

Forum:[Hack Forums](#)

Thread: [Folder Encryption / Source released!](#)

taktaktak - July 5, 2015, 7:15 pm

Greetings all!

I've been working up a new application and figured I would release it here for all of you that might find it usefull! If you do use this program please leave some feadback, I'd like to hear what you think, and it will encourage me to keep developing this project! And just to clarify, I did get the idea from this post: <https://hackforums.net/showthread.php?tid=4888596>, but I'm to lazy to convert the code, so I just made my own version with my twist on it :)

The general Idea of the program is this: You chose a folder, and the folder is encrypted and all the data is stuffed into a "Files.Encrypted" in the same directory as the program. Then after you've done that, create a decryption stub, this stub is only compatable with that "Files.Encrypted" file, they both have the same salt packed inside, and will only work with eachother. Then when you want to unpack the files, run the unpack stub, enter the password it was created with, and it will place the files like they were in the folder, inside a folder located in the same directory as the application called "Decompressed".

Some key features are:

- Always uses 256-bit AES
- As secure as possible (uses standardized algorithms)
- "Polymorphic" / unique output (CBC cipher mode with random IV)
- Minimal RAM overhead. (Writes as it reads)
- No critical information is stored inside the stub

Photos:

Spoiler ([Click to View](#))

[img]http://nodistribute.com/imageresult/?result=JrsmegWuV1CaQM[/img]

<https://github.com/Icemantheditor/FolderEncryption>

I have replaced the DL link to the repository on my github, if you don't want to waste time compiling it, there is a pre-compiled binary inside of bin/Debug/FolderEncryption.exe

VmX5 - July 5, 2015, 8:56 pm

Very good job!

but quick suggestion You SHOULD let the user pick a password or randomly generate it

taktaktak - July 5, 2015, 9:57 pm

(Jul 5, 2015 08:56 PM)VirusCrafter5 Wrote: [\[CITE\]](#) Very good job!

but quick suggestion You SHOULD let the user pick a password or randomly generate it

It is randomly picked, the problem is the password has to be 32 characters long, I guess I could just add say x's to get a short password up to length, and chop long passwords to length behind the scenes to simplify it for the end user.

Hoody - July 5, 2015, 10:03 pm

I have a few suggestions for you.

- 1) Compression. Like gzip
- 2) Store the data inside the decrypter. (makes it easier to move around / share)
- 3) when you start the decrypter have it prompt for the password, that way you cant reverse engineer it to get the password

taktaktak - July 5, 2015, 10:15 pm

(Jul 5, 2015 10:03 PM)Hoody Wrote: [\[CITE\]](#) I have a few suggestions for you.

- 1) Compression. Like gzip
- 2) Store the data inside the decrypter. (makes it easier to move around / share)
- 3) when you start the decrypter have it prompt for the password, that way you cant reverse engineer it to get the password

I've tried Gzip and it didn't help, it actually increased size. I was thinking about that, however people might want to have them separate. And it is like that, the password isn't stored in the stub.

Hoody - July 5, 2015, 10:21 pm

(Jul 5, 2015 10:15 PM)Killpot Wrote: [\[CITE\]](#) I've tried Gzip and it didn't help, it actually increased size. I was thinking about that, however people might want to have them separate. And it is like that, the password isn't stored in the stub.

hmm, ok then :P

well, for gzip. You have to zip the data before you encrypt it, else it will inflate.

taktaktak - July 5, 2015, 10:26 pm

(Jul 5, 2015 10:21 PM)Hoody Wrote: [\[CITE\]](#) hmm, ok then :P

well, for gzip. You have to zip the data before you encrypt it, else it will inflate.

Idk about that, how would I zip a string? Because if you're talking about zipping the file, some can get very very large, to large to read in one string because it would exhaust all memory and throw an exception, so it would have to happen as I write it. And I've tried 3 different methods of gzip ing strings and they all inflated the file size.

Hoody - July 5, 2015, 10:33 pm

(Jul 5, 2015 10:26 PM)Killpot Wrote: [\[CITE\]](#) Idk about that, how would I zip a string? Because if you're talking about zipping the file, some can get very very large, to large to read in one string because it would exhaust all memory and throw an exception, so it would have to happen as I write it. And I've tried 3 different methods of gzip ing strings and they all inflated the file size.

Oh god i cant even remember the last time i used gzip. But from what i recall, you can use a stream and just feed it 10kb or so at a time. So read to the buffer, pass to the stream, dispose the buffer, repeat.

Also, after coding in C for a while. I cringe at the word "string". lol use a byte array :) haha or char* if ya know what i mean ;)

taktaktak - July 5, 2015, 10:37 pm

(Jul 5, 2015 10:33 PM)Hoody Wrote: [\[CITE\]](#) Oh god i cant even remember the last time i used gzip. But from what i recall, you can use a stream and just feed it 10kb or so at a time. So read to the buffer, pass to the stream, dispose the buffer, repeat.

