При запуске программы видим

```
Enter your license key: aaaa-3e33--dd
Invalid license key. Exiting...
```

Значит, существует алгоритм проверки. Идем в гидру искать функцию, которая выводит подобные строки – она может привести к ключу, алгоритму его проверки и тому подобное

Но!

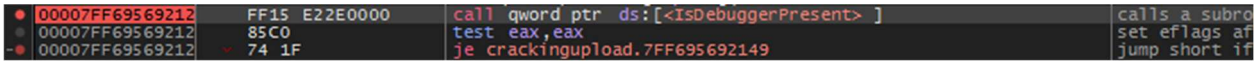
Оказывается, в этой программе существует проверка отладчика

```
local_18 = DAT_140008000 ^ (ulonglong)auStack_b8;
Debug_Checkup= IsDebuggerPresent();
if (Debug_Checkup != 0) {
  FUN_140002d10((basic_ostream<>*)cout_exref,"Debugger detected. Exiting...\n");
                    /* WARNING: Subroutine does not return */
  ExitProcess(1);
}
```

То есть защита от дебаггинга

И, естественно, если запустить её через отладчик, то мы не дойдем даже до ввода пароля – программа увидит активный дебаггер и завершит работу. Значит, надо её обойти – патчем.

Вызов функции IsDebuggerPresent происходит по адресу 140002120



Попробуем сначала заполнить место вызова функции командами NOP, чтобы проверка просто не вызывалась

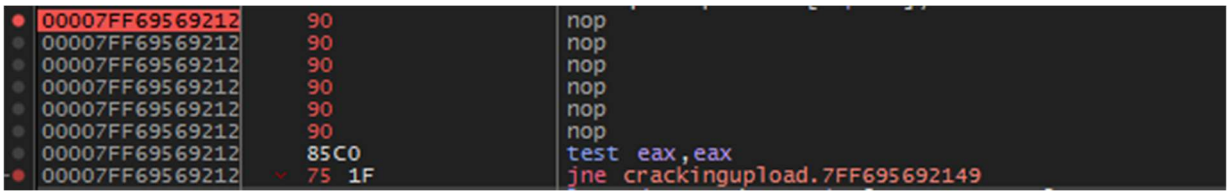
Однако это не помогло

```
g.exe
Debugger detected. Exiting...
```

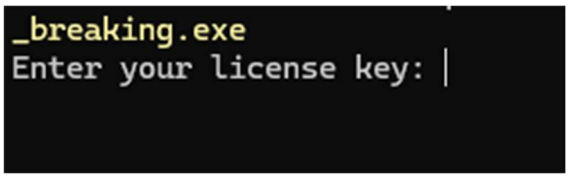
Тогда попробуем изменить инструкцию проверки – je на jne

```
exe
Debugger detected. Exiting...
```

Результат тот же. Значит, надо сделать и то, и другое



Получилось, проверка дебаггера теперь не производится



Значит, можно разбирать программу дальше. Тем более, если в ней есть защита от отладки, то в процессе отладки есть вероятность увидеть чувствительные данные и процессы – посимвольная проверка пароля, зашитый пароль, etc.

Однако я решил посмотреть в отладчике в модуле .exe ссылки на строки, и нашел следующий интересный элемент

Адрес=00007FF6803D216E

Дизассемблированный код=movups xmm0,xmmword ptr ds:[7FF6803D5490]

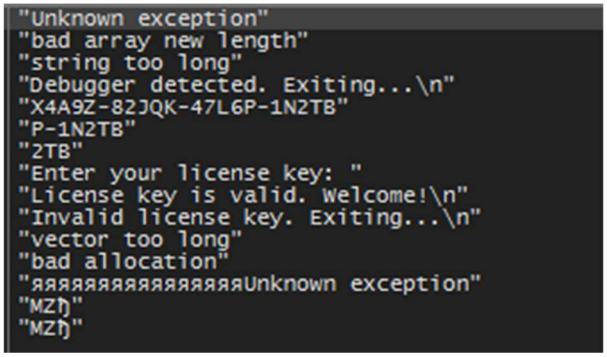
Адрес Строки=00007FF6803D5490

Строка="X4A9Z-82JQK-47L6P-1N2TB"

Выглядит как эталонный или шаблонный ключ, возможно шаблонный, но с эталонным ключом элементной проверки:

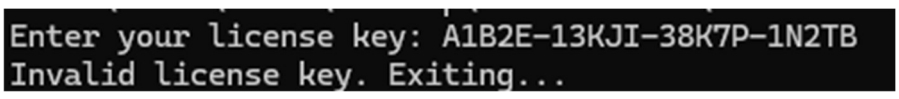
Надо попробовать проверить, ключ действительно должен быть именно таким, или он должен содержать в себе эталонный компонент валидации

В других ссылках на строки в модуле исполняемой программы есть следующие интересующие нас части ключа



