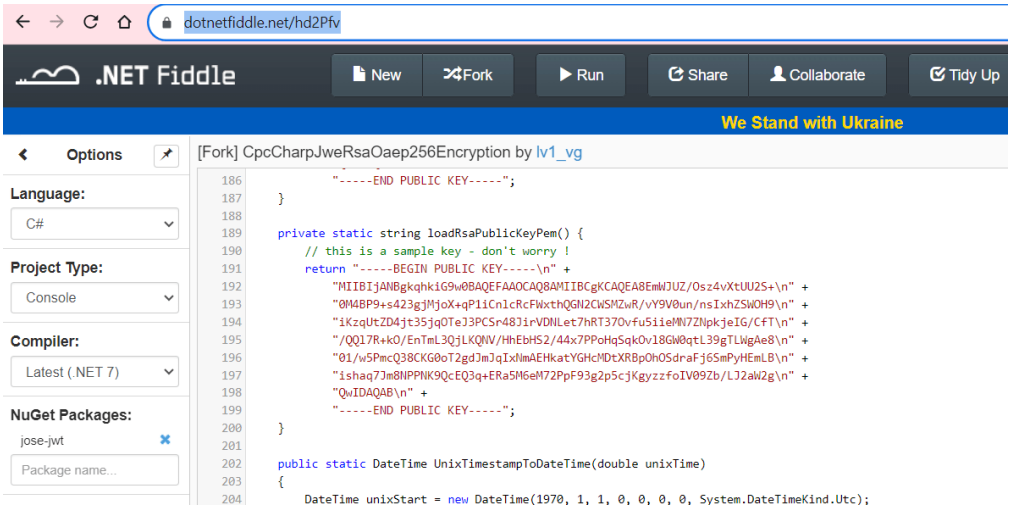
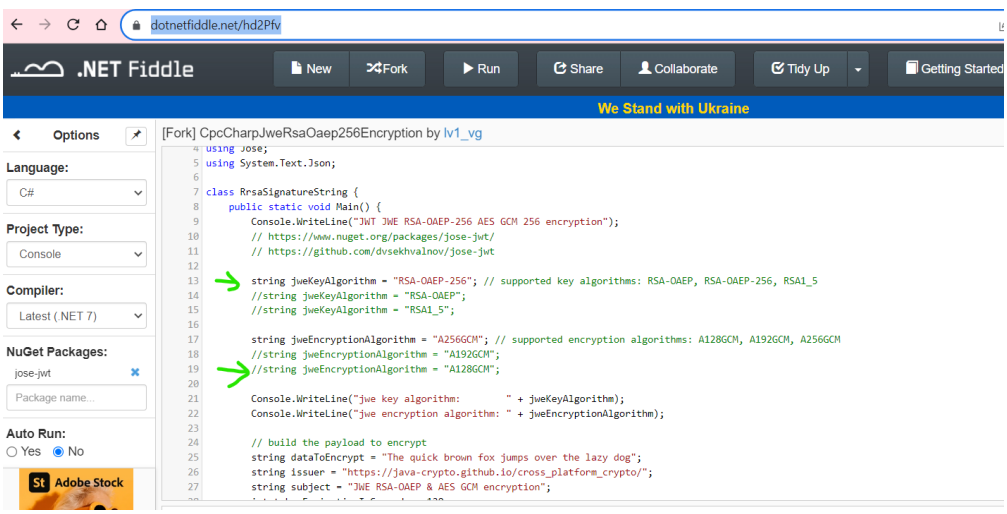
Top 2: Hands On

For testing Cryptographic Weakness you have to manually review the code.

Point 1: Browse Dotnet Fiddle site. https://dotnetfiddle.net/hd2Pfv and check the crypto key length(strong length minimum 256 bytes /2048 bits), like public key or private key length should not short.



