

Re: C# Exceptions

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/pen-test/2006-08/msg00432.html>

- *From:* "3 shool" <3shool@xxxxxxxx>
 - *Date:* Thu, 31 Aug 2006 14:06:05 -0700
-

Ok, I've reached the next level now.

I did a runtime analysis of the exe and some important DLLs as ADMINISTRATOR and found a number of Heap Overruns in the COM (unmanaged) components used by the application. Now the scenario is that the C# application runs with privileges of the user logged-in and not like a service or SYSTEM account so is it possible to do following:

1. Obtain the same Heap Overruns from any LOW privileged user. Even if it was possible would the attacker gain any extra privilege as the application is running as the LOW privileged user.
2. Can these attacks be sent over the network when the application communicates with the Internet services?
3. Any other attacks that are possible here?

Thnx.

On 8/28/06, 3 shool <3shool@xxxxxxxx> wrote:

Thankx for your replies.

My comments in capitals below.

- > In a (web)service orientated architecture, the message integrity is
- > crucial. I would suggest to encrypt the data sent through the network
- > and also digitally sign it. The desktop application validates the
- > signature and if its not valid it will reject the incoming data. If the
- > signature is valid then the app can decrypt the response and process it.

THE COMMUNICATION WITH WEB SERVICES IS OVER HTTPS AND IT HAS DIGITAL CERTIFICATE. HOWEVER I UNDERSTAND SSL CERTIFICATES CAN BE SPOOFED AND PROBABLY IF THE DESKTOP APPLICATION VALIDATES THE SIGNATURE THAT COULD BE SPOOFED TOO?

Re: C# Exceptions

ALSO THE COMMUNICATION WITH DATABASE SERVER IS ENCRYPTED.

- > It's also recommended not to catch general exception like:
- > catch (Exception ex) {}, but catch and handle different kind of
- > exceptions as in : catch (NullReferenceException nullex) {} or catch
- > OverflowException, etc.

SORRY BUT I DIDN'T GET THIS ONE. IF AN APPLICATION THROWS DIFFERENT TYPES OF ACCEPTIONS LIKE NULL, AV, THAT WULD BE GOOD OR HAVING A GENERIC ERROR MESSAGE IS BETTER. I FEEL IT WOULD BE BETTER TO THROW A
A
GENERIC ERROR.

- > Another issue is that through reflection, ildasm you can re-construct
- > the source code of a managed app (see .NET Reflector). It's also

WE ARE ABLE TO RE-CONSTRUCT THE SOURCE CODE OF SOME IMPORTANT DLLS BUT
NOT FOR THE MAIN PROGRAM EXE FILE.

- > possible to patch system assemblies. It is possible to bring the
- > Framework to its knees with fuzzed data. You cant really trust anything,

YES, THAT'S TRUE. IT IS NOT DIFFICULT TO CRASH A C# DESKTOP APPLICATION BY FUZZING TECHNIQUES. ALTHOUGH I BELIEVE IT ISN'T REALLY
A SERIUOS ISSUE AS DOS ATTACK WILL AFFECT ONLY A SINGLE USER AND THE
OTHER IMPORTANT THING IS IT DOESN'T HAVE ANY PORTS OPEN OR SERVICE AVAILABLE ON THE NETWORK.

- > but do your best to detect it and do some defensive coding.

I THINK IT TAKES A LOT OF TIME TO PIN POINT THE CODE DEFECTS THAT CAUSED A CRASH. WOULD IT BE WORTHWHILE TO SPEND THAT MUCH TIME FIXING
IT? ANYWAYS THIS ISSUE IS MAXIMUM GOING TO CAUSE A NON-SERIOUS DOS.

This List Sponsored by: Cenzic

Need to secure your web apps?

Cenzic Hailstorm finds vulnerabilities fast.

Click the link to buy it, try it or download Hailstorm for FREE.

http://www.cenzic.com/products_services/download_hailstorm.php
