

02.01.2024

Курс:

Практическая работа к уроку № Lesson_6

--

Способы обхода активации программ

Задание:

Дана программа task-4. Необходимо получить корректный ключ для вашего имени.

Пример запуска программы:

crackme-4.exe

Name: <your name>

License Key: <license key>

Способы обхода активации программ

Для достижения этой цели может быть использованы различные подходы и способы.

Далее будут рассмотрены следующие способы:

- Продление триального периода
- Создание ключа
- Патчинг
- Перенаправление сетевых запросов

Запускаем в CMD *task-4.exe*

Insert name: Andrew

License key: qwerty

Запускаем в OllyDbg *task-4.exe*

OllyDbg - task-4.exe - [Memory map]

File View Debug Plugins Options Window Help

Address	Size	Owner	Section	Contains	Type	Access	Initial	Mapped as
00010000	00010000				Map 00041004	RW	RW	
00040000	00010000				Map 00041002	R	R	
000F0000	00005000				Priv 00021104	RW Guarded	RW	
00190000	00002000				Priv 00021104	RW Guarded	RW	
00190000	00003000			stack of main thread	Priv 00021104	RW Guarded	RW	
001A0000	00004000				Map 00041002	R	R	
001B0000	00002000				Priv 00021004	RW	RW	
001F0000	00006000				Priv 00021104	RW Guarded	RW	
002CE000	00004000				Priv 00021004	RW	RW	
00400000	00001000			data block of main thread	Priv 00021004	RW	RW	
00401000	00001000	task-4	.text	code	Imag 01001002	R	RWE	
00410000	00000000	task-4	.idata	imports	Imag 01001002	R	RWE	
0041E000	00002000	task-4	.data	data	Imag 01001002	R	RWE	
00430000	00005000				Priv 00021004	RW	RW	
004C0000	0000C000				Map 00041002	R	R	\\Device\\HarddiskVolume2\\Windows\\System32\\locale.n...
005C0000	00005000				Priv 00021004	RW Guarded	RW	
007B0000	00003000				Priv 00021104	RW	RWE	
54300000	00001000	KERNEL32		PE header	Imag 01001002	R	RWE	
75490000	00064000	KERNEL32	.text	code, exports	Imag 01001020	R E	RWE	
75500000	0002F000	KERNEL32	.rdata	imports, exports	Imag 01001002	R	RWE	
75530000	00001000	KERNEL32	.data	data	Imag 01001004	RW	RWE	
75540000	00001000	KERNEL32	.rsro	resources	Imag 01001002	R	RWE	
75550000	00005000	KERNEL32	.reloc	relocations	Imag 01001002	R	RWE	
76FC0000	00001000	KERNELBA		PE header	Imag 01001002	R	RWE	
76FC13000	00001000	KERNELBA	.text	code, exports	Imag 01001002	R	RWE	
77134000	00004000	KERNELBA	.data	data	Imag 01001002	R	RWE	
77180000	00006000	KERNELBA	.idata	imports	Imag 01001002	R	RWE	
7718E000	00001000	KERNELBA	.didat		Imag 01001002	R	RWE	
77190000	00001000	KERNELBA	.rsro	resources	Imag 01001002	R	RWE	
77190000	00002400	KERNELBA	.reloc	relocations	Imag 01001002	R	RWE	
77A90000	00009000				Imag 01001002	R	RWE	
77A90000	00001000	ntdll		PE header	Imag 01001002	R	RWE	
77A91C000	00011C000	ntdll	.text	code, exports	Imag 01001002	R	RWE	
77BBD000	00001000	ntdll	RT		Imag 01001002	R	RWE	
77BBD000	00006000	ntdll	.data	data	Imag 01001002	R	RWE	
77BC4000	00003000	ntdll	.rdata		Imag 01001002	R	RWE	
77BC7000	00001000	ntdll	.000c0g		Imag 01001002	R	RWE	
77BC8000	0000F000	ntdll	.rsro	resources	Imag 01001002	R	RWE	
77C37000	00005000	ntdll	.reloc	relocations	Imag 01001002	R	RWE	
7FE50000	00005000				Map 00041002	R	R	
7FF50000	00001000				Priv 00021004	RW	RW	
7FF61000	00001000				Priv 00021004	RW	RW	
7FF80000	00001000							

- Переходим по этому адресу (С)

00401000

[illegible]

- Точка останова

OllyDbg - task-4.exe - [CPU - main thread, module task-4]

File View Debug Plugins Options Window Help

LEMTW H C / K B R ... S

Registers (FPU)

EAX 0019FFCC task-4.<ModuleEntryPoint>
ECX 004014E1 task-4.<ModuleEntryPoint>
EDX 004014E1 task-4.<ModuleEntryPoint>
EBX 003C0000
ESP 0019FF74
EBP 0019FF80
ESI 004014E1 task-4.<ModuleEntryPoint>
EDI 004014E1 task-4.<ModuleEntryPoint>
EIP 004014E1 task-4.<ModuleEntryPoint>

C 0 ES 002B 32bit 0 (FFFFFFFF)
P 1 CS 002B 32bit 0 (FFFFFFFF)
A 0 SS 002B 32bit 0 (FFFFFFFF)
Z 1 DS 002B 32bit 0 (FFFFFFFF)
S 0 FS 0053 32bit 30F000 (FFF)
T 0 GS 002B 32bit 0 (FFFFFFFF)
D 0
0 0 LastErr ERROR_SEM_NOT_FOUND (0000006B)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0

FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

Run application Main Debug|Run F9

OllyDbg - task-4.exe - [CPU - main thread, module task-4]

File View Debug Plugins Options Window Help

LEMTW H C / K B R ... S

Registers (FPU)

EAX 0041EE00 task-4.0041EE00
ECX 00000000
EDX 5C0D2CC4
EBX 003C0000
ESP 0019FF28
ESI 005131E8
EDI 0051B5B0
EIP 00401051 task-4.00401051

C 0 ES 002B 32bit 0 (FFFFFFFF)
P 1 CS 002B 32bit 0 (FFFFFFFF)
A 0 SS 002B 32bit 0 (FFFFFFFF)
Z 0 DS 002B 32bit 0 (FFFFFFFF)
S 0 FS 0053 32bit 30F000 (FFF)
T 0 GS 002B 32bit 0 (FFFFFFFF)
D 0
0 0 LastErr ERROR_SUCCESS (00000000)
EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0

FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

Try:
Incorrect:
AABBCCDDEEFF
abcdefghijklmnopqrstuvwxyz



Пример **prog_4.1.exe**:

1. Продление триального периода:

Устанавливаем **procmon**

<https://learn.microsoft.com/en-us/sysinternals/downloads/procmon>

```
C:\Users\Администратор\Documents\GB>prog-4.1.exe 2 1

Trial Period: 9 of 10

Result: 3
```

The screenshot shows a Windows Command Prompt window titled "Администратор: Командная строка" and a Process Monitor window titled "Process Monitor - Sysinternals: www.sysinternals.com".

