ET'S

REVERSING SCHOOL

LESSON – 0

NECESSARY TOOLS AND FUNDAMENTALS

Disclaimer

This series has been written solely for educational purposes. There is nothing that a student will be asked to do that is illegal.

No illegal "pirated" versions of software have been distributed. All programs studied in this course have been written by me or others (to whom credit has been given) for the sole intent of being disassembled and studied.

Neither I, nor anyone else involved in the production of this series, assumes any responsibility for what others do with ANY application of the knowledge received herein.

Copyright Notice

eT's Reversing School
by evilTeach

Introduction

Welcome to the world of Reverse Code Engineering ("Reversing"). This series is intended to introduce you to this fascinating field of study. There are many reasons why you may have decided to explore RCE. I hope that one of your reasons is that you enjoy being challenged intellectually. In this program you will have ample opportunity to develop an appreciation for both programmers and "reversers" by diving head-first into some interesting challenges.

The main purpose of this first lesson is to give you a little history into Reversing and to give you time to acquire and prepare the tools that you'll need. **I will not be giving you the necessary tools.** Start to take responsibility for your own education. Find the tools that you need, install them, and maybe even start to play with them. The next few lessons will help you develop your skills with these tools.

Reversing isn't something that you learn by reading. It isn't something that you learn by watching. It's something that you have to learn by **doing**. If you don't try the exercises that I present, you'll be missing out on a great deal of information. No two programs are written quite the same. Each one, even the worst, bug-ridden, undocumented, useless program has something that we can learn from it. Don't skip anything and you'll find yourself growing as you advance from lesson to lesson.

The lessons have been broken up into what I believe to be a very natural progression. Skipping from lesson to lesson would not be very beneficial. This is why I have chosen to only release lessons to students that have completed all work from the previous lesson. "Homework" will need to be assessed before you can move on to the next lesson. Each assessment will include feedback from a senior member in the school. Please take this feedback as it is intended; as constructive criticism.

Please do not share this material with anyone who is not a member of the school. I have to trust you all on this. However, think about the amount of time that I've put into developing this curriculum! I have a vision, and I would really like to see it reach fruition. If you have a friend that really, really, really wants to learn this stuff, have them sign up. They can list you as their preference for a "mentor". When the time comes, if you've advanced far enough, you'll be assigned to assist them in their education.

This is not a definitive work. If you have any input, please send it to me and I'll consider it. If I decide to use it, I will give you credit at the end of the lesson where it is used. Thank you for enrolling in the Reversing School, good luck and good cracking!

Reversing Tools

Here is a list of programs that you should have in your "toolkit". Some of them are essential. Others are optional. The more tools you have a working knowledge of, the more versatile a reverser you'll become.

Most of these tools have alternative options and competing products. I'm going to list the products (and versions) that I use.

Required Tools: These are the core of a reverser's toolkit.

Hex Editor:	HexWorkshop v2.54
Disassembler:	W32DAsm v8.9
Debugger:	SoftICE (we'll use it but it doesn't work in some newer WinOS, so let's not get TOO cozy with it!)
Documentation:	Win32API (get the online help file format)
Journal:	A notebook to keep your notes, observations and experiences — an essential part of learning! (I used to keep mine in Notepad but now I use composition books)

Optional Tools: These can help but won't be used right away.

Filemon	
Regmon	
Borland Resource Workshop	
HIEW	
MASM32	(start learning Assembly Language)
Art of Assembly	Written by Randall Hyde—A comprehensive look at Assembly Language. A new Win32 version is out but I haven't had a chance to look at it yet. I'm sure it's as good as the original!!!

A Brief (Subjective) History

What follows is my exposure to the art of RCE (Reverse Code Engineering). If anyone knows of a "history of reversing" document, please send it to me!! I would love to get permission to use it here.

Reversing has always existed—innovation demands it. As soon as someone creates anything new, others feel obligated to figure out how it was done. Car manufacturers reverse their competitor's products. Generals reverse their opponent's battle plans. Everyone seems to try and reverse a magician's tricks!!! Our target will be computer code.

