IDENTIFYING PASSWORDS ON DISK

Shiva Houshmand
Sudhir Aggarwal
Umit Karabiyik

Florida State University
Department of Computer Science
E-Crime Investigative Technologies Laboratory

Introduction

- **Passwords continue to be the primary mean of authenticating**
- **Sites have developed policies that require more complex passwords**
- **Recommendations to create unique passwords for different accounts**
- **Users are increasingly turning to saving their passwords in some manner**

Surveys

Means for storing passwords
Which method do you use to store passwords?
I remember passwords 71 73 69 80 61 72
In a notepad/sketchbook 23 17 20 14 25 27
Write down passwords in a file / document on your computer 13 6 12 7 11 16
Passwords are stored in the browser 13 10 15 7 11 14
On a piece of paper / sticker next to the computer 12 8 18 7 9 13
Passwords are written down on a cell phone 9 4 3 7 8 13
Passwords are stored on the smartphone 9 4 6 6 6 12
Passwords are stored by specialized software 7 6 10 3 6 8
Global Total N=11583
Western Europe N=2121
North America N=1445
Latin America N=838
EEMEA N=3682
APAC N=3497

Our Aim

- **Investigators are interested to find possible stored passwords on the disk**

- **Given a disk :**

  Analyze the files and return a set of strings that are most probably passwords for the investigator

  - Examining the disk and retrieve tokens
  - Filtering techniques
  - Identifying passwords

Examining the Disk

- Where we look for files on disk:
  - Allocated space
  - Unallocated space
  - Hidden through the operating system

- Using different tools to retrieve files and tokens :
  - Tsk_recover
  - catdoc, docx2txt, xls2txt, unoconv and xls2txt, Unrtf, odt2txt, pdftotext

- Extract **whitespace** separated (space, tab, and newline) strings from each file and keep an associated text file with each token written on a single line.

Initial Filtering

- **Non-printing characters**: not valid ASCII characters for passwords
- **Length**: 6 < Password length < 21
- **Floating point**: xls files contain large number of floating point numbers [-+]? [0-9]* .? [0-9]+ ([eE][-+]?[0-9]+)?
- **Repeated tokens**: We keep one instance of each token in each file
- **Word punctuations**: Tokens that seem to be part of a sentence; any alpha string ending with ;:.,!?)} or starting with (or {.

Specialized Filtering

An extremely prevalent class of tokens is the set of alpha strings

- All-alphas
  - Based on password policies most passwords do not contain only alpha characters
- Sentences
  - Detect sentences using OpenNLP
- Capitalization
  - Filtering only all lowered-case alpha strings
- Dictionary words
  - Filtering those strings that appear in a dictionary
- Multiword
  - Filtering those strings that are not multiword (passphrases) (ex. iloveyou)

Identifying Passwords

How to distinguish passwords from other strings

Construct a *probabilistic context-free grammar** from training on a set of revealed passwords

- Parse every password into base structures and count their frequency.
- Base structures consist of L (alpha sequences), D (digits), S (symbols), M(capitalization)
- Base structure also includes length information

Password12%

$L_8(M_8)D_2S_1$

* M. Weir, S. Aggarwal, B. De Medeiros, B. Glodek, Password cracking using probabilistic context free grammars, IEEE Symposium on Security and Privacy (2009)

Probabilistic password attack

Training

Training Set

tiny99
1pass!
this2!
star99
.
.
.
tree99
burn1!
1star!
down11

Training

$S \rightarrow$	L_4D_2	0.5
$S \rightarrow$	$D_1L_4S_1$	0.25
$S \rightarrow$	$L_4D_1S_1$	0.25
$D_2 \rightarrow$	99	0.7
$D_2 \rightarrow$	11	0.3
$D_1 \rightarrow$	1	0.8
$D_1 \rightarrow$	2	0.2
$S_1 \rightarrow$	!	1.0
$L_4 \rightarrow$	alex	0.1
$S \rightarrow^*$ alex2! With probability $0.25 \times 0.1 \times 0.2 \times 1.0 = 0.005$		

Note: Alpha sequence probabilities come from dictionaries and are equal to $1/n_L$, where n_L is the number of words in the dictionary of length L.

