-----Recap Introduction

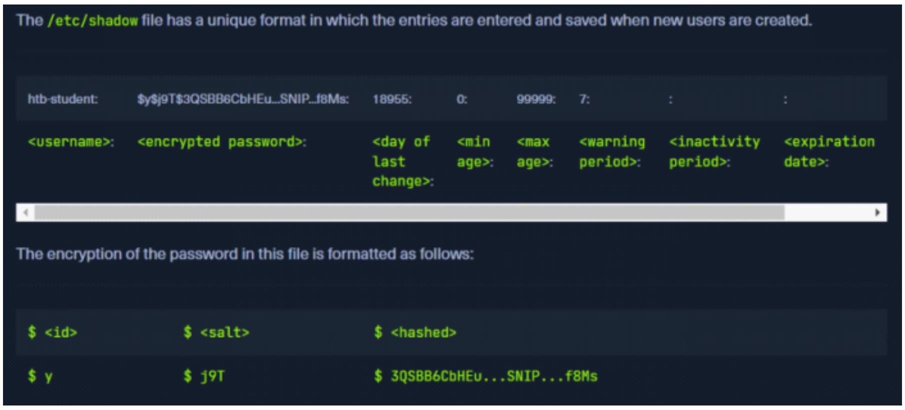
mercredi 9 octobre 2024 5:13 PM

https://attack.mitre.org/techniques/T1003/003/

Module	Content
Credential Storage	

Linux :

/etc/shadow => fields :



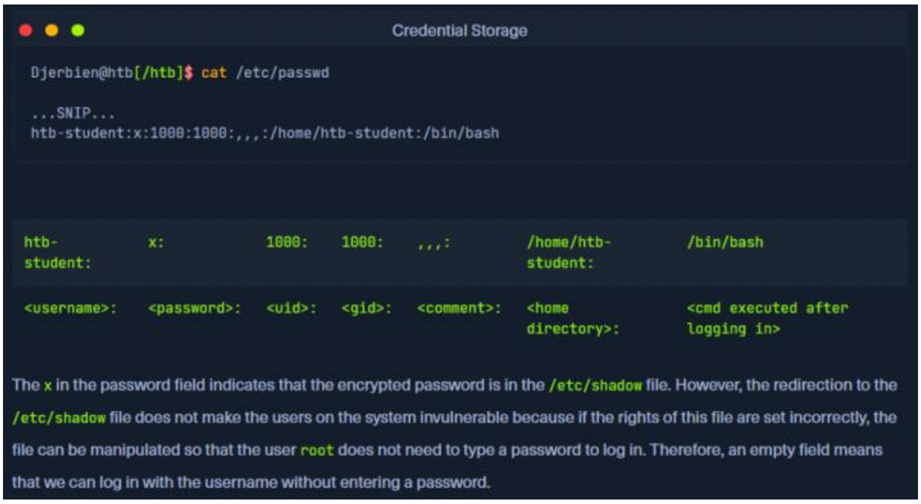
The `/etc/shadow` file has a unique format in which the entries are entered and saved when new users are created.

htb-student:	$y$j9T$3QSBB6CbHEu...SNIP...f8Ms:	18955:	0:	99999:	7:	:	:
`<username>:`	`<encrypted password>:`	`<day of last change>:`	`<min age>:`	`<max age>:`	`<warning period>:`	`<inactivity period>:`	`<expiration date>:`

The encryption of the password in this file is formatted as follows:

`$ <id>`	`$ <salt>`	`$ <hashed>`
`$ y`	`$ j9T`	`$ 3QSBB6CbHEu...SNIP...f8Ms`

/etc/passwd => fields



Credential Storage

```
Djerbien@htb[/htb]$ cat /etc/passwd

...SNIP...
htb-student:x:1000:1000:,,,:/home/htb-student:/bin/bash
```

`htb-student:`	`x:`	`1000:`	`1000:`	`,,,:`	`/home/htb-student:`	`/bin/bash`
`<username>:`	`<password>:`	`<uid>:`	`<gid>:`	`<comment>:`	`<home directory>:`	`<cmd executed after logging in>`

The `x` in the password field indicates that the encrypted password is in the `/etc/shadow` file. However, the redirection to the `/etc/shadow` file does not make the users on the system invulnerable because if the rights of this file are set incorrectly, the file can be manipulated so that the user `root` does not need to type a password to log in. Therefore, an empty field means that we can log in with the username without entering a password.

Windows

LSASS

%SystemRoot%\System32\Lsass.exe

This service is responsible for the local system security policy, user authentication, and sending security audit logs to the Event log.

SAM Database

%SystemRoot%/system32/config/SAM is mounted on HKLM/SAM.