P-1N2TB и 2TB

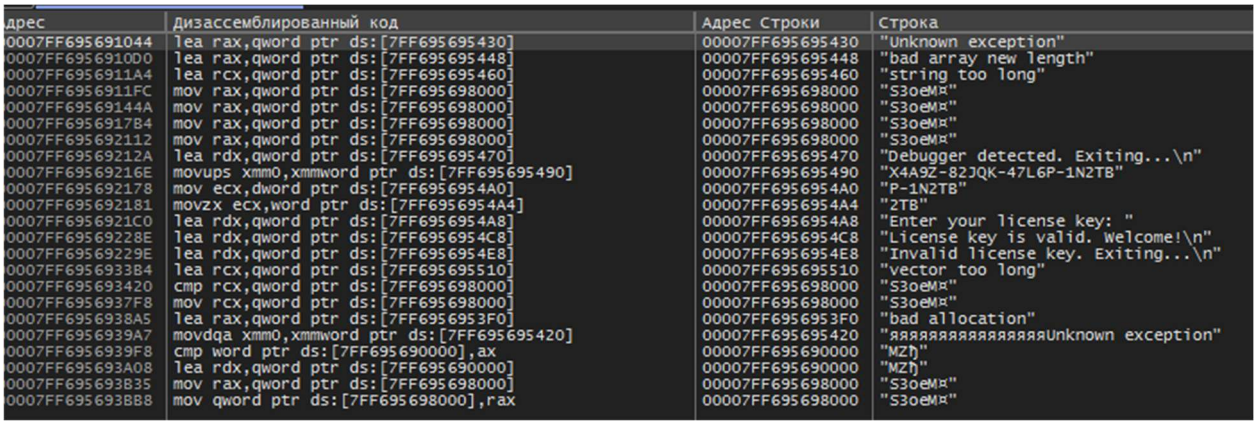
Но введенный ключ с сохранением этих участков ключа оказался не валиден



Значит, в описанной выше строке лежал именно нужный нам статичный лицензионный ключ

Вопрос в следующем – а так ли был необходим патч, если ссылки на строки можно посмотреть в отладчике ДО вызова инструкции проверки на активный дебаггер?

На скриншоте оригинальная программа, без патчей



дрес	Дизассемблированный код	Адрес Строки	Строка
0007FF695691044	lea rax,qword ptr ds:[7FF695695430]	00007FF695695430	"Unknown exception"
0007FF6956910D0	lea rax,qword ptr ds:[7FF695695448]	00007FF695695448	"bad array new length"
0007FF6956911A4	lea rcx,qword ptr ds:[7FF695695460]	00007FF695695460	"string too long"
0007FF6956911FC	mov rax,qword ptr ds:[7FF695698000]	00007FF695698000	"S3oeMR"
0007FF69569144A	mov rax,qword ptr ds:[7FF695698000]	00007FF695698000	"S3oeMR"
0007FF6956917B4	mov rax,qword ptr ds:[7FF695698000]	00007FF695698000	"S3oeMR"
0007FF695692112	mov rax,qword ptr ds:[7FF695698000]	00007FF695698000	"S3oeMR"
0007FF69569212A	lea rdx,qword ptr ds:[7FF695695470]	00007FF695695470	"Debugger detected. Exiting...\n"
0007FF69569216E	movups xmm0,xmmword ptr ds:[7FF695695490]	00007FF695695490	"X4A9Z-82JQK-47L6P-1N2TB"
0007FF695692178	mov ecx,dword ptr ds:[7FF6956954A0]	00007FF6956954A0	"P-1N2TB"
0007FF695692181	movzx ecx,word ptr ds:[7FF6956954A4]	00007FF6956954A4	"2TB"
0007FF6956921C0	lea rdx,qword ptr ds:[7FF6956954A8]	00007FF6956954A8	"Enter your license key: "
0007FF69569228E	lea rdx,qword ptr ds:[7FF6956954C8]	00007FF6956954C8	"License key is valid. Welcome!\n"
0007FF69569229E	lea rdx,qword ptr ds:[7FF6956954E8]	00007FF6956954E8	"Invalid license key. Exiting...\n"
0007FF6956933B4	lea rcx,qword ptr ds:[7FF695695510]	00007FF695695510	"vector too long"
0007FF695693420	cmp rcx,qword ptr ds:[7FF695698000]	00007FF695698000	"S3oeMR"
0007FF6956937F8	mov rcx,qword ptr ds:[7FF695698000]	00007FF695698000	"S3oeMR"
0007FF6956938A5	lea rax,qword ptr ds:[7FF6956953F0]	00007FF6956953F0	"bad allocation"
0007FF6956939A7	movdqa xmm0,xmmword ptr ds:[7FF695695420]	00007FF695695420	"яяяяяяяяяяяяяяяUnknown exception"
0007FF6956939F8	cmp word ptr ds:[7FF695690000],ax	00007FF695690000	"MZђ"
0007FF695693A08	lea rdx,qword ptr ds:[7FF695690000]	00007FF695690000	"MZђ"
0007FF695693B35	mov rax,qword ptr ds:[7FF695698000]	00007FF695698000	"S3oeMR"
0007FF695693BB8	mov qword ptr ds:[7FF695698000],rax	00007FF695698000	"S3oeMR"

И в ней уже лежит ключ. То есть он не собирается во время выполнения программы после проверки на дебаггер или что-то подобное.

Да, возможно, что патчинг программы был избыточным действием, но, по крайней мере, я понял, как обходить такую проверку, так что польза всё равно есть