Lucy Njuguna

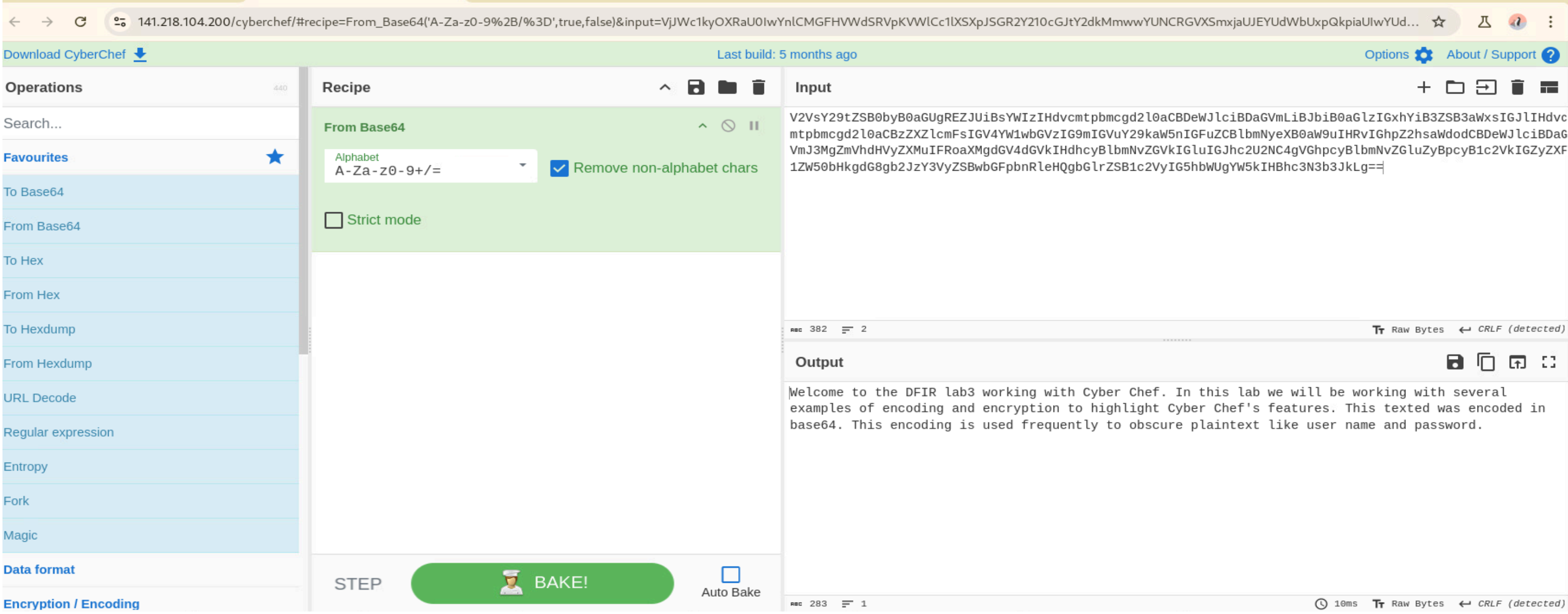
Project: CyberChef & OSINT Analysis

Date: 10/12/2024

1. Base64 Decoding – Welcome Message

Decoded a Base64-encoded welcome message using CyberChef's "From Base64" function. Highlighted how Base64 is used to obscure plaintext data such as usernames and passwords.

1.1 SCREENSHOT : (Base64)



141.218.104.200/cyberchef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true,false)&input=VjJWc1kyOXRaU0IwYnlCMGFHVVdSRVpKVWlCc1lXSXpJSGR2Y210cGJtY2dkMmwwYUNCRGVXSmxjaUJEYUdWbUxpQkpiaUIwYUd…

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

□ Strict mode

```
V2VsY29tZSB0byB0aGUgREZJUiBsYWIzIHdvcmtpbmcgd2l0aCBDeWJlciBDaGVmLiBJbiB0aGlzIGxhYiB3ZSB3aWxsIGJlIHdvc
mtpbmcgd2l0aCBzZXZlcmFsIGV4YW1wbGVzIG9mIGVuY29kaW5nIGFuZCBlbmNyeXB0aW9uIHRvIGhpZ2hsaWdodCBDeWJlciBDaG
VmJ3MgZmVhdHVyZXMuIFRoaXMgdGV4dGVkIHdhcyBlbmNvZGVkIGluIGJhc2U2NC4gVGhpcyBlbmNvZGluZyBpcyB1c2VkIGZyZXF
1ZW50bHkgdG8gb2JzY3VyZSBwbGFpbnRleHQgbGlrZSB1c2VyIG5hbWUgYW5kIHBhc3N3b3JkLg==
```

382 2 Raw Bytes CRLF (detected)

```
Welcome to the DFIR lab3 working with Cyber Chef. In this lab we will be working with several
examples of encoding and encryption to highlight Cyber Chef's features. This texted was encoded in
base64. This encoding is used frequently to obscure plaintext like user name and password.
```

283 1 10ms Raw Bytes CRLF (detected)

Resulting Output: The screenshot shows a basic example of using CyberChef to decode Base64-encoded data.