My first exposure to reversing (although I didn't know what it was called back then) was in the mid eighties. Back then I had a C-64. What a wonderful machine! I taught myself BASIC and then Assembly Language on that old relic. Then, I got into copying games.

Games back then (for those of you too young to remember) came on 5.25" floppy disks. To deter piracy, the software companies developed copy protection schemes. These usually involved intentionally putting errors on the disk. Most disk copiers at the time wouldn't reproduce these errors, ignoring them instead. When running the game, if a disk error didn't occur, the game knew that you were playing an illegal copy and would stop. Of course, it didn't take long for people to figure out what was going on, and to make new utilities that would create the necessary disk errors. (Incidentally, this protection scheme is similar to the concepts used in modern CD copy protection—data is written in areas that (at the time) most CD burning/copying software didn't touch.)

Now, I was and still am a HUGE fan of RPGs. I got hooked on them with the original Ultima (yes episode 1) and played through Ultima 7 before losing interest in that series. Ultima 4 was my first "reverse" of sorts. That game had different disks that you had to make copies of to play the game. Each disk stored information about a different aspect of the game—city disk, dungeon disk, continent disk, etc. I started playing around with a track/sector editor one day and found two different types of data stored on the disk.

The first was lots and lots of text. Turns out that the text was all of the conversations that you could have with people in the game. It didn't take very long to understand how the information was formatted, and then I began changing it. I added my friends into the game, changed the names of towns, and generally had fun with it.

The second type of data that I found was generic ASCII data, in seemingly random patterns. However, by flipping through the sectors, I began to notice patterns in the data. Pulling out some graph paper I started "mapping" the data stored on the disk. It turned out that the data contained all of the maps for the game. It was a simple tile pattern. The data on disk said which graphics to draw on the screen. For example:

```
00000000000000000000000000000000
00000000000000111100000000000000
00000000000011111111100000000000
00000001111222221222211100000000
00000012222222222222222211110000
01111122222222222222222222211110
01222222222222333333333222222210
12222222223333333333333333332221
```

The '0' would represent deep water, the '1' shallow water, '2' would represent grass, and '3' would be bushes. Thus, by experimentation, I was able to decode the whole scheme from which the world was built. Then I had LOTS of fun changing the maps.

Ok, so what's the point here? Well, lots of the work in reversing boils down to recognizing patterns and finding them in a program. I didn't really understand that back then, but I realize now that I was on the right track. Experimentation is another key element. I formed a hypothesis about the data on the disk, and then changed it to see the results. It didn't hurt that I was already intimately familiar with the world of Ultima 4; I recognized the maps of towns from the general ASCII data. To really develop strong reversing skills on a Windows platform, you'll need to become fairly familiar with the OS and how it operates.

Scroll forward almost 10 years. Time spent on other distractions! In 1994 I had finished working on my BA and decided to buy a computer. I was WAY out of the loop! I bought a 486-66 PC with Windows 3.1 OS. First thing I encountered was that there wasn't really a programming language built in. In the 80's, C-64's and Apples had BASIC built in! I started doing research and realized that "most" programming seemed to be done in C (or C++). I had never even heard of these languages at the time. So…being the bizarre person that I am, I HAD to rush out and learn them. I bought a book by Microsoft Press about Visual C++ (Learn VC++ I think). Anyway, it came with VC++ 1.0 compiler. I then bought other books on C/C++ (the MS book didn't teach CRAP—what else is new?) Finally, I got my hands on VC++ 5.0 and really sunk my teeth into the modern age of Windows programming.

Now, I really recommend that anyone who wants to learn RCE should be familiar with Win32 programming. The Win32 API is written in C, so that's a good language to learn. However, MFC (Microsoft Foundation Class) is designed for C++ programmers, making Win32 programming easier. Of course, you could (and should) eventually learn Win32 Assembly Language. You'll be mucking about in ASM code quite a bit, so understanding how to code in ASM will help you understand what's going on.

Well then, I can't remember exactly when it was, or even how exactly, but I realized one day that it was possible to get a demo version of a program "patched" and that there were sites on the internet that had such patches. It might have been an old version of PSP that I first downloaded a patch for. Regardless of what program it was, here was my first exposure to the field of reversing in Windows.

