Sensitive Data Exposure

Overview

Sensitive Data Exposure involves the unintentional exposure of sensitive data which was not properly protected or cryptographically-secured. If sensitive data is stored in plain text or encrypted with a weak or deprecated encryption algorithm it can easily be recovered following an attack such as SQL injection. For instance, if a password database uses simple hashes to store user passwords and an attacker is somehow able to gain access to the database, the attacker may be able to look up these hashed passwords in something called a rainbow table. A rainbow table is a precomputed table containing the output of cryptographic hash functions, specifically used for cracking password hashes. If the hashed passwords are listed in the rainbow table, the attacker now has a list of viable user passwords.

There is a wide spectrum of encryption algorithms available. In this project, we decided to utilize the following encoding scheme and encryption algorithms in order to test their abilities at preventing Sensitive Data Exposure following an SQL injection.

Base64

Base64 is a binary-to-text encoding scheme designed to transfer data stored in a binary format over channels which support textual data. It is widely used on the Web due to its ability to embed image, sound, HTML and CSS files, and is also commonly used for sending email attachments. It translates binary data represented in an ASCII string format into a radix-64 representation. Base64 is not an encryption algorithm per se - it simply encodes some piece of data into an alternate syntax and it can be decoded by anyone.

Message-Digest Algorithm 5 (MD5)

The MD5 message-digest algorithm is a commonly-used hash function, originally designed to be a secure cryptographic hash for authenticating digital signatures. It is able to process any length of message as input and generate a 128-bit hash (or ‘message digest’) as output. It is no longer considered to be a secure cryptographic function due its numerous vulnerabilities. For instance, MDS is particularly vulnerable to collision attacks; that is, when two distinct inputs produce identical hashes. It can still, however, be used as a non-cryptographic checksum to verify data integrity and detect unintentional data corruption.

Secure Hash Algorithm 256 (SHA-256)

SHA-256 is a member of the family of Secure Hash Algorithms (SHA), a set of cryptographic hash functions designed by the National Security Agency (NSA). The number ‘256’ which forms part of its name indicates that it produces a hash value of 256 bits. It is a fast and very popular cryptographic hashing function with a wide range of applications, including password hashing in

Linux systems and verification of Bitcoin transactions. SHA-256 is constructed using the Merkle-Damgård structure, a method of building a collision-resistant hash function from a one-way compression function. Although this method of construction makes it secure, it actually makes it vulnerable to length extension attacks.

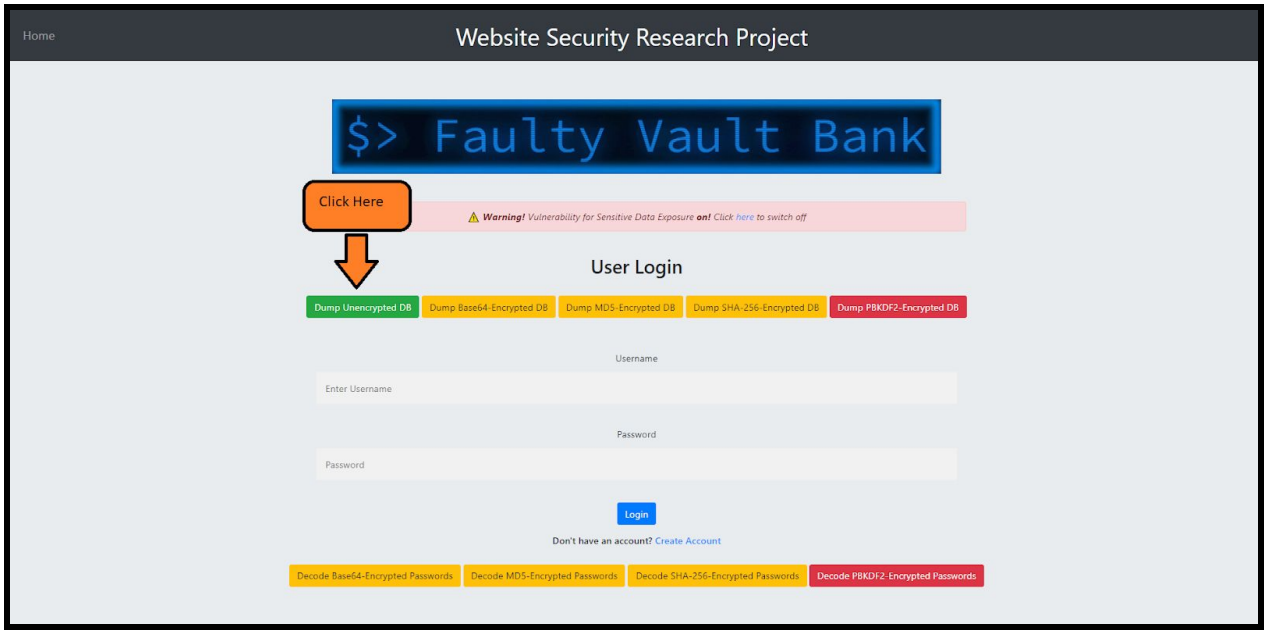
Password-Based Key Derivation Function 2 (PBKDF2)

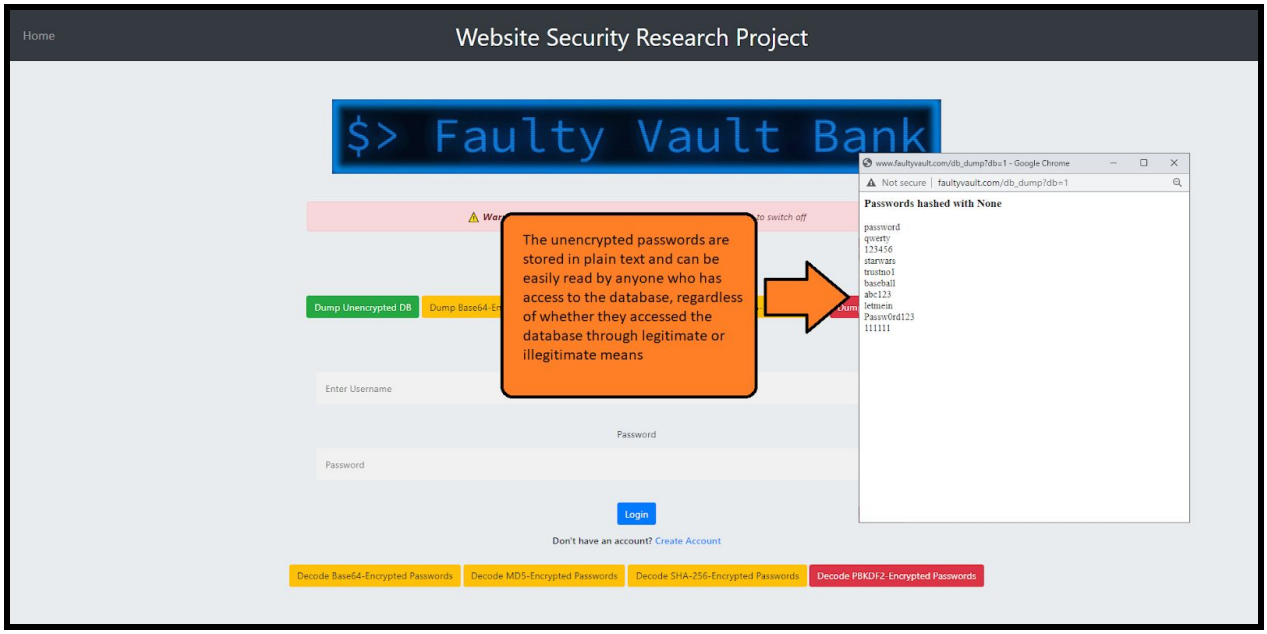
PBKDF2 is a key derivation function which is part of the RSA Laboratories' Public-Key Cryptography Standards (PKCS) series. RFC 8018 (2017) recommends PBKDF2 for password hashing. PBKDF2 is resistant to dictionary and rainbow table attacks. It utilizes a pseudorandom function such as a hash-based message authentication code (HMAC) and applies this to the inputted password. It then adds a salt value; that is, a random string of data, to the input. It repeats this process over and over again in order to generate a derived key which it uses as its cryptographic key in successive operations. PBKDF2 is slow by design. Fortunately, the additional computational work required to produce the cryptographic key, known as key stretching, makes password cracking much more difficult.

Attempts to Decrypt the Passwords

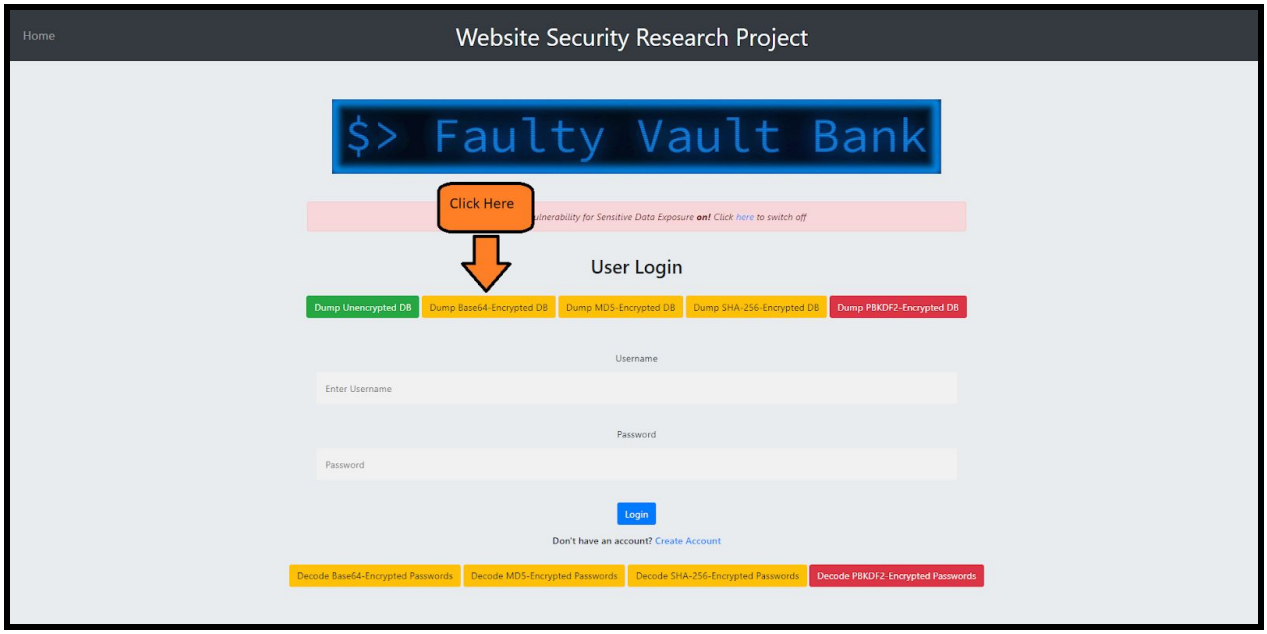
In the following scenario the website's database was dumped following an SQL injection, exposing user passwords. In each case, the passwords are encrypted with different encryption or encoding algorithms, and in one case the passwords stored in plain text. We attempt to crack the passwords.

