Password Security: A Case History by Robert Morris and Ken Thompson (Bell Labs)

Presented by Matthew Little

What is the Presentation?

I. Brief Intro

II. Early Design Flaws

III. Attacks

IV. Solutions and Current State

V. Conclusion

The Beautiful Struggle

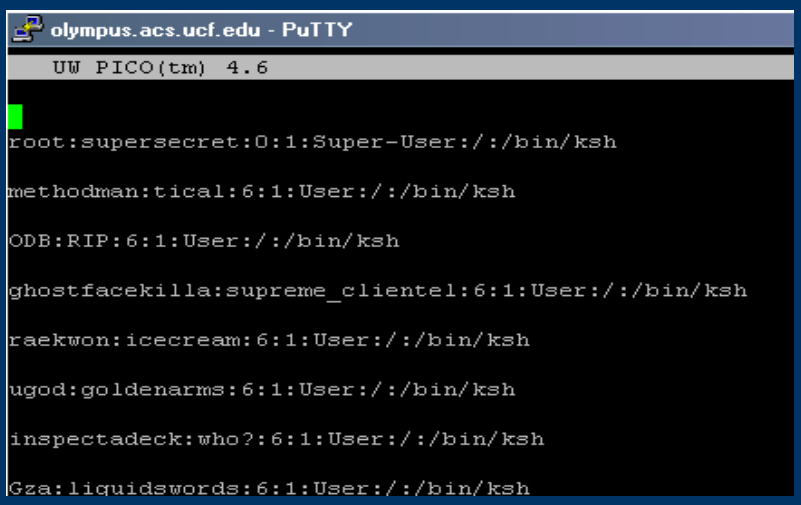
"This competition has been in the same vein as the competition of long standing between manufacturers of armor plate and those of armor piercing shells."

Origins

Early Attempts at password security had serious flaws. Old versions of Unix used a clear text password file.

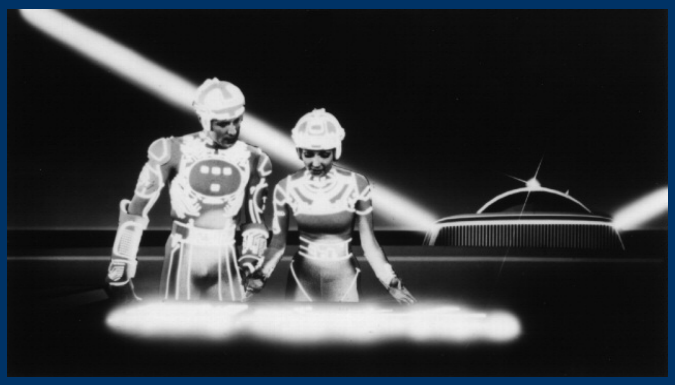
What if the Wu Tang Clan had an old Unix Server?

Their password file might look like this:



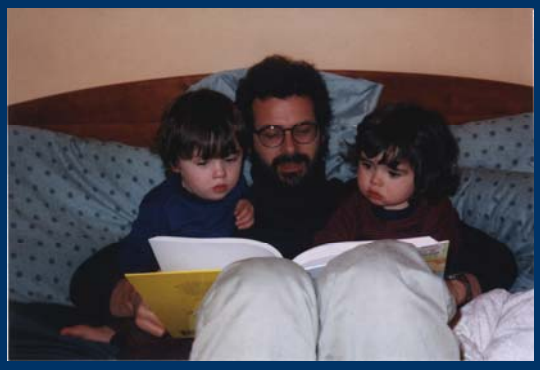
Cool Dude...so what's the problem?

At MIT, in the early 60's, a software bug caused ***the password file to be*** printed on every terminal when it was logged into.



Back that File Up?

In the old days they were backing up their files on magnetic tape, so anyone with physical access to the tape could read anything on it with no restrictions.



