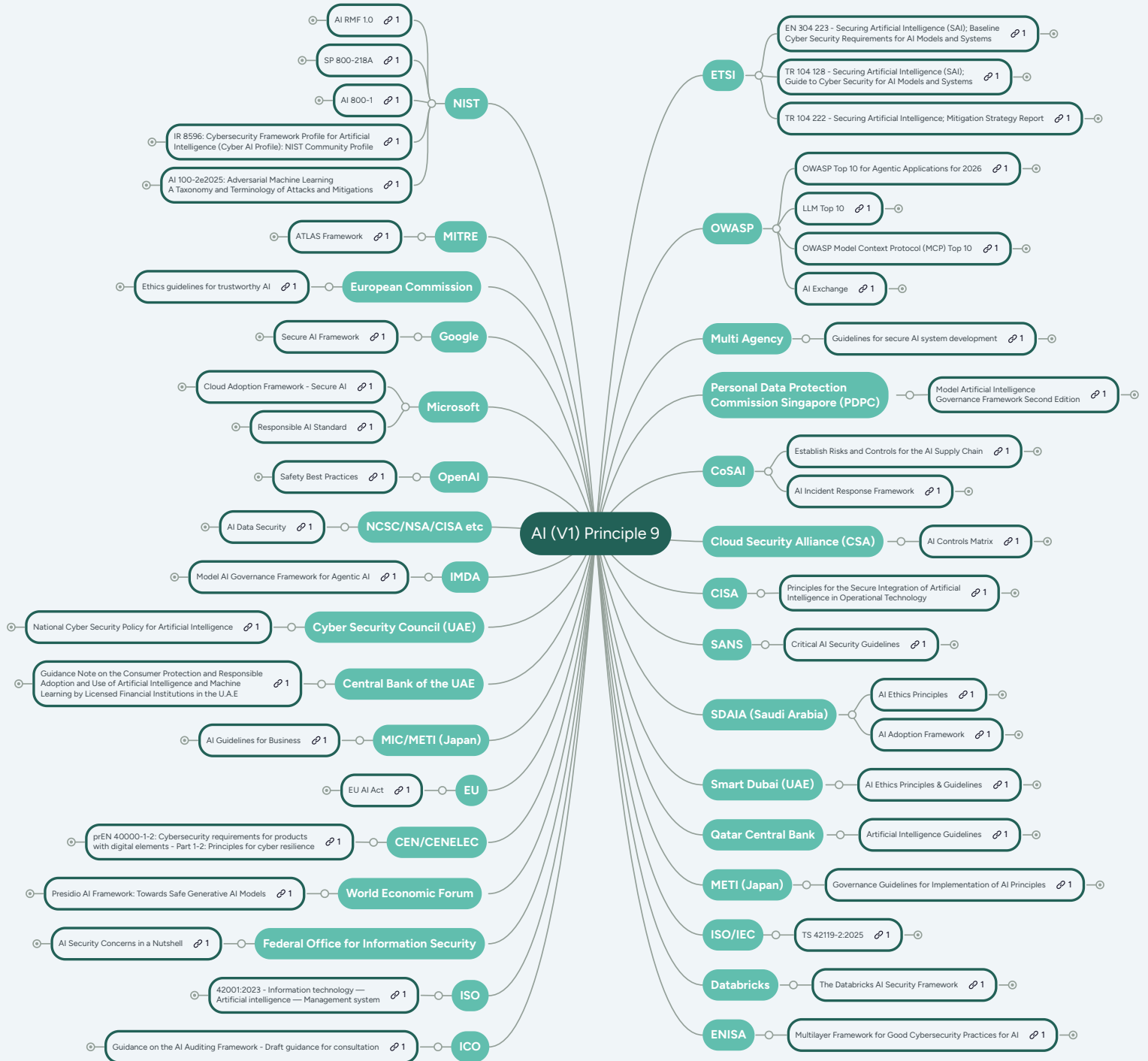




AI (V1) Principle 9

1. ETSI

1.1. EN 304 223 - Securing Artificial Intelligence (SAI); Baseline Cyber Security Requirements for AI Models and Systems

Link: https://www.etsi.org/deliver/etsi_en/304200_304299/304223/02.01.01_60/en_304223v020101p.pdf

Link: https://www.etsi.org/deliver/etsi_en/304200_304299/304223/02.01.01_60/en_304223v020101p.pdf

1.1.1. Provision 5.2.5-1

Developers shall ensure that all models, applications and systems that are released to System Operators and/or End-users have been tested as part of a security assessment process.

1.1.2. Provision 5.2.5-2

System Operators shall conduct testing prior to the system being deployed with support from Developers.

1.1.3. Provision 5.2.5-2.1

For security testing, System Operators and Developers should use independent security testers with technical skills relevant to their AI systems.

1.1.4. Provision 5.2.5-3

Developers should ensure that the findings from the testing and evaluation are shared with System Operators, to inform their own testing and evaluation.

1.1.5. Provision 5.2.5-4

Developers should evaluate model outputs to ensure they do not allow System Operators or End-users to reverse engineer non-public aspects of the model or the training data.

1.1.6. Provision 5.2.5-4.1

Additionally, Developers should evaluate model outputs to ensure they do not provide System Operators or End-users with unintended influence over the system.

1.2. TR 104 128 - Securing Artificial Intelligence (SAI); Guide to Cyber Security for AI Models and Systems

Link: https://www.etsi.org/deliver/etsi_tr/104100_104199/104128/01.01.01_60/tr_104128v010101p.pdf

Link: https://www.etsi.org/deliver/etsi_tr/104100_104199/104128/01.01.01_60/tr_104128v010101p.pdf

1.2.1. Provision 5.2.5-1

"Developers shall ensure that all models, applications, and systems that are released have been tested as part of a security assessment process." (ETSI TS 104 223 [i.1])

Related threats/risks:

Without rigorous security assessments by qualified experts understanding AI and classic risks, AI systems can be vulnerable to exploits, unauthorized access, or data breaches, leading to potential data loss, manipulation, or misuse of the AI model.

Example Measures/Controls 1:

Implement Security Assessment Processes for All Releases: Establish a mandatory security assessment process for all models, applications, and systems prior to release, covering areas like access control, data integrity, and adversarial AI attacks. Scope testing based on the threats identified in threat models.

Example Measures/Controls 2:

Use Offensive Security Assessments with Penetration Testing and Red Teaming: Use periodic penetration testing to evaluate the AI system's resilience and red teaming for models.

1.2.2. Provision 5.2.5-2

"System Operators shall conduct testing prior to the system being deployed with support from Developers." (ETSI TS 104 223 [i.1])

Related threats/risks:

Without pre-deployment testing, systems can fail to meet operational and security requirements, leading to compromised performance, reduced reliability, or security vulnerabilities once deployed.

Example Measures/Controls:

Implement Comprehensive Pre-Deployment Testing: Complement development-time testing with running benchmarking emulation suites covering functional, performance, and security tests to confirm that the system meets intended requirements before deployment. These can either be in-house or independent third-party benchmarks. Integrate these tests into the development pipeline to ensure issues are identified and resolved before release.

1.2.3. Provision 5.2.5-2.1

"For security testing, System Operators and Developers should use independent security testers with technical skills relevant to their AI systems." (ETSI TS 104 223 [i.1])

Related threats/risks:

Without independent security testing, systems can overlook critical vulnerabilities, especially those requiring specialized expertise, increasing the risk of undetected flaws and potential exploitation.

Example Measures/Controls:

Engage Independent Security Testers for Pre-Deployment Testing: Use independent security testers with AI expertise to validate security measures, providing an unbiased review of system security.

