HACKATON OWADE

Catedra DARS -UNIR

Universidad de Alcalá

LazaRus
clínica de datos

¿Por qué Owade?

- Necesidad de extraer las contraseñas que los principales aplicativos guardan en los sistemas Windows.
- Las contraseñas van cifradas con el login del usuario.
- Para sacarlas en la actualidad se necesitaría virtualizar una imagen forense para utilizar las típicas herramientas de sacar passwords sobre el sistema funcionado y con el usuario logado en el sisitema

Extracción de Contraseñas Aplicativos

- Las contraseñas de los principales aplicativos se pueden extraer de dos maneras diferentes:

1. **Online**: utilizando las funciones de la Data Protection API. Se necesita que el sistema operativo este funcionado y el usuario este logado en él. Herramientas de Security Xplode Tools
2. **Offline**: a partir de una imagen forense, sacar todos los datos necesarios. OWADE

Nociones básicas de Criptografía

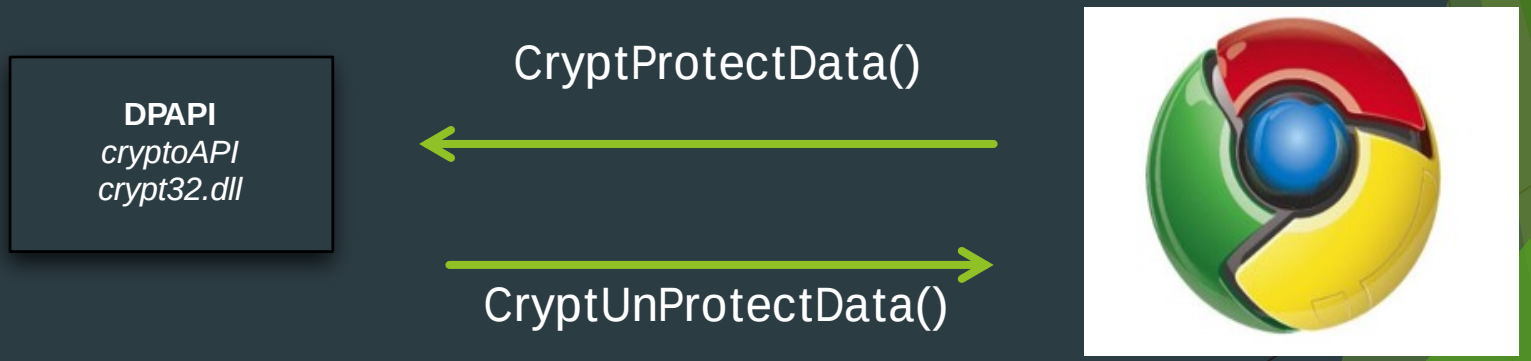
- **CBC (Cipher Block Chaining):** es un cifrador de bloque y necesita vector inicial IV.
- **SALT:** entropía que le añade a un cifrado
- **ITERACIONES**: número de rondas que aplica una función
- **SHA1:** función hash que produce un resumen de 160 bits (20bytes)
- **PBKDF v2** (Password Based Key Derivator Function): necesita una key, sal y número de iteraciones para poder crear una clave de cifrado.
- **DES:** Data Encryption Estándar (Cifrado Simétrico)

Nomenclatura de Windows

- **GUID**: Global Unique Identifier
- **SID**: Security Identifier
- **CREDHIST**: contiene hashes de passwords antiguos ->
  - X:\Users\[User]\AppData\Roaming\Microsoft\Protect\credhist
- **MASTER KEY**: 512 bits aleatorios
  - X:\Users\[User]\AppData\Roaming\Microsoft\Protect\[SID}
- **PREFERRED**: archivo donde está la MasterKey Actual con un timestamp.
  - X:\Users\[User]\AppData\Roaming\Microsoft\Protect\[SID}
- **SAM** (Security Accounts Manager) : contiene LM y NTLM hashes de los passwords
  - X:\Windows\System32\Config

DATA PROTECTION API - DPAPI

- Api de Windows para poder cifrar estructuras DATA BLOB.
- 3DES-CBC: cifra, descifra, cifra. Windows utiliza la versión de 3 claves.



