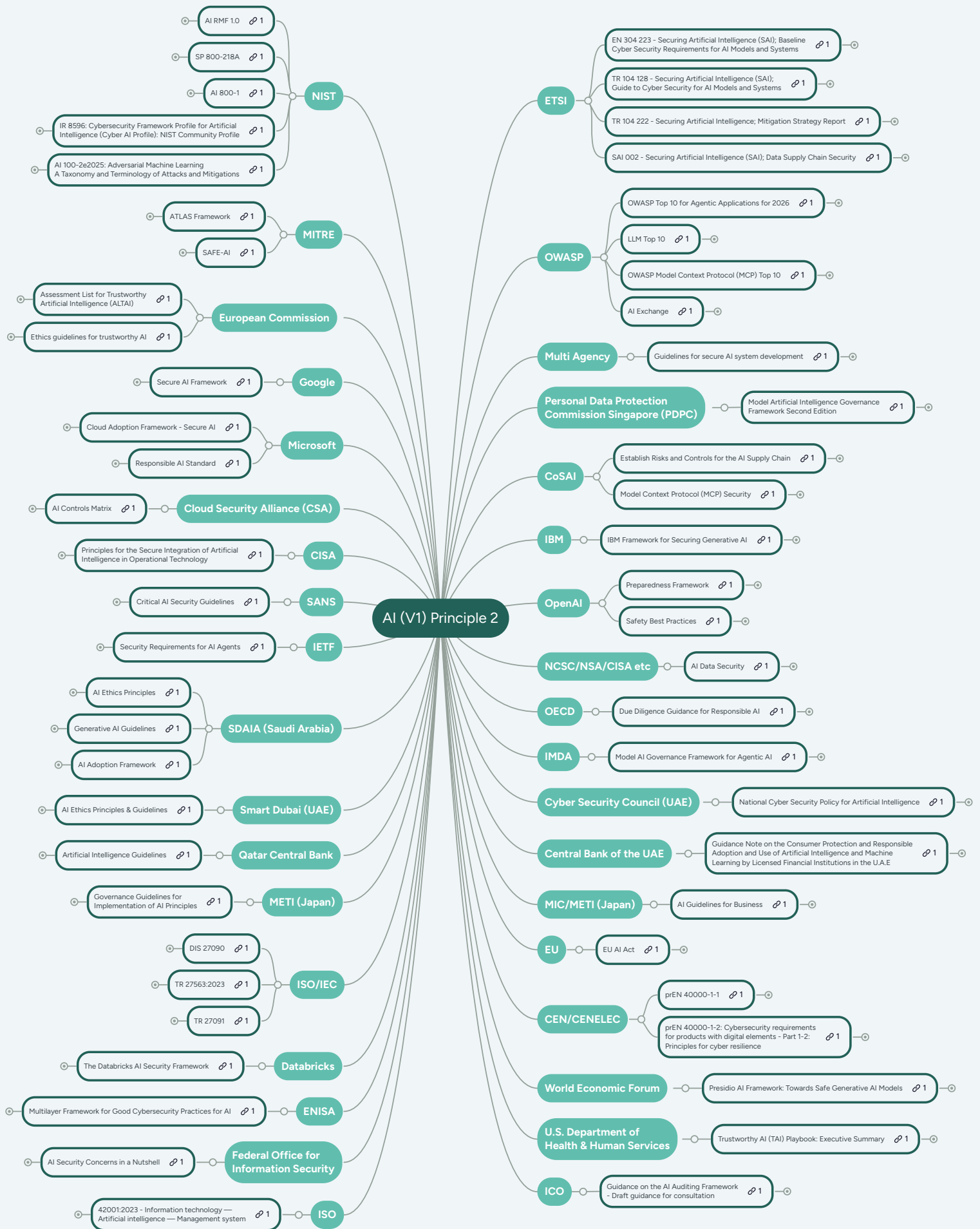




AI (V1) Principle 2

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 5.1.2-1

As part of deciding whether to create an AI system, a System Operator and/or Developer shall conduct a thorough assessment that includes determining and documenting the business requirements and/or problem they are seeking to address, along with associated AI security risks and mitigation strategies.

1.1.2. Provision 5.1.2-1.1

Where the Data Custodian is part of a Developer's organization, they shall be included in internal discussions when determining the requirements and data needs of an AI system.

