Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Online Examination
PeCoH Deliverable D4.4

Nathanael Hübbe, Julian Kunkel

Work Package:	WP4
Responsible Institution:	DKRZ
Date of Submission:	November 2019

Chapter 1

Introduction

This white paper covers the deliverable D4.4 of the PeCoH project.

In this deliverable, we also describe the prototypical process to facilitate online examination of the certificates based on the skill tree developed within the project. Detailed emphasis is given to the code written. Most importantly, we describe the overall architecture and security concept of the code, and we analyze what guarantees this security concept delivers.

Chapter 2

Goal and Threat Model

The aim of the online examination is to attest users their knowledge of HPC skills within their browser and to then being able to produce certificates. The utilization of the browser for conducting the online test in a low-entry-threshold way was decided as the certificates should be accessible to anyone worldwide and free of charge.

Trust in the generated certificates, however, is important. A certificate attesting that a user has a certain skill is only as good as the trust that the user actually possesses this skill. From the technical side, it is impossible to control what the user does with a browser-based test, how they come up with their answers, and how they send their results back to be graded. Most importantly, it is impossible to detect whether a user actually is who they claim to be, anyone can hire anyone else knowledgeable enough to take the test for them. But even a genuine user may have acquired a certificate long time ago using a nearly temper-proof exam but now forgotten most about the skill as it wasn't trained.

We discussed with peers and inspected existing solutions that aim to mitigate cheating attempts, but found that no solution (even physical presence) doesn't guarantee genuine results. Therefore, we approach this problem pragmatically using a combination of technology and process strategy.

2.0.1 Process Strategy

Potential cheating attempts are mitigated as follows:

- Creation of a big question pool and drawing a random subset for a test. Each question contains more answers than shown.
- Users must register (temporarily) in order to attempt a certificate, the name/date printed on the certificate will match the registration.
- Certificates can be technically verified.
- Skill verification, e.g., by providing the technical information of the certificates a quick test can be performed.

Particularly, skill verification is suitable to verify someones knowledge but also your own knowledge. In order to ensure that the skills of a candidate are genuine, e.g., a interview panel can request to generate a quick test (e.g., 5 min) for all given certificates. The candidate then has to fill the test and the system shows the achieved score (and likeliness of possessing the skills) for the candidate after the test.

2.0.2 Technical Solution

There are quite a few things that can be controlled, and that should be controlled in order to give the resulting certificates some credibility. The first threat to be addressed is the use of scripting to extract the internal database of questions and their respective correct answers. Obviously, to protect this database, it must not be transferred to the user's browser at all. The questions for each user must be transferred, but the correct answers must be kept secret. To make it hard to launch differential attacks against the answer catalogue, the user must not be in control of the set of questions they have to answer. It must be possible to detect whether a user has modified the list of questions on their test, and such erroneous submissions must be rejected by the server.

It must also be impossible to automatically take many tests in a short time frame, answering them randomly, and building a catalogue of questions and correct answers by differential analysis. To this end, some kind of rate limiting must be present, and no precise grade should be returned to the users. All the user should be able to see is, whether they passed the test or not. As it is not the goal of the HPC certification to train the user, information about correct or incorrect answers are hidden to the user.

Chapter 3

Architecture of our Solution

To address the various issues described in the previous section, we decided to make a firm separation between the server and the client code, ensuring that no trust is given to the client code.

The solution consists of a frontend allowing users to register for a test, then the technical solution to perform the actual test and submit the solution.

From the user perspective, the process is as follows:

- A user register for conducting the examination; therefore, enters personal information into a form and submit it.
- The server will send an email with a link to the personal examination. This link is valid for 24 hours and contains in encrypted form all details provided by the user. Hence, the server does not store any information, yet.
- Once the link in the email is clicked, it will lead to a webpage with the examination questions, a time limit for the particular examination will be set, e.g., 60 minutes using the tool as described below. The server will also store the user information in a temporary database to indicate when the test was started.
- The user submits the test. The test will be automatically marked.
- We will verify the results and generate a certificate. The certificate will include a unique URL allowing anyone to verify the correctness of the certificate.

