CENTRE FOR SECURE INFORMATION TECHNOLOGIES

Queen's University Belfast

Evolution of the Password

Dr Philip O'Kane

Lecturer at Queen's University Belfast
MSc Applied Cybersecurity Director

The Evolution of Password

- The Evolution of Password as a means of authentication
- What is:
  - Authentication
  - Authorization
- Password
  - A little history of passwords
  - Password usage
  - Password storage
  - Human behaviour
  - Targeting the weakness link – humans & passwords

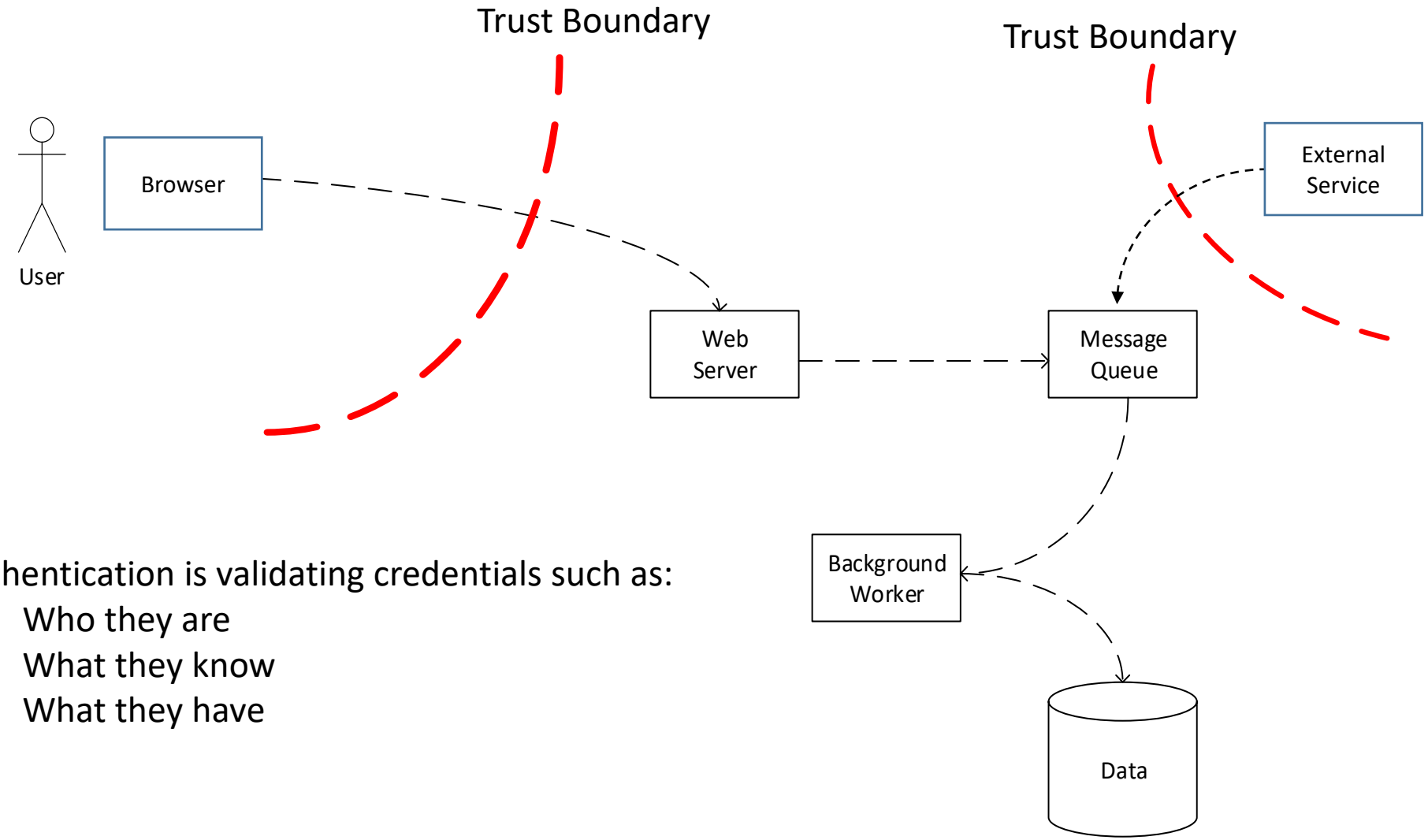
Authentication

- Authentication enables organizations to secure their system by permitting only authenticated users or processes to access protected resources.

- Authentication is a process of determining:
  - Who someone is
  - What something is

    OR
  - Who or what they declare themselves to be

Authentication



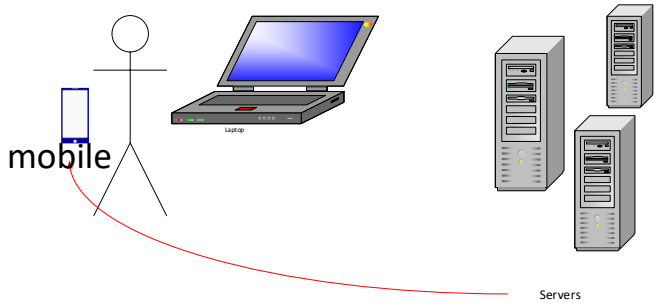
- Authentication is validating credentials such as:
  - Who they are
  - What they know
  - What they have

Authentication

- Identify someone
  - Something they know – password or pin

qwerty
password1

  - Something they have – mobile phone (one time password)



  - Something they are – Turing test

Authentication

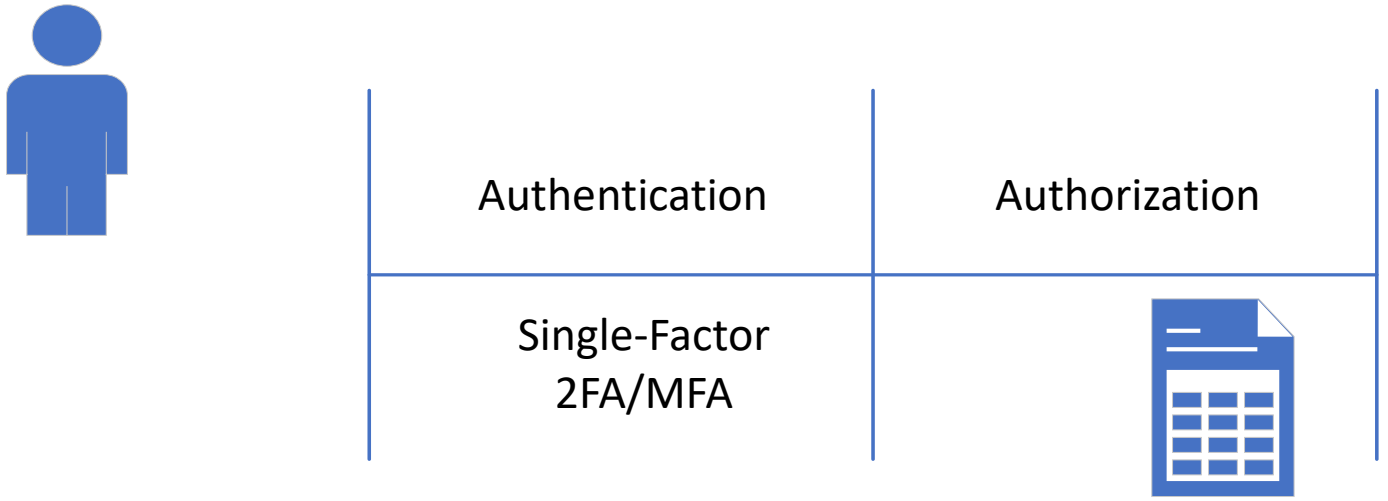
- Single-Factor
  - Username/User ID and password to verify their identity.
- Two-Factor (2FA)
  - Using a combination of two factors
    - Something they know = PIN, password
    - Something they have = bank card, mobile phone (text)

One time password

- Multi-Factor (MFA)
  - Uses multiple factors that are independent of each other
    - passwords
    - SMS – out of band SMS text message
    - Turing test

Authorization

- Authorization is the process of giving someone/something permission access a resource.
  - It occurs after identity has been validated
  - Verifies the rights to access resources, determined by business logic
  - Access Control List (ACL):
    - A table that tells the Operating System (OS) which access rights each user has to a particular system object.



Compatible Time-Sharing System

- Massachusetts Institute of Technology (MIT) Computation Centre was among the first to develop a 'Compatible Time-Sharing System' (CTSS). CTSS operated from1961 until 1973[1].

- MIT had developed an extensive CTSS (at that time) that allowed multiple researchers access to computation resources.
- However, they shared a mainframe as well as a single disk for data storages (files).