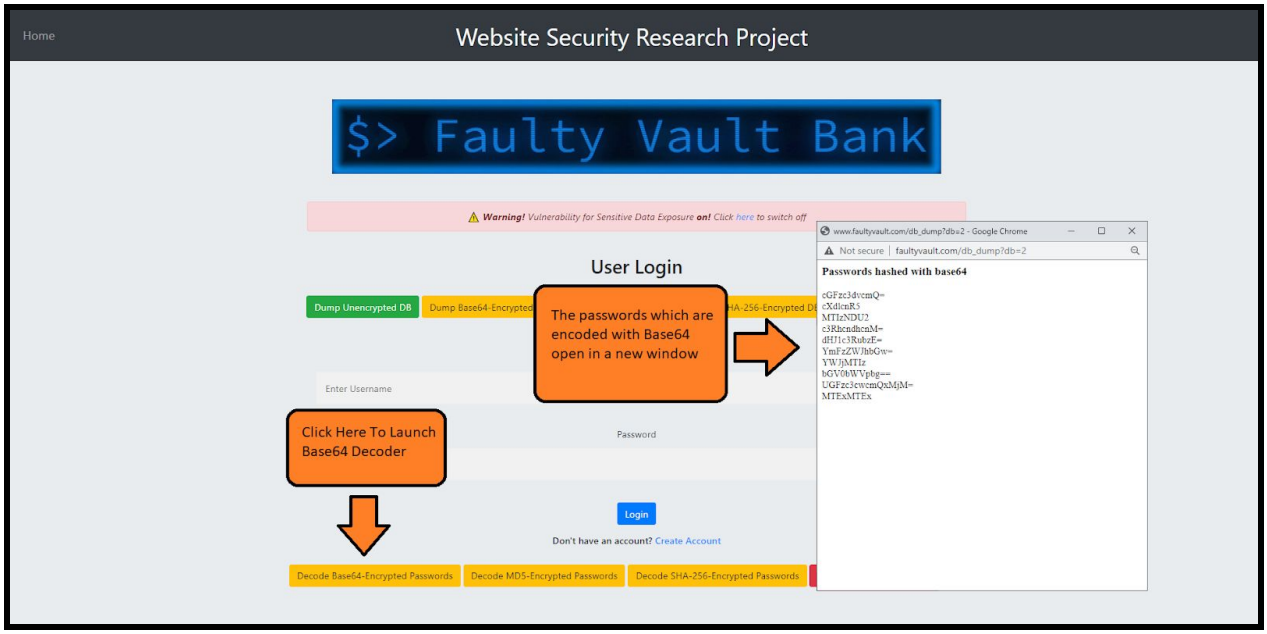
Unencrypted Passwords Stored in the Database

