Brains over Brawn: Intelligent Password Recovery

Sean Segreti
Security Consultant – KoreLogic
whoami

• Security consultant – penetration tester

• Fortune 500 and government clients

• Spoken at USENIX and BSides conferences

• Graduate of UMD & CMU in ECE
DEFCON - Crack Me If You Can
Password Cracking Contest

DARPA Cyber Fast Track: PathWell

KoreLogic Password Recovery Service
Why Talk about Passwords?
How we compromise systems

- 0-day, new unpublished exploits. < 2%
- Web Application flaws ~ 10%
- Malware/Phishing ~ 13%
- Guessing/cracking credentials ~ 75%
How Useful is Intelligent Cracking?

• We performed a pro bono project ~1 year ago for a law firm
  • Needed help cracking a password-protected PDF file

• The law firm used a commercial cracking tool for 81 days unsuccessfully

• We cracked it in 30 minutes on one machine

Effective guessing strategies are crucial
A brief history of password cracking

Cracking in 2017

What blue teams can do
• The first computer password – 1961
A Brief History

- The first computer password – 1961
- MIT’s Compatible Time-Sharing System (CTSS)
A Brief History

- The first computer password – 1961
- MIT’s Compatible Time-Sharing System (CTSS)
- First password breach – 1962
A Brief History

- The first computer password – 1961
- MIT’s Compatible Time-Sharing System (CTSS)
- First password breach – 1962
- Software bug in CTSS reveals password store

Alan Scherr
A Brief History

- The first computer password – 1961
  - MIT’s Compatible Time-Sharing System (CTSS)
- First password breach – 1962
  - Software bug in CTSS reveals password store
- First accidental password breach – 1966
A Brief History

- The first computer password – 1961
  - MIT’s Compatible Time-Sharing System (CTSS)
- First password breach – 1962
  - Software bug in CTSS reveals password store
- First accidental password breach – 1966
  - CTSS “quote of the day”
A Brief History

- The first computer password – 1961
  - MIT’s Compatible Time-Sharing System (CTSS)
- First password breach – 1962
  - Software bug in CTSS reveals password store
- First accidental password breach – 1966
  - CTSS “quote of the day”
- Password hashing introduced
**Hash**: A one-way function which is considered impossible to invert

<table>
<thead>
<tr>
<th>Input</th>
<th>Cryptographic Hash Function</th>
<th>Digest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fox</td>
<td></td>
<td>DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17</td>
</tr>
<tr>
<td>The red fox jumps over the blue dog</td>
<td>cryptographic hash function</td>
<td>0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC</td>
</tr>
<tr>
<td>The red fox jumps over the blue dog</td>
<td>cryptographic hash function</td>
<td>8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819</td>
</tr>
<tr>
<td>The red fox jumps over the blue dog</td>
<td>cryptographic hash function</td>
<td>FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45</td>
</tr>
<tr>
<td>The red fox jumps over the blue dog</td>
<td>cryptographic hash function</td>
<td>8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C</td>
</tr>
</tbody>
</table>
### Simple Hash Examples

