GPP Password Retrieval with PowerShell

Archived: 2026-04-05 17:39:46 UTC

Last week, I read a great post entitled "Exploiting Windows 2008 Group Policy Preferences" that I wish I saw sooner. The article included a nice Python script to accomplish the task of decrypting passwords that were set using the GPP feature in Windows 2008 domains. However, it looked like something that would be handy to have in a PowerShell script. Before I continue, I would like to point out the updated disclaimer, it certainly applies to this post.

You should read the original article, but the quick summary is that its possible for any authenticated user (this includes machine accounts) on the domain to decrypt passwords that are enforced with Windows 2008 Group Policy Preferences. From my experience, this practice is common for larger domains which need to set different local administrator ("500" account) passwords for different OUs.

Python is an excellent scripting language, but PowerShell has two notable advantages in this specific use-case. First, PowerShell does not require any additional libraries since it has access to the entire .NET framework. Second, PowerShell is installed by default on all modern Windows systems to include Windows Server 2008 so it can be used right from the machine you are on.

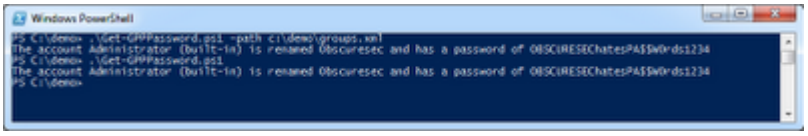
The following Get-GPPPassword PowerShell script can be used by penetration testers to elevate to local administrator privileges (on your way to Domain Admin) by downloading the "groups.xml" file from the domain controller and passing it to the script. The files are typically found in:

\\domain\SYSVOL\domain\Policies\{*}\Machine\Preferences\Groups\Groups.xml

Get-GPPPassword (Use Updated Version)

To run the function, just copy and paste the text into powershell and type 'Get-GPPPassword'. This will in effect bypass the ExecutionPolicy.

Writing this script ended up not being as easy as I originally thought mostly due to never dealing with .NET and crypto before. I would like to thank Matt Graeber for solving the null IV issue, Mike Santiago for general code improvements and of course Emilien Giraul (and the Sogeti ESEC Pentest team for their detailed writeup).



Try it out and let me know what you think.

Update 26 May 2012
You can also download the maintained version of the script from the PowerSploit repository on GitHub. It already has some great scripts for Windows post-exploitation on it!

Update 16 June 2012
Updated the script block with the improvements from Matt Graeber. Matt wrapped it into a function and apparently saved a puppy by creating a new object (avoiding the use of write-host).

Update 3 July 2013
I have reorganized and rewritten the script. You can find the updated version and read about it here.
-Chris