Probabilistic password attack

Generating the guesses

S→	L_4D_2	0.5
S→	$D_1L_4S_1$	0.25
S→	$L_4D_1S_1$	0.25
D_2→	99	0.7
D_2→	11	0.3
D_1→	1	0.8
D_1→	2	0.2
S_1→	!	1.0
L_4→	alex	0.1

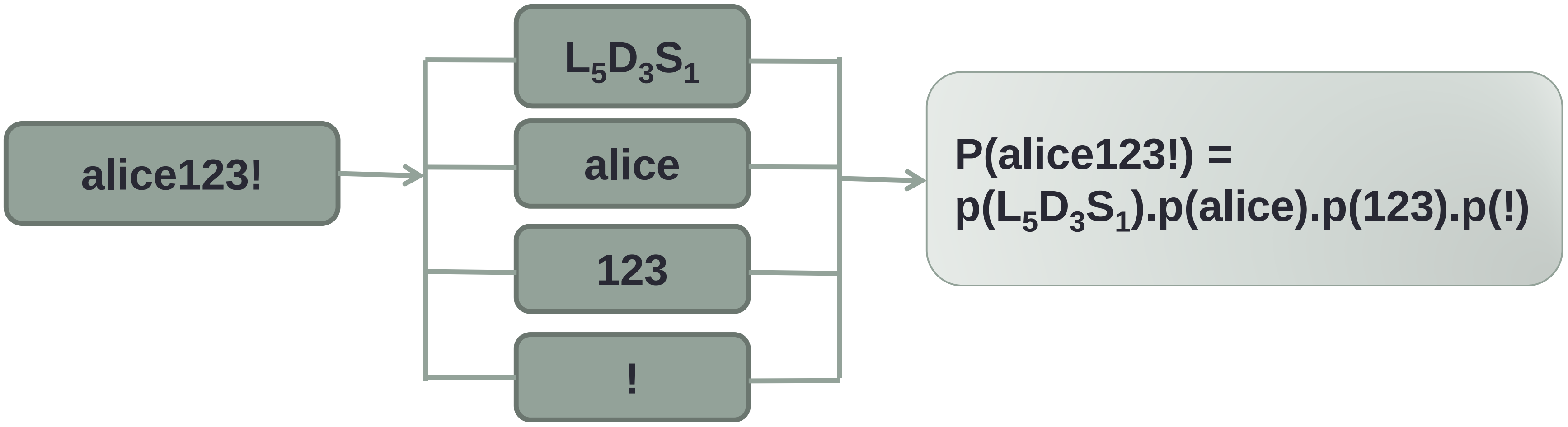
Guessing →

alex 99 andy 99 beta 99 …	0.035
1 alex ! 1 andy ! … alex 1 ! andy 1 ! …	0.02
alex 11 andy 11 …	0.015
2 alex ! 2 andy ! … alex 2 ! andy 2 ! ….	0.005

Identifying Passwords

How to distinguish passwords from other strings

- Using a probabilistic context-free grammar trained on a set of real user passwords, we can calculate the probability of any string.



Ranking algorithms

Outputting the top N tokens as the potential password set

- **Top Overall**:
  - The N highest probability tokens from the whole disk

- **Top percent**:
  - An equal percentage of the highest probability tokens of each file

- **Top 1-by-1**:

  Choose the highest probability token from each file and sort them

  Choose the second highest probability token from each file and sort

Experiment 1

Data Disk Image Size	#Files Analyzed	# Passwords Added
1 GB	1194	1000
500 MB	571	500
250 MB	426	250
100 MB	143	100
50 MB	108	50

- Reveled password sets to choose passwords from:
  Yahoo (300 thousand)

Initial Filtering Experiment

Percentage reduction of tokens due to each filter

Disk / Filter	50 MB	100 MB	250 MB	500 MB	1 GB
Non-printing	0	0	0	0.0015	0
Length	59.65	65.57	60.34	40.75	53.08
Floating point	1.05	0.45	20.71	46.87	28.21
Repeated token	85.04	82.79	73.78	75.63	70.10
Word punctuations	68.96	11.90	8.27	6.28	20.42
All-alphas	77.89	73.11	60.66	31.95	33.71