In case a user doesn't pass the test, s/he can repeat the test after a cooldown period. At the moment, we consider two weeks a suitable time. This is tracked by the temporary database, once a user starts a test, the information is added to the database. The outcome of the test will be added to the database once the user submits the test. In case, the user fails, the entry remains in the database for two weeks and is then removed allowing to retry it.

3.0.1 Registration for a Test

First, a user has to select a certificate to be tested, then a disclaimer is presented that also contains information about the process:

Disclaimer

Privacy statement

The certificate

The certificate will show your name and affiliation, the month the exam was conducted and a description of the particular certificate. It will also include a unique URL that can be given to a third-party allowing it to verify that you have obtained the certificate. It will be produced as a PDF and transferred via email to you.

Data recorded

To generate a meaningful certificate, we need the following personal information: **name**, **affiliation** and **email address**. On the server side, we will also attach the **date** the test was submitted to this metadata. When you interact with the webserver, the typical server logs will be recorded as stated in the privacy statement. These data will not be correlated to your personal information or the tests submitted here.

Data processing

We will use your personal data and the score to generate a certificate. This process may involve manual marking and approval of a staff member and the automatic generation of the certificate.

Your name and email is used solely to contact you and send you the test results and certificate (in case of your failure, we will send you some information that allows the discussion of your test results). We will not disclose your personal information, the results, or the generated certificate with any third party. In case of a successful examination, this information will be deleted from the server immediately after the results are processed and the email has been sent to you. In case of an unsuccessful examination, to prevent re-examination within a cooldown period, we will store a hash of your name together with the date of the examination for a duration that is the resit period of the individual certificate. After that duration, we will delete the information such that you can be re-examined.

We will record the information of (date, affiliation), i.e., how many users with a certain affiliation have obtained a certain certificate. We will use this database of affiliations and date to promote the certification program (e.g., by mentioning that someone from an affiliation have obtained a certain certificate). This data will **not** contain your name, email address, or the score.

We will record the achieved score separately from any personal information with the answered questions for the purpose to optimize the examination.

Terms of this service

We will take industry-typical precautions to prevent any cybercrime including theft of your personal data while your data is stored on an IT system involved in the data processing. However, in case of any data loss, theft of the data provided, we do not take liability for data loss or thefts caused by cybercrime. Note that includes cases in which minor careless actions of staff enabled third-parties to steal or compromise the information.

Process

1. Once you are ready, you register for conducting the examination; therefore, enter your personal information into this form and submit it.
2. We will immediately send you an email with a link to your personal examination. This link is valid for 24 hours. *Check your spam folder if it does not arrive!*

3. The link will lead to a webpage with your examination questions, you have a **time limit for this particular examination of 60 minutes**.
4. We will mark your exam and send your results or your certificate (typically, this takes a week).
5. The certificate will include a unique URL allowing anyone to verify that you obtained the certificate and the date.

Examination for "*The basic certificate*"

This multiple-choice test contains questions to obtain the certificate "*The basic certificate*" with the ID 4711.

(this is just a demonstration page, here we will normally find a description about the certificate)

Time limit: 60 minutes

Score to pass: 70%

Name		(will be on the certificate)
E-mail		(will be used to send you the certificate / the results)
Affiliation		(will be used for statistical purposes)

I agree to the privacy policy and terms of this service ☐

Press **Register for this examination** to transfer your request for examination together with the provided **personal information** to the server.

Register for this examination!

3.0.2 Conducting the Test

The steps for performing a test are as follows:

- The client requests an instance of the test from the server, and the server returns an individualized test to the client in JSON format. This individualized test includes only the questions, no answers. It also contains a number of values for security checking:
  - A nonce that was used to generate the question selection.
  - A cryptographic hash of the concatenation of server secret and the question catalogue from which the questions were selected.
  - A timeout stamp in UTC.
  - Any other test generation parameters like the number of questions.
  - A signature on the tuple (Nonce, Hash, Timeout). This signature is a cryptographic hash computed using a secret value which does not leave the server, ensuring that only the server itself can check the signature.
- The client Javascript code presents the questions to the user, allowing them to answer, records the answers, and adds them to the individualized test to form a submission.
- The submission is sent back to the server.
- The server checks its signature of the Nonce, Hash, and Timeout fields, rejecting the submission if these have been tampered with.
- The server checks the timeout, rejecting the submission if it was not handed in in a timely manner.
- The server uses the Nonce and Hash to recreate the individualizes test from scratch.
- The recreated individualized test is compared to the questions in the submission, and any discrepancy leads to a rejection of the submission.
- Finally, if all the checks above have succeeded, the server actually checks the correctness of the answers and calculates a grade.

