Optimice FAQ

1. **How to start using it?**

   To use Optimice plugin you will need the following:

   - IDA (preferably 6.0+) and IDApython 1.5.0+
   - NASM assembler (http://www.nasm.us/)
     - NOTE: If you install NASM into a non-standard directory (different from "*C:\Program Files\nasm\nasm.exe*") you will need to edit Assembler.py file and replace '***nasm = "C:\\Program Files\\nasm\\nasm.exe***"' with appropriate file path like "*/usr/bin/nasm*".

   Next step is to either download archive[1] or fetch the latest code (with fresh fixes) from SVN[2]. Save and unpack files to some directory and start up IDA.

   Once in IDA, it is necessary to initialize *Optimice* and bind it to some hotkey. Default hotkey is **Alt+N**, but if you prefer some other key combination, you can edit Main.py and in function *setHotkey*() replace "**ALT-N**" combination with some other key combination[3]. To actually bind hotkey it is necessary to run **Main.py** script either from *File->Script File* or using hotkey **Alt+F7**. After initialization, following message will appear in IDA's output window:

   ```
   Optimice v0.13 initialized
   To optimize use ALT+N hotkey and position cursor at the beginning of code!
   ```

   Additionally, two entries will be added to IDA toolbar menu in *Edit->Plugins*. First one - "*Optimice: Manage functions*" - presents an overview of all optimized functions. Using this entry, user can manage (delete) arbitrary functions or even all Optimice data associated with IDB (more in next entry). Second entry is "*Optimice: Optimize (ALT+N)*" that can be used along with the hotkey, in order to invoke optimization process.

2. **How are sessions for different IDBs managed?**

   Optimice stores its data in two locations:

   - Segment inside IDB
     - This data is the part that you see as a product of successful optimization. One large segment usually contains all optimized functions for a single IDB. If you decide to remove all *Optimice* data, you can do it from

[1] http://code.google.com/p/optimice/downloads/list
[2] http://code.google.com/p/optimice/source/checkout
[3] It should be noted that some IDA versions have problems with assigning hotkeys to already existing combinations. In this case, you have to either remove old hotkey or find some free hotkey combination.

"*Manage Optimice*" interface by selecting arbitrary function, right clicking on it and selecting "*Delete segment*".

- *Optimice_ ** directory that is created in directory where IDB resides
    - This directory contains assembly source files that are compiled and imported into IDA as optimized functions, object files and list files. This directory also contains configuration files and state information that assembler uses to keep track of optimized functions and segment state.

3. What optimizations are supported?

- Following optimizations are currently supported:
    - Dead code elimination based on taint analysis
    - Unconditional JMP removal
    - Conditional JMP removals
        - Complementary conditions
        - Static flag conditions
    - Several heuristic methods
        - PUSH/POP removal
        - PUSH/RET to JMP modification
        - Symetric nop removal

4. What to do when assembler fails?

When NASM fails to assemble optimized code, you will encounter a message that informs you of failed assembling and Optimice will automatically open file with the code that has errors. You can then inspect reported errors and hopefully manually resolve problems. Problems usually arise from a fake code branches that contain fake/dead code that wasn't detected. In cases like this, you can replace problematic instruction with a RET instruction and try to assemble code again.

Another problem that can arise is that of different syntax in IDA and NASM. To fix these errors, you can modify syntax or operand prefixes that cause the problem and try to compile it again. If it fails again and you would like to skip that particular instruction, you can comment out that code and resume compilation.
If you encounter errors with compilation, please send sample code for me to fix it :)

5. How to add IDA comments from assembly?

One useful feature that is available when manually editing code is to add comments that will be imported into IDB after compilation. This can be a case when you deobfuscate or simplify code manually (regex style) and want to make notes as to what you did. Comments are added by inserting desired text between two ### strings. For example a

"*jmp L00003004 ; ###FAKE 2 JMP###\n*" code will be compiled and imported to IDA as a "*jmp loc_3004 ; FAKE 2 JMP*". Comments are resolved from assembly code and imported to IDA along with opcodes.