

Tutorial will describe manually unpacking yoda Protector newest version 1.03.3. Tutorial will focus on main yP problem, running protected file under debugger.

## Introduction:

As I said, yoda's Protector 1.03.3 is last yoda protector version and author has decided to stop project. This tutorial will show how anti-debug tricks can be easy avoided and bypassed.

Target and some tools are needed :

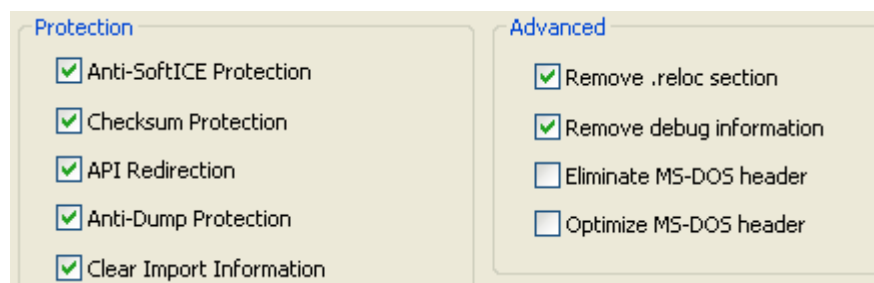
- OllyDbg 1.10
- LordPE
- ImpREC or PE Tools
- Windows XP
- Target : Yoda's Protector 1.03.3

yoda Protector is based on yoda's Cryptor frame, only that new tricks are added. Old tricks are :

- PE header erasing (which is pointless)
- CRC checking (code and file)
- IsDebuggerPresent check
- API redirecting and destroying import information.

New tricks are terminating Olly and possible freeing Windows XP.

## Yoda's Options and new tricks :

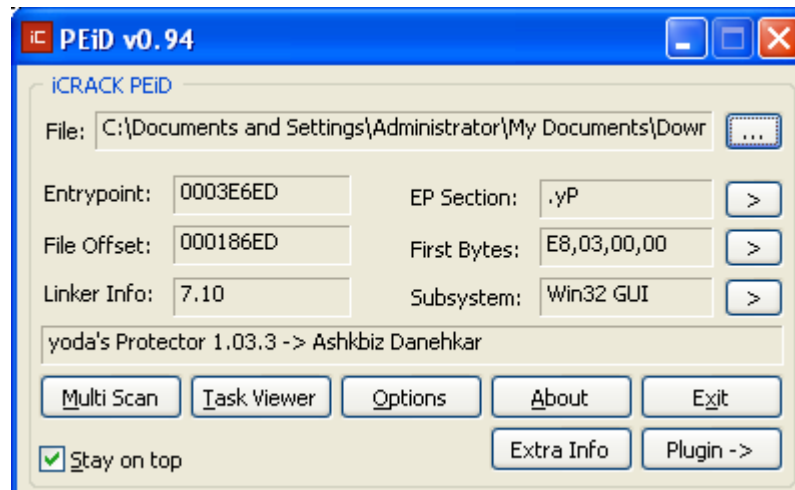


Let's see how Olly is killed. Protector is using combination of API's to get PID number of all running processes. Then it search for process that started (ollydbg in our case) and terminate it. It compares PID of that process with it's own PID. If those PID's are not same, it will terminate that process.

Second trick is more annoying. Protector will use BlockInput API before any other check. That API blocks input devices (mouse, keyboard, ..) so we are blocked from our system. Then protector will do other checks and decrypting. If in meantime protector stops on some exception or Olly is found, our system will wait for us to take action but we cannot do nothing except restart windows. If everything is passed fine, protector will again use BlockInput API to unblock input devices. Pretty smart trick.