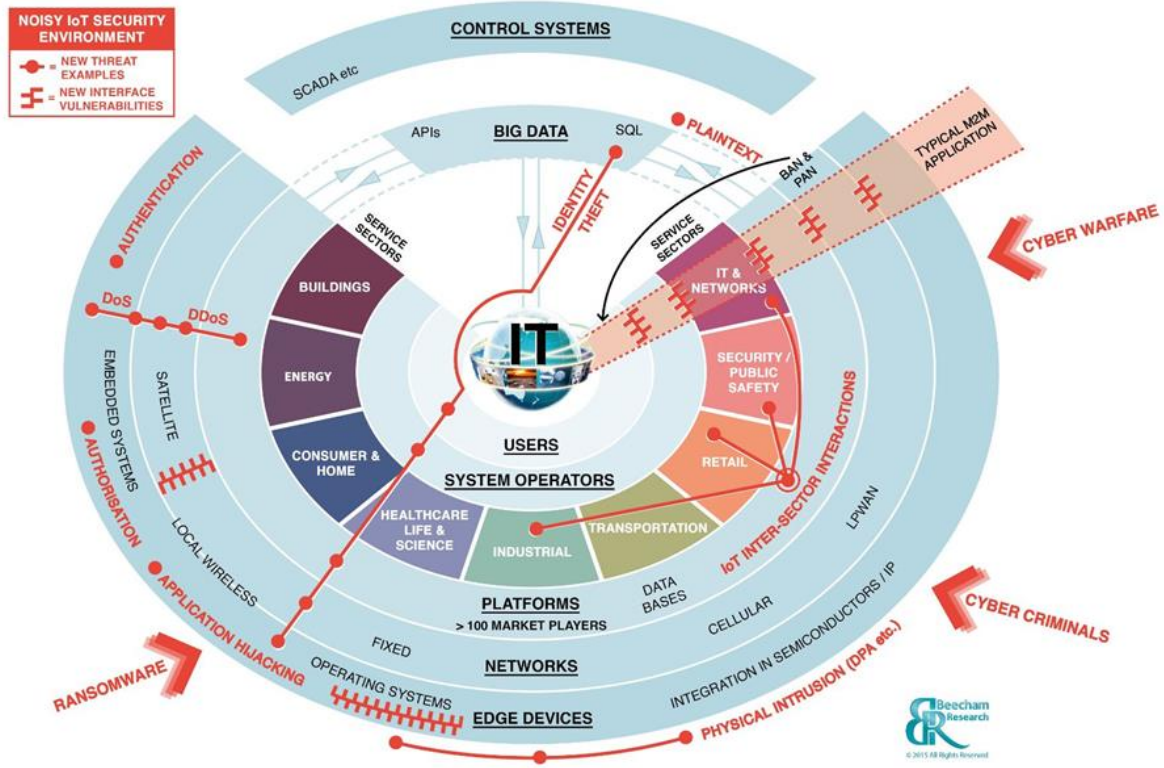
Compatible Time-Sharing System

- Password mechanism used to share a mainframe
  - used to restrict access to specific files and allotted time slots.
  - Primarily used by University professors and researchers
  - In a time before mainstream Hacking.

- Due to the simplicity
  - passwords became the default method of computer security.

Evolution of Password

- Passwords have gone beyond the original researchers:
  - Used by a wide range of individuals and systems to:
    - e-commerce (banking, shopping etc.)
    - Company information and systems
    - Critical infrastructure systems, Nuclear power station etc.
    - Emails accounts (AOL, Hotmail, Gmail etc.)
    - University - student information.



Hacking in the 1980s

- Examples of real-life hackers
  - 1981- ‘Chaos Computer Club’ (Germany)
  - 1982- ‘The 414*s*’
  - 1984- ‘Legion of Doom’, ‘Cult of the Dead Cow’
  - etc.
  - etc.
  - 1994- Russian hackers stole $10 millions from Citibank
  - etc.
  - 2014- Worldwide global cost of cybercrime is estimated at $445 billion
  - 2018- Worldwide global cost of cybercrime is estimated at $600 billion

The 414s

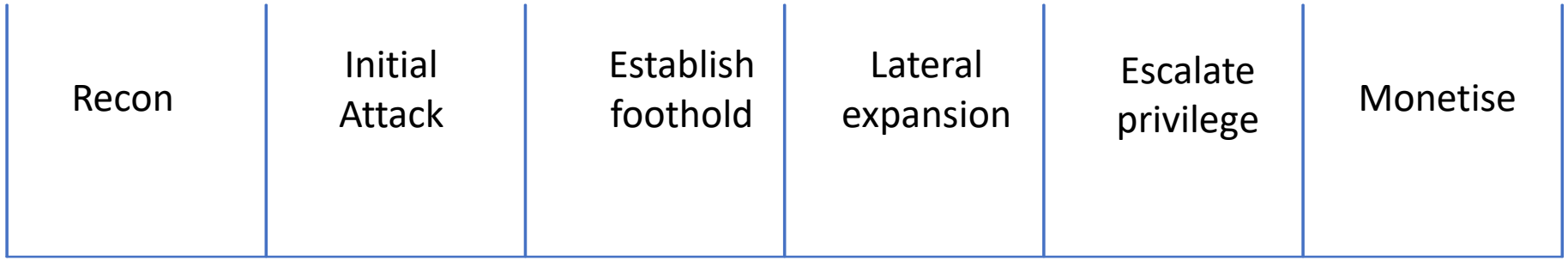
Carbanak

- Carbanak
  - Dangerous combination of hackers and professional criminals
  - A criminal gang that carries out Advanced Persistent Threat (APT) that often targets financial institutions (2014).
  - Kaspersky lab believe them to be a Russian/UK Cyber Crime organisation
  - The criminal manipulate their access to respective networks to steal money:
    - Manipulated databases to orchestrate an attack
    - ATMs – instruct ATMs to dispense cash, with money mules collecting money depositing the money to the criminal accounts.

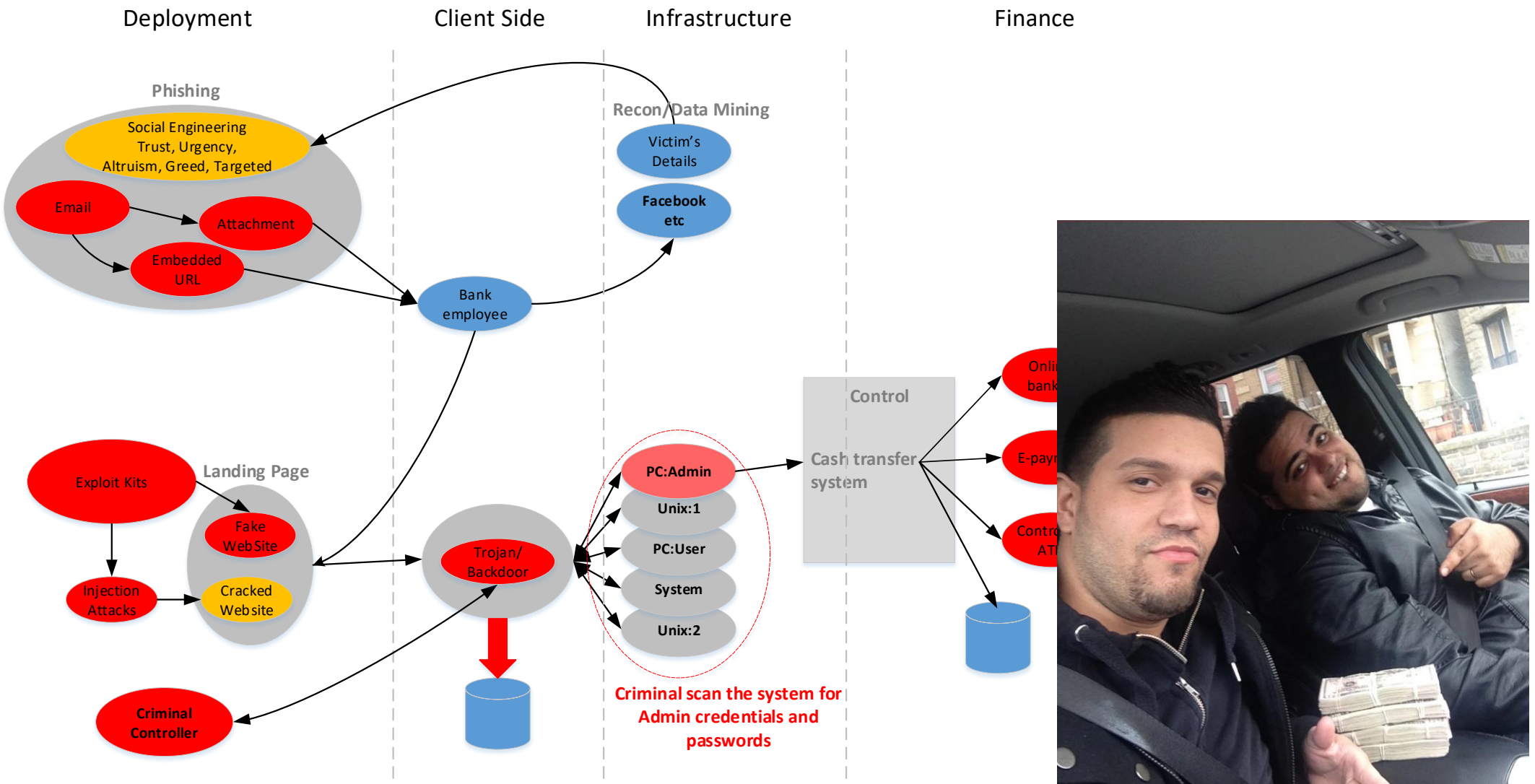
Carbanak creates file:

Family	File name	Size (bytes)	MD5
Trojan[Backdoor]/Win32.Carbanak	file.exe	404,992	a2643fe61f4b65704cfe1ebc55e2b301

