SecurityTube Python Scripting Expert (SPSE)

SecurityTube Python Scripting Expert

http://www.securitytube.net

Vivek Ramachandran
Course Instructor

Module 6: Reverse Engineering

SecurityTube Python Scripting Expert

Part 1: Portable Executable Basics

http://www.securitytube.net

Vivek Ramachandran
Course Instructor

PE File Format

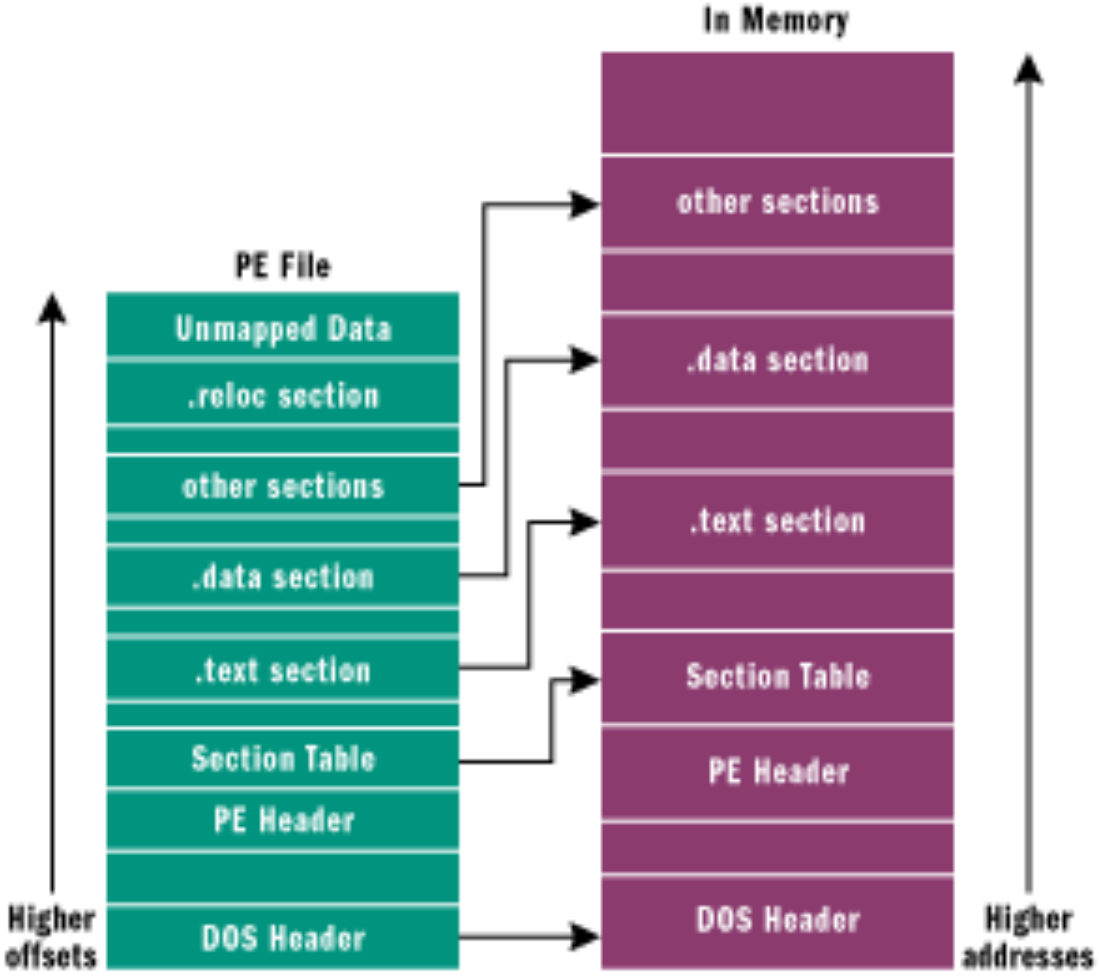
- PE is the native file format for Windows executable
- Portable across different version of windows
- Can be an EXE, DLL etc.

Understanding the PE file format

- Any executable would have to have a mix of executable code and data it uses
- Hence, PE should have minimum 1 code and 1 data section
- Windows NT defined 9 different sections possible in a PE
- Definitive guide to the PE

http://msdn.microsoft.com/en-us/magazine/cc301805.aspx

How does a PE look like in File vs Memory?



