Exercises on Cryptography: intro

CATALDO BASILE

< CATALDO.BASILE@ POLITO.IT >

POLITECNICO DI TORINO

Objectives of the exercises

- improving programming skills & mapping theory into practice
  - the field is very narrow: the cryptography

1. learn how to implement crypto programs in two languages
   - C: the standard for high-efficiency and custom solutions
   - Python: the most used language in the offensive security field (or quick&dirt prototyping)

2. understanding attacks against crypto by implementing them
   - learning how to mount these attacks helps understand crypto best practice as well
     - by seeing how to violate the security properties (they claim to satisfy)
       - ...attacks are interesting *per se*
   - being ready if you have to mount them
   - it’s not penetration testing, but
     - forces a change of perspective
     - it’s a first step towards offensive security in the Cybersecurity Engineering MSc…

The role of Python

implementing complex attacks in C is very time consuming

- …in other words, it’s crazy!

Python is the *de facto* standard for implementing attacks

- for attackers, the rule is “the faster, the better”
  - the sooner you exploit the vulnerabilities, the more you earn
  - ...also valid for CTF players
- Python performance is reasonable in most cases
  - some Python libraries run faster than *not-so-optimized* C code
  - plenty of libraries for attacking purposes
    - **hint**: don’t reinvent the wheel, look for the best library first
- Python proposes a different approach to programming
  - …more Google (or GenAI) dependent
- some students may have not seen Python in their careers
  - ...but we are computer engineers and languages are just languages...

Key takeaways

- competencies in a crucial field of computer system security
  - …in (hope) a less boring way
- an alternative approach to problem-solving
  - “normal” engineers →
    - “from requirements + design + implementation” = constructive approach
  - “attackers” →
    - from implementation + (maybe some requirements and context info) →
      - misuse a system = purposes are different than the ones it was proposed
- ...helpful to complete cybersecurity profiles

- the first step towards approaching the world of the CTFs
  - solve introductory challenges in the crypto area

Exercises classes: the program

"flipped classroom" teaching paradigm

- phase 0: introductory data, explanation of the approach,
  - the study material is provided (slides and videos)
- phase 1: access the material and study yourself
- phase 2: interact with your colleagues and me to solve issues
- phase 3: face-to-face classes to consolidate learning results
  - check that the level of preparation is enough to pass the exam
    - solve more complex exercises and use the knowledge you studied
    - solve typical exam exercises
  - tools will help make lectures more interactive
- I am evaluating other online support tools
  - to improve interactions among students and with me
  - Slack experience is, in general, poor in the classes

Why flipped classroom?

on paper... this approach grants better results

- higher success rate, better level of knowledge, competencies

not all the students appreciated it... [*quotes from the CPD questionnaires*]

  - classes are not all the weeks / want to have more constant contact with the teacher / more continuous classes
  - it was a completely failed experiment (*...well that was too much* ☺)

nonetheless, stats from the last two years contradict these claims

- a higher number of students were able to provide solutions to the exercises
  - 15% → 9% reduction in blank answers (2022)
- students got higher scores for the exercises' questions
  - C exercises are usually easier and standard, the improvement was minor
  - improvement was more consistent for the Python part
    - despite the last two years' exercises being more complex
  - even better last year with the CTF

Main topics

Part I (week1-week5)

- C programming with OpenSSL
  - symmetric and asymmetric crypto primitives, hashes, and MACs
    - then build more complex protocols based on the primitives

Part II (week6-week13)

- Python programming and attacks
  - Python basics: symmetric and asymmetric crypto primitives, hashes, MACs, servers, connections
  - Attacks:
    - symmetric crypto
      - block ciphers (ECB mode, CBC mode, ...)
      - stream ciphers (keystream reuse, statistical attacks, ...)
      - hashes (collisions, length extension, ...)
    - asymmetric crypto
      - RSA (factorization, primes/modules, decryptions) + some theory for the most advanced ones

Cryptography CTF

- two years ago, the CryptoCTF was added
  - https://cryptoctf.m0lecon.it/

- complement the material with practical exercises
  - conceived to play with the course topics
  - it’s not intended to measure your absolute strength to attack crypto
    - you can play real CTFs to measure this!