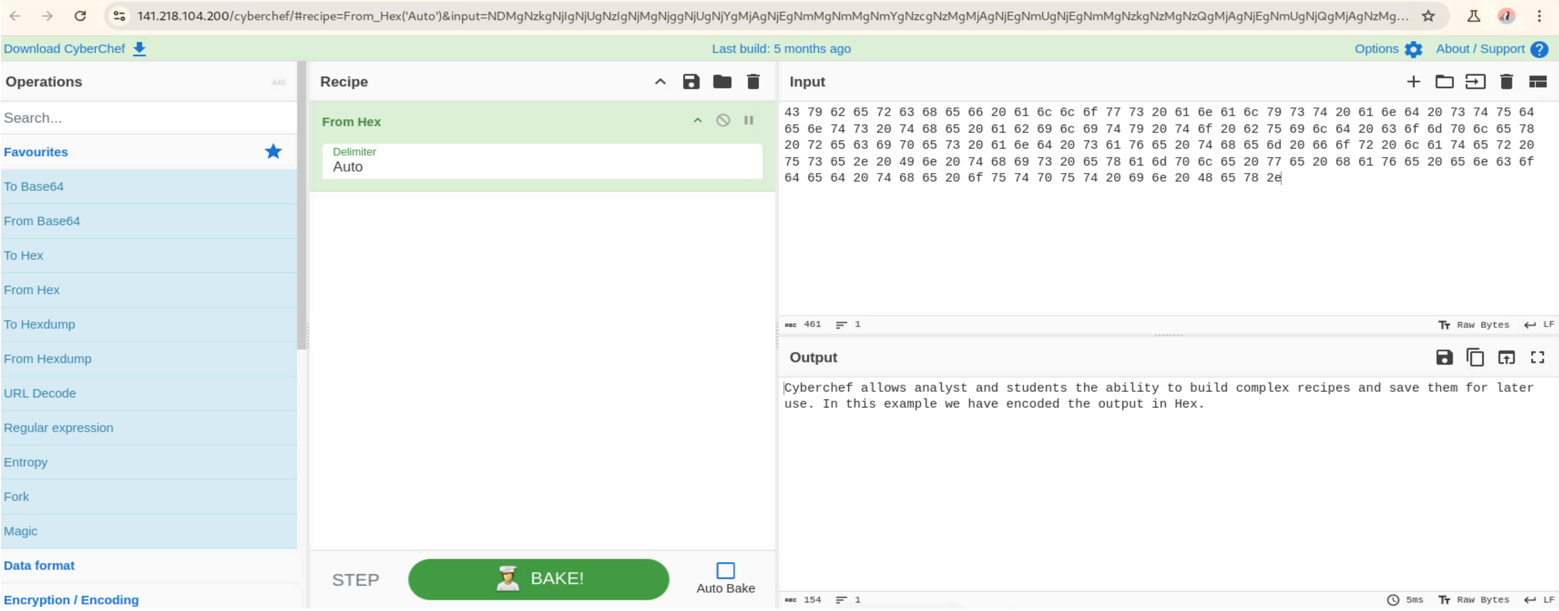
Base64 is an encoding scheme used to convert binary data into a text format using alphanumeric characters, which is often utilized to encode data such as usernames and passwords.

The recipe used was the **"From Base64"** function, which decodes Base64-encoded data back into its original form. The decoded text is a welcome message highlighting CyberChef's capabilities.

2. Hexadecimal Decoding – Recipe Output

Converted Hex-encoded data back to readable text using CyberChef. Discussed how Hex is used to represent binary data in forensics.

2.1. SCREENSHOT (Hex)



141.218.104.200/cyberchef/#recipe=From_Hex('Auto')&input=NDMgNzkgNjIgNjUgNzIgNjMgNjggNjUgNjYgMjAgNjEgNmMgNmMgNmYgNzcgNzMgMjAgNjEgNmUgNjEgNmMgNzkgNzMgNzQgMjAgNjEgNmUgNjQgMjAgNzMg...

Delimiter
Auto

Input

```
43 79 62 65 72 63 68 65 66 20 61 6c 6c 6f 77 73 20 61 6e 61 6c 79 73 74 20 61 6e 64 20 73 74 75 64
65 6e 74 73 20 74 68 65 20 61 62 69 6c 69 74 79 20 74 6f 20 62 75 69 6c 64 20 63 6f 6d 70 6c 65 78
20 72 65 63 69 70 65 73 20 61 6e 64 20 73 61 76 65 20 74 68 65 6d 20 66 6f 72 20 6c 61 74 65 72 20
75 73 65 2e 20 49 6e 20 74 68 69 73 20 65 78 61 6d 70 6c 65 20 77 65 20 68 61 76 65 20 65 6e 63 6f
64 65 64 20 74 68 65 20 6f 75 74 70 75 74 20 69 6e 20 48 65 78 2e
```

