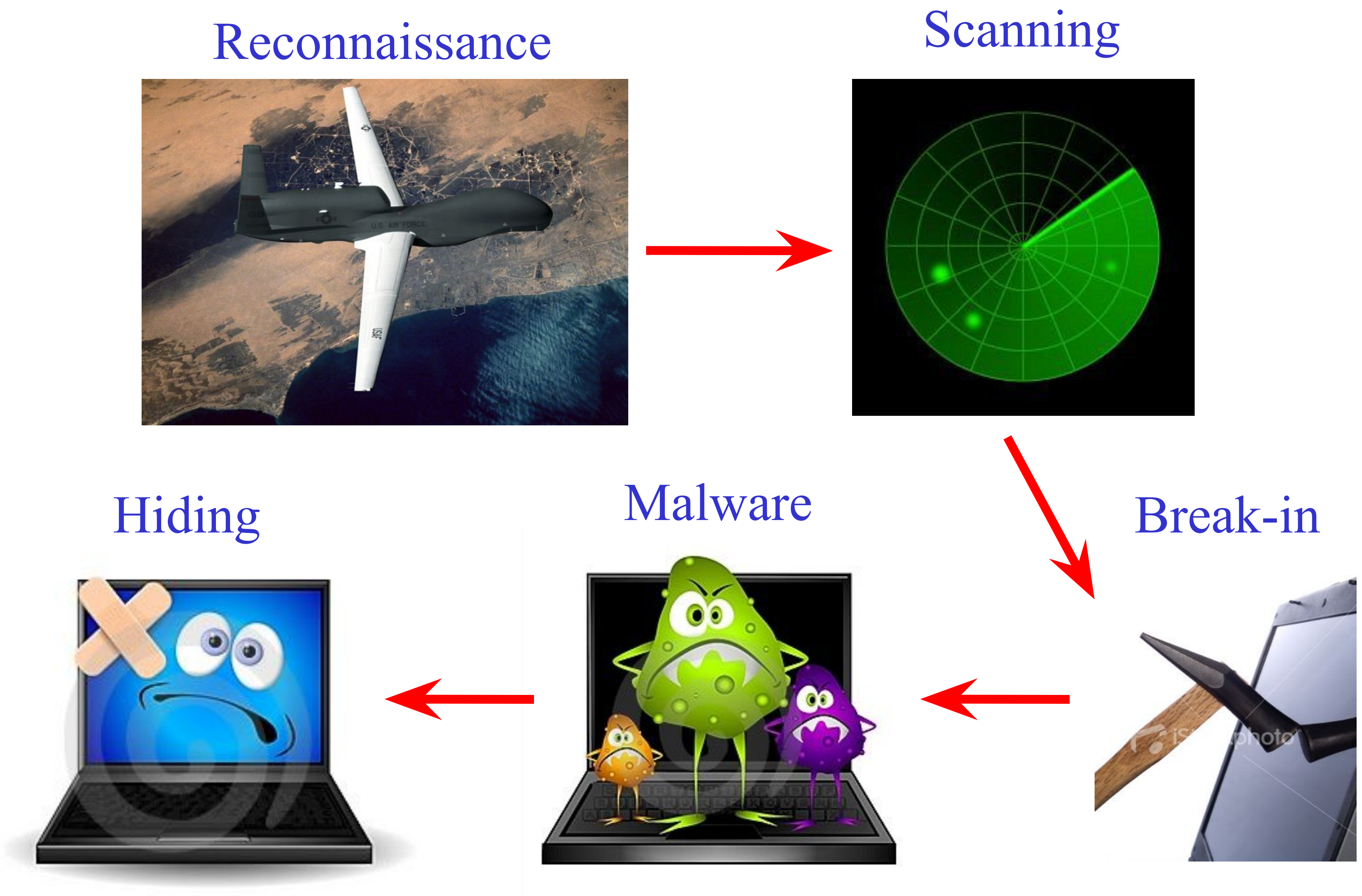
CS4238: Computer Security Practice

Lecture 5: Password Attacks, Binary Analysis and Fuzzing

Slides by: LIANG Zhenkai,
Roland YAP & SUFATRIO

Big Picture of Attacks



Reconnaissance

Scanning

Break-in

Malware

Hiding

Progress Overview

- System attacks and defenses:
  - Reconnaissance
  - Scanning
  - Automated vulnerability finding
  - Automated exploitation
  - *Vulnerability discovery, e.g. fuzzing*
  - Attacks to gain access, e.g., buffer overflow attacks and defenses
  - **Maintaining access, e.g. password attacks, malware planting**

Password Attacks

Background

- *Question: Why password attacks?*
  - Suppose we already can own a host
  - Possible next step(s)?
  - Importance of password file:
    on the exploited host, other hosts
- UNIX/Linux user & password files
- https://wiki.archlinux.org/index.php/Su

Authentication Mechanisms

- Something you know: password, PIN
- Something you have: smart card, private key, phone
- Something you are: biometrics