Command Prompt:

```
C:\Users\Администратор\Documents\GB>prog-4.1.exe

Usage:

prog-4.1.exe <number> <number>

Activation:

prog-4.1.exe <license key>

C:\Users\Администратор\Documents\GB>

Trial Period: 9 of 10

Result: 3

C:\Users\Администратор\Documents\GB>

Trial Period: 8 of 10

Result: 5

C:\Users\Администратор\Documents\GB>

Trial Period: 7 of 10

Result: 5

C:\Users\Администратор\Documents\GB>
```

Process Monitor:

Time of Day	Process Name	PID	Operation	Path
17:15:38,8086413	prog-4.1.exe	4712	RegOpenKey	HKLM\SOFTWARE\Microsoft\Win
17:15:38,8088563	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControlSet\Q
17:15:38,8089038	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControlSet\Q
17:15:38,8089361	prog-4.1.exe	4712	RegSetInfoKey	HKLM\System\CurrentControlSet\Q
17:15:38,8089532	prog-4.1.exe	4712	RegQueryValue	HKLM\System\CurrentControlSet\Q
17:15:38,8091113	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControlSet\Q
17:15:38,8091326	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControlSet\Q
17:15:38,8091568	prog-4.1.exe	4712	RegSetInfoKey	HKLM\System\CurrentControlSet\Q
17:15:38,8091970	prog-4.1.exe	4712	RegQueryValue	HKLM\System\CurrentControlSet\Q
17:15:38,8092722	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControlSet\Q
17:15:38,8092924	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControlSet\Q
17:15:38,8093153	prog-4.1.exe	4712	RegSetInfoKey	HKLM\System\CurrentControlSet\Q
17:15:38,8093310	prog-4.1.exe	4712	RegQueryValue	HKLM\System\CurrentControlSet\Q
17:15:38,8093539	prog-4.1.exe	4712	RegQueryValue	HKLM\System\CurrentControlSet\Q
17:15:38,8093750	prog-4.1.exe	4712	RegCloseKey	HKLM\System\CurrentControlSet\Q
17:15:38,8094116	prog-4.1.exe	4712	RegCloseKey	HKLM\System\CurrentControlSet\Q
17:15:38,8094584	prog-4.1.exe	4712	RegOpenKey	HKLM\SYSTEM\CurrentControlSet
17:15:38,8094887	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControlSet\Q
17:15:38,8096078	prog-4.1.exe	4712	RegQueryValue	HKLM\System\CurrentControlSet\Q
17:15:38,8096736	prog-4.1.exe	4712	QueryNameInfo...	C:\Windows\SysWOW64\sspicli.d
17:15:38,8098371	prog-4.1.exe	4712	RegQueryValue	HKLM\System\CurrentControlSet\Q
17:15:38,8099256	prog-4.1.exe	4712	QueryNameInfo...	C:\Windows\SysWOW64\sechost
17:15:38,8100182	prog-4.1.exe	4712	RegOpenKey	HKLM\SYSTEM\CurrentControlSe
17:15:38,8100500	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControlSet\

Showing 48 897 of 175 611 events (27%)

Add Filter -> Process Name: prog-4.1.exe

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time of Day	Process Name	PID	Operation	Path
17:15:38,7698512	prog-4.1.exe	4712	Process Start	
17:15:38,7698594	prog-4.1.exe	4712	Thread Create	
17:15:38,7718202	prog-4.1.exe	4712	Load Image	C:\Users\Администратор\...
17:15:38,7718963	prog-4.1.exe	4712	Load Image	C:\Windows\System32\ntldr...
17:15:38,7719644	prog-4.1.exe	4712	Load Image	C:\Windows\SysWOW64\ntldr...
17:15:38,7721786	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControl...
17:15:38,7722044	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControl...
17:15:38,7722306	prog-4.1.exe	4712	RegQueryValue	HKLM\System\CurrentControl...
17:15:38,7722510	prog-4.1.exe	4712	RegCloseKey	HKLM\System\CurrentControl...
17:15:38,7722762	prog-4.1.exe	4712	RegOpenKey	HKLM\SYSTEM\CurrentControl...
17:15:38,7722938	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControl...
17:15:38,7723550	prog-4.1.exe	4712	RegOpenKey	HKLM\SYSTEM\CurrentControl...
17:15:38,7723724	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControl...
17:15:38,7723908	prog-4.1.exe	4712	RegQueryValue	HKLM\System\CurrentControl...
17:15:38,7724078	prog-4.1.exe	4712	RegCloseKey	HKLM\System\CurrentControl...
17:15:38,7729675	prog-4.1.exe	4712	CreateFile	C:\Windows\...
17:15:38,7732218	prog-4.1.exe	4712	Load Image	C:\Windows\System32\wow64...
17:15:38,7734156	prog-4.1.exe	4712	Load Image	C:\Windows\System32\wow64...
17:15:38,7747845	prog-4.1.exe	4712	CreateFile	C:\Windows\System32\wow64...
17:15:38,7756180	prog-4.1.exe	4712	CreateFile	C:\Windows\...
17:15:38,7756689	prog-4.1.exe	4712	QueryNameInfo...	C:\Windows\...
17:15:38,7757072	prog-4.1.exe	4712	CloseFile	C:\Windows\...
17:15:38,7776479	prog-4.1.exe	4712	RegOpenKey	HKLM\Software\Microsoft\...
17:15:38,7776955	prog-4.1.exe	4712	RegQueryValue	HKLM\SOFTWARE\Microsoft\...
17:15:38,7777161	prog-4.1.exe	4712	RegQueryValue	HKLM\SOFTWARE\Microsoft\...
17:15:38,7777442	prog-4.1.exe	4712	RegCloseKey	HKLM\SOFTWARE\Microsoft\...
17:15:38,7779194	prog-4.1.exe	4712	Load Image	C:\Windows\System32\wow64...
17:15:38,7784033	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControl...
17:15:38,7784362	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControl...
17:15:38,7784675	prog-4.1.exe	4712	RegSetInfoKey	HKLM\System\CurrentControl...
17:15:38,7784895	prog-4.1.exe	4712	RegQueryValue	HKLM\System\CurrentControl...
17:15:38,7785149	prog-4.1.exe	4712	RegCloseKey	HKLM\System\CurrentControlSet\Contr...
17:15:38,7787490	prog-4.1.exe	4712	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Con...
17:15:38,7787762	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControlSet\Contr...
17:15:38,7788556	prog-4.1.exe	4712	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Con...
17:15:38,7788760	prog-4.1.exe	4712	RegOpenKey	HKLM\System\CurrentControlSet\Contr...
17:15:38,7789003	prog-4.1.exe	4712	RegSetInfoKey	HKLM\System\CurrentControlSet\Contr...
17:15:38,7789167	prog-4.1.exe	4712	RegQueryValue	HKLM\System\CurrentControlSet\Contr...
17:15:38,7789394	prog-4.1.exe	4712	RegCloseKey	HKLM\System\CurrentControlSet\Contr...
17:15:38,7800108	prog-4.1.exe	4712	CreateFile	C:\Users\Администратор\Documents\...

