Have You Been Pwned?
the Aftermath of Data Leakage

Amrit Kumar & Cédric Lauradoux

Privatics team
INRIA Rhône-Alpes

27th February 2014

Recent Leakages

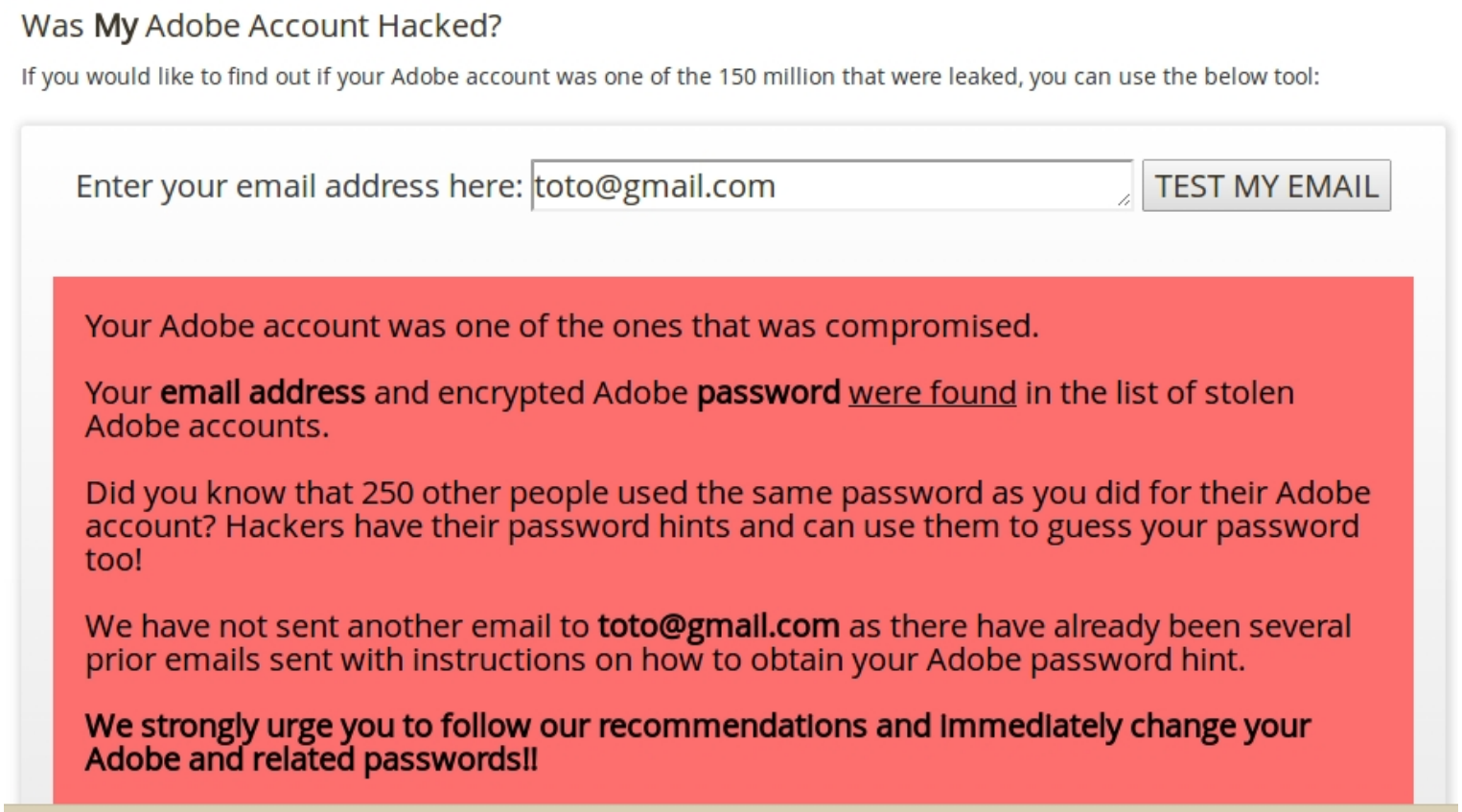
Victim	Data	Amount (million)
Adobe	username, email, encrypted pwd, pwd hint	150
LinkedIn	hashed password	6.5
Snapchat	user name, telephone number, city	4.6
Last.fm	hashed password	2.5
eHarmony	hashed password	1.5
Apple	UDID, device name, device type, etc.	1
coming soon	...	