- Somewhere you are: location-limited channels
- Someone you know: social authentication
- Some system vouches for you: single sign-on, PKI certificate

Guessing Passwords

- Using **default** password:
  - http://www.phenoelit-us.org/dpl/dpl.html
- Password guessing via **login/online attacks**:
  - Some tools: Brutus, THC Hydra
  - Guess passwords from a dictionary, list of weak passwords
  - Support many login protocols
  - Slow, a few seconds for each login attempt
  - May result in **account lockouts**

Unix Passwords

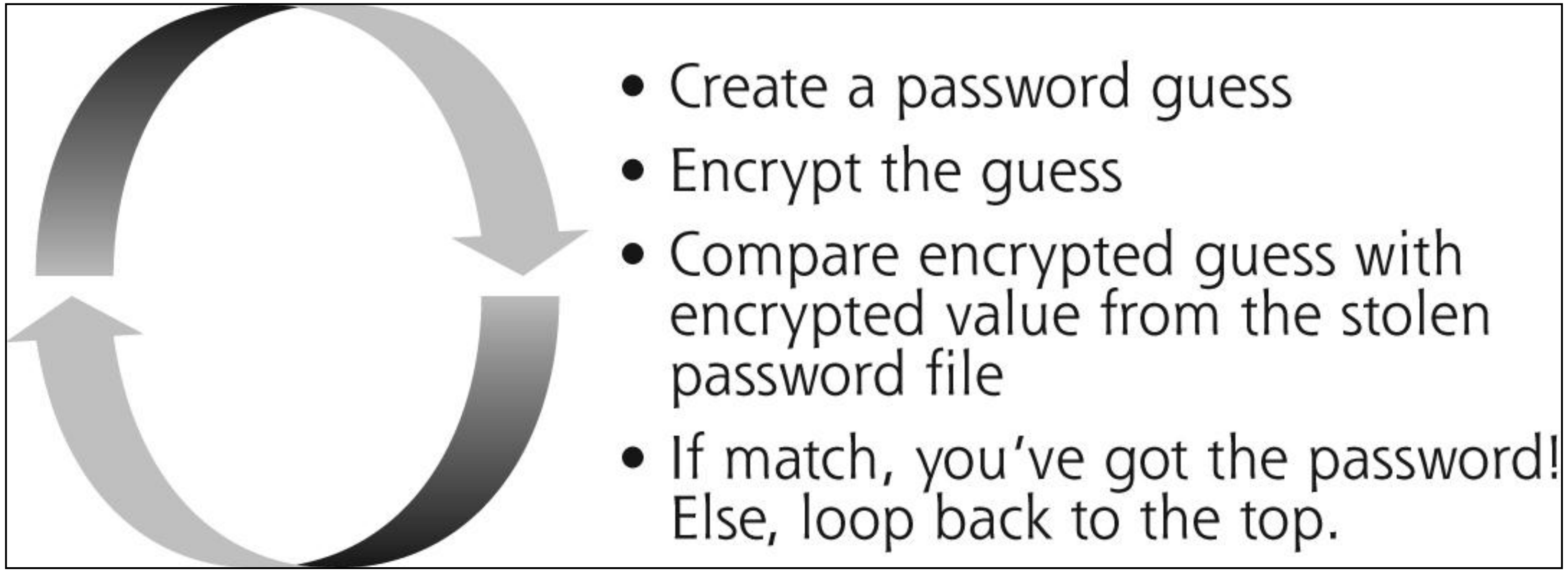
- **Public** file: `/etc/passwd`
  - See `man 5 passwd`
  - Entry’s fields: login name, **x**, uid, gid, home directory, shell
- **Private/protected** file: `/etc/shadow`
  - No access by non-root users
  - See `man 5 shadow` for the fields
  - Entry’s fields (separated by “:”):
    login name, hashed password, date of last password change, minimum password age, maximum password age, password warning period, password inactivity period, account expiration date, reserved field