Администратор: Командная строка

Trial Period: 2 of 10

Result: 5

C:\Users\Администратор\Documents\GB>prog-4.1.exe 3 2

Trial Period: 1 of 10

Result: 5

C:\Users\Администратор\Documents\GB>prog-4.1.exe 3 2

Trial Period: 0 of 10

Result: 5

C:\Users\Администратор\Documents\GB>prog-4.1.exe 3 2

The number of starts has expired.

C:\Users\Администратор\Documents\GB>prog-4.1.exe 3 2

The number of starts has expired.

C:\Users\Администратор\Documents\GB>prog-4.1.exe 3 2

The number of starts has expired.

C:\Users\Администратор\Documents\GB>

HKLM\System\CurrentControlSet\Contr... SUCCESS

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... SUCCESS

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\...

HKLM\SYSTEM\CurrentControlSet\Contr... REPARSE

HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... SUCCESS

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... SUCCESS

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... SUCCESS

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... SUCCESS

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... SUCCESS

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND

HKLM\SYSTEM\CurrentControlSet\Con... REPARSE

HKLM\System\CurrentControlSet\Contr... SUCCESS

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

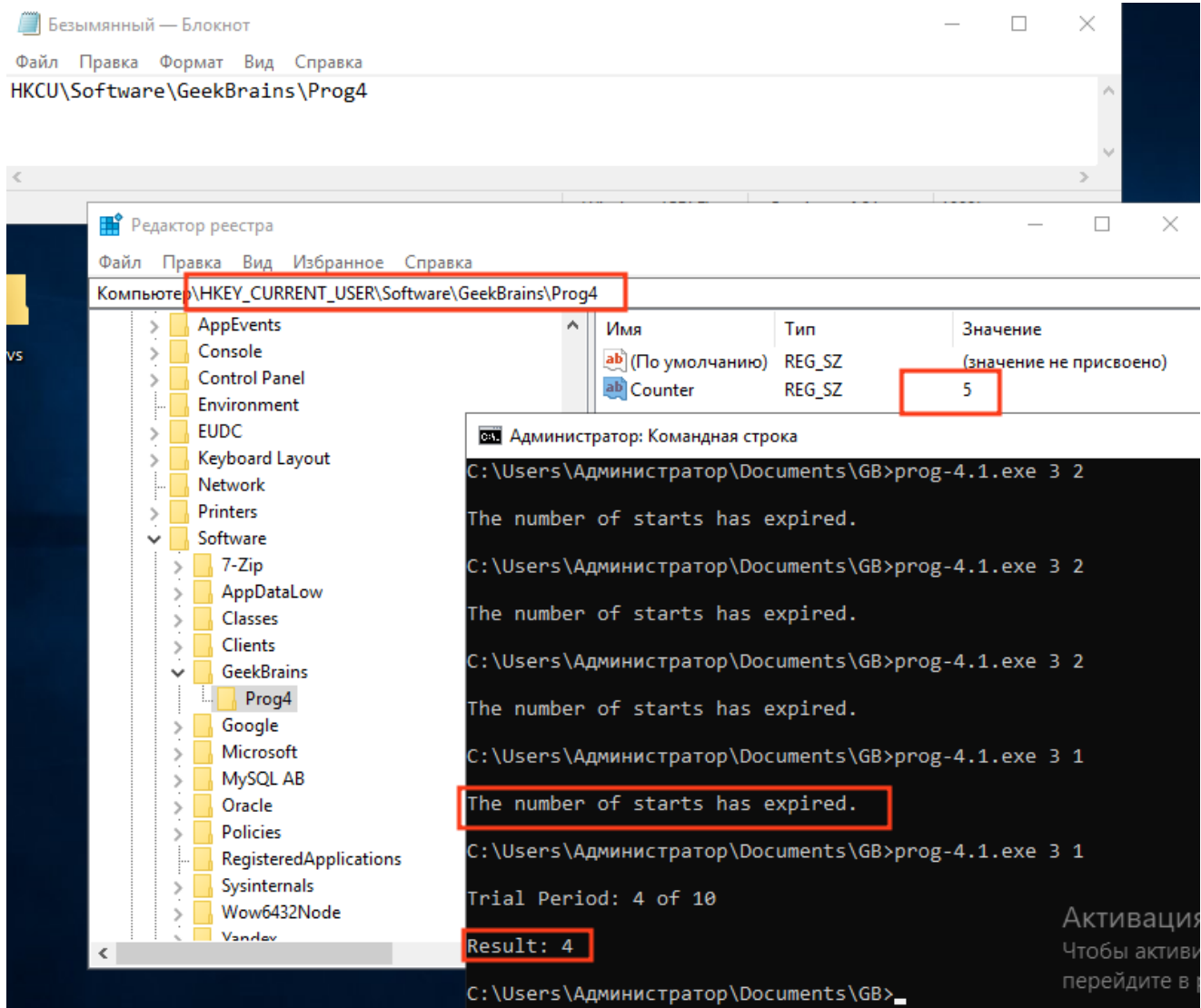
Time of Day	Process Name	PID	Operation	Path
17:22:18,6232206	prog-4.1.exe	2868	QueryValue	C:\Users\Администратор\Documents\GB>prog-4.1.exe 3 2
17:22:18,6233583	prog-4.1.exe	2868	RegOpenKey	
17:22:18,6233858	prog-4.1.exe	2868	RegOpenKey	Trial Period: 1 of 10
17:22:18,6234138	prog-4.1.exe	2868	RegSetInfoKey	
17:22:18,6234309	prog-4.1.exe	2868	RegQueryValue	Result: 5
17:22:18,6234534	prog-4.1.exe	2868	RegCloseKey	
17:22:18,6236798	prog-4.1.exe	2868	RegQueryValue	C:\Users\Администратор\Documents\GB>prog-4.1.exe 3 2
17:22:18,6237407	prog-4.1.exe	2868	QueryValue	
17:22:18,6240980	prog-4.1.exe	2868	QueryValue	Trial Period: 0 of 10
17:22:18,6241470	prog-4.1.exe	2868	RegOpenKey	
17:22:18,6241899	prog-4.1.exe	2868	RegQueryValue	
17:22:18,6242058	prog-4.1.exe	2868	RegQueryValue	Result: 5
17:22:18,6242330	prog-4.1.exe	2868	RegOpenKey	
17:22:18,6242581	prog-4.1.exe	2868	RegSetInfoKey	C:\Users\Администратор\Documents\GB>prog-4.1.exe 3 2
17:22:18,6243453	prog-4.1.exe	2868	RegQueryValue	
17:22:18,6243731	prog-4.1.exe	2868	RegQueryValue	The number of starts has expired.
17:22:18,6244065	prog-4.1.exe	2868	RegCloseKey	
17:22:18,6244345	prog-4.1.exe	2868	RegQueryValue	C:\Users\Администратор\Documents\GB>prog-4.1.exe 3 2
17:22:18,6244496	prog-4.1.exe	2868	RegQueryValue	