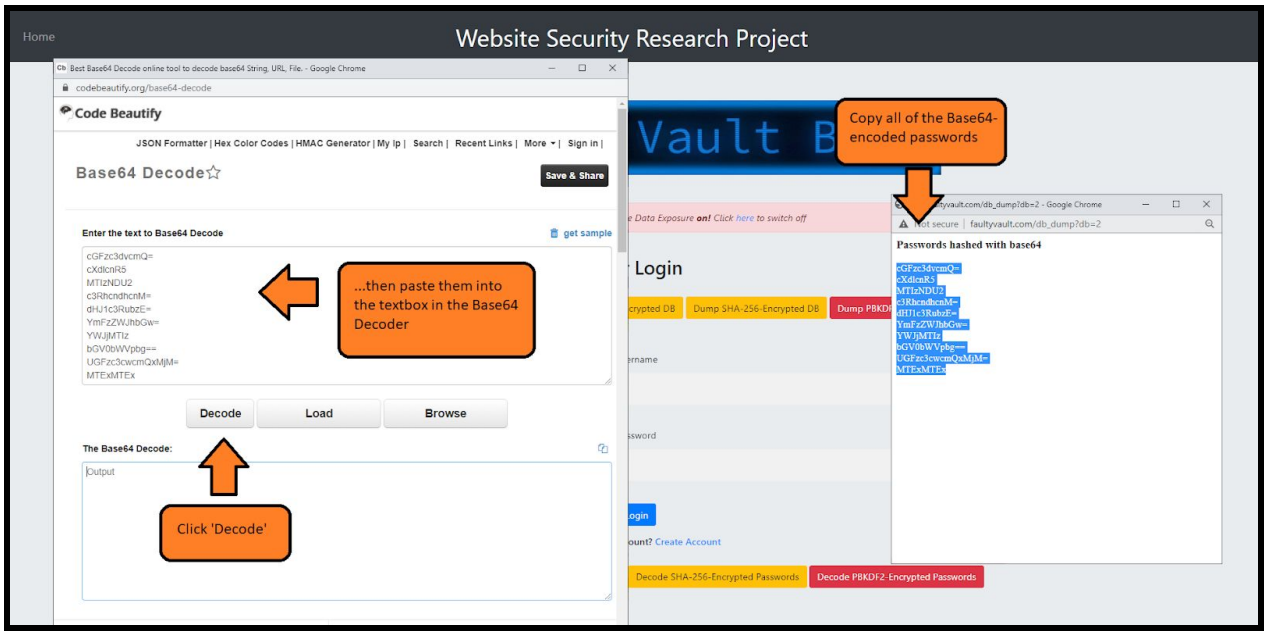


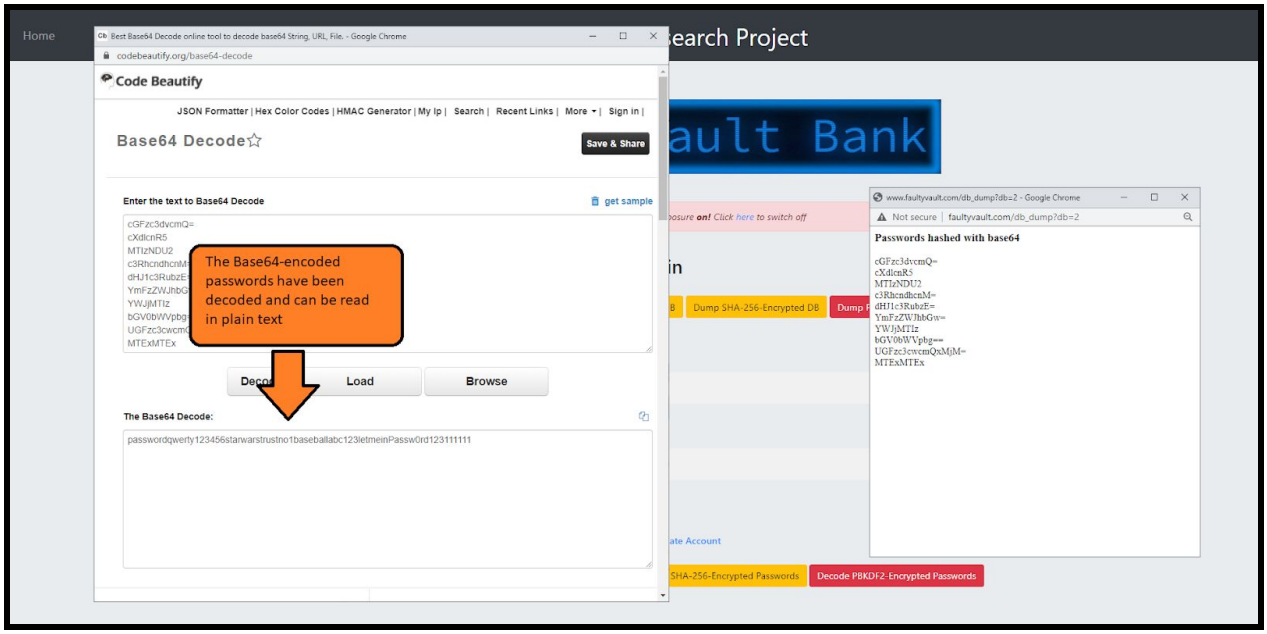


Base-64-Encrypted Passwords Stored in the Database

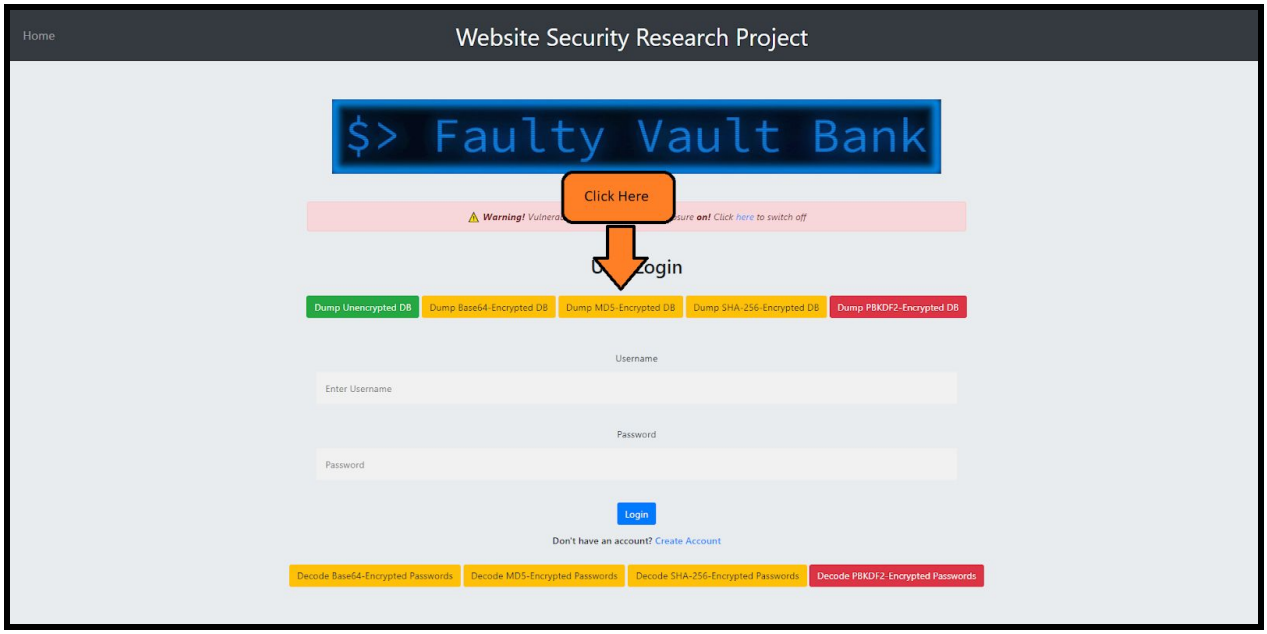


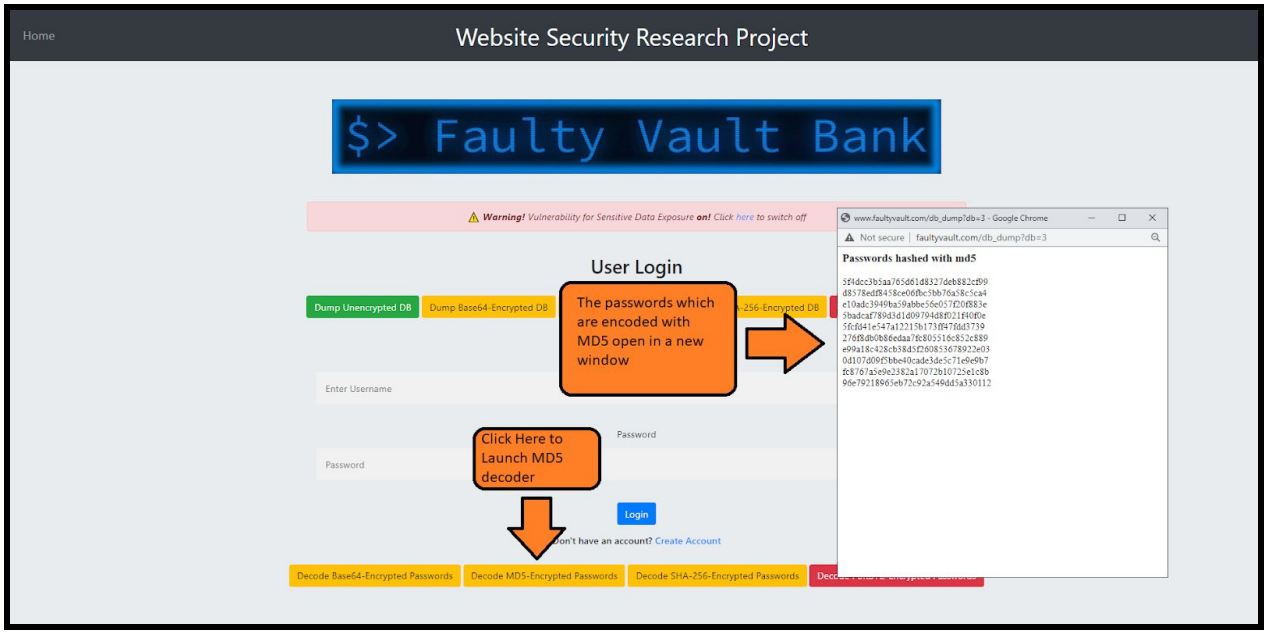


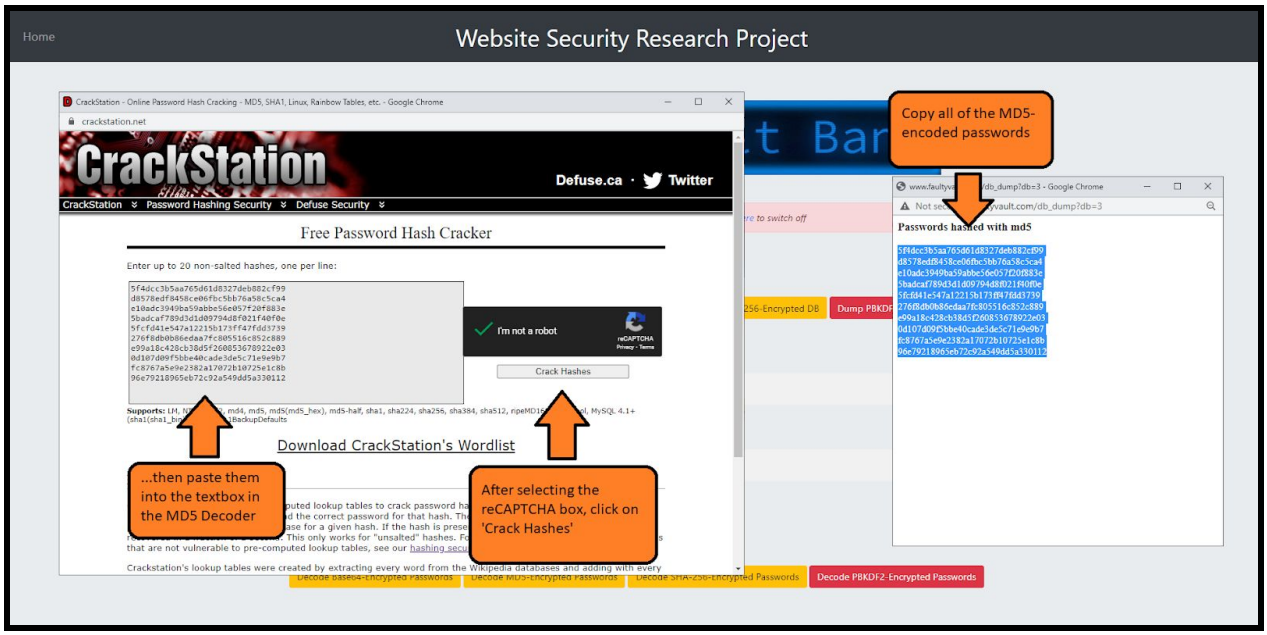

Home
Website Security Research Project
Best Base64 Decode online tool to decode base64 String, URL, File. - Google Chrome
codebeautify.org/base64-decode
Code Beautify
JSON Formatter | Hex Color Codes | HMAC Generator | My Ip | Search | Recent Links | More | Sign in |
Base64 Decode☆
Save & Share
Enter the text to Base64 Decode
get sample
cGFzc3dvcmQ=
cXdlcnR5
MTIzNDU2
c3RhcndhcnM=
dHJ1c3RubzE=
YmFzZWJhbGw=
YWJjMTIz
bGV0bWVpbg==
UGFzc3cwcmQxMjM=
MTExMTEx
...then paste them into the textbox in the Base64 Decoder
Decode
Load
Browse
The Base64 Decode:
Output
Click 'Decode'
Copy all of the Base64-encoded passwords
Vault B
Login
Passwords hashed with base64
Decode SHA-256-Encrypted Passwords
Decode PBKDF2-Encrypted Passwords

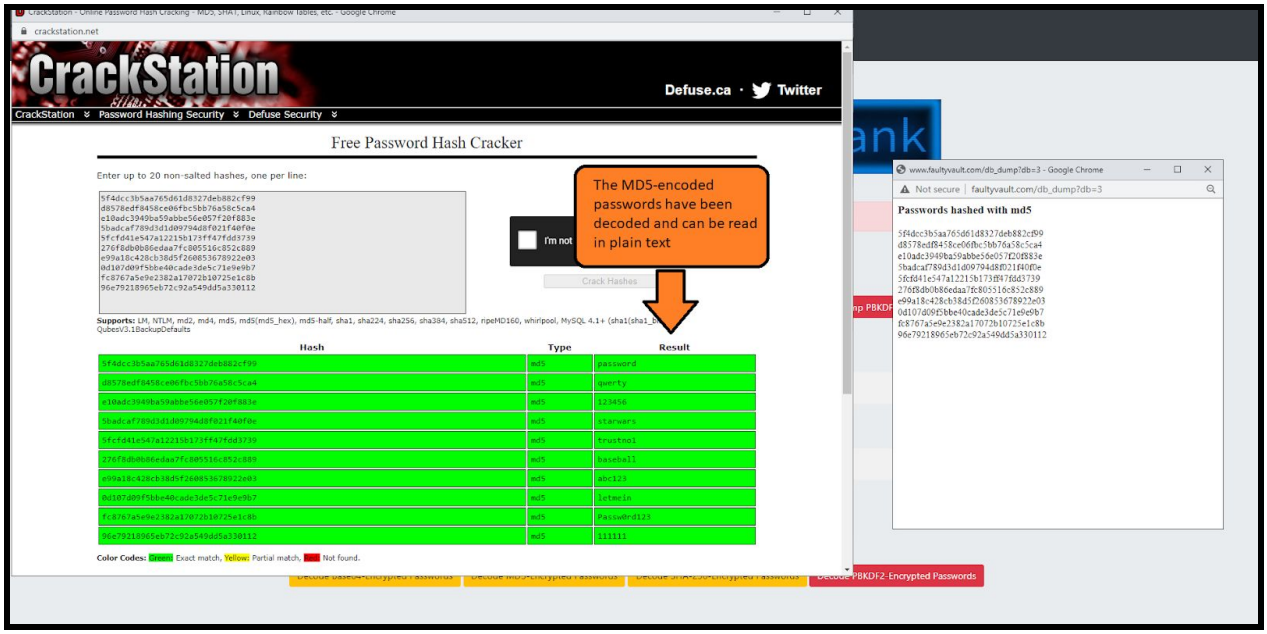

Home
Best Base64 Decode online tool to decode base64 String, URL, File. - Google Chrome
codebeautify.org/base64-decode
Code Beautify
JSON Formatter | Hex Color Codes | HMAC Generator | My Ip | Search | Recent Links | More | Sign in |
Base64 Decode☆
Save & Share
Enter the text to Base64 Decode
get sample
The Base64-encoded passwords have been decoded and can be read in plain text
Load
Browse
The Base64 Decode:
passwordqwerty123456starwarstrustno1baseballabc123letmeinPassw0rd123111111
ault Bank
Passwords hashed with base64
Decode SHA-256-Encrypted Passwords
Decode PBKDF2-Encrypted Passwords