Output

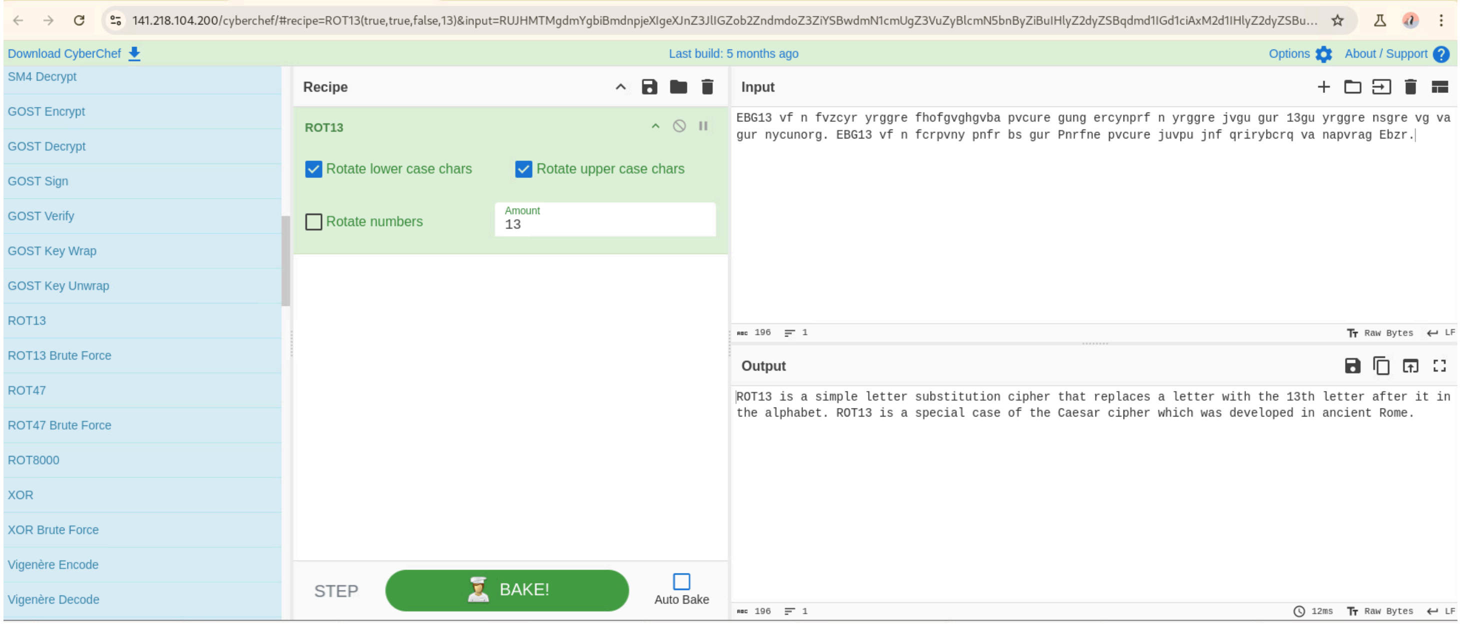
```
Cyberchef allows analyst and students the ability to build complex recipes and save them for later
use. In this example we have encoded the output in Hex.
```

Resulting Output: **Hex** is a base-16 encoding system where each byte is represented by two hexadecimal digits. It's commonly used in computing and digital forensics to represent binary data in a more human-readable format. The decoded message highlights the flexibility of CyberChef for creating complex processing recipes that can be saved and reused.

3. ROT13 Decryption – Caesar Cipher Variant

Used the ROT13 function in CyberChef to decode a simple substitution cipher. Explained how ROT13 is a Caesar cipher used for basic obfuscation.

3.1. SCREENSHOT (ROT13)



```
141.218.104.200/cyberchef/#recipe=ROT13(true,true,false,13)&input=RUJHMTMgdmYgbiBmdnpjeXIgeXJnZ3JlIGZob2ZndmdoZ3ZiYSBwdmN1cmUgZ3VuZyBlcmN5bnByZiBuIHlyZ2dyZSBqdmd1IGd1ciAxM2d1IHlyZ2dyZSBu...
```

ROT13

☑ Rotate lower case chars ☑ Rotate upper case chars

☐ Rotate numbers Amount 13

Input

```
EBG13 vf n fvzcyr yrggre fhofgvghgvba pvcure gung ercynprf n yrggre jvgu gur 13gu yrggre nsgre vg va
gur nycunorg. EBG13 vf n fcrpvny pnfr bs gur Pnrfne pvcure juvpu jnf qrirybcrq va napvrag Ebzr.
```