<table>
<thead>
<tr>
<th>Hash Type</th>
<th>Hash Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD5SUM</td>
<td>ace: 360e2ece07507675dced80ba867d6dcd</td>
</tr>
<tr>
<td></td>
<td>acme: 53bce4f1dfa0fe8e7ca126f91b35d3a6</td>
</tr>
<tr>
<td></td>
<td>apple: 1f3870be274f6c493e31a0c6728957f</td>
</tr>
<tr>
<td></td>
<td>apple1: 26402cc420affb03828fc87f89996f0b</td>
</tr>
<tr>
<td></td>
<td>avalanche: 3188c31a53151637bcfe0b97ef70a397</td>
</tr>
<tr>
<td>SHA1SUM</td>
<td>ace: d86caede0264d429ed6b1d3fe83ec87a18eed990</td>
</tr>
<tr>
<td></td>
<td>acme: 293abb6b76d7791c0732cc517d38c4b5c734b87f</td>
</tr>
<tr>
<td></td>
<td>apple: d0be2dc421be4fcd0172e5afceea3970e2f3d940</td>
</tr>
<tr>
<td></td>
<td>apple1: 9663ea9a5e57758c0fb927047c5f68788ece4f49</td>
</tr>
<tr>
<td></td>
<td>avalanche: dd99c25c4f87c03bd08c6475bb933b3a8e370b00</td>
</tr>
</tbody>
</table>
HMAC-SHA256 (key = $salt)                OpenCart                              Android PIN
HMAC-SHA256 (key = $pass)                Redmine                               Juniper IVE                                     Keepass 1 (AES/Twofish) and Keepass 2 (AES)
HMAC-SHA1 (key = $salt)                  Mediawiki B type                      Juniper Netscreen/SSG (ScreenOS)                Blockchain, My Wallet
HMAC-SHA1 (key = $pass)                  Django (PBKDF2-SHA256)                Cisco-IOS                                       Bitcoin/Litecoin wallet.dat
HMAC-MD5 (key = $salt)                   Django (SHA-1)                        Cisco-ASA                                       1Password, cloudkeychain
sha512($salt.unicode($pass))            xt:Commerce                           AIX {ssha512}                                   Lastpass + Lastpass sniffed
sha512(unicode($pass).$salt)            osCommerce                            AIX {smd5}                                      PDF 1.7 Level 8 (Acrobat 10 - 11)
sha256($salt.unicode($pass))            Wordpress                             OSX v10.8, OSX v10.9, OSX v10.10                PDF 1.7 Level 3 (Acrobat 9)
sha256(unicode($pass).$salt)            Joomla > 2.5.18                       OSX v10.7                                       PDF 1.1 - 1.3 (Acrobat 2 - 4), collider #2
sha256($salt.$pass)                     Joomla < 2.5.18                       OSX v10.4, OSX v10.5, OSX v10.6                 PDF 1.1 - 1.3 (Acrobat 2 - 4), collider #1
sha256($pass.$salt)                     WBB3 (Woltlab Burning Board)          sha512crypt $6$, SHA512(Unix)                   PDF 1.1 - 1.3 (Acrobat 2 - 4), collider #1
sha1($salt.$pass.$salt)                 IPB (Invison Power Board)             sha256crypt $5$, SHA256(Unix)                   PDF 1.1 - 1.3 (Acrobat 2 - 4), collider #1
sha1(md5($pass))                        MyBB                                  bcrypt $2*$, Blowfish(Unix)                     MS Office 2013
sha1(sha1($pass))                       vBulletin > v3.8.5                    md5crypt $1$, MD5(Unix)                         MS Office 2010
sha1($salt.unicode($pass))              vBulletin < v3.8.5                     BSDiCrypt, Extended DES                         MS Office 2007
sha1(unicode($pass).$salt)              MyBB                                 descrypt, DES(Unix), Traditional DES           MS Office 2003
sha1($salt.$pass)                       Gmail < 2.5.18                         Gravitycrypt $1$, MD5(Unix)                    MS Office 2002
sha1($pass.$salt)                       Gmail > 2.5.18                         Gravitycrypt $1$, MD5(Unix)                    MS Office 2001
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 2000
sha1(md5($salt.$pass))                  Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1999
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1998
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1997
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1996
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1995
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1994
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1993
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1992
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1991
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1990
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1989
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1988
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1987
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1986
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1985
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1984
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1983
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1982
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1981
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1980
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1979
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1978
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1977
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1976
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1975
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1974
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1973
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1972
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1971
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1970
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1969
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1968
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1967
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1966
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1965
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1964
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1963
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1962
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1961
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1960
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1959
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1958
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1957
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1956
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1955
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1954
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1953
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1952
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1951
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1950
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1949
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1948
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1947
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1946
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1945
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1944
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1943
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1942
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1941
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1940
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1939
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1938
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1937
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1936
sha1($salt.unicode($pass))              Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1935
sha1($salt.$pass)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1934
sha1($pass.$salt)                       Wordpress                             Gravitycrypt $1$, MD5(Unix)                    MS Office 1933
Hashing assumptions
Hashing assumptions

