Quarks PwDump - Quarkslab's blog

By Sébastien Kaczmarek

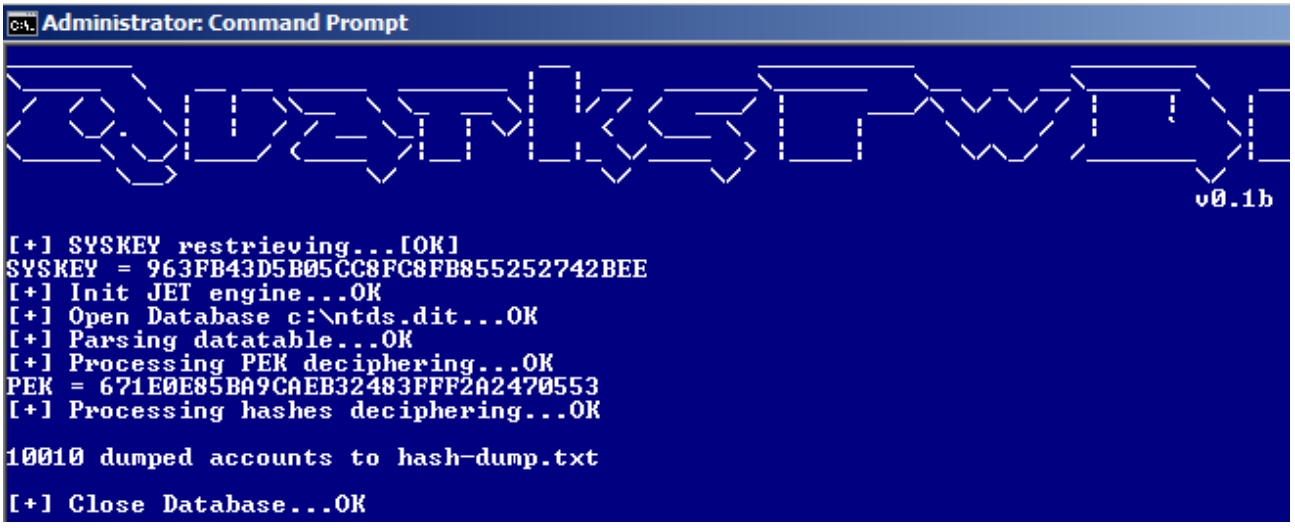
Archived: 2026-04-05 21:43:47 UTC

Quarks PwDump is new open source tool to dump various types of Windows credentials: local account, domain accounts, cached domain credentials and bitlocker. The tool is currently dedicated to work live on operating systems limiting the risk of undermining their integrity or stability. It requires administrator's privileges and is still in beta test.

Quarks PwDump is a native Win32 open source tool to extract credentials from Windows operating systems.

It currently extracts : Local accounts NT/LM hashes + history Domain accounts NT/LM hashes + history stored in NTDS.dit file Cached domain credentials Bitlocker recovery information (recovery passwords & key packages) stored in NTDS.dit

JOHN and LC format are handled. Supported OS are Windows XP / 2003 / Vista / 7 / 2008 / 8



Why another pwdump-like dumper tool?

- No tools can actually dump all kind of hash and bitlocker information at the same time, a combination of tools is always needed.
- Libesedb (http://sourceforge.net/projects/libesedb/) library encounters some rare crashs when parsing different NTDS.dit files.
- It's safer to directly use Microsoft JET/ESE API to parse databases originally built with same functions.

- Bitlocker case has been added even if some specific Microsoft tools could be used to dump those information. (Active Directory addons or VBS scripts)

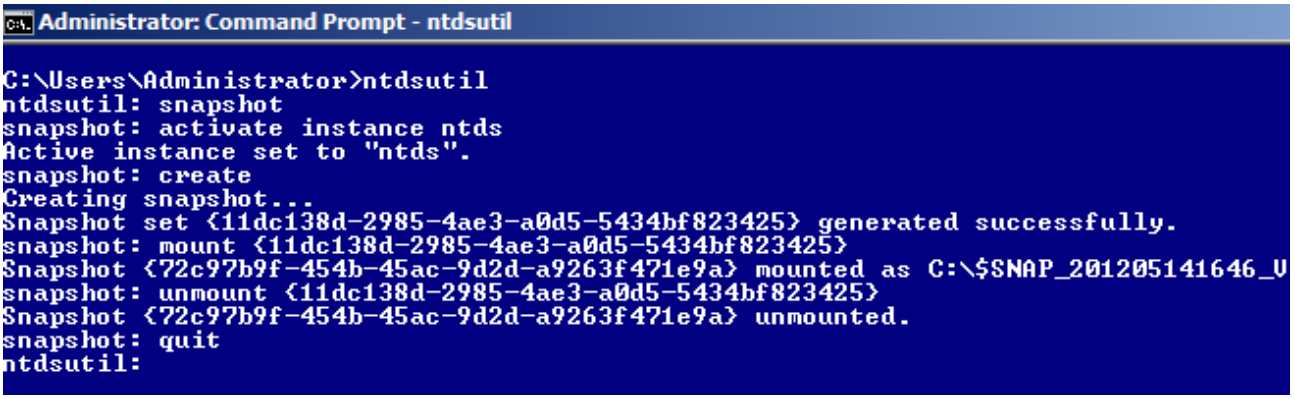
The tool is currently dedicated to work live on operating systems limiting the risk of undermining their integrity or stability. It requires administrator's privileges.