1.2.4. Provision 5.2.5-3

"Developers should ensure that the findings from the testing and evaluation are shared with System Operators, to inform their own testing and evaluation." (ETSI TS 104 223 [i.1])

Related threats/risks:

Without access to previous testing findings, System Operators can lack critical insights into known vulnerabilities or limitations, and mitigations that cannot be seen as adequate by operators leading to potential gaps in security coverage.

Example Measures/Controls:

Establish a Process for Sharing Testing Results: Create a standardized process for sharing all relevant testing results and evaluation findings with System Operators of testing reports and ensure timely access for all relevant stakeholders.

1.2.5. Provision 5.2.5-4

"Developers should evaluate model outputs to ensure they do not allow System Operators or End-users to reverse engineer non-public aspects of the model or the training data." (ETSI TS 104 223 [i.1])

Related threats/risks:

Without adequate output evaluation, Operators or End-users can reverse engineer model internals or correlate outputs with sequences of predesigned inputs, enabling them to reconstruct the model or training data. This can erode commercial advantage, allow the creation of shadow models, or facilitate attacks, even affecting "open source" models if only parts, such as architecture or weights, are released.

Example Measures/Controls 1:

Conduct Security Reviews for Output Sensitivity: Engage with AI experts to evaluate model outputs and identify any information that could reveal internal model structures, ensuring sensitive aspects remain non-public.

Example Measures/Controls 2:

Integrate Adversarial Testing and Robustness Evaluation: Implement adversarial testing to assess the resilience of AI systems against attempts to reverse engineer or manipulate model outputs. This involves simulating attacks that exploit output data to uncover model weaknesses or extract sensitive information. Open-source tools like ART and TextAttack can help to develop and run these tests.

1.2.6. Provision 5.2.5-4.1

"Additionally, Developers should evaluate model outputs to ensure they do not provide System Operators or End-users with unintended influence over the system." (ETSI TS 104 223 [i.1])

Related threats/risks:

Without evaluating for unintended influence, Operators or End-users can exploit system behaviour to align outputs with personal or malicious goals, leading to bias, misuse, or compromised trust in the system.

Example Measures/Controls:

Implement Safeguards Against Manipulation: Set controls to prevent Operators or End-users from adjusting inputs in ways that could intentionally influence the AI system's outcomes.

1.3. TR 104 222 - Securing Artificial Intelligence; Mitigation Strategy Report

Link: https://www.etsi.org/deliver/etsi_tr/104200_104299/104222/01.02.01_60/tr_104222v010201p.pdf

Link: https://www.etsi.org/deliver/etsi_tr/104200_104299/104222/01.02.01_60/tr_104222v010201p.pdf

1.3.1. 6.2.2 - 8

Adversarial testing is closely related to adversarial training described earlier. Adversarial testing consists of testing the robustness of a trained model with respect to adversarial examples. These attacks can be created using a variety of methods, in particular projected gradient descent. Adversarial testing allows quantifying the robustness of a model to the considered attacks. If the testing results show that the robustness of the model is low, the model can be hardened more thoroughly, using adversarial training.

2. OWASP

2.1. OWASP Top 10 for Agentic Applications for 2026

Link: <https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/>

Link: <https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/>

2.1.1. ASI01: Agent Goal Hijack - 8

Conduct periodic red-team tests simulating goal override and verify rollback effectiveness.

2.1.2. ASI05: Unexpected Code Execution (RCE) - 2

Prevent direct agent-to-production systems and operationalize use of vibe coding systems with pre-production checks: including the guidelines of this entry with security evaluations, adversarial unit tests and detection of unsafe memory evaluators.

2.1.3. ASI05: Unexpected Code Execution (RCE) - 3

Ban eval in production agents: Require safe interpreters, taint-tracking on generated code.

2.1.4. ASI08: Cascading Failures - 9

Digital twin replay and policy gating: Re-run the last week's recorded agent actions in an isolated clone of the production environment to test whether the same sequence would trigger cascading failures. Gate any policy expansion on these replay tests passing predefined blast-radius caps before deployment.

2.2. LLM Top 10

Link: <https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/>

Link: <https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/>

2.2.1. LLM01: Prompt Injection - 3

Define sensitive categories and construct rules for identifying and handling such content. Apply semantic filters and use string-checking to scan for non-allowed content. Evaluate responses using the RAG Triad: Assess context relevance, groundedness, and question/answer relevance to identify potentially malicious outputs.

2.2.2. LLM01: Prompt Injection - 7

Perform regular penetration testing and breach simulations, treating the model as an untrusted user to test the effectiveness of trust boundaries and access controls

2.2.3. LLM03: Supply Chain - 3

Apply comprehensive AI Red Teaming and Evaluations when selecting a third party model. Decoding Trust is an example of a Trustworthy AI benchmark for LLMs but models can finetuned to by pass published benchmarks. Use extensive AI Red Teaming to evaluate the model, especially in the use cases you are planning to use the model for.

2.2.4. LLM04: Data and Model Poisoning - 8

Test model robustness with red team campaigns and adversarial techniques, such as federated learning, to minimize the impact of data perturbations.

2.2.5. LLM10: Unbounded Consumption - 2

Restrict or obfuscate the exposure of `logit\_bias` and `logprobs` in API responses. Provide only the necessary information without revealing detailed probabilities.

2.3. OWASP Model Context Protocol (MCP) Top 10

Link: <https://owasp.org/www-project-mcp-top-10/>

Link: <https://owasp.org/www-project-mcp-top-10/>

2.3.1. MCP02:2025 - Privilege Escalation via Scope Creep - 5

Automated Entitlement Reviews & Drift Detection Periodically (and on change) run entitlement audits to find scope expansions. Alert on permission increases and requires a documented justification and approval.

2.4. AI Exchange

Link: https://owaspai.org/docs/ai_security_overview/

Link: https://owaspai.org/docs/ai_security_overview/

2.4.1. 1.3. Controls to limit the effects of unwanted behaviour - CONTINUOUS VALIDATION

Continuous validation: by frequently testing the behaviour of the model against an appropriate test set, it is possible to detect sudden changes caused by a permanent attack (e.g. data poisoning, model poisoning), and also some robustness issues against for example evasion attacks.

Continuous validation is a process that is often in place to detect other issues than attacks: system failures, or the model performance going down because of changes in the real world since it was trained (model drift, model staleness). There are many performance metrics available and the best ones are those that align with the goal. These metrics pertain to correctness, but can also link to other aspects such as unwanted bias towards protected attributes.

Note that continuous validation is typically not suitable for detecting backdoor poisoning attacks, as these are designed to trigger with very specific input that would normally not be present in test sets. In fact, such attacks are often designed to pass validation tests.

2.4.2. 1.3. Controls to limit the effects of unwanted behaviour - UNWANTED BIAS TESTING

Unwanted bias testing: By doing test runs of the model to measure unwanted bias, unwanted behaviour caused by an attack can be detected. The details of bias detection fall outside the scope of this document as it is not a security concern - other than that, an attack on model behaviour can cause bias.

3. Multi Agency

3.1. Guidelines for secure AI system development

Link: <https://www.ncsc.gov.uk/files/Guidelines-for-secure-AI-system-development.pdf>

Link: <https://www.ncsc.gov.uk/files/Guidelines-for-secure-AI-system-development.pdf>

3.1.1. Release AI responsibly

You release models, applications or systems only after subjecting them to appropriate and effective security evaluation such as benchmarking and red teaming (as well as other tests that are out of scope for these guidelines, such as safety or fairness), and you are clear to your users about known limitations or potential failure modes. Details of open-source security testing libraries are given in the further reading section at the end of this document.

