Handling Passwords with Security and Reliability in Background Processes

Don Libes
libes@nist.gov

National Institute of Standards and Technology

Overview

- Tools
- Techniques for Dealing with Passwords in Background
- Other Fun Scripts for Dealing with Passwords

Tcl – Tool Control Language

- Tcl is an interpreted language framework

```
set name "Don"

if {$a == sin($x)/2} {
    puts "hello world"
}

set pets {dog cat canary}
```

- Tcl is unassuming
- Tcl is designed to be extended

Expect Commands Are Simple

- Start an Interactive Process

```
spawn ftp
```

- Send a Command

```
send "dir\r"
send "get $file\r"
```

- Wait for a Response

```
expect {
    "550*ftp> "   cd_failed
    "200*ftp> "   cd_succeeded
    timeout       timed_out
}
```

Automating /bin/passwd

```
# script to automate /bin/passwd
spawn passwd $user
expect "password:"
send "$password\r"
expect "password:"
send "$password\r"
expect eof

# but how do we get password in??
```

Using Passwords Later

```
# prompt user for password
set password [getpass]

sleep ...

spawn telnet
expect "login:"
send "$user\r"
expect "Password:"
send "$password\r"

# now finish task
```

Using Passwords in the Background

```
spawn telnet
interact ~~ return
if [fork] exit
disconnect
# everything hereafter is in background

expect ...
send ...
```

What If Passwords Change?

```
while 1 {
    send "$password\r"
    expect {
        "sorry" {
            find-human
            set passwd [getpass-from-human]
        }
        "$prompt"
    }
}
```

Communicating From The Background With Kibitz

- kibitz – allows two people to share an application

```
. . .
spawn kibitz –noproc $user
interact –u $proc –o ~ {
    return
}
```

Daemon

- gopher or mosaic – can’t interact with processes

passmass

- Problem: yppasswd only works on systems which cooperate
- Solution: passmass – set password on multiple hosts

```
passmass host1 host2 host3

passmass host1
  -user dlibes usenix.org \
  -user libes01 host.mit.edu \
  -telnet -program "set password" vmshost
```

cryptdir

- Problem: crypt only encrypts a single file at a time
- Solution: cryptdir – encrypts a whole directory at a time
  - won’t encrypt an already encrypted file
  - decrypts, too, of course

mkpasswd

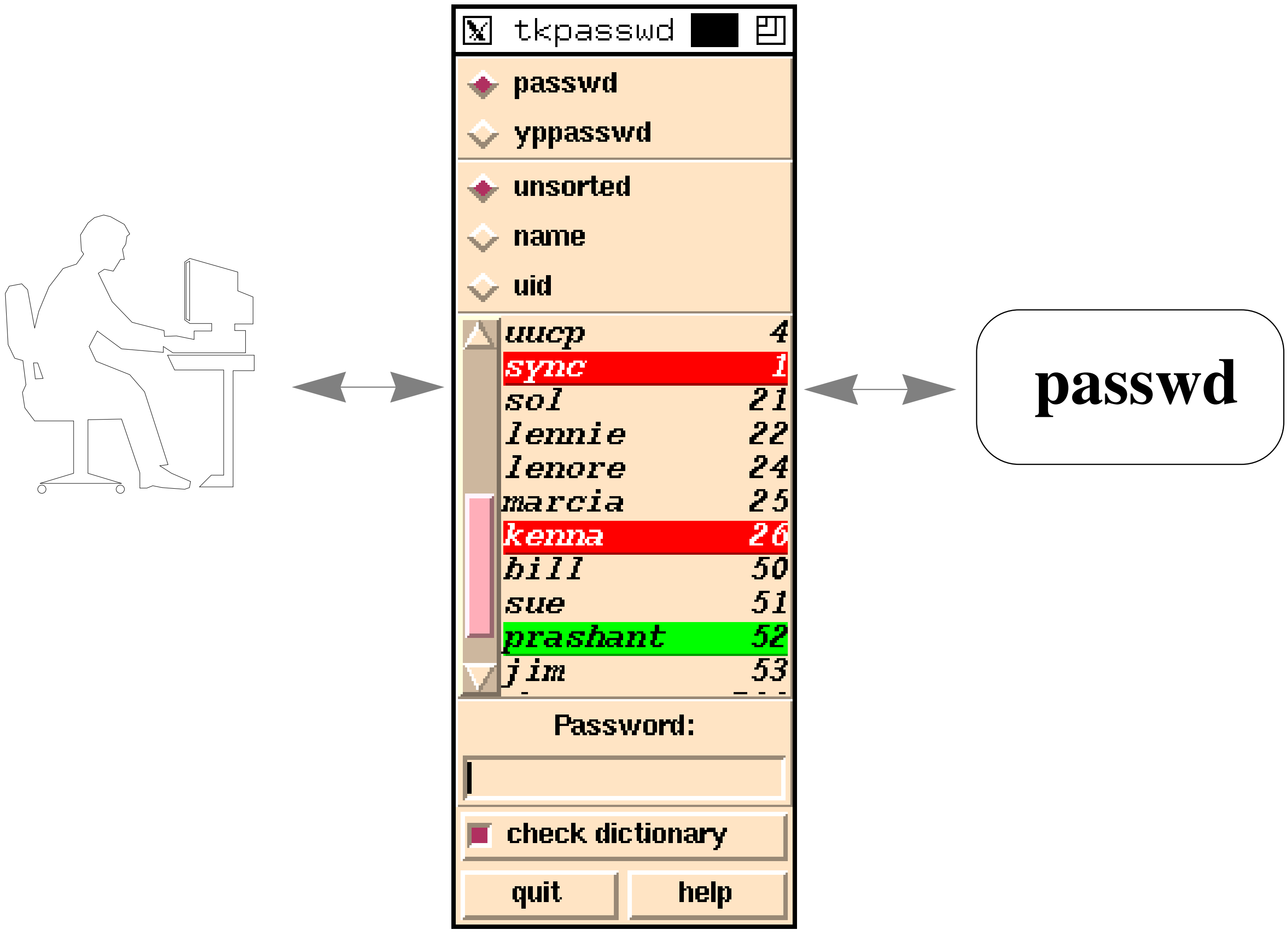
- Generates an unguessable password.

```
% mkpasswd
p1U6hpuBv
% mkpasswd -d 5
kM72Le499
```