We plan to make it work full offline, for example on a disk image.

How does it internally work?

Case #1: Domain accounts hashes are extracted offline from NTDS.dit

It's not currently full offline dump cause Quarks PwDump is dynamically linked with ESENT.dll (in charge of JET databases parsing) which differs between Windows versions. For example, it's not possible to parse Win 2008 NTDS.dit file from XP. In fact, record's checksum are computed in a different manner and database files appear corrupted for API functions. That's currently the main drawback of the tool, everything should be done on domain controller. However no code injection or service installation are made and it's possible to securely copy NTDS.dit file by the use of Microsoft VSS (Volume Shadow Copy Service).



Case #2: Bitlocker information dump

It's possible to retrieve interesting information from ActiveDirectory if some specific GPO have been applied by domain administrators (mainly "Turn on BitLocker backup to Active Directory" in group policy). Recovery password: it's a 48-digits passphrase which allow a user to mount its partition even if its password has been lost. This password can be used in Bitlocker recovery console.

Key Package : it's a binary keyfile which allow an user to decipher data on a damaged disk or partition. It can be used with Microsoft tools, especially Bitlocker Repair Tool.

For each entry found in NTDS.dit, Quarks PwDump show recovery password to STDOUT and keyfiles (key packages) are stored to separate files for each recovery GUID: {GUID_1}.pk, {GUID_2}.pk,...

```
[+] Init JET engine...OK
[+] Open Database c:\ntds.dit...OK
[+] Parsing datatable...OK

------------------------------------------- BEGIN DUMP ---------------------------
Bitlocker entry
        Volume GUID: {A177E194-2CA0-4B32-BD97-00A524D17F6D}
        Recovery GUID: {E3CA8A93-18F2-4DA5-980D-7AB2C68CE9F3}
        Recovery password: 336864-339284-434379-590227-152372-370062-690305-329241
        Key-package: saved to binary file E3CA8A93-18F2-4DA5-980D-7AB2C68CE9F3.pk
Bitlocker entry
        Volume GUID: {D38CF049-A557-4B1F-BD0E-D0F9DCA56B47}
        Recovery GUID: {A7CC4474-F931-4699-975B-5E70373F765A}
        Recovery password: 147829-129635-145222-191301-038203-016885-108174-149303
        Key-package: saved to binary file A7CC4474-F931-4699-975B-5E70373F765A.pk
-------------------------------------------- END DUMP ----------------------------
[+] Close Database...OK
```

Volume GUID: an unique value for each BitLocker-encrypted volume. Recovery GUID: recovery password identifier, it could be the same for different encrypted volumes.

Quarks PwDump does no retrieve TPM information yet. When ownership of the TPM is taken as part of turning on BitLocker, a hash of the ownership password can be taken and stored in AD directory service. This information can then be used to reset ownership of the TPM. This feature will be added in a further release.

In an enterprise environment, those GPO should be often applied in order to ensure administrators can unlock a protected volume and employers can read specific files following an incident (intrusion or various malicious acts for example).

Case #3: Local account and cached domain credentials

There aren't something really new here, a lot of tools are already dumping them without any problems. However we have choosed an uncommmon way to dump them, only few tools use this technique.

Hashes are extracted live from SAM and SECURITY hive in a proper way without code injection/service. In fact, we use native registry API, especially RegSaveKey() and RegLoadKey() functions which require SeBackup and SeRestore privileges. Next we mount SAM/REGISTRY hives on a different mount point and change all keys ACL in order to extend privileges to Administrator group and not LocalSystem only. That's why we choose to work on a backup to preserve system integrity.

Writing this tool was not a really difficult challenge, windows hashes and bitlocker information storage methodology are mostly well documented. However it's an interesting project to understand strange Microsoft's implementation and choices for each kind of storage:

- High level obfuscation techniques are used for local and domain accounts hashes: many constants, atypical registry value name, useless ciphering layer, hidden constants stored in registry Class attribute,...However, it can be easily defeated.
- Used algorithms differ sometimes between windows version and global credentials storage approach isn't regular. We can find exhaustively: RC4, MD5, MD4, SHA-256, AES-256, AES-128 and DES.
- Bitlocker information are stored in cleartext in AD domain services.

Project is still in beta test and we would really appreciate to have feedbacks or suggestions/comments about potential bugs.

Binary and source code are available on GitHub (GNU GPL v3 license):

Quarks PwDump v0.1b: https://github.com/quarkslab/quarkspwdump

For NTDS parsing technical details, you can also refer to MISC MAG #59 article by Thibault Leveslin. Finally, we would like to greet NTDS hash dump (Csaba Barta), libesedb and creddump authors for their excellent work.

If you would like to learn more about our security audits and explore how we can help you, get in touch with us!
