PRACTICE QUESTIONS

EC-Council

Certified Offensive AI Security Professional (COASP)

PUBLISHED BY aiexamcerts

EDITION	Free Set 1
QUESTIONS	10
ISSUED	November 8, 2026
EXAM	EC-Council Certified Offensive AI Security Professional (COASP)

Shared by **aiexamcerts** on aiexamcerts

10 practice questions for the EC-Council Certified Offensive AI Security Professional (COASP) exam, each followed by its answer and a short explanation. More free sets and the full practice exam: **aiexamcerts.com/ec-council/certified-offensive-ai-security-professional-coasp**. Spotted a mistake? Email support@aiexamcerts.com with the exam name and the question number.

---

1. What is the key difference between a white-box and a black-box assessment of an AI system?

- **A** White-box testing runs only after deployment, while black-box testing runs only during training
- **B** White-box testing gets internal details, black-box testing sees only the user interface
- **C** Black-box testing is used only for models hosted on premises, because the vendor hides hardware
- **D** White-box testing cannot find real flaws, because the tester already knows how the code works

Answer: B

White-box assessments give the tester internal knowledge, which allows deeper analysis, while black-box assessments reflect what an outside user or attacker can reach. The option claiming that white-box testing cannot find real flaws is the most tempting misconception, but visible internals often expose weaknesses that black-box work would miss. Timing and hosting claims mix up when and where testing happens with how much knowledge the tester has.

---

2. Publishing a model's exact version and full system prompt in public documentation carries no security risk, since users can already see the chat output.

- **A** True
- **B** False

Answer: B (False)

A system prompt can reveal guardrail design, tool names, and internal business rules, and an exact model version points attackers to known weaknesses for that build. Chat output shows answers, not the hidden configuration, so comparing it to published text does not hold. Publishing these details should be a deliberate decision, not an assumption that they are harmless.

---

3. A fuzzer run against a text-to-SQL feature produces generated statements that delete rows. Which control addresses the root cause?

A Run generated statements only with read-only credentials in a sandbox, with an operation allowlist in place

B Increase the fuzzer's speed so more statements are tested before a deletion occurs

C Move the database to a different storage vendor so the deletions are harder to repeat in production

D Turn off logging of generated statements so the deletions do not show up in reports

Answer: A

Model output is untrusted input to the database, so least-privilege credentials and an allowlist of operations stop generated statements from doing harm. Speeding up the fuzzer only finds the same problem faster, and it is the most tempting response because it looks like progress. Changing vendors does not change what the feature is allowed to execute, and hiding statements from logs removes the evidence the team needs.

---

4. A web page that an assistant fetches for a user contains hidden text telling the assistant to pass along the user's chat history. Which category does this describe?

A Direct prompt injection typed by the account owner into the chat box of the assistant interface

B Model extraction carried out through repeated API queries

C Indirect prompt injection, where instructions arrive through content the model processes

D Training data poisoning performed during a fine-tuning run

Answer: C

Indirect prompt injection hides instructions in content the model reads on the user's behalf, such as web pages, email, or documents, so the attacker never talks to the model directly. Direct prompt injection is the tempting label, but in that case the person typing the prompt supplies the instructions. Model extraction and training-data poisoning are different categories that act on outputs or the training process.

---

5. A model inversion attack tries to reconstruct representative examples of the _____ data from the model's outputs.

Answer: training

Model inversion recovers features or representative samples of the data a model learned from, which can expose sensitive attributes. It differs from membership inference, which only asks whether a known record was included. Limiting output detail and training with differential privacy are the usual mitigations.

---

6. Changing the labels of a small set of training examples so that one class is systematically misclassified is an example of label _____.

Answer: flipping (also accepted: flip)

Label flipping alters the ground truth attached to selected records, which pushes the model toward wrong decisions for targeted inputs. It differs from clean-label poisoning, where the labels stay correct and the content is crafted. Auditing label distributions against an approved version helps detect it.

---

7. A model is used as an automated judge that scores outputs from another model. Which risk applies to the judge itself?

A The judge runs out of memory whenever it reads short answers from the other model

B The judge's scores are stored in the wrong time zone in the evaluation log

C The scored output can contain text crafted to manipulate the judge's verdict

D The judge refuses to score any output that mentions a product name in passing

Answer: C

A judge reads candidate output as input, so adversarial text in that output can steer the verdict, which is a manipulation risk at the scoring step. Memory limits and time zones are operational issues, not security risks of the judging design. Refusing to score outputs that mention a product name is a behavior problem, and it is the most tempting thing to notice, but it is not the manipulation risk the question describes.

---

8. What is an AI bill of materials (AIBOM) most useful for during a vulnerability disclosure?

A Calculating the average inference cost per request for the finance team's budget

B Replacing code review for the model-serving code with a signed list of approvals

C Finding every deployment that uses an affected model, dataset, or library

D Storing production API keys in one shared file so teams can rotate them together

Answer: C

An AIBOM records the models, datasets, libraries, and versions a system depends on, so affected components can be found quickly when a flaw is disclosed. Cost reporting is the most tempting use, but it is a financial metric and not supply chain visibility. Replacing code review and storing keys in a shared file do not make components traceable, and the second creates new exposure.

---

9. What does the risk-based approach of the EU AI Act imply for security testing of AI systems?

A Higher-risk systems face stricter requirements, including risk management and testing

B All AI systems are banned unless their source code is published openly

C Only systems that run on GPUs fall under the regulation at all

D Security testing is optional for every category, because the Act focuses mainly on product labeling

Answer: A

The Act sets obligations by risk tier, so systems in higher-risk categories need more rigorous risk management and testing than lower-risk ones. Banning systems unless their code is open is the most extreme misreading, and the Act does not work that way. Hardware choice and labeling-only enforcement are equally inaccurate descriptions of its scope.

---

10. Which evidence best helps reconstruct how a poisoned document entered a retrieval index?

A The model's temperature setting at the moment the affected answer was produced for that user

B The number of tokens in the final answer shown to the user

C The browser version the support agent used on the day the ticket was opened

D Ingestion logs with the adding account or pipeline, timestamps, and the content hash

Answer: D

Ingestion logs and a content hash tie the document to a writer, a time, and an exact version, which is what an investigator needs to trace the entry path. Temperature is the most tempting setting to inspect, but it affects variation in wording and not how a document got into the index. Token counts and browser versions do not show how content was written into the system.