kaspersky
expert training

Advanced malware analysis techniques

Course
program

№	Track	What you will learn/practice	Lesson	Practice
0	Course overview	• About your trainer • Course roadmap • Course structure	Course overview	—
			Virtual lab introduction	—
1	Intro	• Routine IDA Pro tasks: navigation, functions, code and data manipulation • Advanced features of IDA Pro: structure types, fields, shifted structure pointers • Code and data flow analysis • Stack arithmetics	Intro: Mission briefing	Intro: exercise 1
			Intro: solution for exercise 1. Next steps	Intro: exercise 2
			Intro: solution for exercise 2. Pointer into the middle of a structure	Intro: exercise 3
			Intro: solution for exercise 3. Stack frame and stack pointer	Intro: exercise 4
			Intro: solution for exercise 4. Further analysis	Intro: exercise 5
2	Shell	• Code and data flow analysis • Stack mechanics and data layout • Manual reconstruction of data structures	Shell: mission briefing	Shell: exercise 1
			Shell: solution for exercise 1. Further analysis	Shell: exercise 2
			Shell: solution for exercise 2. Conclusion	Shell: exercise 3
3	Msfvenom	• Analyzing PowerShell scripts • Decoding Msfvenom (Metasploit) payloads • Manual reconstruction of data structures	Msfvenom: mission briefing	Msfvenom: exercise 1
			Msfvenom: solution for exercise 1. Further steps	Msfvenom: exercise 2
			Msfvenom: Solution for exercise 2. Conclusion	—

№	Track	What you will learn/practice	Lesson	Practice
4	Bangladesh GPCA	• Code and data flow analysis • Recognizing a well-known encryption algorithm • Automating decryption with a decoding framework	Bangladesh GPCA: mission briefing	Bangladesh GPCA: exercise 1
			Bangladesh GPCA: solution for exercise 1. Decryption	Bangladesh GPCA: exercise 2
			Bangladesh GPCA: solution for exercise 2. Decoding framework	Bangladesh GPCA: exercise 3
			Bangladesh GPCA: solution for exercise 3. Conclusion	—
5	Regin driver	• Analyzing a homebrew crypto algorithm • Raw offset - virtual address conversions • Automating decryption of PE files	Regin driver: mission briefing	Regin driver: exercise 1
			Regin driver: solution for exercise 1. Next steps	Regin driver: exercise 2
			Regin driver: solution for exercise 2. Conclusion	—
6	Decrypt strings	• Analyzing a homebrew crypto algorithm • Automating decryption of Mach-O files • Processing multiple encrypted strings, referenced as function arguments	Decrypt strings : mission briefing	Decrypt strings: exercise 1
			Decrypt strings: solution for exercise 1. Next steps	Decrypt strings: exercise 2
			Decrypt strings: solution for exercise 2. Conclusion	—
7	Driver	• Processing encrypted strings, preparing the sample for the analysis Applying structures, enumerations • Re-creating a C++ class/structure • In-depth reverse engineering of a sample	Driver: mission briefing	Driver: exercise 1
			Driver: solution for exercise 1. Next steps	Driver: exercise 2
			Driver: solution for exercise 2. Next steps	Driver: exercise 3
			Driver: solution for exercise 3	—

№	Track	What you will learn/practice	Lesson	Practice
			Driver: next steps	Driver: exercise 4
			Driver: solution for exercise 4. Next steps	Driver: exercise 5
			Driver: solution for exercise 5. Next steps. Part 1	—
			Driver: next steps. Part 2	—
			Driver: next steps . Part 3	Driver: exercise 6
			Driver: solution for exercise 6. Conclusion	—
8	Miniduke	• Processing a custom assembly-coded shellcode • Extracting opcode information without a disassembler • Reconstructing a custom API hashing algorithm • Exporting information to IDA via an IDC script	Miniduke: mission briefing	Miniduke: exercise 1
			Miniduke: solution for exercise 1. Next steps	Miniduke: exercise 2
			Miniduke: solution for exercise 2. Next steps	Miniduke: exercise 3
			Miniduke: solution for exercise 3. Conclusion	—
9	Rocra	• Extracting a binary payload from the RTF document • Analyzing an exploit's shellcode payload • Extracting the final payload from the document	Rocra: mission briefing	Rocra: exercise 1
			Rocra: solution for exercise 1. Next steps	Rocra: exercise 2
			Rocra: solution for exercise 2. Next steps	Rocra: exercise 3
			Rocra: solution for exercise 3. Conclusion	—
10	Cobalt	• Using oletools to inspect an OLE2 container	Cobalt: mission briefing	Cobalt: exercise 1
			Cobalt: solution for exercise 1. Conclusion	—

№	Track	What you will learn/practice	Lesson	Practice
11	Cloud Atlas	• Extracting binary data from a crafted RTF document • Using oletools to inspect an OLE2 container • Analyzing binary and scriptable (VBS) payloads	Cloud Atlas: mission briefing	Cloud Atlas: exercise 1
			Cloud Atlas: solution for exercise 1. Next steps	Cloud Atlas: exercise 2
			Cloud Atlas: solution for exercise 2. Next steps	Cloud Atlas: exercise 3
			Cloud Atlas: solution for exercise 3. Next steps	—
12	Miniduke PDF	• Analyzing a malicious PDF document • Inspecting a ROP-building Javascript • Reconstructing a ROP chain	Miniduke PDF: mission briefing	Miniduke PDF: exercise 1
			Miniduke PDF: solution for exercise 1. Next steps	Miniduke PDF: exercise 2
			Miniduke PDF: solution for exercise 2. Conclusion	—
13	Ragua Py2exe	• Extracting a py2exe binary • Decompiling Python bytecode	Ragua Py2exe: mission briefing	Ragua Py2exe: exercise 1
			Ragua Py2exe: solution for exercise 1. Conclusion	—
14	Cridex	• Dynamically unpacking / decrypting • Windows executables	Cridex: mission briefing	Cridex: exercise 1
			Cridex: solution for exercise 1. Conclusion	—
15	Carbanak	• Analyzing and dynamically unpacking / decrypting Windows .NET executables	Carbanak: mission briefing	Carbanak: exercise 1
			Carbanak: solution for exercise 1. Conclusion	—
16	Snake	• Analyzing Golang samples • Mapping basic Golang structures • Extracting and decrypting Golang string literals	Snake: mission briefing	Snake: exercise 1
			Snake: solution for exercise 1. Conclusion	—
			Course summary	—

Thank you!

kaspersky.com

Discord server: kas.pr/g2j8

Help page: kas.pr/ii9f

kaspersky