Output

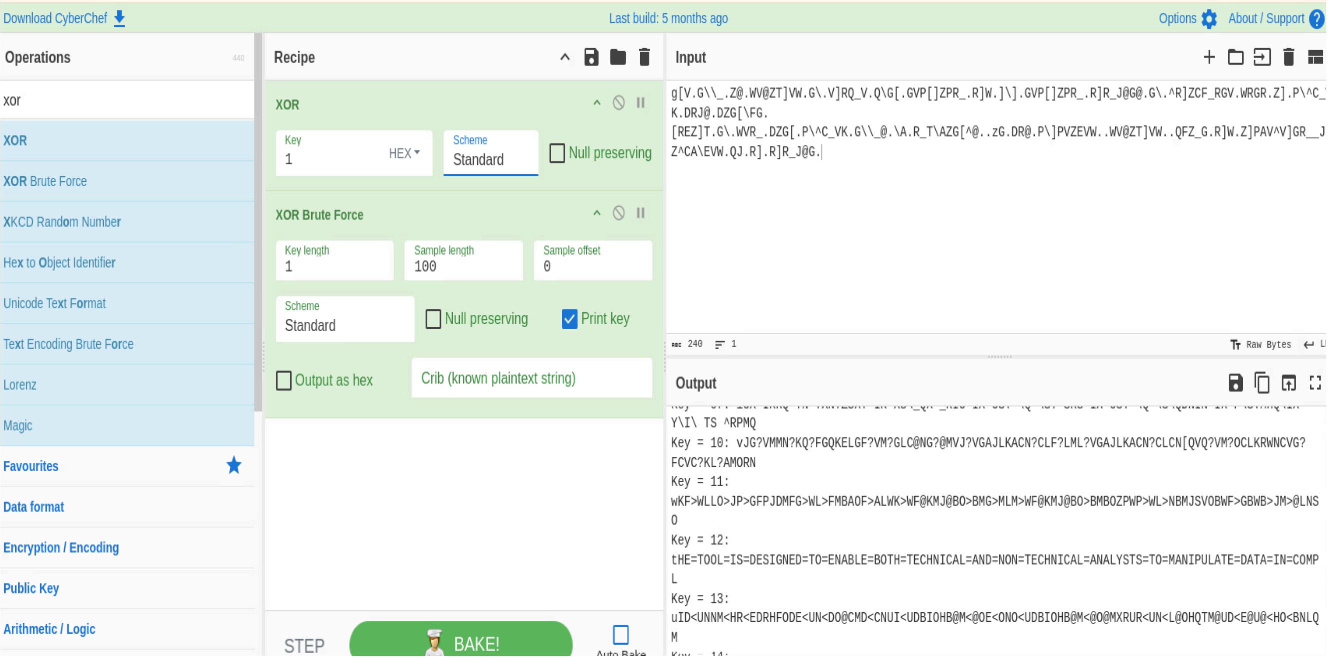
```
ROT13 is a simple letter substitution cipher that replaces a letter with the 13th letter after it in
the alphabet. ROT13 is a special case of the Caesar cipher which was developed in ancient Rome.
```

Resulting Output: **ROT13** is a substitution cipher that replaces each letter with the one 13 places after it in the alphabet. Since the alphabet has 26 letters, applying ROT13 twice returns the original text. The text input provided was encoded using ROT13, meaning every letter was rotated 13 places forward in the alphabet.

4. XOR Brute Force Decryption – First Attempt

Applied an XOR brute force method to decrypt a message. The 12th key produced a meaningful partial message.

4.1 SCREENSHOT (XOR)



xor

XOR — Key: 1 HEX — Scheme: Standard — □ Null preserving

XOR Brute Force — Key length: 1 — Sample length: 100 — Sample offset: 0 — Scheme: Standard — □ Null preserving — ☑ Print key — □ Output as hex — Crib (known plaintext string)

Input

```
g[V.G\\_.Z@.WV@ZT]VW.G\.V]RQ_V.Q\G[.GVP[]ZPR_.R]W.]\].GVP[]ZPR_.R]R_J@G@.G\.^R]ZCF_RGV.WRGR.Z].P\^C_
K.DRJ@.DZG[\FG.
[REZ]T.G\.WVR_.DZG[.P\^C_VK.G\\_@.\A.R_T\AZG[^@..zG.DR@.P\]PVZEVW..WV@ZT]VW..QFZ_G.R]W.Z]PAV^V]GR__J
Z^CA\EVW.QJ.R].R]R_J@G.
```

240 1

Output

```
Y\I\ TS ^RPMQ
Key = 10: vJG?VMMN?KQ?FGQKELGF?VM?GLC@NG?@MVJ?VGAJLKACN?CLF?LML?VGAJLKACN?CLCN[QVQ?VM?OCLKRWNCVG?
FCVC?KL?AMORN
Key = 11:
wKF>WLLO>JP>GFPJDMFG>WL>FMBAOF>ALWK>WF@KMJ@BO>BMG>MLM>WF@KMJ@BO>BMBOZPWP>WL>NBMJSVOBWF>GBWB>JM>@LNS
O
Key = 12:
tHE=TOOL=IS=DESIGNED=TO=ENABLE=BOTH=TECHNICAL=AND=NON=TECHNICAL=ANALYSTS=TO=MANIPULATE=DATA=IN=COMP
L
Key = 13:
uID<UNNM<HR<EDRHFODE<UN<DO@CMD<CNUI<UDBIOHB@M<@OE<ONO<UDBIOHB@M<@O@MXRUR<UN<L@OHQTM@UD<E@U@<HO<BNLQ
M
```

