Post-Exploitation with WCE v1.2

Pass-the-Hash. Pass-the-ticket & more…

Date: 01-07-2011

Author:
Hernan Ochoa
<hernan@ampliasecurity.com>
Windows Authentication

- The cleartext password is not sent
- The NTLM protocol is used

**SAM Database**

<table>
<thead>
<tr>
<th>User</th>
<th>Hash LM</th>
<th>Hash NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>cgarcia</td>
<td>A6BCD..</td>
<td>B0FD1..</td>
</tr>
<tr>
<td>ajuarez</td>
<td>B90DF..</td>
<td>CCDF..</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
NTLM Authentication

Let's break down the process:

1. **Init connection**
   - The client sends a challenge random `C` to the server.

2. **Responds**
   - The server sends `cgarcia, R`.

3. **Sends**
   - The client calculates `R = f(LMHash(pwd1)/NTHash(pwd1), C)`.

4. **SAM Database**
   - The client looks up the user database for `cgarcia` and `ajuarez`.

5. **Calculate**
   - `R' = f(SAM[Lmhash/nthash], C)` for both users.

6. **Comparison**
   - If `R' == R`, access granted.
   - If `R' <> R`, access denied.

**Example**:

- **cgarcia**:
  - LMHash = A6BCD6F1D1
  - NTHash = B0FD1CDE

- **ajuarez**:
  - LMHash = B90DF1E2F3
  - NTHash = CCDF2345

- **User** | **Hash LM** | **Hash NT**
  --|---|---
  cgarcia | A6BCD6F1D1 | B0FD1CDE
  ajuarez | B90DF1E2F3 | CCDF2345
  ... | ... | ...
After compromising a Windows box...

- ‘Dump’ the SAM
  - `pwdump3/3e/4/5/6/7, fgdump, etc`
    - Administrator:500:0102030405060708090A0B0C0D0E0F10:102030405060708090A0B0C0D0E0F10

- Crack/Brute-Force hashes to obtain cleartext password
  - Takes time.. (e.g.: pentest time is limited)
  - No guarantee the password will be obtained
  - Rainbow tables were not widely used
    - Less computing power, storage, etc
    - Still, not the answer to everything
Pass-The-Hash Technique

Published by Paul Ashton in 1997

http://www.securityfocus.com/bid/233/discuss
Imhash = LMHash(“pwd1”)
nthash = NTHash(“pwd1”)

\[ R = f(\text{lmhash/nthash}, C) \]

- Cleartext password is not needed for NTLM auth
- Only lmhash/nthash are needed
- No need to crack/brute-force
- Just use the hashes directly

SAM Database

<table>
<thead>
<tr>
<th>User</th>
<th>Hash LM</th>
<th>Hash NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>cgarcia</td>
<td>A6BCD..</td>
<td>B0FD1..</td>
</tr>
<tr>
<td>ajuarez</td>
<td>B90DF..</td>
<td>CCDF..</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

\[ R' = f(\text{SAM[lmhash/nthash]}, C) \]
- If \( R' = R \) => Access Granted
- If \( R' \neq R \) => Access Denied
Pass-the-hash: ‘exploitation’

- **Modified Smbclient** (SAMBA)
  
  - `smbclient //192.168.1.120/c$ -U Administrator -p 4ECC0E7568976B7EAAD3B435B51404EE:551E3B3215FFD87F5E037B3E3523D5F6`

- **Samba-TNG**

- **Many 3rd-party SMB+NTLM stacks**
  
  - Python, Ruby, Java, etc
3rd-party SMB+NTLM stacks: Limitations

• Limited and partial functionality

• Always running behind Windows
  – New functionality has to be implemented, some by reverse engineering
    • Complex, requires time and effort

• Cannot use native Windows applications
  – They ask for username and cleartext password, not hashes...
Enter, *Windows Credentials Editor*...
What is WCE?

• Tool to manipulate Windows logon sessions
  – Add, list, delete, modify
  – Obtain credentials associated with logon sessions
  – Pass-the-hash (NTLM)
  – Pass-the-ticket (Kerberos)
Pre-WCE/Pass-the-hash attacks

• Without WCE..
  – Crack/Brute-force hashes to obtain cleartext password
  – Crack/Brute-force ‘encrypted’ hashes (C,R->NTLM) to obtain cleartext password
  – Use 3rd-party SMB+NTLM stacks, w/limited and partial functionality
  – More difficult/not possible to do pass-the-hash while pivoting among Windows boxes
Post-WCE/Pass-the-hash attacks

• With WCE...
  