Initial Filtering Experiment

Reduction of tokens due to all filters

Disk	50 MB	100 MB	250 MB	500 MB	1 GB
# Before filtering (millions)	2.45	2.16	6.76	28.84	49.41
# After filtering (millions)	0.07	0.050	0.25	1.38	3.21
Total reduction (percent)	97.15	97.68	96.35	95.21	93.50

Experiment 2: Ranking Algorithms

- Stored 5 and 15 passwords in our disks

- Reveled password sets to choose passwords from:
  - CSDN (300 thousand)
  - Rockyou (1 million)

- Returned N potential passwords when N =1000, 2000, 4000, 8000, 16000

Ranking Algorithms

Storing 5 passwords from CSDN

N	Algorithm \ Disk	50 MB	100 MB	250 MB	500 MB	1 GB
N=1000	Top overall	1	2	0	0	2
	Top percent	2	3	1	1	2
	Top 1-by-1	**5**	**3**	**2**	**3**	**3**
N=2000	Top overall	1	2	0	0	2
	Top percent	5	3	1	1	2
	Top 1-by-1	**5**	**4**	**2**	**3**	**4**
N=4000	Top overall	5	2	0	0	2
	Top percent	5	3	2	1	2
	Top 1-by-1	**5**	**5**	**3**	**4**	**4**
N=8000	Top overall	5	3	0	0	2
	Top percent	5	3	2	1	3
	Top 1-by-1	**5**	**5**	**4**	**4**	**5**
N=16000	Top overall	5	4	0	0	2
	Top percent	5	4	2	3	3
	Top 1-by-1	**5**	**5**	**4**	**5**	**5**

Average Recall

40%
56%
92%

Ranking Algorithms

Storing 15 passwords from CSDN

N	Algorithm \ Disk	50 MB	100 MB	250 MB	500 MB	1 GB
N=1000	Top overall	1	7	0	2	2
	Top percent	4	10	2	3	3
	Top 1-by-1	**11**	**12**	**7**	**8**	**9**
N=2000	Top overall	1	9	0	2	2
	Top percent	9	10	2	4	5
	Top 1-by-1	**12**	**14**	**9**	**9**	**11**
N=4000	Top overall	11	10	0	2	2
	Top percent	10	11	3	5	6
	Top 1-by-1	**15**	**15**	**12**	**10**	**12**
N=8000	Top overall	13	11	0	2	2
	Top percent	11	11	8	5	8
	Top 1-by-1	**15**	**15**	**13**	**10**	**14**
N=16000	Top overall	15	14	0	2	2
	Top percent	12	14	9	8	8
	Top 1-by-1	**15**	**15**	**13**	**11**	**14**

