E.P.E.Ph	
7D1	00 10 00 8d 45 ec 50 ff 15 b8 02 01 00 85 c0 0f 85 a0 01 00 00 39 45	....E.P.....9E	
7E8	f4 0f 84 a4 01 00 00 39 45 f8 0f 84 8e 01 00 00 ff 75 f4 e8 75 08 00	.....9E.....U..3..	
7FF	00 85 c0 89 45 f8 0f 84 7b 01 00 00 a1 a0 13 01 00 8b 48 04 8b 49 0c	....E...C.....H..I	
816	8d 55 fc 52 50 ff 51 2c 84 c0 59 59 0f 84 5e 01 00 00 8b 45 fc 8b 55	..U.RP.Q...YY..^.....E..U	
82D	84 c0 59 59 0f 84 24 01 00 00 8b 55 f8 8b 45 fc 8b 52 08 8b 48 04 8b	..YY..\$....U..E..R..H...	
844	49 04 8b 49 0c 83 c2 44 52 50 ff 91 38 01 00 00 84 c0 59 59 0f 84 fd	I..I...DRP..8.....YY...	
85B	00 00 00 8b 45 fc 8b 48 04 8b 49 04 8b 49 0c 68 30 06 01 00 50 ff 91	....E...H..I..I.h0...P...	
872	38 01 00 00 84 c0 59 59 0f 84 db 00 00 00 8b 45 fc 8b 48 04 8b 49 04	8.....YY.....E..H..I	
889	8b 49 0c 68 01 00 00 50 ff 91 2c 01 00 00 84 c0 59 59 0f 84 b9 00	..I.h...P.....YY...	
8A0	00 00 8b 45 fc 8b 48 04 8b 49 04 8b 49 0c 6a 00 50 ff 91 2c 01 00 00	....E..H..I..I..j.P.....	
8B7	84 c0 59 59 0f 84 9a 00 00 00 8b 45 fc 8b 48 04 8b 49 04 8b 49 0c 6a	..YY.....E..H..I..I...	
8CE	02 50 ff 91 24 01 00 00 84 c0 59 59 74 7f 8b 45 fc 8b 48 04 8b 49 04	..P..\$....YYt...E..H..I	
8E5	8b 49 0c 6a 01 50 ff 91 24 01 00 00 84 c0 59 59 74 64 8b 45 fc 8b 48	..I..j.P..\$....YYtd.E..H	
8FC	04 8b 49 04 8b 49 0c 68 7e 07 01 00 50 ff 91 20 01 00 00 84 c0 59 59	....I..I.h...P.....YY...	
913	74 46 8b 45 fc 8b 48 04 8b 49 04 8b 49 0c 6a 07 68 31 c4 00 00 68 01	tF.E..H..I..I..j.h1...h	
92A	00 00 7f 50 ff 91 e0 00 00 83 c4 10 85 c0 75 20 8b 45 fc 8b 48 04	...P.....U..E..H	
941	8b 49 04 8b 49 0c 68 74 13 01 00 50 ff 91 bc 01 00 00 84 c0 59 59 74	..I..I.h...P.....YY...	
958	02 fe c3 8b 45 fc 8b 40 04 8b 40 04 8b 40 0c 8d 4d fc 51 ff 50 34 59	....E...@...@...M.Q.P4V	
96F	8b 4d f4 85 c9 74 06 ff 15 b0 02 01 00 8a c3 5b c9 c3 cc 55 8b ec 83	..M...t.....I...U...	
986	ec 0c 53 8d 45 f4 50 8d 45 f8 50 6a 01 68 01 00 00 7f ff 35 a0 13 01	..S.E.P.E.P.j.h...5...	
99D	00 c6 45 ff 00 32 db e8 b3 08 00 00 85 c0 75 26 83 7d f8 05 75 10 56	..E...2.....u&...u.V	
9B4	00 75 14 57 0f 00	..U.W.....2...u.E.P.	
9CB	7d 09 00 00 84 db	3.....U.....a.....	
9E2	00 00 e8 3f fd ff	...?...hddk...5a...j...	
	50251 (Regin module)		

Most of the shared code belongs to the function that accesses the system keyboard driver:



