Amplia Security - Research - WCE FAQ

By ampliasecurity.com

What is WCE?

Windows Credentials Editor (WCE) is a security tool that allows to list Windows logon sessions and add, change, list and delete associated credentials (e.g.: LM/NT hashes, Kerberos tickets and cleartext passwords).

The tool allows users to:

- - Perform Pass-the-Hash on Windows
- - 'Steal' NTLM credentials from memory (with and **without** code injection)
- - 'Steal' Kerberos Tickets from Windows machines

- - Use the 'stolen' kerberos Tickets on other Windows or Unix machines to gain access to systems and services
- - Dump cleartext passwords stored by Windows authentication packages

WCE is a security tool widely used by security professionals to assess the security of Windows networks via Penetration Testing.

What is the current version?

The current version of WCE 32bit is v1.42beta; you can download it here and the current version of WCE 64bit is v1.42beta; you can download it here. Since version 1.4beta there is also a "Universal Binary" which runs on both 32bit and 64bit platforms; you can download it here.

Who should use WCE?

WCE is aimed at security professionals and penetration testers. It is basically a post-exploitation tool to 'steal' and reuse NTLM hashes, Kerberos tickets and plaintext passwords which can then be used to compromise other machines. Under certain circumstances, WCE can allow you to compromise the whole Windows domain after compromising only one server or workstation.

What Operating Systems does WCE support?

WCE supports Windows XP, Windows 2003, Vista, Windows 7 and Windows 2008 (all SPs, 32bit and 64bit versions).

Is WCE like cachedump?

NO. Cachedump obtains NTLM credentials from the Windows Credentials Cache (aka logon cache, logon information cache, etc).
This cache can be disabled and it is very often disabled by network/domain/windows administrators (see http://support.microsoft.com/kb/172931).
WCE will be able to steal credentials even when this cache is disabled.
WCE obtains NTLM credentials from memory, which are used by the system to perform SSO; it uses a series of techniques the author of WCE developed and published some years ago.

Also, cachedump does not allow you to perform Pass-the-hash, nor does it allow you to 'steal' and reuse Kerberos tickets.

Is WCE like pwdump?

NO. Pwdump dumps NTLM credentials from the local SAM. WCE dumps credentials from memory; which are used by the system to perform SSO; it uses a series of techniques the author of WCE developed and published some years ago.
This is one of the reasons why you may be able to compromise the whole Windows domain after compromising a regular server; NTLM credentials stored in memory and obtained by WCE could have been left there, for

example, by Domain Administrators that connected to the server using RDP. In this scenario, pwdump will only allow you to obtain the NTLM credentials of the local SAM, that will probably be useless, since the server is not the domain controller.
Also, pwdump does not allow you to perform Pass-the-hash, nor does it allow you to 'steal' and reuse Kerberos tickets.

Is WCE like Cain & Abel?

No. WCE and Cain&Abel; are two different tools with different functionality. In fact, Cain&Abel; does not implement any of the functionality implemented by WCE, for example:

* It does not implement Pass-the-Hash natively in Windows
* It does not dump NTLM hashes stored in memory (it dumps local and remote SAMs, which is **not** the same thing. For more information read Post-Exploitation with WCE)
* It does not implement Pass-the-Ticket for Kerberos
* It does not dump cleartext logon passwords stored in memory
* etc.

This does not imply WCE is a *better* tool, Cain&Abel; also implements many things WCE does not, they are just different tools with different functionality. You should use both!

Where can I find more information about how WCE works?

"WCE Internals" presentation. RootedCon 2011; Madrid, Spain. (download)

Where can I find information on how to use WCE on a pentest?

"Post-Exploitation with WCE" presentation, UBA 2011 - Spanish (download) - English (download)

What privileges do I need to run WCE?

You need local administrator privileges to run WCE and be able to steal NTLM credentials from memory. This is a post-exploitation tool.
You also need local administrator privileges to perform Pass-The-Hash (change your current NTLM credentials, or launch a new program in a new Windows logon session with the NTLM credentials specified).

How do I list NTLM credentials in memory?

By default, WCE lists NTLM credentials in memory, no need to specify any options.
For example:

C:\Users\test>wce.exe
WCE v1.2 (Windows Credentials Editor) - (c) 2010,2011 Amplia Security - by Hernan Ochoa (hernan@ampliasecurity.com)
Use -h for help.

theuser:amplialabs:01FC5A6BE7BC6929AAD3B435B51404EE:0CB6948805F797BF2A82807973B89537

C:\Users\test>

In this case, only one user/credential set is listed. If there are more in memory, more will be displayed.