- If the system has been assigned to a **workgroup**, it handles the **SAM database** locally and stores all existing users locally in this database.
- However, if the system has been joined to a **domain**, the Domain Controller (DC) must validate the credentials from the Active Directory database (ntds.dit), which is stored in **%SystemRoot%\ntds.dit.**

Credential Manager

```
PS  C:\Users\[Username]\AppData\Local\Microsoft\[Vault/Credentials]\
```

Credential Manager is a feature built-in to all Windows operating systems that **allows users to save the credentials they use to access various network resources and websites**. Saved credentials are stored based on user profiles **in each user's Credential Locker**. Credentials are **encrypted** and **stored** in the path above

There are various methods to decrypt credentials saved using Credential Manager. We will practice hands-on with some of these methods in this module.

NITDS

Each Domain Controller hosts a file called NTDS.dit that is kept synchronized across all Domain Controllers with the exception of Read-Only Domain Controllers. NTDS.dit is a database file that stores the data in Active Directory, including but not limited to:

- User accounts (**username** & **password hash)**
- Group accounts
- Computer accounts
- Group policy objects

John the ripper

Cracking Modes

Single Crack Mode is one of the most common John modes used when attempting to crack passwords using a single password list. **It is a brute-force attack**, meaning **all passwords on the list are tried, one by one, until the correct one is found**.

```
$ john --format=<hash_type> <hash or hash_file>
$ john --format=sha256       hashes_to_crack.txt
```

When we run the command, John will read the hashes from the specified file, and then it will try to crack them **by comparing them to the words in its built-in wordlist** and **any additional wordlists** specified with the **--wordlist** option. Additionally, It will use any rules set with the **--rules** option (if any rules are given) to generate further candidate passwords.

John will output the cracked passwords to the console and the file **"john.pot" (~/.john/john.pot)** to the current user's home directory

Hash Format	**Example Command**	**Description**
Afs	john --format=afs hashes_to_crack.txt	AFS (Andrew File System) password hashes
Bfegg	john --format=bfegg hashes_to_crack.txt	bfegg hashes used in Eggdrop IRC bots
Bf	john --format=bf hashes_to_crack.txt	Blowfish-based crypt(3) hashes
Bsdi	john --format=bsdi hashes_to_crack.txt	BSDi crypt(3) hashes
crypt(3)	john --format=crypt hashes_to_crack.txt	Traditional Unix crypt(3) hashes
Des	john --format=des hashes_to_crack.txt	Traditional DES-based crypt(3) hashes
Dmd5	john --format=dmd5 hashes_to_crack.txt	DMD5 (Dragonfly BSD MD5) password hashes
Dominosec	john --format=dominosec hashes_to_crack.txt	IBM Lotus Domino 6/7 password hashes
EPiServer SID hashes	john --format=episerver hashes_to_crack.txt	EPiServer SID (Security Identifier) password hashes
Hdaa	john --format=hdaa hashes_to_crack.txt	hdaa password hashes used in Openwall GNU/Linux
hmac-md5	john --format=hmac-md5 hashes_to_crack.txt	hmac-md5 password hashes