Spies have broken into a magnetic tape archive. They are reading and stealing passwords for malicious use.

We're talkin Remote Login?

Some programs used to log into remote servers, programs such as telenet, have a huge design flaw.

Clear Text

Worst Idea Ever!!!!!!!!!

Smells like Hacker Spirit

Because we are logging in 'remotely', we have to worry about 'Sniffers' (people monitoring the traffic on the network...looking for passwords sent in 'clear text'.)

Some Body Dial 911

Because we OBVIOUSLY need help

Encryption to the Rescue!

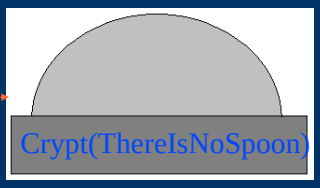
Even if they can see your password, it won't help.

“I don't even see the code...all I see is Blonde, Brunette, Redhead...”-Cypher

One way Hash to protect your...PASS

Password Enter

ThereIsNoSpoon

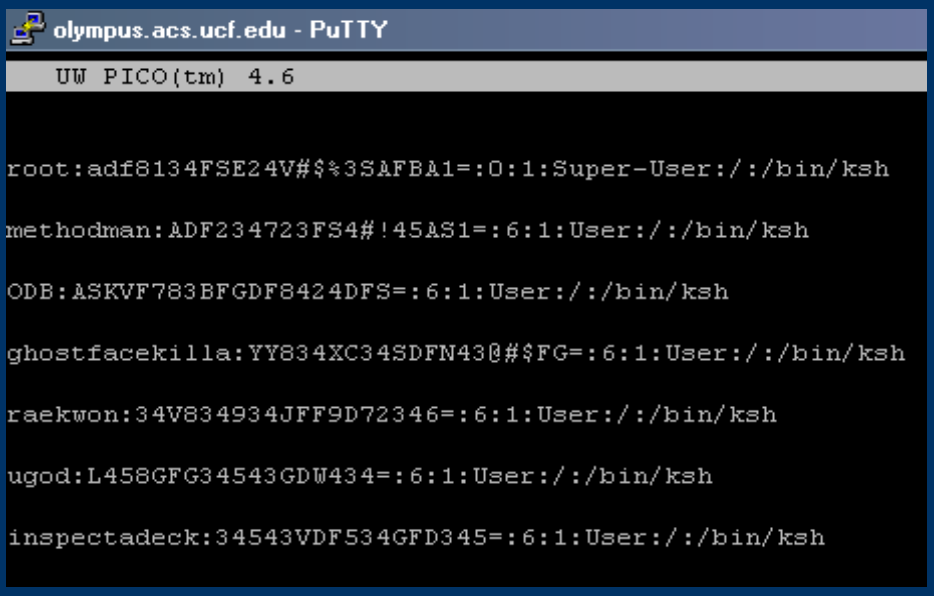


Cryptword Leave

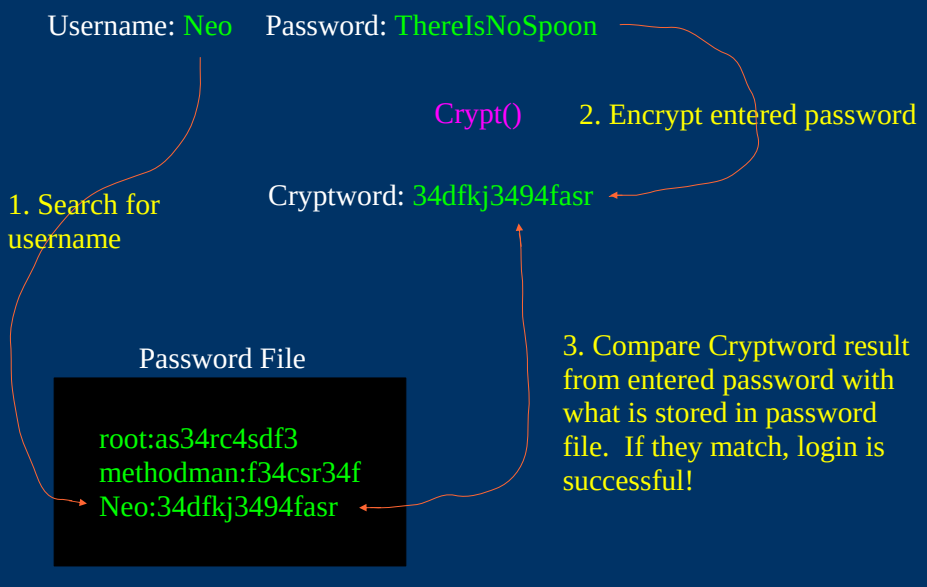
AD43V4EV43V

The UNIX crypt ( ) function takes the user's password as the encryption key and uses it to encrypt a 64-bit block of zeros. The resulting 64-bit block of text is then encrypted again with the user's password; the process is repeated a total of 25 times. The final 64 bits are unpacked into a string of 11 printable characters that are stored in the /etc/passwd file.