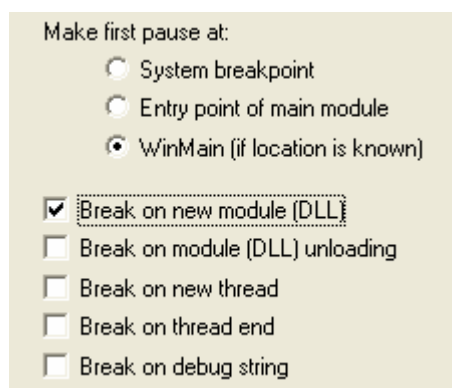
### Target analysing :

– First, we use PeiD to analyse and get more target's information :

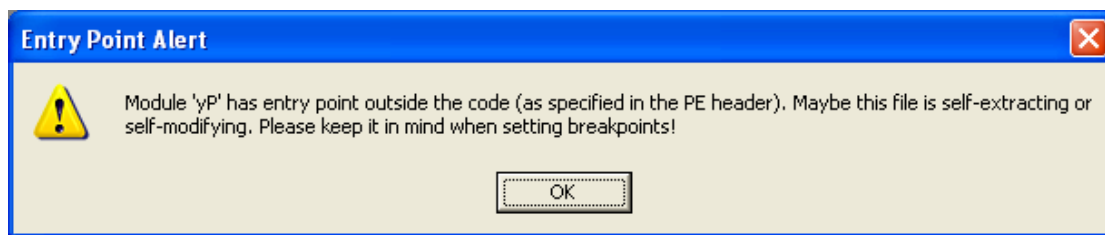


### Reaching OEP :

– Before manual unpacking, you need configure OllyDBG's options follow below image to OllyDBG break at module **user32.DLL**, prevent **BlockInput** API excutation:



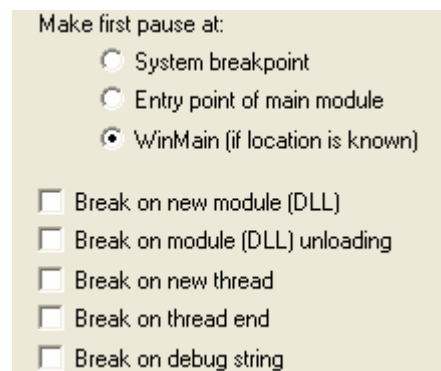
– Load target into OllyDBG, you see a meesagebox appear, press OK to continue :



– Then press F9 untill you see that User32.dll is loaded:

|          |          |          |          |                                 |                                     |
|----------|----------|----------|----------|---------------------------------|-------------------------------------|
| 00400000 | 00047000 | 0043E6ED | yp       |                                 | C:\Documents and Settings\Administr |
| 77D40000 | 00090000 | 77D50EB9 | User32   | 5.1.2600.2180 (xpsp_sp2_rtm.040 | C:\WINDOWS\system32\User32.dll      |
| 7C800000 | 000F4000 | 7C80B436 | kernel32 | 5.1.2600.2180 (xpsp_sp2_rtm.040 | C:\WINDOWS\system32\kernel32.dll    |
| 7C900000 | 000B0000 | 7C913156 | ntdll    | 5.1.2600.2180 (xpsp_sp2_rtm.040 | C:\WINDOWS\system32\ntdll.dll       |

– After, you configure OllyDBG's options as original :



– After that we can uncheck option for breaking on new module. Now we need just to patch BlockInput API so it doesn't block devices. Simply select, "go to, expression" and enter BlockInput. Ok and we land in **user32.dll** on that API (this looks on my system):

|          |             |      |                 |
|----------|-------------|------|-----------------|
| 77D9C61A | B8 36110000 | mov  | eax, 1136       |
| 77D9C61F | BA 0003FE7F | mov  | edx, 7FFE0300   |
| 77D9C624 | FF12        | call | dword ptr [edx] |
| 77D9C626 | C2 0400     | retn | 4               |

– To kill API, just NOP all to RETN 4:

|          |             |      |                 |  |
|----------|-------------|------|-----------------|--|
| 77D9C61A | B8 36110000 | mov  | eax, 1136       | <div> <div>Backup</div> <div>Copy</div> <div>Binary</div> <div>Assemble</div> <div>Label</div> <div>Comment</div> <div>Breakpoint</div> </div> |
| 77D9C61F | BA 0003FE7F | mov  | edx, 7FFE0300   |  |
| 77D9C624 | FF12        | call | dword ptr [edx] |  |
| 77D9C626 | C2 0400     | retn | 4               |  |
| 77D9C629 | 90          | nop  |                 |  |
| 77D9C62A | 90          | nop  |                 | <div> <div>Edit</div> <div>Fill with 00's</div> <div>Fill with NOPs</div> <div>Binary copy</div> </div>  |
| 77D9C62B | 90          | nop  |                 |  |
| 77D9C62C | 90          | nop  |                 |  |