4. Personal Data Protection Commission Singapore (PDPC)

4.1. Model Artificial Intelligence Governance Framework Second Edition

Link: <https://www.pdpc.gov.sg/-/media/files/pdpc/pdf-files/resource-for-organisation/ai/sgmodelaigovframework2.pdf>

Link: <https://www.pdpc.gov.sg/-/media/files/pdpc/pdf-files/resource-for-organisation/ai/sgmodelaigovframework2.pdf>

4.1.1. Data for Model Development - d) Different datasets for training, testing, and validation

Different datasets are required for training, testing, and validation. The model is trained using the training data, while the model's accuracy is determined using the test data. Where applicable, the model could also be checked for systematic bias by testing it on different demographic groups to observe whether any groups are being systematically advantaged or disadvantaged.

Finally, the trained model can be validated using the validation dataset. It is considered good practice to split a large dataset into subsets for these purposes, if it does not lead to a significant reduction in the quality of data in terms of accuracy and representation. However, where this is not possible (e.g. if the organisation is not working with large datasets or are using pre-trained models as in the case of transfer learning), organisations are encouraged to be cognisant of the risks of systematic bias and put in place appropriate safeguards.

4.1.2. Data for Model Development - e) Periodic reviewing and updating of datasets

It would be prudent for datasets (including training, testing, and validation datasets) to be reviewed periodically to ensure accuracy, quality, currency, relevance and reliability. Where necessary, the datasets can be updated with new input data obtained from actual use of the AI models deployed in production. When such new input data is used, organisations need to be aware of potential bias as using new input data that has already gone through a model once could create a reinforcement bias.

4.1.3. Repeatability - a)

Conducting repeatability assessments for commercial deployments in live environments to ensure that deployments are repeatable;

4.1.4. Repeatability - a)

Performing counterfactual fairness testing. Counterfactual fairness testing ensures that a model's decisions are the same in both the real world and in a counterfactual world where attributes deemed sensitive (such as race or gender) are altered;

4.1.5. Reproducibility - a)

Testing whether specific contexts or particular conditions would need to be taken into account to ensure reproducibility;

4.1.6. Reproducibility - b)

Putting in place verification methods to ensure different aspects of the AI model's reliability and reproducibility;

5. CoSAI

5.1. Establish Risks and Controls for the AI Supply Chain

Link: <https://github.com/cosai-oasis/ws1-supply-chain/blob/main/risks-and-controls-for-the-ai-supply-chain-v1.md>

Link: <https://github.com/cosai-oasis/ws1-supply-chain/blob/main/risks-and-controls-for-the-ai-supply-chain-v1.md>

5.1.1. 3.1.2 Model Training - Compromised Evaluation

Non-reproducible evaluation processes allowing fraudulent results to promote malicious models:
Tamper-proof documentation of evaluation protocols and results

5.1.2. 3.1.4 AI Infrastructure Supply Chain - Serving Infrastructure Compromise

Vulnerabilities in model serving platforms that may affect multiple hosted models:
Evaluate isolation mechanisms, request penetration testing reports, implement client-side verification

5.1.3. 3.1.4 AI Infrastructure Supply Chain - Development Environment Artifacts

Backdoors or vulnerabilities introduced during model development persisting into production:
Request development security documentation, perform independent model validation and testing

5.1.4. 3.2.1 Supply Chain Security for Data - Data Leakage (output)

Unintended exposure of sensitive information via model outputs:

- Implement robust output filtering mechanisms
- Regularly audit and monitor model outputs
- Establish rate limiting on similar queries

5.2. AI Incident Response Framework

Link: <https://github.com/cosai-oasis/ws2-defenders/blob/main/incident-response/AI%20Incident%20Response.md>

Link: <https://github.com/cosai-oasis/ws2-defenders/blob/main/incident-response/AI%20Incident%20Response.md>

5.2.1. 3.3.3. Containment, Eradication, and Recovery Phase - Eradication Procedures - Verification Testing

- Penetration testing
- Attack reproduction attempts
- Red-team exercises
- Issue verification

6. Cloud Security Alliance (CSA)

6.1. AI Controls Matrix

Link: <https://cloudsecurityalliance.org/artifacts/ai-controls-matrix>

Link: <https://cloudsecurityalliance.org/artifacts/ai-controls-matrix>

6.1.1. A&A-02

Conduct independent audit and assurance assessments according to relevant standards at least annually.

6.1.2. A&A-03

Perform independent audit and assurance assessments in response to significant changes or emerging risks and according to risk-based plans and policies.

6.1.3. A&A-05

Define and implement an Audit Management process aligned with global auditing standards, to support audit planning, risk analysis, security control assessment, conclusion, remediation schedules, report generation, and review of past reports and supporting evidence.

6.1.4. AIS-05

Implement a testing strategy, including criteria for acceptance of new information systems, upgrades and new versions, which provides application security assurance and maintains compliance while meeting organizational delivery goals. Automate when applicable and possible.

6.1.5. CCC-02

Establish, maintain and implement a defined quality change control, approval and testing process incorporating baselines, testing, and release standards.

6.1.6. CEK-09

Audit encryption and key management systems, policies, and processes with a frequency that is proportional to the risk exposure of the system with audit occurring preferably continuously but at least annually and after any security event(s).

6.1.7. DCS-14

Secure, monitor, maintain, and test utilities services for continual effectiveness at planned intervals.

6.1.8. GRC-11

Regularly evaluate AI systems, models, datasets & algorithms for bias and fairness to ensure compliance with ethical standards.

6.1.9. SEF-04

Follow a structured approach to evaluate the effectiveness of incident response plans at planned intervals or upon significant changes.

6.1.10. TVM-06

Define, implement and evaluate processes, procedures and technical measures for the periodic performance of penetration testing by independent third parties.

7. CISA

7.1. Principles for the Secure Integration of Artificial Intelligence in Operational Technology

Link: <https://www.cisa.gov/sites/default/files/2026-01/joint-guidance-principles-for-the-secure-integration-of-artificial-intelligence-in-operational-technology-508cV2.pdf>

Link: <https://www.cisa.gov/sites/default/files/2026-01/joint-guidance-principles-for-the-secure-integration-of-artificial-intelligence-in-operational-technology-508cV2.pdf>

7.1.1. 3.3 Conduct Thorough AI Testing and Evaluation

Operators should initially conduct tests of the AI system on infrastructure specifically designed for testing.

8. SANS

8.1. Critical AI Security Guidelines

Link: <https://sansorg.egnyte.com/dl/bvkYQxrW8QMj>

Link: <https://sansorg.egnyte.com/dl/bvkYQxrW8QMj>

8.1.1. 6.1 Regularly Test and Tune LLM Applications/Models

LLM applications and, if possible, the underlying models they employ should be regularly tested to ensure the application's alignment to confirm it behaves as expected and desired. Though models employed should have been red teamed throughout their development prior to deployment, regular assessments of the deployed models and applications should still be performed. Test results could suggest the need for additional mitigations, tuning, or, if applicable, (re)training, to ensure a trustworthy implementation. In addition to red teaming, organizations should conduct regular penetration testing of the AI infrastructure, including vectorDBs, APIs, and connected systems. AI penetration testing differs from traditional pen testing due to the dynamic and complex nature of AI models. Current approaches involve a combination of automated tools and manual techniques, which we expect to change as this capability matures in the industry

9. SDAIA (Saudi Arabia)

9.1. AI Ethics Principles

Link: <https://sdaia.gov.sa/en/SDAIA/about/Documents/ai-principles.pdf>

Link: <https://sdaia.gov.sa/en/SDAIA/about/Documents/ai-principles.pdf>

9.1.1. Principle 2 – Privacy & Security - Build and Validate - 3

The AI System should be tested to ensure that the combination of available data does not reveal the sensitive data or break the anonymity of the observation.