How may a user know if his personal data has leaked ?

Alerting Websites : Good

If data leaked, user needn't download huge leaked database.

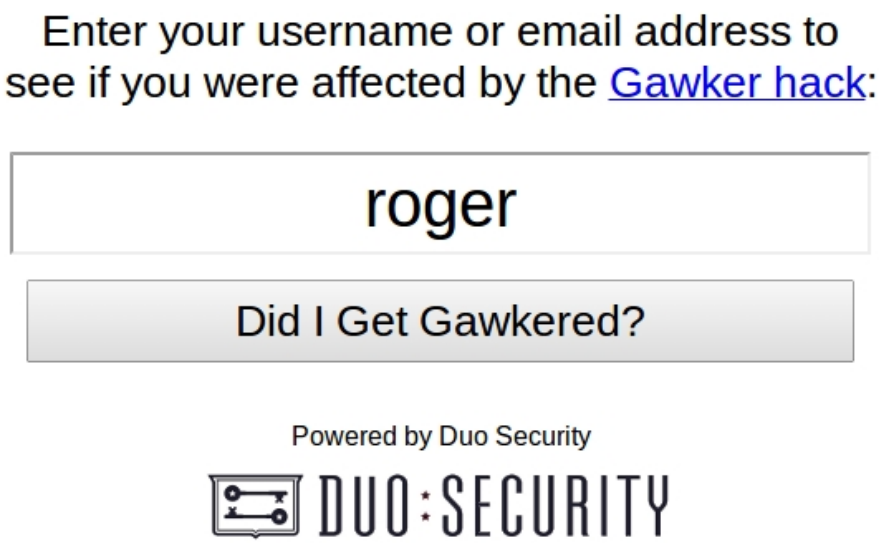
Example : `lastpass.com/adobe`

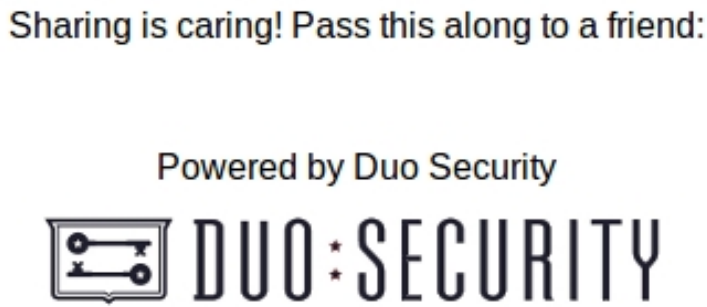


Alerting Websites : Phishing

If data isn't leaked, website learns a new data (possibly your password).

Example. didigetgawkered.com





Alerting Websites Cont.

Websites	Type	Database(s)	https	Statement	Answer	Descrip.
adobe.cynic.al	S	Adobe	✗	✗	✓	✓
bit.ly/1by3hd9	S		✓	✓	✓	✗
lucb1e.com/credgrep	S		✗	✗	✗	✗
adobe.breach.il.ly	S		✗	✗	✗	✗
snapcheck.org	S	SnapChat	✗	✗	✓	✗
findmysnap.com	S		✗	✗	✓	✗
lookup.gibsonsec.org	S		✗	✗	✓	✗
didigetgawkered.com	S	Gawker	✗	✗	✗	✗
lastpass.com	A	6	✓	✓	✓	✗
haveibeenpwned.com	A	9	✓	✓	✓	✓
bit.ly/1fj0SqV	A	12	✓	✗	✗	✗
dazzlepod.com/disclosure	A	28	✗	✓	✓	✗
bit.ly/1aJubEh	A	3346/194	✓	✗	✓	✗
hacknotifier.com	H	Unknown	✗	✓	✓	✗
pwnedlist.com/query	H	Unknown	✓	✗/✓	✓	✓
sicherheitstest.bsi.de	H	Botnets	✓	✓	✓	✗

Problem

Scenario

- Server hosts a public (leaked) database.
- Client wishes to know if his data belongs to the database.

Problem Statement

Can the client

- privately query a database,
- and remain more efficient than asking for the database itself ?