Source: MSDN

Sections in a PE file

- Code Sections
  - .text – executable instructions
- Data Sections
  - .bss – uninitialized data e.g. static variables
  - .rdata - read only data e.g. constants
  - .data – all other variables
- Export / Import Data
  - .edata - name+addresses of export functions
  - .idata - contains Import Address Table and others
- Debug Section
  - .debug – debug info for the file
- Resource Section
  - .rsrc – contains resource information
- Relocation Section
  - .reloc - relocation information
- Thread Local Storage template
  - .tls - template section for thread data storage

Analyzing a PE file

- Use a software to understand the headers and format
  - PEBrowse
  - Peview
- Search, Download and Install

Module 6: Reverse Engineering

SecurityTube Python Scripting Expert

Part 2: PE Analysis with pefile

http://www.securitytube.net

Vivek Ramachandran
Course Instructor

DIY Installation

- Install Immunity Debugger + Python
- Install pefile

Ripping a PE file in Python

- pefile is GREAT! ☺
- easy and simple to use
- comprehensive in coverage

Exercise

- Take a DLL name as input and check if a given PE imports it and print the list of imports

Module 6: Reverse Engineering

SecurityTube Python Scripting Expert

Part 3: Disassembling Code with Pydasm

http://www.securitytube.net

Vivek Ramachandran
Course Instructor

Installation of Pydasm

- It's a painful challenge ☺
- Install Pydasm (libdasm) on Windows 7 to use Python 2.7

Pydasm

- Uses libdasm
- can disassemble instructions
- can be easily incorporated into your programs

Exercise

- Create a simple program which can disassemble the first 200 bytes of executable code

- Create simple shellcode for a windows bind shell and then use pydasm to disassemble it

Module 6: Reverse Engineering

SecurityTube Python Scripting Expert

Part 4: PyDbg Basics

http://www.securitytube.net

Vivek Ramachandran
Course Instructor

Exercise

- Install Pydbg to run on Python 2.7.x
- Spend time researching online ☺

What is Debugging?

- "De" "Bug" (reduce number of bugs) ☺
- Systematic process of analyzing a program to find and fix bugs
- Debugger aids in this analysis by allowing programmer to inspect a running process

Breakpoints

- Breakpoints allow you to freeze a program in execution and allows you to inspect things such as the
  - variables
  - memory dump
  - stack
  - processor registers
  - etc.

Type of Breakpoints

- Software Breakpoints
  - when a specific instruction is to be executed
  - any number can be applied
  - cannot be used to check for memory access
- Hardware Breakpoints
  - applied for instruction execution
  - memory access (read / write)
  - maximum number depends on Architecture
  - Uses debug registers to create breakpoints

First and Second Chance Exceptions

- In a program running in a debugger, when an exception occurs, the debugger gets notified immediately
  - first chance exception
  - exception handling code is not invoked yet
- If program does not handle exception, then debugger gets notified again
  - second chance exception

- In most cases, we are interested in only Second Chance Exceptions

PyDbg

- Python based, Open Source User Mode debugger

- Allows us to script and automate debugging tasks when reversing or doing exploit research

Analyzing Crashes using Pydbg

- Will use a binary which is vulnerable to buffer overflow and try and catch exception

Exercise

- Modify the code to take a file location as input and then automatically runs to file

Exercise

- Modify the program to include full crash dump details. Explore how you can get all the info using the utils module

Module 6: Reverse Engineering

SecurityTube Python Scripting Expert

Part 5: Monitoring API Calls

http://www.securitytube.net

Vivek Ramachandran
Course Instructor

Malware Analysis and Reversing

- One of the most important tasks is to monitor API calls
- e.g. What does the malware send / recv over the network?
- Which registry keys is it setting?
- Which files is it opening?

PyDbg to the Rescue!

- Simple Steps for API Monitoring:
  - Find Address of Function you are interested in monitoring
  - set a breakpoint
  - dump context and inspect when breakpoint is hit

Exercise

- For both send / recv calls read the arguments from the stack when the breakpoint is hit and print the contents out in an intelligible way coherent with the API documentation

Analyzing Malware

- Create API monitors for the following:
  - Registry writes to “run on login/boot”
  - Opening / Writing of files
  - Send / Recv on network
- Once you create the above framework, get a malware or program sample and test against it

Module 6: Reverse Engineering

SecurityTube Python Scripting Expert

Part 6: Malware Analysis with Sandbox

http://www.securitytube.net

Vivek Ramachandran
Course Instructor

Sandbox

- Mechanism to observe the behavior of a program without allowing it to cause damage
- Monitoring of
  - API calls
  - Data sent / received over the network
  - Files, registry and other modifications
- Used in the development of AV signatures

Cuckoo Sanbox

- http://www.cuckoobox.org/
- Allows you to submit malware and look at results
- Infrastructure can be set using Virtualbox

Exercise

- Setup Cuckoo Box

- Analyze a program with it and log
  - API calls
  - Files it reads / writes
  - Host it communicates with