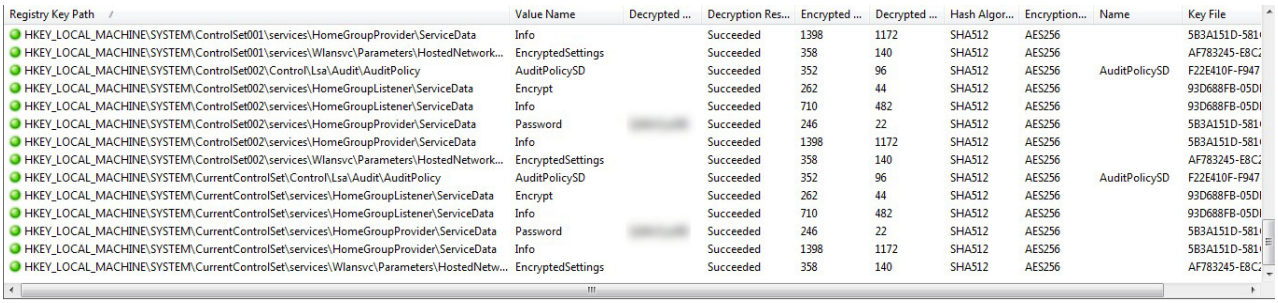
«Секретики» DPAPI. Взгляд на осла / Хабр

habr.com/ru/articles/437390

karelovao



Registry Key Path	Value Name	Decrypted ...	Decryption Res...	Encrypted ...	Decrypted ...	Hash Algor...	Encryption...	Name	Key File
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\HomeGroupProvider\ServiceData	Info		Succeeded	1398	1172	SHA512	AES256		5B3A151D-581
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\Wlansvc\Parameters\HostedNetwork...	EncryptedSettings		Succeeded	358	140	SHA512	AES256		AF783245-E8C
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Control\Lsa\Audit\AuditPolicy	AuditPolicySD		Succeeded	352	96	SHA512	AES256	AuditPolicySD	F22E410F-F947
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\services\HomeGroupListener\ServiceData	Encrypt		Succeeded	262	44	SHA512	AES256		93D688FB-05D
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\services\HomeGroupListener\ServiceData	Info		Succeeded	710	482	SHA512	AES256		93D688FB-05D
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\services\HomeGroupProvider\ServiceData	Password	[redacted]	Succeeded	246	22	SHA512	AES256		5B3A151D-581
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\services\HomeGroupProvider\ServiceData	Info		Succeeded	1398	1172	SHA512	AES256		5B3A151D-581
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\services\Wlansvc\Parameters\HostedNetwork...	EncryptedSettings		Succeeded	358	140	SHA512	AES256		AF783245-E8C
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Audit\AuditPolicy	AuditPolicySD		Succeeded	352	96	SHA512	AES256	AuditPolicySD	F22E410F-F947
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\HomeGroupListener\ServiceData	Encrypt		Succeeded	262	44	SHA512	AES256		93D688FB-05D
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\HomeGroupListener\ServiceData	Info		Succeeded	710	482	SHA512	AES256		93D688FB-05D
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\HomeGroupProvider\ServiceData	Password	[redacted]	Succeeded	246	22	SHA512	AES256		5B3A151D-581
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\HomeGroupProvider\ServiceData	Info		Succeeded	1398	1172	SHA512	AES256		5B3A151D-581
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Wlansvc\Parameters\HostedNetw...	EncryptedSettings		Succeeded	358	140	SHA512	AES256		AF783245-E8C

В дополнение к нашей прошлой статье про расшифровку DPAPI-блобов расскажем еще о двух случаях, с которыми нам пришлось столкнуться. Речь пойдет о сохраненных паролях в браузерах MS IE11 и Edge.

Стратегия остается прежней – будем все расшифровывать в режиме offline. Для этого необходимо забрать нужные файлы.

В зависимости от операционной системы (Windows 7 или выше) сохраненные пароли следует искать в двух местах:

В случае Windows 7 это ветка реестра

```
HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\IntelliForms\Storage2
```

В случае Windows 8 и выше — хранилище Windows Vault.

Так же следует отметить что на Windows 7 пароли http basic авторизации так же хранятся в Windows Vault, так что забрать его не помешает в любом случае.

Ну и по старой доброй традиции — все это конечно же шифруется через DPAPI-механизмы.

Теперь рассмотрим алгоритм расшифровки более подробно.

Windows 7 + IE11 (Edge)

Как уже упоминалось выше, пароли хранятся в реестре текущего пользователя и представляют из себя DPAPI-блобы, зашифрованные мастер-ключом пользователя.

Но есть важное отличие — при шифровании пароля применяется энтропия. Энтропия — это URL, по которому вводится пароль в формате `("https://url"+"\x00").lower().encode("utf-16-le")`.

Для расшифровки пароля нужно знать полный URL! Иначе никак.

Но, чтобы IE сам знал как расшифровывать пароль — этот URL хешируется и сохраняется в реестре в качестве имени ключа с DPAPI-blob.

