

## Модульный троян для скрытого доступа к компьютеру

Archived: 2026-04-05 21:04:02 UTC



Как показывает практика, получить скрытый удаленный доступ к вашему компьютеру, заполучить пароли, записать видео или звук уже не является тяжело решаемой задачей. И в принципе, кроме персональной осведомленности в вопросах информационной безопасности, иного действенного средства не придумаешь. В данной статье предлагаем ознакомиться с базовым анализом многомодульной троянской программы и немного сблизиться с актуальными в нынешнее время техниками и средствами, используемыми злоумышленниками.

Совсем недавно мы наткнулись на два интересных образца вредоносных программ. Наряду с тем, что вредоносы распространялись в виде документов MS Word с эксплойтом «на борту», их структура предполагала модульность, а степень выявления антивирусными продуктами была невысока в связи с достаточно высоким уровнем обфускации [1] за счет многократного шифрования. Мы решили взяться за изучение этого malware, а результат изложить в этой статье.

Так выглядели два файла, попавшие в наше поле зрения:

**Reported for TRMSCD3L Licence expired.doc**

**Police report remit expired Licence.doc**

Очень часто, исходя из названия, можно дедуктивно догадаться, «кем» и кому предназначен тот или иной вредоносный файл (к примеру: «Договор.doc», «Рахунок на оплату.doc» – для бухгалтеров, «Приказ НБУ №159.doc» – банкам, «Списки захопл. в зоні АТО.doc» – военным, и т.д.). В нашем случае мы можем только догадываться, так как:

- слово «Licence» – написано с ошибкой;
- слова «Licence expired» и «expired Licence» говорят о том, что тематика может касаться срока действия какой-то лицензии;
- строка «TRMSCD3L», если поискать в Google, вообще ассоциируется со SWIFT CODE банка TRUST MERCHANT BANK SARL, находящегося в Демократической республике Конго (г. Лубумбаши).

В общем, в данном случае понять трудно. Видимо, атакующий очень хорошо знал жертву, так как при именовании вредоносных файлов указал целый набор едва связанных посылов.

Как правило, изучение вредоносных программ осуществляется с помощью динамического (если возможно) и статического анализа. В статье опишем результаты изучения файла «Reported for TRMSCD3L Licence expired.doc».

### Базовый динамический анализ

При открытии файла с помощью MS Word происходит эксплуатация уязвимости (если Вы на своем ПК не обновляли MS Word с 2012 года) и на компьютере создается файл с произвольным именем:

```
Path: C:\Users%\USERNAME%\AppData\Local\Microsoft\Windows\  
FileName: yoymbgp.exe  
Md5: 11b6a2ea17d18c09cc274731f05e89b5
```

