

The Black Art of Reverse engineering

(Handout)

Here's a list of common things that you'll find useful when you start doing reversing on your own. Hope it helps!

Common Opcodes

- **CMP A, B** : One of the most frequent opcodes you will encounter. It's a mnemonic for "compare". It basically subtracts the two arguments (A-B) and sets the appropriate flags. This command is usually followed by a jump instruction.
- **JE, JNE, JG, JNG, etc** : Jump on equal, jump on not-equal, jump on greater, jump on not-greater, etc. These instructions are used to direct the program flow depending on the flags. The flags can be set purposely by adding or subtraction two numbers or can be set by the **CMP** instruction. The **CMP** along with one of the jumps is the most common sequence you'll find in almost any executables.
- **ADD, SUB, XOR, etc** : These and many others constitute the "math" instructions in the opcodes set. Usually their mnemonic suggests their function, like **ADD A,B** will add A and B and put the result in A.
- **MOV A, B** : Used to move data between memory. The value of B is copied into A. B's value remains unchanged. The arguments can be register names or memory locations. The data to be moved can even be just a number,
- **PUSH** : This is used to "push" data onto the stack. Normally the function arguments are put into the stack by the use of this command.
- **CALL <location>** : This is used to call a function at the specified location. The function arguments are expected to be present in the stack during the time of call.

The arguments are to be pushed onto the stack in the opposite order in which they appear in the source code. Eg. A function “func(int a, int b)” is to be called as

```
PUSH b
PUSH a
CALL <func>
```

Useful Olly Commands

PURPOSE	METHOD	DETAILS
View all text strings in the executable	Rt click > Search for > All referenced text strings	Double clicking on the text line shows the place where the string is used. Useful for quickly finding the place of interest in the exe.
View all function calls	Rt click > Search for > All intermodular calls	Again double clicking takes us to the place where the call takes place. Very useful in finding specific function calls.
Put a breakpoint	Rt click > Breakpoint > Toggle (Shortcut F2)	Extremely useful in stopping the program flow in areas of interest. Conditional breakpoints and memory access breakpoints give us additional control.
Save changes	Rt click > Copy to executable	Used to save the modifications you made. Otherwise when you restart the exe, all the changes are lost.
Modify existing code	Double click on the line which you want to modify	Helps when you want to make changes to the code. Remember to keep the “Fill with NOPs” checked, or else you will end up with corrupted code. Also if the line you want to add is bigger than the current line, then it will overwrite to the next one, erasing it. Everytime you restart the program, the changes are reverted back.

Useful Links

- **NEW2CRACKING** – a extensive collection of tutorials for starters -
<http://new2cracking.cjb.net/>
- **ASSEMBLY LANGUAGE REFERENCE** – an exhaustive list of assembly language opcodes with explanations -
<http://www.woodmann.com/crackz/Tutorials/Drme2.htm>
- **REVERSE ENGINEERING TOOLS** – a good collection of tools useful in reversing -
<http://www.woodmann.com/crackz/Tools.htm>
- **CRACKMES.de** – a great site to find all the targets to try your hand on, beginning from easy to very difficult - <http://crackmes.de/>
- **MSDN** – The Microsoft Developer Network documentation. Useful for referencing syntaxes of Windows function calls. – <http://www.msdn.com>
- **CODE BREAKER'S JOURNAL** – An excellent site for interesting articles on reversing - <http://www.codebreakers-journal.com/>
- **CRACKZ'S REVERSE ENGINEERING PAGE** – A great site to find all you will ever need to learn in reversing, only you'll have to search for yourself -
<http://www.woodmann.com/crackz/index.html>

There is an infinite amount of material available online if you are interested in learning. Just searching will usually get you what you're looking for. If you really want to learn something new and can't find a starting place, feel free to mail me anytime.

COMPILED BY:

Sachin S. Shirwalkar

(sachins@iitb.ac.in)

<http://www.civil.iitb.ac.in/~d3sachin>