Also, after coding in C for a while. I cringe at the word "string". lol use a byte array :) haha or char* if ya know what i mean ;)

Hmm sounds interesting, and simple enough, I'll have to give it a shot. And sadly I have a little

experience in c++ so kinda :P

BahNahNah - July 5, 2015, 10:39 pm

(Jul 5, 2015 09:57 PM)Killpot Wrote: [\[CITE\]](#) It is randomly picked, the problem is the password has to be 32 characters long, I guess I could just add say x's to get a short password up to length, and chop long passwords to length behind the scenes to simplify it for the end user.

The password does not need to be 32 characters.

either use PasswordDeriveBytes or hash it to 32 bit.

taktaktak - July 5, 2015, 10:44 pm

(Jul 5, 2015 10:39 PM)BahNahNah Wrote: [\[CITE\]](#) The password does not need to be 32 characters. either use PasswordDeriveBytes or hash it to 32 bit.

Actually you just reminded me that I switched encryption methods, I totally forgot to update that, thanks I suppose.

Hoody - July 5, 2015, 10:45 pm

(Jul 5, 2015 10:39 PM)BahNahNah Wrote: [\[CITE\]](#) The password does not need to be 32 characters. either use PasswordDeriveBytes or hash it to 32 bit.

But. Bannawwa (hard W)

you sure you mean 32 bit? thats.. 4 bytes... like 0x12345678

tbh im talking out of my ass. but for real. i doubt they have a 32bit password

BahNahNah - July 5, 2015, 10:47 pm

(Jul 5, 2015 10:45 PM)Hoody Wrote: [\[CITE\]](#) But. Bannawwa (hard W)

you sure you mean 32 bit? thats.. 4 bytes... like 0x12345678

tbh im talking out of my ass. but for real. i doubt they have a 32bit password

I dont remember. I know that one of the key sizes for Rijndael is the same as an MD5 hash (128 bits i think).

Hoody - July 5, 2015, 10:55 pm

(Jul 5, 2015 10:47 PM)BahNahNah Wrote: [\[CITE\]](#) I dont remember. I know that one of the key sizes for Rijndael is the same as an MD5 hash (128 bits i think).

well.. i think hes using aes256? im too lazy to check the OP. but if so. lets do some math :)

bit / 8 = byte

now you see bannawwa.

Code ____32__

8|256

24

16

16

0

256 / 8 = 32

so OP here needs to use 32 characters.

taktaktak - July 5, 2015, 11:08 pm

(Jul 5, 2015 10:55 PM)Hoody Wrote: [\[CITE\]](#) well.. i think hes using aes256? im too lazy to check the OP. but if so. lets do some math :)

bit / 8 = byte

now you see bannawwa.

Code ____32_

8|256

24

16

16

0

256 / 8 = 32

so OP here needs to use 32 characters.

(Surprise that's what I said)

(Jul 5, 2015 10:55 PM)Hoody Wrote: [\[CITE\]](#) well.. i think hes using aes256? im too lazy to check the OP. but if so. lets do some math :)

bit / 8 = byte

now you see bannawwa.

Code ____32_

8|256

24

16

16

0

256 / 8 = 32

so OP here needs to use 32 characters.

I tried adding in your gzip and it was just giving me trouble, if people want to compress the data then ill just let them do it with a third party program.

Randy_Savage_xX - July 6, 2015, 8:12 pm

Thanks man! It works great!

taktaktak - July 8, 2015, 7:57 pm

Source has been released folks, enjoy!

tweety - July 8, 2015, 8:13 pm

You have so many obsolete operations in your code. Catching things and proceeding with operations or not error handling correctly. Using the My Namespace should be avoided at all costs. I mean..

PHP Code

```
<?php
```

```
Directory.CreateDirectory(Directory.GetCurrentDirectory +
```

```
"\Decompressed\" + path.Substring(0, path.Length - seguro(seguro.Length - 1).Length - 1))
```

What? Path.Combine()? String.Format()?

You didn't even write the encryption method: <https://hackforums.net/showthread.php?tid=1489558>

taktaktak - July 8, 2015, 9:14 pm

(Jul 8, 2015 08:13 PM)Tweety. Wrote: [\[CITE\]](#) You have so many obsolete operations in your code. Catching things and proceeding with operations or not error handling correctly. Using the My Namespace should be avoided at all costs. I mean..

PHP Code

```
<?php
```

```
Directory.CreateDirectory(Directory.GetCurrentDirectory +  
"\\Decompressed\\" + path.Substring(0, path.Length - seguro(seguro.Length - 1).Length - 1))
```

What? Path.Combine()? String.Format()?