Помимо этого, с целью обеспечения выживаемости, файл добавляется в автозагрузку (рис. 1), для чего в реестр вносятся соответствующие изменения (имя значения также является произвольным):

```
RegKey Data: C:\Users%\USERNAME% \AppData\Local\Microsoft\Windows\ yoymbgp.exe  
RegKey Data Type: REG_SZ  
RegKey Value Name: atk17n5rAA  
RegKey Name: HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

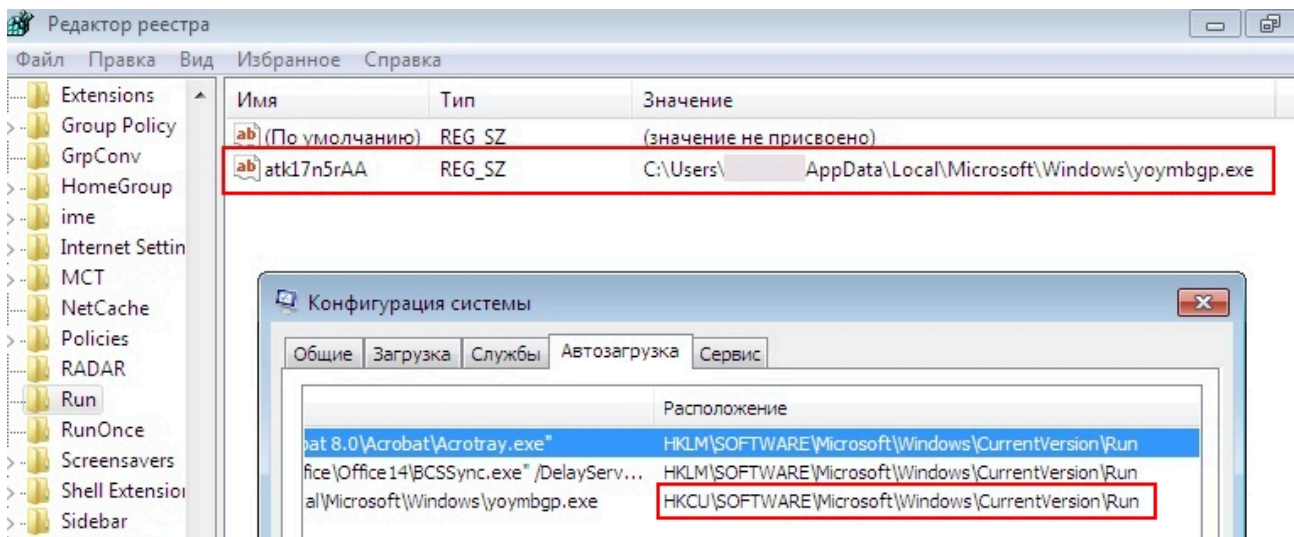


Рис. 1

Хоть это и примитивно, но мы советуем посматривать в автозагрузку вашего ПК и не допускать присутствия там неведомых вам программ. Чтобы проверить автозагрузку можно воспользоваться как штатным функционалом вашей операционной системы (в командной строке: regedit, msconfig), так и сторонним программным обеспечением.

На следующем этапе вредоносная программа запускает легитимные процессы svchost.exe и внедряет в них вредоносный код, с целью его последующей расшифровки. Эта процедура повторяется несколько раз (вначале статьи мы упоминали о многократном шифровании). Также, на этом этапе осуществляется взаимодействие между зараженным компьютером и сервером управления, откуда вредонос скачивает модули. Пример такого сетевого взаимодействия:

HTTP GET-запросы:

```
hxxp:// 95.211.204.14:80/m/228131.zip  
hxxp:// 95.211.204.14:80/m/721118.zip  
hxxp:// 95.211.204.14:80/m/958232.zip  
hxxp:// 95.211.204.14:80/m/855787.zip  
hxxp:// 95.211.204.14:80/m/5594516.zip
```

HTTP POST-запросы:

```
hxxp://95.211.204.14:443/$rdgate?ACTION=HELLO  
hxxp://95.211.204.14:443/$rdgate?ACTION=START&ID=877A74B0199241848C931A962ADC55EB  
hxxp://95.211.204.14:80/test.php  
hxxp://95.211.204.14:80/f9S52tseWPcDgVbEY+EbYJ4RhUnZm=AUM9a6oYAc0AV1yFsXJvsCWAAbvctbESCEjhquFuqiNqF7U
```

## Статический анализ

Данный образец вредоносной программы был изрядно запакован – прежде чем приступить к его детальному изучению нам пришлось снять три «слоя» обфускации:

```
[binary_layer1.exe][6197736aec27686efb6f63e75cac0abd]  
[binary_layer2.exe][6c7441ec1b630fd299d8196367aefeea]  
[binary_layer3.exe][9347cb20a1ec90c61e1ff8ded379fc81]
```

Последующее изучение проводилось в отношении распакованного образца «binary\_layer3.exe». При запуске вредоносной программы первым делом она осуществляет проверку на предмет наличия на зараженном ПК необходимых модулей («check\_payload\_plugin»), и, при отсутствии, инициирует их скачивание с сети Интернет. Специально для скачивания создается отдельный поток по адресу 0x00419A60 (мы его назвали «create\_thread\_3»).

