ScyllaHide v1.1 - Documentation

Developed by NtQuery and cypher

2014-08-15

Contents

1 Features

1.1 Anti-Anti-Debug

1.1.1 Process Environment Block (PEB)

1.2 Special

1.2.1 DLL Injection

1.2.2 Prevent Thread Creation

1.2.3 RunPE Unpacker

1.3 OllyDbg v1 Specific

1.3.1 Remove entry point breakpoint

Some protectors use Thread-Local-Storage (TLS) as entrypoint and check for breakpoints at the normal PE entrypoint address. You must remove the PE entrypoint to hide your debugger. This option is necessary for VMProtect.

1.3.2 Fix Olly Bugs

This option fixes various OllyDbg bugs: ForegroundWindow, PE fix for NumOfRvaAndSizes, FPU bug, sprintf bug

1.3.3 x64 single-step fix

OllyDbg doesn't work very well on x64 operating systems. This option fixes the most annoying bug.

1.3.4 Skip Entrypoint outside code

Annoying warning can be skipped.

1.3.5 Ignore bad PE image

Annoying warning can be skipped.

1.3.6 Skip compressed code warning

Annoying warning can be skipped with a default behaviour.

1.3.7 Skip "load dll" warning

Annoying warning can be skipped with a default behaviour.

1.3.8 Break on TLS

This options sets a breakpoint to any available Thread-Local-Storage (TLS) address. This is necessary for various protectors e.g. VMProtect.

1.3.9 Advanced CTRL+G

Replaces the default OllyDbg "Go to Address" dialog. Now you can enter RVA and offset values. Be sure to select the correct module.

1.3.10 Change window caption

Change the OllyDbg window caption. This can be useful against e.g. FindWindow anti-debug tricks. You don't need to enable this, if you have the NtUser* hooks enabled! Hint: You can use it to make the currently used profile visible.

1.4 OllyDbg v2 Specific

1.4.1 Change window caption

Change the OllyDbg window caption. This can be useful against e.g. FindWindow anti-debug tricks. You don't need to enable this, if you have the NtUser* hooks enabled! Hint: You can use it to make the currently used profile visible.

1.5 IDA Specific

1.6 x64dbg Specific

1.7 TitanEngine Specific

2 Advanced Information

2.1 Special PEB Fix Information

There is a special piece of code inside the debug loop of the plugins and it seems like there is a bug:

```
    if (pHideOptions.PEBHeapFlags)
    {
        if (specialPebFix)
        {
            StartFixBeingDebugged(ProcessId, false);
            specialPebFix = false;
        }

        if (debugevent->u.LoadDll.lpBaseOfDll == hNtdllModule)
        {
            StartFixBeingDebugged(ProcessId, true);
            specialPebFix = true;
        }
    }
```

But this code is correct and very important. This nice trick removes heap artifacts (You can read more about it here: http://pferrie.tripod.com/papers/unpackers.pdf "The heap"). Themida and other protectors are checking for heap artifacts. Instead of manually wiping the artifacts, the code prevents the heap artifact creation.