I began to realize that there were people that had the ability to get down into someone else's program and figure out how it worked; even CHANGE how it worked! My thirst for knowledge kicked in again! I had to know how to do this. I started looking around, not even knowing what I was looking for. I kept looking for "how to patch" and would get sites with patches. One day I stumbled on a site (now forgotten) that had a link to a site by some guy named Fravia+. I didn't understand very much on the site and eventually lost the link (only to find the site again months later). A version of the site still exists as a mirror, but it isn't updated anymore. If you've never heard of Fravia+ or the +HCU, I recommend searching the web for more info. Fravia+ has moved on and has a new site named SearchLores (www.searchlores.org) all about searching the internet for information.

On Fravia's site I did find a bunch of tutorials, written by +ORC. Now, if you haven't heard of him—well then you HAVE to do some research. I don't believe that I would be going out on a limb to say that +ORC is **the father of Windows (and DOS) reversing**. His tutorials broke new ground, sharing his knowledge with anyone willing to try and learn from them. Apparently he wrote a series of tutorials throughout the mid 90's. According to Fravia+ and others in the +HCU, +ORC has retired and has disappeared from the internet scene (apparently in 1998?) Reading through +ORC's lessons I got a glimpse into the world of Reversing. However, it took a while before I could grasp some of the beauty that +ORC put into his writing.

Once I found +ORC's work (I found the ORC packs which included tools and target programs the he was writing about—they're still around on the net—search for them!) I started going through them and began searching for other tutorials on the internet as well.

In my quest for new knowledge I found a site called the Newbie Project. It was a site containing a series of target programs along with specific objectives to achieve during the reversing process. It also contained a message board where different "students" involved in the project could share their ideas. It was an incredible experience. Being able to share your ideas with others, bounce questions off each other, and see how others approached the same problem was an educational experience. The program lasted through 7 or 8 projects before it was dropped (too much work administrating it all I believe). Unfortunately, I found the site too late and only took part in project's 7 and 8.

One reverser who ran the project was named Sandman. He had an amazing site for newbie reversers which is sadly also gone from the internet (as far as I can tell.) It contained a wealth of knowledge that helped me on my way to evolving as a reverser.

The next stage in my development came when I found the Immortal Descendants website (also gone from the web). Up until that point, all of my experience had been in reversing protection schemes ("cracking") or working on "crack-mes". I started to get bored because, frankly, most protection schemes are lame. At Immortal Descendants I found a new class of challenges called "ReverseMes". The difference was that ReverseMe's had a much more challenging goal, adding to or repairing the functionality of a small program. This went way beyond the simple task of bypassing a security feature in a program. This was getting down into the heart of a program. What a great challenge!

Alas, even the Immortal Descendants site is gone now, and all of this knowledge is scattered all over the internet. I hope to pool some of that knowledge in this series. I believe that the path I followed, by luck or by fate, was a good one. The experiences that I've had will become the foundation of this curriculum.

In this brief history I have skipped many important people and groups. Many of them were instrumental in the evolution of RCE. I apologize if anyone feels they belong here. Please remember that this is MY view of the history—I guess it's the history of my experiences in RCE. I hope you find this series useful!

evilTeach, Feb 2002

"Give a man a crack, and he'll be hungry again tomorrow,
teach him how to crack, and he'll never be hungry again" - +ORC

"Knowledge is now free at last, everything should be free from now on, enjoy knowledge and life and work for everybody else" - +ORC

Exercises

In each lesson there will be a series of exercises that you will be asked to complete. Each exercise will focus on the topics covered in that lesson and previous lessons. Completed exercises must be submitted for review before you will receive the next lesson. The intention here is that you will learn better incrementally. Trying to master too much at once is not a good thing. Better to grow in steps, making sure that you're not missing anything along the way.

For this lesson there is only one exercise:

Find the required software mentioned earlier in the lesson. Try to find the optional materials as well. Don't install all of the software. Some of it, SoftICE for example, isn't that easy to install (on some machines). We'll cover installation in later issues as the need arises.

Good luck and see you next lesson!