Descifrando un Data BloB

- MasterKey File contiene las Masterkeys
- Windows renueva las MasterKeys cada 3 meses

Descifrando un Data BloB

Data blob

MasterKey
GUID

Master key file

3DES

Master Encryption
KEY

PBDKF

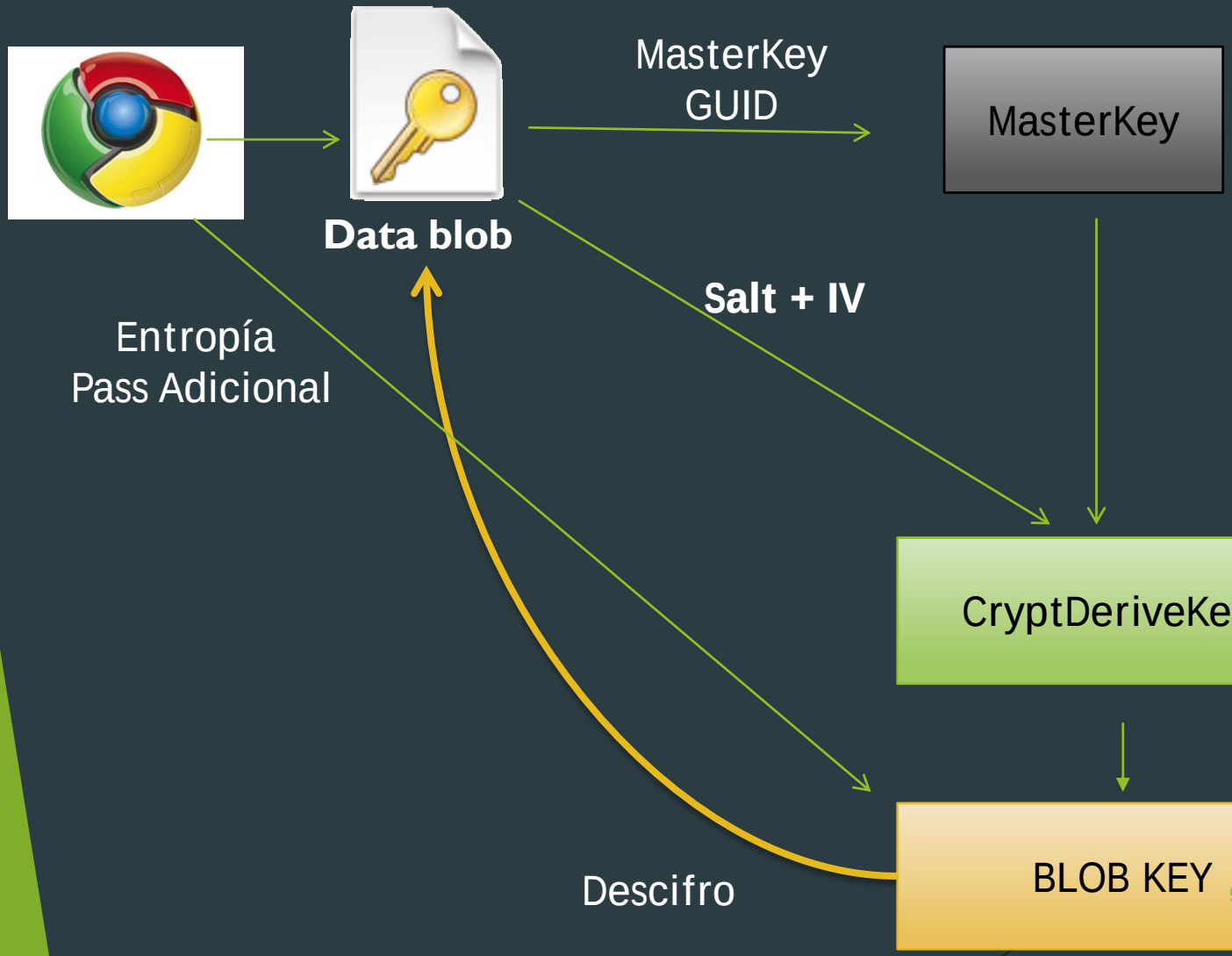
SHA1
(Password)

