

OSD — Windows User Mode Exploit Development Preparation Training

Delivered from Hacky Security's own curriculum, by instructors who also work our security operations centre. Classroom, online live, or at your office.

EXPERT

At a glance

COURSE	OSD — Windows User Mode Exploit Development	WHAT IS INCLUDED ✓ Instructor-led delivery by practitioners who also work our SOC ✓ Hands-on lab access for the duration of the course ✓ Course materials and lab guides ✓ Hacky Security certificate of completion ✓ Post-course support by email
TRACK	OffSec	
DOMAIN	Cyber Security	
LEVEL	Expert	
DURATION	On request	
DELIVERY	Classroom · Online live · On-site at your office	
LANGUAGE	English	
CLASS SIZE	Small cohorts — everyone gets lab time	

Who this is for

This is an expert course. It suits people already working in or moving into Cyber Security who want structured, hands-on preparation rather than a video library. If you are unsure whether the level fits your team, send us the job titles and we will tell you honestly — including if a different course would serve you better.

Course outline

About this course map

OSD is the exploit-development exam — the module where you stop using other people's exploits and start writing your own. It is the "-300" course EXP-301 (Windows User Mode Exploit Development) and one of the three certifications that make up OffSec's OSCE³. It answers a question penetration testing never quite reaches: when there is no public exploit for the vulnerability in front of you, can you build one?

This is the most low-level course in the OffSec catalogue. It lives in the debugger, in assembly, and in the details of how Windows lays out memory and defends it. The preparation programme assumes you are comfortable in C and can read x86 assembly, or are willing to get there fast, because everything after the first week depends on it.

The exam is 48 hours. You are given vulnerable applications and you write working exploits for them, defeating the mitigations Windows puts in the way. Then 24 hours to document exactly how each exploit works.

Module 01 — Tools of the trade

The environment exploit development actually happens in, and getting fluent before the hard part.

- WinDbg and the debugger workflow: breakpoints, memory inspection, stepping
- Reading x86 assembly well enough to follow a program's real behaviour
- The Windows memory model: the stack, the heap, and how a process is laid out
- Setting up a development lab that mirrors the target

Module 02 — Stack buffer overflows, properly

The overflow OSCP treats as a checkbox, taken apart to its foundations.

- How the stack works during a function call, frame by frame
- Overwriting the saved return address and taking control of execution
- Finding the offset, controlling EIP, and locating your buffer in memory
- Bad characters: finding them, and why one missed byte breaks everything

Module 03 — Shellcode and payload construction

Getting from "I control execution" to "my code runs."

- What shellcode is at the byte level, and why you cannot always use a canned one
- Encoding around bad characters and space constraints
- Socket reuse and staging when the payload will not fit
- Writing small custom shellcode when you have to

Module 04 — Structured Exception Handler overflows

An older but exam-relevant technique, and the reasoning behind it.

- How Windows exception handling works, and how an overflow abuses it
- Overwriting the SEH chain and redirecting execution through it
- POP POP RET and finding a usable gadget
- When SEH is the path and when it is a dead end

Module 05 — Defeating DEP with ROP

Data Execution Prevention stops your shellcode running on the stack. Return-Oriented Programming is how you get around it — and it is the intellectual core of the course.

- Why DEP breaks the classic overflow, in mechanical detail
- Return-Oriented Programming: chaining existing code fragments into a program of your own
- Finding gadgets, and building a chain that calls the right Windows function
- Using VirtualProtect or VirtualAlloc to make your shellcode executable anyway

Module 06 — Bypassing ASLR

Address Space Layout Randomisation moves everything so you cannot predict where anything is. This module is how you find it anyway.

- What ASLR randomises and what it does not
- Information leaks: reading an address out of the target to defeat the randomisation
- Non-randomised modules and partial overwrites
- Combining an ASLR bypass with a ROP chain into one working exploit

Module 07 — Format string and other memory-corruption bugs

Vulnerability classes beyond the stack overflow.

- Format string vulnerabilities: reading and writing arbitrary memory
- Reversing a closed binary to find the vulnerability in the first place
- Reading disassembly to understand a function's behaviour without source

Module 08 — Reverse engineering for the bug

Finding the vulnerability yourself, which the exam requires — the applications are not pre-marked.

- Static analysis with a disassembler: mapping a program's structure
- Dynamic analysis: watching it run to find where user input reaches something dangerous
- Identifying the vulnerable function and the exact conditions that trigger it

Module 09 — Building the full exploit under exam conditions

Chaining every technique into a reliable, working exploit — which is what is graded.

- From crash to controlled execution to a stable shell
- Making the exploit reliable rather than lucky
- Documenting the exploit so someone else could reproduce it — the reporting half of the exam

Sitting the official exam

Exam voucher sold separately. We buy it from the certification body on your behalf and send you the code — you book your own exam date. This course is Hacky Security's own preparation training. The certification itself is awarded by the certification body, not by us.

Investment

Priced per cohort. Tell us how many people and which delivery mode and we will send a quote the same working day.

Book this course

Tell us how many people, where, and when — we will come back with dates and a quote within one working day.

+91-8091900310 • info@hackysecurity.com

<https://hackysecurity.com/courses/osed/>

CHANDIGARH • SHIMLA • DELHI • DUBAI • LONDON

Independent preparation training delivered from Hacky Security's own curriculum. Hacky Security Private Limited is not affiliated with or endorsed by the certification body named in this document. Certification names and codes are the trademarks of their respective owners and are used only to describe what this course prepares you for. Prices, durations and outlines are indicative and may change; the quotation we send you is the binding one. Issued 19 September 2026.