EXECryptor 2.x – Отключение проверки системной даты

17.05.2008, kioresk (RAZUM)

В этой небольшой статье рассматривается легкий способ отключения проверки изменения системной даты (clock manipulation), которая была добавлена во 2-й версии EXECryptor'а (криптора).

Теория

Проверка системной даты расположена внутри функции EC_GetDate, которая при вызове возвращает текущую дату. Кроме своего непосредственного вызова, EC_GetDate используется еще внутри следующих функций:

1. EC_GetTrialDaysLeft
2. EC_VerifySerialNumber
3. EC_VerifySerialNumberW

Конечно, можно найти начало функции EC_GetDate и пропатчить ее, чтобы она просто возвращала текущую дату. Но тогда для каждой исследуемой программы придется заморачиваться и искать где же она расположена. Поэтому нужен более простой и универсальный способ, и такой как раз есть.

В начале EC_GetDate вызывается функция, которая проверяет под какой операционной системой запущено приложение – Win9x или WinNT. Назовем ее Check_NT.

В зависимости от значения, возвращаемого функцией Check_NT, криптор выполняет различный код:

1. для WinNT – выполняет кучу кода (вызывая при этом GetCurrentProcessId, EnumWindows, GetWindowThreadProcessId, OpenProcess, NtQuerySystemInformation, GetTickCount, GetCurrentProcess, GetProcessTimes), проверяет не была ли изменена системная дата, и в конце концов возвращает текущую дату.
2. для Win9x – угадайте что? Правильно – просто возвращает текущую дату.

Это и есть тот самый универсальный способ – для отключения проверки системной даты надо просто пропатчить функцию Check_NT, чтобы она всегда сообщала, что мы работаем под Win9x.

Подробности

Для начала рассмотрим начало функции EC_GetDate, взятой из консольной части криптора версии 2.4.1.

```
0046DF04 EXECryptor_GetDate proc near
0046DF04
0046DF04             push    ebp
0046DF05             mov     ebp, esp
0046DF07             add     esp, 0FFFFFFA4h
0046DF0A             call    Check_NT
0046DF0A
0046DF0F             test    al, al
0046DF11             jz      Win9x
0046DF11
0046DF11             ; Тут идет код, выполняемый под WinNT (пропускаем его)

                     ...

0046E0C1 Win9x:
0046E0C1             call    RunInThread
0046E0C1
0046E0C6             test    al, al
0046E0C8             jz      short loc_46E0D4
```

```
0046E0C8
0046E0CA             lea     eax, [ebp-24h]
0046E0CD             push    eax                     ; lpSystemTimeAsFileTime
0046E0CE             call    GetSystemTimeAsFileTime
0046E0CE
0046E0D3             retn
0046E0D3
0046E0D4 ; ---------------------------------------------------------------------
0046E0D4
0046E0D4 loc_46E0D4:
0046E0D4             call    RunInThread
0046E0D4
0046E0D9             test    al, al
0046E0DB             jz      short loc_46E0FC
0046E0DB
0046E0DD             lea     eax, [ebp-24h]
0046E0E0             push    eax                     ; lpLocalFileTime
0046E0E1             lea     eax, [ebp-24h]
0046E0E4             push    eax                     ; lpFileTime
0046E0E5             call    FileTimeToLocalFileTime
0046E0E5
0046E0EA             lea     eax, [ebp-3Ch]
0046E0ED             push    eax                     ; lpFatTime
0046E0EE             lea     eax, [ebp-3Ah]
0046E0F1             push    eax                     ; lpFatDate
0046E0F2             lea     eax, [ebp-24h]
0046E0F5             push    eax                     ; lpFileTime
0046E0F6             call    FileTimeToDosDateTime
0046E0F6
0046E0FB             retn
0046E0FB
0046E0FC ; ---------------------------------------------------------------------
0046E0FC
0046E0FC loc_46E0FC:
0046E0FC             movzx   eax, word ptr [ebp-3Ah]
0046E100             mov     [ebp-1Ch], eax
0046E103             mov     eax, [ebp-1Ch]
0046E106             mov     esp, ebp
0046E108             pop     ebp
0046E109             retn
0046E109
0046E109 EXECryptor_GetDate endp
```

В приведенном коде несколько раз вызывается процедура RunInThread. Что она делает:

1. создает отдельный поток, который выполняет код, расположенный после вызова этой функции (т.е. код, который идет после test al, al – jz … и до первого retn)
2. по завершению потока управление передается коду, указанному в jz …

Все знают что криптор выполняет кучу кода в потоках, но зачем это ему? – все просто, чтобы не срабатывали железные бряки, поставленные на код (расположенный после вызова RunInThread) и используемые в нем API функции.

