

Re: Securing win32_process.create ?

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.wmi/2009-02/msg00016.htm>

- *From:* Gerry Hickman <gerry666uk2@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 10 Feb 2009 21:02:28 +0000
-

Hi,

The reason you are having problems is essentially because you're expecting an admin operation (mounting a fake partition) to be able to be performed by a user who does not have admin rights. That's impossible using the normal impersonation mechanism of WMI and PsExec (their strengths).

Looking at TrueCrypt, it certainly does not look like it's been designed for remote instantiation, nor to run as a non-interactive remote process (the only type supported by WMI).

Surely the better solution is to use encryption that has an API and is designed to work over a network and without a UI?

The reason there's no documentation for this is that it's not how you are supposed to use WMI.

If we forget TrueCrypt for a minute, there is a way to run a process as a different identity as the caller.

- A) Using COM/DCOM client/server programming
- B) Creating a service wrapper with communications and use a different logon credential
- C) Run a scheduled job with communications as a different user

Matt Brown – nyc wrote:

Thanks for the reply back.

I'm looking to remote mount a truecrypt drive by executing the truecrypt process on the remote:

I currently have a py script prompting my user(s) for a password, which then gets piped into a command line for a program called cpau (to run a process as a different user locally), which in turn runs psExec to execute the process with the cpau provided user context change.

The remote system is a file server, and a single one at that, where the truecrypt file contain is to be mounted.

Do you understand now that I'm not attempting to be malicious, nor

Re: Securing win32_process.create ?

would I have to hack anything in order to adjust the DCOM ACLs or namespace ACLs.

The lack of documentation surprises me. I am looking for the most granular thing, and am looking to do it once. It is fairly obscured right now, but it isn't truly as secure as I'd like it to be. I attempted to put more granular restrictions, by not allowing the "granting" the user the SeDenyInteractiveLogonRight privilege; via Active Directory User profile and by ntrights.exe. However, psexec will not run properly unless this setting is set.

Basically, it is always more secure to provide granular access; because right now, my users can, in essence, administer one of my servers; and nobody likes that.

I'm glad to be having a dialog about this after all the dead ends i've hit.

Any further input is appreciated!

Thanks,

Matt

On Feb 4, 2:56 pm, Gerry Hickman <gerry666...@xxxxxxxxxxxxxxxxxxxx> wrote:

Hi,

Can you perhaps re-word what you are actually trying to do? I don't remember you saying the client user should not have admin rights to the target machine?

It would be an odd case, since DCOM and namespace security would need hacked to allow it, then you'd have to ask why a non-administrative user would be allowed to run a process on a remote machine and you'd also have to ask what that process would be able to do? e.g. it would not be able to start and stop services (in the normal way) without Admin rights. You'd also have to change the above mentioned security settings on every remote machine. It's also worth noting this kind of thing has become much more difficult with Vista/2008.

Matt Brown – nyc wrote:

Thanks for replying... My purpose is to avoid exactly that. I do not want to user to have administrative rights to the remote

Re: Securing win32_process.create ?

machine.

I had to go the other way to work around the issue of the explicit

lack of documentation of this:

http://forum.sysinternals.com/forum_posts.asp?TID=17497&PID=87377#87377

<http://forums.truecrypt.org/viewtopic.php?t=14335>

Can anyone answer my original question?

On Jan 28, 2:59 pm, Gerry Hickman

<gerry666...@xxxxxxxxxxxxxxxxxxx>

wrote:

Hi,

I'm not sure what you're trying to do. If you just want to create a process, why not just connect to the machine with Admin rights, and then create the process?

Matt Brown – nyc wrote:

Hello,

I've performed research for some time in an attempt to secure the create method of win32_process and came up with little headway:

–<http://msdn.microsoft.com/en-us/library/aa393266.aspx>

–[http://msdn.microsoft.com/en-us/library/aa393613\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa393613(VS.85).aspx)(root

\CIMV2, as stated in previously listed MSDN article)

Permissions have been assigned mirroring the local Administrators group.

However, when executing the win32_process.create method, an error is received: 8, "Unknown Failure."

Does anyone know what namespace or other access the win32_process.create accesses?

Does anyone know how to further determine the error?

Any input is appreciated.

Thanks very much,

Matt Brown

Re: Securing win32_process.create ?

--
Gerry Hickman (London UK)

--
Gerry Hickman (London UK)

--
Gerry Hickman (London UK)

.