

EEYE: Windows VDM TIB Local Privilege Escalation

Source: <http://www.derkeiler.com/Mailing-Lists/NT-Bugtraq/2004-04/0012.html>

From: Marc Maiffret (*mmaiffret_at_EEYE.COM*)

Date: 04/13/04

Date: Tue, 13 Apr 2004 13:06:48 -0700

To: NTBUGTRAQ@LISTSERV.NTBUGTRAQ.COM

Windows VDM TIB Local Privilege Escalation

Release Date:

April 13, 2004

Date Reported:

February 9, 2004

Severity:

Medium (Local Privilege Escalation to Kernel)

Vendor:

Microsoft

Systems Affected:

Windows NT 4.0

Windows 2000

Description:

eEye Digital Security has discovered a second local privilege escalation vulnerability in the Windows kernel that would allow any user capable of executing code to elevate that code to the highest possible local privilege level (kernel). For instance, a malicious user with legitimate access to a machine, or a remote attacker or worm payload able to obtain unprivileged access through an unrelated exploit, could use this vulnerability to wholly compromise a Windows NT 4.0 or Windows 2000 system.

The problem lies in a certain area of the Windows kernel that supports 16-bit code executing in a Virtual DOS Machine (VDM). By causing the processor to execute code in Virtual86 (essentially "16-bit emulation") mode without first initializing a VDM for the process, specific routines in the Windows 2000 kernel code may be caused to dereference a null pointer, which actually functions as a pointer to attacker-controlled data if memory is allocated at virtual address 0. (On Windows NT 4.0,

the pointer can be controlled directly by the user.) Other pointers and fields at offsets from the VDM data address may then be supplied with specially–crafted data, in order to write to arbitrary locations in kernel memory.

Technical Description:

A Virtual DOS Machine is simply a collection of data structures that, among other things, instructs the kernel how to behave when an exception occurs within Virtual 8086–mode code. Typically, the state of the VDM and V86–mode code execution is handled using the NtVdmControl() API exported by NTDLL.DLL, but it is sufficient to call NtContinue() with a CONTEXT structure that properly supplies CS:EIP, SS:ESP, and EFLAGS with the Virtual–8086 Mode flag (bit 17) set, in order to switch the calling thread into Virtual 8086 mode and bypass VDM initialization entirely.

When a thread is executing in V86 mode and something bad happens, the first thing most kernel fault handlers do is to check for the V86 flag in the EFLAGS stored on the stack, dealing with the exception differently based on whether or not it's set. For instance, KiTrap0D (the General Protection Fault handler) emulates the behavior of certain IOPL–restricted privileged instructions if they occur in V86 mode (e.g., POPF is considered a privileged instruction during V86–mode execution). In some cases, it attempts to consult VDM information for the current process — on Windows 2000, by first dereferencing the "VdmObjects" field of the current thread's associated EPROCESS structure ([[[FFDFF124h]+44h]+1DCh]) and then using other pointers and data relative to that address.

As mentioned above, however, the "VdmObjects" pointer on Windows 2000 will be 0 if NtVdmControl() has not yet been used to initialize it. Of course, because V86–mode code needs the low end of memory to be addressable, 0 is in fact a perfectly valid base for a chunk of virtual memory, provided that ZwAllocateVirtualMemory() is called with a base address of 1..(4KB–1) to allocate it. So, if a region of memory is allocated at 0 prior to causing a V86–mode fault, then the kernel will attempt to access user–controlled memory, which it treats with as much trust and lack of validation as a kernel–controlled data structure. Yes, sometimes even null pointers are exploitable.

Among other things, the "VdmObjects" data structure features a pointer to a "VDM TIB" data area ([[[FFDFF124h]+44h]+1DCh]+98h) on Windows 2000) that contains CONTEXT structures which the kernel routine VdmSwapContexts() references in certain circumstances. (On Windows NT 4.0, this pointer is in the user–land TIB at offset F18h and is therefore naturally under user control.) The "VDM TIB" pointer is not validated during the interesting portion of KiTrap0D, so it can point to an arbitrary address in user or kernel memory. This can allow all sorts of bad things to happen. Continuing the GPF example from above, a POPFD instruction (for instance) encountered during V86–mode execution will cause the effective context at the time of the fault to be stored at offset +CD0h within the data area (+AD0h for Windows NT 4.0), then the

NT–Bugtraq: EEYE: Windows VDM TIB Local Privilege Escalation

context at offset +A04h is retrieved for the purpose of restoring when KiTrap0D exits. The selector values in this latter context are sanitized in order to have CPL/DPL=3, but it doesn't really matter because the context stored at offset +CD0h can be written to any location in user or kernel memory, including the IDT or a process's LDT.

Of course, writing arbitrary data to an arbitrary location in kernel memory is the last thing that happens to one's machine before it officially becomes the attacker's machine, so the only thing left to talk about is what an attacker can do with unfettered kernel–level access on a system. For more information on that subject, please visit www.rootkit.com.

Protection:

Retina Network Security Scanner has been updated to identify this vulnerability.

Vendor Status:

Microsoft has released a patch for this vulnerability. The patch is available at:

<http://www.microsoft.com/technet/security/bulletin/MS04–011.msp>.

Credit:

Derek Soeder

Related Links:

Retina Network Security Scanner – Free 15 Day Trial

<http://www.eeye.com/html/Products/Retina/download.html>

Greetings:

VI*500; LWV, GW, and JMM; Ralf Brown and Potemkin's Hackers Group; FTM (can't wait for the next album!!); and Victor (who got away), Lucky (who stayed), and Lynda (who has cared for them all). =) The House of Quality, the House of Information Technology, and the Research Projects.

Copyright (c) 1998–2004 eEye Digital Security

Permission is hereby granted for the redistribution of this alert electronically. It is not to be edited in any way without express consent of eEye. If you wish to reprint the whole or any part of this alert in any other medium excluding electronic medium, please email alert@eEye.com for permission.

Disclaimer

The information within this paper may change without notice. Use of this information constitutes acceptance for use in an AS IS condition. There are no warranties, implied or express, with regard to this information. In no event shall the author be liable for any direct or indirect damages whatsoever arising out of or in connection with the use or spread of this information. Any use of this information is at the user's own risk.

NT-Bugtraq: EEYE: Windows VDM TIB Local Privilege Escalation

Feedback

Please send suggestions, updates, and comments to:

eEye Digital Security
<http://www.eEye.com>
info@eEye.com

NTBugtraq Editor's Note:

Wondering how to unsubscribe from NTBugtraq? Just send a message to Listserv@listserv.ntbugtraq.com with unsubscribe ntbugtraq in the message body, you don't need a subject line. If it says you aren't subscribed, you've either subscribed with a different email address or your address has changed somehow. Just email Russ.Cooper@rc.on.ca and I'll remove you.