9.1.2. Principle 5 – Reliability & Safety - Prepare Input Data - 2

It is crucial for the build and validate step to test how the system behaves under outlier events, extreme parameters, etc. In this step, stress test data should be prepared for extreme scenarios.

9.1.3. Principle 5 – Reliability & Safety - Build and Validate - 1

To develop a sound and functional AI system that is both reliable and safe, the AI system's technical construct should be accompanied by a comprehensive methodology to test the quality of the predictive data-based systems and models according to standard policies and protocols.

9.1.4. Principle 6 – Transparency & Explainability - Prepare Input Data - 2

The data sets should be assessed in the context of their accuracy, suitability, validity, and source. This has a direct effect on the training and implementation of these systems since the criteria for the data's organization, and structuring must be transparent and explainable in their acquisition and collection adhering to data privacy regulations and intellectual property standards and controls.

9.1.5. Principle 7 – Accountability & Responsibility - Build and Validate - 3

The decisions should be supported with quantitative (performance measures on train/test datasets, consistency of the performance on different sensitive groups, performance comparison for each set of hyperparameters, etc.) and qualitative indicators (decisions to mitigate and correct unintended risks from inaccurate predictions).

9.2. AI Adoption Framework

Link: <https://sdaia.gov.sa/en/SDAIA/about/Files/AIAdoptionFramework.pdf>

Link: <https://sdaia.gov.sa/en/SDAIA/about/Files/AIAdoptionFramework.pdf>

9.2.1. 5.1.2 Privacy and Safety - Testing Risk Scenarios regularly

Simulating failure or outage scenarios is a best practice for ensuring infrastructure readiness. These tests include: Data center failures, connection losses, or security breaches, measuring system resilience, identifying vulnerabilities, and updating contingency plans based on real outcomes.

10. Smart Dubai (UAE)

10.1. AI Ethics Principles & Guidelines

Link: <https://www.digitaldubai.ae/docs/default-source/ai-principles-resources/ai-ethics.pdf>

Link: <https://www.digitaldubai.ae/docs/default-source/ai-principles-resources/ai-ethics.pdf>

10.1.1. 1.1.1.1

AI developer organisations and AI operator organisations should undertake reasonable data exploration and/or testing to identify potentially prejudicial decision-making tendencies in AI systems arising from biases in the data.

10.1.2. 1.2.2.8

AI operator organisations should subject AI systems informing significant decisions to quality checks at least as stringent as those that would be required of a human being taking the same decision.

10.1.3. 1.2.3.1

When AI systems are used for critical decisions, external auditing of the AI systems in question should be used as a means to ensure meaningful standards of transparency and accountability are upheld.

11. Qatar Central Bank

11.1. Artificial Intelligence Guidelines

Link: https://www.qcb.gov.qa/Services/Financial%20Technology/QCB_Artificial_Intelligence_Guideline.pdf

Link: https://www.qcb.gov.qa/Services/Financial%20Technology/QCB_Artificial_Intelligence_Guideline.pdf

11.1.1. 12.8

The external and internal auditors of the Entity must be able to review the accounting records and internal controls of the outsourcing service provider.

11.1.2. 15.3

An Entity should ensure it has different Data Sets for training, validation, and testing.

11.1.3. 15.4

The Entity must use the Training Data and Validation Data to train the AI System. Such data should be subject to internal data quality control procedures.

11.1.4. 15.5

An Entity may ensure the model is reviewed to identify any unintuitive or false causal relationships. The validation may be carried out by an independent function within an Entity or by an external organization.

11.1.5. 15.6

The Entity must use the test data to determine the accuracy of the AI System.

11.1.6. 15.10

An Entity should ensure Testing Data Sets are sufficiently relevant, representative and have the appropriate statistical properties, including as regards the Customers of whom the AI System is intended to be used (i.e., how relevant the data and inferences drawn from the data are to the AI System).

11.1.7. 15.12

An Entity should, where relevant, conduct rigorous, independent validation and testing of material trained AI Models to ensure the accuracy, appropriateness, and reliability of the models prior to deployment.

11.1.8. 15.13

An Entity should ensure the model is reviewed to identify any unintuitive or false causal relationships. The validation may be carried out by an independent function within an Entity or by an external organization.

12. METI (Japan)

12.1. Governance Guidelines for Implementation of AI Principles

Link: https://www.meti.go.jp/shingikai/mono_info_service/ai_shakai_jisso/pdf/20220128_2.pdf

Link: https://www.meti.go.jp/shingikai/mono_info_service/ai_shakai_jisso/pdf/20220128_2.pdf

12.1.1. Action Target 6-1

Companies that develop and operate AI systems should, under the leadership of top management, conduct re-evaluations, update their understanding, obtain new points of views, or take other relevant actions with respect to Action Targets 1-1 through 1.3, in a timely manner. When implementing Action Target 5-2, they should also consider obtaining opinions not only for the current AI management system and the operation of such system, but also conducive to review of entire AI governance, including analyses of conditions and risks.

13. ISO/IEC

13.1. TS 42119-2:2025

Link: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:ts:42119:-2:ed-1:v1:en>

Link: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:ts:42119:-2:ed-1:v1:en>

13.1.1. 5.4

Risk-based testing

13.1.2. 5.5

Test processes

13.1.3. 5.6

Test documentation

13.1.4. 5.7

Testing stakeholders

13.1.5. 7.1

Introduction to test approaches

13.1.6. 7.2

Test levels

13.1.7. 7.3

Test types

13.1.8. 7.4

Test design techniques and measures

13.1.9. A.2

Purpose of testing

13.1.10. A.3

Software testing in context

13.1.11. A.4

Static and dynamic testing

13.1.12. A.5

14. Databricks

14.1. The Databricks AI Security Framework

Link: <https://www.databricks.com/sites/default/files/2025-02/databricks-ebook-dasf-2.pdf>

Link: <https://www.databricks.com/sites/default/files/2025-02/databricks-ebook-dasf-2.pdf>

14.1.1. DASF 38: Platform security — penetration testing, red teaming, bug bounty and vulnerability management

Mitigating attacks on infrastructure hosting AI services, including AI red teaming for large language models, is crucial for safe model development and deployment. Regular security and penetration testing help identify and address infrastructure vulnerabilities before attackers can exploit them.

Databricks operates a formal, documented vulnerability management program overseen by the Chief Security Officer (CSO). The program is management-approved, reviewed annually, and communicated to all relevant internal parties. The policy mandates that vulnerabilities be addressed based on severity: critical vulnerabilities within 14 days, high-severity vulnerabilities within 30 days, and medium-severity vulnerabilities within 60 days.

AI red teaming, especially for large language models, is an essential component of ensuring model safety and security. Databricks conducts regular AI red teaming on models and systems developed internally.

14.1.2. DASF 45: Evaluate models

Model evaluation is a critical component of the machine learning lifecycle. It provides data scientists with the tools to measure, interpret and explain the performance of their models. MLflow plays a critical role in accelerating model development by offering insights into the reasons behind a model's performance and guiding improvements and iterations. MLflow offers many industry-standard native evaluation metrics for classical machine learning algorithms and LLMs, and also facilitates the use of custom evaluation metrics.

