

# Диалоги о Impacket-secretsdump

 [habr.com/ru/articles/703332](https://habr.com/ru/articles/703332)

hotmanchester



[hotmanchester](#) 3 дек 2022 в 22:30

5 мин

6.6K

Тutorial

“Ты — спящий гений, степень пробуждения которого зависит от количества твоих осознанных действий.”

## Введение

Доброго всем времени суток! Зачастую, получается так, что изобретать свой велосипед нет никакой необходимости, и намного удобнее взять уже чье-то готовое решение, а уж тем более если это решение «прошло огонь, воду, и медные трубы». При этом, есть великий соблазн использовать его, не тратя времени на понимание принципа работы, руководствуясь истиной: «Я под капот не полезу, все равно я в этом ничего не понимаю! Работает? Ну и отлично!»

В сегодняшней статье речь пойдет о всем знакомой утилите из пакета **Impacket** под названием **secretsdump**. Безусловно, эта статья не раскроет **всех** теоретических аспектов работы данного скрипта, но нацелена на повышение осознанности при его использовании! В любом случае, нет предела совершенству, и, при желании, можно еще больше углубить свои знания, начав разбираться самостоятельно! (будем считать, что это «своего рода» трамплин)

[Ссылочка на сам проект](#), если вдруг у вас возникнет непреодолимый интерес покопаться в строках программного кода!

## И для чего нам все это?

«Утилита позволяет сдать хэши с удаленного Domain Controller без запуска на нем каких-либо агентов.» Конечно, это не единственный путь для достижения похожего результата. Но secretsdump любят и используют по всему миру, за возможность **удаленного** дампа чувствительной информации с контроллера домена!

[Ссылочка на плеяду методов дампа хэшей с Windows](#)

После корректной работы **impacket-secretsdump**, у нас появляется возможность осуществить атаки: Pass-the-Hash (для Lateral Movement), Golden Ticket (поставив галочку напротив пункта «осуществить persistence») да и вообще, по сути взять

домен под свой контроль и «чувствовать себя вольготно».

## А что делается то?

1. SAM и LSA secrets + cached creds считываются из реестра, затем «кусты реестра» [hives – о боже как я не хочу переводить] сохраняются на целевой системе в папке %SYSTEMROOT%\Temp и уже оттуда происходит считывание всех оставшихся данных.

«Говоря о программах для анализа реестра, под выражением «куст реестра» обычно понимается один из файлов SOFTWARE, SAM, SECURITY, SYSTEM и так далее. То есть кустом реестра является файл, в котором хранятся ключи корневого уровня или ключи корневого уровня.»

2. С NTDS.dit действует опционально:

2.1) Извлекает имена доменных пользователей и хэши их паролей + ключи Kerberos используя [MS-DRDS] вызов процедуры DRSGetNCChanges(), при этом репликация происходит только необходимых нам атрибутов (об этом чуть ниже)

2.2) Извлекает NTDS.dit с помощью vssadmin [подробнее по той самой ссылке про методы дампа хэшей], при этом его запуск происходит с помощью smbexec.

## Что получается на выходе?

Теперь настало время для детального анализа «выходной продукции» после работы (успешной) данной утилиты!

Сперва происходит извлечение ключа загрузки целевой машины. Он находится в **SYSTEM hive** в **HKLM\SYSTEM**. Конечно если мы хотим выполнить это действие вручную, то для этого мы просто выполняем команду **reg save HKLM\SYSTEM** после чего копируем его с целевой машину любым доступным для нас способом. Ключ загрузки понадобится позже для расшифровки файла **ntds.dit**.

```
[*] Target system bootKey: 0x1c1f9e2a97ca38f5906e2a9422e3df85
```

извлечение ключа загрузки целевой машины

Далее осуществляется дамп файла SAM находящегося в **HKLM\SAM**. The Security Account Manager (SAM) – это файл реестра, который хранит **пароли локальных пользователей компьютера**. Хранятся они, конечно, в зашифрованном виде.

```
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:eddeefbe1249d34d8d7a3db98e762bf3 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
```

