

Search for



Advanced Search

- OEM Connect Program
- Sales and Marketing
- Technical
- Strategic Guidance
- Product Guide

Worldwide Sites

**Where are you within OEM Connect?**

- Sales and Marketing
- Technical
- Strategic Guidance
- Product Guide
- General

Learn more about this and other improvements to the OEM Connect website!

[Strategic Guidance > Road Map](#)

**Camouflaged Code: Hiding Code to Protect It**

By Robert L. Scheier  
June 2003

Last year, **software pirates stole at least \$13 billion USD worth of code worldwide** ranging from operating systems to office applications and games. The size of the loss is both staggering and far reaching.

Software piracy robs OEMs of revenue they could otherwise earn bundling software with their hardware. It also costs them sales to vendors who market "bare bones" systems, where unscrupulous users load unauthorized copies of operating systems and applications after they get their hands on the hardware.

**The Hidden Costs of Software Piracy**

- ▶ Software vendors lost \$13 billion in sales during 2002
- ▶ Hardware OEMs are denied software sales that otherwise could be bundled with hardware
- ▶ Because of the bugs in hacked software, OEMs, solution providers and customers spend more to support it
- ▶ Countries where piracy runs rampant lose out on tax revenues and jobs

Sources: Business Software Alliance, IDC, Microsoft Corp.

For these reasons and more, Microsoft is developing and deploying techniques designed to make **software harder to steal or to hack**. These techniques go under the general name "code obfuscation" because they involve disguising code to make it harder for hackers to identify which parts of a program to attack. Or even to tell when they've succeeded in hacking the software.

"Obviously, if somebody is loading bare bones computers with unauthorized operating system software or desktop applications and selling those computers, they're competing unfairly with OEMs that are playing by the rules," says Bob Kruger, vice president for enforcement at the Business Software Alliance (BSA), a Washington, D.C.-based trade organization devoted to reducing software piracy. Piracy also **drives up support costs** for vendors because cracked or modified software tends to be less stable than legitimate code.

The impact of piracy is even greater when considering the role software development and other IT services play in global economies. An April 2003 study by research firm IDC for the BSA found that countries with the lowest piracy rates "enjoy larger IT sectors accompanied by greater tax bases, more jobs and other economic benefits" than other countries where piracy is common. The reason: **legitimate customers pay taxes** on their software rather than steal it.

**Obfuscation Defined**

To avoid these impacts, a growing number of software

developers are adopting code obfuscation techniques. Code obfuscation means "taking existing code and putting it through a series of transformations that make it **very difficult for hackers to understand**," explains Mariusz Jakubowski of Microsoft Research.

One candidate for obfuscation, he says, is the code that checks to see whether a game CD has the proper modification to the physical tracks on its surface to prove it is an original disk rather than a copy. Without obfuscation, "a hacker could look for that check, get rid of it, and make the CD run" even if it were an illegal copy, says Jakubowski.

One popular obfuscation technique is to encrypt the bytes that make up an application, decrypting them only when the code is run. In this example, "you don't care much about the cryptographic strength," says Jakubowski. "**The only purpose is to hide the code**, at least temporarily, before it runs. When it is finished running, you encrypt it again."

While this method won't stop a dedicated and skilled attacker, Jakubowski says many crackers will not be able to get past this kind of encryption, or won't be sufficiently motivated to put in the time required to do so.

---

#### Stop the Debuggers

Because **debugging tools can be used to hack into software**, Microsoft is also working on ways to detect whether a user is running a debugger to analyze and change the code in an application. Says Jakubowski: "If our software is being debugged, we can respond in some way, such as generating a message that says 'Please turn off your debugger and then restart the application.' Or more silently and less obviously, we can make the program crash."

Shining Light on Code Obfuscation	
What It Is	Disguises software code so it is harder to understand
What It Does	Makes code harder for hackers to modify or attack
Where It Works	Can be applied to machine code, source code or both

Source: Microsoft Corp.

On the source code level, Microsoft is developing ways to encode, or mask, program variables and control flow. A program variable is the string of characters that identify a value in a program. Hackers can **look for these in the code and exploit them illegitimately**.

For example, the variable "count" might be the number of times a trial version of an application can run before it must be purchased; or "arms" might be the names of different weapons carried by an airplane in a computer game. "If a hacker sees the counter going down every time he runs the program, he can see that this part of the program holds the number of runs remaining," says Jakubowski.