Average Recall

37.3%

57.3%

89.3%

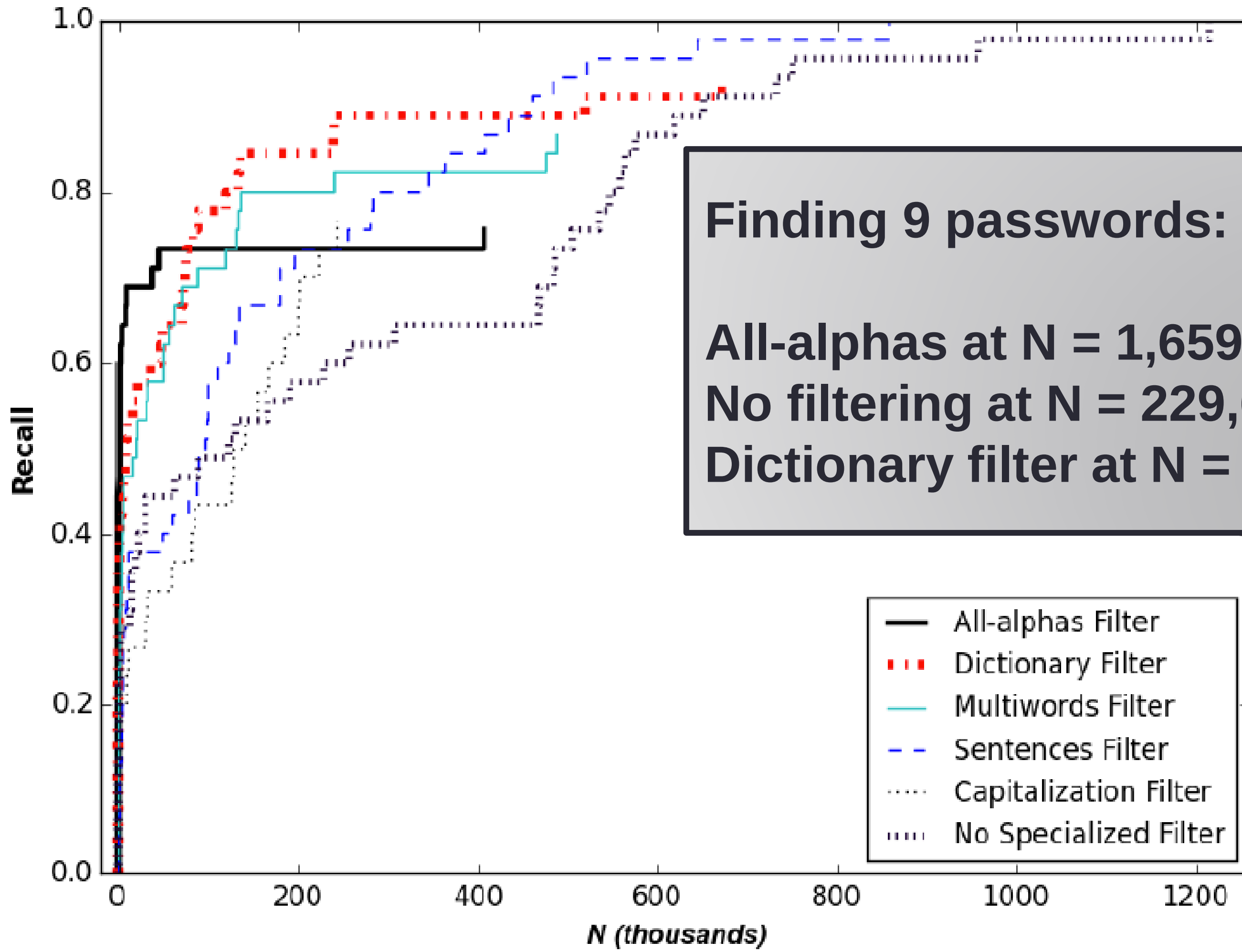
Experiment 3: Specialized Filtering

Storing 15 passwords from Rockyou

	Filter / Algorithm	No Filter (15)	Capitalization (11)	Multiwords (14)	Dictionary (14)	Sentences (15)	All-alphas (11)
N=1000	Top overall	0	2	0	0	0	5
	Top percent	1	1	3	3	2	1
	Top 1-by-1	**2**	**2**	**4**	**4**	**0**	**8**
N=2000	Top overall	0	2	0	0	0	5
	Top percent	1	2	3	3	2	2
	Top 1-by-1	**2**	**2**	**4**	**5**	**0**	**10**
N=4000	Top overall	0	2	0	0	0	5
	Top percent	2	3	3	3	3	4
	Top 1-by-1	**2**	**2**	**5**	**5**	**1**	**10**
N=8000	Top overall	0	2	0	0	0	5
	Top percent	4	4	5	5	3	7
	Top 1-by-1	**2**	**2**	**7**	**7**	**1**	**10**
N=16000	Top overall	0	2	0	0	0	5
	Top percent	4	4	5	5	3	7
	Top 1-by-1	**4**	**5**	**8**	**8**	**7`**	**10**