You didn't even write the encryption method: <https://hackforums.net/showthread.php?tid=1489558>

I never claimed I did write the encryption method, I have no idea how to write them. And yeah, i'm not happy with what you quoted, it was simply a 1-3 day project and it was just to get it working, but thats no excuse for shitty code.

LUOIdRentA - July 9, 2015, 3:00 am

(Jul 5, 2015 09:57 PM)Killpot Wrote: [\[CITE\]](#) It is randomly picked, the problem is the password has to be 32 characters long, I guess I could just add say x's to get a short password up to length, and chop long passwords to length behind the scenes to simplify it for the end user.

Allow the user to enter their own password (it can be whatever length) and then hash it to MD5 so that it's always 32 characters long. Then obviously when the user is going to decrypt hash what the user has entered as the password into MD5 and then compare the two strings.

Nikey646 - July 9, 2015, 3:48 am

(Jul 9, 2015 03:00 AM)thevegita730 Wrote: [\[CITE\]](#) Allow the user to enter their own password (it can be whatever length) and then hash it to MD5 so that it's always 32 characters long. Then obviously when the user is going to decrypt hash what the user has entered as the password into MD5 and then compare the two strings. Just a FYI: The MD5 hash function is no longer considered secure. Not only is it vulnerable to collisions, but it has huge rainbow tables available, and adding a salt won't help due to the collisions. Not to mention that it is entirely too easy to brute force a MD5 hash on modern hardware.

Edit: What is the purpose of the compiler in this program? You don't modify the Source what so ever, infact it almost seems like you just wanted a compiler as to not provide two .exe...

LUOIdRentA - July 9, 2015, 4:51 am

(Jul 9, 2015 03:48 AM)Nikey646 Wrote: [\[CITE\]](#) Just a FYI: The MD5 hash function is no longer considered secure. Not only is it vulnerable to collisions, but it has huge rainbow tables available, and adding a salt won't help due to the collisions. Not to mention that it is entirely too easy to brute force a MD5 hash on modern hardware.

Edit: What is the purpose of the compiler in this program? You don't modify the Source what so ever, infact it almost seems like you just wanted a compiler as to not provide two .exe...

I understand what you're saying, and you're absolutely right, but what I was referring to was not meant to be used to store the password, but simply be used so that the user can input a password of any length, the program will then turn that into a MD5 hash, and encrypt with that. Later when the user wants to decrypt, they must input the same password and the program will once again, turn it into a MD5 hash and attempt to decrypt the files. There's no real need to even store this anywhere...

The main purpose was to get passed OP's issue of not being able to allow the user to input their own password

Nikey646 - July 9, 2015, 5:36 am

(Jul 9, 2015 04:51 AM)thevegita730 Wrote: [\[CITE\]](#) I understand what you're saying, and you're absolutely right, but what I was referring to was not meant to be used to store the password, but simply be used so that the user can input a password of any length, the program will then turn that into a MD5 hash, and encrypt with that. Later when the user wants to decrypt, they must input the same password and the program will once again, turn it into a MD5 hash and attempt to decrypt the files. There's no real need to even store this anywhere...

The main purpose was to get passed OP's issue of not being able to allow the user to input their own password. What you are failing to see is the MD5 hash becomes the encryption key for the encrypted files. This means that the key is subjected to every form of attack that the MD5 hashing system is. What's worse is the Stub code includes the encryption and decryption code, this means anyone can easily rip that code out and use an application to brute force the file.

The problem with that "PolyAES" class, is that it claims to be "poly", but all it actually is, is a standard implementation of AES with unique Key/IV, ensuring that it's harder to break the encryption, however half of this is meaningless if you use something that can be collided with such ease as MD5, and what makes it even more meaningless is that the IV (from what I can tell) is stored in the file.

Simply put: MD5 is not secure for any form of protection, be it as a Key to an encryption, or a hashed version of a password.

Quick Edit: But talking about how secure this is, is entirely meaningless since it does not even use a Secure RNG, or a larger-than-generic Key Map. I'm pretty damn certain that any password generated by this application is as insecure as the average password of this site's userbase.

OffCode - July 9, 2015, 5:45 am

That's looks so nice man! Keep up good work!

taktaktak - July 9, 2015, 6:14 am

(Jul 9, 2015 03:48 AM)Nikey646 Wrote: [\[CITE\]](#) Edit: What is the purpose of the compiler in this program? You don't modify the Source what so ever, infact it almost seems like you just wanted a compiler as to not provide two .exe...

Yeah, it does not need it now, but earlier I was passing a salt and vector onto the stub which required it to be compiled, now it doesn't, but I just keep it there because I'm not happy with the current encryption method and plan to forge up a new one later.