14.1.3. DASF 47: Compare LLM outputs on set prompts

New, no-code visual tools allow users to compare models' output based on set prompts, which are automatically tracked within MLflow. With integration into Mosaic AI Model Serving, customers can deploy the best model to production. The AI Playground is a chat-like environment where you can test, prompt and compare LLMs.

14.1.4. DASF 49: Automate LLM evaluation

The "LLM-as-a-judge" feature in MLflow 2.8 automates LLM evaluation, offering a practical alternative to human judgment. It's designed to be efficient and cost-effective, maintaining consistency with human scores. This tool supports various metrics, including standard and customizable GenAI metrics, and allows users to select an LLM as a judge and define specific grading criteria.

14.1.5. DASF 66: Use Human-in-the-loop feedback

End-user feedback is invaluable for understanding how your GenAI application performs in real-world scenarios. MLflow provides tools to capture, store, and analyze feedback directly from the users of your deployed applications.

15. ENISA

15.1. Multilayer Framework for Good Cybersecurity Practices for AI

Link:

<https://www.enisa.europa.eu/sites/default/files/publications/Multilayer%20Framework%20for%20Good%20Cybersecurity%20Practices%20for%20AI.pdf>

Link:

<https://www.enisa.europa.eu/sites/default/files/publications/Multilayer%20Framework%20for%20Good%20Cybersecurity%20Practices%20for%20AI.pdf>

15.1.1. Security testing

Security testing of AI has some commonalities with security testing of traditional systems, but also provides new challenges and requires different approaches, due to:

- significant differences between subsymbolic AI and traditional systems that have strong implications on their security and on how to test their security properties;
- non-determinism that may result from self-learning, i.e. AI-based systems may evolve over time and as a consequence, security properties may degrade;
- the test oracle problem, where assigning a test verdict is different and more difficult for AI-based systems, since not all expected results are known a priori;
- data-driven algorithms, where in contrast to traditional systems, (training) data forms the behaviour of subsymbolic AI.

15.1.2. From the lab to the market 3

Do you have/promote testing environments, like sandboxes/cyber ranges/simulation platforms to test and evaluate AI vulnerabilities before market? How?

15.1.3. Infrastructure 2

Have you specified/defined measurements and KPIs which the AI stakeholders can use to assess the appropriateness of the controls undertaken?

15.1.4. Regulation 2

Have national auditors and certification and accreditation bodies been established for assessing the security of the AI systems?

15.1.5. Regulation 3

How do you evaluate security of the AI systems (e.g. via conformity assessment, certification, standards compliance, risk assessment, etc.)?

15.1.6. Regulation 4

What are the obligations you have imposed for testing, risk management, documentation and human oversight throughout the AI systems' life cycle to ensure continuous data and training model integrity?

16. ICO

16.1. Guidance on the AI Auditing Framework - Draft guidance for consultation

Link: <https://ico.org.uk/media2/about-the-ico/consultations/2617219/guidance-on-the-ai-auditing-framework-draft-for-consultation.pdf>

Link: <https://ico.org.uk/media2/about-the-ico/consultations/2617219/guidance-on-the-ai-auditing-framework-draft-for-consultation.pdf>

16.1.1. Detective Controls - 2

Regularly test, assess and evaluate the effectiveness of any security measures they have put in place (eg through techniques such as penetration testing).

17. ISO

17.1. 42001:2023 - Information technology — Artificial intelligence — Management system

Link: <https://www.iso.org/standard/42001>

Link: <https://www.iso.org/standard/42001>

17.1.1. 9.2

Internal audit

17.1.2. 9.3.3

Management review results

18. Federal Office for Information Security

18.1. AI Security Concerns in a Nutshell

Link: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/KI/Practical_AI-Security_Guide_2023.pdf?__blob=publicationFile&v=5

Link: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/KI/Practical_AI-Security_Guide_2023.pdf?__blob=publicationFile&v=5

18.1.1. 5.3 Defending against Poisoning and Backdoor Attacks - Search for Triggers

The triggers used for backdoors rely on logical shortcuts between the target class and the input. To find a shortcut, one must determine the minimal input change required to shift the model's prediction. If such a change is minimal, a backdoor may have been found [20]. Another method for the image domain is to randomly mask parts of an input image and examine how the model's prediction changes. If the input image contains a trigger, masking it will change the model's prediction [21].

19. World Economic Forum

19.1. Presidio AI Framework: Towards Safe Generative AI Models

Link: https://www3.weforum.org/docs/WEF_Presidio_AI%20Framework_2024.pdf

Link: https://www3.weforum.org/docs/WEF_Presidio_AI%20Framework_2024.pdf

19.1.1. Red teaming and reinforcement learning from human feedback (RLHF)

Performing red teaming early, especially during finetuning and validation of the building phase, is crucial for preventing adverse outcomes and ensuring model safety. Addressing vulnerabilities and ethical concerns earlier in the life cycle demonstrates a commitment to security and ethics while building trust among stakeholders. For foundation models, tests should cover prompt injection, leaking, jailbreaking, hallucination, IP and personal information (PI) generation, as well as identifying toxic content. While red teaming is effective for known vulnerabilities, it may have limitations in identifying unknown risks, especially before mass release.

19.1.2. Shifting left for optimized risk mitigation

The term "shift-left" describes implementing quality assurance and testing measures earlier in a product cycle. The core objective is proactively identifying and managing potential risks, increasing efficiency and cost-effectiveness. This well-established concept applies to various technologies and processes, including software engineering.

20. CEN/CENELEC

20.1. prEN 40000-1-2: Cybersecurity requirements for products with digital elements - Part 1-2: Principles for cyber resilience

Link: <https://genorma.com/en/standards/pren-40000-1-2>

Link: <https://genorma.com/en/standards/pren-40000-1-2>

20.1.1. 7.5

Cybersecurity Verification and validation

21. EU

21.1. EU AI Act

Link: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202401689

Link: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202401689

21.1.1. 9.7 Risk Management System

Testing procedures may include testing in real-world conditions in accordance with Article 60.

21.1.2. 9.8 Risk Management System

The testing of high-risk AI systems shall be performed, as appropriate, at any time throughout the development process, and, in any event, prior to their being placed on the market or put into service. Testing shall be carried out against prior defined metrics and probabilistic thresholds that are appropriate to the intended purpose of the high-risk AI system.

21.1.3. 15.2 Accuracy, Robustness and Cybersecurity

To address the technical aspects of how to measure the appropriate levels of accuracy and robustness set out in paragraph 1 and any other relevant performance metrics, the Commission shall, in cooperation with relevant stakeholders and organisations such as metrology and benchmarking authorities, encourage, as appropriate, the development of benchmarks and measurement methodologies.

22. MIC/METI (Japan)

22.1. AI Guidelines for Business

Link: https://www.meti.go.jp/shingikai/mono_info_service/ai_shakai_jisso/pdf/20240419_9.pdf

Link: https://www.meti.go.jp/shingikai/mono_info_service/ai_shakai_jisso/pdf/20240419_9.pdf

22.1.1. Safety - 1 (a)

Ensure that the AI system/service is sufficiently fulfilling the requirements, including the accuracy of outputs (reliability).

22.1.2. Safety - 1 (b)

Ensure that the performance level is maintained under various circumstances, and that grossly incorrect judgments are not made for irrelevant events (robustness).