```

CODE: 0041C222      mov     [edx], al
CODE: 0041C224      push   0
CODE: 0041C226      push   0
CODE: 0041C228      mov     eax, ds:off_41E364
CODE: 0041C22D      push   eax
CODE: 0041C22E      mov     ecx, offset sub_41BE34
CODE: 0041C233      xor     edx, edx
CODE: 0041C235      xor     eax, eax
CODE: 0041C237      call   sub_4041C8
CODE: 0041C23C      call   check_payload_plugin
CODE: 0041C241      test   al, al
CODE: 0041C243      jz     short loc_41C24C
CODE: 0041C245      call   create_thread_2
CODE: 0041C24A      jmp    short loc_41C25F
CODE: 0041C24C      ; -----
CODE: 0041C24C      loc_41C24C: ; CODE XREF: sub_41C0B4+18F↑j
CODE: 0041C24C      call   create_thread_3
CODE: 0041C251      jmp    short loc_41C25F
CODE: 0041C253      ; -----
CODE: 0041C253
    
```

Рис. 2

Условием выполнения потока 2 («create\_thread\_2») является наличие в реестре скачанных и записанных в реестр модулей вредоносной программы (рис. 3-4):

```

RagKey Name:      HKCU\Software\Google\Update\network\secure

0041A087  C745 F8 010000 MOV  DWORD PTR SS:[EBP-8],80000001
0041A08E  E8 B9D5FEFF  CALL svchost.0040764C
0041A093  8946 F0      MOV  DWORD PTR SS:[EBP-10],EAX
0041A096  33C0        XOR  EAX,EAX
0041A098  55          PUSH EBP
0041A099  68 3DA24100 PUSH svchost.0041A23D
0041A09E  64:FF30     PUSH DWORD PTR FS:[EAX]
0041A0A1  64:8920     MOV  DWORD PTR FS:[EAX],ESP
0041A0A4  8D55 E0      LEA EDX,[EBP-20]
0041A0A7  B8 80A24100 MOV  EAX,svchost.0041A280
0041A0AC  E8 C326FFFF  CALL svchost.0040C774
0041A0B1  8B55 E0      MOV  EDX,DWORD PTR SS:[EBP-20]
0041A0B4  8B46 F8      MOV  EAX,DWORD PTR SS:[EBP-8]
0041A0B7  E8 5C6FFFFF  CALL svchost.00411018
0041A0BC  8946 F4      MOV  DWORD PTR SS:[EBP-0C],EAX
0041A0BF  33C0        XOR  EAX,EAX
0041A0C1  55          PUSH EBP
0041A0C2  68 F5A04100 PUSH svchost.0041A0F5
0041A0C7  64:FF30     PUSH DWORD PTR FS:[EAX]
    
