

Sensitive Data Exposure

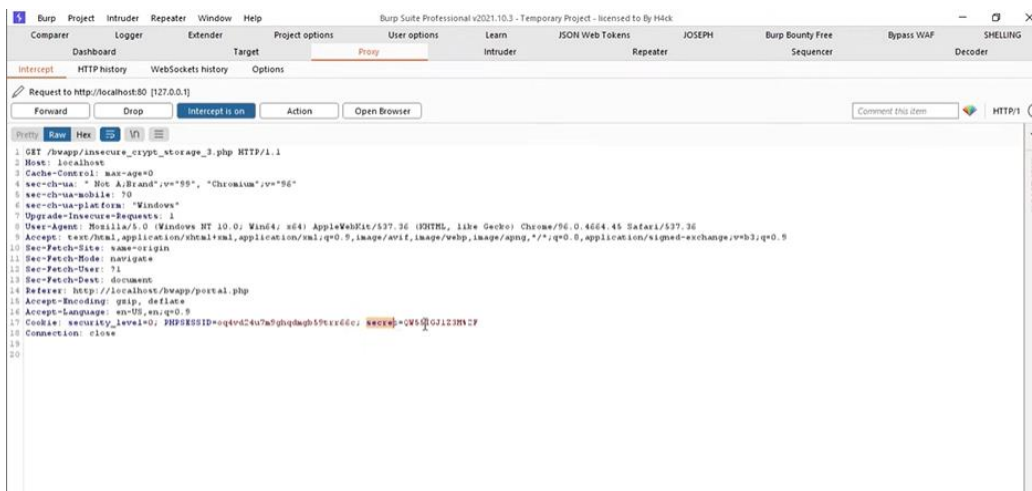
base64 encoding – POC

Step 1: Open BWAPP and select Base64 Encoding and click on hack

Security Level : Low



Step 2 : Capture the request in BurpSuite , here we’ve got a cookie-header and secret named parameter



The hint was to decrypt the parameter

Step 3: Then using decoder feature in burpsuite ‘smartdecoder’, identify the type of encoding in this case it’s base 64



Step 4: After selecting Base 64 encoding , it will decode

Here after decoding the message was “Any bugs?”

QWSSIGHZ3M

Any bugs?

Text

Hex

Decode as ...

Encode as ...

Hash ...

Smart decode

Text

Hex

Decode as ...

Encode as ...

Hash ...

Smart decode

Here we exposed sensitive encrypted data by decoding it

Security Level: Medium/High

Step 1: After capturing requests in medium level, this time we got some bigger encoded value

```
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/bwapp/insecure_crypt_storage_3.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=oq4vd24u7m9ghqdmgb59trr66c; security_level=1; secret=83785efbbef1b4cdf3260c5e6505f7d2261f738d
Connection: close
```

Step 2: Use Hash analyserf to find out the type of hash

Tool to identify hash types. Enter a hash to be identified.

83785efbbef1b4cdf3260c5e6505f7d2261f738d

Analyze

Hash:	83785efbbef1b4cdf3260c5e6505f7d2261f738d
Salt:	Not Found
Hash type:	SHA1(or SHA 128)
Bit length:	160
Character length:	40
Character type:	hexidecimal

In this case it is SHA1 with bit length – 160

Step 3: After finding out the type of hash just decrypt the hash

Decrypt Hash Results for: 83785efbbef1b4cdf3260c5e6505f7d2261f738d

Algorithm	Hash	Decrypted
sha1	83785efbbef1b4cdf3260c5e6505f7d2261f738d	Any bugs?

Here we successfully decrypted sensitive data