SID

MasterKey

Usuario

Descifrando un Data BloB



Descifrando un DataBlob

Todo deriva de las credenciales del usuario:

1. El password se puede sacar del fichero SAM
2. El password se pueda sacar del fichero de hibernación : hasheado con SHA1
3. El password se puede sacar del fichero CREDHIST que contiene los hashes de los passwords que tuvo un usuario, SHA1 y NTLM.

Descifrando un DataBlob de

Estructura

```
struct dpapi_blob_t {
DWORD cbProviders;
GUID *arrProviders; // Crypto Providers GUIDs
DWORD cbKeys;
GUID *arrKeys; // MarteKeys GUIDs
DWORD ppszDataDescrSize;

    WCHAR *ppszDataDescr;  //Descripción
    DWORD idCipherAlgo; // Id de Cifrado utilizado
    DWORD idHashAlgo; //Id de Hash utilizado
    BYTE *pbSalt; //Salt
    BYTE *pbCipher; //Datos Cifrados
    BYTE *pbHMAC; //HMAC de verificación
    };
```

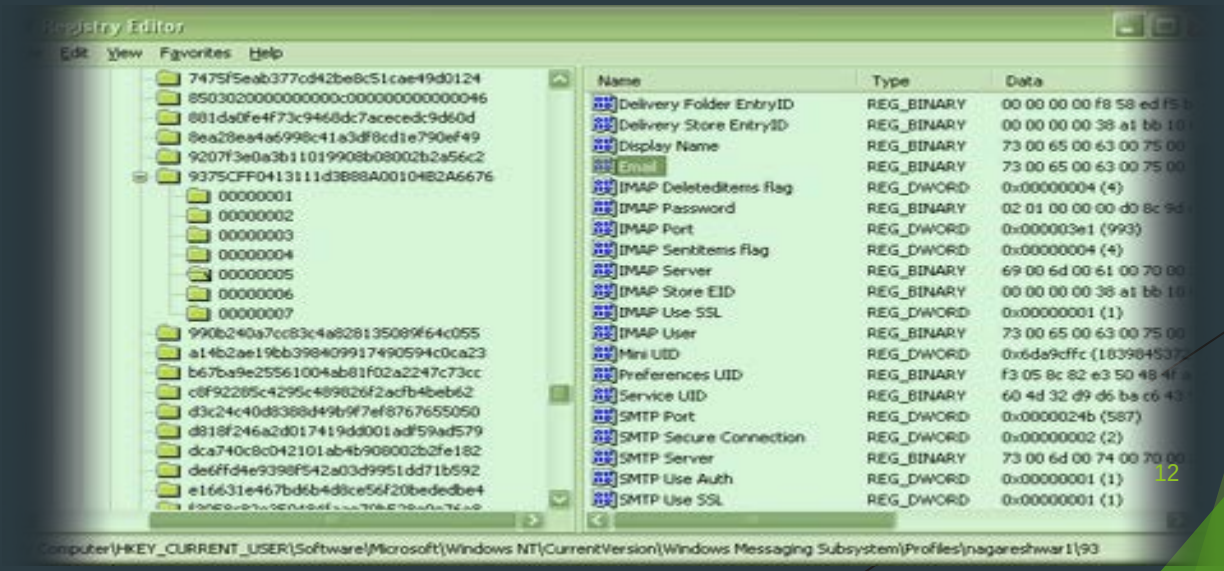
Microsoft Outlook

- Outlook almacena las credenciales en el registro:

HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows Messaging Subsystem\Profiles