```

Рис. 3

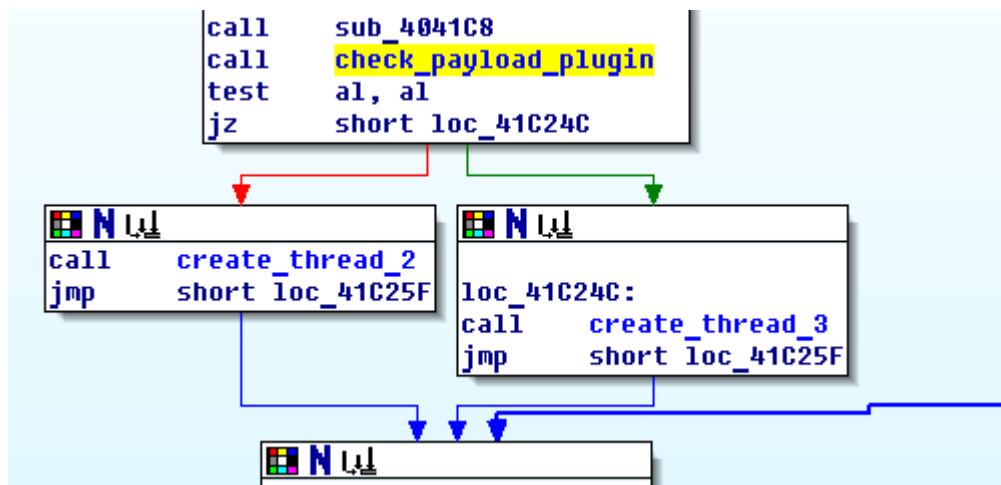


Рис. 4

Пример состояния реестра после скачивания необходимых модулей приведен на рис. 5. Все скачанные модули хранятся в реестре в зашифрованном виде. Имена ключей в реестре также являются произвольными.

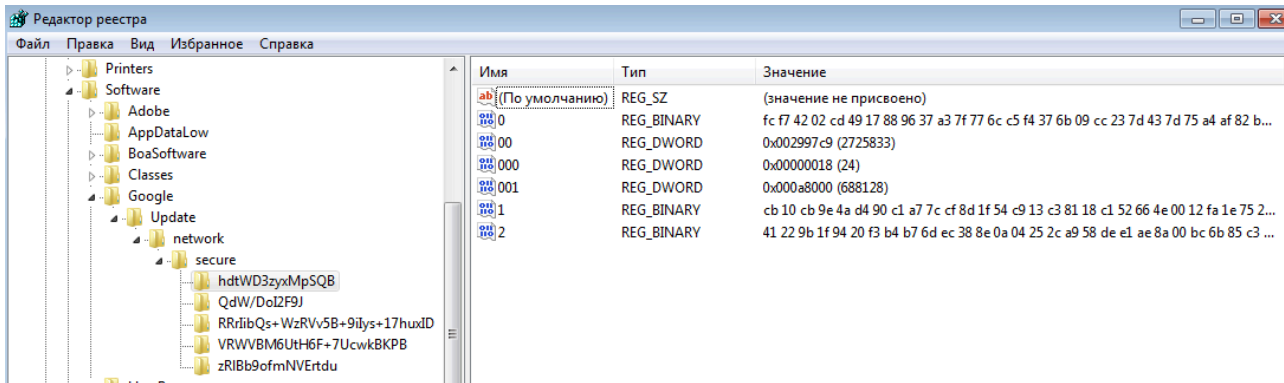
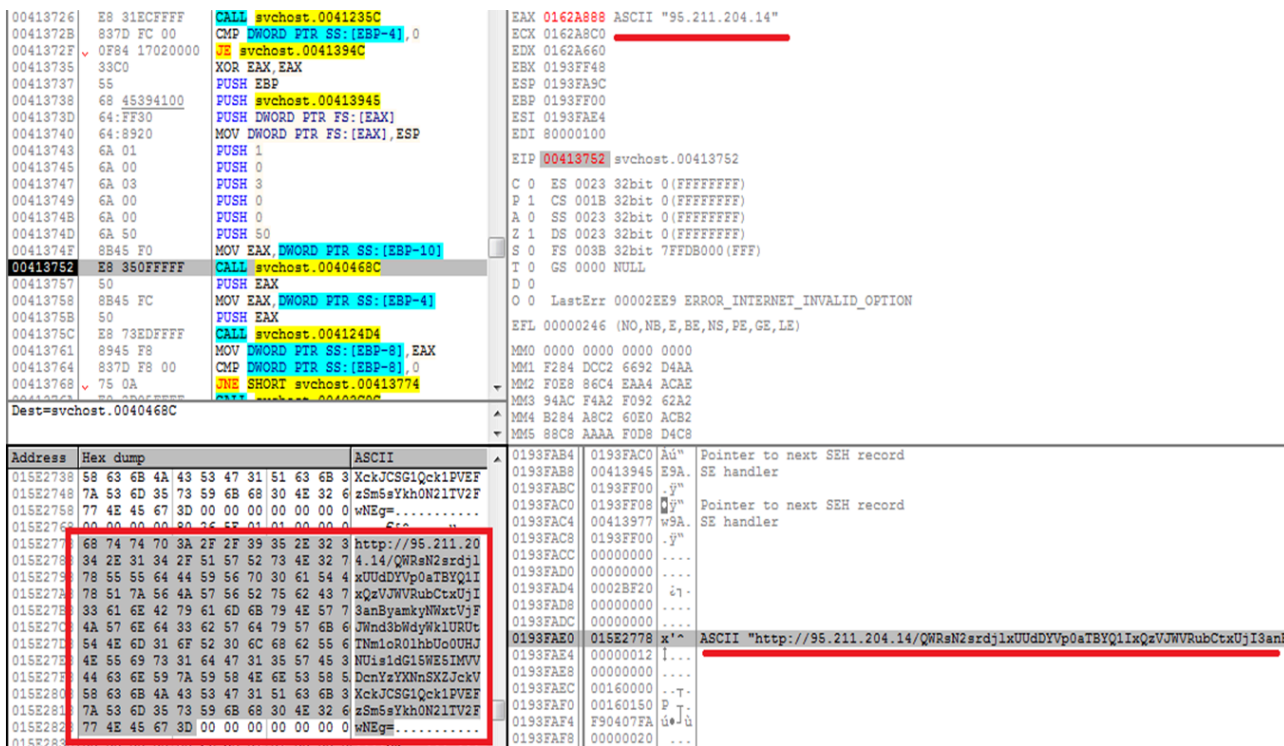


Рис. 5

Некоторые артефакты сетевого взаимодействия зараженного ПК и сервера управления (IP-адрес сервера управления, а также URL, используемый для отправки HTTP-запроса):



