(https://profile.intra.42.fr/searches) gemerald

(https://profile.intra.42.fr)

Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defences during the Covid crisis allows more flexibility so you can progress into your curriculum, but also brings more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the benefits of the entire community.

SCALE FOR PROJECT FT_SSL_DES (/PROJECTS/42CURSUS-FT_SSL_DES)

You should evaluate 1 student in this team

★

Git repository

```
git@vogsphere.msk.21-school.ru:vogsphere/intra-uuid-5d07537
```

Introduction

In order to maintain high evaluation standards, you are expected to:: Stay polite, courteous, respectful and constructive at every moment of the discussion. Trust between you and our community depends on your behaviour. Highlight the flaws and issues you uncover in the turned-in work to the evaluated student or team, and take the time to discuss every aspect extensively. Please take into account that discrepancies regarding the expected work or functionnalities definitions might occur. Keep an open mind towards the opposite party (is he or she right or wrong?), and grade as honestly as possible. 42's pedagogy only makes sense if peer-evaluations are carried out seriously.

Guidelines

You must grade only what exists in the GiT repository of the student or team. Be careful to check the GiT repository's ownership:: is it the student's or team's repository, and for the right project? Check thoroughly that no wicked aliases have been used to trick you into grading something other than the genuine repository. Any script supposed to ease the evaluation provided by one party must be thoroughly checked by the other party in order to avoid unpleasant situations. If the student in charge of the grading hasn't done the project yet, it is mandatory that he or she reads it before starting the evaluation. Use the available flags on this scale to tag an empty work, a non functional work, a coding style ("norm") error if applicable, cheating, and so on. If a flag is set, the grade is 0 (or -42 in case of cheating). However, cheating case excluded, you are encouraged to carry on discussing what went wrong, why, and how to address it, even if the grading itself is over.

Attachments

subject.pdf (https://cdn.intra.42.fr/pdf/pdf/13245/en.subject.pdf)

Mandatory Part.

The basics. Without mastering them, you are nothing.

Did they extend the executable from the previous project?

When you run `./ft_ssl invalid_param`, does it show an appropriate error message and display the list of available commands? Are those commands the same as required in the previous project? If you run one of the commands, does it behave as it should?

✓ Yes ✗ No

Is BASE64 working?

Test their compiled executable against the `base64` executable. And against OpenSSL. And against itself. You can never be too sure in this line of work. echo "repeat after me 'encoding is not encryption'" | base64 echo "repeat after me 'encoding is not encryption'" | ./ft_ssl base64 diff <(echo -n abc | openssl base64) <(echo -n abc | ./ft_ssl base64) echo "All your base are belong to us. You have no chance make your time." | ./ft_ssl base64 | ./ft_ssl base64 -d

✓ Yes ✗ No

Do they have a flag in their base?

Or like, four of them? Because capture the flag is my favorite game. echo "best game in the franchise" | ./ft_ssl base64 -e -o "halo1" ./ft_ssl base64 -d -i "halo1"

Rate it from 0 (failed) through 5 (excellent)

○

DES-eptively easy!

Does the bare minimum, their ECB mode, work? Is the padding correct? export DES_KEY="C0FFEE69DEADBEEF" echo "foo bar" | ./ft_ssl des-ecb -k "$DES_KEY" | openssl des-ecb -d -K "$DES_KEY" echo "foo bar" | openssl des-ecb -K "$DES_KEY" | ./ft_ssl des-ecb -d -k "$DES_KEY" echo "smol" | ./ft_ssl des-ecb

-k "$DES_KEY" | openssl des-ecb -d -K "$DES_KEY" echo "smol" | openssl des-ecb -K "$DES_KEY" | ./ft_ssl des-ecb -d -k "$DES_KEY" echo "testing the key now" | openssl des-ecb -K "ABCD" | ./ft_ssl des-ecb -d -k "ABCD" echo "More key tests" | openssl des-ecb -K "FFFFFFFFFFFFFFFF" | ./ft_ssl des-ecb -d -k "FFFFFFFFFFFFFFFF" echo "what kind of lock takes no key?" | openssl des-ecb -K "00000000" | ./ft_ssl des-ecb -d -k "00000000"

✓ Yes ✕ No

DES simple flags!

You had to do most of these for base64 so you have no excuse. TESTER! Come up with your best test cases to really thwart their

-a -d -e -i and -o flags! Don't forget, you still have $DES_KEY and base64 to work with!

Rate it from 0 (failed) through 5 (excellent)

○

YOU SHALL NOT PASS(word)!

Does their program handle passwords the same (slimey) way as OpenSSL? echo "$(curl 'https://www.peereboom.us/assl/assl/html/openssl.html')" > original.html echo "password" > password_file \# very secure openssl des-ecb -p -in original.html -out ciphertext.html -pass "pass::$(cat password_file)" ./ft_ssl des-ecb -d -i ciphertext.html -o decrypted.html -p "$(cat password_file)" -s (Copy the salt used by OpenSSL here!) open *.html #/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome *.html # TODO make sure this line above makes it through correctly! You should be presented with three websites. Two of them should be a nice readable rant about the terrors of using OpenSSL. The second should be a nightmare resembling the actual OpenSSL source.

✓ Yes ✕ No

Are they on the Cipher BlockChain?

Does their DES-CBC work? Can they explain what CBC does? echo "Make sure they really tell you what is going on 'under the hood' when you use CBC mode, and how it is more secure than ECB" | openssl des-cbc -K "BABE101010FACADE" -iv "77696E6B66616365" | ./ft_ssl des-cbc -k "BABE101010FACADE" -v "77696E6B66616365" -d Test this also with a password entry against itself and OpenSSL to make sure the IV generation works just as well as the password.

✓ Yes ✕ No

Bonus Part.

The golden rule.

Can they do the DES three times?

Does their Triple DES work in E-D-E mode? Can they explain how it works? Does it compare against OpenSSL, both by direct key input and by password?

✓ Yes ✕ No

Other Block Cipher Modes

Can they explain what a block cipher mode is? Can they explain the differences between the extra modes they added and ECB and CBC? Did they include it also in their 3DES implementation?

Rate it from 0 (failed) through 5 (excellent)

○

Ratings

Don’t forget to check the flag corresponding to the defense

✔ Ok ★ Outstanding project

Empty work Incomplete work No author file Invalid compilation Norme

Cheat Crash Incomplete group ⊘ Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

General term of use of the site (https://signin.intra.42.fr/legal/terms/6)
Privacy policy (https://signin.intra.42.fr/legal/terms/5)
Legal notices (https://signin.intra.42.fr/legal/terms/3)
Declaration on the use of cookies (https://signin.intra.42.fr/legal/terms/2)
Rules of procedure (https://signin.intra.42.fr/legal/terms/4)
Terms of use for video surveillance (https://signin.intra.42.fr/legal/terms/1)