– Press F2 to set BP at RETN 4 :

|          |         |        |
|----------|---------|--------|
| 77D9C61A | 90      | nop    |
| 77D9C61B | 90      | nop    |
| 77D9C61C | 90      | nop    |
| 77D9C61D | 90      | nop    |
| 77D9C61E | 90      | nop    |
| 77D9C61F | 90      | nop    |
| 77D9C620 | 90      | nop    |
| 77D9C621 | 90      | nop    |
| 77D9C622 | 90      | nop    |
| 77D9C623 | 90      | nop    |
| 77D9C624 | 90      | nop    |
| 77D9C625 | 90      | nop    |
| 77D9C626 | C2 0400 | retn 4 |

– We have killed this API and with that we avoid blocking devices, but we need to prevent Olly killing. There is similar simple solution for that. yoda uses CreateToolhelp32Snapshot to get all processes and couple others to walk through all processes. But it uses GetCurrentProcessId to get PID of itself. Then yoda will check is process who started it has same PID as itself (ei. did protected file started through some debugger or not) and if not, it will terminate that process. We can do next to prevent killing Olly.

– Open LordPE or PE Tools and get PID of OllyDbg.exe. In my case, PID is 888 (your computer is different).

|   |          |          |          |
|---|----------|----------|----------|
| c:\program files\internet download manager\i...   | 000008A4 | 00400000 | 00040000 |
| c:\program files\microsoft office\office10\win... | 000009A4 | 30000000 | 00A41000 |
| e:\cracking\tools\ollydbg collection\ollydbg_...  | 00000888 | 00400000 | 00164000 |
| e:\cracking\tools\lord pe 1.4\lordpe.exe          | 00000E64 | 00400000 | 00036000 |

– "Go to , expression", enter **GetCurrentProcessId** and click ok. You are in API:

|          |                |      |                         |
|----------|----------------|------|-------------------------|
| 7C80994E | 64:A1 18000000 | mov  | eax, dword ptr fs:[18]  |
| 7C809954 | 8B40 20        | mov  | eax, dword ptr [eax+20] |
| 7C809957 | C3             | retn |                         |

– That api will return PID of protected file, but I will patch it to return Olly PID. And our protected file will think that it is Olly itself, check:

|          |                |      |                         |  |
|----------|----------------|------|-------------------------|--|
| 7C80994E | 64:A1 18000000 | mov  | eax, dword ptr fs:[18]  |  |
| 7C809954 | 8B40 20        | mov  | eax, dword ptr [eax+20] |  |
| 7C809957 | C3             | retn |                         |  |
| 7C809958 | 90             | nop  |                         |  |
| 7C809959 | 90             | nop  |                         |  |
| 7C80995A | 90             | nop  |                         |  |
| 7C80995B | 90             | nop  |                         |  |

– You edit instruction at address **7C80994E** follow :

**MOV EAX, OllyDBG's PID (mean is 888)**

|          |             |      |          |
|----------|-------------|------|----------|
| 7C80994E | B8 88080000 | MOV  | EAX, 888 |
| 7C809953 | 90          | NOP  |          |
| 7C809954 | 90          | NOP  |          |
| 7C809955 | 90          | NOP  |          |
| 7C809956 | 90          | NOP  |          |
| 7C809957 | C3          | RETN |          |

– Continue, you must kill **IsDebuggerPresent** by one of all :