```

• CODE:0041A7D1      push    ebp
• CODE:0041A7D2      push    offset loc_41A804
• CODE:0041A7D7      push    dword ptr fs:[eax]
• CODE:0041A7DA      mov     fs:[eax], esp
• CODE:0041A7DD      cmp     [ebp+var_C], 0
• CODE:0041A7E1      jz     short loc_41A7EE
• CODE:0041A7E3      mov     edx, [ebp+var_10]
• CODE:0041A7E6      mov     eax, [ebp+var_C]
• CODE:0041A7E9      call   enum_modules_in_registry
    
```

Рис. 7

0041A8E7	8D55 C0	LEA EDX, [EBP-40]	
0041A8EA	B8 5CAA4100	MOV EAX, svchost.0041AA5C	ASCII "JxSoqd7H1PzCcC"
0041A8EF	E8 801EFFFF	CALL svchost.0040C774	
0041A8F4	8B45 C0	MOV EAX, DWORD PTR SS:[EBP-40]	
0041A8F7	8D55 C4	LEA EDX, [EBP-3C]	
0041A8FA	E8 05DFFEFF	CALL svchost.00408804	
0041A8FF	8B45 C4	MOV EAX, DWORD PTR SS:[EBP-3C]	ASCII "BOT_ENGINE"
0041A902	8D55 C8	LEA EDX, [EBP-38]	
0041A905	E8 62DFFEFF	CALL svchost.0040886C	
0041A90A	8B55 C8	MOV EDX, DWORD PTR SS:[EBP-38]	
0041A90D	58	POP EAX	
0041A90E	E8 C59CFE9F	CALL svchost.004045D8	

Рис. 8

```

mov     eax, [ebp+var_0]
call   sub_410F74
test    al, al
jnz    loc_4198EB
    
```

```

mov     [ebp+var_1E], 0B483h
mov     [ebp+var_1C], 1D09h
mov     [ebp+var_1A], 30CCh
call   sub_407640
mov     [ebp+var_18], eax
xor     edx, edx
push   ebp
push   offset loc_419E45
push   dword ptr fs:[edx]
mov     fs:[edx], esp
mov     eax, [ebp+var_10]
call   sub_407810
    
```

Рис. 9

После успешной расшифровки данных, записанных под непонятными именами в реестр, получаем читаемый и говорящий сам за себя список модулей.

- Bot\_Engine.bin**
- PONY\_STEALER.bin**
- REMOTE\_DESKTOP\_SERVICE.bin**
- SECURITY.bin**
- VNC\_HIDE\_DESKTOP.bin**

Алгоритм расшифровки модулей может быть описан таким псевдокодом:

### Переменные:

enc_byte	зашифрованный байт
dec_byte	расшифрованный байт
HIBYTE	извлекает старший байт из данного 16-разрядного значения
LOBYTE	извлекает младший байт из данного 16-разрядного значения
HIWORD	извлекает старшее слово из данного 32-разрядного значения
LOWORD	извлекает младшее слово из данного 32-разрядного значения
_DWORD	двойное слово (4 байта)
_WORD	слово (2 байта)
key	ключ, состоящий из 6 байт
size	размер зашифрованного массива

### Псевдокод:

```

v0 := *(_DWORD*) key;
v1 := *(_WORD*) (key + 4);
for (i := 0; i < size; i++)
{
    dec_byte := enc_byte ^ HIBYTE(v1);
    v1 := HIWORD(v0) + LOWORD(v0) * (uint8)(LOBYTE(v1) + dec_byte);
    ++enc_byte;
    ++dec_byte;
    --size;
}

```

Рассмотрим полученные модули по порядку.

### Bot\_Engine.bin

Этот модуль должен быть запущен под названием svchost.exe или explorer.exe, иначе он заблаговременно закончит свое исполнение. Пример проверки имени процесса представлен на рис. 10.

