

Forum:[Hack Forums](#)

Thread: [Static and Dynamic Analysis Evasion \(2025\)](#)

skyt__ - October 19, 2025, 1:20 pm

Static and Dynamic Analysis Evasion (2025)

Overview

Anti viruses and sandboxes statically and dynamically assess assemblies to determine how dangerous they are. However, in a dynamic environment such as a sandbox, if the program detects its being analyzed and stops the execution of malicious code, it will appear to be a safe assembly. In this brief series, we will explore methods to detect if a program is in a sandbox environment or in a virtual machine.

1. Anti Sandbox

There are many ways to detect if the program is inside of a sandbox. One of these ways is by getting the module handle of commonly loaded dlls in sandboxes.

Here is an example of how to check for these dlls in c++

```
Code std::vector<std::string> dlls = {
    "avghookx.dll", // Avast/AVG
    "avghooka.dll", // Avast/AVG
    "snxhk.dll",    // Sandboxie
    "sf2.dll",      // Jujubox
    "sbiedll.dll",  // Sandboxie
    "dbghelp.dll",  // Debugging tools
    "api_log.dll",  // Jujubox
    "dir_watch.dll", // Jujubox
    "pstorec.dll",  // Zenbox
    "vmcheck.dll",  // VMware / VM-detection
    "wpespy.dll",   // WPE Pro
    "cmdvrt64.dll", // Comodo
    "cmdvrt32.dll", // Comodo
    "sxin.dll"      // Jujubox
};

bool detect_sandbox() {
    for (const auto& dll : dlls)
    {
        if (GetModuleHandleA(dll.c_str()) != nullptr)
            return true;
    }

    return false;
}
```

2. Anti Debug

Checking if a program is debugging your code can be rather complex or straightforward.

One of the most simple ways to check for a debugger is by using the default debugapi function

CheckRemoteDebuggerPresent.

You can call it in c++ like this

```
Code bool detect_debugger() {  
    BOOL is_debugger_present = FALSE;  
  
    if (CheckRemoteDebuggerPresent(GetCurrentProcess(), &is_debugger_present )) {  
        return true;  
    }  
  
    return false;  
}
```

You can also check for kernel debuggers in c++ by doing the following

```
Code bool detect_kernel_debugger() {  
    const ULONG_PTR user_shared_data = 0x7FFE0000;  
  
    const UCHAR kd_debugger_enabled_byte = *(UCHAR*)(user_shared_data + 0x2D4); // offset of  
the field  
  
    const bool kd_debugger_enabled = (kd_debugger_enabled_byte & 0x1) == 0x1;  
    const bool kd_debugger_not_present = (kd_debugger_enabled_byte & 0x2) == 0;  
  
    if (kd_debugger_enabled || !kd_debugger_not_present)  
        return true;  
  
    return false;  
}
```

3. Anti Virtual Machine

The methods for detecting virtual machines can vary. Instead of explaining thoroughly the methods that can be used, I will recommend a github repository under the name

"VMAware."

VMAware is a program for detecting if the program is present in a virtual machine or a real computer. It consists of one C header file that you can include to detect for virtual machines. You can use the function "VM::detect()" to run all the detection methods for a virtual machine and return the result.

Well Thanks to the guy with the sniper as his username for the inspiration.

ClickAds - October 19, 2025, 5:02 pm

Informative details from Member ?

Bastille - October 20, 2025, 9:53 am

Nice introduction to the topic, hope to see your threads improving over time.

Thanks [???????????](#) for inspiring other members ("the guy with the sniper as his username" as he

says)

???????????? - October 24, 2025, 11:53 am

(Oct 20, 2025 09:53 AM) Bastille Wrote: [\[CITE\]](#) Nice introduction to the topic, hope to see your threads improving over time.

Thanks [????????????](#) for inspiring other members ("the guy with the sniper as his username" as he says)

I am happy to see people posting quality things. :)