Hmailserver	john --format=hmailserver hashes_to_crack.txt	hmailserver password hashes
Ipb2	john --format=ipb2 hashes_to_crack.txt	Invision Power Board 2 password hashes
Krb4	john --format=krb4 hashes_to_crack.txt	Kerberos 4 password hashes
Krb5	john --format=krb5 hashes_to_crack.txt	Kerberos 5 password hashes
LM	john --format=LM hashes_to_crack.txt	LM (Lan Manager) password hashes
lotus5john	--format=lotus5 hashes_to_crack.txt	Lotus Notes/Domino 5 password hashes
Mscash	john --format=mscash hashes_to_crack.txt	MS Cache password hashes
Mscash2	john --format=mscash2 hashes_to_crack.txt	MS Cache v2 password hashes
Mschapv2	john --format=mschapv2 hashes_to_crack.txt	MS CHAP v2 password hashes
Mskrb5	john --format=mskrb5 hashes_to_crack.txt	MS Kerberos 5 password hashes
Mssql05	john --format=mssql05 hashes_to_crack.txt	MS SQL 2005 password hashes
Mssql	john --format=mssql hashes_to_crack.txt	MS SQL password hashes
mysql-fast	john --format=mysql-fast hashes_to_crack.txt	MySQL fast password hashes
Mysql	john --format=mysql hashes_to_crack.txt	MySQL password hashes
mysql-sha1	john --format=mysql-sha1 hashes_to_crack.txt	MySQL SHA1 password hashes
NETLM	john --format=netlm hashes_to_crack.txt	NETLM (NT LAN Manager) password hashes
NETLMv2	john --format=netlmv2 hashes_to_crack.txt	NETLMv2 (NT LAN Manager version 2) password hashes
NETNTLM	john --format=netntlm hashes_to_crack.txt	NETNTLM (NT LAN Manager) password hashes
NETNTLMv2	john --format=netntlmv2 hashes_to_crack.txt	NETNTLMv2 (NT LAN Manager version 2) password hashes
NEThalfLM	john --format=nethalflm hashes_to_crack.txt	NEThalfLM (NT LAN Manager) password hashes
Md5ns	john --format=md5ns hashes_to_crack.txt	md5ns (MD5 namespace) password hashes
Nsldap	john --format=nsldap hashes_to_crack.txt	nsldap (OpenLDAP SHA) password hashes
Ssha	john --format=ssha hashes_to_crack.txt	ssha (Salted SHA) password hashes
NT	john --format=nt hashes_to_crack.txt	NT (Windows NT) password hashes
Openssha	john --format=openssha hashes_to_crack.txt	OPENSSH private key password hashes
Oracle11	john --format=oracle11 hashes_to_crack.txt	Oracle 11 password hashes
Oracle	john --format=oracle hashes_to_crack.txt	Oracle password hashes
Pdf	john --format=pdf hashes_to_crack.txt	PDF (Portable Document Format) password hashes
phpass-md5	john --format=phpass-md5 hashes_to_crack.txt	PHPass-MD5 (Portable PHP password hashing framework) password hashes
Phps	john --format=phps hashes_to_crack.txt	PHPS password hashes
pix-md5	john --format=pix-md5 hashes_to_crack.txt	Cisco PIX MD5 password hashes
Po	john --format=po hashes_to_crack.txt	Po (Sybase SQL Anywhere) password hashes
Rar	john --format=rar hashes_to_crack.txt	RAR (WinRAR) password hashes
raw-md4	john --format=raw-md4 hashes_to_crack.txt	Raw MD4 password hashes
raw-md5	john --format=raw-md5 hashes_to_crack.txt	Raw MD5 password hashes
raw-md5-unicode	john --format=raw-md5-unicode hashes_to_crack.txt	Raw MD5 Unicode password hashes
raw-sha1	john --format=raw-sha1 hashes_to_crack.txt	Raw SHA1 password hashes
raw-sha224	john --format=raw-sha224 hashes_to_crack.txt	Raw SHA224 password hashes
raw-sha256	john --format=raw-sha256 hashes_to_crack.txt	Raw SHA256 password hashes
raw-sha384	john --format=raw-sha384 hashes_to_crack.txt	Raw SHA384 password hashes

raw-sha512	john --format=raw-sha512 hashes_to_crack.txt	Raw SHA512 password hashes
salted-sha	john --format=salted-sha hashes_to_crack.txt	Salted SHA password hashes
Sapb	john --format=sapb hashes_to_crack.txt	SAP CODVN B (BCODE) password hashes
Sapg	john --format=sapg hashes_to_crack.txt	SAP CODVN G (PASSCODE) password hashes
sha1-gen	john --format=sha1-gen hashes_to_crack.txt	Generic SHA1 password hashes
Skey	john --format=skey hashes_to_crack.txt	S/Key (One-time password) hashes
Ssh	john --format=ssh hashes_to_crack.txt	SSH (Secure Shell) password hashes
Sybasease	john --format=sybasease hashes_to_crack.txt	Sybase ASE password hashes
Xsha	john --format=xsha hashes_to_crack.txt	xsha (Extended SHA) password hashes
Zip	john --format=zip hashes_to_crack.txt	ZIP (WinZip) password hashes

Tool	**Description**
Pdf2john	Converts **PDF documents** for John
Ssh2john	Converts **SSH private keys** for John
Mscash2john	Converts MS Cash hashes for John
Keychain2john	Converts OS X keychain files for John
Rar2john	Converts **RAR archives** for John
Pfx2john	Converts **PKCS#12 files** for John
truecrypt_volume2john	Converts TrueCrypt volumes for John
Keepass2john	Converts KeePass databases for John
Vncpcap2john	Converts VNC PCAP files for John
Putty2john	Converts PuTTY private keys for John
Zip2john	Converts **ZIP archives** for John
Hccap2john	Converts **WPA/WPA2 handshake captures** for John
Office2john	Converts MS Office documents for John
Wpa2john	Converts **WPA/WPA2 handshakes** for John

More tools can be found using ;

```
   $ locate *2john*
```

```
$ <tool> <file_to_crack> > file.hash

$ pdf2john server_doc.pdf > server_doc.hash

$ john server_doc.hash
                # OR
$ john --wordlist=<wordlist.txt> server_doc.hash
```

.