APT

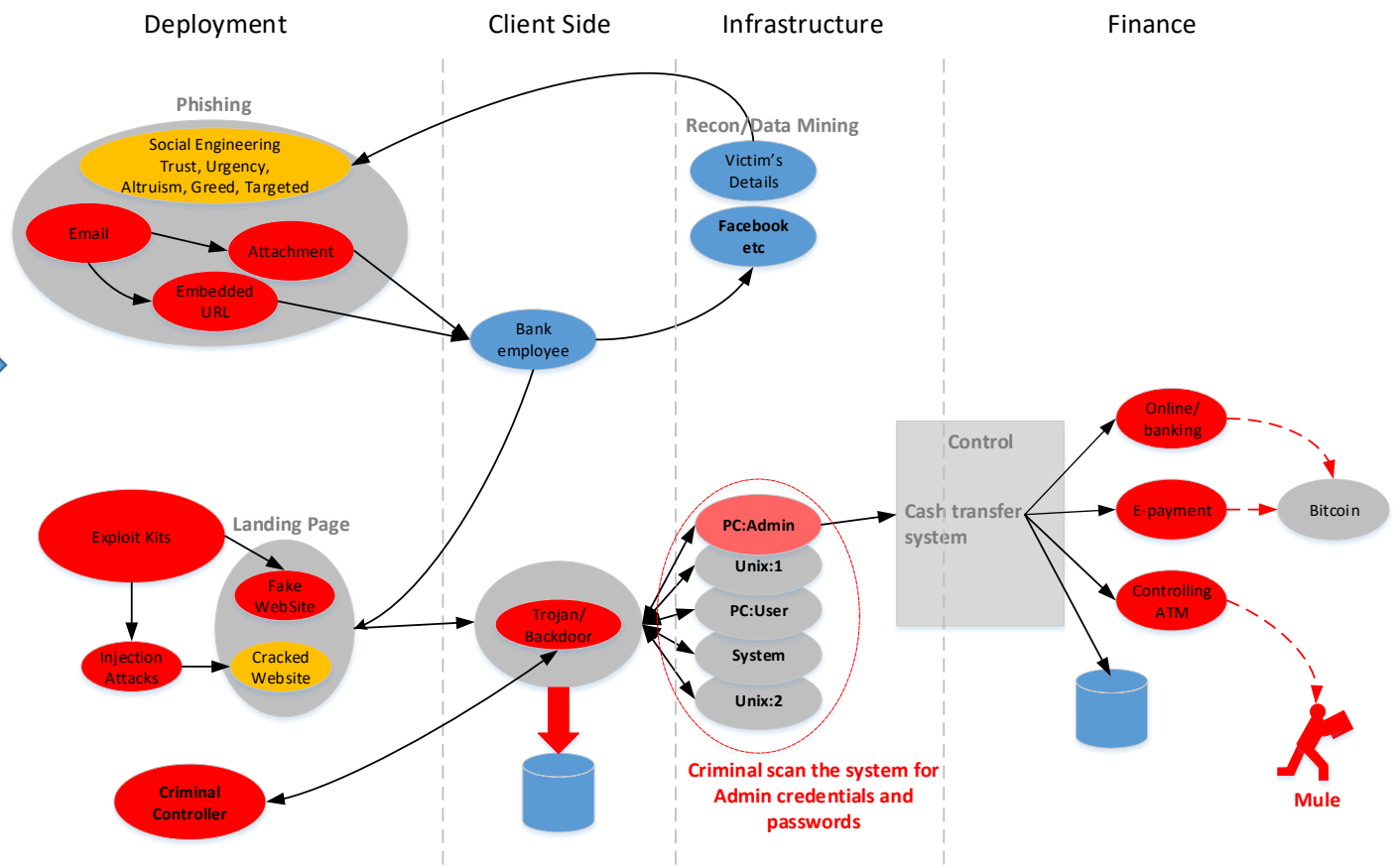

Recon
Initial Attack
Establish foothold
Lateral expansion
Escalate privilege
Monetise

Carbanak Attack



Picture ref: https://www.nytimes.com/2013/05/10/nyregion/eight-charged-in-45-million-global-cyber-bank-thefts.html

Hackers to Attackers



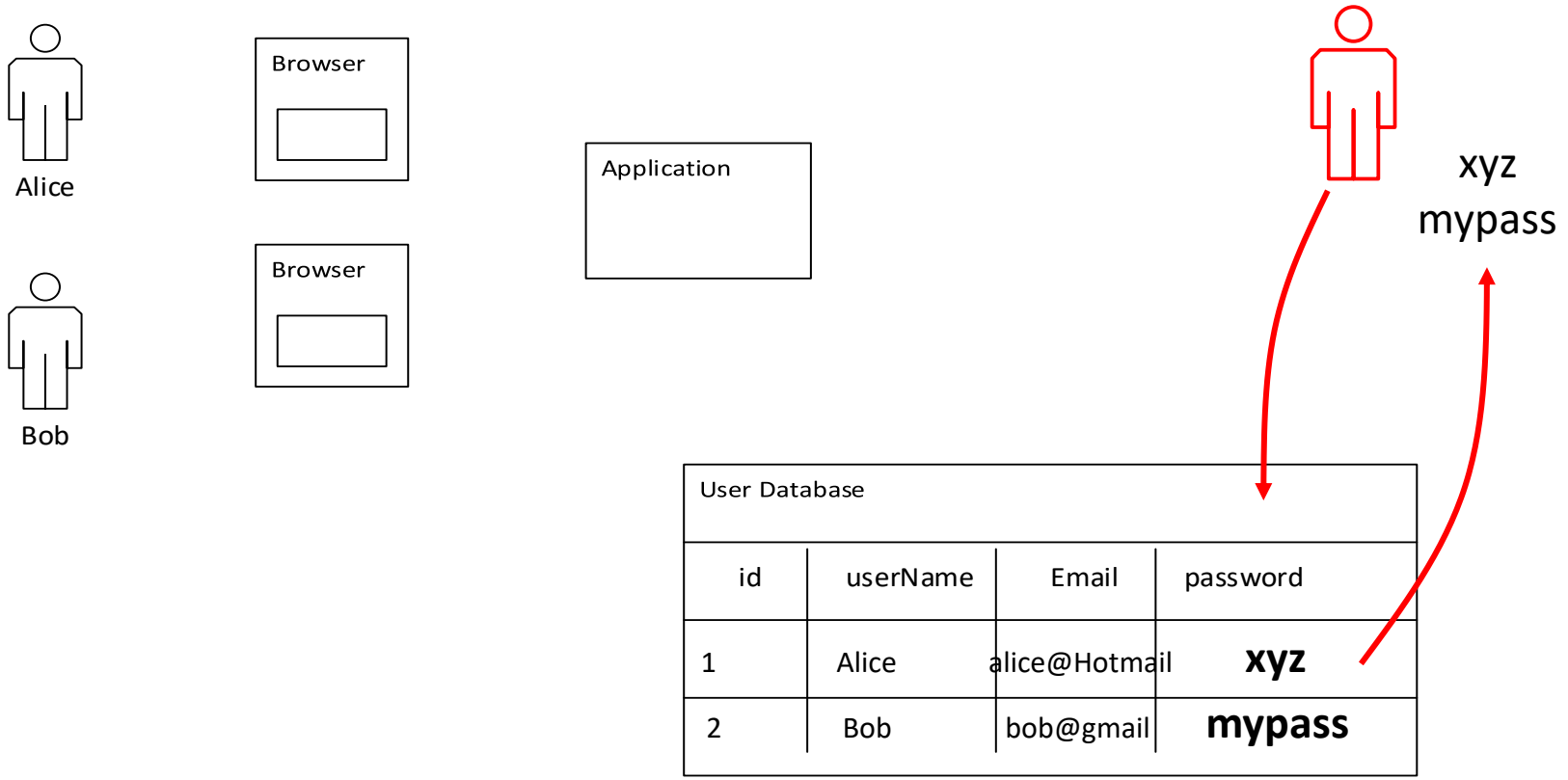
Internet a Perfect for Crime

- Todays Internet is:
  - Target rice environment
    - Banking eCommerce (Amazon, eBay, Insurance, etc.)
    - Media (Jobs, LinkedIn)
    - Social media (Facebook, Twitter etc.)
    - The human element (victim)
  - Perfect place for crime
    - Anonymous access to vast resources
    - The availability of criminal tools
    - No national or political boundaries
    - Law enforcement is limited
    - Numerous opportunities for money laundering

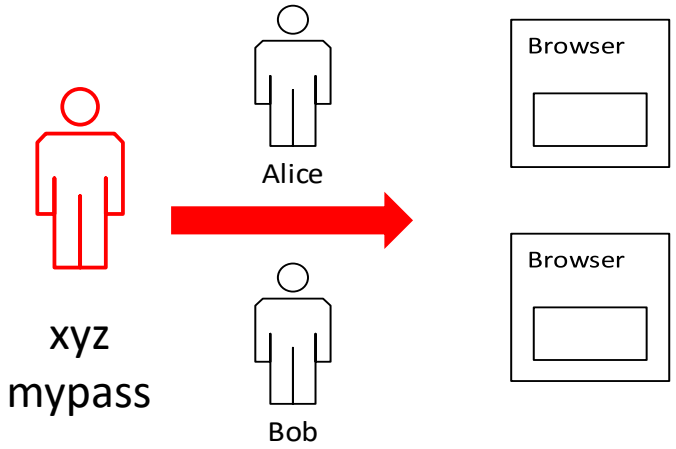
Breaches

- But how safe is it?
  - Recent breaches include:
    - BlueKai (2020, potentially 2b recorded left unprotected)
    - Capital One (2019, 100m details stolen)
    - British Airways (2019, 380K details stolen)
    - Marriot Hotels (2018, 500m details stolen)
- Cost – runs into Billons
  - Fix the breach and cover account monitoring for the victims
  - Actual losses credit (hackers monetising the crime)