Рассмотрим небольшой пример. Для сайта `https://rdot.org/forum/` сохраненный пароль будет выглядеть так:

```
A88E21329B5372B856CE238B79D1F28D8EA1FD359D    REG_BINARY
01000000D08C9DDF0115D1118C7A00C......BC310C51EE0F9B05D
```

где
A88… — это хешированный URL `https://rdot.org/forum/`
01000000D08C… — DPAPI-блоб, содержащий username и пароль

Алгоритм хеширования URL незамысловатый. Подробнее о нем можно почитать в ЦРУ-шных наработках Vault7.

На питоне он выглядит следующим образом:

```
import hashlib
url = "https://rdot.org/Forum/".lower() + "\x00"
url_utf_16_le = url.encode("utf-16-le")
sha1obj = hashlib.sha1(url_utf_16_le)
urldigest = sha1obj.digest()
checksum = 0
len(urldigest)
for abyte in urldigest:
   checksum = (checksum + (ord(abyte))) & 0xFF
hash = sha1obj.hexdigest().upper()
cksum = "%02X" % checksum
reg_value_name = "%s%s" % (hash, cksum)
print reg_value_name
```

Список последних 50-ти введенных URL можно почерпнуть так же из реестра:

```
HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\typedurls
```

Вернемся к примеру. Допустим нам необходимо найти в реестре сохраненный пароль от `https://rdot.org/forum/`.

Подставив значение URL в скрипт конвертации — мы получим значение

```
A88E21329B5372B856CE238B79D1F28D8EA1FD359D
```

Ключ с этим наименованием нам необходимо отыскать в реестре

```
req query "HKEY_USERS\<SID>\Software\Microsoft\Internet
Explorer\IntelliForms\Storage2"
```

Если такой ключ найден — его необходимо скопировать в файл в виде hex-значений (т.е. интерпретировав значение ключа как hex blob) и произвести расшифровку как DPAPI-blob с применением энтропии: `("https://rdot.org/forum/".lower() + "\x00").encode("utf-16-le")`

Для расшифровки можно воспользоваться dpapick, внеся соответствующие изменения для учета энтропии в расшифровке.

В файле examples/filegeneric.py вызов функции

```
 probe.try_decrypt_with_password(options.password, mkp, options.sid)
```

заменить на

```
probe.try_decrypt_with_password(options.password, mkp, options.sid, entropy=
("https://rdot.org/forum/".lower() + "\x00").encode("utf-16-le"))
```

и после этого вызвать dpapick как обычно:

```
 ./filegeneric.py --sid <SID> --masterkey <mk dir> --password <..> --inputfile
<dpapi blob from registry>
```

Если мастер-ключ расшифровался верно, то на выходе получим сохраненные логин и пароль (после некоторого количества служебных бинарных данных).

Windows 8.1 и выше

В случае сохранения паролей на Win8 и выше пароли от http форм, равно как и http basic авторизации, хранятся в Windows Vault. И что хорошо — вместе с паролем сохраняется и полный URL сайта, к которому он подходит.

Сам Vault шифруется двухступенчато — сначала весь блок данных шифруется AES'ом, а симметричный ключ для расшифровки шифруется DPAPI и сохраняется в файл. Полностью алгоритм по шифровке-расшифровке описан в статье ребят из Zena Forensics.

Ими же разработаны специальные декрипторы для Windows Vault на основе dpapick (dpapilab). Их можно взять на гите ZF или скачать форк с нашего гитхаба.

Хранилище Vault расположено в профиле пользователя:

```
C:\Users\<user>\AppData\Local\Microsoft\Vault\<GUID>\
```

Внутри файл .vpol — DPAPI-блоб, зашифрованный ключом пользователя, и хранящий AES-key для расшифровки .vcrd

Для расшифровки Vault необходимо запустить:

```
./vaultdec.py --masterkey <mk dir> --sid <SID> --password <pass>  <VAULT DIR>
```

Вместо пароля можно применить доменный ключ, как было показано в предыдущей статье. Так же следует отметить, что если в машина в домене и включена политика Credential Roaming, то данные Windows Vault будут храниться в ldap. Про это можно прочитать в первой нашей статье про DPAPI.

Маленькое дополнение: для корректной работы скрипта Вам скорее всего потребуется установить старые питоновские либы:

```
apt install python-construct.legacy
```

Шпаргалка

Для расшифровки паролей IE, Edge а так же сохраненных в Windows паролей Вам необходимо забрать:

каталог с Vault

```
c:\Users\<user>\AppData\Local\Microsoft\Vault\<GUID>\
```

каталог с мастер-ключами

```
c:\Users\<user>\AppData\roaming\microsoft\Protect\<SID>\
```

содержимое ключей реестра

```
HKEY_USERS\<SID>\Software\Microsoft\Internet Explorer\IntelliForms\Storage2
HKEY_USERS>\<SID>\Software\Microsoft\Internet Explorer\typedurls
```

Помимо этого, необходимо знать пароль пользователя или доменный dpapi backup-ключ для расшифровки без пароля.