17:22:18,6244764	prog-4.1.exe	2868	RegOpenKey	The number of starts has expired.
17:22:18,6245001	prog-4.1.exe	2868	RegSetInfoKey	
17:22:18,6245185	prog-4.1.exe	2868	RegQueryValue	C:\Users\Администратор\Documents\GB>prog-4.1.exe 3 2
17:22:18,6245441	prog-4.1.exe	2868	RegQueryValue	
17:22:18,6245781	prog-4.1.exe	2868	RegCloseKey	The number of starts has expired.
17:22:18,6266987	prog-4.1.exe	2868	Thread	
17:22:18,6271731	prog-4.1.exe	2868	Load Image	C:\Users\Администратор\Documents\GB>prog-4.1.exe 3 1
17:22:18,6276907	prog-4.1.exe	2868	Thread	
17:22:18,6279613	prog-4.1.exe	2868	Thread	The number of starts has expired.
17:22:18,6311005	prog-4.1.exe	2868	Thread	
17:22:18,6312305	prog-4.1.exe	2868	Process	
17:22:18,6312733	prog-4.1.exe	2868	RegOpenKey	C:\Users\Администратор\Documents\GB>
17:22:18,6312952	prog-4.1.exe	2868	RegQueryValue	HKLM\System\CurrentControlSet\Servi... SUCCESS Type: REG_BINA...
17:22:18,6313188	prog-4.1.exe	2868	RegSetValue	HKLM\System\CurrentControlSet\Servi... SUCCESS Type: REG_BINA...
17:22:18,6313803	prog-4.1.exe	2868	RegCloseKey	HKLM\System\CurrentControlSet\Servi... SUCCESS
17:22:18,6314673	prog-4.1.exe	2868	CloseFile	C:\Windows\...
17:22:18,6316023	prog-4.1.exe	2868	CloseFile	C:\Users\Администратор\Documents\...
17:22:18,6383784	prog-4.1.exe	2868	RegCloseKey	HKLM\SOFTWARE\Microsoft\Window... SUCCESS
17:22:18,6384169	prog-4.1.exe	2868	RegCloseKey	HKLM\System\CurrentControlSet\Contr... SUCCESS
17:22:18,6384348	prog-4.1.exe	2868	RegCloseKey	HKCU\...

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

Showing 1 638 of 297 442 events (0.5%)

Backed by virtual memory



2. Создание ключа:

Запускаем в CMD **prog-4.1.exe**

```
C:\Users\Администратор\Documents\GB>prog-4.1.exe

Usage:
    prog-4.1.exe <number> <number>

Activation:
    prog-4.1.exe <license key>

C:\Users\Администратор\Documents\GB>prog-4.1.exe 123123
License key is incorrect.
```


Мы остановились перед вызовом функции. Видим, что пока в командной строке не было записей.

0040151F	> 75 14	JNZ SHORT prog-4_1.00401535	
00401521	. 68 00904100	PUSH prog-4_1.00419000	[Arg1 = 00419000 ASCII "License key is incorrect."]
00401526	. E8 65010000	CALL prog-4_1.00401690	printf
0040152B	. 83C4 04	ADD ESP,4	
0040152E	. 33C0	XOR EAX,EAX	
00401530	> E9 0F010000	JMP prog-4_1.00401644	
00401535	> E8 86FCFFFF	CALL prog-4_1.004011C0	

Обратимся к коду, который идет перед тем, как будет напечатана строка "License key is incorrect.", видим вызов функции 0x00401050 и условный переход. Видим, что переход

JNZ указывает на код, где должна быть напечатана нужная нам строка.