**Method 1** : (Manual Fix)

– Press CTRL + G, and type **IsDebuggerPresent**, you NOP and edit to **MOV EAX, 0** :

|          |             |            |
|----------|-------------|------------|
| 7C812E03 | B8 00000000 | MOV EAX, 0 |
| 7C812E08 | 90          | NOP        |
| 7C812E09 | 90          | NOP        |
| 7C812E0A | 90          | NOP        |
| 7C812E0B | 90          | NOP        |
| 7C812E0C | 90          | NOP        |
| 7C812E0D | 90          | NOP        |
| 7C812E0E | 90          | NOP        |
| 7C812E0F | 90          | NOP        |
| 7C812E10 | C3          | RETN       |

**Method 2** : using OllyDBG's plugin

– We need use plugin to hide Olly from IsDebuggerPresent check :

|                |           |
|----------------|-----------|
| IsDebugPresent | Hide      |
| Labeler        | Restore   |
| Labelmaster    | Option    |
| LoadDll        |           |
| NonaWrite      | About     |
| ODbgScript     | Dumper    |
| OllyDump       |           |
| OllyMachine    | ExtraHide |

– Press F9 to run target (remember you break at BP on BlockInput – RETN 4). We will stop two times on bp on patched BlockInput API.

|          |                 |                             |
|----------|-----------------|-----------------------------|
| 7C90EB94 | C3              | RETN                        |
| 7C90EB95 | 8DA424 00000000 | LEA ESP, DWORD PTR SS:[ESP] |
| 7C90EB9C | 8D6424 00       | LEA ESP, DWORD PTR SS:[ESP] |
| 7C90EBA0 | 90              | NOP                         |

– Press ALT + M to open "Memory Map" window and place memory bp on access on first section and run. OEP is reached:

|          |          |        |       |                   |                                 |    |        |
|----------|----------|--------|-------|-------------------|---------------------------------|----|--------|
| 00400000 | 00001000 | yP     |       | PE header         | Imag                            | RW | RWE    |
| 00401000 | 00033000 | yP     |       | code, data        |                                 |    |        |
| 00434000 | 00005000 | yP     | .rsrc | code, resource3s  | Actualize                       |    |        |
| 00439000 | 00005000 | yP     | .x01  | code              | View in D1s4ss3mbl3r            |    | Enter  |
| 0043E000 | 00009000 | yP     | .yP   | code, imports     | Dvmp in DSM                     |    |        |
| 00450000 | 00009000 |        |       |                   | Dvmp                            |    |        |
| 00510000 | 00002000 |        |       |                   | Search                          |    | Ctrl+B |
| 00520000 | 00103000 |        |       |                   |                                 |    |        |
| 00630000 | 000F6000 |        |       |                   |                                 |    |        |
| 00930000 | 00100000 |        |       |                   |                                 |    | F2     |
| 00A30000 | 08000000 | rsaenh |       | PE header         |                                 |    |        |
| 0FFD0000 | 00001000 | rsaenh |       | code, imports, ex | Set m3m0ry br34kp01nt on access |    |        |
| 0FFD1000 | 00021000 | rsaenh | .text |                   | Set m3m0ry br34kp01nt on write  |    |        |