  – Do *Pass-the-hash* directly with the hashes
    • No need to attempt to crack/brute-force hashes
    • Will be able to use them even if you cannot crack them
  
  – 3\textsuperscript{rd}-party SMB+NTLM stacks problems eliminated
  
  – Easier to do *Pass-the-hash* while pivoting among Windows boxes
Demo #1
WCE: Pass-the-hash
WCE: ‘Steal’ Credentials from memory

• New ‘attack’ implemented by WCE
  – **It is not** *Pass-the-hash*, it’s another technique..
  – Sometimes the two are confused, but they are not the same..

• Allows you to obtain usernames and NTLM hashes stored in memory
WCE: ‘Steal’ Credentials from memory

• Why are they in memory?
  • NTLM auth package
  • SSO

MyDomain/cgarcia
h1 = LMHash(“pwd1”)
h2 = NTHash(“pwd1”)

Network
NTLM

cgarcia
pwd1
When are they stored in memory?

- Interactive logon sessions at the console
- Remote logon sessions via RDP
- RunAs
- Windows Services running under specific user accounts
- Windows APIs used by applications
- Etc.
WCE: ‘Steal’ Credentials from memory

When are they stored in memory?

• Interactive logon sessions at the console

MyDomain/cgarcia
h1 = LMHash(“pwd1”)
h2 = NTHash(“pwd1”)

cgarcia
pwd1
WCE: ‘Steal’ Credentials from memory

When are they stored in memory?

• Remote logon sessions via RDP

Via RDP:
MyDomain/cgarcia
h1 = LMHash("pwd1")
h2 = NTHash("pwd1")
WCE: ‘Steal’ Credentials from memory

When are they stored in memory?

- RunAs

WD-PC/muser
h1 = LMHash("pwd4")
h2 = NTHash("pwd4")
WCE: ‘Steal’ Credentials from memory

When are they stored in memory?

- Windows Services

MyDomain/cgarcia

h1 = LMHash("pwd1")

h2 = NTHash("pwd1")
WCE: ‘Steal’ Credentials from memory

When are they stored in memory?

- Windows APIs used by applications

Example:

```c
MyDomain/cgarcia
h1 = LMHash("pwd1")
h2 = NTHash("pwd1")
```

LogonUser Function

The `LogonUser` function attempts to log a user on to the local computer. The local computer is the computer from which `LogonUser` was called. You cannot use `LogonUser` to log on to a remote computer. You specify the user with a user name and domain and authenticate the user with a `plaintext` password. If the function succeeds, you receive a handle to a token that represents the logged-on user. You can then use this token handle to impersonate the specified user or, in most cases, to create a process that runs in the context of the specified user.

Syntax

```c
BOOL LogonUser(
    __in  LPTSTR lpstrUsername,
    __in_opt  LPTSTR lpstrDomain,
    __in  LPTSTR lpstrPassword,
    __in  DWORD dwLogonType,
    __in  DWORD dwLogonProvider,
    __out  PHANDLE phToken
);
```
WCE: ‘Steal’ Credentials from memory

• WCE can obtain the LM Hash..
  – By default, nowadays, Windows does not store the LM hash in the SAM
    • It is weak; “easy” to crack
WCE: ‘Steal’ Credentials from memory

• WCE can obtain the LM Hash from memory
  – Windows generates and stores the hashes in memory, including the LM hash
    • Interactive sessions
    • Others previously mentioned
  – Possible to crack it and obtain cleartext password to use in places where NTLM is not the auth method
WCE: ‘Steal’ Credentials from memory

Example:

The LM Hash is not in the SAM

```
Administrator:500:N0 PASSW0RD**************************************************************************:N0 PASSW0RD**************************************************************************: 
Guest:501:N0 PASSW0RD**************************************************************************:N0 PASSW0RD**************************************************************************: 
muser:1000:N0 PASSW0RD**************************************************************************:9BF617CAEFC9DFE18995B5A300174176**: 
Completed.
```

Pwdump output

The LM Hash is in memory

```
C:\Users\muser\wce\wce
WCE v1.2 (Windows Credentials Editor) - (c) 2010,2011 Amplia Security - by Hernan Ochoa (hernan@ampliasecurity.com)
Use -h for help.
muser:wd-PC:A3283469F98CF766AAD3B435B51404EE:9BF617CAEFC9DFE18995B5A300174176
```

WCE output
When are they **NOT** stored in memory?