004014FD	> 837D 08 02	CMP DWORD PTR SS:[EBP+8],2	
00401501	> 75 61	JNZ SHORT prog-4_1.00401564	
00401503	. B9 04000000	MOV ECX,4	
00401508	. C1E1 00	SHL ECX,0	Shift constant out of range 1..31
0040150B	. 8B55 0C	MOV EDI,DWORD PTR SS:[EBP+C]	
0040150E	. 8B040A	MOV EAX,DWORD PTR DS:[EDX+ECX]	
00401511	. 50	PUSH EAX	
00401512	. E8 39FBFFFF	CALL prog-4_1.00401050	[Arg1 = prog-4_1.00401050]
00401517	. 83C4 04	ADD ESP,4	
0040151A	. 0FB6C8	MOVZX ECX,AL	
0040151D	. 85C9	TEST ECX,ECX	
0040151F	> 75 14	JNZ SHORT prog-4_1.00401535	
00401521	. 68 00904100	PUSH prog-4_1.00419000	[Arg1 = 00419000 ASCII "License key is incorrect."]
00401526	. E8 65010000	CALL prog-4_1.00401690	printf
0040152B	. 83C4 04	ADD ESP,4	

Давайте разберемся, от чего зависит результат условия. После вызова функции происходит запись значения из регистра AL в регистр ECX. Далее мы видим оператор TEST,

который выполняет проверка на нулевое значение в регистре ECX. И прыжок будет, если

результат будет не 0. Мы выяснили, что решение о корректности ключа принимает функция 0x00401050. Давайте посмотрим на содержимое этой функции. Кликаем правой кнопкой на инструкции CALL и выбираем Follow.

The screenshot shows a debugger window with three main panes. The top pane displays assembly code with addresses, hex values, and mnemonics. The middle pane shows the state of registers (EAX, ECX, EDI, ESP, EBP, ESI, EDI, EIP) and their values. The bottom pane shows the ASCII representation of the data being processed, including the string "License key is incorrect...".

Address	Hex dump	ASCII
00419000	4C 69 63 65 6E 73 65 74 2E 0A 00 00 00 00 00 00	License key is incorrect...The program has been activated....
00419010	69 6E 63 6F 72 72 61 60 20 68 65 79 20 69 73 20	Unknown error (license)....The number of start s has expired...Unknown error (trial).....Res
00419020	20 70 72 6F 72 61 60 20 68 65 79 20 69 73 20	
00419030	6E 20 61 63 74 69 76 61 74 65 64 21 0A 00 00 00	
00419040	0A 55 6E 68 6E 6F 77 6E 20 65 72 72 6F 72 21 20	
00419050	28 6C 63 65 65 73 65 20 6A 00 00 0A 54 65 65	
00419060	20 6E 75 6D 00 72 20 6F 6E 20 73 74 61 72 74	
00419070	73 20 68 61 73 20 65 78 70 69 72 65 64 2E 0A 00	
00419080	0A 55 6E 68 6E 6F 77 6E 20 65 72 72 6F 72 21 20	
00419090	28 74 72 69 61 6C 29 0A 00 00 00 0A 52 65 73	

Поставим точку останова перед инструкцией MOV и перезапустим программу. Выполняем инструкцию MOV и видим, что адрес в регистре EAX указывает на ключ, который мы передали при запуске программы.

Вызываем эту функцию и видим, что в регистре EAX появилось значение 6. Это значение

похоже на длину нашей строки. Давайте изменим длину ключа (blabla1) и перезапустим

программу.

Видим, что теперь в регистре EAX значение 7, значит функция с адресом 0x004036E0 это

функция strlen. Добавим комментарий.

Далее мы видим, что длина сравнивается со значением 0x13 (19). Мы эту проверку не проходим и попадаем в конец функции и функция завершается со значением 0 в регистре EAX. А мы помним, что нам необходимо добиться того, чтобы функция вернула ненулевое значение.

Ставим ключ с длиной в 19 символов и перезапускаем программу. Мы прошли первую проверку, теперь мы знаем, что ключ должен состоять из 19 символов.

Проверяем ключ в командной строке.

```
prog-4.1.exe 1234567890123456789
License key is incorrect.
```

Ключ всё ещё неправильный. Продолжаем анализ функции.

Далее в регистр ECX помещается значение 1, затем за счет сдвига битов влево на 2 позиции мы получаем значение 4 в регистре ECX. Затем в регистр EDX попадает указатель на ключ. Далее в регистр EAX попадает значение, который находится в строке на 4-й позиции. Это байт 35, который по ASCII таблице равен 5.

И происходит сравнение с байтом 0x2D. И мы попадаем в конец функции с нулевым значением в регистре EAX. Делаем выводы, что 4-й байт должен быть 0x2D. По таблице ASCII это знак “-”.

Изменяем ключ (1234-67890123456789) и перезапускаем программу.

Мы прошли проверку и далее видим еще несколько таких же проверок. На 9-й и 14-й байты. Проверяем ключ в командной строке.

```
prog-4.1.exe 1234-6789-1234-6789
License key is incorrect.
```

Ключ всё ещё неправильный. Продолжаем анализ функции.

Изменяем ключ (1234-6789-1234-6789) и перезапускаем программу.

Далее мы видим набор инструкций, который очень похож на цикл с предусловием.

Выставляет локальная переменная в 0. Затем сравнивается с 0x14 (20). Это длина нашего ключа.

Далее мы видим условный переход JGE, который прыгает на инструкцию MOV, которая кладет значение 1 в регистр AL и функция завершает свою работу. Это то, что нам нужно.

Получается, нам нужно оставаться в цикле пока счетчик не достигнет 20.

Итак, мы попали на первую итерацию цикла. В регистр ECX мы кладем указатель на ключ.

