

Forum:[Hack Forums](#)

Thread: [disk, directory, file GCM Mode encryption/decryption exe, c++ - POC](#)

SovietBastard - August 18, 2024, 4:34 pm

First of all, im quite new here, so dont kill me if i violate some kind of rule im not aware of, any comment or opinion or even critique is appreciated and keep in mind this is just something i did on the fly to share with people here as i saw someone in the other threads complaining that nobody is posting anything, i decided to make a little contribution, if you like it, let me know, if you dont like it, let me know as well, im trying to make a point here, showing how to do c++ full disk encryption using 2 other modules of mine, one for encryption and decryption, the other for filesystem operations(I/O...), encryption is performed using CGM(galois-counter mode) mode of operation, which is an authenticated encryption/decryption cipher, it combines the counter(CTR) mode with a galois field multiplication for authentication and relies on cipher tags which will be used to authenticated the block cipher. GCM mode use PBKDF2 HMAC(hash-based message authentication code), this is used to derive the final key, a salt is added to the cipher just to make happy everyone, and yes.... the IV is unique for every counter block generated, this is the standard for serious security, well i don't know if is standard everywhere, but definitely if i have GCM mode available along with CBC and CTR modes, i will always select GCM, it has this balance which is perfect between efficiency, security and integrity, definitely good choice if you want performance and speed of execution, and.... for the algorithm, used AES, standard shit, the algorithm can be changed dynamically at runtime and can be changed with (mars, twofish or rc6).

was thinking about CBC mode, but it does not support authentication, but ... it supports more algorithms than GCM, also GOST mode which is pretty good. and right now im thinking about adding all 6 modes of operation(cbc, gcm, aex, cfb, ofb and ctr), im still thinking about it, it is possible that i will end up using GCM mode with AES anyway so fuck that... im writing this on a linux machine x86-64 bit system, but there shouldn't be any compatibility issues if you take the source code and re-compile it on your system, better if use gcc or clang compiler because i usually make use of inline compiler optimization attributes to optimize where possible... Librar.y crypto++ must be installed on your system or compilation will fail, to install it on linux type "apt install libcrypto++-dev" or something like that, and it will install the library which the cryptography module heavily relies on for its compatibility, availability and efficiency. The encrypted data can be decrypted only if the key match, to decrypt just rerun the program and execute "decrypt" on the target with the correct key and it will decrypt the content if the key is good. Side project folks.

Last thing is... due to some issues with large files when reading their content, module FSController uses memory to store buffers and stream content, this increased the speed of execution a lot.

Run:

Code \$./diskcrypt [additional-flags]

Common syntax:

Code \$./diskcrypt --target target_path --secret 123secret --encrypt

\$./diskcrypt --target target_path --secret 123secret --decrypt

OR

\$./diskcrypt --interface // easier for someone out there...

Flags:

the following flags can be used for execute&exit or persistent execution mode:

--help
--recursive
--verbose
--backup
--backup-path <PATH>
--secret <SECRET>
--target <TARGET_PATH>
--direct
--encrypt
--decrypt
--use-atomic
--interface

--recursive: encrypt/decrypt recursively
--verbose: print output
--backup: create backup before
--backup-path: set backup path, will create if not found
--secret: set secret key
--target: set target path
--direct: direct execution, will no stop execution for confirmations
--encrypt: encryption of target
--decrypt: decryption of target
--use-atomic: sets verbose to false and --direct to true
--interface uses a cool interface for command push

Use --interface for persistent execution, or --direct for single execution.

below attached some screenshots, showing the command&control console interface and other useful information that might interest you.

LAUNCH

SHOW COMMANDS, OPTIONS...

NON ENCRYPTED DIRECTORY(RECURSIVE)

ENCRYPTED

all files are encrypted, but the difference is only observable on the image file on my system

Source code and binary: <https://github.com/Somorpher/Diskrypt>

dobroy_000 - August 19, 2024, 4:45 am

nice share bro, looks like you're busy these days ah?