Protecting Passwords



Protecting Passwords

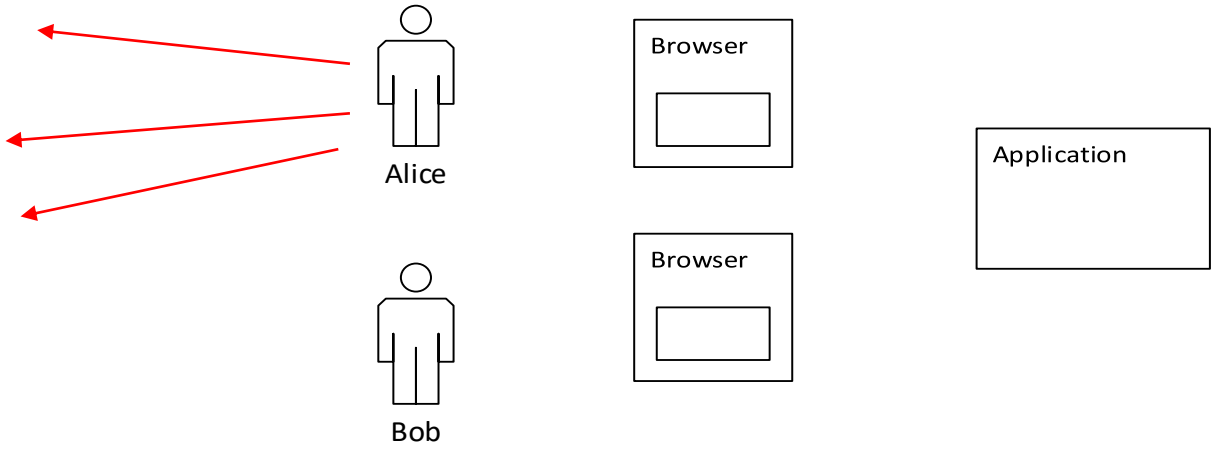


Application

User Database

id	userName	Email	password
1	Alice	alice@Hotmail	**xyz**
2	Bob	bob@gmail	**mypass**

Protecting Passwords



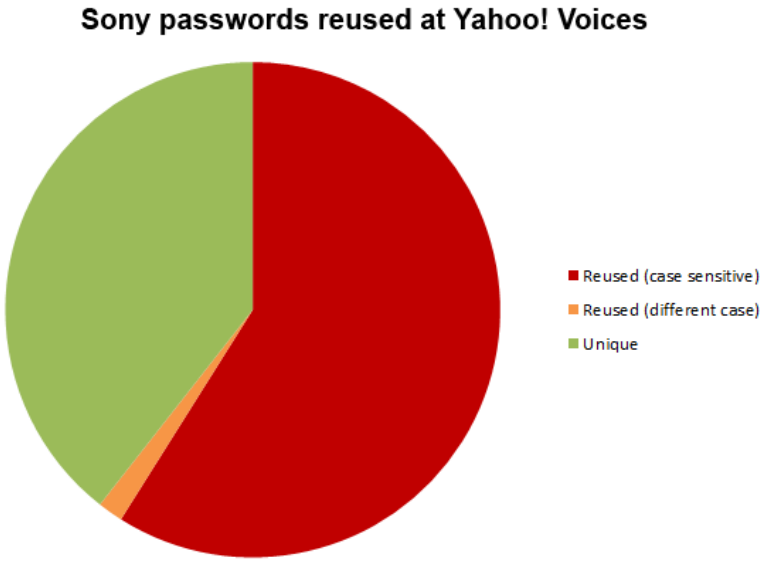
User Database			
id	userName	Email	password
1	Alice	alice@Hotmail	**xyz**
2	Bob	bob@gmail	**mypass**

- Wider impact:
  - Many users the same password for multiple accounts

Password Reuse

- Stats
  - 1% of passwords contain non-alphanumeric character
  - 4% contain two character types
  - 93% are 6 to 10 characters long

- A year after the Sony breach[1]:
  - *“59% of people were still using the exact same password on Yahoo! Voices.”*
  - A further 2% of passwords only differed by case.



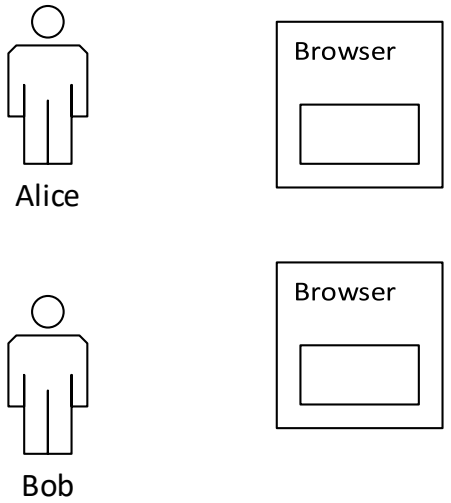
[1] https://www.troyhunt.com/what-do-sony-and-yahoo-have-in-common/

Protecting Passwords

- Given the:
  - Numerous data breaches
  - User poor password habits
    - We create simple easy to remember passwords
    - We use the same password across multiple accounts
    - We are very slow about updating our passwords, even after a breach
- The bad guys know our weakness and target password

