

[REVS] WLSI – Windows Local Shellcode Injection

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2006-03/msg00045.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxx>
 - *Date:* 16 Mar 2006 14:21:58 +0200
-

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

WLSI – Windows Local Shellcode Injection

SUMMARY

This paper describes a new technique to create reliable local exploits for Windows operating systems, the technique uses some Windows operating systems design weaknesses that allow low privileged processes to insert data on almost any Windows processes no matter if they are running under high privileges. We all know that local exploitation is much easier than remote exploitation but it has some difficulties. After a brief introduction and a description of the technique, a couple of samples will be provided so the reader will be able to write his/her own exploits.

DETAILS

Introduction:

When writing a local Windows exploit you can face many problems:

- Different return addresses.
- Because different Windows versions.
- Because different Windows service pack level.
- Because different Windows languages.
- Limited space for shellcode.
- Null byte restrictions.
- Character set restrictions.

[REVS] WLSI – Windows Local Shellcode Injection

– Buffer overflows/exploits protections.

To bypass those restrictions an exploit has to use many different return addresses and/or techniques. After you finish reading this paper you won't have to worry any more about that because it will be very easy to write a 100% reliable exploit that will work on any Windows version, service pack level, language, etc. and could bypass buffer overflows/exploits protections since the code won't be executed from the stack nor the heap and it won't use a fixed return address.

This technique relies in the use of Windows LPC (Local/Lightweight Procedure Call), this is an inter-process communication mechanism, RPC (Remote Procedure Call) uses LPC as a transport for local communications. LPC allow processes to communicate by "messages" using LPC ports. LPC is not well documented and here won't be detailed but you can learn more at the links listed on references section. LPC ports are Windows objects, servers (processes) can create named LPC ports to which clients (processes) can connect by referencing their names. You can see processes LPC ports using Process Explorer from <http://www.sysinternals.com/> by selecting a process in the upper panel and then looking at the lower panel at the Type column, they are identified by the word Port, you can see the port name, handle and by double clicking you can see additional information like permissions, etc. LPC is heavily used by Windows internals, also by OLE/COM, etc. this means that almost every Windows process has a LPC port. LPC ports can be protected by ACLs so sometimes a connection can not be established if the client process doesn't have proper permissions. To use this technique we will need to use a couple of APIs that will be detailed below.

To read more : <http://www.argeniss.com/research/WLSI.zip>
<http://www.argeniss.com/research/WLSI.zip>

ADDITIONAL INFORMATION

The information has been provided by <mailto:cesarc56@xxxxxxxx> Cesar.
The original article can be found at:
<http://www.argeniss.com/research/WLSI.zip>
<http://www.argeniss.com/research/WLSI.zip>

=====

This bulletin is sent to members of the SecuriTeam mailing list.
To unsubscribe from the list, send mail with an empty subject line and body to:
list-unsubscribe@xxxxxxxxxxxxxxxxx
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxxx

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.