• **Preimage resistance**
  • It should be hard to find a given hash’s preimage
Hashing assumptions

- **Preimage resistance**
  - It should be hard to find a given hash’s preimage

- **Second preimage resistance**
  - Given a preimage, it should be hard to find another with the same hash
- **Preimage resistance**
  - It should be hard to find a given hash’s preimage

- **Second preimage resistance**
  - Given a preimage, it should be hard to find another with the same hash

- **Collision resistant**
  - It should be hard to find any two preimages with the same hash
  - “Birthday attack”
A Brief History

- Password hashing introduced
  - UNIX Crypt(1) – 1974
  - Data Encryption Standard (DES) – 1979
Hashing Mishaps

- Early VAX systems running VMS – 1978
  - CRC32 based hashes relatively easy to find collisions for two different words
  - “penetration” and “prepituitary” both have the same hash “BF6A229E”

- UNIX DES – 1979
  - EFF shows that the 56-bit key can be brute-forced – 1999

- Microsoft LANMAN – Early 1990s
  - Split into two case-insensitive 7+ character DES hashes
Early mainstream cracking tools – 1980-2000

- Rainbow tables – 1980
- Computer Oracle & Password System (COPS) – 1989
- UNIX Crack – 1990
- Cracker Jack – 1993
Simple Dictionary Cracking
Simple Dictionary Cracking

hash: 297ee0aabc73ab6fc23bb819c8e42fac
Simple Dictionary Cracking

Dictionary

- ace
- abacus
- amen
- apple
- avert
- ...

hash: 297ee0aabc73ab6fc23bb819c8e42fac
Simple Dictionary Cracking

hash: 297ee0aabc73ab6fc23bb819c8e42fac

Dictionary

- ace
- abacus
- amen
- apple
- avert
- ...

ace
Simple Dictionary Cracking

Dictionary

- ace
- abacus
- amen
- apple
- avert
- ...

Hash Function

hash: 297ee0aabc73ab6fc23bb819c8e42fac
Simple Dictionary Cracking

Dictionary

- ace
- abacus
- amen
- apple
- avert
- ...

Hash Function

hash: 297ee0aabc73ab6fc23bb819c8e42fac

360e2ece07507675dc6d80ba867d6dcd
Simple Dictionary Cracking

Dictionary

ace
abacus
amen
apple
avert...

Hash Function

hash: 297ee0ab073ab6fc23bb819c8e42fac

Matches
297ee0ab073ab6fc23bb819c8e42fac?
Simple Dictionary Cracking

Dictionary

ace
abacus
amen
apple
avert
...

Hash Function

hash: 297ee0aabc73ab6fc23bb819c8e42fac

Matches
297ee0aabc73ab6fc23bb819c8e42fac?

Yes

360e2ece07507675dced80ba867d6dcd
Simple Dictionary Cracking

Dictionary

- ace
- abacus
- amen
- apple
- avert
- ...

Hash Function

Hash: 360e2ece07507675dced80ba867d6dc

Matches?

hash: 297ee0aabc73ab6fc23bb819c8e42fac

Cracked!
Simple Dictionary Cracking

Dictionary

ace  
abacus  
amen  
apple  
avert  
...

Hash Function

ace

360e2ece07507675dced80ba867d6dcd

Matches

hash: 297ee0aabc73ab6fc23bb819c8e42fac

Yes

Cracked!

No

297ee0aabc73ab6fc23bb819c8e42fac

?
Simple Dictionary Cracking

Dictionary

- ace
- abacus
- amen
- apple
- avert
- ...

Hash Function

Hash: 360e2ce07507675dced80ba867d6dcd

Matches 297ee0aabc73ab6fc23bb819c8e42fac?