Затем прибавляем счетчик. Видим, это смещение по строке. Первый байт ключа кладем в

регистр EAX. Прибавляем счетчик. Затем в регистр ECX кладем второй байт ключа. В регистре EDX храним сумму первых двух байтов. В регистр ECX кладем третий байт и снова прибавляем его к значению в регистре EDX. Затем тоже самое с третьим байтом. Далее мы кладем четвертый байт в регистр ECX. И теперь с помощью инструкции LEA мы складываем значение в регистре EDX и ECX и вычитаем число 0xC0. В результате получаем число 0x0A (10) в регистре EDX и выполняем сравнение с 0x0A. Они равны и происходит переход к началу цикла. Получается, мы посчитали сумму первых 4 байт и вычли из суммы 192.

Переходим ко второй фазе цикла. Мы увеличиваем счетчик на 5 и на этот раз складываем

следующие 4 числа в ключе. В сумме вы получили 1E и вышли из цикла со значением 0 в регистре AL.

Получается, что цикл с каждым проходом считает сумму цифр в каждом блоке, вычитает

192 и сравнивает с 10. Первый блок прошел проверку, а второй уже нет. Нам ничего не мешает сделать второй блок и последующие таким же, как первый.

Проверяем ключ в командной строке.

```
prog-4.1.exe 1234-1234-1234-1234  
The program has been activated!
```

ASCII код цифры 1 - это 0x31 (49), цифры 2 - это 0x32 (50) и т.д. Для того, чтобы получить

число, нужно вычесть 48 из каждого байта. Например, 49 - 48 - это 1, 50 - 48 - это 2 и т.д. Если мы складываем 4 числа в каждом блоке, то в конце нужно вычесть 4 раза по

48. А это и есть 192, которые мы видели в коде Давайте составим другой ключ, но с той же суммой в блоке.

```
prog-4.1.exe 0334-1234-1234-1234  
The program has been activated!
```

- Сумма цифр в каждом блоке должна быть равно 10.

2. Перенаправление сетевых запросов:

Запустить программу prog-4.4.1, убедиться в том, что она работает правильно.

<https://1.eu.dl.wireshark.org/win64/all-versions/>

Запускаем сниффер. Выбираем сетевой интерфейс, через который мы выходим в сеть Интернет. Видим, что через сетевой интерфейс проходит очень много сетевых пакетов. Сейчас хорошо бы добавить фильтр, но мы пока ничего не знаем о протоколе взаимодействия. TCP это или UDP. Используется ли доменное имя или обращение происходит сразу по IP-адресу. Если программа взаимодействует с удаленным сервером в сети Интернет, то у нее точно в коде или каком-нибудь конфиге должен быть указан IP-адрес или доменное имя удаленного сервера, иначе она не будет знать, как с ним связаться. В нашем случае программа prog-4.4.1.exe состоит из одного файла. Давайте с помощью утилиты Strings получим все строки, которые есть в бинарном файле и попробуем найти что-нибудь похожее на IP-адрес или доменное имя.

Рекомендации по установке утилиты Strings.

Ссылка: <https://docs.microsoft.com/en-us/sysinternals/downloads/strings>

Запускаем утилиту Strings.

```
strings64.exe prog-4.4.1.exe > out.txt
```

Открываем файл out.txt и видим более 2 тыс. строк. Это очень много для того, чтобы искать в ручном режиме. Давайте сократим область поиска.

Если доменное имя есть в составе программы, то оно находится в секции данных.

Запускаем утилиту Pupy. Открываем файл prog-4.4.1.exe. Выбираем Section Headers. Видим, что секция данных в файле начинается со смещения 0x17a00.

Утилита Strings принимает смещение только в десятичной системе счисления, поэтому открываем калькулятор и переводим число из шестнадцатичной в десятичную.

Получаем число 96768. Запускаем снова утилиту Strings.

```
strings64.exe -f 96768 prog-4.4.1.exe > out.txt
```

Открываем файл out.txt и видим всего 42 строки. И почти сразу находим доменное имя check-key.example.com. Это говорит о том, что программа, скорее всего, обращается к серверу по доменному имени. Значит перед тем, как отправить запрос на проверку ключа, происходит разрешение доменного имени для получения IP-адреса. А для

разрешения доменных имен используется протокол DNS.

Возвращаемся к снифферу и добавляем фильтр по протоколу DNS.

dns:

Давайте еще раз попробуем активировать нашу программу и убедимся в том, что действительно доменное имя check-key.example.com используется программой при активации.

prog-4.4.1.exe blabla

В сетевом трафике мы быстро находим DNS-запрос. В ответе мы видим, что этому доменному имени соответствует IP-адрес 192.168.1.8. Давайте посмотрим, какие сетевые пакеты были отправлены по этому IP-адресу.

dns || ip.dst == 192.168.1.5

Мы видим, что было установлено TCP-соединение с сервером и в одном из пакетов отправлен ключ в открытом виде - "blabla". Обратите внимание, что порт назначения 1337, это нам понадобится позже.

В ответ была отправлена строка "false" и соединение завершилось. Мы видим, что никаких механизмов контроля целостности сообщений не используется. Мы знаем, что в случае неправильного ключа сервер отвечает строкой "false", но при этом не знаем, что он отвечает, когда ключ правильный.

Давайте откроем программу prog-4.4.1.exe под отладчиком и найдем код, который обрабатывает ответы от сервера.

Переходим в секцию кода и пробуем найти код, который ссылается на строку "License key

is incorrect". Поставим точку останова перед условным переходом и перезапустим программу в режиме активации.

Функция 0x00401050 вернула нулевое значение через регистр EAX. Давайте перейдем внутрь этой функции, поставим точку останова и перезапустим программу.

Видим, что внутри функции происходит отправка сетевых пакетов, далее мы получаем ответ от сервера и выполняем сравнение двух строк с помощью функции strcmp.

Сравнение ответа происходит со строкой "true". А если мы указываем при активации программы ключ "blabla", то мы получаем от сервера строку "false".

Наша задача сделать так, чтобы сервер прислал строку "true".

Давайте выполним перенаправление запроса на наш сервер.

Специально для этой цели была разработана программа prog-4.4.2, которая умеет отвечать на все запросы строкой "true".