Sample Shadow Entry

- user1:$6$yonrs//S$bUdht9fglwJW0LduAxEJpcExtMfKokFMJoT8tGkKLx5xFGJk22/trPstOHXr4PdBID0AV1xko5LfFVDwW.aJS.:17275:0:99999:7:::
- The second (hashed password) field:
  - Find its format information: `man 3 crypt`
  - Format used: `$id$salt$hashed-key`
  - `id`: ID of the hash-method used (1=MD5, 5=SHA-256, 6= SHA-512, …)
  - `salt`: up to 16 chars drawn from the set [a-zA-Z0-9./]
  - `hashed-key`: hash of the password (e.g. 22 chars for MD5, 43 chars for SHA-256, 86 chars for SHA-512)
- Unshadow: replace x in `passwd` with the hash password

Cracking Passwords

- Prerequisite:
  attacker has access to password database
- See it by yourself: `/etc/shadow`



- Create a password guess
- Encrypt the guess
- Compare encrypted guess with encrypted value from the stolen password file
- If match, you've got the password! Else, loop back to the top.

- **Note**: Password is **hashed**, and *not* **encrypted**

Password-Cracking Tools

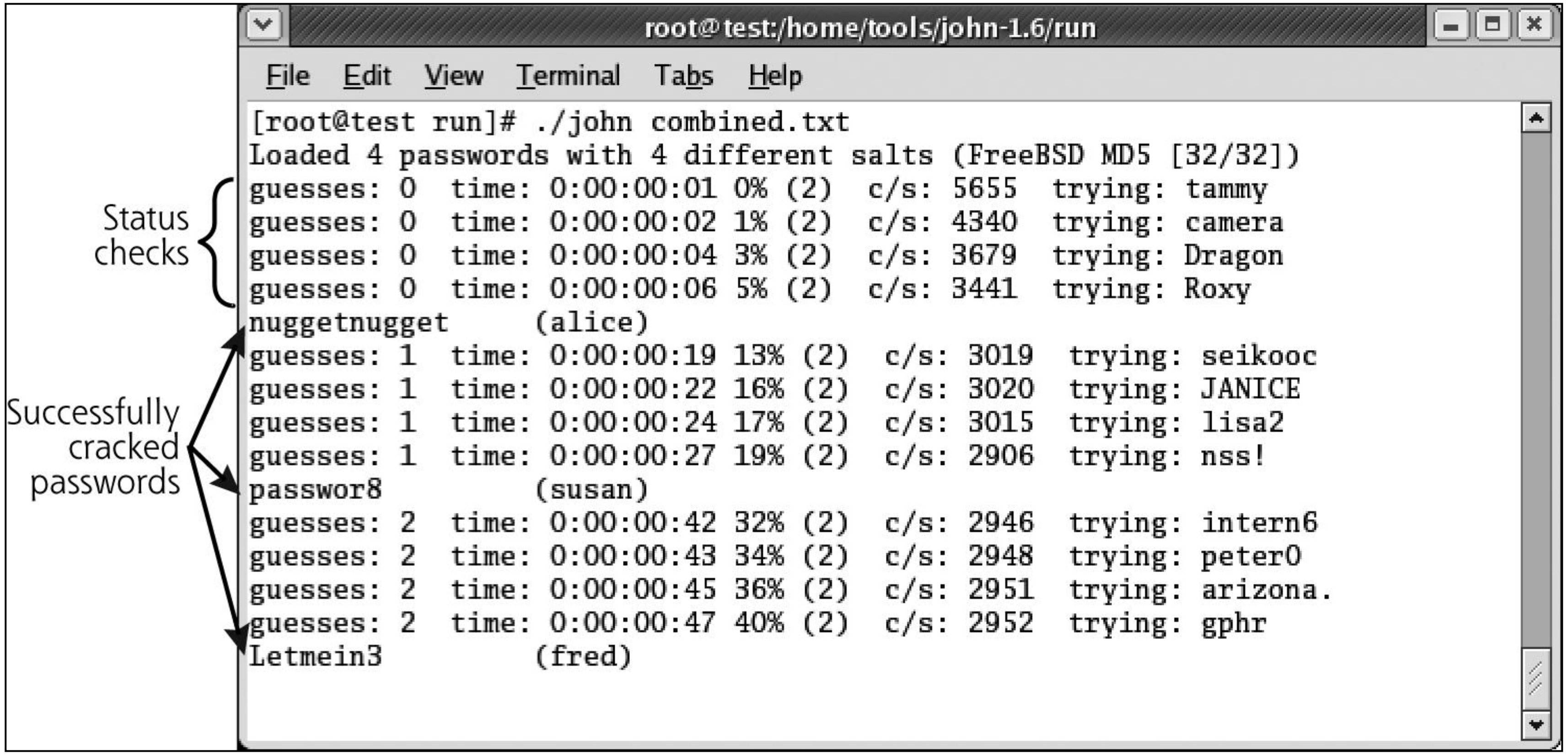
- Forming **password guesses**:
  - From dictionaries: **dictionary attack**
  - Brute force: **brute force attack**
  - Hybrid approaches combining both: **hybrid attack**
- Popular **password crackers**
  - Cain
  - **John the Ripper (JtR)**
  - Pandora
  - LC5

John the Ripper (JtR)

- A free, high quality password cracker
- Written by Solar Designer and team
- Run on many operating systems:
  - Linux, UNIX, Windows, DOS
- Crack password of various UNIX variants
  - Crack Windows password through plugin
- Create a hidden folder `.john`:
  - File `.john/john.pot:` stores **cracked entries**
  - **Delete** it **after scanning** your own system!

Example of John the Ripper

```
unshadow /etc/passwd /etc/shadow > combined.txt
john combined.txt
```



root@test:/home/tools/john-1.6/run
File Edit View Terminal Tabs Help
Status checks
Successfully cracked passwords

```
[root@test run]# ./john combined.txt
Loaded 4 passwords with 4 different salts (FreeBSD MD5 [32/32])
guesses: 0  time: 0:00:00:01 0% (2)  c/s: 5655  trying: tammy
guesses: 0  time: 0:00:00:02 1% (2)  c/s: 4340  trying: camera
guesses: 0  time: 0:00:00:04 3% (2)  c/s: 3679  trying: Dragon
guesses: 0  time: 0:00:00:06 5% (2)  c/s: 3441  trying: Roxy
nuggetnugget     (alice)
guesses: 1  time: 0:00:00:19 13% (2)  c/s: 3019  trying: seikooc
guesses: 1  time: 0:00:00:22 16% (2)  c/s: 3020  trying: JANICE
guesses: 1  time: 0:00:00:24 17% (2)  c/s: 3015  trying: lisa2
guesses: 1  time: 0:00:00:27 19% (2)  c/s: 2906  trying: nss!
passwor8         (susan)
guesses: 2  time: 0:00:00:42 32% (2)  c/s: 2946  trying: intern6
guesses: 2  time: 0:00:00:43 34% (2)  c/s: 2948  trying: peter0
guesses: 2  time: 0:00:00:45 36% (2)  c/s: 2951  trying: arizona.
guesses: 2  time: 0:00:00:47 40% (2)  c/s: 2952  trying: gphr
Letmein3         (fred)
```