MD5-Encrypted Passwords Stored in the Database

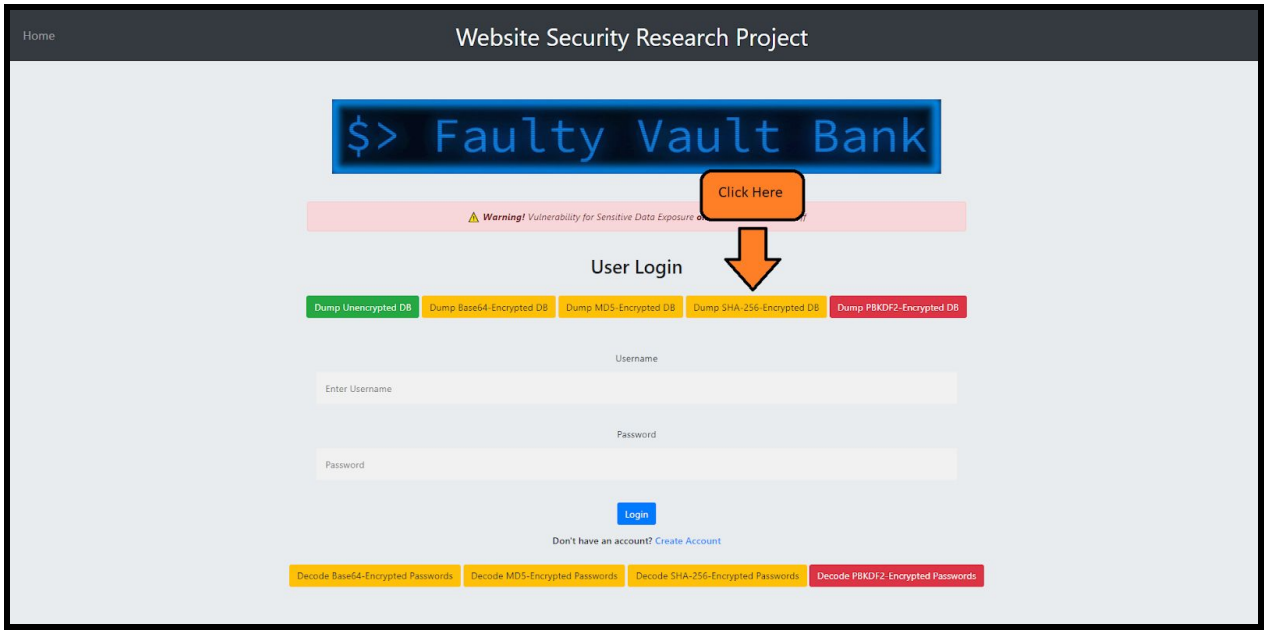


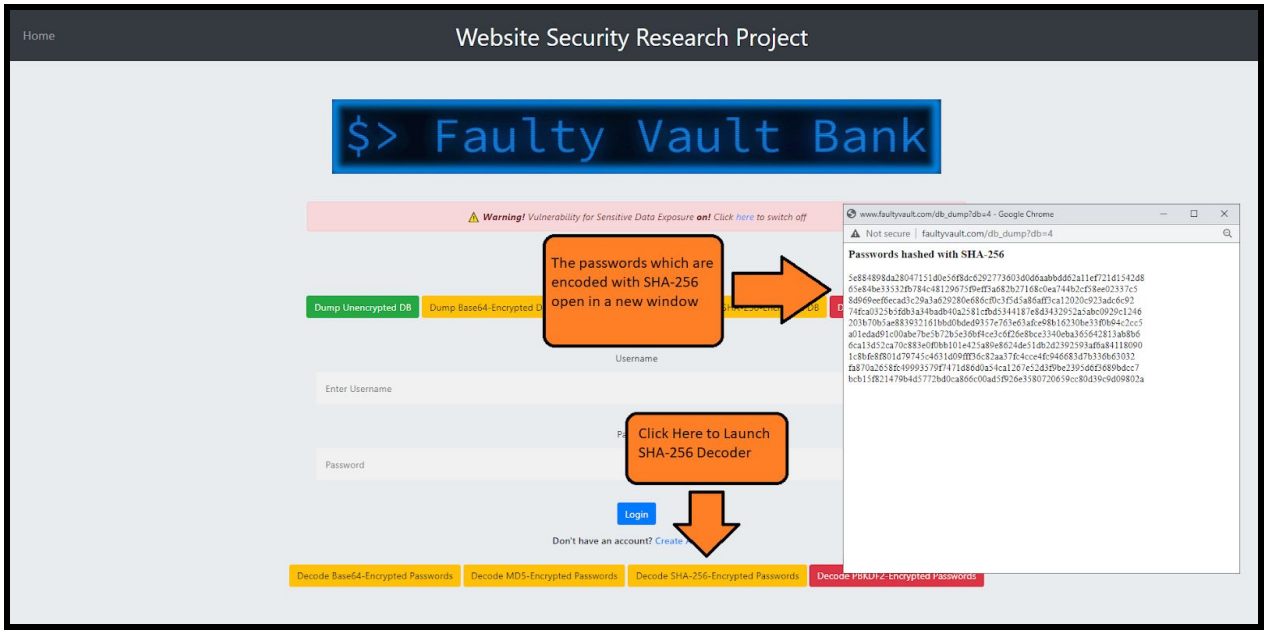


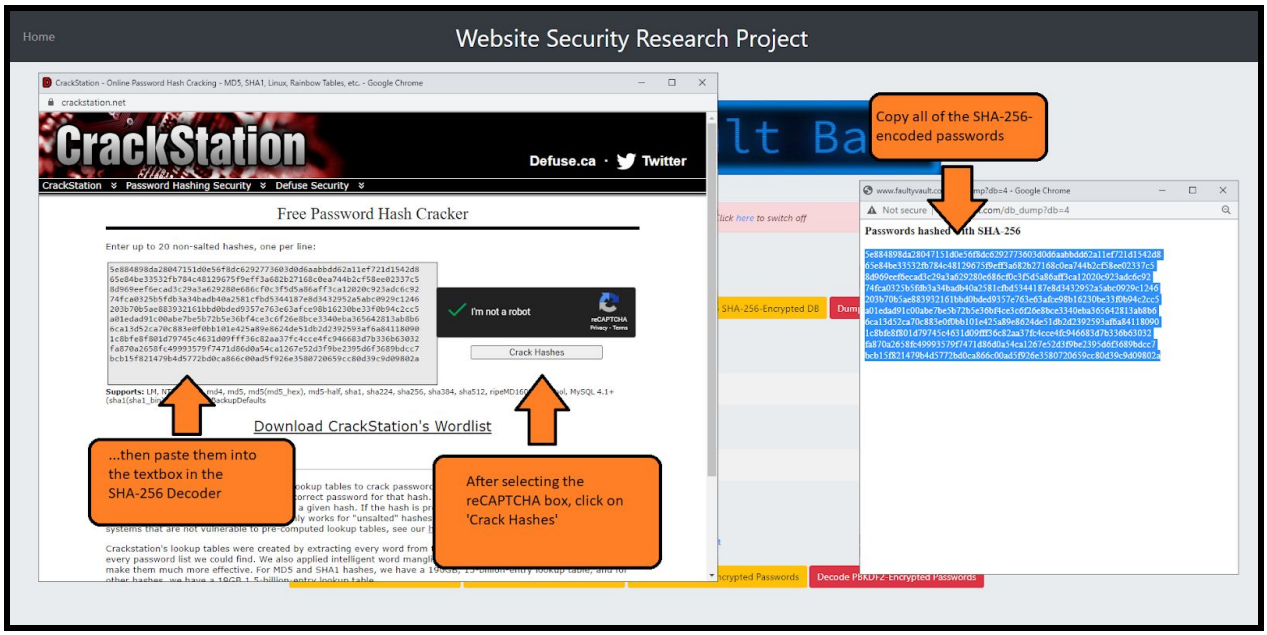

Home
Website Security Research Project
CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc. - Google Chrome
crackstation.net
CrackStation
Defuse.ca · Twitter
CrackStation · Password Hashing Security · Defuse Security
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
5f4dcc3b5aa765d61d8327deb882cf99
d8578edf8458ce06fbc5bb76a58c5ca4
e10adc3949ba59abbe56e057f20f883e
5badcaf789d3d1d09794d8f021f40f0e
5fcfd41e547a12215b173ff47fdd3739
276f8db0b86edaa7fc805516c852c889
e99a18c428cb38d5f260853678922e03
0d107d09f5bbe40cade3de5c71e9e9b7
fc8767a5e9e2382a17072b1072 5e1c8b
96e79218965eb72c92a549dd5a330112
I'm not a robot
reCAPTCHA
Crack Hashes
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults
Download CrackStation's Wordlist
...then paste them into the textbox in the MD5 Decoder
After selecting the reCAPTCHA box, click on 'Crack Hashes'
Copy all of the MD5-encoded passwords
Passwords hashed with md5
Decode PBKDF2-Encrypted Passwords