Запускаем наш сервер:

prog-4.4.2.exe

Итак, сейчас сервер работает локально на нашем компьютере, теперь нам надо перенаправить запрос на наш сервер. Это можно очень легко сделать, если добавить следующую запись в файл `C:\Windows\System32\hosts`

```
127.0.0.1 check-key.example.com
```

Проверим, что теперь доменное имя `check-key.example.com` разрешается IP-адресом `127.0.0.1`.

Открываем командную строку

```
ping check-key.example.com
```

Видим, что теперь доменному имени соответствует IP-адрес `127.0.0.1`.

Давайте попробуем активировать программу.

```
prog-4.4.1.exe blabla
```

The program has been activated

Откроем консоль, где был запущен сервер. Видим, что он принял запрос и ответил строкой `"true"`.

Открываем сниффер и видим, что новых запросов на настоящий сервер не выполнялось.

4. Патчинг

Откроем программу `prog-4.1` под отладчиком и перейдем к функции, где программа печатает строку `"License key is incorrect."`. Зададим в качестве аргумента произвольный ключ (`blabla`).

Поставим точку останова сразу после вызова функции проверки ключа.

Мы видим, что далее проверяется значение, которое вернула функция. Значение кладется в регистр `ECX`, происходит проверка на ноль и если не ноль, то проверка успешно пройдена.

Какие есть варианты патчинга?

1. Сделать так, чтобы функция возвращала всегда 1.
2. Сделать так, чтобы в регистр `ECX` всегда попадала 1 перед проверкой.
3. Изменить условный переход `JNZ` на безусловный `JMP`.

Все эти варианты должны работать. Но самый простой вариант, как можно активировать программу, перед инструкцией `JNZ` изменить состояние флага `ZF`. Давайте сделаем это.

Доходим до инструкции `JNZ`. Кликаем правой кнопкой на флаге `ZF` - `Reset`. И видим в консоли надпись - `"The program has been activated!"`.

Проверяем работу программы в командной строке.

prog-4.1.exe 1 2

Result: 3

Соответствующие изменения были сделаны в реестре и программа считается активированной.

То, что мы сейчас сделали нельзя сохранить в файле. Для того, чтобы изменения можно

было сохранить, нужно изменять инструкции в коде программы.

Давайте удалим записи с реестра, которые сделала программа и восстановим первоначальное состояние.

Открывает regedit и удаляем ветку (HKEY_CURRENT_USER\Software\GeekBrains\Prog4). Мы видим инструкцию MOVZX ECX, AL которая загружает значение в регистре AL в регистр ECX. Она занимает 3 байта. Во время патчинг нельзя менять количество байт, иначе поедут смещения и мы превратим программу в кирпич.

Итак, у нас есть три байта и нам нужно записать значение 1 в регистр ECX. Сразу напрашивается инструкция MOV ECX, 1.

Давайте найдем похожую инструкцию в коде нашей программы и посмотри ее опкод.

Ctrl + F (MOV ECX, 1) - ничего не найдено. Такой инструкции в коде программы нет.

Давайте попробуем найти не с единицей, а с двойкой.

Ctrl + F (MOV ECX, 2). Такая инструкция есть. Ее опкод b9 02000000 и такая инструкция

нам не подойдет, так как она занимает 5 байт. Первый байт это инструкция и 4 байта это единица.

Возвращаемся к нашему коду. Кликаем правой кнопкой - Go to - Origin.

В такие моменты нужно проявить фантазию и придумать, как за 3 байта получить значение 1 в регистре ECX.

Я придумал следующее. Первая инструкция XOR ECX, ECX, ее опкод 33C9 и составляет 2

байта, а вторая это инструкция INC ECX, ее опкод 41 и составляет 1 байт. В сумме это 3 байта и нам это подходит.

Давайте проверим.

Кликаем правой кнопкой на инструкции Binary - Edit (33C941).

Видим, что код изменился. Давайте выполним эти инструкции. Мы прошли проверку ключа.

Это вариант уже может быть сохранен на диск.

Кликаем правой кнопкой на любое место в коде Copy to executable - All modifications - Copy

All. Появляется новое окно, кликаем на коде правой кнопкой и Save file. Указываем имя новой программы prog-4.1-any-key.exe.

Проверяем работу программы в командной строке.

```
prog-4.1-any-key.exe qwerty
```

```
The program has been activated!
```

Отлично! Программа теперь активируется с любым ключом.

А теперь давайте усложним задачу и сделаем так, чтобы программа вообще не требовала активации.

Возвращаемся в отладчик. Откатываем изменения.

Окно Patches - Restore Original Code.

Далее по коду мы видим строку - "The number of starts has expired.", которая оповещает нас о том, что количество запусков программы исчерпано. Это происходит тогда когда, значение параметра Counter равно 0.

Будет ли напечатана эта строки или нет зависит от функции 0x00401300, которая, видимо, проверяет счетчик.

Мы точно не знаем, что делает эта функция, поэтому просто убрать вызов этой функции будет неправильно.

Мы пойдем по-другому пути. Сразу после вызова функции мы видим условный переход

JNZ. Именно от него зависит, попадем мы на код, которая завершает работу программы или нет.

Нам важно, чтобы этот переход осуществлялся всегда. Заменяем условный переход JNZ на безусловный JMP.

Кликаем правой кнопкой на инструкции Binary - Edit (EB14).

Видим, что код изменился.

Давайте изменим состояние счетчика в реестре на ноль, чтобы спровоцировать состояние, когда количество попыток запуска закончится.
Counter - 0.

Давайте выполним эти инструкции. Видим в консоли, что программа успешно отработала, несмотря на то, что счетчик равен 0. Мы прошли проверку ключа. Это вариант уже может быть сохранен на диск.

Кликаем правой кнопкой на любое место в коде Copy to executable - All modifications - Copy

All. Появляется новое окно, кликаем на коде правой кнопкой и Save file. Указываем имя новой программы prog-4.1-cracked.exe.

