Ethical Hacking mit Python und Kali Linux

Kali Linux ist DAS Hacker Linux mit allen nötigen Tools um einen vernünftigen Pentest zu machen.
Know your Enemy

Heise Security Tips

Nmap

nmap -O ip-adresse # gibt unter anderem das Betriebssystem wieder
nmap -sV ip-adresse # gibt Versionen wieder

Subdomains finden

Subfinder findet alles

docker pull projectdiscovery/subfinder
docker run projectdiscovery/subfinder -d example.com

Metasploit

Burpsuite

Netcat

Social-Engineering Toolkit

Link Simulating Basic Attack

Nikto

Scanning von Websiten

John the ripper

Download

Filtermasken erstellen

Filtermasken

Zun‰chst muss man aus einer Datei, die mit einem Passwort gesch¸tzt ist, den Hashwert herausbekommen. Das funktioniert mit

`zip2john verschluesselt.zip > hast.txt` f¸r ein Zip File

`pdf2john verschluesselt.pdf > hast.txt` sollte eigentlich das Passwort einer Zip-Datei herausfinden

Danach wird der Hashwert mit **John the Ripper** gekrackt:

`john hash.txt`

pdf2john funktionert nicht

```
git clone "https://github.com/magnumripper/JohnTheRipper.git" && cd JohnTheRipper/src && ./configure && sudo make -s clean && sudo make -sj4
```

John Maske

für nur buchstaben und Zahlen

`--mask=[A-Za-z0-9] --max-length=6`

Cheatsheet

oder als PDF jtr-cheat-sheet.pdf

John Resources

- John jumbo dev release
- John binaries
- John docs
- John docs
- Password Analysis and Cracking Kit
- Mangling Rules Generation

John Installation

Genau Beschreibung der Installation

```
git clone https://github.com/openwall/john -b bleeding-jumbo /data/tools/john ; cd /data/tools/john/src/ ;
./configure && make -s clean && make -sj4 ; cd ~
```

John Modes

- Wordlist mode (dictionary attack) - `john --wordlist=<wordlist> <hash>`
- Mangling rules mode - `john --wordlist=<wordlist> --rules:<rulename> <hash>`
- Incremental mode - `john --incremental <hash>`
- External mode - `john --external:<rulename> <hash>`
- Loopback mode (use .pot files) - `john --loopback <hash>`
- Mask mode - `john --mask=?1?1?1?1?1?1?1?1 -1=[A-Z] -min-len=8 <hash>`
- Markov mode - `calc_stat <wordlist> markovstats` `john -markov:200 -max-len:12 --mkv-stats=markovstats <hash>`

- Prince mode - `john --prince=<wordlist> <hash>`

Refer the link for more examples.

CPU and GPU options

- List opencl devices - `john --list=opencl-devices`
- List formats supported by opencl - `john --list=formats --format=opencl`
- Use multiple CPU - `john hashes --wordlist:<wordlist> --rules:<rulename> --dev=2 --fork=4`
- Use multiple GPU - `john hashes --format:<openclformat> --wordlist:<wordlist> --rules:<rulename> --dev=0,1 --fork=2`

Rules

- Single
- wordlist
- Extra
- Jumbo (Single, wordlist and Extra)
- KoreLogic
- All (Single, wordlist, Extra and KoreLogic)

Incremental modes

- Lower (26 char)
- Alpha (52 char)
- Digits (10 char)
- Alnum (62 char)

New rule

```
[List.Rules:Tryout]
l                   [convert to lowercase]
u                   [convert to uppercase]
c                   [capitalize]
l r                 [lowercase and reverse (palindrome)]
l Az"2015"          [lowercase and append "2015" at end of word]
l A0"2015"          [lowercase and prepend "2015" at end of word]
d                   [duplicate]
A0"#"Az"#"          [append and prepend "#"]
```

- Display password candidates - `john --wordlist=<wordlist> --stdout --rules:Tryout`
- Generate password candidates - `john --wordlist=<wordlist> --stdout=8 --rules:Tryout`

Other rules

```
C     [lowercase first char, uppercase rest]
t     [toggle case of all chars]
TN    [toggle case of char in position N]
r     [reverse word - test123 -> 321tset]
d     [duplicate word - test123 -> test123test123]
f     [reflect word - test123 -> test123321tset]
{     [rotate word left - test123 -> est123t]
}     [rotate word right - test123 -> 3test12]
$X    [append word with X]
^X    [prefix word with X]
[     [remove first char]
]     [remove last char]
DN    [delete char in posision N]
xNM   [extract from position N till M chars]
iNX   [insert X in place of N and shift rest right]
oNX   [overwrite N with X]
S     [shift case - test123 -> TEST!@#]
V     [lowercase vowels, uppercase consonents - test123 -> TeST123]
R     [shift each char right, using keyboard key - test123 -> yrdy234]
L     [shift each char left, using keyboard key - test123 -> rwar012]
<N    [reject words unless less than length N]
>N    [reject words unless greater than length N]
N     [truncate to length N]
```

New charset

```
john --make-charset=set.char
```

Create `john.conf` with character set config.

```
# Incremental modes
[Incremental:charset]
```

```
File = $JOHN/set.char
MinLen = 0
MaxLen = 30
CharCount = 80
```

```
john --incremental=charset <hash>
```

Wordlists

- Sort wordlist - `tr A-Z a-z < <wordlist> | sort -u > <new-wordlist>`
- Generate wordlist using POT - `cut -d: -f2 john.pot | sort -u > pot.dict`
- Generate candidate pwd for slow hash - `john --wordlist=<wordlist> --stdout --rules:Jumbo | unique -mem=25 <unique-wordlist>`

External mode

- Create complex password list - link
- Generate wordlist according to complexity filter - `./john --wordlist=<wordlist> --stdout --external:<filter> > <filtered-wordlist>`
- Use adjacent keys on `keyboard` - `john --external:Keyboard <hash>`

Misc Options

- Hidden options - `john --list=hidden-options`
- Display guesses - `john --incremental:Alpha -stdout -session=s1`
- Generate guesses with external programs - `crunch 1 6 abcdefg | ./john hashes -stdin -session=s1`
- Save session - `john hashes -session=name`
- Restore session - `john --restore:name`
- Show cracked passwords - `john hashes --pot=<pot> --show`

Dictionaries

- Generate wordlist from wikipedia - `wget https://raw.githubusercontent.com/zombiesam/wikigen/master/wwg.py ; python wwg.py -u http://pt.wikipedia.org/wiki/Fernando_Pessoa -t 5 -o fernandopessoa -m3`
- Aspell dictionary - `apt-get install aspell-es` `aspell dump dicts` `aspell -d es dump master | aspell -l es expand | awk 1 RS=" |\n" > aspell.dic`

Version #3
Erstellt: 19 October 2022 10:52:19 von Herr_admin
Zuletzt aktualisiert: 19 October 2022 11:18:20 von hermann

Revision #4
Created 21 March 2023 07:55:14 by Hermann
Updated 7 July 2025 22:38:22 by Hermann