23. Central Bank of the UAE

23.1. Guidance Note on the Consumer Protection and Responsible Adoption and Use of Artificial Intelligence and Machine Learning by Licensed Financial Institutions in the U.A.E

Link: <https://rulebook.centralbank.ae/en/rulebook/guidance-note-consumer-protection-and-responsible-adoption-and-use-artificial-intelligence>

Link: <https://rulebook.centralbank.ae/en/rulebook/guidance-note-consumer-protection-and-responsible-adoption-and-use-artificial-intelligence>

23.1.1. 3. Fairness/Non-Discrimination and Ethics - c

AI deployed should be subject to periodic testing, i.e. once a year or each time a model is upgraded, materially changed or a new one is introduced, to identify and remediate undue/unintended embedded biases or discriminatory outcomes. The deployment of AI should reflect the institution's ethical standards and code of conduct. Decisions made or supported by AI should be consistent with the duty to act honestly, fairly and in the best interests of consumers.

23.1.2. 5. Data Quality, Privacy and Security - d

Robustness and safety should be integral to AI development. AI should be subject to stress testing and validation to ensure they operate reliably under a range of scenarios and do not produce unsafe or unintended outputs. Institutions should incorporate operational resilience measures—such as redundancy, contingency planning and incident response—to minimise disruption or harm to consumers from system failures or cyber-attacks.

Where an LFI is developing AI internally, it should consider the use of third-party independent reviews to check suitability, security and reliability.

24. Cyber Security Council (UAE)

24.1. National Cyber Security Policy for Artificial Intelligence

Link: https://csc.gov.ae/documents/38662/0/National+Cyber+Security+Policy+for+Artificial+Intelligence_v1.1.pdf/4e02b32e-9f62-948d-4bc8-b580d596451b?t=1766994254544

Link: https://csc.gov.ae/documents/38662/0/National+Cyber+Security+Policy+for+Artificial+Intelligence_v1.1.pdf/4e02b32e-9f62-948d-4bc8-b580d596451b?t=1766994254544

24.1.1. 2.4.4

The entity shall ensure the security of AI/ML systems through comprehensive testing and validation, providing assurance of their resilience against potential threats.

24.1.2. 3.1.4 Change Management and Reporting - 3

All manual changes should be tested in a secure environment before deployment to identify and address any unexpected impacts or vulnerabilities introduced by the change.

24.1.3. 3.2.3 Patch Management - 3

All patches and updates should undergo testing and verification in a controlled environment before deployment in production systems to prevent unintended impacts on system performance and availability.

24.1.4. 3.2.4 Vulnerability Management for AI/ML Systems - 2

The entity should employ an appropriate combination of testing tools, methodologies, and external audits to identify vulnerabilities in AI/ML infrastructure and applications.

24.1.5. 3.2.5 Application Security - 2

The entity should perform security testing, including code reviews and static/dynamic analysis, to identify and remediate potential security flaws in AI/ML applications prior to deployment.

24.1.6. 3.2.6 Network Security for AI/ML Infrastructure - 6

The entity should routinely conduct vulnerability assessments and penetration tests on the AI/ML network infrastructure to identify and remediate potential security weaknesses.

24.1.7. 3.4.4 Testing and Validation of AI/ML Systems - 1

The entity should establish and maintain a robust framework for the testing and validation of AI/ML systems, ensuring their accuracy, reliability, and performance meet defined criteria.

24.1.8. 3.4.4 Testing and Validation of AI/ML Systems - 2

All AI/ML systems should undergo rigorous testing before deployment to verify their functionality, security, and resilience against potential threats.

24.1.9. 3.4.4 Testing and Validation of AI/ML Systems - 3

The testing process should consider various scenarios and inputs, including edge cases, to evaluate the AI/ML systems' behavior under different conditions.

24.1.10. 3.5.2 Defending Against AI/ML Attacks - 3

Regular security testing, including penetration testing and red teaming exercises, should be conducted on AI/ML systems to identify potential weaknesses and implement necessary defenses.

25. IMDA

25.1. Model AI Governance Framework for Agentic AI

Link: <https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/mgf-for-agentic-ai.pdf>

Link: <https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/mgf-for-agentic-ai.pdf>

25.1.1. 2.3.2 Before deploying, test agents

Organisations should test agents for safety and security before deployment. This provides confidence that the agents work as expected and controls are effective. Best practices on software and LLM testing are still relevant, such as unit and integration testing for software systems, as well as selecting representative datasets, and useful metrics and evaluators for LLM testing. Organisations can refer to previous guidance, such as the Starter Kit for testing of LLM-based apps for safety and reliability.

26. NCSC/NSA/CISA etc

26.1. AI Data Security

Link: https://media.defense.gov/2025/May/22/2003720601/-1/-1/0/CSI_AI_DATA_SECURITY.PDF

Link: https://media.defense.gov/2025/May/22/2003720601/-1/-1/0/CSI_AI_DATA_SECURITY.PDF

26.1.1. 3.11 Edge Cases

Identify and mitigate edge cases that can cause models to malfunction.

27. OpenAI

27.1. Safety Best Practices

Link: <https://platform.openai.com/docs/guides/safety-best-practices>

Link: <https://platform.openai.com/docs/guides/safety-best-practices>

27.1.1. Adversarial testing

We recommend "red-teaming" your application to ensure it's robust to adversarial input. Test your product over a wide range of inputs and user behaviors, both a representative set and those reflective of someone trying to 'break' your application. Does it wander off topic? Can someone easily redirect the feature via prompt injections, e.g. "ignore the previous instructions and do this instead"?

28. Microsoft

28.1. Cloud Adoption Framework - Secure AI

Link: <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/scenarios/ai/secure>

Link: <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/scenarios/ai/secure>

28.1.1. Discover AI security risks 3 - Test AI models for security vulnerabilities

AI models contain unique vulnerabilities like data leakage, prompt injection, and model inversion that attackers can exploit. Real-world testing uncovers risks that static reviews can't detect. Test models for vulnerabilities using data-loss-prevention techniques and adversarial simulations, and red team both generative AI and nongenerative AI models to simulate real attacks.

28.2. Responsible AI Standard

Link: <https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/final/en-us/microsoft-brand/documents/Microsoft-Responsible-AI-Standard-General-Requirements.pdf?culture=en-us&country=us>

Link: <https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/final/en-us/microsoft-brand/documents/Microsoft-Responsible-AI-Standard-General-Requirements.pdf?culture=en-us&country=us>

28.2.1. A3.4

Document an evaluation plan for each of the performance metrics and error types.

28.2.2. A3.5

Use the methods defined in requirement A3.4 to conduct evaluations. Document the pre-release results of the evaluations. Determine and document how often ongoing evaluation should be conducted to continue supporting this Goal.

28.2.3. A5.6

Conduct evaluations defined by requirement A5.4 using a near-release version of the system. Document the results.

28.2.4. T1.5

Conduct evaluations defined by requirement T1.3. Document the pre-release results of the evaluations. Determine and document how often ongoing evaluation should be conducted to continue supporting this Goal.

28.2.5. T3.4

Define and document Responsible Release Criteria to achieve this Goal.

28.2.6. T3.5

Conduct evaluations defined by requirement T3.3. Document the pre-release results of the evaluations. Determine and document how often ongoing evaluation should be conducted to continue supporting this goal.

28.2.7. F1.2

Evaluate all data sets to assess inclusiveness of identified demographic groups and collect data to close gaps. Document this process and its results.

28.2.8. F1.3

Define and document the evaluation that you will perform to support this Goal. Include:

- 1) any system components to be evaluated, in addition to the whole system,
- 2) the metrics to be used to evaluate the system components and the whole system, and
- 3) a description of the data set to be used for this evaluation.

28.2.9. F1.4

Define and document Responsible Release Criteria to achieve this Goal, as follows:

For each metric, document:

- 1) any target minimum performance level for all groups, and
- 2) the target maximum (absolute or relative) performance difference between groups.