- We can secure passwords
  - Cryptology:
    - Hashing

Protecting Passwords



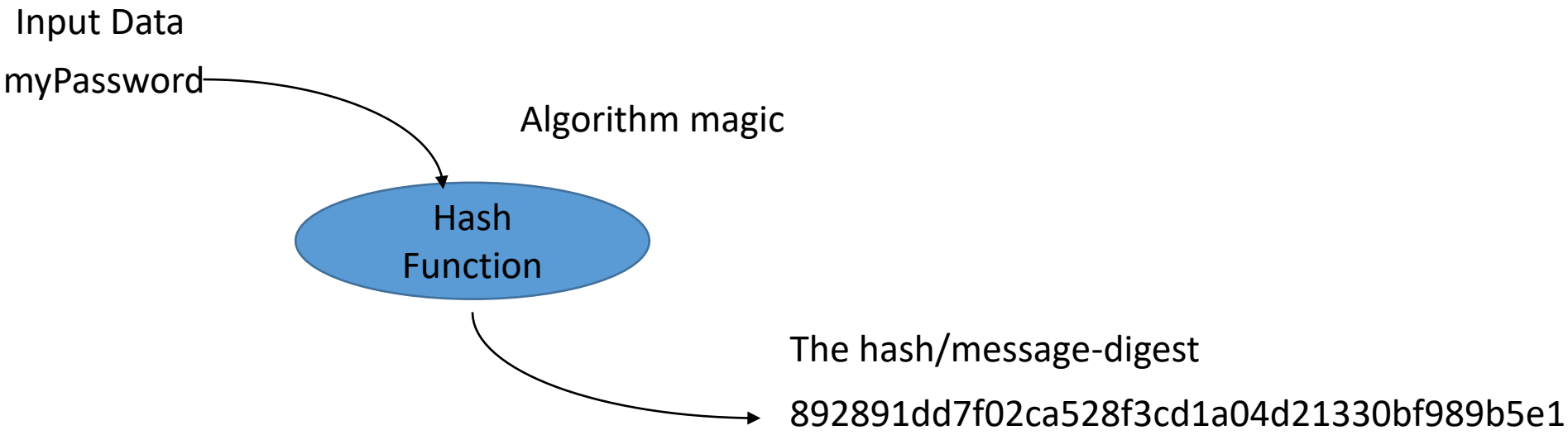
Not acceptable

User Database

id	userName	Email	password
1	Alice	alice@Hotmai	xyz
2	Bob	bob@gmail	mypass

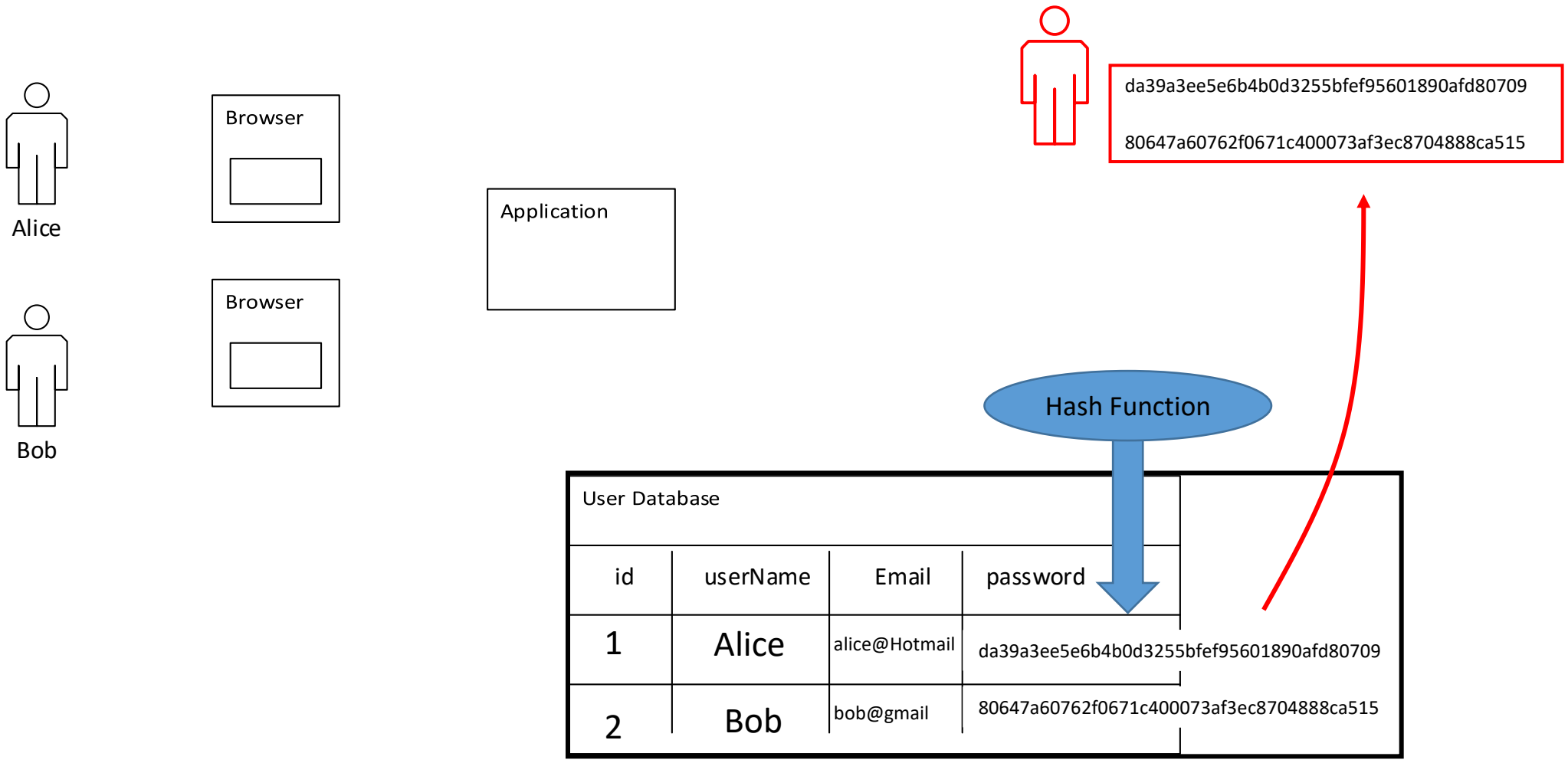
Hashing Algorithm

- Hashing algorithm is a complex mathematic function that transforms an input (string) in to a seemingly random sequence of numbers



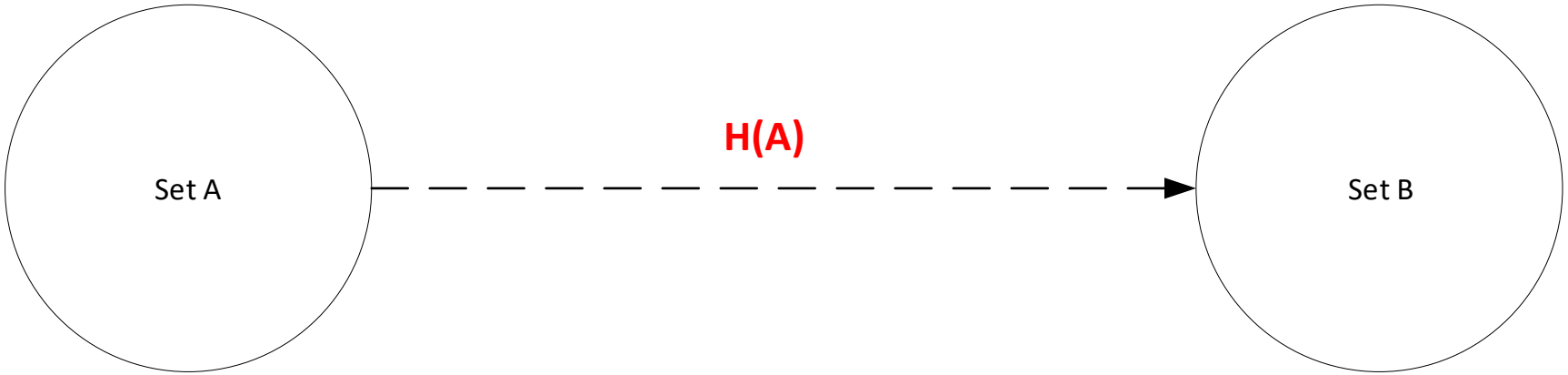
- **Encryption**: is often seen as a way of temporarily storing data until it is needed and then is unencrypted
- **Hashing**: is One-way -> you do not unencrypt (or un-hash) the hashed data

Protecting Passwords



Hash Function

- A hash function compresses a set (information) to another set usually a shorter set.



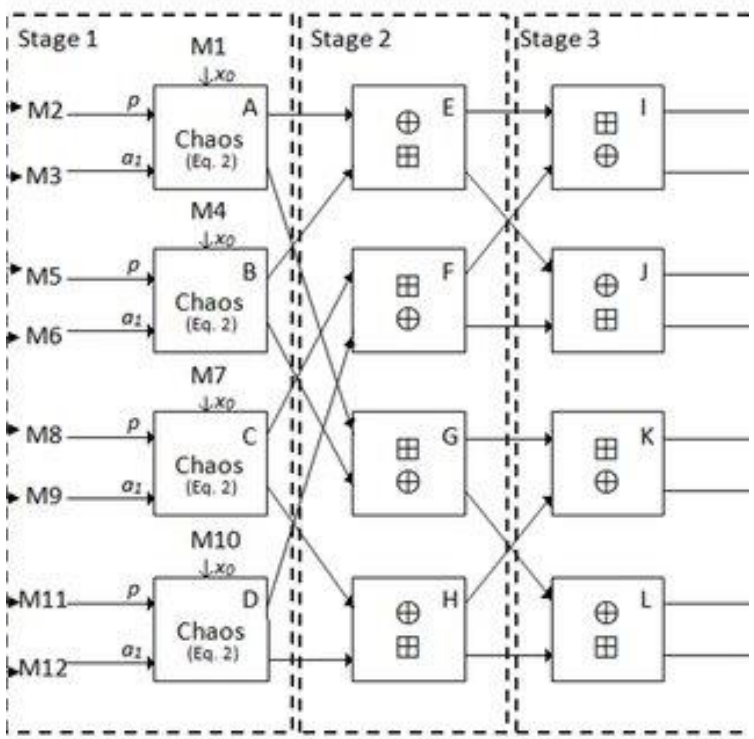
Set A (Input Data)	Set B (Hash)
Philip	da39a3ee5e6b4b0d3255bfef95601890afd80709
Phi1ip	9ed7bcc82bc8cc78aa550908f37a532bf79112e5
Philip1	80647a60762f0671c400073af3ec8704888ca515
Philip3	4e1456cb612fa7eebaac4714668aa7209de183da

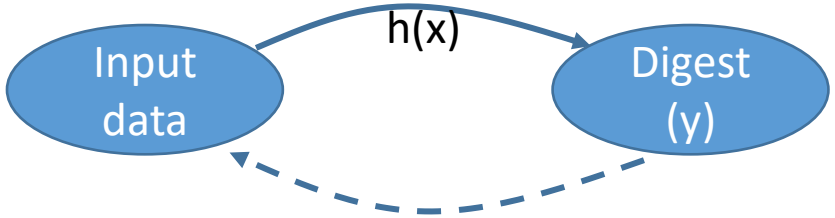
Cryptographic Hash Function

- Hash property
  - What does hashing give us?
    - Hides the password
  - Hash functions are normally public knowledge
    - What if a hacker could determine the original password (message) from the hash (digest)?
- Properties:
  - Deterministic
  - Collision Resistance (CR)
    - Target Collision Resistance (TCR), weak collision resistance
  - Looks random
    - Non-Malleability
  - Public
    - Onewayness

Cryptographic Hash Function

- Infeasible to reverse engineer x
  - Algorithm is complex and difficult to reverse

