Journeyman-Windows Capabilities Developer Exam Task List

> This page lists the Master Training Task List (MTTL) items that are associated with the Journeyman Windows Capabilities Developer Evaluation.

MTTL References By Exam Module

User Mode – Module #1:

Developers will demonstrate the ability to understand, use, and modify the Windows registry. (IV 1 C)

Developers will be able to create, load, and use Dynamic Link Libraries (DLLs). (IV 2 I)

Developers will be able to create, access, and manipulate keys in the registry using Win32 and Native APIs. (IV 2 M)

User Mode – Module #2:

Developers will demonstrate the use of the Native Application Programming Interface (API) function calls for Windows Systems programming. (IV 2 A)

Developers will use the Win32 Application Programming Interface (API) function calls for Windows Systems programming. (IV 2 B)

Developers will be able to create, load, and use Dynamic Link Libraries (DLLs). (IV 2 I)

Developers will be able to create, access, and manipulate keys in the registry using Win32 and Native APIs. (IV 2 M)

Driver Module:

Developers will demonstrate their ability to load and unload drivers from user and kernel space. (IV 4 B)

Developers will be able to setup and perform Debugging of a Windows Driver. (IV 4 S)

Developers will demonstrate the ability to write kernel code to handle interrupts (software/hardware). (IV 4 U)

Developers will demonstrate the ability to use Windbg to debug kernel code and perform Crash Dump Analysis. (IV 4 W)

VR/RE Module:

Developers will be able to Disassemble Binary using command line tools and Static Analysis. (III 6 E)

Developers will apply Static Analysis concepts. (III 6 F)

Developers will perform Dynamic Analysis concepts. (III 6 G)

Developers will combine Dynamic and Static Analysis to reverse binaries. (III 6 H)

Developers will demonstrate the ability to identify common vulnerabilities in Source code. (V 8 A)

Developers will demonstrate the ability to implement Basic Fuzzing Algorithms. (V 8 B)

Developers will demonstrate the ability to implement Shell Code Generation. (V 8 Q)