Wu Tang Clan ain't nothin to Hash wit?



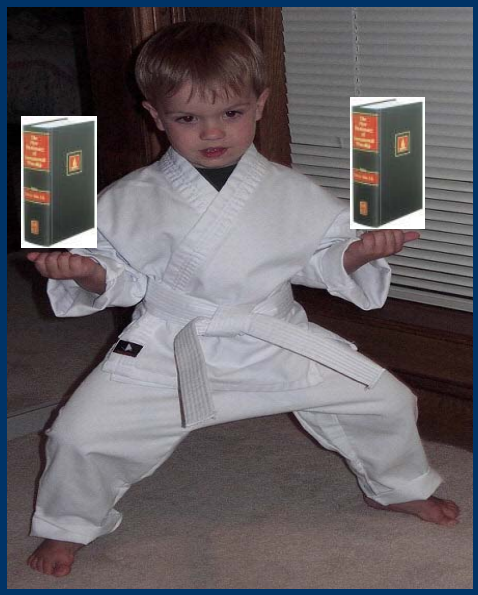
Who's on First!



“I'm Winston Wolf, I solve problems.”

With Encryption, we can allow access to the password file and send encrypted passwords over the network and not worry about a thang...right?

Kieeee Yaaaahhh...Dictionary Attack!



We are still vulnerable to”**Dictionary Attacks**”!

Sticks and Stones...

Hackers know the Encryption Algorithm. The Unix source code is readily available.

They use a “Dictionary”, a compiled catalog of likely Passwords.

Then they encrypt their dictionary of words using the same algorithm crypt().

Finally, with access to the password file, they search for encrypted matches...gotta match? gotta password!

“I know Kung Fu”-Neo
“...Show Me”-Morpheous

Dictionary

Cat
Dog
Hat
Boy

Crypt()

Dictionary w/ Cryptword

Cat 7z4s
Dog 7czq
Hat n81d
Boy 7bbe

Password File

root:n84z
michael:7bbe
yao:ab42
Ichiro:14cu
dmx:lup9

“So Michael's password must be Boy! Who would have guessed?”

Shake it like a Salt Shaker

The System can extend passwords by appending a 'Salt'. The 'Salt' is a 12-bit number between 0-4095. That means the same password can encrypt in 4096 different ways.

