# ASL IT SECURITY

# XTREME XPLOIT DEVELOPMENT

## V 2.0

**Overview:** The most dangerous threat is the one which do not have a CVE. Until now developing reliable exploits was a skill which only a small “elite” group of hackers knew. Now with our Xtreme Xploit Development training you can develop highly technical skills of vulnerability discovery, analysis and exploit writing. This training does not teach you how to use exploits, but how to write your own reliable exploits. Learn the latest exploit development techniques.

**Course Description:** Xtreme Xploit Development is the most detailed and advanced exploit development training. Starting from assembly language basics the course moves to shellcode development, fuzzing for exploit discovery, writing reliable exploit, bypassing security measures and anti-virus evasions. The training is developed and delivered by accomplished exploit writers and reverse engineers who had developed and published in various common softwares and Microsoft products. After this training exploit writing and vulnerability research will not be a dark art for you. Each exercise in this training is hands-on with live examples and test cases.

**Who Should Take This Course:**

- Red Team members, who want to pen-test custom binaries and exploit custom built applications.
- Bug Hunters, who want to find new vulnerabilities and write exploits for all the crashes they find.
- Members of military or government cyber warfare units.
- Members of reverse engineering research teams.
- Pen-testers, Security analysts, Reverse Engineers, who want to take their skills to the next level.

**Student Requirements:**

Knowledge of c/c++

Knowledge of OS

Knowledge of any scripting language like perl/python/ruby

**Trainers:**

**Abhishek Sahni** has seven years of experience in web application penetration testing. He had reported security vulnerabilities to yahoo and AOL. Abhishek had conducted successful security trainings on various topics for government agencies in India and abroad. He had performed many web application penetration tests and faced really challenging scenarios and found methods to overcome them.

**Abhishek Lyall** is an experienced penetration tester and exploit writer. He found many vulnerabilities in MS Office products and other common softwares and wrote exploits for them. He is also well versed with very advanced exploit writing techniques like Return Oriented Programming, Jit Spraying etc and egghunting. He has also found many 0 days in Government websites and reported them and assisted in patching those vulnerabilities.

## Module 1: Introduction to assembly language

1. Windows assembly
2. Linux assembly
3. Basic assembly language programs

## Module 2: Shellcoding

1. Windows shellcode
2. Linux shellcode
3. Generating shellcodes with metasploit
4. Reverse shell and bind shell
5. Download_execute
6. Complete Meterpreter
7. Using and modifying egghunter
8. SEH omelets
9. Writing own shellcode

## Module 3: Debuggers

1. Windbg
2. Ollybdg
3. Immunity Debugger

4. Gdb
5. Useful Debugger plugins

**Module 4: Introduction to exploits**

1. Places to find public exploits
2. Exploits usage
3. Introduction to metasploit exploits

**Module 5: Vulnerability Discovery**

1. Introduction to fuzzing and fault injection
2. Protocol fuzzing
3. File format fuzzing
4. Fuzzing frameworks (peach, sully)
5. Writing fuzzers
6. Writing metasploit fuzzers
7. Crash dump analysis

**Module 6: Exploiting Buffer overflows**

1. Writing stack based overflows exploits
2. SEH based buffer overflow

3. Bypassing SafeSEH
4. Exploiting heap overflows
5. Developing browser and PDF exploits
6. Heap spraying
7. Jit Spraying
8. Writing Integer overflow exploits
9. Writing exploits for Metasploit
10. Porting existing exploits to Metasploit
11. Debugger plugins for automated exploit generation

**Module 7: Bypassing protections**

1. Defeating DEP using Ret2libC
2. Return Oriented Programming
3. Finding ROP gadgets using debugger plugins
4. Bypassing ASLR
5. Bypassing EMET
6. Exploiting unicode overflows
7. Writing venetian Shellcode
8. Alphanumeric Shellcode

### Module 8:  Making Exploits more reliable

1. Choosing return addresses
2. Identifying bad characters

### Module 9: Other exploits

1. Java applet infection
2. DLL hijacking
3. Logical flaws

### Module 10: Reverse engineering for exploit creation

1. Binary diffing and patch diffing
2. Finding malware samples from wild
3. Finding and replacing JS in pdf malwares
4. PDF analysis tools
5. Finding and replacing shellcode in malwares
6. Finding embedded executables
7. Replacing embedded executables
8. Evading antivirus signatures

## Module 11: Post exploitation

1. Different kind of shellcodes
2. Meterpreter as a complete backdoor
3. Advanced RATS
4. Other payloads