28.2.10. F1.5

Evaluate the system according to the defined Responsible Release Criteria.

28.2.11. F2.2

Evaluate all data sets to assess inclusiveness of identified demographic groups and collect data to close any gaps. Document this process and its results.

28.2.12. F2.3

Define and document the evaluation that you will perform to support this Goal. Include:

- 1) any system components to be evaluated, in addition to the whole system,
- 2) the metrics to be used to evaluate the system components and the whole system, and
- 3) the data set to be used for this evaluation.

28.2.13. F2.4

Define and document Responsible Release Criteria to achieve this Goal, as follows:

For each metric, document the target maximum difference (absolute or relative) between the rates at which resources and opportunities are allocated to groups.

28.2.14. F2.5

Evaluate the system according to the defined Responsible Release Criteria.

28.2.15. F3.2

Define and document any system components to be evaluated, in addition to the whole system.

28.2.16. F3.3

Define and document a plan to evaluate the system components and the whole system for risks of stereotyping, demeaning, and erasing the prioritized identified demographic groups.

28.2.17. F3.4

Evaluate the system according to the plan defined in requirement F3.3.

28.2.18. RS1.2

Evaluate training and test data sets to ensure that they include representation of the intended uses, operational factors, and an appropriate range of settings for each factor. Document the evaluation.

28.2.19. RS1.6

Define and document an evaluation plan based on requirements RS1.1, RS1.3, RS1.4, and RS1.5, to include the environment in which the system will be evaluated.

28.2.20. RS1.7

Evaluate the system according to the evaluation plan defined in requirement RS1.6 to ensure reliable and safe system behavior. Document the pre-release results of the evaluation. Determine and document how often ongoing evaluation should be conducted to continue supporting this goal.

28.2.21. RS3.5

Conduct all evaluations tagged as Ongoing Evaluation Checkpoints in other Goals on an ongoing basis.

29. Google

29.1. Secure AI Framework

Link: <https://www.saif.google/secure-ai-framework>

Link: <https://www.saif.google/secure-ai-framework>

29.1.1. Model and Data Access Controls

Minimize internal access to models, weights, datasets, etc. in storage and in production use.

29.1.2. Red Teaming

Identify security and privacy improvements through self-driven adversarial attacks on AI infrastructure and products.

29.1.3. Product Governance

Validate that all AI models and products meet the established security and privacy requirements.

30. European Commission

30.1. Ethics guidelines for trustworthy AI

Link: <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>

Link: <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>

30.1.1. 1.7.1 Auditability

Auditability entails the enablement of the assessment of algorithms, data and design processes. This does not necessarily imply that information about business models and intellectual property related to the AI system must always be openly available. Evaluation by internal and external auditors, and the availability of such evaluation reports, can contribute to the trustworthiness of the technology. In applications affecting fundamental rights, including safety-critical applications, AI systems should be able to be independently audited.

30.1.2. 2.1.4 Testing and Validating

Testing and validation of the system should occur as early as possible, ensuring that the system behaves as intended throughout its entire life cycle and especially after deployment. It should include all components of an AI system, including data, pre-trained models, environments and the behaviour of the system as a whole. The testing processes should be designed and performed by an as diverse group of people as possible. Multiple metrics should be developed to cover the categories that are being tested for different perspectives. Adversarial testing by trusted and diverse “red teams” deliberately attempting to “break” the system to find vulnerabilities, and “bug bounties” that incentivise outsiders to detect and responsibly report system errors and weaknesses, can be considered. Finally, it must be ensured that the outputs or actions are consistent with the results of the preceding processes, comparing them to the previously defined policies to ensure that they are not violated.

30.1.3. 2.1.5 Quality of Service Indicators

Appropriate quality of service indicators can be defined for AI systems to ensure that there is a baseline understanding as to whether they have been tested and developed with security and safety considerations in mind. These indicators could include measures to evaluate the testing and training of algorithms as well as traditional software metrics of functionality, performance, usability, reliability, security and maintainability.

31. MITRE

31.1. ATLAS Framework

Link: <https://atlas.mitre.org/mitigations>

Link: <https://atlas.mitre.org/mitigations>

31.1.1. AML.M0008 - Validate AI Model

Validate that AI models perform as intended by testing for backdoor triggers, potential for data leakage, or adversarial influence. Monitor AI model for concept drift and training data drift, which may indicate data tampering and poisoning.

32. NIST

32.1. AI RMF 1.0

Link: <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf>

Link: <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf>

32.1.1. GOVERN 4.3

Organizational practices are in place to enable AI testing, identification of incidents, and information sharing.

32.1.2. GOVERN 5.1

Organizational policies and practices are in place to collect, consider, prioritize, and integrate feedback from those external to the team that developed or deployed the AI system regarding the potential individual and societal impacts related to AI risks.

32.1.3. MAP 2.3

Scientific integrity and Test, Evaluation, Verification, and Validation (TEVV) considerations are identified and documented, including those related to experimental design, data collection and selection (e.g., availability, representativeness, suitability), system trustworthiness, and construct validation.

32.1.4. MEASURE 1.3

Internal experts who did not serve as front-line developers for the system and/or independent assessors are involved in regular assessments and updates. Domain experts, users, AI actors external to the team that developed or deployed the AI system, and affected communities are consulted in support of assessments as necessary per organizational risk tolerance.

32.1.5. MEASURE 2.1

Test sets, metrics, and details about the tools used during TEVV are documented.

32.1.6. MEASURE 2.2

Evaluations involving human subjects meet applicable requirements (including human subject protection) and are representative of the relevant population.

32.1.7. MEASURE 2.3

AI system performance or assurance criteria are measured qualitatively or quantitatively and demonstrated for conditions similar to deployment setting(s). Measures are documented.

32.1.8. MEASURE 2.5

The AI system to be deployed is demonstrated to be valid and reliable. Limitations of the generalizability beyond the conditions under which the technology was developed are documented.

32.1.9. MEASURE 2.6

The AI system is evaluated regularly for safety risks – as identified in the MAP function. The AI system to be deployed is demonstrated to be safe, its residual negative risk does not exceed the risk tolerance, and it can fail safely, particularly if made to operate beyond its knowledge limits. Safety metrics reflect system reliability and robustness, real-time monitoring, and response times for AI system failures.

32.1.10. MEASURE 2.7

AI system security and resilience – as identified in the MAP function – are evaluated and documented.

32.1.11. MEASURE 2.13

Effectiveness of the employed TEVV metrics and processes in the MEASURE function are evaluated and documented.

32.2. SP 800-218A

Link: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218A.pdf>

Link: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218A.pdf>

32.2.1. PO.4.1

Define criteria for software security checks and track throughout the SDLC.

Implement guardrails and other controls throughout the AI development life cycle, extending beyond the traditional SDLC.

32.2.2. PW.2.1

Have 1) a qualified person (or people) who were not involved with the design and 2) automated processes instantiated in the toolchain review the software design to confirm and enforce that it meets all of the security requirements and satisfactorily addresses the identified risk information.

32.2.3. PW.3.3

Include adversarial samples in the training and testing data to improve attack prevention.

Use a process and corresponding controls to test the adversarial samples and put appropriate guardrails on training and testing use.

32.2.4. PW.7.2

Perform the code review and/or code analysis based on the organization's secure coding standards, and record and triage all discovered issues and recommended remediations in the development team's workflow or issue tracking system.

Scan all AI models for malware, vulnerabilities, backdoors, and other security issues in accordance with the organization's code review and analysis policies or guidelines.

