App Academy

Authentication // The Black Hat Angle

A Variety of Compromises

- Server Compromise
  - Server Logs
  - Corrupt your code
  - Can lead to a Database Compromise
- Session Compromise
  - CSRF, SQL Injections
- Database Compromise
  - Best Case: all user info, passwords, credit card #, SSN, etc.
  - Worst Case: all user info, password hashes

If your password’s aren’t hashed

id	username	password	createdAt	updatedAt
1	idbentley	uber733t	2020-09-04	2020-09-04
2	jortiz	a8skd8tew2	2020-09-04	2020-09-04
3	jmriley	arrakisbound	2020-09-04	2020-09-04

If your password’s are hashed

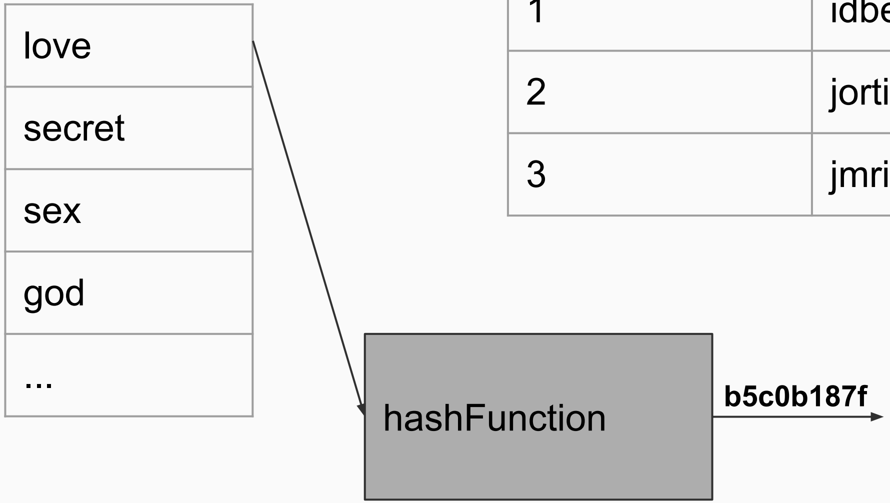
id	username	hashedPassword	createdAt	updatedAt
1	idbentley	**f8e731c33**	2020-09-04	2020-09-04
2	jortiz	**2e25c7168**	2020-09-04	2020-09-04
3	jmriley	**d44afde51**	2020-09-04	2020-09-04

The Four Most Common Passwords

Enter Rainbow Tables

id	username	hashedPassword
1	idbentley	**f8e731c33**
2	jortiz	**2e25c7168**
3	jmriley	**d44afde51**