Resulting Output: THE TOOL IS DESIGNED TO ENABLE BOTH TECHNICAL AND NON-TECHNICAL ANALYSTS TO MANIPULATE DATA

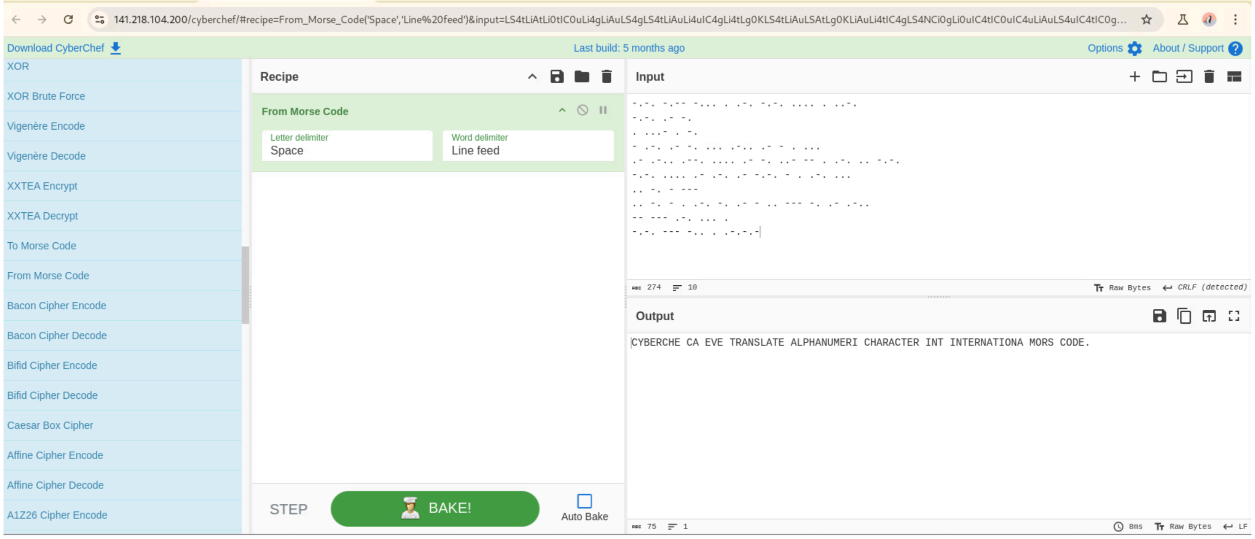
The method employed was an **XOR Brute-Force attack**, which gradually tests different potential keys to unlock the message.

The 12th key successfully revealed part of the original message.

5. Morse Code Decoding – Signal Analysis

Decoded Morse code using CyberChef's translation recipe. Demonstrated the conversion of encoded sequences into readable text.

5.1 SCREENSHOT (Morse Code)



```
141.218.104.200/cyberchef/#recipe=From_Morse_Code('Space','Line%20feed')&input=LS4tLiAtLi0tIC0uLi4gLiAuLS4gLS4tLiAuLi4uIC4gLi4tLg0KLS4tLiAuLSAtLg0KLiAuLi4tIC4gLS4NCi0gLi0uIC4tIC0uIC4uLiAuLS4uIC4tIC0g...
```

From Morse Code

Letter delimiter: Space

Word delimiter: Line feed

Input

```
-.-. -.-- -... . .-. -.-. .... . ..-.
-.-. .- -.
. ...- . -.
- .-. .- -. ... .-.. .- - . ...
.- .-.. .--. .... .- -. ..- -- . .-. .. -.-.
-.-. .... .- .-. .- -.-. - . .-. ...
.. -. - ---
.. -. - . .-. -. .- - .. --- -. .- .-..
-- --- .-. ... .
-.-. --- -.. . .-.-.-
```

274 10

Output

```
CYBERCHE CA EVE TRANSLATE ALPHANUMERI CHARACTER INT INTERNATIONA MORS CODE.
```

75 1 8ms

Resulting Output: **CYBERCHE CAN EVE TRANSLATE CHARACTER INT INTERNATIONA MORS CODE**

Morse code is a method of encoding text characters into sequences of dots and dashes to represent letters, numbers, and punctuation.

The recipe decodes the Morse code into text based on standard international Morse code conventions.

Each sequence of dots and dashes between spaces is converted into its corresponding alphanumeric character.