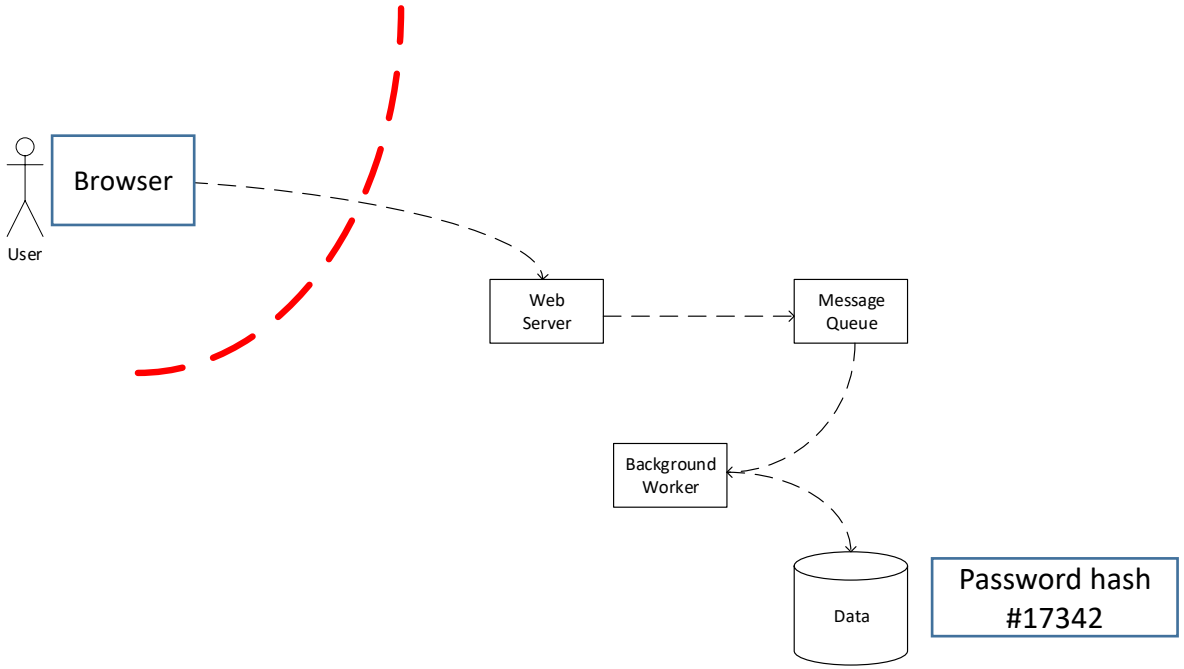




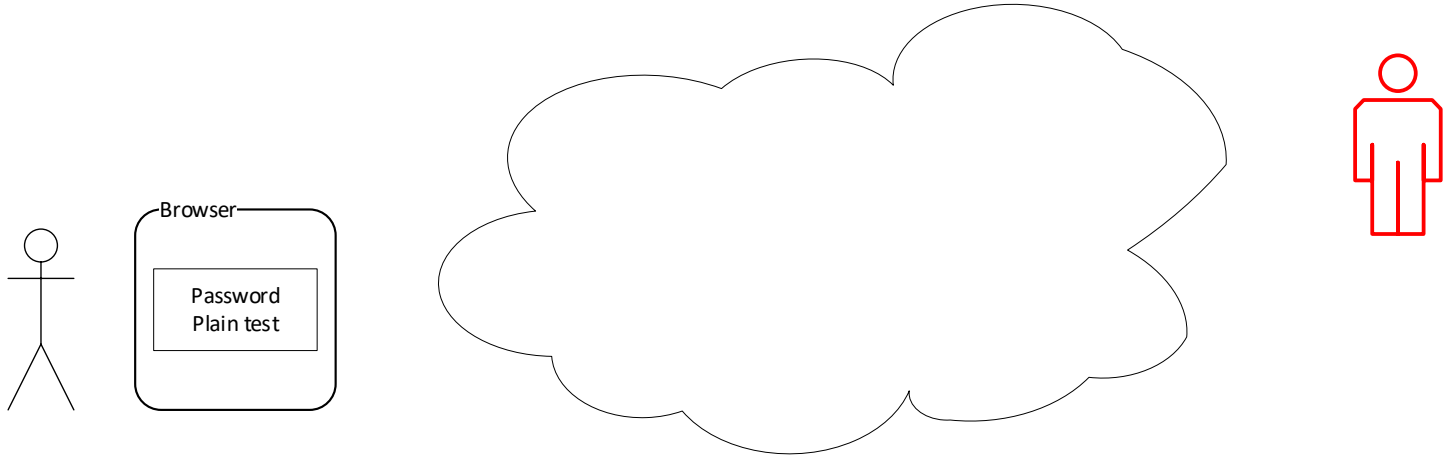
Set A (Input Data)	Set B (Hash)
Philip	da39a3ee5e6b4b0d3255bfef95601890afd80709
Phi1ip	9ed7bcc82bc8cc78aa550908f37a532bf79112e5
Philip1	80647a60762f0671c400073af3ec8704888ca515
Philip3	4e1456cb612fa7eebaac4714668aa7209de183da

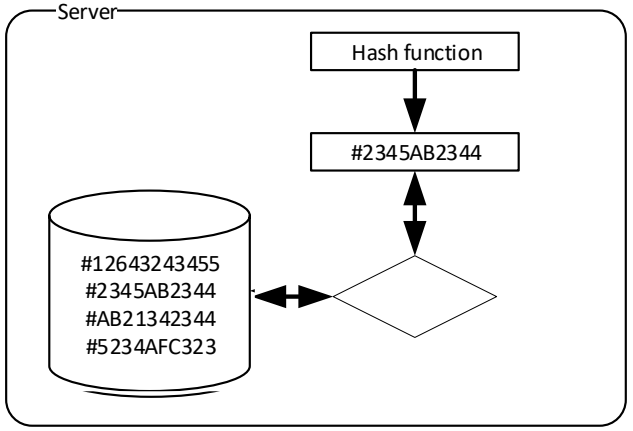
Cryptographic Hash Function

- Properties:
  - Deterministic
  - Looks random
  - Public
    - Onewayness

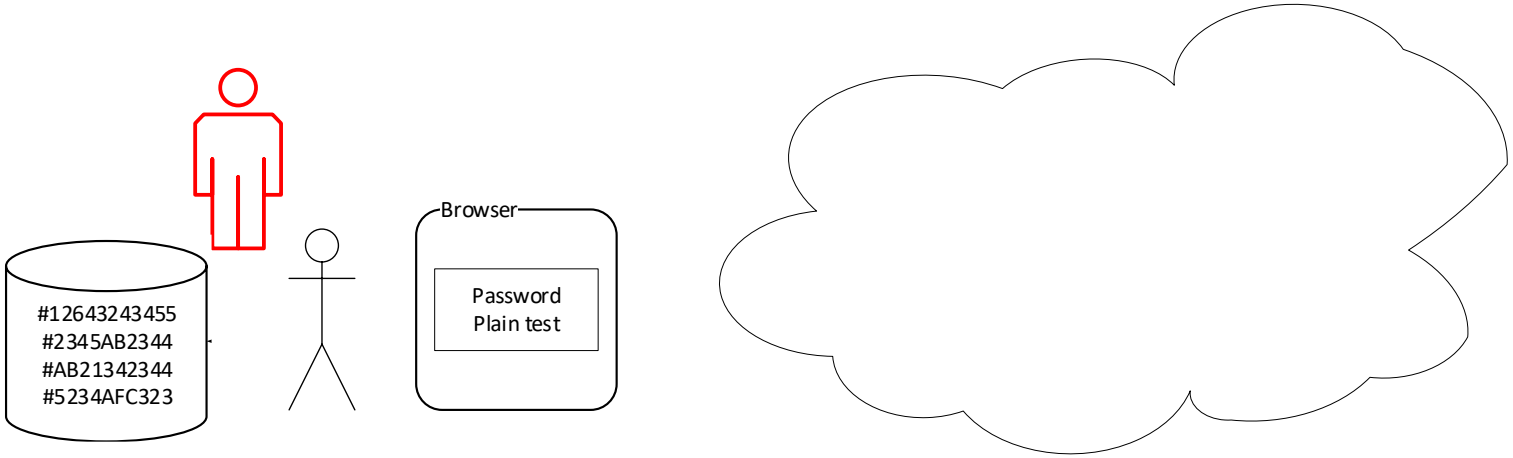


Password Storage

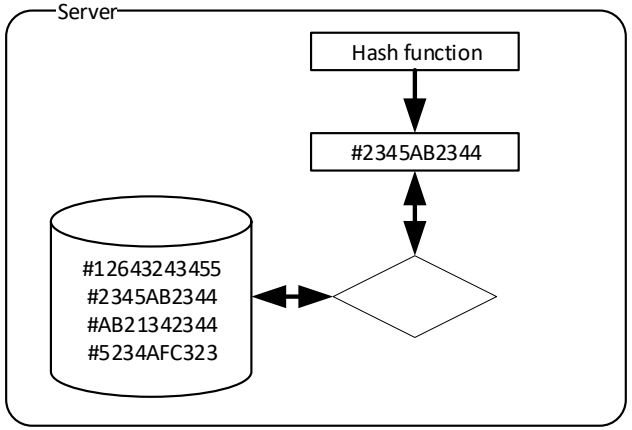




Password Storage



In theory the hash will be no good to the hacker?