Username: Neo
Password w/o Salt: ThereIsNoSpoon
Cryptword: AD43V4EV43V

Pick a 'Salt' based on the time of day

Password w/ Salt: ThereIsNoSpoon0, ThereISNoSpoon1, ThereIsNoSpoon2, etc
Cryptword: BAI4BYZ4B4K, DJOE4N47XTYW, EXYE2845SS78G, etc

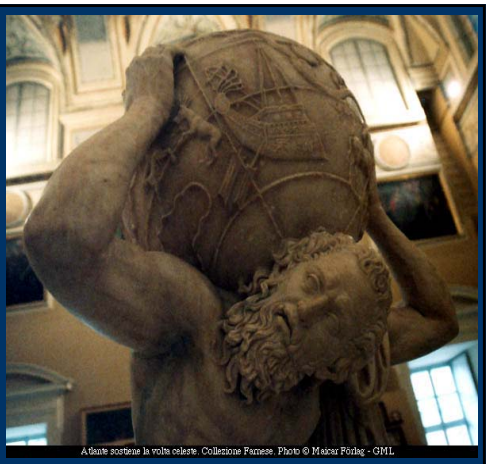
Player Haters

Cat	7z4s
Dog	7czq
Hat	n81d
Boy	7bbe

4 Entries

With 'Salt', Dictionary size must be increased by a factor of 4096

Cat0, Cat1, Cat2, ..., Cat4095
Dog0, Dog1, Dog2, ..., Dog4095
Hat0, Hat1, Hat2, ..., Hat4095
Boy0, Boy1, Boy2, ..., Boy4095



4 x 4096 = 16384 Entries

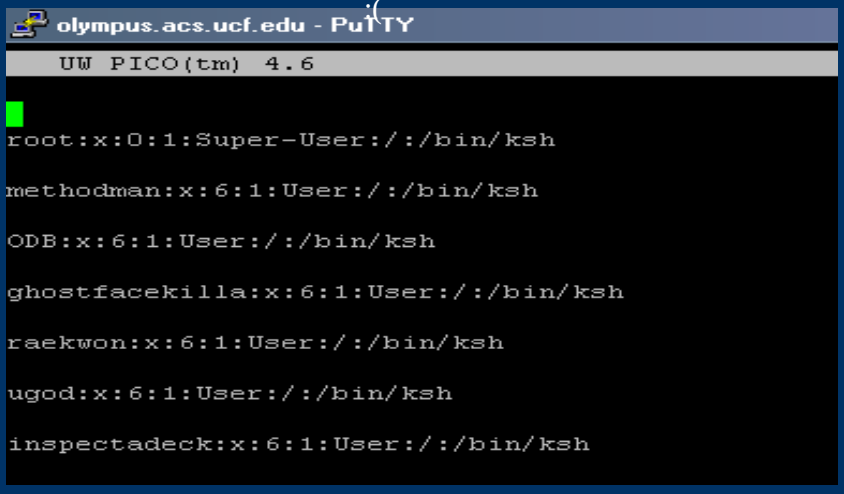
16K+ Entries for a Dictionary of just 4 likely Passwords!

You Can't Hack What You Can't See

Shadow Files have restricted permissions that prevent them from being read by intruders. The encrypted password is stored only in the shadow password file, /etc/shadow, and not in the /etc/passwd file. The passwd file is maintained as a world-readable file because it contains information that various programs use. The shadow file can only be read by root and it does not duplicate the information in the passwd file. It only contains passwords and the information needed to manage them.

Now Wu Tang Clan ain't nothin to hack wit!

The Password File has become totally useless to hackers :(



Instead of holding clear text passwords or Cryptwords, it now holds 'x'!

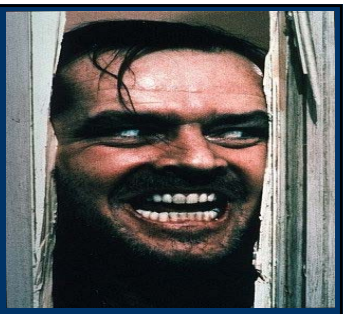
Hiding in the Shadows

olympus.acs.ucf.edu - PuTTY

```
  UW PICO(tm) 4.6

username: root
password: vdf4eadf434fa
salt: 2

username: methodman
password: 434rnc442345f
salt: 4894

username: ODB
password: 34cdfer84dfsf
salt: 873

username: raekwon
password: 34r8ck443fkdk
salt: 27
```

All the good stuff is locked up in the Shadow File...Restricted Access

Can I come in?



