WinDbg: User & Kernel Mode Debugging

Vivek Ramachandran

SWSE, SMFE, SPSE, SISE, SLAE, SGDE Course Instructor

Certifications: http://www.securitytube-training.com

Pentester Academy: http://www.PentesterAcademy.com

Course Introduction

Course Objective

- Learn user and kernel mode debugging with WinDbg
- Examine CPU state, memory, program disassembly, data etc.
- Understand Windows Internals, kernel data structures, device drivers, crash analysis

Usefulness in Infosec?

- Exploit Research
- Reverse Engineering
- Malware Analysis
- Windows Internals and Rootkits

Course Outline

- Lab Setup
- WinDbg Basics
- Understanding Processes and Threads
- Debugging Multi-processor systems
- Symbols and Symbol Servers
- Debugging
  - With source
  - Without source

Course Outline (contd.)

- User Mode Debugging
  - Breakpoints
  - Watches
  - Examining CPU
  - Examining Memory – Stack, Heap, Code
  - Threads and associated storage
  - Modifying registers, data etc.
  - Disassembling

Course Outline (contd.)

- Kernel Mode Debugging
  - Kernel debug setup
  - Windows Internals basics
  - Device Driver basics
  - Kernel data structures
  - Process and Thread data structures
  - Interesting APIs and Subsystems
  - Rootkit analysis

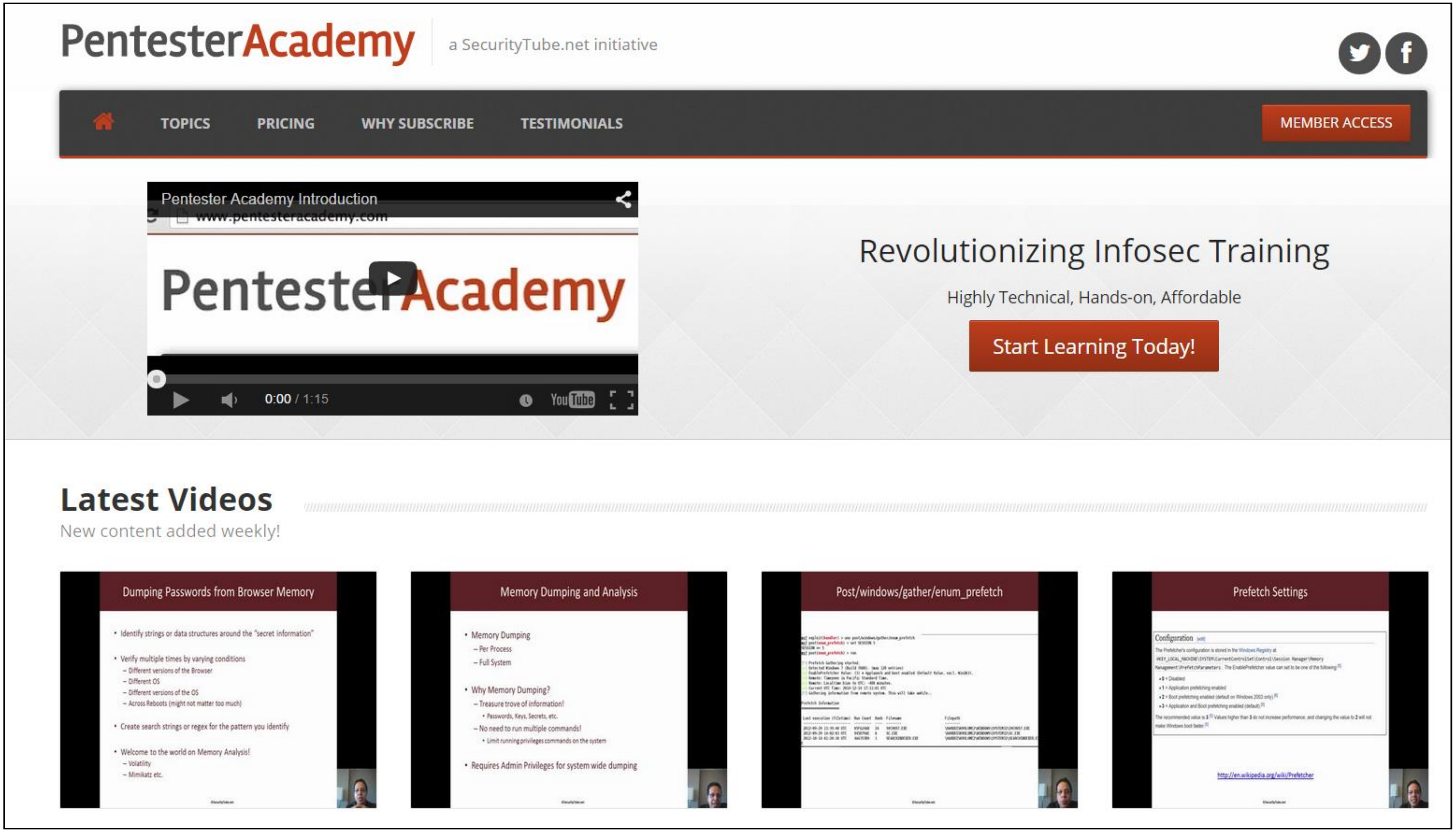
Course Outline (contd.)