32.2.5. PW.8.1

Determine whether executable code testing should be performed to find vulnerabilities not identified by previous reviews, analysis, or testing and, if so, which types of testing should be used.

Include AI models in code testing policies and guidelines. Several forms of code testing can be used for AI models, including unit testing, integration testing, penetration testing, red teaming, use case testing, and adversarial testing.

32.2.6. PW.8.2

Scope the testing, design the tests, perform the testing, and document the results, including recording and triaging all discovered issues and recommended remediations in the development team's workflow or issue tracking system.

Test all AI models for vulnerabilities in accordance with the organization's code testing policies or guidelines.

Retest AI models when they are retrained or new data sources are added.

32.2.7. RV.1.2

Review, analyze, and/or test the software's code to identify or confirm the presence of previously undetected vulnerabilities.

Scan and test AI models frequently to identify previously undetected vulnerabilities.

Rely mainly on automation for ongoing scanning and testing, and involve a human-in-the-loop as needed.

Conduct periodic audits of AI models.

32.2.8. RV.3.3

Review the software for similar vulnerabilities to eradicate a class of vulnerabilities, and proactively fix them rather than waiting for external reports.

32.3. AI 800-1

Link: <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.800-1.ipd2.pdf>

Link: <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.800-1.ipd2.pdf>

32.3.1. Practice 1.1: Anticipate model capabilities - 6

Consider involving experts in anticipating model capabilities, such as by potentially granting them access to intermediate model checkpoints for open-ended experimentation to identify other ways the model could be misused to cause harm.

32.3.2. Practice 3.1: Assess misuse risk from threat actors gaining unauthorized access to the model - 4

When relevant, consider using cybersecurity red teams and penetration testing to assess how difficult it would be for an actor to circumvent security measures.

32.3.3. Practice 4.1: Evaluate model capabilities on tasks relevant to assessing misuse risk - 1

Evaluate the model's capabilities throughout the development process: during training, after training, and when integrated into a downstream system or interface.

32.3.4. Practice 4.1: Evaluate model capabilities on tasks relevant to assessing misuse risk - 2

Consider using automated or otherwise less expensive tests on tasks related to real-world harm to indicate that a model lacks dangerous capabilities. If initial tests are inconclusive or indicate a potential risk, conduct more costly and precise measurements.

32.3.5. Practice 4.1: Evaluate model capabilities on tasks relevant to assessing misuse risk - 3

Consult relevant experts when creating evaluations to ensure that they are indicative of real-world scenarios.

32.3.6. Practice 4.1: Evaluate model capabilities on tasks relevant to assessing misuse risk - 4

Maximize model performance on evaluation tasks by adjusting prompts, software scaffolding, fine-tuning the model, or other means.

32.3.7. Practice 4.1: Evaluate model capabilities on tasks relevant to assessing misuse risk - 5

If a gap exists between the effort or resources applied by evaluators to maximize system performance and the resources that could be applied by a threat actor, account for that gap when interpreting evaluation results.

32.3.8. Practice 4.1: Evaluate model capabilities on tasks relevant to assessing misuse risk - 6

Avoid overlap between data used to train a model and data used in capability evaluations and measure the extent of any overlap.

32.3.9. Practice 4.2: Red-team safeguards - 1

Evaluate whether an adequately resourced red team can misuse the model to achieve a predetermined goal or accomplish a related proxy task in a realistic deployment context. If using proxy tasks, ensure that they are at least as easy to achieve as undesirable real-world outcomes, considering both the inherent complexity of the task and any additional difficulty introduced by safeguards.

32.3.10. Practice 4.2: Red-team safeguards - 2

Assemble red teams based on their ability to succeed at the determined task, considering factors such as relevant domain expertise, their independence from the model developer, diversity of perspectives, and lack of incentives that conflict with their red-teaming goal.

32.3.11. Practice 4.2: Red-team safeguards - 3

Provide the red team with a clearly defined task. Determine a performance metric for measuring the red team's success and provide incentives and accountability for task completion.

32.3.12. Practice 4.2: Red-team safeguards - 4

Compare the red team's expertise, resources, and time available to those of a relevant threat actor. Ensure that the red team is adequately resourced to the extent possible and consider providing alternative and/or additional resources to the red team or making the red team's task easier in other ways (such as providing additional access to the model or dividing complex tasks into constituent pieces) to compensate for any remaining gaps between the red team and threat actors.

32.3.13. Practice 4.2: Red-team safeguards - 5

Provide the red team with at least as much information about the model (or downstream system if testing at the integration level) as would be available to an attacker and make explicit any respects in which the red team lacks full information about the design of safeguards.

32.3.14. Practice 4.2: Red-team safeguards - 8

Consider testing the model at different levels, such as the model without safeguards, the model with safeguards, and the model integrated into a downstream system or interface.

32.3.15. Practice 5.2: Assess misuse risk based on implemented safeguards - 3

If additional safeguards are added in response to identified risks, consider re-assessing misuse risk prior to deployment by carrying out red-teaming exercises (Practice 4.2) with the additional safeguards in place.

32.3.16. Practice 6.1: Monitor for evidence of misuse - 3

Periodically assess the effectiveness of misuse detection systems, including through red teaming.

32.3.17. Practice 6.2: Respond to incidents of model misuse - 4

Consider carrying out drills to practice responding to time-sensitive and safety-critical scenarios of misuse, if appropriate given the level of risk of the deployment.

32.4. IR 8596: Cybersecurity Framework Profile for Artificial Intelligence (Cyber AI Profile): NIST Community Profile

Link: <https://csrc.nist.gov/pubs/ir/8596/iprd>

Link: <https://csrc.nist.gov/pubs/ir/8596/iprd>

32.4.1. ID.IM-01

Improvements are identified from evaluations

32.4.2. ID.IM-02

Improvements are identified from security tests and exercises, including those done in coordination with suppliers and relevant third parties

32.4.3. RS.MA-02

Incident reports are triaged and validated

32.5. AI 100-2e2025: Adversarial Machine Learning A Taxonomy and Terminology of Attacks and Mitigations

Link: <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2025.pdf>

Link: <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2025.pdf>

32.5.1. 2.3.3 Model inspection and sanitization

Model inspection analyzes the trained MLmodel before its deployment to determine whether it was poisoned. An early work in this space is NeuronInspect [168], which is based on explainability methods to determine different features between clean and backdoored models that are subsequently used for outlier detection. DeepInspect [78] uses a conditional generative model to learn the probability distribution of trigger patterns and performs model patching to remove the trigger. Xu et al. [416] proposed the Meta Neural Trojan Detection (MNTD) framework, which trains a meta-classifier to predict whether a given ML model is backdoored (or “Trojaned,” in the authors’ terminology). This technique is general and can be applied to multiple data modalities, such as vision, speech, tabular data, and NLP. Once a backdoor is detected, model sanitization can be performed via pruning [407], retraining [429], or fine-tuning [217] to restore the model’s accuracy

32.5.2. 3.3.3 Direct Prompting Attacks - Interventions during evaluation (4)

Evaluations can measure the vulnerability of models to query-based attacks, which can then inform trust and affordance decisions, as well as developer and user education. Evaluations can include broad automated vulnerability assessments [72, 107, 324] as well as targeted expert red teaming [381] and bug bounties [16]. Current evaluation approaches, though a useful tool, may underestimate vulnerabilities accessible to actors with more time, resourcing, or luck. Evaluations measure model vulnerabilities at a particular moment in time; assessments may change if new attacks are developed, additional data is collected post-training, or model capabilities are improved. Continuous evaluations following deployment can help combat these challenges.