b5c0b187f



love

secret

sex

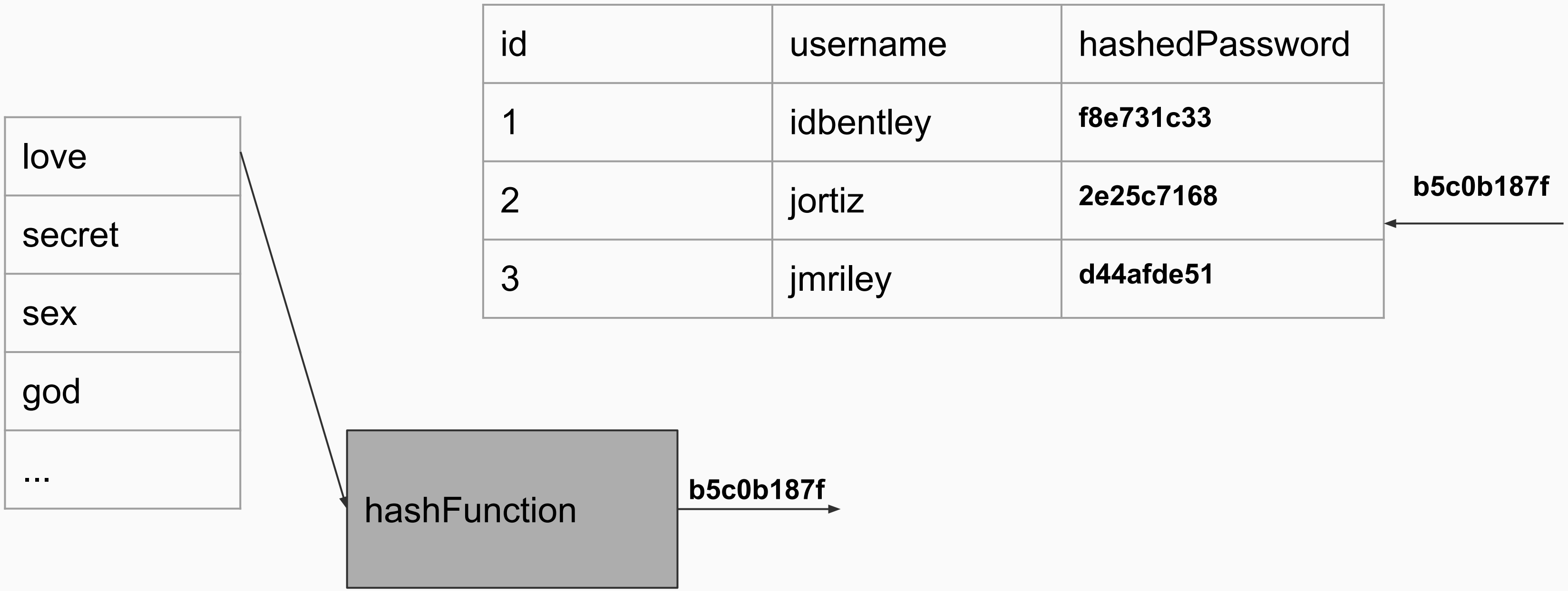
god

...

hashFunction

b5c0b187f

Enter Rainbow Tables



id	username	hashedPassword
1	idbentley	**f8e731c33**
2	jortiz	**2e25c7168**
3	jmriley	**d44afde51**

b5c0b187f

love
secret
sex
god
...

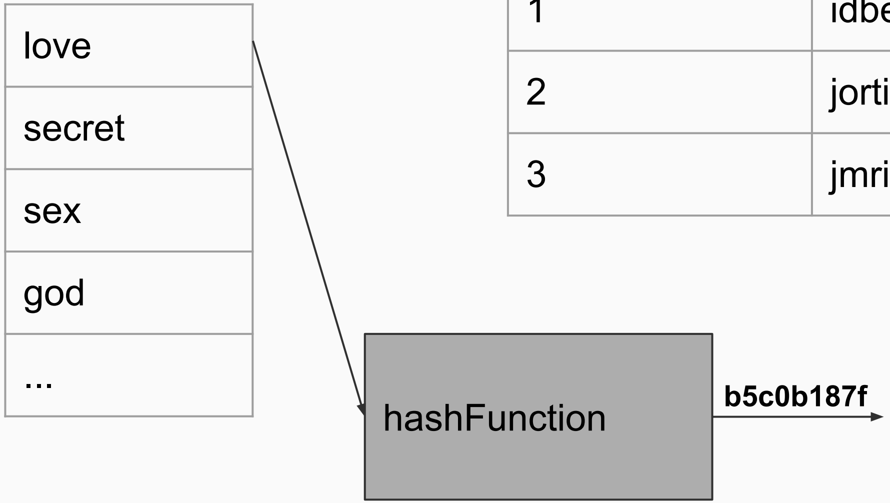
hashFunction

b5c0b187f

Enter Rainbow Tables

id	username	hashedPassword
1	idbentley	f8e731c33
2	jortiz	2e25c7168
3	jmriley	d44afde51

b5c0b187f



love

secret

sex

god

...