- Case Studies:
  - Buffer Overflows
  - Malicious programs
    - Local
    - Network based communication
    - Rootkits and Kernel mode backdoors

Lab Setup

- Windows 8.1 x86_64
- Windows 10 x86_64
  - Technical or Insider preview
- Visual Studio 2013 / 2015 Professional
  - 90 day trial will do
  - Windows SDK
  - Windows DDK
  - Installing Help locally

Pentester Academy



PentesterAcademy | a SecurityTube.net initiative

Revolutionizing Infosec Training

Highly Technical, Hands-on, Affordable

Latest Videos

New content added weekly!

Dumping Passwords from Browser Memory

- Identify strings or data structures around the "secret information"
- Verify multiple times by varying conditions
  - Different versions of the Browser
  - Different OS
  - Different versions of the OS
  - Across Reboots (might not matter too much)
- Create search strings or regex for the pattern you identify
- Welcome to the world on Memory Analysis!
  - Volatility
  - Mimikatz etc.

Memory Dumping and Analysis

- Memory Dumping
  - Per Process
  - Full System
- Why Memory Dumping?
  - Treasure trove of information!
    - Passwords, Keys, Secrets, etc.
  - No need to run multiple commands!
    - Limit running privileges commands on the system
- Requires Admin Privileges for system wide dumping

Post/windows/gather/enum_prefetch

```
msf exploit(handler) > use post/windows/gather/enum_prefetch
msf post(enum_prefetch) > set SESSION 1
SESSION => 1
msf post(enum_prefetch) > run

[*] Prefetch Gathering started.
[+] Detected Windows 7 (Build 7600). (max 128 entries)
[+] EnablePrefetcher Value: (3) = Applaunch and boot enabled (Default Value, excl. Win2K3).
[+] Remote: Timezone is Pacific Standard Time.
[+] Remote: Localtime bias to UTC: -480 minutes.
[+] Current UTC Time: 2014-12-14 17:11:01 UTC
[*] Gathering information from remote system. This will take awhile..

Prefetch Information
====================

Last execution (filetime)  Run Count  Hash  Filename        Filepath
-------------------------  ---------  ----  --------        --------
[illegible]
```

Prefetch Settings

Configuration [edit]

The Prefetcher's configuration is stored in the Windows Registry at HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters. The EnablePrefetcher value can set to be one of the following:[5]

- 0 = Disabled
- 1 = Application prefetching enabled
- 2 = Boot prefetching enabled (default on Windows 2003 only)[6]
- 3 = Application and Boot prefetching enabled (default)[5]

The recommended value is 3.[5] Values higher than 3 do not increase performance, and changing the value to 2 will not make Windows boot faster.[5]

http://en.wikipedia.org/wiki/Prefetcher