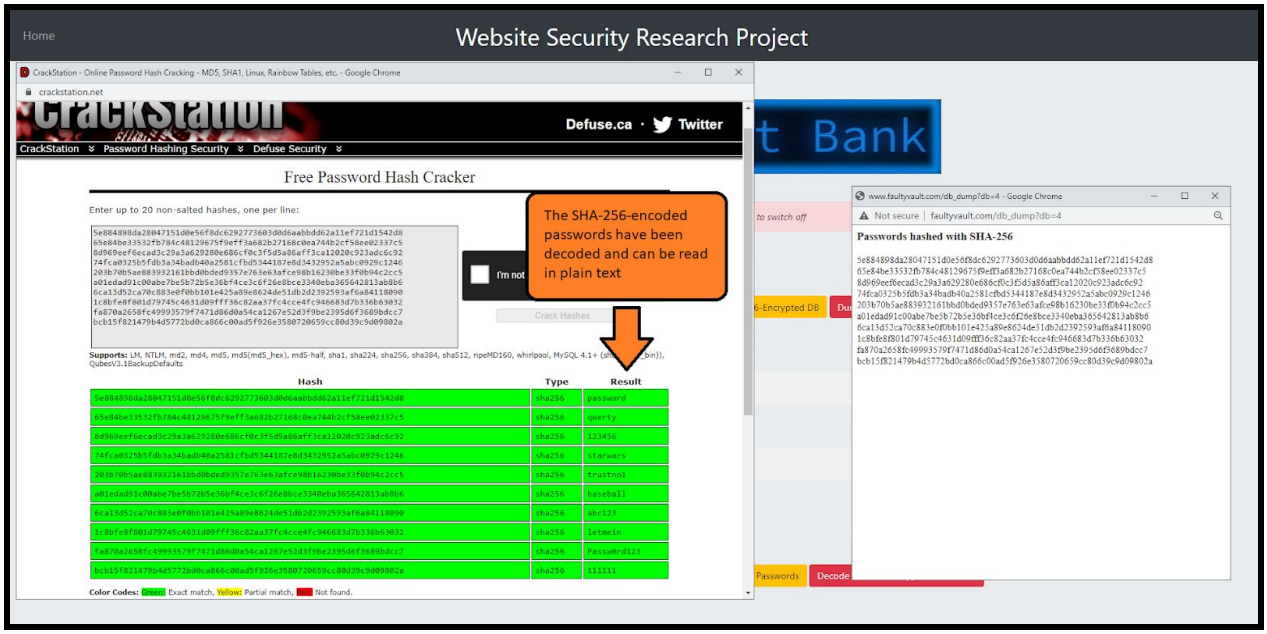

CrackStation
Defuse.ca · Twitter
CrackStation · Password Hashing Security · Defuse Security
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
Crack Hashes
The MD5-encoded passwords have been decoded and can be read in plain text
Hash
Type
Result
5f4dcc3b5aa765d61d8327deb882cf99 md5 password
d8578edf8458ce06fbc5bb76a58c5ca4 md5 qwerty
e10adc3949ba59abbe56e057f20f883e md5 123456
5badcaf789d3d1d09794d8f021f40f0e md5 starwars
5fcfd41e547a12215b173ff47fdd3739 md5 trustno1
276f8db0b86edaa7fc805516c852c889 md5 baseball
e99a18c428cb38d5f260853678922e03 md5 abc123
0d107d09f5bbe40cade3de5c71e9e9b7 md5 letmein
fc8767a5e9e2382a17072b1072 5e1c8b md5 Password123
96e79218965eb72c92a549dd5a330112 md5 111111
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.
Passwords hashed with md5
Decode PBKDF2-Encrypted Passwords

SHA-256-Encrypted Passwords Stored in the Database

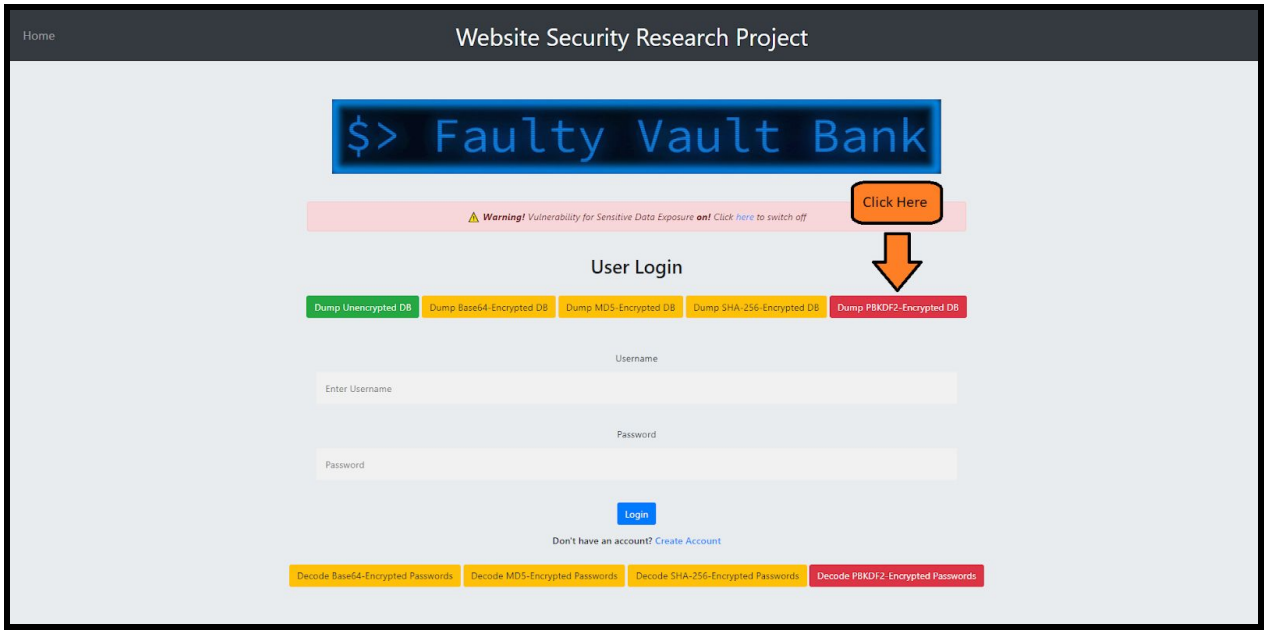


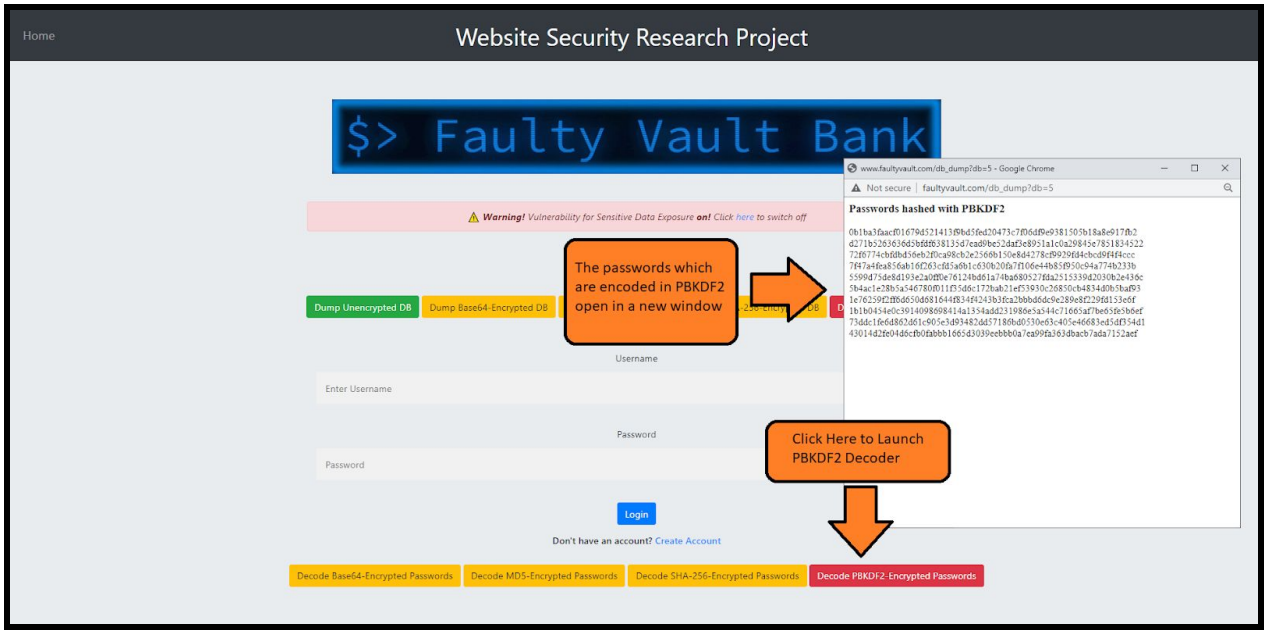


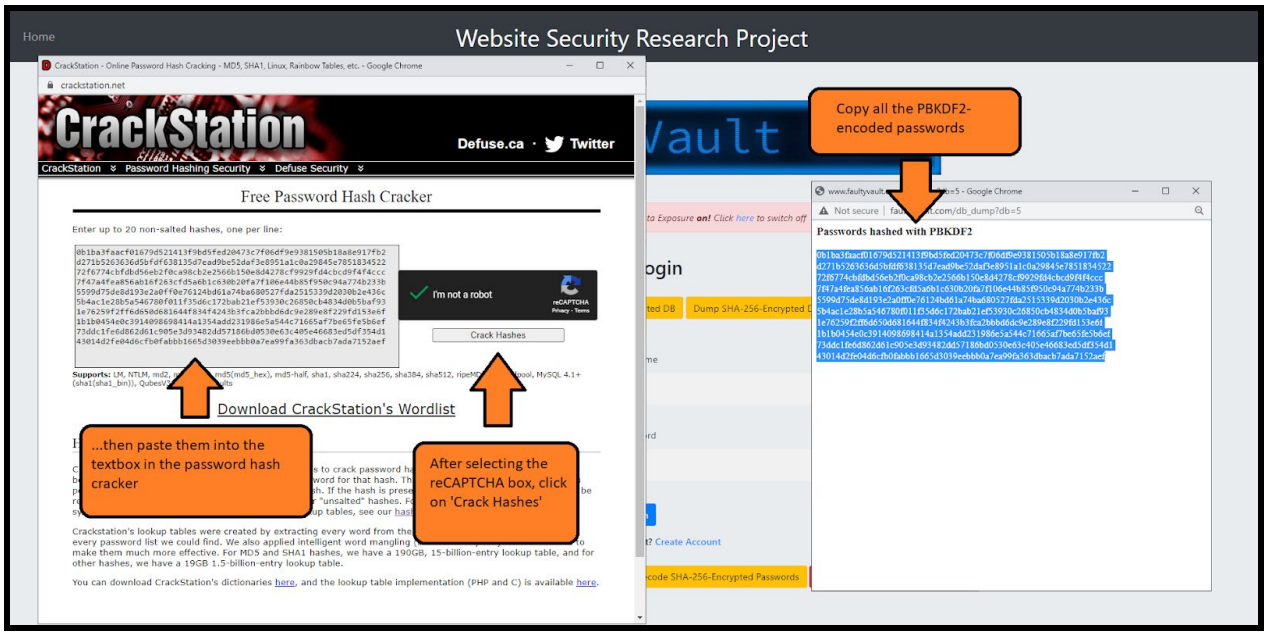

Home
Website Security Research Project
CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc. - Google Chrome
crackstation.net
CrackStation
Defuse.ca · Twitter
CrackStation » Password Hashing Security » Defuse Security »
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
I'm not a robot
reCAPTCHA
Crack Hashes
Download CrackStation's Wordlist
...then paste them into the textbox in the SHA-256 Decoder
After selecting the reCAPTCHA box, click on 'Crack Hashes'
Copy all of the SHA-256-encoded passwords
Passwords hashed with SHA-256