How do I change my current NTLM credentials?

wce.exe -s <username>:<domain>:<lmhash>:<nthash>
For example:

C:\Users\test>wce.exe -s
testuser:amplialabs:01FC5A6BE7BC6929AAD3B435B51404EE:0CB6948805F797BF2A82807973B89537

WCE v1.2 (Windows Credentials Editor) - (c) 2010,2011 Amplia Security - by Hernan Ochoa
(hernan@ampliasecurity.com)
Use -h for help.

Changing NTLM credentials of current logon session (00024E1Bh) to:
Username: testuser
domain: amplialabs
LMHash: 01FC5A6BE7BC6929AAD3B435B51404EE
NTHash: 0CB6948805F797BF2A82807973B89537
NTLM credentials successfully changed!

C:\Users\test>

How do I create a new logon session and launch a program with new NTLM credentials?

wce.exe -s <username>:<domain>:<lmhash>:<nthash> -c <program>
For example:

C:\Users\test>wce.exe -s
testuser:amplialabs:01FC5A6BE7BC6929AAD3B435B51404EE:0CB6948805F797BF2A82807973B89537 -c
cmd.exe

WCE v1.2 (Windows Credentials Editor) - (c) 2010,2011 Amplia Security - by Hernan Ochoa
(hernan@ampliasecurity.com)
Use -h for help.

Changing NTLM credentials of new logon session (000118914h) to:
Username: testuser
domain: amplialabs
LMHash: 01FC5A6BE7BC6929AAD3B435B51404EE
NTHash: 0CB6948805F797BF2A82807973B89537
NTLM credentials successfully changed!

C:\Users\test>

At this point, a new cmd.exe instance will be launched and network connections using NTLM initiated from that instance will use the NTLM credentials specified. Of course, you can run any program, not just cmd.exe.

This is feature is very useful, because you can do many tests and do Pass-the-Hash with many different users without having to change your current Windows logon session and credentials.

How can I generate NTLM hashes with WCE? (for testing purposes)

wce.exe -g <cleartext password>
For example:

C:\Users\test>wce.exe -g mypassword

WCE v1.2 (Windows Credentials Editor) - (c) 2010,2011 Amplia Security - by Hernan Ochoa (hernan@ampliasecurity.com)
Use -h for help.

Password: mypassword
Hashes: 74AC99CA40DED420DC1A73E6CEA67EC5:A991AE45AA987A1A48C8BDC1209FF0E7

C:\Users\test>

While testing WCE, and other things, it is very common to have the need to generate LM and NT hashes from a password. This can be done using the '-g' parameter as shown above.

What is 'safe mode'?

WCE is the first and only tool that can read NTLM credentials stored by Windows in memory **without** injecting code. WCE is able to locate and understand the undocumented structures used by Windows to store the credentials, find encryption keys and decrypt credentials **just by reading** the system's memory.

This technique is very very safe (after all, the tool is just reading memory; thus the name 'safe mode') and tries to ensure that the system where WCE is executed will not crash. This is extremely important if you are a penetration tester and want to run WCE without risking a server crash.

WCE will automatically attempt to use this technique first when obtaining NTLM credentials from memory; however it will also automatically attempt code injection if the first technique failed.

For this reason, if you want tu ensure WCE will **only** attempt to obtain NTLM credentials by reading memory (without code injection), you can use the **-f** switch (Force 'safe mode').

Example:

C:\Users\test>wce.exe -f
WCE v1.2 (Windows Credentials Editor) - (c) 2010,2011 Amplia Security - by Hernan Ochoa

(hernan@ampliasecurity.com)
Use -h for help.

theuser:amplialabs:01FC5A6BE7BC6929AAD3B435B51404EE:0CB6948805F797BF2A82807973B89537

C:\Users\test>

Having said that, you use the tool under your own risk; no guarantee is given.

How can I write hashes obtained by WCE to a file?

Use the -o switch. For example:

C:\>wce -o output.txt
WCE v1.2 (Windows Credentials Editor) - (c) 2010,2011 Amplia Security - by Hernan Ochoa
(hernan@ampliasecurity.com)
Use -h for help.

C:\>type output.txt
test:AMPLIALABS:0102030405060708090001020304050 6:989712345678650198127345768901 02
C:\>

How can I dump logon cleartext passwords with WCE?