Using John the Ripper

- Some useful John's **parameters**:
  - `-h`: help
  - `--users=`<*user*>: crack the password of **user**
  - `--wordlist=`<*file*>: use the given **wordlist file**
  - `--show`[`=LEFT`]: show cracked/uncracked passwords
- Oher popular password **dictionary files**:
  - Rockyou, Cain & Abel, Hotmail, …
  - See: https://wiki.skullsecurity.org/Passwords
- Can also generate a *custom* word list:
  gather words from a target site's home page

Cracking Modes of John the Ripper

- John has different ***cracking modes***:
  - Specify the desired mode using its flag
- John's **default order** of cracking modes:
  - ***Single-crack*** mode
  - ***Wordlist*** mode
  - ***Incremental*** mode

Cracking Modes of John the Ripper

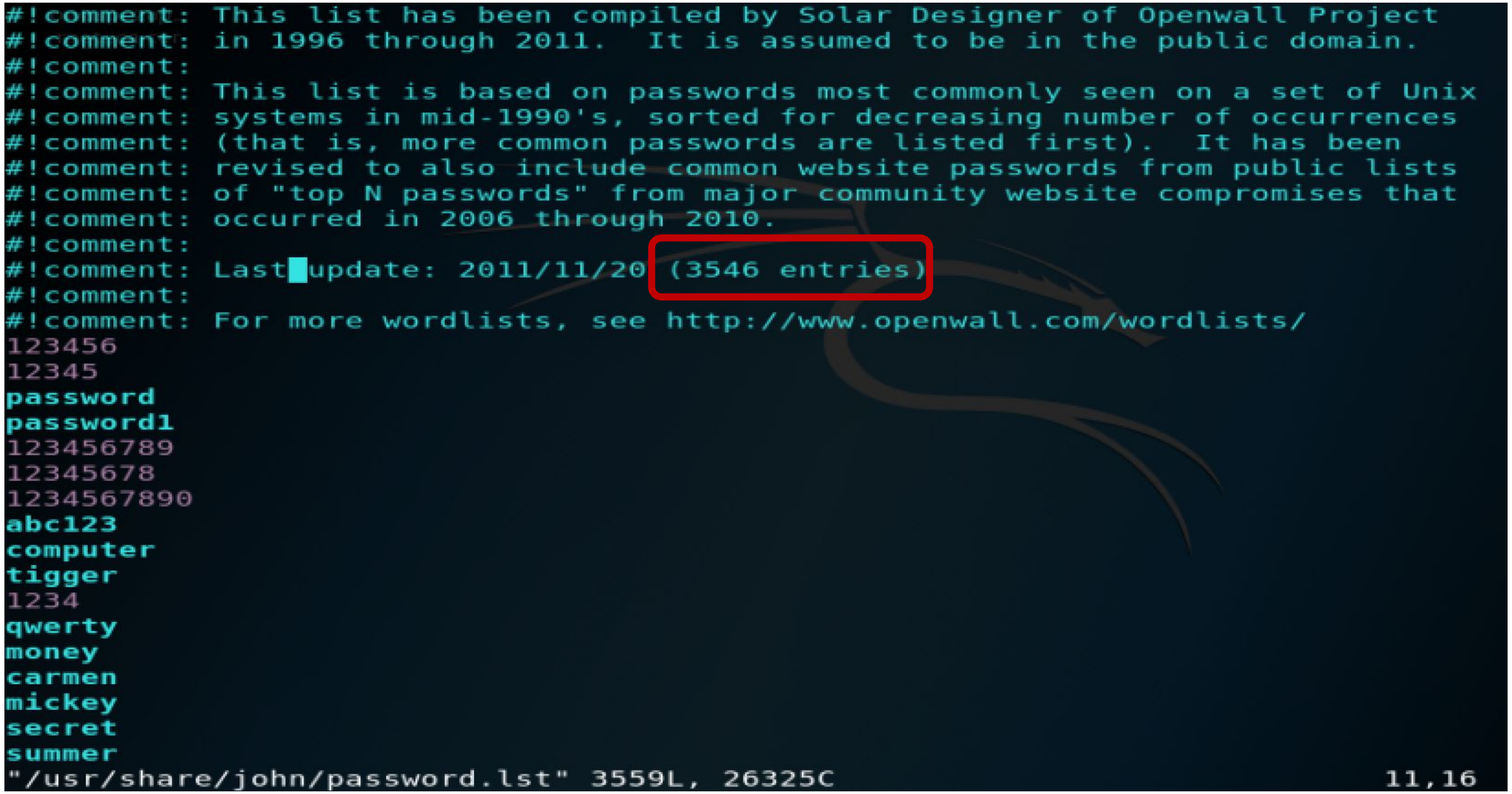
- **Single-crack** mode (`--single`):
  - Uses the **login names**, **"GECOS"/"Full Name"** fields, and users' **home directory** names as candidate passwords
  - Also applies a large set of ***mangling rules***: used to modify/mangle **a possible password** and produce **multiple candidate passwords**
  - For JtR's **rules**: see https://www.openwall.com/john/doc/RULES.shtml
  - Is faster than wordlist mode