дамп файла SAM

Когда пользователь домена проходит аутентификацию на контроллере домена при входе в систему, его учетные данные сохраняются на локальном компьютере по умолчанию (Cached Credentials: username:hash). Это позволяет пользователю

входить под своей учетной записью на компьютер из домена даже в том случае, если Контроллер недоступен (проблема с сетью, или DC выключен)

```
[*] Dumping cached domain login information (domain/username:hash)
```

Cached Credentials

Сохраненные пароли хранятся в ветке  
реестра **HKEY\_LOCAL\_MACHINE\Security\Cache**

## LSA secrets

LSA секреты - это специальное защищенное хранилище важной информации, которая используется системой Local Security Authority (LSA) в Windows. LSA предназначено для

управления локальной политикой безопасности системы, аудита, авторизации, входа пользователей в систему, хранения приватных данных. Чувствительные данные пользователей и системы хранятся в Secrets. Доступ ко всем секретным данным имеет только SYSTEM, но при желании в properties мы можем выставить доступ к ним, для того чтобы ознакомиться с содержимым папки Security.

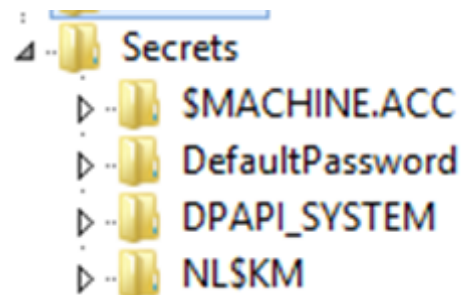
```
[*] Dumping LSA Secrets
```

Дамп LSA секретов

HKLM/Security/Policy/Secrets

Именно эти ветки реестра **impacket-secretsdump** успешно извлек и расшифровал.

Стоит сказать, что зашифрованы они на system bootkey.



содержимое LSA secrets

```
EVIL\SRV-DC01$:aad3b435b51404eeaad3b435b51404ee:b8f30c124145e3de53e7053acdef64e1 :::
```

Секрет от машинной учетной записи

В данном случае это секрет машинной учетной записи, который сменяется каждые 30 дней. Существует возможность использования хэша машинной учетной записи в атаках, связанных с делегацией.

DPAPI используется для защиты учетных данных пользователей, хранящихся на Windows хосте.

```
[*] DPAPI_SYSTEM
```

дамп DPAPI secrets

DPAPI используется при защите следующих персональных данных:

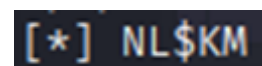
- Пароли и данные автозаполнения форм в Internet Explorer, Google Chrome

- Пароли учетных записей почты в Outlook, Windows Mail, Windows Mail, и т.д.
- Пароли учетных записей встроенного менеджера FTP
- Пароли доступа к общим папкам и ресурсам
- Пароли и ключи учетных записей беспроводной сети Ключи шифрования в Windows CardSpace и Windows Vault
- Пароли соединений удаленного доступа к рабочему столу, .NET Passport
- Приватные ключи Системы Шифрования Файлов (EFS), шифрования почты S-MIME, другие сертификаты пользователя, SSL/TLS в Internet Information Services EAP/TLS и 802.1x (VPN и WiFi аутентификация)
- Сетевые пароли в Credential Manager
- Персональные данные любого приложения, программно защищенные при помощи API функции CryptProtectData. Например, в Skype, Windows Rights Management Services, Windows Media, MSN messenger, Google Talk и др.

Список поражает, не так ли?!)