Проверяем работу программы в командной строке.

prog-4.1-cracked.exe 1 2

Result: 3

Как мы видим, программа работает несмотря на то, что счетчик равен нулю.

```
// Исходный код программы prog-4.1.c

#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#pragma comment(lib, "Advapi32.lib")

void usage(char *);
bool check_key(char *);
int sum(int, int);
bool check_numbers(char *, char *);
bool license(void);
bool trial(void);
bool check_trial(void);
bool isRegistered(void);

int main(int argc, char* argv[]) {

    if (argc == 1 || argc > 3) {
        usage(argv[0]);
        return 0;
    }

    if (argc == 2) {
        if (!check_key(argv[1])) {
            printf("\nLicense key is incorrect.\n");
            return 0;
        }

        if (license()) {
            printf("\nThe program has been activated!\n");
        } else {
            printf("\nUnknown error! (license)\n");
        }
        return 0;
    }
}
```

```

        if (isRegistered()) {
            if (!check_trial()) {
                printf("\nThe number of starts has expired.\n");
                return 0;
            }
        } else {
            if (!trial()) {
                printf("\nUnknown error! (trial)\n");
                return 0;
            }
        }

        if (!check_numbers(argv[1], argv[2])) {
            usage(argv[0]);
            return 0;
        }

        printf("\nResult: %d\n", sum(atoi(argv[1]), atoi(argv[2])));
    }

void usage(char *program_name) {
    printf("\nUsage:\n");
    printf("\n\t%s <number> <number>\n", program_name);
    printf("\nActivation:\n");
    printf("\n\t%s <license key>\n", program_name);
}

bool isRegistered(void) {
    HKEY hkey;
    LPCTSTR PATH = TEXT("Software\\GeekBrains\\Prog4");
    LPCTSTR NAME = TEXT("Counter");

    LONG nError = RegOpenKeyEx(HKEY_CURRENT_USER, PATH, 0, KEY_READ,
&hkey);
    if (nError != ERROR_SUCCESS) {
        return false;
    }

    nError = RegGetValue(hkey, NULL, NAME, RRF_RT_REG_SZ, 0, NULL,
NULL);
    if (nError != ERROR_SUCCESS) {
        RegCloseKey(hkey);
        return false;
    }

    RegCloseKey(hkey);

    return true;
}

```



```

bool check_key(char* key) {

    if (strlen(key) != 19) {
        return false;
    }

    if (key[4] != '-' || key[9] != '-' || key[14] != '-') {
        return false;
    }

    for (int block_offset = 0; block_offset < 20; block_offset += 5) {
        if ((key[block_offset] + key[block_offset+1] +
key[block_offset+2] + key[block_offset+3] - 192) != 10) {
            return false;
        }
    }

    return true;
}

int sum(int a, int b) {
    return a + b;
}

bool check_numbers(char *num1, char *num2) {
    for (int i = 0; i < strlen(num1); i++) {
        if (num1[i] < 48 || num1[i] > 57) {
            return false;
        }
    }

    for (int i = 0; i < strlen(num2); i++) {
        if (num2[i] < 48 || num2[i] > 57) {
            return false;
        }
    }

    return true;
}

bool license(void) {

    HKEY hkey;
    LPCTSTR PATH = TEXT("Software\\GeekBrains\\Prog4");
    LPCTSTR NAME = TEXT("Counter");

    char value[] = "10000";

    LONG nError = RegCreateKeyEx(HKEY_CURRENT_USER, PATH, 0, NULL,
REG_OPTION_NON_VOLATILE, KEY_WRITE, NULL, &hkey, NULL);

```

```

        if (nError != ERROR_SUCCESS) {
            return false;
        }

        nError = RegSetValueExA(hkey, NAME, 0, REG_SZ, value,
sizeof(value));
        if (nError != ERROR_SUCCESS) {
            RegCloseKey(hkey);
            return false;
        }

        RegCloseKey(hkey);

        return true;
    }

bool trial(void) {

    HKEY hkey;
    LPCTSTR PATH = TEXT("Software\\GeekBrains\\Prog4");
    LPCTSTR NAME = TEXT("Counter");

    char value[] = "9";

    LONG nError = RegCreateKeyEx(HKEY_CURRENT_USER, PATH, 0, NULL,
REG_OPTION_NON_VOLATILE, KEY_WRITE, NULL, &hkey, NULL);
    if (nError != ERROR_SUCCESS) {
        return false;
    }

    nError = RegSetValueExA(hkey, NAME, 0, REG_SZ, value,
sizeof(value));
    if (nError != ERROR_SUCCESS) {
        return false;
    }

    printf("\nTrial Period: %d of 10\n", atoi(value));

    RegCloseKey(hkey);

    return true;
}

bool check_trial(void) {

    HKEY hkey;
    LPCTSTR PATH = TEXT("Software\\GeekBrains\\Prog4");
    LPCTSTR NAME = TEXT("Counter");

    char value[64];

```

```

DWORD value_size = 64;

LONG nError = RegOpenKeyEx(HKEY_CURRENT_USER, PATH, 0, KEY_READ,
&hkey);
if (nError != ERROR_SUCCESS) {
    return false;
}

nError = RegGetValue(hkey, NULL, NAME, RRF_RT_REG_SZ, 0, value,
&value_size);
if (nError != ERROR_SUCCESS) {
    RegCloseKey(hkey);
    return false;
}

int tmp = atoi(value);

if (tmp == 0) {
    RegCloseKey(hkey);
    return false;
}

itoa(--tmp, value, 10);

nError = RegOpenKeyEx(HKEY_CURRENT_USER, PATH, 0, KEY_WRITE,
&hkey);
if (nError != ERROR_SUCCESS) {
    RegCloseKey(hkey);
    return false;
}

nError = RegSetValueExA(hkey, NAME, 0, REG_SZ, value,
sizeof(value));
if (nError != ERROR_SUCCESS) {
    RegCloseKey(hkey);
    return false;
}

RegCloseKey(hkey);

if (tmp < 10) {
    printf("\nTrial Period: %d of 10\n", atoi(value));
}

return true;
}

```

Компиляция:

```
> cl prog-4.1.c -Od /Gs- /GS- /link /DYNAMICBASE:NO /NXCOMPAT:NO
```

Запуск:

```
> prog-4.1.exe
```

Выполнил: AndreiM