3.1 Security Aspects

Because of the signature that must be included in the submission, the client cannot successfully hand in submissions to individualized tests that were not generated by the server itself. And, even though the server does not store any information about the individualized tests it creates, the signed timeout allows it to reject submissions that were not handed in within the provided time frame.

There is still a danger of a client acquiring a complete list of exam questions through repeated requests for individualized tests. Thus, if the hash of the question catalogue did not include the server secret and did not include enough questions, it would be possible to brute force the answers using the question catalogue hash in an offline brute force attack. The inclusion of the server secret ensures that an offline attacker cannot

compute the effective hash function that is used to identify the question catalogue. To request the questions, a user has to register with the name and email which provides some level of protection against brute-force attempts. The way we have implemented it, it is possible to acquire the complete list of questions, but impossible to deduce the correct answers to the questions.

3.2 Components

There are four software components in our solution:

- A `bash` script called `makeCatalogue` to create a hashed question catalogue. This basically walks a hierarchy of directories and concatenates all the questions file it finds within.
- A `python` program called `examiner` that is both used to create individualized tests from a hashed question catalogue, and to check/grade submissions.
- A `PHP` script `examiner.php` that provides a web-interface for the `examiner`.
- A `Javascript` program `testGui.js` that can be included in a web page to provide the front end where the user can enter their answers and submit their test.

The `examiner` is the only component in this architecture that serves in two very distinct roles. However, these two roles both require creating an individualized test from a question catalogue and a Nonce, which is actually the main work of checking the integrity of a test submission. As such these two functions were combined in one program to avoid code duplication.

3.3 Question Format

The question catalogue is a set of questions with possible answers. Currently, all these questions are "select multiple" questions, where each question defines a number of correct and incorrect answers, from which a random subset is selected.

Each answer possibility is displayed as a checkbox and has a likelihood of exactly 50% to be correct. This 50% likelihood is independent of other answer possibilities of the same question, and is enforced by the test generation algorithm. As such, if a question has five possible answers displayed, and the user knows that four are correct, the fifth answer still has exactly a 50% chance of being correct. Both, the case that no answer is correct and that all answers are correct, are equally likely, and occur with a probability of 1/32 for a five answer question. This ensures that users cannot guess at a correct answer because the other answer possibilities are clearly wrong, or vice versa.

It may be beneficial to expand the number of possible question formats in the future. But for now, the "select multiple" format seems quite effective at providing meaningful test results.

Chapter 4

State of the Implementation

The `makeCatalogue` script and the `examiner` program are fully functional for handling of "select multiple" questions. The `testGui.js` implements basic functionality but would require significant honing to make it pleasing to use. A test deployment was made on `https://www.hpc-certification.org/examiner/` that demonstrates the overall process.

What is still missing, is a large number of good test questions to feed the system. We created an initial repository of questions that are now managed by the HPC Certification Forum. As these questions and answers are sensitive information, they cannot be revealed to the wider public.

Chapter 5

Summary

We have designed a concept for online examinations. This concept addresses a number of hard questions concerning the security and meaningfulness of online exams. We believe that we found good solutions to some of these problems, and designed a workable concept. This concept has been implemented, and was found to be effective. A test deployment was made on `https://www.hpc-certification.org/examiner/` but for the acceptance of a certificate, a good question catalogue must be created.

In the future, we will expand the system to generate the short questionnaires that can be used to verify skills.

Acknowledgement

The PeCoH project has received funding from the German Research Foundation (DFG) under grants LU 1353/12-1, OL 241/2-1, and RI 1068/7-1.