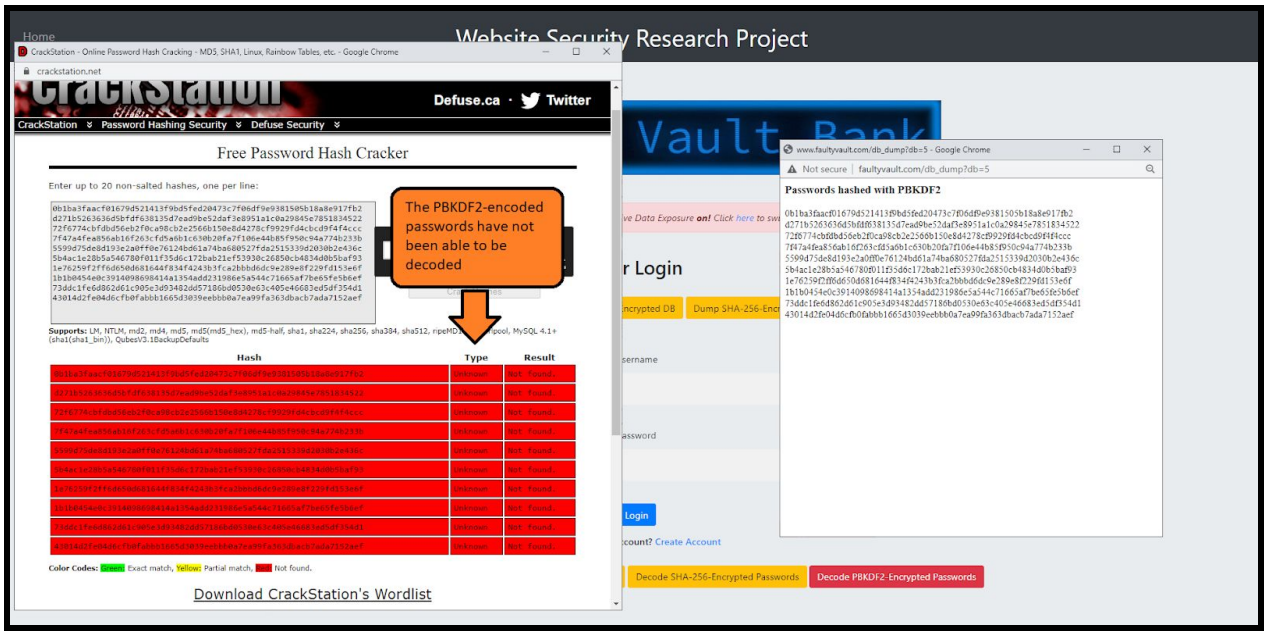

Home
Website Security Research Project
CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc. - Google Chrome
crackstation.net
CrackStation
Defuse.ca · Twitter
CrackStation » Password Hashing Security » Defuse Security »
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
Crack Hashes
The SHA-256-encoded passwords have been decoded and can be read in plain text
Hash
Type
Result
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.
Passwords hashed with SHA-256

PBKDF2-Encrypted Passwords Stored in the Database






Home
Website Security Research Project
CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc. - Google Chrome
crackstation.net
CrackStation
Defuse.ca · Twitter
CrackStation » Password Hashing Security » Defuse Security »
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
I'm not a robot
reCAPTCHA
Crack Hashes
Download CrackStation's Wordlist
...then paste them into the textbox in the password hash cracker
After selecting the reCAPTCHA box, click on 'Crack Hashes'
Copy all the PBKDF2-encoded passwords
Passwords hashed with PBKDF2
Dump SHA-256-Encrypted
Create Account
Decode SHA-256-Encrypted Passwords


Home
Website Security Research Project
CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc. - Google Chrome
crackstation.net
CrackStation
Defuse.ca · Twitter
CrackStation » Password Hashing Security » Defuse Security »
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
The PBKDF2-encoded passwords have not been able to be decoded
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults
Hash
Type
Result
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.
Download CrackStation's Wordlist
Passwords hashed with PBKDF2
Decode SHA-256-Encrypted Passwords
Decode PBKDF2-Encrypted Passwords

Hardening Website

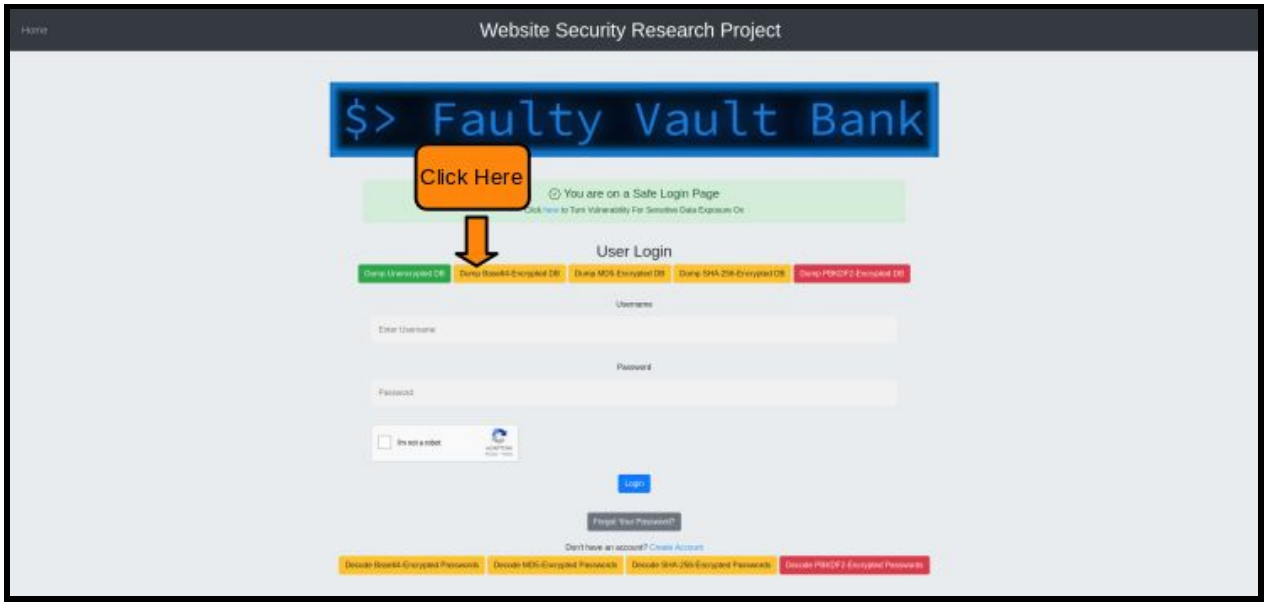
It is important to identify which data in your web application could be classified as sensitive and treat it accordingly, taking into account the relevant privacy laws and regulatory requirements. Sensitive data should not be stored or cached unnecessarily and should be safely discarded by the application or browser as soon as feasibly possible. Furthermore, sensitive data should never be stored or transmitted in plain text. All data in transit should be encrypted in accordance with a secure protocol such as TLS.

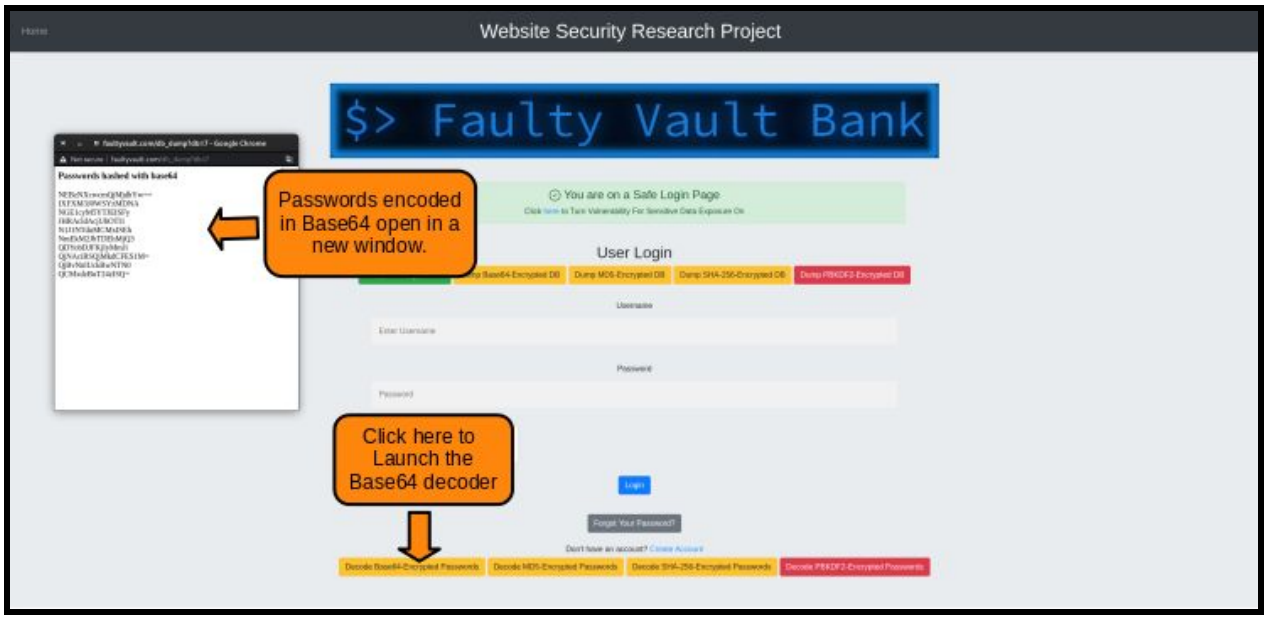
The best strategy of defense against sensitive data exposure is one which is multi-tiered. In addition to the above, implementation of a strong password policy plus encryption of stored passwords with a secure salted hashing function such as PBKDF2 is an effective way to safeguard sensitive data.