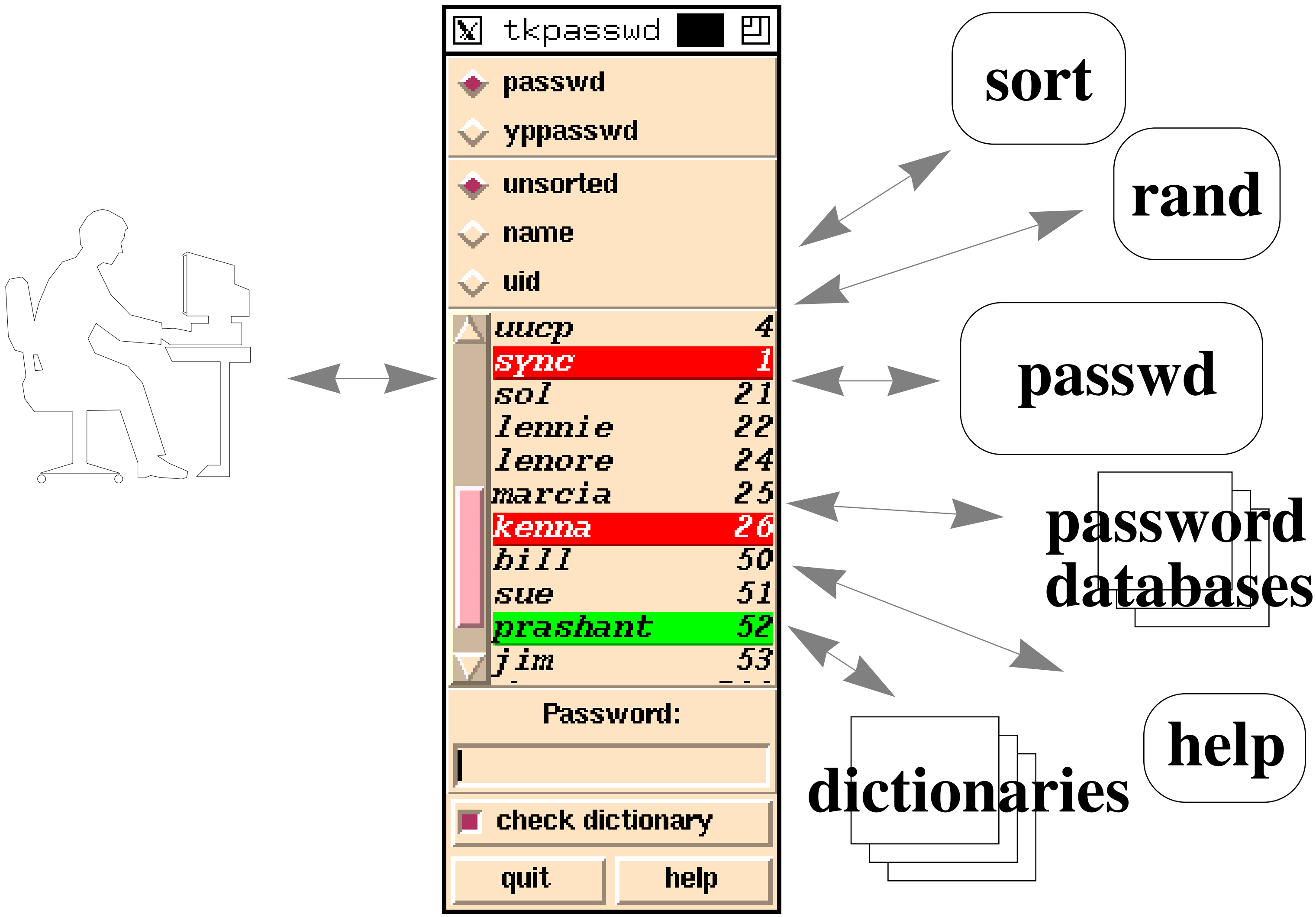
- Generates and sets password

```
% mkpasswd -d 5 don
h3U6g591X
```

tkpasswd


tkpasswd
passwd
yppasswd
unsorted
name
uid
uucp 4
sync 1
sol 21
lennie 22
lenore 24
marcia 25
kenna 26
bill 50
sue 51
prashant 52
jim 53
Password:
check dictionary
quit
help
passwd

tkpasswd


tkpasswd
passwd
yppasswd
unsorted
name
uid
uucp 4
sync 1
sol 21
lennie 22
lenore 24
marcia 25
kenna 26
bill 50
sue 51
prashant 52
jim 53
Password:
check dictionary
quit
help
sort
rand
passwd
password databases
dictionaries
help

Summary

- Many ways to automate background procs that require passwords
- Techniques are non-obvious ...
- ... but easy when you have the right tools

- Expect is in the public domain and is available from ftp.cme.nist.gov. Further information in the Proceedings.