Point 2: Verify Strong Cryptography

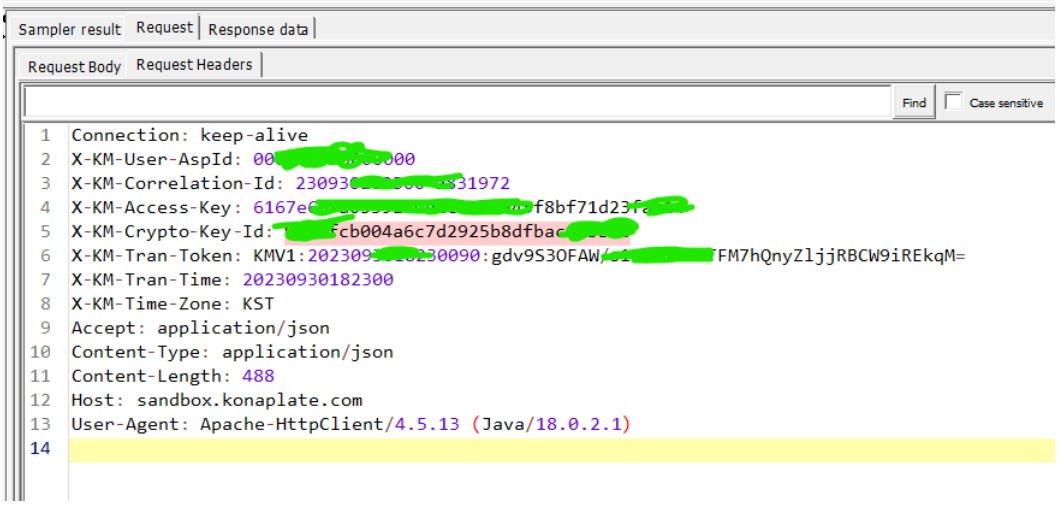


List of Weak Cryptography:

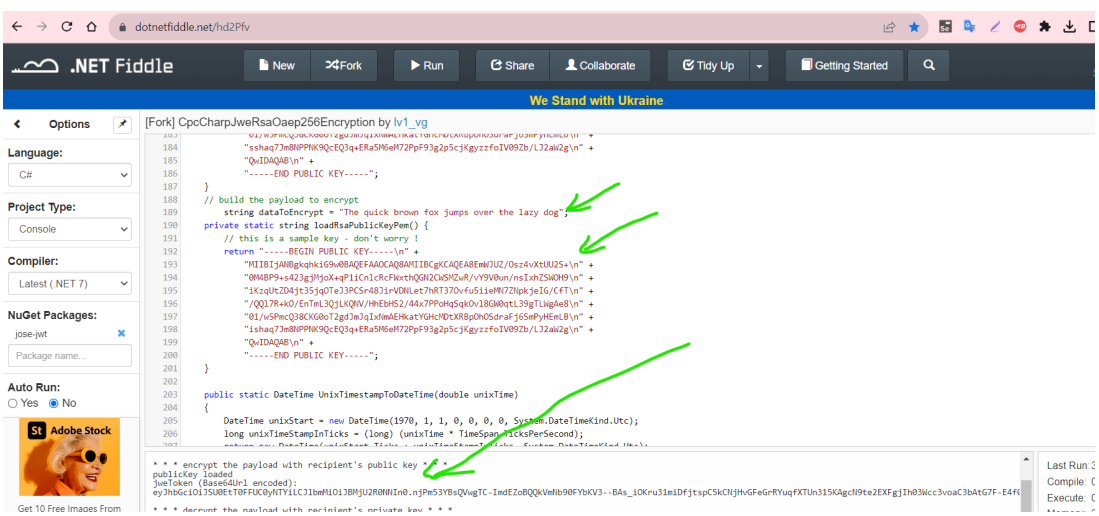
1. Data Encryption Standard (DES): DES was once a widely used symmetric encryption algorithm, but it is now considered weak due to its small key size (56 bits). It is susceptible to brute force attacks, and its use is strongly discouraged.
2. Triple Data Encryption Algorithm (3DES): While 3DES is an improvement over DES, it is considered outdated as well. It uses the same DES algorithm three times with different keys but has vulnerabilities compared to modern encryption algorithms like AES.
3. Rivest Cipher (RC4): RC4 was once used for SSL/TLS encryption in web browsers and other applications, but multiple vulnerabilities have been discovered in its implementation. It is now considered insecure, and its use is discouraged.
4. MD5 (Message Digest 5): MD5 is a hash function that has known collision vulnerabilities. It is no longer considered secure for cryptographic purposes like digital signatures.
5. SHA-1 (Secure Hash Algorithm 1): SHA-1 is another hash function that has known collision vulnerabilities. It is being phased out in favor of more secure hash functions like SHA-256.
6. WEP (Wired Equivalent Privacy): WEP was an early security protocol for wireless networks, but it has well-documented weaknesses. It is easily cracked, and its use is strongly discouraged.

7. SSL and Early TLS Versions: Older versions of the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols, such as SSLv2, SSLv3, and early TLS 1.0, have known vulnerabilities. It is recommended to use modern TLS versions (e.g., TLS 1.2 or TLS 1.3) for secure communication.
8. RSA Key Sizes Below 2048 Bits: As computing power has increased, RSA key sizes below 2048 bits are no longer considered sufficiently secure for long-term protection of data. Key sizes of 2048 bits or higher are recommended.
9. Diffie-Hellman Key Exchange with Small Parameters: Using small parameters in the Diffie-Hellman key exchange can make it vulnerable to attacks. Stronger and larger parameter sizes should be used to enhance security.
10. Insecure Password Hashing Algorithms: Hashing algorithms like MD5 and SHA-1 should not be used for securely hashing passwords. Instead, stronger and slower hash functions like bcrypt or Argon2 should be used.