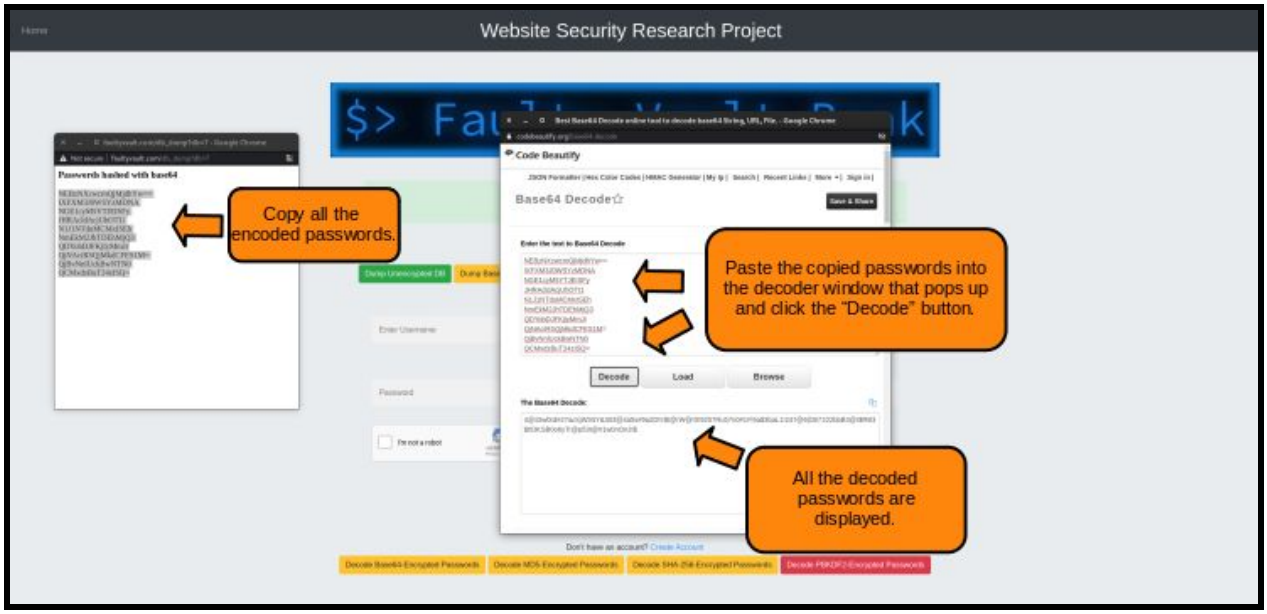
Attempts to Decrypt the Passwords Following Website Hardening

Base-64-Encrypted Passwords Stored in the Database

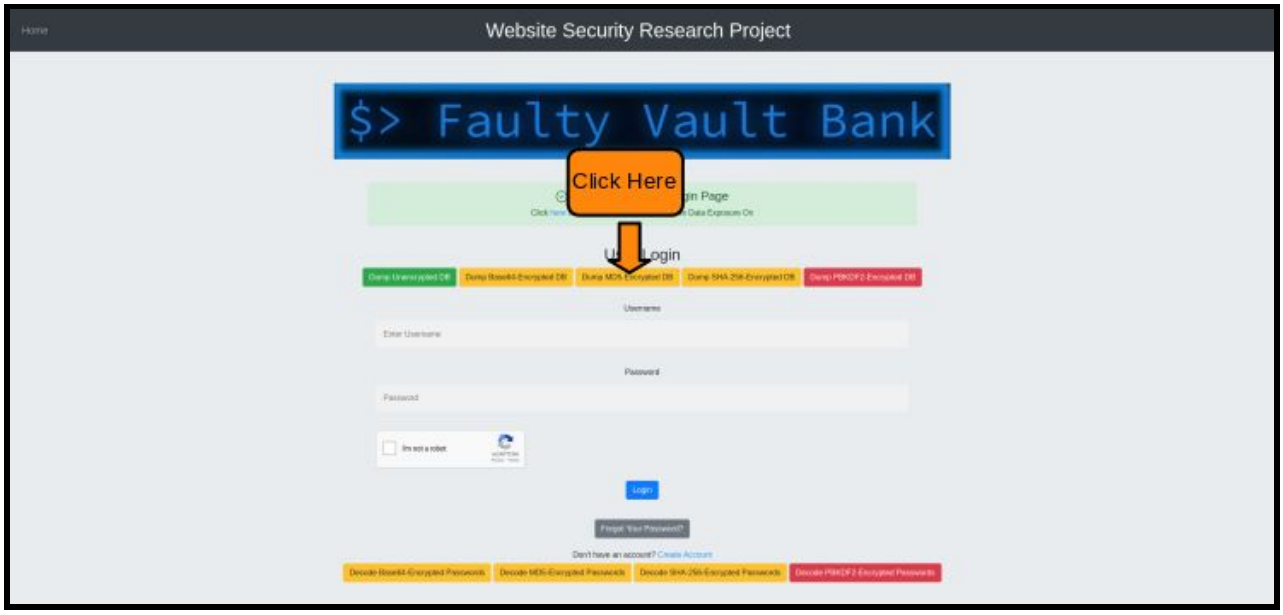
***Base64 is not a hashing algorithm and should never be used to “encrypt” passwords.

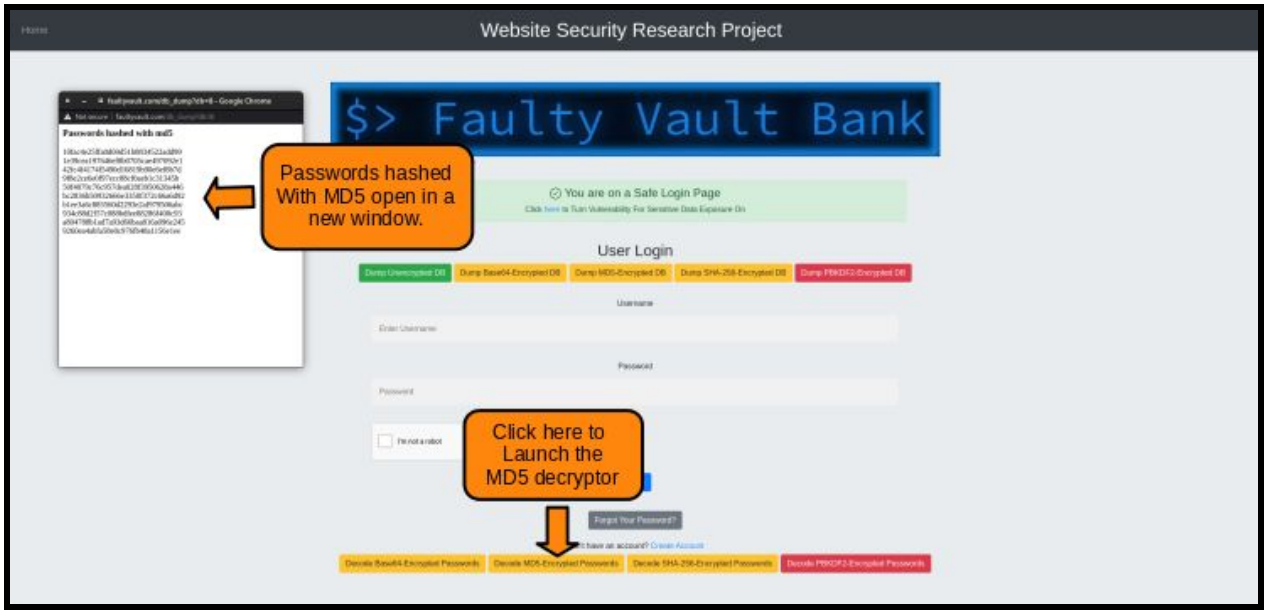



Home
Website Security Research Project
$> Faulty Vault Bank
Passwords encoded in Base64 open in a new window.
You are on a Safe Login Page
User Login
Click here to Launch the Base64 decoder

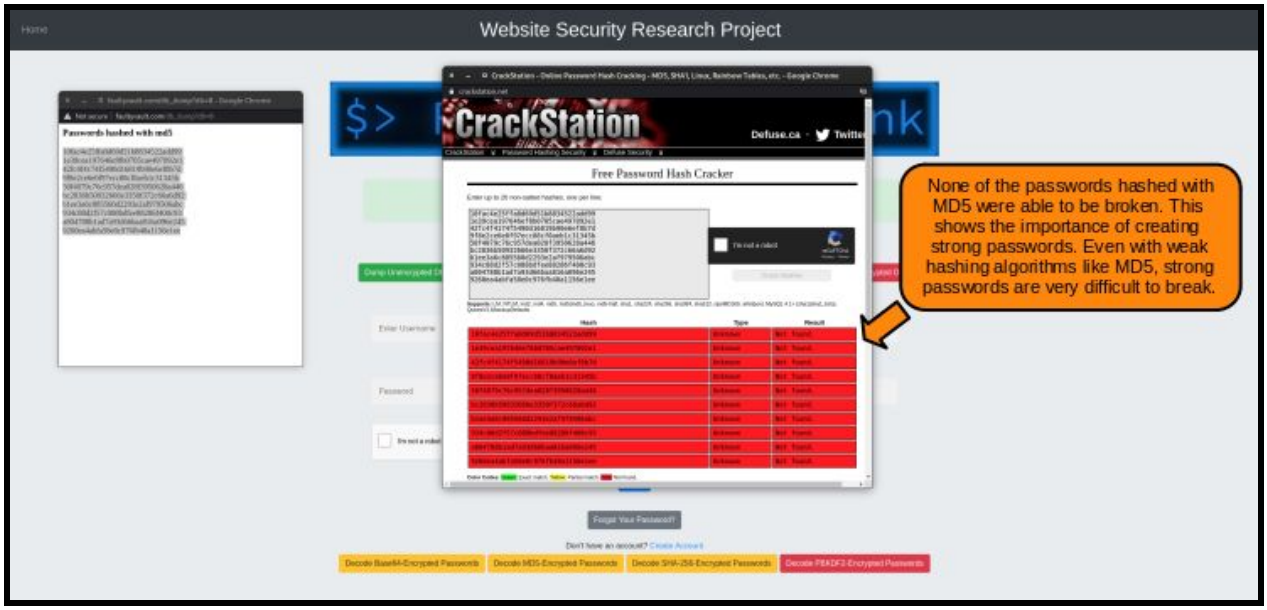

Home
Website Security Research Project
Copy all the encoded passwords.
Code Beautify
Base64 Decoder
Paste the copied passwords into the decoder window that pops up and click the "Decode" button.
Decode
Load
Browse
All the decoded passwords are displayed.