- divided into two parts
  - PART 1: C programming challenges
    - write a C program able to generate the required output → i.e., print the flag
      - select pieces from the solution and hash them to obtain the flag
        - according to some rules
  - PART 2: attacks (to be implemented in Python)
    - different attacks against (symmetric, asymmetric) crypto
      - if you can successfully mount the attack, you’ll get the flag

This is the CTF of the Cryptography course 2024

Cryptography CTF

last year the CryptoCTF was officially recognized

- incentives for people playing the CTF:
  - 3 pt if you complete all the intended challenges before the deadline
    - (1pt) C programming
    - (1pt) symmetric crypto attacks
    - (1pt) asymmetric crypto attacks
  - 2 pt bonus to students who solve the last challenges and have completed the great majority of the exercises (to be selected based on the CTF scoreboard)
- the part of the exercise grade will be computed as
  - *(exam score + bonus) mod 12*
    - but *max(exam score + bonus) = 14.5*

Cryptography2024
Not secure 130.192.5.212:8080
All Bookmarks
Cryptography2024 Users Scoreboard Challenges
CTFd
Politecnico di Torino
1859
Powered by CTFd

This is the CTF of the Cryptography course 2024

Cryptography CTF

- some comments
  - C challenges are not fully compliant with the *CTF best practice*
    - the objective is to “invite” you to write C code
  - Python challenges are much better
    - some will be basic, some medium-level, some a bit more advanced
      - the latter may require personal study effort and may cover more than is required to pass the exam
- **the CryptoCTF is OPTIONAL**
  - you can pass the exam studying as you did for all other exams
    - and also reach *30 e lode*
  - playing the CTF may require more hours than usually associated with the CFUs
    - but don’t complain with me if you discover CTFs are highly addictive

This is the CTF of the Cryptography course 2024

Anti-cheating

last year a non-negligible portion of the students cheated

- easy to ask your friends to pass the FLAGS
  - I would like to trust you, not to implement methods to randomize strings and detect cheaters...

...but... students will be randomly selected to perform a quick verification

- a few minutes of discussion with me (last year on a VC)
- you can collaborate with your colleagues, and you are encouraged to do so
  - especially when they are more experts than you
- but, in the end, you need to understand what they did / what the solve does
- ***Remember****: the purpose is being ready for the exam in June/July*

if you didn't understand the attack, don't submit the flag

if during the verification it's clear the you didn't understand the solve for flags

- **your final score will be -5** and you will not be allowed to continue with the CTF
- I will evaluate sending names of the cheating students to the **Commissione Disciplinare**

this year **a larger portion of students** will be involved in the check

Anti-cheating

last year a non-negligible portion of the students cheated

- easy to ask your friends to pass the FLAGS
  - I would like to trust you, not to implement methods to randomize strings and detect cheaters, but...

stud

- a f
- yo
  - e
- bu

The CTF score is in the range [-5,+5]

- ***Remember***: *the purpose is being ready for the exam in June/July*

if you didn’t understand the attack, don’t submit the flag

if during the verification it’s clear the you didn’t understand the solve for flags

- **your final score will be -5** and you will not be allowed to continue with the CTF
- I will evaluate sending names of the cheating students to the **Commissione Disciplinare**

this year **a larger portion of students** will be involved in the check

More resources for your study

github of the course

- learn how to move inside it
  - as this will be accessible during the exam
  - https://github.com/aldobas/cryptography-03lpyov-exercises

manuals and reference documentation

- OpenSSL, Python libraries for crypto and attacks

you should not remember by heart the function prototypes and parameters

- the examples and the public manuals are there for this reason

more tools are under investigation

- slack/discord
- wooclap/moodle

Environment for the exercises

- reference architecture: **Kali Linux 2024.3+**
  - VM available for most hypervisors
    - https://www.kali.org/get-kali/#kali-virtual-machines
  - or install on multi-boot (do you really want to do this in 2024?)
    - https://www.kali.org/get-kali/#kali-bare-metal
    - or live (discouraged unless you really want to use persistence)
    - https://www.kali.org/get-kali/#kali-live
- the Python 3 interpreter
    - additional packages will be proposed and added using *pip install*
- OpenSSL and OpenSSL for developers
  - install from sources or Linux repositories
- **WARNING**: you may also want to use Windows, MACs, etc., but exercises will not be tested on these platforms (Windows+WSL2 should work)
  - everything "should" work, but if it does not, you have to solve issues yourself…
    - I use VSC as an editor + gcc command line from a shell for C programs
    - VSC and PyCharm for Python