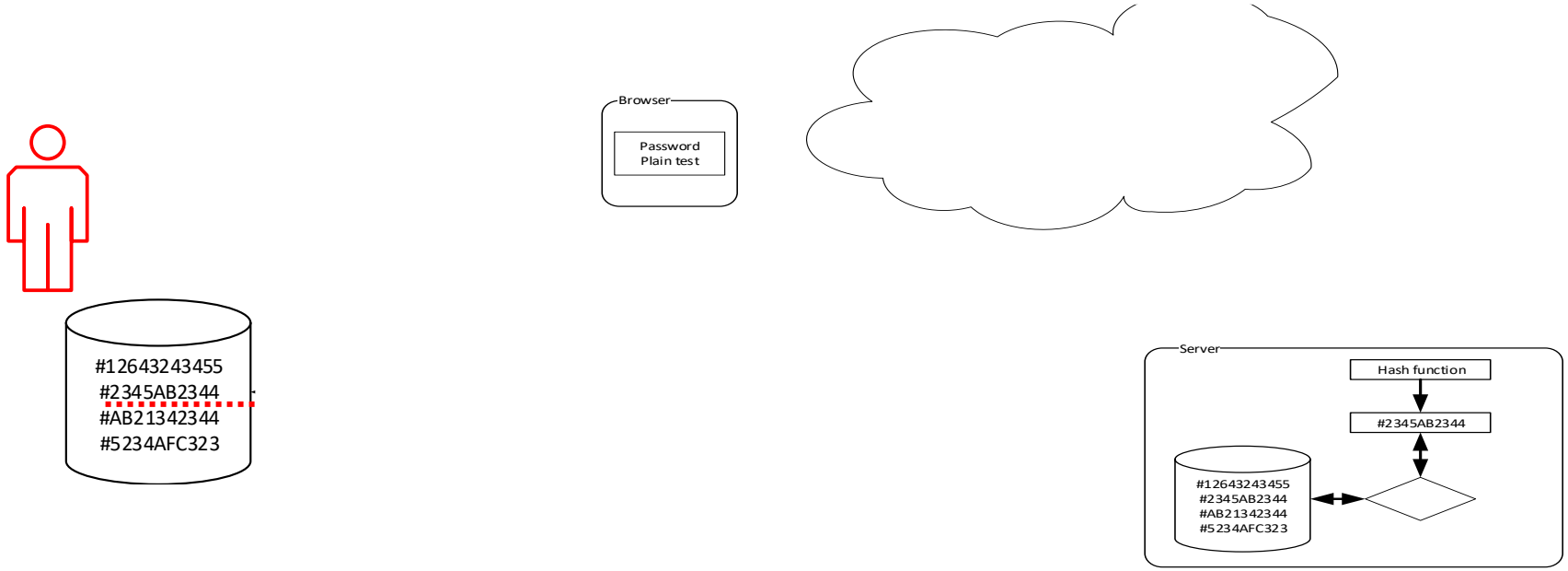
Reverse Lookup Table

Hash value	Original password
5413ee24723bba2c5a6ba2d0196c78b3ee4628d1	myPassword
7af2d10b73ab7cd8f603937f7697cb5fe432c7ff	Admin123
cd027069371cdb4f80c68dcfb37e6f4a1bdb0222	User123
521f17fee5fc459d7458c18b5220fc10412bed1e	myPa55w0rd
7d018bb3df0e523692845af1f27e992ce8a41650	mySecret
f8ec29af355cd3fb52ddaf5767134061a8d3ea13	tooManyPasswords

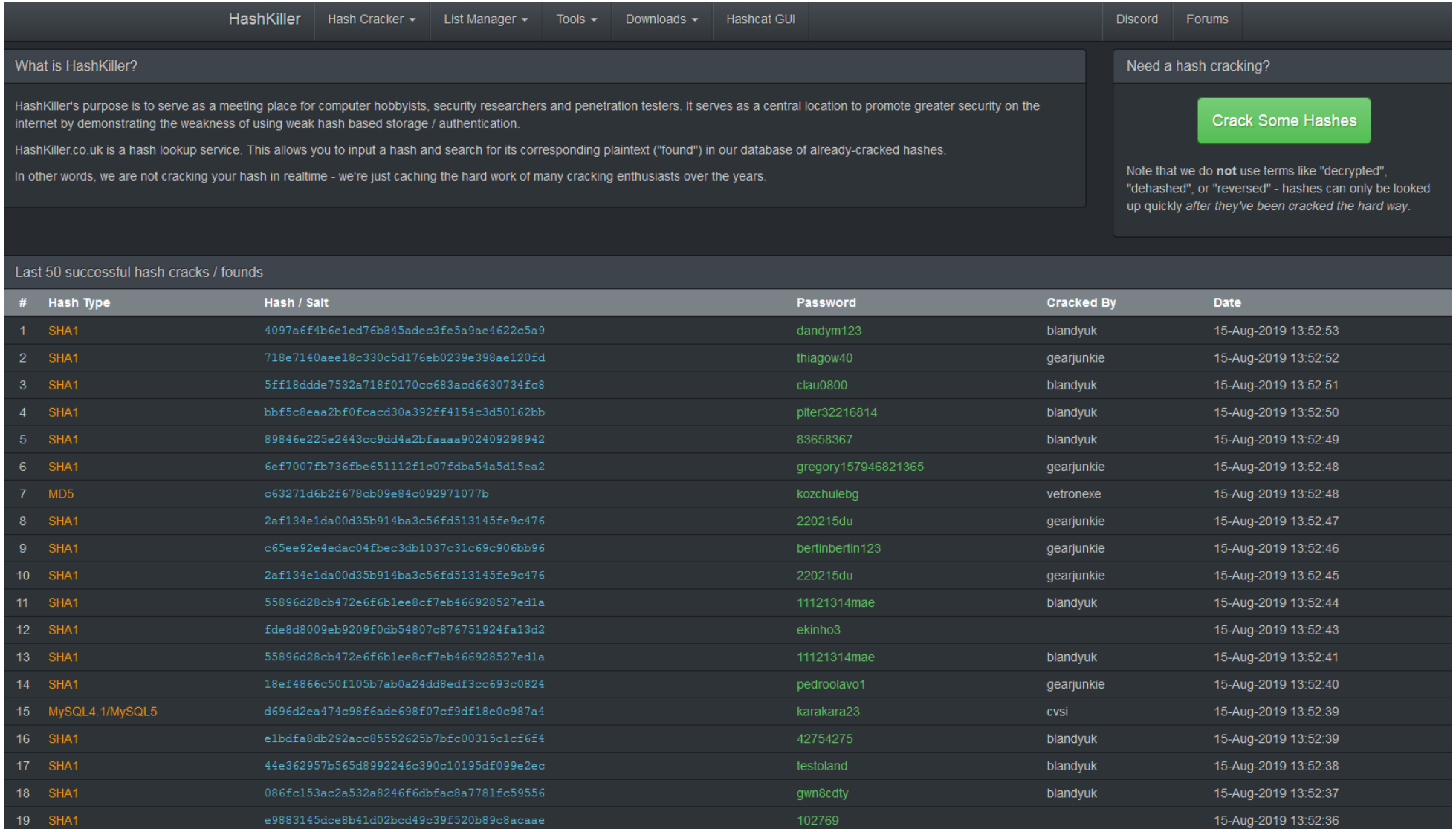
These tables can contain 100s millions of entries

Reverse Lookup Table

Hash value	Original password
5413ee24723bba2c5a6ba2d0196c78b3ee4628d1	myPassword
7af2d10b73ab7cd8f603937f7697cb5fe432c7ff	Admin123
cd027069371cdb4f80c68dcfb37e6f4a1bdb0222	User123
521f17fee5fc459d7458c18b5220fc10412bed1e	myPa55w0rd
2345ac23443412341a23232323232323232323	mySecretpassWord
f8ec29af355cd3fb52ddaf5767134061a8d3ea13	tooManyPasswords



Reverse Lookup Tables



HashKiller | Hash Cracker | List Manager | Tools | Downloads | Hashcat GUI | Discord | Forums

What is HashKiller?

HashKiller's purpose is to serve as a meeting place for computer hobbyists, security researchers and penetration testers. It serves as a central location to promote greater security on the internet by demonstrating the weakness of using weak hash based storage / authentication.

HashKiller.co.uk is a hash lookup service. This allows you to input a hash and search for its corresponding plaintext ("found") in our database of already-cracked hashes.

In other words, we are not cracking your hash in realtime - we're just caching the hard work of many cracking enthusiasts over the years.

Need a hash cracking?