Point 3: Verify **X-Type option** must in the request headers which should be encrypted



Point 4: Ensure **Initializing Vector (IV)** is used in that way when different cyphertext will be generated though public key and plaintext is same.



Point 5: Analyzing JWT Session

request e jodhi csrf token thake tahole issue na

Md. Farhan Islam 9/4 5:30 PM

Request e Access Token Ache bhaia

CSRF token name e kichu nai

GET /api/assets/details/%27+--+&password= HTTP/2
Host: kiwasystem.com
Cookie: _ga=GA1.1.1595788934.1693822789;
IP_PORTAL_ACCESS_TOKEN=eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJsZWRAeW9wbWFpbC5jb20iLCJjbGllbnRJZ
CI6IjAzMmFjMzExLTk3MzktNGVkMS05MjdiLThiNTZjZDQyNzA1NyIsImNyZWF0ZWQiOjE2OTM4MjY2MTc4
MTgsInJhbmRvbVZhbHVlIjoiMTY5MzgyNjYxNzgxOCIsInJvbGVMaXN0IjoiNDI2NGNiYTgtZjRhZS00YTk5LTg
wZGMtODgyZTRiYzEzMjlhOmFsbC1hY2Nlc3MiLCJ0eXBlIjoiQUNDRVNTX1RPS0VOIiwidXNlcklkIjoiODE3MjE
0NzAtYjQyYi00Y2M4LTkwZjEtMzJjYmRiOWVhZjRiIiwiYXNwSWQiOiIxMDAwMTIzNDU2MTIzNDUiLCJwaG9u
ZSI6bnVsbCwiZGVzaWduYXRpb24iOm51bGwsInVzZXJUeXBlIjoiQ19TVSIsImV4cCI6MTY5MzgyNjkxNywiZ
W1haWwiOm51bGx9.DXlpht8drb6BdcpLKqBqPUl3zSiN5RCaJj9qjlfe2eoeFCxXqYCoUIoDR7ra-v-n1mUk4F9-
vu78Qmrk2pWXpA;

jwt.io

JWT

Debugger Libraries Introduction Ask

Crafted by auth0 by Okta

Encoded PASTE A TOKEN HERE

eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJsZWRAeW
9wbWFpbC5jb20iLCJjbGllbnRJZCI6IjAzMmFjM
zExLTk3MzktNGVkMS05MjdiLThiNTZjZDQyNzA1
NyIsImNyZWF0ZWQiOjE2OTM4MjY2MTc4MTgsInJ
hbmRvbVZhbHVlIjoiMTY5MzgyNjYxNzgxOCIsIn
JvbGVMaXN0IjoiNDI2NGNiYTgtZjRhZS00YTk5L
TgwZGMtODgyZTRiYzEzMjlhOmFsbC1hY2Nlc3Mi
LCJ0eXBlIjoiQUNDRVNTX1RPS0VOIiwidXNlckl
kIjoiODE3MjE0NzAtYjQyYi00Y2M4LTkwZjEtMz
JjYmRiOWVhZjRiIiwiYXNwSWQiOiIxMDAwMTIzN
DU2MTIzNDUiLCJwaG9uZSI6bnVsbCwiZGVzaWdu
YXRpb24iOm51bGwsInVzZXJUeXBlIjoiQ19TVSI
sImV4cCI6MTY5MzgyNjkxNywiZW1haWwiOm51bG
x9.DXlpht8drb6BdcpLKqBqPUl3zSiN5RCaJj9q
jIfe2eoeFCxXqYCoUIoDR7ra-v-n1mUk4F9-
vu78Qmrk2pWXpA

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

{
"alg": "HS512"
}

PAYLOAD: DATA

"clientId": "032ac311[redacted]0cd427057",
"created": 1693826617818,
"randomValue": "1693826617818",
"roleList": "4264cba8[redacted]02e4bc1329a:all-access",
"type": "ACCESS_TOKEN",
"userId": "8172147[redacted]cbdb9eaf4b",
"aspId": "10[redacted]2345",
"phone": null,
"designation": null,
"userType": "C_SU",
"exp": 1693826917,
"email": null
}

Note: A CSRF token can protect CSRF attack though session hijak.