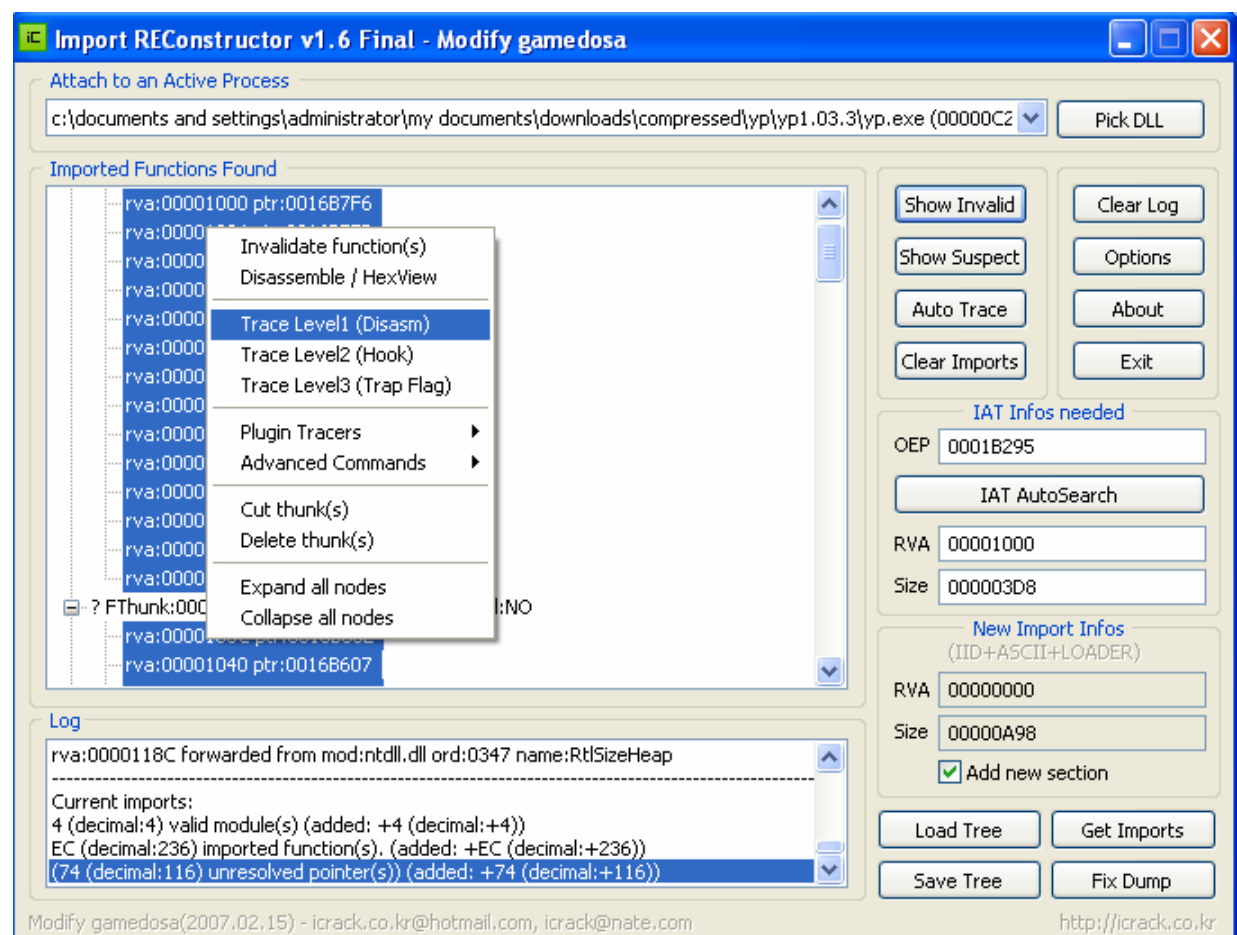
– Press F9 to RUN target, you break target's OEP :

|          |               |      |                            |  |                        |
|----------|---------------|------|----------------------------|--|------------------------|
| 0041B295 | 6A 60         | PUSH | 60                         |  |                        |
| 0041B297 | 68 D05B4000   | PUSH | 405BD0                     |  |                        |
| 0041B29C | E8 670D0000   | CALL | 0041C008                   |  | yP.0041C008            |
| 0041B2A1 | BF 94000000   | MOV  | EDI, 94                    |  |                        |
| 0041B2A6 | 8BC7          | MOV  | EAX, EDI                   |  |                        |
| 0041B2A8 | E8 C3FCFFFF   | CALL | 0041AF70                   |  | yP.0041AF70            |
| 0041B2AD | 8965 E8       | MOV  | DWORD PTR SS:[EBP+18], ESP |  |                        |
| 0041B2B0 | 8BF4          | MOV  | ESI, ESP                   |  |                        |
| 0041B2B2 | 893E          | MOV  | DWORD PTR DS:[EBI], EDI    |  |                        |
| 0041B2B4 | 56            | PUSH | ESI                        |  |                        |
| 0041B2B5 | FF15 48114000 | CALL | NEAR DWORD PTR DS:[401148] |  | kernel32.GetVersionExA |