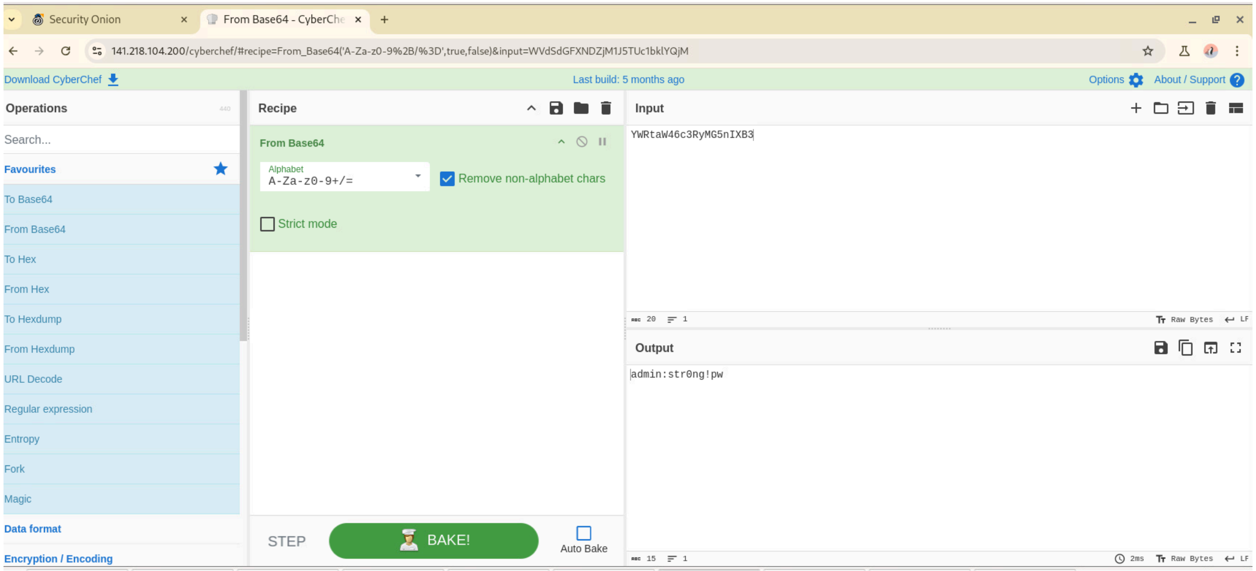
6. Base64 Credential Decoding – Username and Password Extraction

Decoded a Base64 string that revealed login credentials:

- **Username: admin**

- **Password: str0ng!pw**

6.1 SCREENSHOT



```
141.218.104.200/cyberchef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true,false)&input=WVdSdGFXNDZjM1J5TUc1bklYQjM
```

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

☐ Strict mode

Input

```
YWRtaW46c3RyMG5nIXB3
```

Output

```
admin:str0ng!pw
```

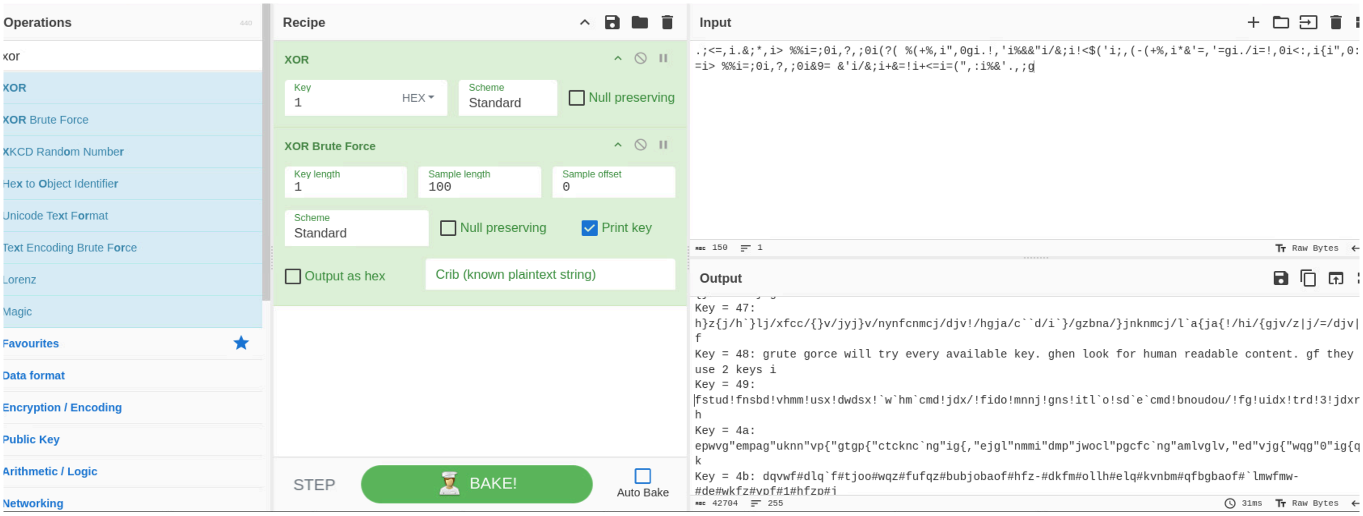
The decoded output was:

- **Username: admin**
- **Password: str0ng!pw**

The recipe used was the **"From Base64"** function, which decodes Base64-encoded data back into its original form, which is often utilized to encode data such as usernames and passwords, as shown in the decoded output