```

* CODE: 0042D204      mov     edx, [ebp-2Ch]
* CODE: 0042D207      pop     eax
* CODE: 0042D208      call   @@LStrCmp      ; __linkproc__ LStrCmp
- CODE: 0042D20D      jz     short loc_42D249
* CODE: 0042D20F      lea   eax, [ebp-38h]
* CODE: 0042D212      call   sub_42CC20
* CODE: 0042D217      mov   eax, [ebp-38h]

```

Рис. 10-1

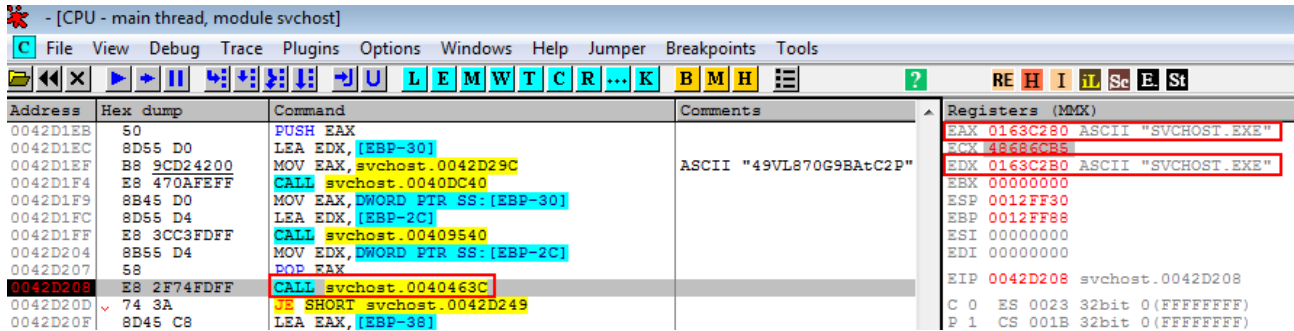


Рис. 10-2

Основным предназначением модуля является запуск других модулей. Пример создания потоков для других модулей отображен на рис. 11.

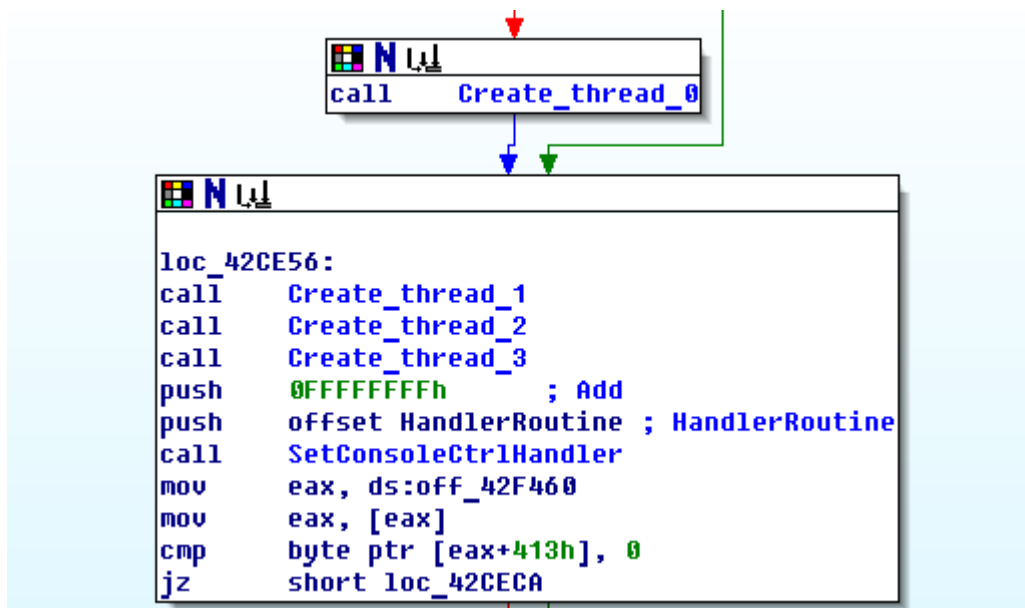


Рис. 11

### PNY\_STEALER.bin

Этот модуль представляет собою не что иное, как одноименную вредоносную программу Pony, функционал которой предусматривает хищение логинов, паролей, кошельков криптовалют. На момент исследования административная панель сервера управления Pony находилась по адресу: <http://95.211.204.14/test.php>

### REMOTE\_DESKTOP\_SERVICE.bin

Модуль, обеспечивающий скрытый удаленный доступ к компьютеру жертвы посредством протокола RDP и создания обратного соединения (back-connect).

Адрес для обратного соединения:

95.211.204.14:443

### SECURITY.bin

Данный модуль обеспечивал проверку фактов присутствия на компьютере антивирусного программного

обеспечения и других, обеспечивающих безопасность компьютера, программ (рис. 12).

Address	Hex dump	ASCII
01685E90	00 00 00 00 0A 00 00 00 61 7A 74 66 76 67 79 67	.....aztfvgyg
01685EA0	79 69 00 46 69 6C 65 73 5C 54 72 65 6E 64 20 4D	yi.Files\Trend M
01685EB0	69 63 72 6F 5C 41 6E 74 69 53 70 79 77 61 72 65	icro\AntiSpyware
01685EC0	00 69 53 70 79 77 61 72 65 00 45 00 11 5F 68 01	.iSpyware.E. h
01685ED0	00 00 00 00 06 00 00 00 7E 8C 50 2C 18 8E 00 34	....~...~GP,  Z.4
01685EE0	C6 9F 89 E3 4D FE 00 73 5C 43 68 65 63 6B 50 6F	EYkãMp.s\CheckPo