Cracking Modes of John the Ripper

- **Wordlist** mode (`--wordlist`):
  - Uses a ***wordlist***:
    a text file containing one word per line
  - The default but limited wordlist: `password.lst`
  - Should be no duplicate lines: **no** sorting done!
  - **The order** matters: most likely candidate first
  - Can use "***word mangling rules***"

John the Ripper: Default Wordlist

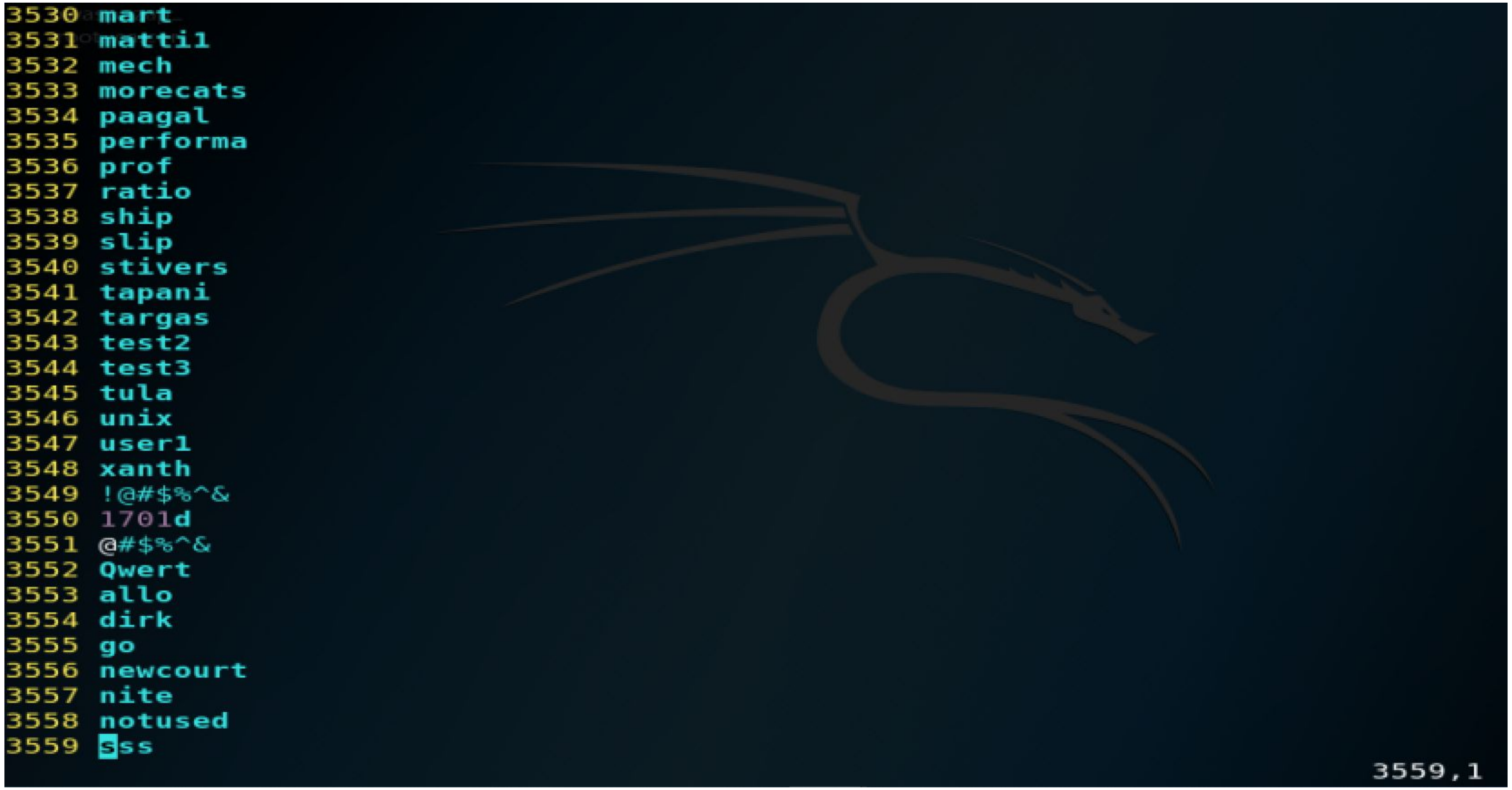
File `/usr/share/john/password.lst` (*beginning*)



```
#!comment: This list has been compiled by Solar Designer of Openwall Project
#!comment: in 1996 through 2011.  It is assumed to be in the public domain.
#!comment:
#!comment: This list is based on passwords most commonly seen on a set of Unix
#!comment: systems in mid-1990's, sorted for decreasing number of occurrences
#!comment: (that is, more common passwords are listed first).  It has been
#!comment: revised to also include common website passwords from public lists
#!comment: of "top N passwords" from major community website compromises that
#!comment: occurred in 2006 through 2010.
#!comment:
#!comment: Last update: 2011/11/20 (3546 entries)
#!comment:
#!comment: For more wordlists, see http://www.openwall.com/wordlists/
123456
12345
password
password1
123456789
12345678
1234567890
abc123
computer
tigger
1234
qwerty
money
carmen
mickey
secret
summer
"/usr/share/john/password.lst" 3559L, 26325C                            11,16
```

John the Ripper: Default Wordlist

File `/usr/share/john/password.lst` (*ending*)



```
3530 mart
3531 mattil
3532 mech
3533 morecats
3534 paagal
3535 performa
3536 prof
3537 ratio
3538 ship
3539 slip
3540 stivers
3541 tapani
3542 targas
3543 test2
3544 test3
3545 tula
3546 unix
3547 user1
3548 xanth
3549 !@#$%^&
3550 1701d
3551 @#$%^&
3552 Qwert
3553 allo
3554 dirk
3555 go
3556 newcourt
3557 nite
3558 notused
3559 sss
                                                              3559,1
```

Cracking Modes of John the Ripper

- **Incremental** mode (`--incremental`):
  - The most powerful cracking mode
  - Tries all possible character combinations
  - However, it is assumed that the cracking will never terminate
- Additional usage examples: https://www.openwall.com/john/doc/EXAMPLES.shtml
- Reference: https://www.openwall.com/john/doc