7. XOR Brute Force – Second Attempt and Observations

7.1. SCREENSHOT (XOR)



XOR
Key: 1 HEX
Scheme: Standard
□ Null preserving

XOR Brute Force
Key length: 1
Sample length: 100
Sample offset: 0
Scheme: Standard
□ Null preserving
☑ Print key
□ Output as hex
Crib (known plaintext string)

Input

```
.;<=,i.&;*,i> %%i=;0i,?,;0i(?( %(+%,i",0gi.!,'i%&&"i/&;i!<$('i;,(-(+%,i*&'=,'=gi./i=!,0i<:,i{i",0:
=i> %%i=;0i,?,;0i&9= &'i/&;i+&=!i+<=i=(",:i%&'.,;g
```

150 1

Output

```
Key = 47:
h}z{j/h`}lj/xfcc/{}v/jyj}v/nynfcnmcj/djv!/hgja/c``d/i`}/gzbna/}jnknmcj/l`a{ja{!/hi/{gjv/z|j/=/djv|
f
Key = 48: grute gorce will try every available key. ghen look for human readable content. gf they
use 2 keys i
Key = 49:
fstud!fnsbd!vhmm!usx!dwdsx!`w`hm`cmd!jdx/!fido!mnnj!gns!itl`o!sd`e`cmd!bnoudou/!fg!uidx!trd!3!jdxr
h
Key = 4a:
epwvg"empag"uknn"vp{"gtgp{"ctcknc`ng"ig{,"ejgl"nmmi"dmp"jwocl"pgcfc`ng"amlvglv,"ed"vjg{"wqg"0"ig{q
k
Key = 4b: dqvwf#dlq`f#tjoo#wqz#fufqz#bubjobaof#hfz-#dkfm#ollh#elq#kvnbm#qfbgbaof#`lmwfmw-
#de#wkfz#vpf#1#hfzp#i
```

42704 255 31ms

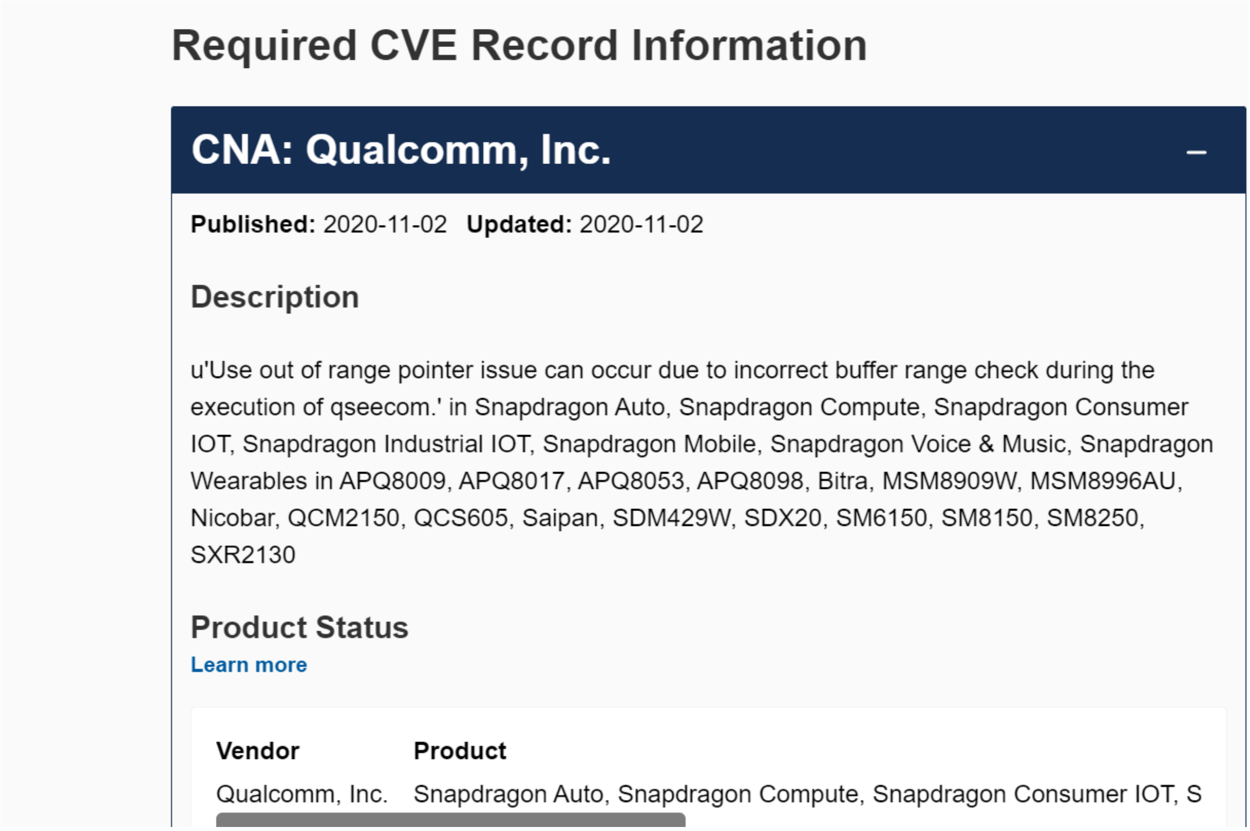
Resulting Output: grute gorce will try every available key. Then look for human-readable content. gf, they use 2 keys I

The method applied was an XOR Brute Force attack, which gradually tests different potential keys to unlock the message.