01685EF0	69 6E 74 5C 5A 6F 6E 65 41 6C 61 72 6D 00 6F 6E	int\ZoneAlarm.on
01685F00	65 41 6C 61 72 6D 00 00 BD 7D DF 00 11 5E 68 01	eAlarm..h}S. h
01685F10	00 00 00 00 2F 00 00 00 25 53 59 53 5F 44 49 53	..../...%SYS_DIS
01685F20	4B 25 3A 5C 50 72 6F 67 72 61 6D 20 46 69 6C 65	K%:\Program File
01685F30	73 5C 41 76 69 72 61 5C 41 6E 74 69 56 69 72 20	s\Avira\AntiVir
01685F40	44 65 73 6B 74 6F 70 00 69 43 00 00 B0 59 68 01	Desktop.iC..°Yh

Рис. 12

### VNC\_HIDE\_DESKTOP.bin

Модуль, обеспечивающий скрытый удаленный доступ к компьютеру жертвы посредством протокола VNC и создания обратного соединения (back-connect).

Адрес для обратного соединения:

95.211.204.14:8098

Вместо заключения.

Данная статья создавалась с целью демонстрации, на примере актуальных атак, возможностей, тактик и техник заинтересованных лиц по получению скрытого доступа к компьютеру исследуемого объекта, а также для повышения уровня всеобщей осведомленности в вопросах информационной безопасности.

Рассмотренная в статье атака была зафиксирована в первой декаде октября, исходя из чего, мы допускаем, что актуальность приведенных индикаторов компрометации (IP-адреса, в частности) может быть под вопросом.

Следует отметить, что согласно данным Passive DNS (грубо говоря – истории соответствия доменных имен и IP-адресов) в первом квартале 2014 года IP-адрес 95.211.204.14 соответствовал доменному имени severodvinsk-bux.ru, которое принадлежит веб-сайту, специализирующемуся на рекламе. Это может означать ровным счетом ничего (так как прошло больше года, да и сайт сей меняет «айпишники» как перчатки), но, для полноты картины, мы добавили эту частичку информации.

### Отдел реагирования на инциденты CyS Centrum

Использованные материалы:

[1] <https://ru.wikipedia.org/wiki/Обфускация>