Private Information Retrieval

"Private Information Retrieval", by Chor et al, FOCS'95.

Scenario

- Server has a database $\mathcal{DB} = \{x_1, \ldots, x_n\} \in \{0,1\}^n$.
- Client has an index $1 \leq i \leq n$ and wishes to retrieve x_i.

Privacy

- Information-theoretic privacy : no information on i should be revealed. Chor et al. '98, Devet et al. '12.
- Computational privacy : i cannot be retrieved by a PPT server. Kushilevitz and Ostrovsky '97, Cachin et al. '99, Chang '04, Lipmaa '05, Gentry and Ramzan '05.

Trivial PIR : Send the database, communication complexity n.

PIR Cont.

PIR Variants

- Single server/multiple non-communicating servers.
- Private block retrieval (PBR) :
  $\mathcal{DB} = \{x_1, \ldots, x_n\} \in \{0,1\}^{\ell \cdot n}$, $1 \leq i \leq n$.
- PIR by Keywords :
  $\mathcal{DB} = \{x_1, \ldots, x_n\} \in \{0,1\}^{\ell \cdot n}$, $kw \in \{0,1\}^{\ell}$.
  "Private Information Retrieval by Keywords ", Chor et al. 1997
  - Store database in hash table.
  - Invoke PBR on digest.

PIR in Our Context ?

Is PIR applicable ?

- Original PIR :
  NO, user doesn't have physical address.
- PIR by keywords :
  NO, keywords statically defined $\Rightarrow$ prevents non-membership query.
- IDEA : Use Bloom filter to represent data
  Allows to apply original PIR + low memory footprint.

Bloom Filters

"Space/Time Trade-offs in Hash Coding with Allowable Errors", Burton Bloom, Communications of the ACM, 1970.

Tools

- S : a set of n elements.
- k hash functions with range $0, 1, \ldots, m-1$.
  hash functions $h_1, \ldots, h_k$ can be truncated SHA-1.
- b : array of bits of size m, initialized to 0.

Working Mechanism

- Insertion : find k hash-indices and set these bits to 1.
- Query : find k hash-indices and check if all these bits are set to 1.

False positive

Truncation $\Rightarrow$ Collisions $\Rightarrow$ False positive f.

PIR with Bloom Filter

- Server builds the Bloom filter b using k hash functions $\{h_1, h_2, \ldots, h_k\}$
- Client for a data y generates $\{h_1(y), h_2(y), \ldots, h_k(y)\}$.
- For each $1 \leq i \leq k$, Client invokes a single-server PIR and retrieves $b[h_i(y)]$.
- If $b[h_i(y)] = 0$ for any i, then y is not in the database.

Client			**Server**
y			b
		$\texttt{for } i \in \{1, \ldots, k\}$	
$q = h_i(y)$		$\xrightarrow{PIR(q)}$	$r = b_q$
NO	if $r = 0$	$\xleftarrow{r}$	
YES	else if $(i = k)$		

Possible Solutions

- k invocations of single server PIR to Bloom filter.
  $\Rightarrow$ computational privacy + communication complexity sublinear/square logarithmic.
- Send the complete filter.
  $\Rightarrow$ complete privacy + low communication complexity than trivial PIR.

Which is the most time efficient ?

Performance Evaluation

Building Bloom filter ($f = 2^{-128}$), SHA-1.

Database	Size	n	m (MB)	**Build time (mins)**	# CPU
SNAPCHAT	49 MB	4609621	102	6	1
LINKEDIN	259 MB	6458019	142	10	1
ADOBE	3.3 GB	153004872	412	198	1
ADOBE	3.3 GB	153004872	412	50	4

$f = 2^{-64}$

Database	m (MB)	**Build time (mins)**
SNAPCHAT	52	2
LINKEDIN	72	3
ADOBE	206	72

$f = 2^{-32}$

Database	m (MB)	**Build time (mins)**
SNAPCHAT	26	1
LINKEDIN	36	1.5
ADOBE	102	30

Conclusion

- Data leakages, the panorama.
- Privately alerting a user : PIR with Bloom filter vs. filter transmission.
- Compression techniques on Bloom filter.
- Experimental evaluation of techniques.

Thank you !