- Outlook2013

HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\0003



Microsoft Outlook

- Data Blob parseada :

```
cbProviders: 1
arrProviders[0]: df9d8cd0-1501-11d1-8c7a-00c04fc297eb
cbKeys: 1
arrKeys[0]: 0c33334d-bb5c-4ad4-939d-6382a0e83c2a
dwDescriptionSize: 28
szDescription: POP3 Password
idCipherAlgo: 0x00006603
dwKeyLen: 168
dwSaltLen: 16
pbSalt: c284caa059f1a4e068d738686146dca8
idHashAlgo: 0x00008004
dwHashLen: 160
dwDataLen: 16
pbData: 2539035ae5e5e3e02b33bc936ae6c550
dwCipherLen: 24
pbCipher: dddb57889d1cc1fc6a51a91761e1561e14e330490e7adbb2
dwCrcLen: 20
pbCrc: 844b6abcb408ce127ddcbdc80027c0a42fa06f83
```

Chrome Cached Passwords

- Fichero SQLite **Login Data** en
  - C:\Documents and Settings\<username>\Local Settings\Application Data\Google\Chrome\User Data\Default \
  - C:\Users\<username>\Appdata\Local\Google\Chrome\User Data\Default

DPAPI
DATA BLOB

Table: logins

	origin_url	action_url	username_element	username_value	password_element	password_value
1	https://ssl.reddit.com/login	https://ssl.reddit.com/post/login	user	th_away123	passwd	r

iCloud Apple token decryption

- Fichero necesario:
  - com.apple.AOSKit.plist
  - Entropia del propio aplicativo:

```
APPLE_ENTROPY = (''
    '\x1D\xAC\xA8\xF8\xD3\xB8\x48\x3E\x48\x7D\x3E\x0A\x62\x07\xDD\x26'
    '\xE6\x67\x81\x03\xE7\xB2\x13\xA5\xB0\x79\xEE\x4F\x0F\x41\x15\xED'
    '\x7B\x14\x8C\xE5\x4B\x46\x0D\xC1\x8E\xFE\xD6\xE7\x27\x75\x06\x8B'
    '\x49\x00\xDC\x0F\x30\xA0\x9E\xFD\x09\x85\xF1\xC8\xAA\x75\xC1\x08'
    '\x05\x79\x01\xE2\x97\xD8\xAF\x80\x38\x60\x0B\x71\x0E\x68\x53\x77'
    '\x2F\x0F\x61\xF6\x1D\x8E\x8F\x5C\xB2\x3D\x21\x74\x40\x4B\xB5\x06'
    '\x6E\xAB\x7A\xBD\x8B\xA9\x7E\x32\x8F\x6E\x06\x24\xD9\x29\xA4\xA5'
    '\xBE\x26\x23\xFD\xEE\xF1\x4C\x0F\x74\x5E\x58\xFB\x91\x74\xEF\x91')
```

Mozilla Firefox

- Contraseñas almacenadas en el el perfil del usuario de Firefox X:\Users\{usuario}\AppData\Roaming\Mozilla\Firefox\Profiles\{random}.default
- logins.json
- signons.sqlite

Internet Explorer

- Passwords de autocompletar en el fichero de registro ntuser.dat
- DPAPI blob almacenada en el registro:
  - HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\IntelliForms\Storage2

Wifi Passwords

- Perfiles de las redes Wifi almacenados en:
  - C:/ProgramData/Microsoft/Wlansvc
- Los perfiles van cifrados con las MasterKey del Usuario SYSTEM
  - Masterkey System: X:/Windows/System32/Microsoft/Protect/S-1-5-18/User
- Necesitan el fichero registro SYSTEM y Security.

1.ONLINE: Data Protection API

```
DATA_BLOB blobEncrypted;
DATA_BLOB blobDecrypted;
blobEncrypted.pbData = &buf[1];
blobEncrypted.cbData = dwBufSize -1;

if (CryptUnprotectData(&blobEncrypted, 0, 0, 0, 0,
CRYPTPROTECT_UI_FORBIDDEN, &blobDecrypted))
    {
    wcout << "Decrypted Password: " << (wchar_t*)blobDecrypted.pbData;
    exit(0);
    } else
    {
    cout << "CryptUnprotectData error: " << GetLastError() << endl;
    }
```

DPAPI
cryptoAPI
crypt32.dll

CryptProtectData()

CryptUnProtectData()