- Yes
  - Cracked!
- No
  - Fetch a new guess and try again

Hash: 297ee0aabc73ab6fc23bb819c8e42fac
Simple Dictionary Cracking

Dictionary

- ace
- abacus
- amen
- apple
- avert...

Hash Function

Hash: $360e2ece07507675dced80ba867d6dcd$

Matches?

hash: $297ee0aabc73ab6fc23bb819c8e42fac$

Yes

Cracked!

No

Fetch a new guess and try again
Simple Dictionary Cracking

Dictionary

- ace
- abacus
- amen
- apple
- avert
- ...

Hash Function

Hash: 13f27b1072bbf7719d0d267b083ff91c

Matches: 297ee0aabc73ab6fc23bb819c8e42fac

Yes → Cracked!

No → Fetch a new guess and try again

hash: 297ee0aabc73ab6fc23bb819c8e42fac
Simple Dictionary Cracking

Dictionary

<table>
<thead>
<tr>
<th>ace</th>
<th>abacus</th>
<th>amen</th>
<th>apple</th>
<th>avert</th>
</tr>
</thead>
</table>

Hash Function

- Hash: 297ee0aabc73ab6fc23bb819c8e42fac

Matches?

- Yes
- No

Fetch a new guess and try again

Cracked!
Simple Dictionary Cracking

Hash Function

Dictionary

- ace
- abacus
- amen
- apple
- avert
- ...

hash: 297ee0aabc73ab6fc23bb819c8e42fac

Matches?

297ee0aabc73ab6fc23bb819c8e42fac

Yes

Cracked!

No

Fetch a new guess and try again
Simple Dictionary Cracking

Dictionary

ace
abacus
amen
apple
avert
...

Hash Function

765522d1be9232d1e2236c53d7780970

Matches 297ee0aabc73ab6fc23bb819c8e42fac ?

hash: 297ee0aabc73ab6fc23bb819c8e42fac

Cracked!

Fetch a new guess and try again

Yes

No

Hash Function

avert

ace
abacus
amen
apple
avert
...
Simple Dictionary Cracking

Dictionary

ace
abacus
amen
apple
avert
...

Hash Function

9e925e9341b490bfd3b4c4ca3b0c1ef2

Matches
hash: 297ee0aabc73ab6fc23bb819c8e42fac

Cracked!
Simple Dictionary Cracking

Dictionary

ace
abacus
amen
apple
avert
...

Hash Function

89fe7b5ca56cdee4e750f4eb3ab12fbb

Matches

297ee0aabc73ab6fc23bb819c8e42fac

hash: 297ee0aabc73ab6fc23bb819c8e42fac

Fetch a new guess and try again

Yes
Cracked!

No

297ee0aabc73ab6fc23bb819c8e42fac

Yes
Cracked!
Simple Dictionary Cracking

Dictionary

Hash Function

Matches

hash: 297ee0aabc73ab6fc23bb819c8e42fac

Fetch a new guess and try again

Cracked!
Simple Dictionary Cracking

Dictionary

ace
abacus
amen
apple
avert
...

Hash Function

3103a7d8f63cb184287c558ca51c6160

Matches

297ee0aabc73ab6fc23bb819c8e42fac

hash: 297ee0aabc73ab6fc23bb819c8e42fac

Cracked!
Simple Dictionary Cracking

Dictionary
- ace
- abacus
- amen
- apple
- avert
- ...

Hash Function
- hash: 297ee0aabc73ab6fc23bb819c8e42fac

Matches
- 297ee0aabc73ab6fc23bb819c8e42fac

Yes
- Cracked!

No
- Fetch a new guess and try again

- 297ee0aabc73ab6fc23bb819c8e42fac: maryland
Cracking Stronger Passwords
What if… “Maryland”?

“maryland” hash: 297ee0aabc73ab6fc23bb819c8e42fac
“Maryland” hash: 11ea71f253acf02a2d126a0901e765a7
What if… “Maryland”?

Dictionary

ace
abacus
amen
apple
avert
...