John the Ripper: Sample Log #1

```
0:00:00:00 Starting a new session
0:00:00:00 Loaded a total of 1 password hash
0:00:00:00 Cost 1 (iteration count) is 5000 for all loaded hashes
0:00:00:00 - UTF-8 input encoding enabled
0:00:00:00 - Passwords will be stored UTF-8 encoded in .pot file
0:00:00:00 - Rules/masks using ISO-8859-1
0:00:00:00 - Hash type: sha512crypt, crypt(3) $6$ (lengths up to 79)
0:00:00:00 - Algorithm: SHA512 128/128 AVX 2x
0:00:00:00 - Candidate passwords will be buffered and tried in chunks of 64
0:00:00:00 - Configured to use otherwise idle processor cycles only
0:00:00:00 Proceeding with "single crack" mode
0:00:00:00 - 1081 preprocessed word mangling rules
0:00:00:00 - Allocated 1 buffer of 8 candidate passwords
0:00:00:00 - Rule #1: ':' accepted as ''
0:00:00:00 - Rule #2: '-s x**' rejected
0:00:00:00 - Rule #3: '-c (?a c Q' accepted as '(?acQ’
…
…
0:00:00:00 - Rule #15: '-c )?a r l' accepted as ')?arl'
0:00:00:00 - Rule #16: '-: <* !?A l p' accepted as '<*!?Alp'
0:00:00:00 + Cracked root
0:00:00:00 Session completed
```

John the Ripper: Sample Log #2

```
0:00:00:00 Starting a new session
0:00:00:00 Loaded a total of 1 password hash
…
…
0:00:00:00 Proceeding with "single crack" mode
0:00:00:00 - 1081 preprocessed word mangling rules
0:00:00:00 - Allocated 1 buffer of 8 candidate passwords
0:00:00:00 - Rule #1: ':' accepted as ''
0:00:00:00 - Rule #2: '-s x**' rejected
0:00:00:00 - Rule #3: '-c (?a c Q' accepted as '(?acQ'
...
...
0:00:00:57 - Oldest still in use is now rule #1079
0:00:00:57 - Rule #1081: 'l Az"1900" <+' accepted as 'lAz"1900"<+'
0:00:00:57 - Oldest still in use is now rule #1080
0:00:00:57 - Processing the remaining buffered candidate passwords, if any
0:00:00:57 Proceeding with wordlist mode
0:00:00:57 - Rules: Wordlist
0:00:00:57 - Wordlist file: /usr/share/john/password.lst
0:00:00:57 - 57 preprocessed word mangling rules
0:00:00:57 - Rule #1: ':' accepted as ''
0:00:00:57 + Cracked user1
0:00:00:57 Session completed
```

John the Ripper: Sample Log #3

```
0:00:00:00 Starting a new session
0:00:00:00 Loaded a total of 1 password hash
0:00:00:00 Cost 1 (iteration count) is 5000 for all loaded hashes
0:00:00:00 - UTF-8 input encoding enabled
0:00:00:00 - Passwords will be stored UTF-8 encoded in .pot file
0:00:00:00 - Hash type: sha512crypt, crypt(3) $6$ (lengths up to 79)
0:00:00:00 - Algorithm: SHA512 128/128 AVX 2x
0:00:00:00 - Candidate passwords will be buffered and tried in chunks of 64
0:00:00:00 - Configured to use otherwise idle processor cycles only
0:00:00:00 Proceeding with wordlist mode
0:00:00:00 - Wordlist file: rockyou.txt
0:00:00:00 - No word mangling rules
0:00:02:19 + Cracked user2
0:00:02:19 Session completed
```

Defenses against Password-Cracking Attacks

- Strong password policy
- User awareness
- Password filtering/metering software
- User authentication tools in addition to passwords
- Do ***your own*** password-cracking tests
- Protect your encrypted or hashed password files: including on your backup disks/tapes