– You dump Full by PE Tool or Lord PE, default it save with name **dumped.exe** :

| Path                                     | PID      | Image Base | Image Size |
|--|----------|------------|------------|
| [System Idle Process]                    | 00000888 | 30000000   | 00A41000   |
| c:\documents and settings\administrato   |          | 0000       | 00047000   |
| c:\program files\techsmith\snagit 8\сна  |          | 0000       | 00630000   |
| c:\program files\techsmith\snagit 8\tsch |          | 0000       | 0000E000   |

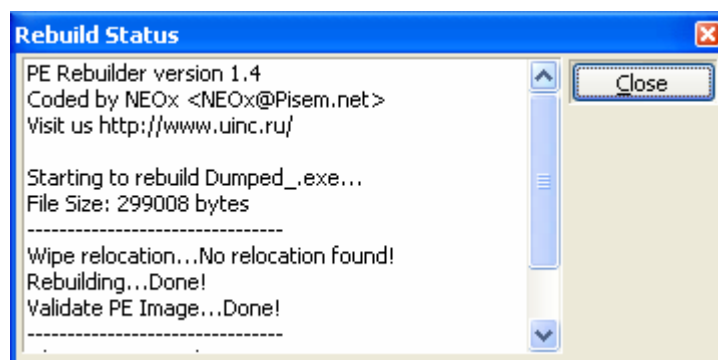
– Using ImpREC to Fix Import. There are very bad import, right click choose **Trace Level1 (Disasm)** follow :



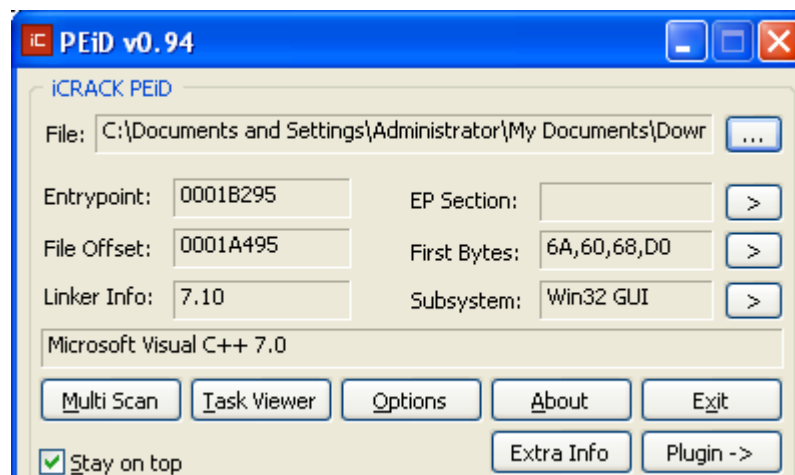
– After fixing :

```
+ advapi32.dll FThunk:00001000 NbFunc:E (decimal:14) valid:YES
+ comctl32.dll FThunk:0000103C NbFunc:4 (decimal:4) valid:YES
+ gdi32.dll FThunk:00001050 NbFunc:14 (decimal:20) valid:YES
+ glu32.dll FThunk:000010A4 NbFunc:2 (decimal:2) valid:YES
+ kernel32.dll FThunk:000010B0 NbFunc:56 (decimal:86) valid:YES
+ opengl32.dll FThunk:0000120C NbFunc:1C (decimal:28) valid:YES
+ shell32.dll FThunk:00001280 NbFunc:4 (decimal:4) valid:YES
+ user32.dll FThunk:00001294 NbFunc:4A (decimal:74) valid:YES
+ comdlg32.dll FThunk:000013C0 NbFunc:2 (decimal:2) valid:YES
+ imagehlp.dll FThunk:000013CC NbFunc:2 (decimal:2) valid:YES
```

– Press **Fix Dump**, and choose file **dumped.exe** to finish Fix IAT. You can use Lord PE or PE Tools to Rebuild PE :



– Using PeiD retest and you know Yoda's is coded by language **Microsoft Visual C++ 7.0**.



**Thanks to haggar's tutorial about Yoda's Protector 1.03.3**

– Unpacking done !