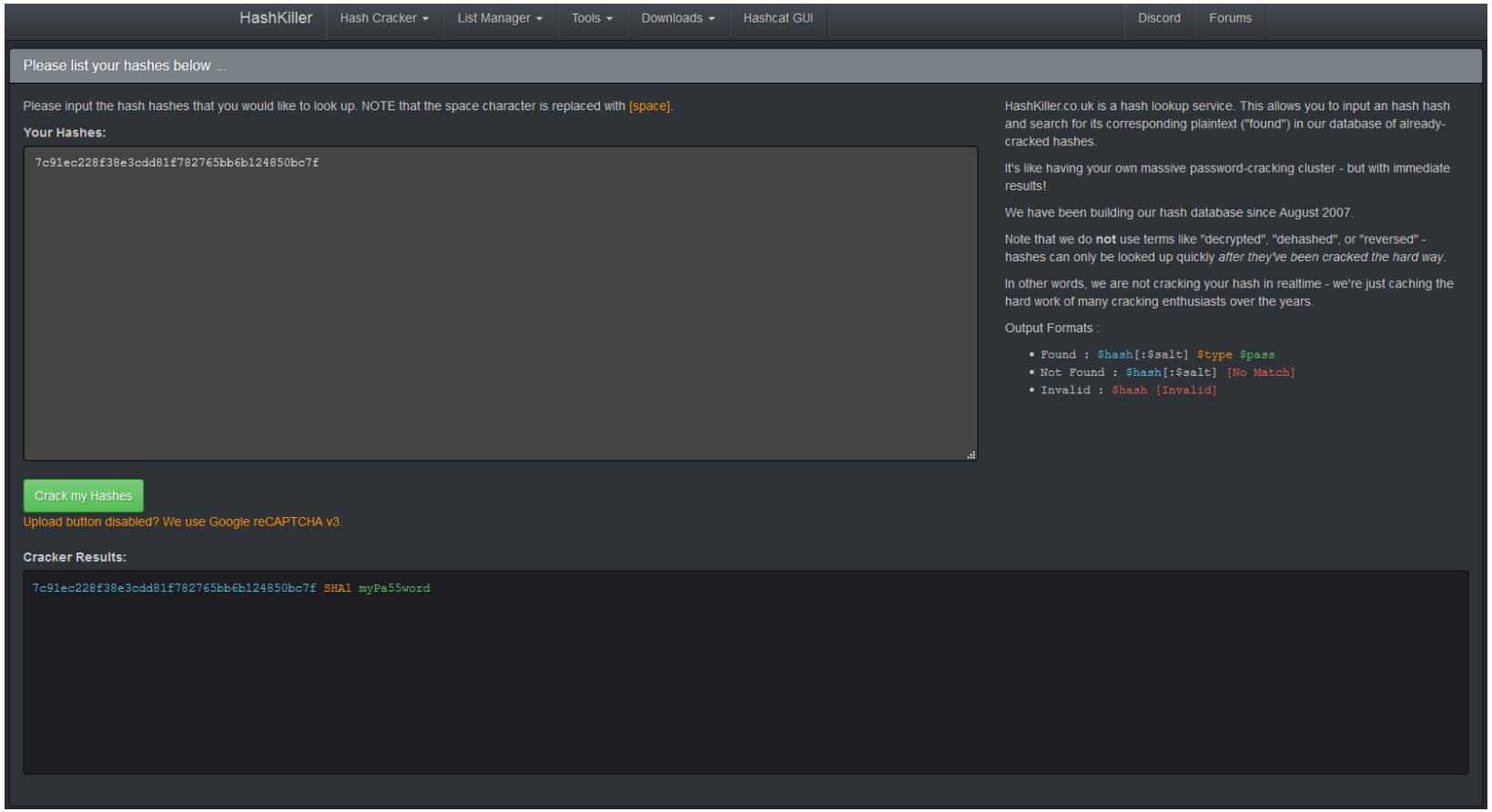
Crack Some Hashes

Note that we do **not** use terms like "decrypted", "dehashed", or "reversed" - hashes can only be looked up quickly *after they've been cracked the hard way*.

Last 50 successful hash cracks / founds

#	Hash Type	Hash / Salt	Password	Cracked By	Date
1	SHA1	4097a6f4b6e1ed76b845adec3fe5a9ae4622c5a9	dandym123	blandyuk	15-Aug-2019 13:52:53
2	SHA1	718e7140aee18c330c5d176eb0239e398ae120fd	thiagow40	gearjunkie	15-Aug-2019 13:52:52
3	SHA1	5ff18ddde7532a718f0170cc683acd6630734fc8	clau0800	blandyuk	15-Aug-2019 13:52:51
4	SHA1	bbf5c8eaa2bf0fcacd30a392ff4154c3d50162bb	piter32216814	blandyuk	15-Aug-2019 13:52:50
5	SHA1	89846e225e2443cc9dd4a2bfaaaa902409298942	83658367	blandyuk	15-Aug-2019 13:52:49
6	SHA1	6ef7007fb736fbe651112f1c07fdba54a5d15ea2	gregory157946821365	gearjunkie	15-Aug-2019 13:52:48
7	MD5	c63271d6b2f678cb09e84c092971077b	kozchulebg	vetronexe	15-Aug-2019 13:52:48
8	SHA1	2af134e1da00d35b914ba3c56fd513145fe9c476	220215du	gearjunkie	15-Aug-2019 13:52:47
9	SHA1	c65ee92e4edac04fbec3db1037c31c69c906bb96	bertinbertin123	gearjunkie	15-Aug-2019 13:52:46
10	SHA1	2af134e1da00d35b914ba3c56fd513145fe9c476	220215du	gearjunkie	15-Aug-2019 13:52:45
11	SHA1	55896d28cb472e6f6b1ee8cf7eb466928527ed1a	11121314mae	blandyuk	15-Aug-2019 13:52:44
12	SHA1	fde8d8009eb9209f0db54807c876751924fa13d2	ekinho3		15-Aug-2019 13:52:43
13	SHA1	55896d28cb472e6f6b1ee8cf7eb466928527ed1a	11121314mae	blandyuk	15-Aug-2019 13:52:41
14	SHA1	18ef4866c50f105b7ab0a24dd8edf3cc693c0824	pedroolavo1	gearjunkie	15-Aug-2019 13:52:40
15	MySQL4.1/MySQL5	d696d2ea474c98f6ade698f07cf9df18e0c987a4	karakara23	cvsi	15-Aug-2019 13:52:39
16	SHA1	e1bdfa8db292acc85552625b7bfc00315c1cf6f4	42754275	blandyuk	15-Aug-2019 13:52:39
17	SHA1	44e362957b565d8992246c390c10195df099e2ec	testoland	blandyuk	15-Aug-2019 13:52:38
18	SHA1	086fc153ac2a532a8246f6dbfac8a7781fc59556	gwn8cdty	blandyuk	15-Aug-2019 13:52:37
19	SHA1	e9883145dce8b41d02bcd49c39f520b89c8acaae	102769		15-Aug-2019 13:52:36

Reverse Lookup Tables



Attackers

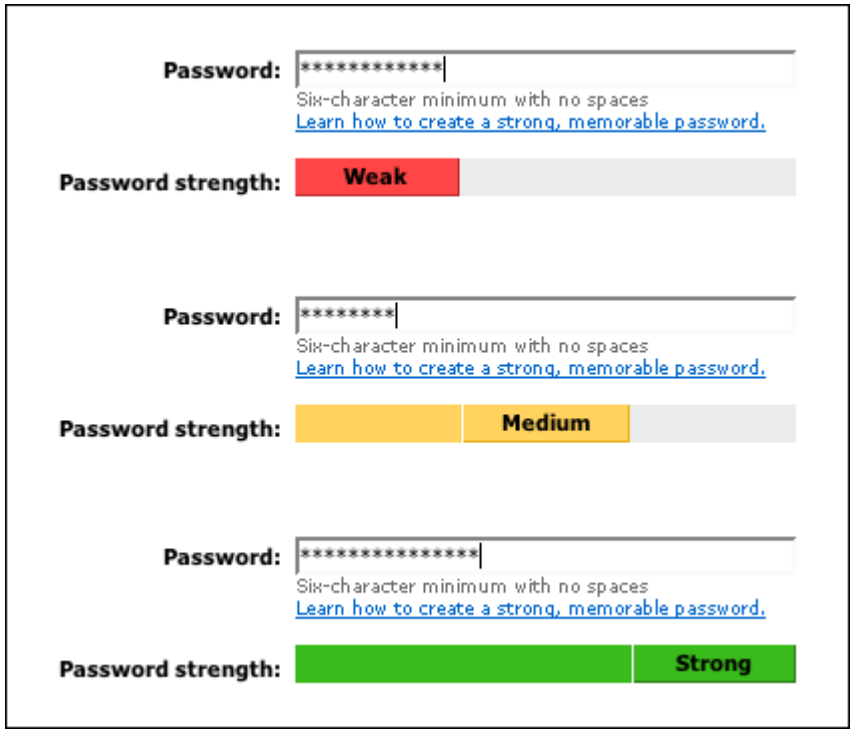
- Dictionary attack
  - List for words and word-pattern: Example: password, pas55w0rd etc.
- Brute force attack
  - Try everything, Fuzz the pattern
- Reverse Lookup tables attack
- Social Engineering/Phishing
- Malware (key loggers, memory scrappers)
- Offline cracking
- Spidering

Human Behaviour

- Humans:
  - Use the same password for multiple accounts
  - Use simple easy to remember passwords
  - Fail to update the passwords
- Password resets:
  - Leak information on social media
  - Link accounts with each other
  - Poor security questions

Password strength

- Password strength checker[1]



Weak password pattern	Is it memorable?	Time to crack
A common word (example: december)	Yes	18 milliseconds
An easily-typed spatial word (example: qwerty)	Yes	10 milliseconds
The family dog (example: rex)	Yes	27 milliseconds
An important number, such as DOB, Wedding (example: 03261981)	Memorable to the user	2.213 seconds
A word with trivial letter→number substitutions (example: pa55w0rd)	Sort of memorable, but you may forget which letters are substituted for numbers.	639 milliseconds

[1] http://designinginterfaces.com/patterns/password-strength-meter/

Password Compromise

- Data breach
  - System vulnerabilities
  - “According to Verizon, 81% of all data breaches take advantage of stolen or weak passwords.”
- Weak passwords
  - Attackers target our behaviour
- Poor software design/implementation
- Password reset methods
  - Security questions, humans leak information
- Malware
  - Keyloggers, especially if you use public computers

Weak	Still easy to crack	A little better
BankLogin	BankLogin13	BankLogin!3

- Passphrase
- Use randomly chosen words - four

Weak password pattern	Is it memorable?	Time to crack
Four + random words	To you	Years ++

Weak password pattern	Is it memorable?	Time to crack
A common word (example: december)	Yes	18 milliseconds
An easily-typed spatial word (example: qwerty)	Yes	10 milliseconds
The family dog (example: rex)	Yes	27 milliseconds
An important number, such as DOB, Wedding (example: 03261981)	Memorable to the user	2.213 seconds
A word with trivial letter→number substitutions (example: pa55w0rd)	Sort of memorable, but you may forget which letters are substituted for numbers.	639 milliseconds

Principles of software security Architecture

1) Economy of Mechanism
2) Fail-safe defaults (permission based access control)
3) Complete Mediation (check permission before access to objects)
4) Open design (design should not be secret)
5) Separation of Privilege (Multiple conditions required to complete task)
6) Least Privilege (minimum rights)
7) Least Common Mechanism (minimize shared subsystems/roles)
8) Psychological acceptability (usability)
9) Defence in Depth

Psychological acceptability (usability)

Hit any key to continue

Ref:clipartxtras.com