Hash Function

“maryland” hash: 297ee0aabc73ab6fc23bb819c8e42fac
“Maryland” hash: 11ea71f253acf02a2d126a0901e765a7

297ee0aabc73ab6fc23bb819c8e42fac

Matches

11ea71f253acf02a2d126a0901e765a7

No

Yes

Cracked!

Fetch a new guess and try again
What if… “Maryland”?

Dictionary

ace
abacus
amen
apple
avert
...

Hash Function

“maryland” hash: 297ee0aabc73ab6fc23bb819c8e42fac
“Maryland” hash: 11ea71f253acf02a2d126a0901e765a7

Fetch a new guess and try again

No

Matches 11ea71f253acf02a2d126a0901e765a7?

Yes

Cracked!

With this approach, if the password is not in the dictionary, it will not crack.
More Guesses: Mangling Rules

Dictionary

ace
abacus
amen
apple
avert
...

Rules

- Capitalize the first character
- Append a digit
- Append a 4-digit year
- Append birthdays
- Capitalize the first & Append a digit
...

Extended Dictionary

Ace
ace1
ace2
ace3
...
ace1970
ace1971
...
ace073191
ace080191
ace080291
...
Ace1
Ace2
Ace3
...

Agenda

- A brief history of password cracking
- Cracking in 2017
- What blue teams can do
Modern cracking tools

- Hashcat
- John the Ripper
- L0phtCrack
- Ophcrack
- RainbowCrack
- Cain & Abel
- aircrack-ng

- DaveGrohl
- THC Hydra
- Crowbar
- Brutus
- WFuzz
- Medusa
- And more...
Mangling Rule Possibilities (Simple)

l  convert to lowercase
u  convert to uppercase
c  capitalize
C  lowercase the first character, and uppercase the rest
t  toggle case of all characters in the word
TN toggle case of the character in position N
r  reverse: "Fred" -> "derF"
d  duplicate: "Fred" -> "FredFred"
f  reflect: "Fred" -> "FredderF"
{  rotate the word left: "jsmith" -> "smithj"
}  rotate the word right: "smithj" -> "jsmith"
$X  append character X to the word
^X  prefix the word with character X
Mangling Rule Possibilities (Complex)

S shift case: "Crack96" -> "cRACK(^"
V lowercase vowels, uppercase consonants: "Crack96" -> "CRaCK96"
R shift each character right, by keyboard: "Crack96" -> "Vtsvl07"
L shift each character left, by keyboard: "Crack96" -> "Xeaxj85"

sXY replace all characters X in the word with Y
s?CY replace all characters of class C in the word with Y
@X purge all characters X from the word
@?C purge all characters of class C from the word
!X reject the word if it contains character X
!?C reject the word if it contains a character in class C
/X reject the word unless it contains character X
/?C reject the word unless it contains a character in class C
=NX reject the word unless character in position N is equal to X
=N?C reject the word unless character in position N is in class C
(X reject the word unless its first character is X
(?C reject the word unless its first character is in class C
)X reject the word unless its last character is X
)?C reject the word unless its last character is in class C
%NX reject the word unless it contains at least N instances of X
%N?C reject the word unless it contains at least N characters of class C
Writing Custom Mangling Rules

- Organization’s password policy
- Help Desk reset patterns
- Rules to match previous cracks/patterns
- Cyclical passwords e.g. Spring2016
- Combining rules

```bash
# Capitalize 1st, append Number Number Special
caZ"[0-9][0-9][!$@%.^&()_+-={|\}\;':,/\<\>?`~*]"
```

Thousands of JtR rules like this are available at:

http://contest-2010.korelogic.com/rules.html
Custom Dictionaries

- Natural languages: English, German, French, Chinese, Russia, Italian, etc.
- Twitter/Wikipedia/Pastebin scrapes, Ngrams
- Phrases, quotes, lyrics, movie lines, speeches
- Technical papers, jargon/slang
- Human and pet names
- KRAD h4q0r, 31337
- Targeted dictionaries: internal corporate words, geopolitical regions, sports teams
- Keyboard patterns: qwertyui, asdfzxcv, dvorak
- Public password leaks (see below)
- Your previous password cracks