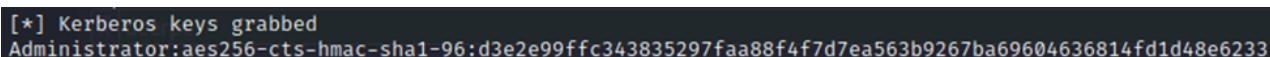
Использовать добытый DPAPI key можем, например, для того, чтобы расшифровать пароли, сохраненные в Google Chrome.

Секрет NL\$KM содержит ключ шифрования кэшированных паролей домена.



Кроме секретов LSA так же происходит дамп ключей Kerberos

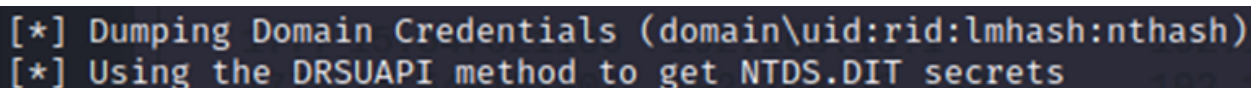
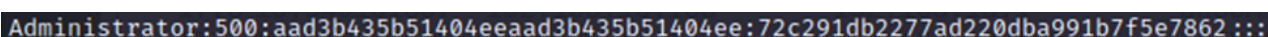
;) )



пример извлеченного ключа Kerberos

## NTDS.dit ?

Ну вот мы и дошли до дампа файла ntds.dit, в котором хранится информация об объектах домена, и содержащего в том числе и имена пользователей + хэши паролей.

Думаю, что всем уже понятно, что мы можем делать с NT hash, поэтому я лишь хотел бы пролить свет на **DRSUAPI method**. Как уже было отмечено ранее, по сути то, что делает **impacket-secretsdump** принято называть репликацией контроллера домена, а в контексте атаки на домен – **DCSync**.

Скажу лишь в двух словах как работает **DCSync**:

1. Происходит обнаружение контроллера домена по доменному имени (так как мы указали ip адрес, то поиски не долгие)
2. Происходит запрос на репликацию учетных данных пользователей к контроллеру домена через функцию **GetNCChanges()** [используя Directory Replication Service (DRS) Remote Protocol]

«Сетевая инфраструктура организации часто нуждается в наличии более чем одного контроллера домена для Active Directory. Таким образом для полноценного функционирования двух и более контроллеров домена необходимо, чтобы AD объекты могли быть реплицированы на все контроллеры домена»

Процесс репликации возложен на DRS Remote Protocol. DRSUAPI - API Microsoft которая реализует процесс репликации. Как и любая API у нее есть ряд функций.

```
DRSUAPI    198 DsBind request
DRSUAPI    210 DsBind response
DRSUAPI    174 DsGetDomainControllerInfo request
DRSUAPI    1090 DsGetDomainControllerInfo response
DRSUAPI    282 DsCrackNames request
DRSUAPI    290 DsCrackNames response
DRSUAPI    406 DsGetNCChanges request
DCERPC     4338 Response: call_id: 5, Fragment: 1st, Ctx: 0
```

Захваченный с помощью Wireshark трафик в процессе работы утилиты

В данном случае мы остановимся лишь на функции DsGetNCChanges(). Клиентский DC отправляет запрос DsGetNCChanges на сервер, когда первый хочет получить обновления объектов AD от второго. Ответ содержит набор обновлений, которые клиент должен применить к своей реплике NC. В случае если, набор обновлений слишком велик для одного ответного сообщения, то выполняется несколько запросов и ответов DsGetNCChanges. Этот процесс называется циклом репликации или просто циклом.

Подробнее о DCSync вы можете почитать [тут](#) и [тут](#)

## Заключение

---

Таким образом, разобравшись с выводом, и немного углубившись в то, что значит каждый символ, мы теперь по достоинству можем оценить тот функционал, который заложили создатели в secretsdump. Безусловного знания не бывает, и возможно, я упустил определенные вещи в данной статье (постарался обильно снабдить ссылками на полезные ресурсы для дальнейшего изучения).