hashFunction

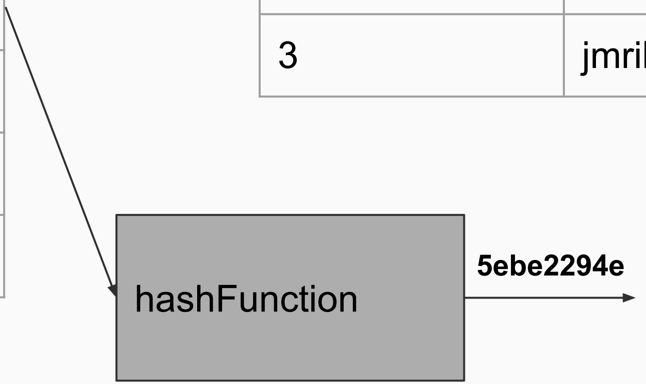
b5c0b187f

Enter Rainbow Tables

id	username	hashedPassword
1	idbentley	**f8e731c33**
2	jortiz	**2e25c7168**
3	jmriley	**d44afde51**

5ebe2294e

love
secret
sex
god
...



Rainbow Tables are Brute Force but Cached

- Hacker must try many combinations before finding a match
- Slower hashing functions, make brute forcing slower
- Hacker keeps track of all combinations

Cache results for efficient use and re-use

guess	hashFunction1	hashFunction2	hashFunction3
love	b5c0b187f	f1fdd3cbe6	4a502bc853
secret	5ebe2294e	419d952a	2f50c26e60d
sex	2de95035	77b6e2569	b761d0e6a
god	1105e690a	434ed1d87a	6ff2f0162
...			

- A hacker can pre-calculate values for a variety of common hashing functions & passwords
- Once a single matching value is found, the hacker can be sure they know exactly what hashing function is used for all passwords

What if our users have worse passwords?

id	username	password	createdAt	updatedAt
1	idbentley	uber733t	2020-09-04	2020-09-04
2	jortiz	a8skd8tew2	2020-09-04	2020-09-04
3	jmriley	arrakisbound	2020-09-04	2020-09-04

What if our users have worse passwords?

id	username	password	createdAt	updatedAt
1	idbentley	(419d952a) secret	2020-09-04	2020-09-04
2	jortiz	(21f5809f2) cat	2020-09-04	2020-09-04
3	jmriley	(7164738b6) bird	2020-09-04	2020-09-04
4	coney	(419d952a) secret	2020-09-04	2020-09-04
5	akeeler	(689f0139) phone	2020-09-04	2020-09-04
6	mreis	(7649ce6) password	2020-09-04	2020-09-04

Why we salt

id	username	password
1	idbentley	(419d952a) secret
2	jortiz	(21f5809f2) cat
3	jmriley	(7164738b6) bird
4	coney	(419d952a) secret
5	akeeler	(689f0139) phone
6	mreis	(7649ce6) password

guess	**hashFunction1**	**hashFunction2**	**hashFunction3**
secret	5ebe2294e	419d952a	2f50c26e60d
...			

Why we salt

id	username	password	salt
1	idbentley	(67e7c619) secret	0a9428e68eb353fe
2	jortiz	(00f3dd5dd) cat	b62dfb9b673bb532
3	jmriley	(b381e31a0) bird	3e5ddb7a8c97d9e
4	coney	(a02b6da4) secret	8013449c20351a0
5	akeeler	(18482bc8) phone	7a9c4c989b2fae0b
6	mreis	(6817ff23) password	9f9ccc20a8f18393

guess	**hashFunction1**	**hashFunction2**	**hashFunction3**
secret	5ebe2294e	419d952a	2f50c26e60d

Can’t precalculate for every salt

guess	hashFx Salt 1	hashFx Salt 2	hashFx Salt 3
love	b5c0b187f	f1fdd3cbe6	4a502bc853
secret	5ebe2294e	419d952a	2f50c26e60d
sex	2de95035	77b6e2569	b761d0e6a
god	1105e690a	434ed1d87a	6ff2f0162
...			

- Each password has its own salt. It’s as if each password is hashed with a different function
- Can’t pre calculate all values for each salt

What safety guarantees does this give us

- If our database is compromised:
  - Individual passwords may be compromised
  - User passwords overall are still protected