- **Network Logons**
  - The hashes never reach the remote server
WCE: ‘Steal’ Credentials from memory

Post-exploitation attack scenario

- Attacker compromises remote Windows box

- Run WCE to obtain credentials stored in memory
- Use those hashes to do Pass-The-Hash with WCE
Demo #2

WCE: ‘Steal’ credentials from memory
WCE: ‘Steal’ Credentials from memory

Especially interesting in Windows Domain Environments

<table>
<thead>
<tr>
<th>User</th>
<th>Hash LM</th>
<th>Hash NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>cgarcia</td>
<td>A6BCD..</td>
<td>B0FD1..</td>
</tr>
<tr>
<td>ajuarez</td>
<td>B0DDF..</td>
<td>CCDF..</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Domain Controller (DC)

SAM

Backups Server

Domain User

Domain User

Domain User
WCE: ‘Steal’ Credentials from memory

WIThOUT WCE..

- Attacker owns remote Windows box
- Only has access to local SAM..
- No Domain Users there, not very useful...

<table>
<thead>
<tr>
<th>User</th>
<th>Hash LM</th>
<th>Hash NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>garcia</td>
<td>A6BCD..</td>
<td>B0FD1..</td>
</tr>
<tr>
<td>juarez</td>
<td>B90DF..</td>
<td>CCDF..</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Domain Controller (DC)

Backups Server

User
Hash LM
Hash NT

Administ rador  F0DE..  CCDF1.. 
Guest           AABCD..  FFD4F..  

SAM

Domain User

Domain User

Attacker
WCE: ‘Steal’ Credentials from memory

WITH WCE..

- Attacker owns Backups server
- Obtains credentials from memory..
  - Probably will find Domain Users there..
  - Can possibly compromise the whole domain!
WCE: ‘Steal’ Credentials from memory

Typical Scenario

- Domain Controller (DC)
- Backups Server
- RDP/Terminal Services Connection
- Domain Administrator
- Domain User
- Attacker
WCE: ‘Steal’ Credentials from memory

Typical Scenario

Domain: Administrator
CDB 70D9EDA812
DBF:C9DF712
D09D7DF1C

Backups Server

Domain Controller (DC)

RDP/Terminal Services Connection

Domain Administrator

Domain User

Attacker
WCE: ‘Steal’ Credentials from memory

Typical Scenario

- **WCE ‘steals’ the Domain Administrator Credentials!**
RDP ´exposes´ credentials

• When you RDP to a remote box, you leave the NTLM hashes of your password in the remote server´s memory

  • NTLM hashes are equivalent to the cleartext password (pass-the-hash+wce)

  • So, we could say you are leaving there your password...
RDP ´exposes´ credentials

• Even when using pass-through authentication

• Credentials (user+domain+hashes) are stored in Domain Controller

• But when you use RDP, they are also left in the memory of the remote Windows box you are RDPing to!

• Be careful where you are RDPing to...
WCE: ‘Steal’ Credentials from memory

RDP ‘exposes’ credentials

• Attacker on the remote Windows box can obtain credentials from memory
  • Local Administrator
  • Regular Domain User w/local admin privs
  • Attacker that owns less secured box, regular users are more vulnerable
  • Etc.
WCE: ‘Steal’ Credentials from memory

RDP ‘exposes’ credentials: Example

- Domain Controller (DC)
- SAM
- Domain User
- Workstation
- Attacker

<table>
<thead>
<tr>
<th>User</th>
<th>Hash LM</th>
<th>Hash NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>cgarcia</td>
<td>A6BCD..</td>
<td>B0FD1..</td>
</tr>
<tr>
<td>ajuarez</td>
<td>B90DF..</td>
<td>CCDF..</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

RDP/Terminal Services Connection
WCE: ‘Steal’ Credentials from memory

RDP ‘exposes’ credentials: Example

Domain Controller (DC)

SAM

User | Hash LM | Hash NT
--- | --- | ---

cgarcia | A6BCD.. | B0FD1..
ajurez | B90DF.. | CCDF..
... | ... | ...