Username: Neo Password: ThereIsNoSpoon

1. Search for username

Saltword: ThereIsNoSpoon362

3. Encrypt Saltword Crypt()

Cryptword: 34dfkj3494fasr

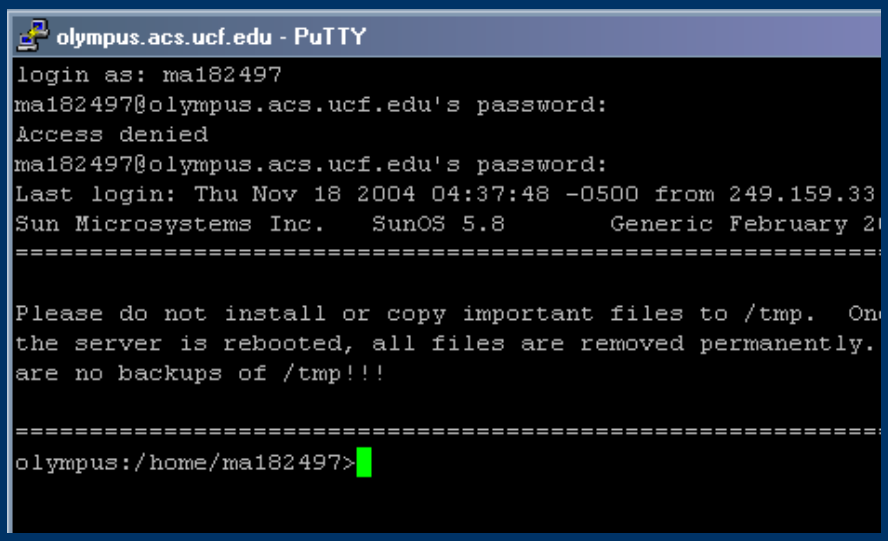
2. Append Salt

Shadow File

Root: as34rc4sdf3: 14
methodman: f34csr34f: 91
Neo: 34dfkj3494fasr: 362

4. Compare Cryptword result from entered password with what is stored in shadow file. If they match, login is successful!

They Spoof!...Made ya look!



What had happened was...

Username: Neo
Password: ThereIsNoSpoon

1

2

Access denied

3

Username:

4

A short shell script captured the user's login password. 1. It does this by spoofing the regular login screen. The user enters their login information at the prompt. 2. Their password is rejected and the script ends, 3. calling the login program, but not before emailing the login and password information to another account. 4. The user then retries their password and everything appears to work as normal.

Ctrl + Alt + Del...What is it good for?

Login:

Ctl + Alt + Del

Some Operating Systems can prevent this by requiring a special key combination to be entered before the login screen is presented, for example Ctl + Alt + Del. If the login prompt appears without having pressed the "secure attention key", you may be getting spoofed. Only the kernel should be able to detect when the "secure attention key" is pressed.

"It is Finished"

Early Design

Plain Text Password Files and Plain Text over the net

Attacks

Sniffing, Dictionary Attacks, and Spoofing

Solutions and Current State

Encryption, Salt, Shadowfile, Ctl + Alt + Del

References

Evolt.org Password encryption: rationale and Java example by James Shvarts
http://www.evolt.org/article/comment/18/60122/

TheFreeDictionarydotCom by Farlex, Login Spoofing
http://encyclopedia.thefreedictionary.com/Login%20spoofing

Practical UNIX and Internet Security Chap 8.6
http://www.unix.org.ua/orelly/networking/puis/ch08_06.htm

Bic, L., and Shaw, A., Operating System Principles pp. 425-426.

Morris, R., and Thompson, K., Password Security: A Case History, Communications of the ACM, 22(11), November 1979, pp. 594-597.

Questions?