The 48th key successfully revealed part of the original message, although it seems to contain some typographical errors, which are likely due to imperfections in the brute force process.

Section 2: Using Open Source Intelligence (OSINT) Research

2.1 CVE Entry Identification – Company and Registration Date



Required CVE Record Information

CNA: Qualcomm, Inc.

Published: 2020-11-02 **Updated:** 2020-11-02

Description

u'Use out of range pointer issue can occur due to incorrect buffer range check during the execution of qseecom.' in Snapdragon Auto, Snapdragon Compute, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wearables in APQ8009, APQ8017, APQ8053, APQ8098, Bitra, MSM8909W, MSM8996AU, Nicobar, QCM2150, QCS605, Saipan, SDM429W, SDX20, SM6150, SM8150, SM8250, SXR2130

Product Status

Learn more

Vendor	Product
Qualcomm, Inc.	Snapdragon Auto, Snapdragon Compute, Snapdragon Consumer IOT, S

Identified that **CVE-2020-3693** is associated with **Qualcomm, Inc.**, registered on **2020-11-02**.

Short discussion

Common Vulnerabilities and Exposures (CVE) List

The CVE List is a public catalog of all known cybersecurity weaknesses in software. Each weakness is allocated its own ID, so it's easier for companies to keep track of what is happening. It is a list of software problems that could be easy access points for hackers if not fixed.

Goel and Mehtre (2015) mention that this list is essential because it helps companies scan their systems for any of these known problems. For example, if a vulnerability in a common software like Google Chrome is published, companies can quickly check if they're using the affected version and patch it to block hackers from exploiting the weakness.

National Vulnerability Database (NVD)

The NVD is run by the National Institute of Standards and Technology (NIST) and works closely with the CVE List. It adds more details to each CVE, such as how dangerous a weakness is, and even suggests ways to fix it. One key tool it offers is the Common Vulnerability Scoring System (CVSS), which rates how critical a vulnerability is. This makes it easier for companies to decide which problems to fix first.

Scarfone and Mell (2009) point out that the CVSS system helps businesses focus on the biggest threats. If a vulnerability is rated as severe, like a 9 out of 10, it means they need to fix it right away. This prioritization helps companies tackle the most dangerous problems first, keeping them one step ahead of hackers.

VirusTotal

VirusTotal is an online virus scanner that lets companies upload files, links, or IP addresses to check if they are infected with malware. VirusTotal uses results from multiple antivirus programs, so it's more thorough than a single antivirus tool.

Santos et al. (2011) explain that VirusTotal helps companies catch threats that might go unnoticed by just one antivirus program. If a company receives a suspicious email with an attachment, they can upload the file to VirusTotal. If multiple scanners flag it as dangerous, the company can block it, preventing a possible attack. This tool helps companies stop viruses and malware in their tracks.

Risk IQ/Microsoft Defender Intelligence Community

Risk IQ, now part of Microsoft Defender, is an internet-wide security guard that provides comprehensive threat intelligence on external threats by scanning the web for potential risks, such as exposed servers or fake websites that look like a particular company's but are actually phishing scams. RiskIQ helps businesses see what's happening outside their network and take action before hackers can do damage.

RiskIQ gives companies a heads-up about potential attacks before they happen. For example, if it finds a phishing website that mimics the company's real site, the business can get it taken down before customers get tricked. This proactive approach keeps companies safer.

Conclusion

Using tools like the CVE List, NVD, VirusTotal, and RiskIQ makes it easier for businesses to defend themselves against cyberattacks. The CVE List and NVD help them track and prioritize which vulnerabilities to fix, while VirusTotal adds extra protection by checking files for malware. RiskIQ's ability to monitor external threats further helps companies prevent attacks before they happen. These tools help keep organizations ahead of hackers and strengthen their overall security.

References

Goel, S., & Mehtre, B. M. (2015). *Vulnerability assessment & penetration testing as a cyber defense technology*. Procedia Computer Science.
https://www.sciencedirect.com/science/article/pii/S1877050915019870

MITRE Corporation. (2023). *Common vulnerabilities and exposures (CVE)*.

https://cve.mitre.org/

National Institute of Standards and Technology. (2023). *National Vulnerability Database (NVD)*.
https://nvd.nist.gov/

Santos, I., Brezo, F., & Bringas, P. G. (2011). *Collective classification for malware detection*. In *IEEE Conference on Systems, Man, and Cybernetics.*

https://www.researchgate.net/publication/221436323_Collective_Classification_for_Unknown_Malware_Detection

VirusTotal. (2023). *VirusTotal - Free online virus, malware, and URL scanner*.
https://www.virustotal.com/

RiskIQ. (2023). *Microsoft Defender Intelligence*.

https://community.riskiq.com/