<pre> 00010756 DeviceKeyboardclass0: ; DATA XREF: sub 00010756 unicode 0, &lt;\Device\KeyboardClass0&gt;,0 00010784 db 2 dup(0) 00010786 ; DATA XREF: sub 00010786 Kbdclass_sys: unicode 0, &lt;kbdclass.sys&gt;,0 000107A0 ; ===== SUBROUTINE ===== 000107A0 ; Attributes: bp-based frame 000107A0 sub_107A0 proc near ; CODE XREF: sub 000107A0 DestinationString= UNICODE_STRING ptr -14h 000107A0 FileObject = dword ptr -0Ch 000107A0 DeviceObject = dword ptr -8 000107A0 var_4 = dword ptr -4 000107A0 push ebp 000107A1 mov ebp, esp 000107A3 sub esp, 14h 000107A6 push ebx 000107A7 push offset DeviceKeyboardclass0 ; S 000107AC lea eax, [ebp+DestinationString] 000107AF push eax ; DestinationStr 000107B0 xor bl, bl 000107B2 call ds:RtlInitUnicodeString 000107B8 lea eax, [ebp+DeviceObject] 000107BB push eax ; DeviceObject 000107BC lea eax, [ebp+FileObject] 000107BF push eax ; FileObject 000107C0 push 100000h ; DesiredAccess 000107C5 lea eax, [ebp+DestinationString] 000107C8 push eax ; ObjectName 000107C9 call ds:IoGetDeviceObjectPointer 000107CF test eax, eax 000107D1 jnz loc_10977 000107D7 cmp [ebp+FileObject], eax 000107DA jz loc_10984 000107E0 cmp [ebp+DeviceObject], eax 000107E3 jz loc_10977 000107E9 push [ebp+FileObject] 000107EC call IoGetBaseFileSystemDeviceObject 000107F1 test eax, eax 000107F3 mov [ebp+DeviceObject], eax 000107F6 jz loc_10977 000107FC mov eax, dword_113C0 00010801 mov ecx, [eax+4] 00010804 mov ecx, [ecx+0Ch] 00010807 lea edx, [ebp+var_4] 0001080A push edx 0001080B push eax 0001080C call dword ptr [ecx+2Ch] 0001080F test al, al 00010811 00010812 00010813 00010819 0001081C mov edx, [ebp+DeviceObject] 0001081F mov edx, [edx+8] 00010822 mov ecx, [eax+4] 00010825 push dword ptr [edx+44h] 00010828 mov ecx, [ecx+4] 0001082B mov ecx, [ecx+0Ch] 0001082E push eax 0001082F call dword ptr [ecx+138h] 00010835 test al, al 00010837 pop ecx 00010838 pop ecx 00010839 jz loc_10963 0001083F mov edx, [ebp+DeviceObject] 00010842 mov eax, [ebp+var_4] </pre>	<pre> 00010660 DeviceKeyboardclass0: ; DATA XREF: sub 00010660 unicode 0, &lt;\Device\KeyboardClass0&gt;,0 0001068E align 10h 00010690 ; DATA XREF: sub 00010690 Kbdclass_sys: unicode 0, &lt;kbdclass.sys&gt;,0 000106AA ; ===== SUBROUTINE ===== 000106AA ; Attributes: bp-based frame 000106AA sub_106AA proc near ; CODE XREF: sub 000106AA DestinationString= UNICODE_STRING ptr -14h 000106AA FileObject = dword ptr -0Ch 000106AA DeviceObject = dword ptr -8 000106AA var_4 = dword ptr -4 000106AA push ebp 000106AB mov ebp, esp 000106AD sub esp, 14h 000106B0 push ebx 000106B1 push offset DeviceKeyboardclass0 ; "\ 000106B6 lea eax, [ebp+DestinationString] 000106B9 push eax ; DestinationStr 000106BA xor bl, bl 000106BC call ds:RtlInitUnicodeString 000106C2 lea eax, [ebp+DeviceObject] 000106C5 push eax ; DeviceObject 000106C6 lea eax, [ebp+FileObject] 000106C9 push eax ; FileObject 000106CA push 100000h ; DesiredAccess 000106CF lea eax, [ebp+DestinationString] 000106D2 push eax ; ObjectName 000106D3 call ds:IoGetDeviceObjectPointer 000106D9 test eax, eax 000106DB jnz loc_10881 000106E1 cmp [ebp+FileObject], eax 000106E4 jz loc_1088E 000106EA cmp [ebp+DeviceObject], eax 000106ED jz loc_10881 000106EF push [ebp+FileObject] 000106F3 call IoGetBaseFileSystemDeviceObject 000106F8 test eax, eax 000106FD mov [ebp+DeviceObject], eax 00010700 jz loc_10881 00010706 mov eax, dword_11508 0001070B mov ecx, [eax+4] 0001070E mov ecx, [ecx+0Ch] 00010711 lea edx, [ebp+var_4] 00010714 push edx 00010715 push eax 00010716 call dword ptr [ecx+2Ch] 00010719 test al, al 0001071B 0001071C 0001071D 00010723 00010726 mov edx, [ebp+DeviceObject] 00010729 mov edx, [edx+8] 0001072C mov ecx, [eax+4] 0001072F push dword ptr [edx+44h] 00010732 mov ecx, [ecx+4] 00010735 mov ecx, [ecx+0Ch] 00010738 push eax 00010739 call dword ptr [ecx+138h] 0001073F test al, al 00010741 pop ecx 00010742 pop ecx 00010743 jz loc_1086D 00010749 mov edx, [ebp+DeviceObject] 0001074C mov eax, [ebp+var_4] </pre>
---	--