Public password leaks:
https://github.com/danielmiessler/SecLists/tree/master/Passwords

Automated web scraping tools:
https://bitbucket.org/mattinfosec/wordhound
Interactive Cracking

“Fire and forget” approaches are inefficient

Real-time tweaks are critical
CUSTOM rules

- Organization’s password policy
- Help Desk reset patterns
- Rules to match previous cracks/patterns
- Cyclical passwords e.g. Spring2016
- Combining rules

CUSTOM dictionaries

- Natural languages: English, German, French, Chinese, Russia, Italian, etc.
- Twitter/Wikipedia/Pastebin scrapes, Ngrams
- Phrases, quotes, lyrics, movie lines, speeches
- Technical papers, jargon/slang
- Human and pet names
- KRAD h4q0r, 31337
- Targeted dictionaries: internal corporate words, geopolitical regions, sports teams
- keyboard patterns: qwertyui, asdfzxcv, dvorak
- Public password leaks
- Your previous password cracks
Guessing and More Guessing

**CUSTOM rules**
- Organization’s password policy
- Help Desk reset patterns
- Rules to match previous cracks/patterns
- Cyclical passwords e.g. Spring2016
- Combining rules

These all increase the chances of recovering the password but they require enormous computational horsepower.

**CUSTOM dictionaries**
- Natural languages: English, German, French, Chinese, Russia, Italian, etc.
- Twitter/Wikipedia/Pastebin scrapes, Ngrams
- Phrases, quotes, lyrics, movie lines, speeches
- Technical papers, jargon/slang
- Human and pet names
- KRAD h4q0r, 31337
- Targeted dictionaries: internal corporate words, geopolitical regions, sports teams
- Keyboard patterns: qwertyui, asdfzxcv, dvorak
- Public password leaks
- Your previous password cracks
We need a bigger boat
CPUs and GPUs

CPU vs GPU Performance in GFLOPS (left) and Memory Bandwidth (right).
The Best GPU for Cracking

NVIDIA GTX 1080 Ti Founders Edition
Stacking GPUs
Dealing With Temperature

<table>
<thead>
<tr>
<th>Session.Name...</th>
<th>oclHashcat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status.........</td>
<td>Exhausted</td>
</tr>
<tr>
<td>Hash.Target....</td>
<td>File (hashes/some_client.hash)</td>
</tr>
<tr>
<td>Hash.Type......</td>
<td>NTLM</td>
</tr>
<tr>
<td>Time.Started...</td>
<td>Fri Jul 18 05:09:51 2014 (3 hours, 46 mins)</td>
</tr>
<tr>
<td>Time.Estimated.</td>
<td>0 secs</td>
</tr>
<tr>
<td>Speed.GPU.#1...</td>
<td>4819.3 MH/s</td>
</tr>
<tr>
<td>Speed.GPU.#2...</td>
<td>4823.0 MH/s</td>
</tr>
<tr>
<td>Speed.GPU.#3...</td>
<td>4830.2 MH/s</td>
</tr>
<tr>
<td>Speed.GPU.#4...</td>
<td>4828.2 MH/s</td>
</tr>
<tr>
<td>Speed.GPU.#5...</td>
<td>4817.5 MH/s</td>
</tr>
<tr>
<td>Speed.GPU.#6...</td>
<td>4825.0 MH/s</td>
</tr>
<tr>
<td>Speed.GPU.#*...</td>
<td>28943.2 MH/s</td>
</tr>
<tr>
<td>Recovered......</td>
<td>301/494 (60.09%) Digests, 0/1 (0.00%) Salts</td>
</tr>
<tr>
<td>Progress.......</td>
<td>392085408000000/392085408000000 (100.00%)</td>
</tr>
<tr>
<td>Skipped.........</td>
<td>0/392085408000000 (0.00%)</td>
</tr>
<tr>
<td>Rejected........</td>
<td>0/392085408000000 (0.00%)</td>
</tr>
<tr>
<td>HWMon.GPU.#1...</td>
<td>82% Util, 78c Temp, 100% Fan</td>
</tr>
<tr>
<td>HWMon.GPU.#2...</td>
<td>81% Util, 81c Temp, 41% Fan</td>
</tr>
<tr>
<td>HWMon.GPU.#3...</td>
<td>82% Util, 84c Temp, 42% Fan</td>
</tr>
<tr>
<td>HWMon.GPU.#4...</td>
<td>81% Util, 84c Temp, 41% Fan</td>
</tr>
<tr>
<td>HWMon.GPU.#5...</td>
<td>82% Util, 83c Temp, 42% Fan</td>
</tr>
<tr>
<td>HWMon.GPU.#6...</td>
<td>81% Util, 80c Temp, 40% Fan</td>
</tr>
</tbody>
</table>
“Founder’s Edition”