Хорошо, но есть же еще софтварные бряки, зачем такие заморочки с потоками? Да есть, и для борьбы с ними в крипторе есть:

1. бряки на код программы – проверка CRC памяти (об отключении которой говорилось ранее)
2. бряки на API функции – проверка кода функции на софтварные бряки перед ее вызовом (поскольку функции вызываются не напрямую, а через переходники)

Так, с бряками разобрались. Теперь вернемся к нашей EC_GetDate, код которой без потоков будет выглядеть следующим образом:

```
EXECryptor_GetDate proc near

             push    ebp
             mov     ebp, esp
             add     esp, 0FFFFFFA4h
             call    Check_NT

             test    al, al
```

```
            jz      Win9x

            ; Here comes part for WinNT

            ...

Win9x:
            lea     eax, [ebp-24h]
            push    eax                     ; lpSystemTimeAsFileTime
            call    GetSystemTimeAsFileTime

            lea     eax, [ebp-24h]
            push    eax                     ; lpLocalFileTime
            lea     eax, [ebp-24h]
            push    eax                     ; lpFileTime
            call    FileTimeToLocalFileTime

            lea     eax, [ebp-3Ch]
            push    eax                     ; lpFatTime
            lea     eax, [ebp-3Ah]
            push    eax                     ; lpFatDate
            lea     eax, [ebp-24h]
            push    eax                     ; lpFileTime
            call    FileTimeToDosDateTime

            movzx   eax, word ptr [ebp-3Ah]
            mov     [ebp-1Ch], eax
            mov     eax, [ebp-1Ch]
            mov     esp, ebp
            pop     ebp
            retn

EXECryptor_GetDate endp
```

Как видно из приведенного кода, функция Check_NT должна возвращать 0 в регистре AL, чтобы выполнялся код, предназначенный для Win9x.

Посмотрим код этой функции.

```
0046B6D0 Check_NT proc near
0046B6D0
0046B6D0             push    ebp
0046B6D1             mov     ebp, esp
0046B6D3             push    ecx
0046B6D4             mov     byte ptr [ebp-1], 0
0046B6D8             mov     eax, cs                     ; проверяем работаем ли мы под WinNT
0046B6DA             xor     al, al
0046B6DC             or      eax, eax
0046B6DE             jnz     short Exit
0046B6DE
0046B6E0             call    Get_Ntdll_ImageBase         ; GetModuleHandle("ntdll.dll")
0046B6E0                                                 ; взводим флаг, что адрес был получен
0046B6E0                                                 ; сохраняем адрес в просторах криптора и
0046B6E0                                                 ; возвращаем его в EAX
0046B6E0
0046B6E5             test    eax, eax
0046B6E7             setnz   byte ptr [ebp-1]
0046B6E7
0046B6EB Exit:
0046B6EB             mov     al, byte ptr [ebp-1]
0046B6EE             pop     ecx
0046B6EF             pop     ebp
0046B6F0             retn
0046B6F0
0046B6F0 Check_NT endp
```

Самый простой способ всегда возвращать 0 – это заменить инструкцию setnz byte ptr [ebp-1] (0F 95 45 FF) на mov byte ptr [ebp-1], 0 (C6 45 FF 00), поскольку крипторовская виртуальная машина не умеет обфуцировать/виртуализовать инструкции setcc и в любой программе эту инструкцию можно найти без труда.

Заключение

Итак, для отключения проверки системной даты нужно:

1. найти инструкцию setnz byte ptr [ebp-1] (0F 95 45 FF)
2. потрейсить/проверить соседний код, чтобы он был похож на код функции Check_NT (поскольку setnz может много где использоваться)
3. заменить ее инструкцией mov byte ptr [ebp-1], 0 (C6 45 FF 00)

Этот метод работает на всех крипторах, начиная с версии 2.2.