Pass-through auth

RDP/Terminal Services Connection

Domain Administrator

Domain User

Attacker

Domain User Workstation
WCE: ‘Steal’ Credentials from memory

RDP ‘exposes’ credentials: Example

Domain Administrator

Domain Controller (DC)

SAM

User | HashLM | HashNT
--- | --- | ---

cgarcia | A6B0D.. | B0F0D1..
ajuarez | B0D0F.. | C0DF..
--- | --- | ---

Domain:Administrator:CDB 70D9EDA812 DBF:C9DF712 D09D7DF1C

Domain:User

Workstation

Attacker

RDP/Terminal Services Connection
WCE: ‘Steal’ Credentials from memory

RDP ‘exposes’ credentials: Example

- Domain Administrator
- Domain Controller (DC)
- SAM
- User | Hash LM | Hash NT
  --- | --- | ---
  cgarcia | A6BC.. | B0FD1..
  ajuarez | B90DF.. | CCDF..
  ... | ... | ...
- Domain User
- Workstation
- Attacker
- Domain: Administrator: CDB 70D9EDA812 DBF:C9DF712 D09D7DF1C
- RDP/Terminal Services Connection
- Pass-through auth
WCE: ‘Steal’ Credentials from memory

RDP ‘exposes’ credentials: Example

<table>
<thead>
<tr>
<th>User</th>
<th>Hash LM</th>
<th>Hash NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>cgarcia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ajuarez</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Domain Controller (DC)

Domain Administrator

Domain User

Attacker

Domain User Workstation

Domain:Administrator: CDB 70D9EDA812 DBF:C9DF712 D09D7DF1C
WCE: ‘Steal’ Credentials from memory

RDP: Disconnect != Log Off
WCE: ‘Steal’ Credentials from memory

RDP: Disconnect != Log Off
WCE: ‘Steal’ Credentials from memory

RDP: Disconnect != Log Off
RDP: Disconnect != Log Off

• ´Disconnect´ leaves NTLM hashes in memory
  • The Logon session is not terminated

• ´Log Off´ terminates the logon session
  • Hashes are erased from memory

• **Always ´Log Off´!**
  • Users tend to just ´Disconnect´...
  • Including Administrators..
WCE: ‘Steal’ Credentials from memory
WCE: ‘Steal’ Credentials from memory

Bug: Zombie Logon Sessions!

Day 1

Domain Controller (DC)

Backups Server

RDP/Terminal Services Connection

Domain Administrator

Domain User

Attacker
WCE: ‘Steal’ Credentials from memory

Bug: Zombie Logon Sessions!

Day 1

Domain: Administrator: CDB 70D9EDA812 DBF:C9DF712 D09D7DF1C

Backups Server

Domain Controller (DC)

RDP/Terminal Services Connection

Domain Administrator

Domain User

Attacker
Bug: Zombie Logon Sessions!

Day 1

• Administrator logs off, disconnects, finished what he/she was doing..

Domain: Administrator: CDB 70D9EDA812 DBF:C9DF712 DO9D7DF1C

Backups Server

Domain Controller (DC)

Domain Administrator

Domain User

Attacker

WCE: ‘Steal’ Credentials from memory
WCE: ‘Steal’ Credentials from memory

Bug: Zombie Logon Sessions!

BUG! Windows does not erase from memory NTLM hashes!

Day 1

Domain: Administrator: CDB 70D9EDA812 DBF:C9DF712 D09D7DF1C

Domain Controller (DC)

Backups Server

Domain Administrator

Domain User

Attacker
WCE: ‘Steal’ Credentials from memory

Bug: Zombie Logon Sessions!

Domain: Administrator
70D9EDA812
DBF:C9DF712
D09D7DF1C

Domain Controller (DC)

Backups Server

Attacker owns server, steals NTLM credentials!

He/She is Domain Admin!

Day 31.. ~ A month later..
WCE: Pass-the-Ticket (Kerberos)

• New attack implemented by WCE v1.2
  – First and only tool that implements this AFAIK

• Post-exploitation

• Equivalent to Pass-The-Hash for NTLM

• You can ‘steal’ Kerberos TGT/tickets & use them in other Windows and *Unix boxes
• ‘Stolen’ tickets can be used to access remote services
  – Example: SMB shares

• The TGT (Ticket Granting Ticket) can be used to create new tickets
  – Gain access to more services / computers
Conclusions

• WCE brings new post-exploitation techniques
  – Pass-the-Hash (NTLM)
  – Steal NTLM from memory
  – Pass-the-ticket (Kerberos)

• Useful for pentests

• You need to know them to defend yourself
  – Not just to attack..
• “WCE Internals” Presentation
  – RootedCon 2011; Madrid, España
  – More technical details about implementation

Questions?

Thank you!

hernan@ampliasecurity.com

@hernano
@ampliasecurity

www.ampliasecurity.com