MD5-Encrypted Passwords Stored in the Database

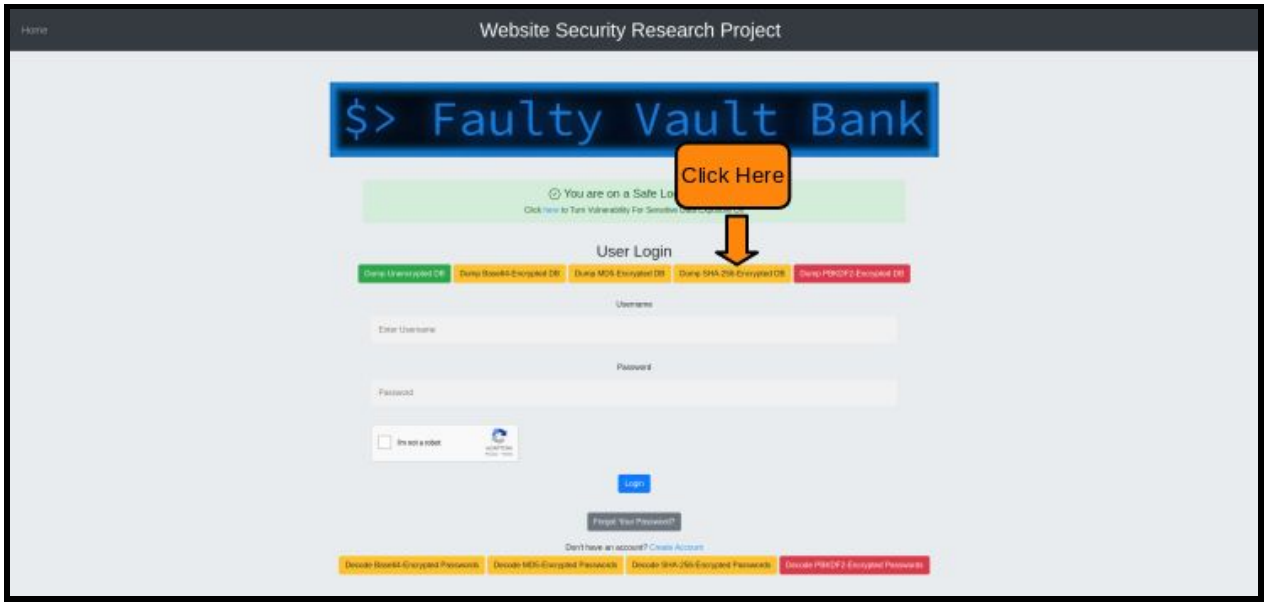


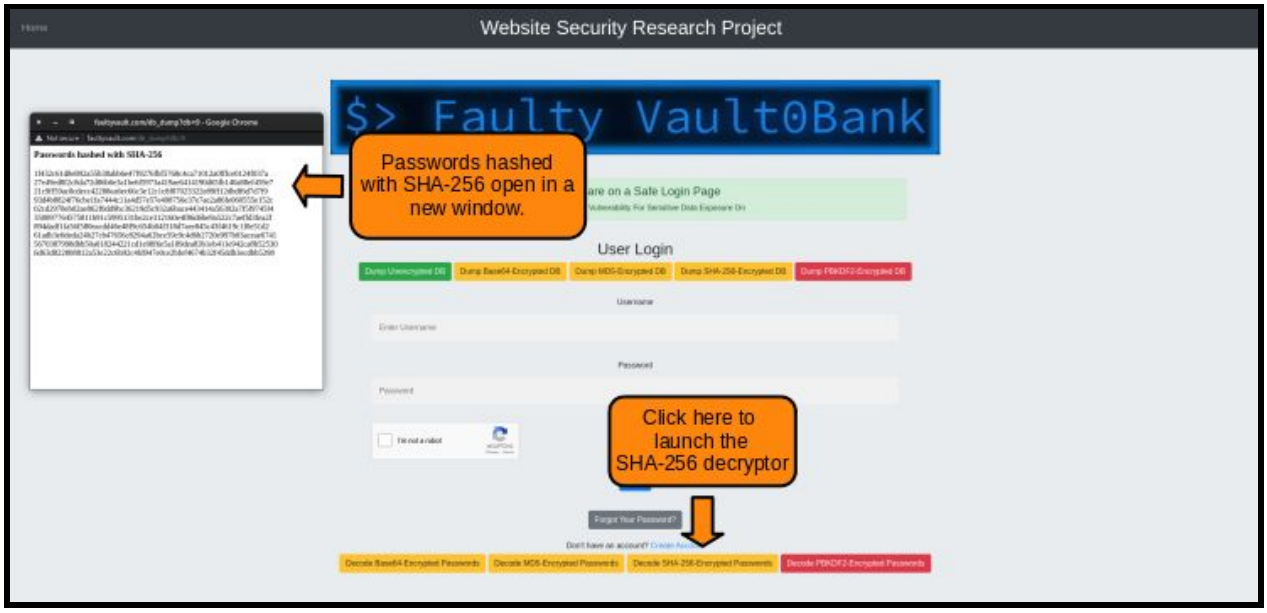



Home
Website Security Research Project
Passwords hashed with md5
Copy all the hashed passwords.
Paste the copied passwords into the decryptor window that pops up. You will need to verify you are not a robot and then click "Crack Hashes"
CrackStation
Free Password Hash Cracker
Download CrackStation's Wordlist
How CrackStation Works

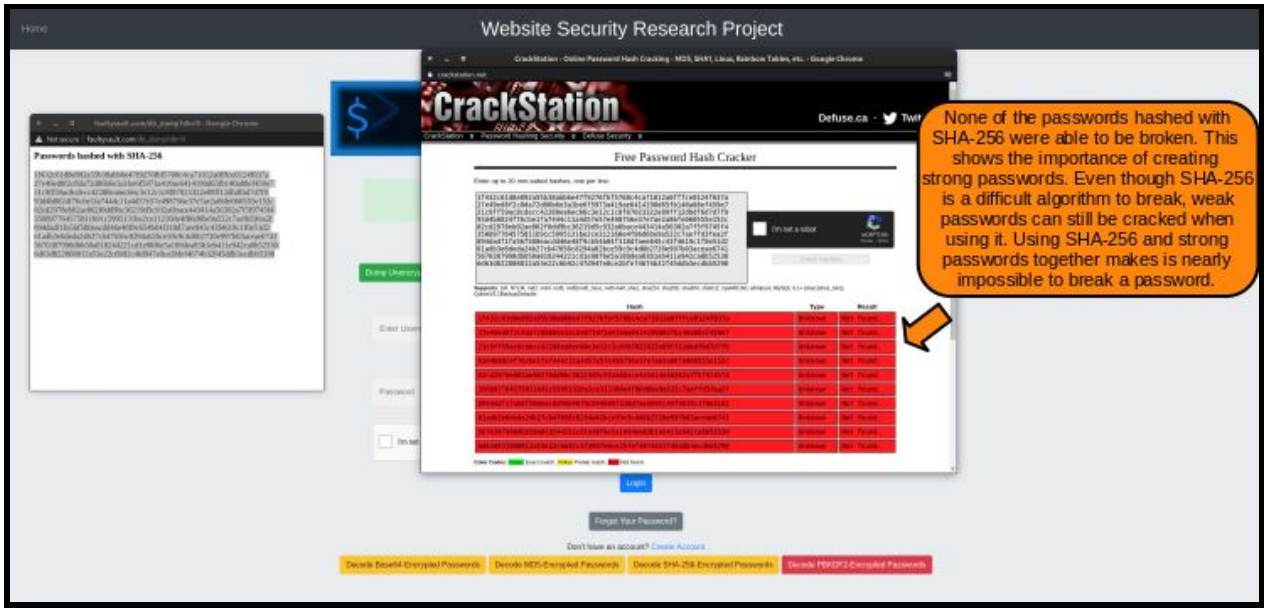

Home
Website Security Research Project
Passwords hashed with md5
CrackStation
Free Password Hash Cracker
None of the passwords hashed with MD5 were able to be broken. This shows the importance of creating strong passwords. Even with weak hashing algorithms like MD5, strong passwords are very difficult to break.

SHA-256-Encrypted Passwords Stored in the Database






Website Security Research Project
Copy all the hashed passwords.
Paste the copied passwords into the decryptor window that pops up. You will need to verify you are not a robot and then click "Crack Hashes"
Passwords hashed with SHA-256
CrackStation
Free Password Hash Cracker
Download CrackStation's Wordlist
How CrackStation Works


Website Security Research Project
Passwords hashed with SHA-256
CrackStation
Free Password Hash Cracker
None of the passwords hashed with SHA-256 were able to be broken. This shows the importance of creating strong passwords. Even though SHA-256 is a difficult algorithm to break, weak passwords can still be cracked when using it. Using SHA-256 and strong passwords together makes is nearly impossible to break a password.

Resources

https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

https://docs.python.org/3/library/hashlib.html

https://searchsecurity.techtarget.com/definition/MD5

https://en.wikipedia.org/wiki/MD5#Overview_of_security_issues

https://www.geeksforgeeks.org/md5-hash-python/

https://www.geeksforgeeks.org/sha-in-python/

https://en.wikipedia.org/wiki/SHA-2

https://www.solarwindsmsp.com/blog/sha-256-encryption

https://crackstation.net/

https://crackstation.net/hashing-security.htm

https://www.safetydetectives.com/blog/the-most-hacked-passwords-in-the-world/

https://stackabuse.com/encoding-and-decoding-base64-strings-in-python/

https://codebeautify.org/base64-decode

https://en.wikipedia.org/wiki/Base64

https://stackoverflow.com/questions/28836837/is-base64-an-encryption-or-encoding-algorithm#:~:text=it%20is%20not%20considered%20as,content%2C%20so%20it's%20not%20encryption.&text=Base64%20is%20such%20an%20encoding,may%20not%20be%20handled%20correctly.

https://en.wikipedia.org/wiki/PBKDF2

https://security.stackexchange.com/questions/16354/whats-the-advantage-of-using-pbkdf2-vs-sha256-to-generate-an-aes-encryption-key

https://ropesec.com/articles/sensitive-data-exposure/

https://en.wikipedia.org/wiki/Rainbow_table