

Raytheon

Blackbird Technologies

SIRIUS Pique Proof-of-Concept Delivery

User-Mode DKOM

Final PoC Report

For

SIRIUS Task Order PIQUE

Submitted to:
U.S. Government

Submitted by:
Raytheon Blackbird Technologies, Inc.
13900 Lincoln Park Drive
Suite 400
Herndon, VA 20171

26 January 2015

This document includes data that shall not be disclosed outside the Government and shall not be duplicated, used, or disclosed—in whole or in part—for any purpose other than to evaluate this concept. If, however, a contract is awarded to Blackbird as a result of—or in connection with—the submission of these data, the Government shall have the right to duplicate, use, or disclose the data to the extent provided in the resulting contract. This restriction does not limit the Government's right to use information contained in these data if they are obtained from another source without restriction.

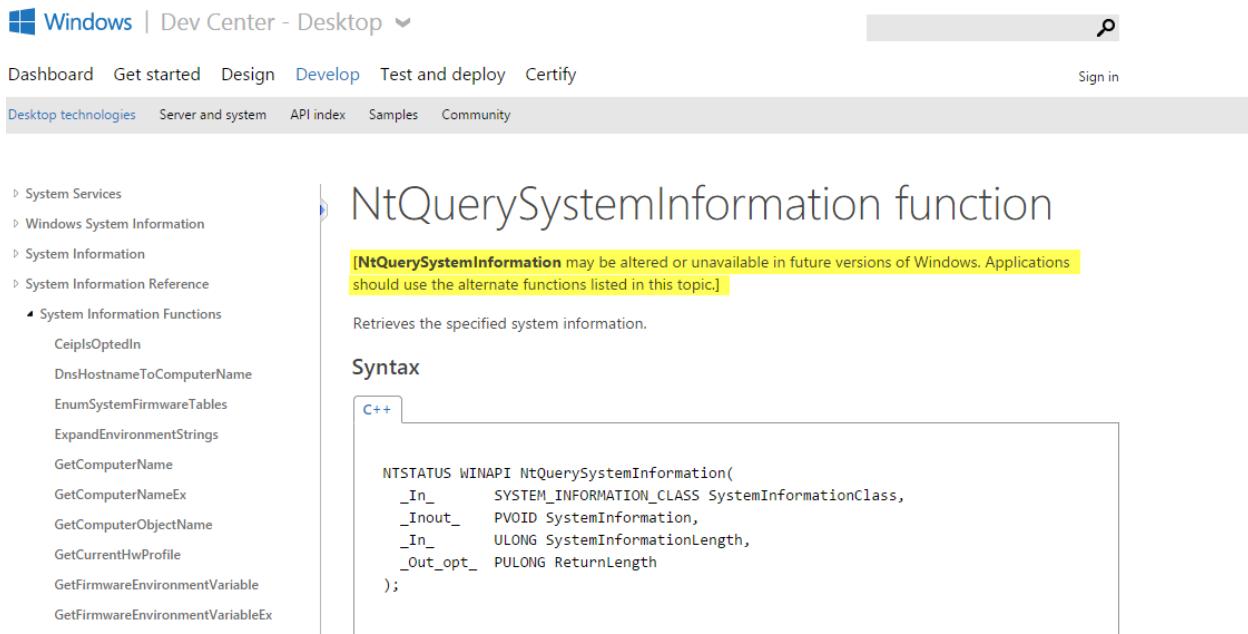
This document contains commercial or financial information, or trade secrets, of Raytheon Blackbird Technologies, Inc. that are confidential and exempt from disclosure to the public under the Freedom of Information Act, 5 U.S.C. 552(b)(4), and unlawful disclosure thereof is a violation of the Trade Secrets Act, 18 U.S.C. 1905. Public disclosure of any such information or trade secrets shall not be made without the prior written permission of Raytheon Blackbird Technologies, Inc.

(U) Table of Contents

(U) Executive Summary.....	3
(U) Current Status.....	3
(U) Next Steps.....	3

(U) Executive Summary

(U) Upon further research into using `NtQuerySystemInformation()` to obtain the NT KernelBase Image address and ultimately the address of the Kernel Processor Control Region (KPCR) and subsequent bypassing ASLR to modify kernel-based pointers to effect process hiding, we have concluded this approach is no longer available for Windows 8.0 and later. Beginning with Windows 8.0, Microsoft no longer allows the use of `NtQuerySystemInformation()` and its replacement API does not support obtaining NT KernelBase Image address, which is crucial to implementing user-mode DKOM. Figure 1 shows Microsoft's warning that `NtQuerySystemInformation` should not be used because it "may be altered or unavailable in future versions of Windows."



The screenshot shows a Microsoft Dev Center page for the Windows API. The left sidebar lists various system information functions. The main content area is titled "NtQuerySystemInformation function". A yellow box contains the deprecation warning: "[`NtQuerySystemInformation` may be altered or unavailable in future versions of Windows. Applications should use the alternate functions listed in this topic.]". Below the warning, the function's purpose is described: "Retrieves the specified system information." A "Syntax" section shows the C++ code for the function:

```
NTSTATUS WINAPI NtQuerySystemInformation(
    _In_      SYSTEM_INFORMATION_CLASS SystemInformationClass,
    _Inout_    PVOID SystemInformation,
    _In_      ULONG SystemInformationLength,
    _Out_opt_  PULONG ReturnLength
);
```

Figure 1. Microsoft Notification of Potential Deprecation

(U) The independent blog site, <http://www.exploit-monday.com/>, has updated a June 2013 blog post on NtQuerySystemInformation() noting that the symbols available in NtQuerySystemInformation() and subsequently contained in uxtheme.dll (64-bit) and combase.dll (32-bit) have been removed and unavailable altogether beginning with Windows 8.0.

Undocumented NtQuerySystemInformation Structures (Updated for Windows 8)

Those familiar with Windows internals are likely to have used the NtQuerySystemInformation function in ntdll. This function is extremely valuable for getting system information that would otherwise not be made available via the Win32 API. The MSDN documentation only documents a minimal subset of the structures returned by this powerful function, however. To date, one of the best references for the undocumented features of this function has been the "Windows NT/2000 Native API Reference." Despite being published in 2000, many of the structures documented in this book are still relevant today. In recent history though, Microsoft has quietly expanded the number of functions returned by NtQuerySystemInformation. Thankfully, the vast majority of them have been made public via symbols present in uxtheme.dll (64-bit structures) and combase.dll (32-bit) structures in Windows 8. At last check, it appears as though Microsoft pulled these symbols from the latest versions of the respective dlls.

@mattifestation

CODE

PowerSploit on GitHub
Window Shellcode in C
Memory-Tools.ps1
Replace-x64-Process.ps1

JOURNEY BACK IN TIME

► 2014 (4)
▼ 2013 (11)

Figure 2. Independent Blogger Confirms Deprecation in Windows 8.0

(U) Current Status

(U) We have the skeleton user-mode DKOM application written and compiled (the current version Microsoft Visual Studio 2013 solution was attached to the January 23, 2015 Interim Report II – PIK_DKOM.sln).

Note: we've written custom _vsprint, memset, and DBGPRINT routines in order to run tests on Windows XP SP2 and earlier to preclude having to pull in the CRT.

(U) Next Steps

(U) Now that NtQuerySystemInformation() has been decremented, the scope of developing the user-mode DKOM exceeds that of a PoC. The development of a user-mode DKOM capability will likely require detailed research into Windows kernel structures and finding an undocumented method for obtaining the KernelBase and KPCR. We recommend the project be allocated, but outside the context of a PoC development due to technical difficulty and anticipated scope of effort.