

The Microsoft Distributed Transaction Coordinator (MSDTC) service loads the file `mtxoci.dll` at when it starts. This file automatically loads (i.e., without the user's consent) `oci.dll` if it is present. Since `oci.dll` is only present when Oracle products are installed, we have the ability to write our own version of `oci.dll` to execute code of our choosing in the context of MSDTC. If `oci.dll` is present in this folder or `System32`, overwriting requires taking ownership from `TRUSTED_INSTALLER` and reconfiguring accesses for Administrator. Additionally, because MSDTC is a Windows service, we can maintain persistence during reboots.

This "exploit" has a series of caveats to consider:

1. When Hikiti is typically deployed, it uses the "StickyKeys" vulnerability to get Administrator access to the computer in question. In testing, Administrator access is required for all steps (e.g., copying `oci.dll` to `C:\Windows\system32\wbem\` and configuring MSDTC)
2. In testing, `oci.dll` would cause the system to hang while it attempts to start MSDTC if we try to open a dialog box. The cause of this is not clear.
3. The `C:\Windows\System32\wbem` folder does not appear to be in the DLL load order path in Windows 8.1 x64. We found that copying `oci.dll` to `C:\Windows\System32` does allow the code to run.

The code was tested to work in Windows 7 x32, Windows 7 x64, and Windows 8.1 x64.

The correct procedure for running this POC listed below.

1. Copy `oci.dll` to `System32\wbem` (or `System32` for Windows 8 and later)
2. Copy `config.bat` to a place of your choosing.
3. Open an Administrator Command Prompt and launch `config.bat`

Optionally, to see proof that the DLL is getting executed, you may open `Dbgview.exe` from Sysinternals and see output as the library attaches and detaches from threads and processes respectively.