Why use buffer overflow exploit?

Asked 12 years, 8 months ago Modified 12 years, 7 months ago

Viewed 562 times

1

I understand the concept of buffer overflow, and acknowledge it can give me the opportunity to execute my own code within a foreign executable.

My question is, cant this simply be done with easier ways ?

Say inject a DLL, and in DLLMain write your malicious code ?

Or play with the disassembly and inject assembly code into executable ?

And even if you got your malicious code working, what damage\profit can you get by the act, that you could not get by editing the disassembly by yourself ?

As far as I understand, the moment you got an executable in your hands you are the master of it, and can add\change\remove code by playing with the disassembly, why make all the effort for searching for exploits ?

Thanks, Michael.

reverse-engineering buffer-overflow exploit

Share

Improve this question

Follow

asked Apr 24, 2012 at 9:51

Michael

806 ● 13 ● 29

2 Answers

Sorted by: Highest score (default)

4

✓

Thing is, you don't generally get the victim to run your executable. So the fact that you can make it malicious is of little value.

Instead you can get the potential victim to use your input: that's why it's so interesting.

Share Improve this answer

Follow

answered Apr 24, 2012 at 9:53

cnicutar

182k ● 26 ● 376 ● 398

I may understand the profit of attacking web\servers based applications. But I still dont understand what profit I can earn by attacking a normal desktop application. Say I bought "Word 2010" and found a buffer overflow exploit in the application, what can I gain by exploiting it ? – Michael Apr 24, 2012 at 10:03

1 @MichaelEngstler Nothing. But what if you attack an application that is already running with higher privileges than you are ? – cnicutar Apr 24, 2012 at 10:10

2 Say you find a buffer overflow in Word 2010 that can be exploited from a specially crafted Word document. Now you just need to trick someone into opening your "document" instead of running your suspicious binary. – Colin Valliant Apr 29, 2012 at 0:21

2

Most of the time, this is due to the skeptical minds of users nowadays towards executables, and how they do not think that a PDF document could contain a virus. In other situations, the only way to deliver the code is through an exploit, such as a buffer/heap/stack overflow.

For example, on Apple iOS devices, the only way to download executable code is through the AppStore. All executables that come this way must be explicitly approved of by Apple. On the other hand, if the user simply visits a link to a maliciously crafted PDF document in MobileSafari, it could allow an attacker to arbitrarily execute code on the device.

This is the case with Comex's JailbreakMe.com site (both v2.0 (Star) and v3.0 (Saffron)). The site has the device load an incredibly intricate PDF file that ultimately leads to jailbreaking the device. There is no chance in the world that Apple would approve of an app that would do the same thing.

Share Improve this answer Follow

answered May 7, 2012 at 22:24

C0deH4cker

4,057 ● 1 ● 25 ● 36