GitHub Gist   Search...   All gists   Back to GitHub   Sign in   Sign up

Instantly share code, notes, and snippets.

Neo23x0 / **Base64_CheatSheet.md**   ☆ Star 267   Fork 43

Last active 2 weeks ago

<> **Code**   Revisions 61   ☆ Stars 267   Forks 43   Embed ▾   `<script>`   Download ZIP

Learning Aid - Top Base64 Encodings Table

<> **Base64_CheatSheet.md**   Raw

Base64 Patterns - Learning Aid

Base64 Code	Mnemonic Aid	Decoded*	Description
`JAB`	🗣 Jabber	`$.`	Variable declaration (UTF-16), e.g. `JABlAG4AdgA` for `$env:`
`TVq`	📺 Television	`MZ`	MZ header
`SUVY`	🚙 SUV	`IEX`	PowerShell Invoke Expression
`SQBFAF`	🍜 Squab favorite	`I.E.`	PowerShell Invoke Expression (UTF-16)
`SQBuAH`	🍜 Squab uahhh	`I.n.`	PowerShell Invoke string (UTF-16) e.g. `Invoke-Mimikatz`
`PAA`	💪 "Pah!"	`<.`	Often used by Emotet (UTF-16)
`cwBhA`	🦁 Chewbaka	`s.a.`	Often used in malicious droppers (UTF-16) 'sal' instead of 'var'
`aWV4`	🤯 Awe version 4	`iex`	PowerShell Invoke Expression

`aQBlA`	💦 Aqua Blah (aquaplaning)	`i.e.`	PowerShell Invoke Expression (UTF-16)
`R2V0`	🤖 R2D2 but version 0	`Get`	Often used to obfuscate imports like GetCurrentThreadId
`dmFy`	👿 defy / demonify	`var`	Variable declaration
`dgBhA`	debugger + high availability	`v.a.`	Variable declaration (UTF-16)
`dXNpbm`	Dixon problem	`usin`	Often found in compile after delivery attacks
`H4sIA`	🚁 HForce (Helicopter Force) I agree		gzip magic bytes (0x1f8b), e.g. `echo 'test' | gzip -cf | base64`
`Y21k`	🎆 Year 21k bug	`cmd`	As used in `cmd.exe /c wscript.exe` or the like
`IAB`	😙 I am bored	`s`	wide lower case `s`, often something like `sEt-iTem`
`cABhAH`	🕋 Kaaba	`p.a.`	wide formatted `param`
`Qzpc`	🖥 Quiz PC	`C:\`	Root of Windows partition (upper case)
`Yzpc`	🖥 Yes PC	`c:\`	Root of Windows partition (lower case)
`UEs`	🏢 Upper East Side	`PK`	ZIP, Office documents
`ey`	🗣 Hey	`{`	Indicates JSON data

* the `.` stands for `0x00` found in UTF-16 encoded text

Often found patterns

Base64 Code	Decoded	Description
`AAAAAAAAAAAA`	`\x00\x00\x00\x00\x00\x00\x00\x00\x00`	Sequence of binary zeros
`////////////`	`\xFF\xFF\xFF\xFF\xFF\xFF\xFF\xFF\xFF`	Sequence of 0xFF bytes
`ICAgICAgICAg`		Sequence of space characters

Cyber Chef Recipe

https://gchq.github.io/CyberChef/#recipe=Fork('%5C%5Cn','%5C%5Cn',false)From_Base64('A-Za-z0-9%2B/%3D',true)&input=SkFCCIRWcQpQQUEKU1VWWQpTUUJGQUYKYVdNAphUUJsQQpSMlYwCmRtRnkKZGdaCaEEKY3dCaEEKZFhOcGGJtCkg0c0lBRIdXc2wwQUF5dEpMUzdoQWdER05iazdCCUUFBQUE9PQ

References

Tweet

Tweet and Thread https://twitter.com/cyb3rops/status/1187341941794660354

JAB

https://www.hybrid-analysis.com/sample/ce0415b6661ef66bbedb69896ad1ece9ee4e6dfde9925e9612aec7bbf1cb7bc5?environmentId=100

PAA

Emotet process command line https://app.any.run/tasks/dfba6d53-7a93-4d8b-86ba-4e737ad06b06/

cwBha

Explanation https://threat.tevora.com/5-minute-forensics-decoding-powershell-payloads/

Sample https://www.hybrid-analysis.com/sample/b744129bfe54de8b36d7556ddfcc55d0be213129041aacf52b7d2f57012caa60?environmentId=100

srcr commented on Oct 15, 2020 · · ·

should MITRE ATT4CK be changed to MITRE ATT&CK? Very nice list though. Danke

Neo23x0 commented on Oct 15, 2020   `Author`   ···

I'll remove the MITRE reference completely. Don't want that anyone tells me that it lacks a (r) character.

ohader commented on Mar 9, 2022   ···

Please add `YTo` and `Tzo`, start of a serialized array and object in PHP (`a:` and `O:`), good indicator when searching for insecure deserialization vulns. PoC at https://3v4l.org/9PI63

dc-secureworks commented on Apr 7, 2022   ···

Please mention that these encodings only hold at the start of the encoded string and will not work consistently at other positions. Most people aren't aware of the 3 byte chunks used to encode base64 and the 3 encodings repetition you get by shifting one byte at a time. Also IAB decodes to 0x20 0x00 not 0x73 0x00.
Here's an example of determining the consistent subset of an encoded string at the 3 offsets. (wide "s" used)
https://gchq.github.io/CyberChef/#recipe=Fork('%5C%5Cn','%5C%5Cn',false)From_Hex('Auto')To_Base64('A-Za-z0-9%2B/%3D')&input=NzMgMGDAgMDAgMDAKNzMgMGDAgNzcgNzcKNzMgMGDAgRkYgRkYKMDAgNzMgMGDAgMDAgMDAKNzcgNzMgMGDAgNzcgNzcKRkYgNzMgMGDAgRkYgRkYKMDAgNzMgMGDAgMDAgMDAKNzcgNzcgNzMgMGDAgNzcgNzcgNzcKRkYgRkYgNzMgMGDAgRkYgRkYgRkYKMDAgMDAgNzMgMGDAgMDAgMDAKNzcgNzcgNzcgNzMgMGDAgNzcgNzcgNzcKRkYgRkYgRkYgNzMgMGDAgRkYgRkYgRkYKMDAgMDAgNzMgMGDAgMDAgMDAKNzcgNzcgNzcgNzcgNzMgMGDAgNzcgNzcgNzcKRkZGRiBGRiBGRiA3MyAwMCBGRiBGRiBGRgoK
Unless you're only checking the start of a buffer you'll need to trim a bit off the front and back to safely detect the consistent substring of what you're looking for. From the example of wide "s" you have cw, MA, zA. I added extra shifts in Cyberchef to illustrate the cycle and how the pattern comes back every 3rd byte.

antoinet commented on Jan 27, 2023   ···

Very cool gist! It occured to me: `ey` misses a quote after the brace: `{"` . And any idea for `aHR0cHM6Ly8=` (`https://`)?

rickhenderson commented on Aug 3, 2023

•••

This is amazing btw.

Sign up for free to join this conversation on GitHub. Already have an account? Sign in to comment

Terms   Privacy   Security   Status   Docs   Contact   Manage cookies   Do not share my personal information

© 2024 GitHub, Inc.