Impeller pushes hot exhaust out of chassis
Robust cooling for 100% load

Blows hot exhaust around chassis
Quiet for gaming
Pre-Built Cracking Platforms

Sagitta HPC “Brutalis” and “Invictus”

https://sagitta.pw/hardware/
Embarrassingly Parallel
Advanced Guessing Techniques

- Mask attacks
  - Brute-forcing character class sequences
- Combinator attacks
  - Combining two dictionaries
- Random mangling rule generation
  - Data-driven rule generation
- Probabilistic approaches
  - Markov models, neural networks
Before DEFCON Crack Me If You Can:

- Investigator at XYZ agency had an encrypted file involved in child exploitation
  - OSX DMG format
  - Needed to crack it, but there were NO tools to do so.

2010:

- Rule generation "barely" existed
- Public datasets were smaller
- Little sharing of techniques/strategies
- The community was scattered
- Little GPU use

2017:

- Cracking teams are well known
- More cracking tools (many are now open source)
- Dictionaries are widely available
  - RockYou, LinkedIn, Yahoo, Ashley Madison
- 3-4 times as many hash types are supported
- We have trouble finding NEW challenges for password cracking contests now.
A brief history of password cracking

Cracking in 2017

What blue teams can do
Blue Team Strategies

Minimal effort:

- Stringent password policy
  - Minimum length 12 characters, multiple character classes
- Mandatory password expiry (90 days recommended)
  - Too short → cyclical patterns
  - Too long → attackers have more time to crack
- Disable LANMAN
- Disable local administrator password in GPO
Blue Team Strategies

Moderate Effort:

- Focus on improving password security awareness for **administrators first**
  - Place a password policy on admins that is more stringent than what is imposed on regular users
- Eliminate accounts whose passwords never expire
  - Only mission-critical accounts should be granted a waiver
  - Do not allow admins to dodge password expiration
- Review corporate policy no less than annually
Significant effort:

- Employ 2-factor authentication where possible
  - **Expensive but effective**
- Use well-researched password strength meters
  - [https://github.com/cupslab/password_meter](https://github.com/cupslab/password_meter)
- Enforce password diversity (PathWell)
  - [https://github.com/KoreLogicSecurity/libpathwell](https://github.com/KoreLogicSecurity/libpathwell)
- Perform periodic password audits
  - Eliminate bad passwords / reward good passwords
  - Track improvement
Improvement Over Time #2

The bar chart shows the percentage improvement over time for 'All Accounts' and 'Admin Accounts'. The bars are color-coded: blue for 'All Accounts' and orange for 'Admin Accounts'. The x-axis represents time periods 1 to 9, while the y-axis represents the percentage improvement ranging from 0.00% to 80.00%.
Improvement Over Time #2

Average Password Length

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
Brains over Brawn: Intelligent Password Recovery

Sean Segreti
Security Consultant – KoreLogic

@CrackMelfYouCan

Contact: ss@korelogic.com