50251.dll (Regin module)

20123.sys ("qwerty")

Most of the "Qwerty" components call plugins from the same pack (with plugin numbers 20121 – 20123), however there is also one piece code that references plugins from the Regin platform. One particular part of code is used in both the "Qwerty" 20123 module and the Regin's 50251 counterpart, and it addresses the plugin 50225 that can be found in the virtual filesystems of Regin. The Regin's plugin 50225 is responsible for

## kernel-mode hooking.

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	push	eax	0001081A	push	eax
00010911	call	dword ptr [ecx+120h]	0001081B	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	77000000h	0001083A	push	77000000h
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")

This is a **solid proof that the Qwerty plugin can only operate as part of the Regin platform**, leveraging the kernel hooking functions from plugin 50225.

As an **additional proof that both modules use the same software platform**, we can take a look at functions exported by ordinal 1 of both modules. They contain the startup code that can be found in any other plugin of Regin, and include the actual plugin number that is registered within the platform to allow further addressing of the module. This only makes sense if the modules are used with the Regin platform orchestrator.

```

000103EE public _50251_1
000103EE proc near ; DATA XREF: off
000103EE arg_0 = dword ptr 4
000103EE arg_4 = dword ptr 8
000103EE mov     eax, [esp+arg_0]
000103F2 mov     ecx, [eax+0Ch]
000103F5 push    ebx
000103F6
000103F7 push    50251
000103FC
00010401 xor     bl, bl
00010403 call    dword ptr [ecx+18h]
00010406 add     esp, 0Ch
00010409 test    al, al
0001040B jz      short loc_1042E
0001040D push    [esp+4+arg_4]
00010411 call    sub_10E5E
00010416 test    al, al
00010418 jz      short loc_1041E
0001041A inc     bl
0001041C jmp     short loc_1042E
0001041E ;
0001041E loc_1041E:
0001041E mov     eax, dword_113C0 ; CODE XREF: _50
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 ; CODE XREF: _50
00010430 pop     ebx
00010431 retn
00010431 _50251_1 endp

0001041A public _20123_1
0001041A proc near ; DATA XREF: off
0001041A arg_0 = dword ptr 4
0001041A arg_4 = dword ptr 8
0001041A mov     eax, [esp+arg_0]
0001041E mov     ecx, [eax+0Ch]
00010421 push    ebx
00010422
00010423 push    20123
00010428
0001042D xor     bl, bl
0001042F call    dword ptr [ecx+18h]
00010432 add     esp, 0Ch
00010435 test    al, al
00010437 jz      short loc_1045A
00010439 push    [esp+4+arg_4]
0001043D call    sub_10C28
00010442 test    al, al
00010444 jz      short loc_1044A
00010446 inc     bl
00010448 jmp     short loc_1045A
0001044A ;
0001044A loc_1044A:
0001044A mov     eax, dword_11508 ; CODE XREF: _20
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 ; CODE XREF: _20
0001045C pop     ebx
0001045D retn
0001045D _20123_1 endp

```

The reason why the two modules have different plugin IDs is unknown. This is perhaps because they are leveraged by different actors, each one with its own allocated plugin ID ranges.

## Conclusions

Our analysis of the QWERTY malware published by Der Spiegel indicates it is a plugin designed to work part of the Regin platform. The QWERTY keylogger doesn't function as a stand-alone module, it relies on kernel hooking functions which are provided by the Regin module 50225. Considering the extreme complexity of the Regin platform and little chance that it can be duplicated by somebody without having access to its sourcecodes, we conclude the QWERTY malware developers and the Regin developers are the same or working together.

Another important observation is that Regin plugins are stored inside an encrypted and compressed VFS, meaning they don't exist directly on the victim's machine in "native" format. The platform dispatcher loads and executes there plugins at startup. The only way to catch the keylogger is by scanning the system memory or decoding the VFSes.

## Appendix (MD5 hashes):

**QWERTY 20123.sys:**

1 0ed11a73694999bc45d18b4189f41ac2
------------------------------------

**Regin 50251 plugins:**

1 c0de81512a08bdf2ec18cb93b43bdc2d
2 e9a43ea2882ac63b7bc036d954c79aa1