The *-w* switch can be used to dump logon passwords stored in cleartext by the *Windows Digest Authentication package*. For example:

C:\>wce -w
WCE v1.3beta (Windows Credentials Editor) - (c) 2010,2011,2012 Amplia Security - by Hernan Ochoa
(hernan@ampliasecurity com)
Use -h for help.

test\MYDOMAIN:mypass1234
NETWORK SERVICE\WORKGROUP:test

This video shows the use of the -w switch in a Windows 2008 Server (watch in 720p for best quality).

How can I prevent WCE dumping my logon password in cleartext?

When you login into a Windows system; your cleartext password is handed over to all the Security Packages installed on the system. This includes the NTLM security package (msv1_0.dll) the Kerberos security package (kerberos.dll), the Digest Authentication Security Package (wdigest.dll) etc. These packages take the cleartext password and basically do what they desire with it. For example, the NTLM security package generates and stores in memory the NTLM hashes discarding the cleartext password, and the Digest Authentication package stores in memory the cleartext password encrypted. The techniques invented by WCE precisely consist in extracting from these packages these credentials stored in memory.

For this reason, one of the ways to prevent WCE dumping your cleartext login password and other credentials is to avoid loading the Security Packages from which WCE retrieves them.

These are defined in registry at the following location:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Security Packages

You will find there a list similar to the following:

kerberos
msv1_0
schannel
wdigest
tspkg
pku2u

You can remove an item from the list and (after rebooting) Windows will not load the corresponding Security Package.

For example, you can remove *wdigest* and *tspkg* and WCE will not be able to dump the cleartext password stored by these packages simply because they will not be loaded anymore.

Although this works, keep in mind that you may encounter problems if your environment uses one of the security packages removed; you will need to test how this affects you specifically. Also, removing fundamental security packages like msv_1_0 can have catastrophic consequences, so apply this technique at your own risk.

WCE is detected by the antivirus/HIPS. what can I do to avoid detection?

Use a *PE Packer*, for example UPX. If *UPX does* not do the trick, try other *PE Packers*, there are many out there. Also, since you need administrator privileges to run *WCE*, try disabling the AV/HIPS before running WCE...

What is GETLSASRVADDR.EXE?

GETLSASRVADDR.exe is a tool (included with WCE) that can be used to obtain automatically the needed addresses for WCE to be able to read logon sessions and NTLM credentials from memory (without code injection) when WCE is not able to do it by itself out-of-the-box.

Addresses obtained can then be used with WCE using the -A switch.
This tool requires the DLLs *symsrv.dll* and *dbghelp.dll* available from the *"Debugging Tools for Windows"* package.

When should I use GETLSASRVADDR.EXE?

Basically, you should use *GETLSASRVADDR.exe* when you want to use 'safe mode' to extract hashes from memory on a system where out-of-the-box WCE is unable to make it work.

GETLSASRVADDR.exe will give you the information WCE needs to get 'safe mode' working.

I can't get GETLSASRVADDR.EXE to work. What's the problem?

The most common source of problems is that you are missing the DLL files *symsrv.dll* and *dbghelp.dll* available from the "Debugging Tools For Windows" package.

This is most likely the case if you are getting the following error message:

Connecting to Microsoft.com symbol server...please wait..
Error: cannot find symsrv.dll
Error: Cannot obtain addresses

Read the presentation WCE Internals for an explanation on why these DLLs are required.
The tool *getlsasrvaddr.exe* is meant to be used in the attacker's machine, and not in compromised machines; so this requirement should not be an issue.

Another common issue is having UAC enabled and not being able to access *c:\windows\system32\lsasrv.dll*. In this case, just copy *lsasrv.dll* to another directory and try again.

Who's the author of WCE? Is he also the author of the PSH Toolkit?

The author of WCE is Hernan Ochoa (hernan [at] ampliasecurity.com); and yes, he is also the author of the now defunct Pass-The-Hash Toolkit.

How is WCE better than the PSH Toolkit?

The Pass-The-Hash (PSH) Toolkit does not work anymore. It does not support newer updates for Windows XP and 2003; and it does NOT support Windows 7 and 2008 at all.

WCE is basically a complete rewrite form scratch, it uses new techniques and does automagically lots of things to make its use easier and to make it work automatically in more platforms. It also works perfectly with all Windows versions, including Windows 7 and 2008; and it is the only tool that is able to read credentials just by reading memory; which is very important to penetration testers, since this means the chances of crashing a server when using WCE are almost zero (although neither the author nor Amplia Security guarantees this in any way; you use the tool at your own risk).

Also, the PSH Toolkit does not allow you to 'steal' and reuse Kerberos tickets.
(Note: remember the author of WCE is also the author of the PSH Toolkit).