Specialized Filtering

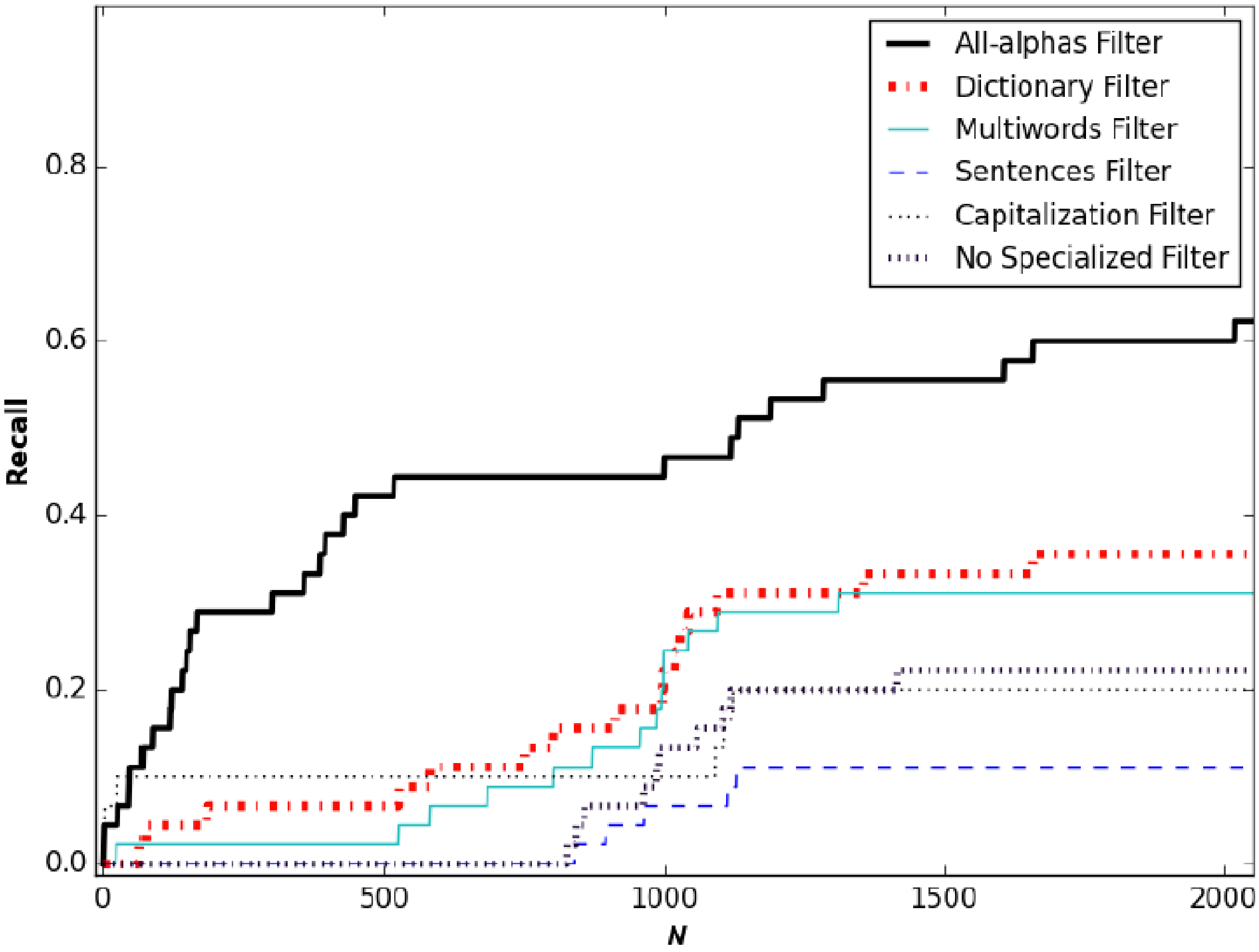
1-by-1 algorithm (several runs)



Finding 9 passwords:

All-alphas at N = 1,659
No filtering at N = 229,671
Dictionary filter at N = 36,240

Specialized Filtering



Example of top 20 potential passwords

Potential passwords	Probability
charles1	6.384 E-6
include3	1.687 E-6
program4	1.610 E-6
carolina23	6.272 E-7
light20	1.112 E-7
program97	7.757 E-8
lyndsay1	7.739 E-8
decagon1	7.739 E-8
dogbloo1	7.739 E-8
example1	7.739 E-8
pdprog1	5.370 E-8
report1	5.370 E-8
cielo123	5.080 E-8
soldiers1	4.044 E-8
bluberry1	4.044 E-8
listeria1	4.044 E-8
compendia1	3.110 E-8
framework1	3.110 E-8
alpha1s	2.972 E-8

Conclusion

- We can successfully identify most of the passwords on disks with large number of tokens.

- We return a relatively small set of potential passwords to be tried based on the investigator’s resources.

- The system can be adapted to work for cellphones and USB drives.