---

#### Symbol Mangling

Through a process known as "symbol mangling" developers can change variable and function names into meaningless strings of characters. Says Jakubowski: "If you have a variable called 'number of times remaining,' for example, you can rename it to something like 'x' or a combination of characters such as '01001000' so to the

cracker all the variables look the same."

Control flow refers to the loops, branches and conditional statements that allow an application to respond to different inputs. A loop, for example, might prompt a user for their password and return an "access denied" message if they fail to enter the proper password. Obscuring the true function of this loop would make it harder for hackers to find it and disable the password-checking function.

In some cases, obfuscation might even include **deliberately adding "spaghetti code"** in the form of unnecessary or redundant loops or jumps from one section of code to another.

Such obfuscation is particularly important in using programming environments such as Microsoft® .NET ([More Info](#)) and Java. If the true names of variables and the control flow are not obscured in these environments, it is **easy to decompile the executable code** into something strongly resembling the original source code.

To help developers prevent this, Microsoft is bundling a product called Dotfuscator Community Edition from PreEmptive Solutions Inc. ([More Info](#)) with its Visual Studio® .NET 2003 ([More Info](#)) development environment. Dotfuscator modifies the Microsoft Intermediate Language (MSIL) code generated by the Visual Studio® .NET compiler so it becomes **difficult for humans to understand**, while still providing the executable code needed for the .NET runtime. Among other features, Dotfuscator compresses the names of properties, fields, methods and classes into new names as short as one character.

Code obfuscation can also be used to protect checksums, a popular method of verifying whether code has been patched, or tampered with. A checksum is a mathematical value based on the values of bits

or bytes within a piece of code that is calculated before the code is run. When the code is actually executed, the system on which it is running recalculates the checksum.

"It's very difficult to make any changes to the bytes and still have the same checksum," explains Jakubowski. Obfuscating the checksums, or the code used to verify the checksums, **helps ensure this tamper-protection remains in force.**

If the checksums don't match -- indicating the code has been tampered with -- developers can cause the program to crash or simply shut it down. To further confuse hackers, Jakubowski says it is good practice to **separate the tamper response from the tamper detection.**

For example, causing the program to crash the day after tampering is detected makes it harder for the hacker to know if his patch worked or when he was detected.

#### Top Three Code Obfuscation Methods

1. Encrypts the code except when it is being run
2. Renames the variables (such as "user name") or control flow features such as loops and branches
3. Encrypts or encodes the checksums that are used to determine if code has been changed

Source: Microsoft Corp.

---

#### Do No Harm

Whatever obfuscation method is used, it's important that no anti-piracy technology make life harder for users, says Gartner Inc. analyst John Pescatore. "If it's done right, if it

**treats people as if they're honest** and only kicks in when it really and truly looks like they're dishonest, it will be widely used."

That's just what Microsoft aims to do. "If these techniques are properly done and well implemented, there should be **no negative impact on ISVs or OEMs**," says Jakubowski. "The only time they're visible is when someone tries to tamper with the code."

---

#### **Additional Web Resources**

[Obfuscating Smart Device Applications, April 2003](#)

How to protect source code with the PreEmptive Dotfuscator obfuscator

[License and Support Information for the Dotfuscator Tool for Visual Studio .NET 2003](#)

Web site of [PreEmptive Solutions Inc.](#), developers of Dotfuscator

[Dotfuscator Community Forum](#) on Yahoo

IDC/BSA study, "[Expanding Global Economies: The Benefits of Reducing Software Piracy](#)"

*Robert L. Scheier is a freelance writer based in Boylston, Mass. He is the former technology editor of Computerworld and previously was a senior editor at VAR Business and PC Week. In addition, he was an analyst for the Hurwitz Group, specializing in databases and middleware.*

Have a question? Want more information? Contact the writers and editors at [oemedit@microsoft.com](mailto:oemedit@microsoft.com).

© 2003 Microsoft Corporation. All rights reserved. This document is for informational purposes only and subject to change without notice. MICROSOFT MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT. The entire risk of the use or the results of the use of this document remains with the user.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Last Updated: June 23, 2003

[Contact Us](#) | [E-mail this Page](#) | [Newsletter](#) | [All Rights Reserved](#) | [Terms of Use](#)

©2003 Microsoft Corporation. All rights reserved. [Terms of Use](#) | [Privacy Statement](#)