super_sayan - August 19, 2024, 4:54 am

(Aug 18, 2024 04:34 PM)SovietBastard Wrote: [\[CITE\]](#) First of all, im quite new here, so dont kill me if i violate some kind of rule im not aware of, any comment or opinion or even critique is appreciated and keep in mind this is just something i did on the fly to share with people here as i saw someone in the other threads complaining that nobody is posting anything, i decided to make a little contribution, if you like it, let me know, if you dont like it, let me know as well, im trying to make a point here, showing how to do c++ full disk encryption using 2 other modules of mine, one for encryption and decryption, the other for filesystem operations(I/O...), encryption is performed using CGM(galois-counter mode) mode of operation, which is an authenticated encryption/decryption cipher, it combines the counter(CTR) mode with a galois field multiplication for authentication and relies on cipher tags which will be used to authenticated the block cipher. GCM mode use PBKDF2 HMAC(hash-based message authentication code), this is used to derive the final key, a salt is added to the cipher just to make happy everyone, and yes.... the IV is unique for every counter block generated, this is the standard for serious security, well i don't know if is standard everywhere, but definitely if i have GCM mode available along with CBC and CTR modes, i will always select GCM, it has this balance which is perfect between efficiency, security and integrity, definitely good choice if you want performance and speed of execution, and.... for the algorithm, used AES, standard shit, the algorithm can be changed dynamically at runtime and can be changed with (mars, twofish or rc6).

was thinking about CBC mode, but it does not support authentication, but ... it supports more algorithms than GCM, also GOST mode which is pretty good. and right now im thinking about adding all 6 modes of operation(cbc, gcm, aex, cfb, ofb and ctr), im still thinking about it, it is possible that i will end up using GCM mode with AES anyway so fuck that... im writing this on a linux machine x86-64 bit system, but there shouldn't be any compatibility issues if you take the source code and re-compile it on your system, better if use gcc or clang compiler because i usually make use of inline compiler optimization attributes to optimize where possible... Librar.y crypto++ must be installed on your system or compilation will fail, to install it on linux type "apt install libcrypto++-dev" or something like that, and it will install the library which the cryptography module heavily relies on for its compatibility, availability and efficiency. The encrypted data can be decrypted only if the key match, to decrypt just rerun the program and execute "decrypt" on the target with the correct key and it will decrypt the content if the key is good. Side project folks.

Last thing is... due to some issues with large files when reading their content, module FSController uses memory to store buffers and stream content, this increased the speed of execution a lot.

Run:

Code \$./diskcrypt [additional-flags]

Common syntax:

Code \$./diskcrypt --target target_path --secret 123secret --encrypt

\$./diskcrypt --target target_path --secret 123secret --decrypt

OR

\$./diskcrypt --interface // easier for someone out there...

Flags:

the following flags can be used for execute&exit or persistent execution mode:

--help

--recursive

--verbose

--backup

--backup-path <PATH>

--secret <SECRET>

--target <TARGET_PATH>
--direct
--encrypt
--decrypt
--use-atomic
--interface

--recursive: encrypt/decrypt recursively
--verbose: print output
--backup: create backup before
--backup-path: set backup path, will create if not found
--secret: set secret key
--target: set target path
--direct: direct execution, will no stop execution for confirmations
--encrypt: encryption of target
--decrypt: decryption of target
--use-atomic: sets verbose to false and --direct to true
--interface uses a cool interface for command push

Use --interface for persistent execution, or --direct for single execution.

below attached some screenshots, showing the command&control console interface and other useful information that might interest you.

LAUNCH

SHOW COMMANDS, OPTIONS...

NON ENCRYPTED DIRECTORY(RECURSIVE)

ENCRYPTED

all files are encrypted, but the difference is only observable on the image file on my system

Source code and binary: <https://github.com/Somorpher/Diskrypt>

my friend, im new here, but looks like you're the ONLY one sharing code on this forum, why is that? do people actually share code here?

SovietBastard - August 19, 2024, 5:12 am

(Aug 19, 2024 04:54 AM)super_sayan Wrote: [\[CITE\]](#) my friend, im new here, but looks like you're the ONLY one sharing code on this forum, why is that? do people actually share code here?

they might do, im just sharing side code i write for fun, if people like it, good,if not, i wont post for long anyway...

super_sayan - August 19, 2024, 5:15 am

(Aug 19, 2024 05:12 AM)SovietBastard Wrote: [\[CITE\]](#) they might do, im just sharing side code i write for fun, if people like it, good,if not, i wont post for long anyway...

how long have you been here? and why you dont want to post anymore if i may ask?

SovietBastard - August 19, 2024, 5:32 am

(Aug 19, 2024 05:15 AM)super_sayan Wrote: [\[CITE\]](#) how long have you been here? and why you dont want to post anymore if i may ask?

1 month i would say, i wont post anymore bcz im really busy atm with other things..

super_sayan - August 19, 2024, 6:07 am

(Aug 19, 2024 05:32 AM)SovietBastard Wrote: [\[CITE\]](#) 1 month i would say, i wont post anymore bcz im really busy atm with other things..

1 month is quite new, what do you do for living?