

Table of Contents

Table of Contents	2
Introduction	4
1. VM'S	5
Introduction	5
Registry Checks:	5
Practical Example	6
Network Adapter MAC Address:	8
How it Works:	9
Why MAC Address Detection?	9
System Information:	9
RAM Detection Logic	10
Explanation of the Code:	10
Why Use RAM Detection?	10
Potential Limitations	11
Provider Detection:	11
Provider Detection Code	11
How the Code Works:	12
Why Use Provider Detection?	12
Potential Limitations:	12
Shared Folders:	12
Shared Folder Detection Code	13
How the Code Works:	14
Why Use Shared Folder Detection?	14
Potential Limitations:	14
System Calls:	14
Common System Calls for VM Detection	14
Practical Example: Using CPUID to Detect VMs	15
How the Code Works:	16
Other System Calls for VM Detection:	16
Why Use System Calls for VM Detection?	17
Limitations:	17
2. Debuggers	17
Introduction	17
Detecting Software Breakpoints:	18
Detecting Hardware Breakpoints:	19
Explanation of the Method:	20
Why Detecting Hardware Breakpoints is Important	20
Is Debugger Present:	20
Explanation of the Method:	20
Importance of the debuggerPresent Method:	21

Anti-Debug Breakpoint Method:	21
Explanation of the Method:	21
Anti-Debug UI Remote Breakin Method:	22
isDebuggerPresentInRemoteProcess	23
Extensions	24
1. Enhanced Registry Checks	24
2. CPU Feature Detection	25
3. Memory and System Resource Monitoring	25
4. Hardware and Peripheral Monitoring	25
5. Behavioral and Timing Analysis	26
6. Advanced Debugger Detection	26
7. Anti-Analysis Techniques	27
8. Shared Resource Detection	27
9. Hardware Fingerprinting	28
10. Windows API Abnormalities	28
Conclusions	29