1.1.3. Provision 5.1.2-2

Developers and System Operators shall ensure that AI systems are designed and implemented to withstand adversarial AI attacks, unexpected inputs and AI system failure.

1.1.4. Provision 5.1.2-3

To support the process of preparing data, security auditing and incident response for an AI system, Developers shall document and create an audit trail in relation to the AI system. This shall include the operation, and lifecycle management of models, datasets and prompts incorporated into the system.

1.1.5. Provision 5.1.2-4

If a Developer or System Operator uses an external component, they shall conduct an AI security risk assessment and due diligence process in line with their existing software development processes, that assesses AI specific risks.

1.1.6. Provision 5.1.2-5

Data Custodians shall ensure that the intended usage of the system is appropriate to the sensitivity of the data it was trained on as well as the controls intended to ensure the security of the data.

1.1.7. Provision 5.1.2-5.1

Organizations should ensure that employees are encouraged to proactively report and identify any potential security risks in AI systems and ensure appropriate safeguards are in place.

1.1.8. Provision 5.1.2-6

Where the AI system will be interacting with other systems or data sources, (be they internal or external), Developers and System Operators shall ensure that the permissions granted to the AI system on other systems are only provided as required for functionality and are risk assessed.

1.1.9. Provision 5.1.2-7

If a Developer or System Operator chooses to work with an external provider, they shall undertake a due diligence assessment and should ensure that the provider is adhering to the present document.

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 5.1.2-1

"As part of deciding whether to create an AI system, a System Operator and/or Developer shall conduct a thorough assessment that includes determining and documenting the business requirements and/or problem they are seeking to address, along with potential AI security risks and mitigation strategies." (ETSI TS 104 223 [i.1])

Related threats/risks:

Without assessing whether an AI system is required to meet the business requirements, systems can be unnecessary or poorly suited for their environment, leading to lack of compliance, unnecessary complexity, increased attack surface, unexpected behaviour, and security vulnerabilities.

Example Measures/Controls 1:

Conduct Business Alignment Review: Use ISO/IEC 25059 [i.16] as a quality model to review and document business requirements for the AI system to ensure that design choices align with the organization's needs and objectives.

Example Measures/Controls 2:

Perform Risk Assessment: Conduct and document an AI-specific risk assessment covering data classifications, logging risks of personal data and their mitigations in DPIAs. Cover expected data volume, types of integration, the model's complexity, architecture, and number of parameters. For more information on these risk factors see the NCSC Principles for Machine Learning [i.5] and NCSC Guidelines for Secure AI system development [i.6].

Example Measures/Controls 3:

Integrate Risk Management and Governance Frameworks: Embed AI system assessments within both a formal Risk Management Framework (RMF) and the AI governance structure to ensure thorough risk evaluation, consistent mitigation, and monitoring before key organizational decisions.

1.2.2. Provision 5.1.2-1.1

"Where the Data Custodian is part of a Developers organization, they shall be included in internal discussions when determining the requirements and data needs of an AI system." (ETSI TS 104 223 [i.1])

Related threats/risks:

Failure to include the Data Custodian in discussions about AI system requirements and data needs can result in noncompliance with data governance policies, inappropriate data usage, or insufficient safeguards for sensitive data, increasing the risk of data breaches or regulatory violations.

Example Measures/Controls:

Ensure collaboration with the Data Custodian: During the design and development phases to define data requirements to identify regulatory compliance requirements. Ensure that Data Custodians are able to balance additional risks to data that come from the AI system with intended mitigations and the business need.

1.2.3. Provision 5.1.2-2

"Developers and System Operators shall ensure that AI systems are designed and implemented to withstand adversarial AI attacks, unexpected inputs and AI system failure." (ETSI TS 104 223 [i.1])

Related threats/risks:

Organizations are not always successful in preventing breaches and so defence in depth requires planning for and responding to compromise. The absence of defence in depth can lead to infringement of applicable regulation.

Example Measures/Controls:

Apply Secure by Design Principles: Integrate security into the AI system's design phase by conducting threat modelling. Threat modelling covers both traditional cyber threats and AI-specific ones that might be introduced by the design choices. Incorporate standardized security controls in the system design controls to mitigate risks. Document each standardized control used in the design phase, and ensure it is integrated with specific test cases to verify its effectiveness during system testing. Ensure monitoring controls as well as incident response and recovery from failures are addressed in threat mitigation and they are documented in the system's design.

1.2.4. Provision 5.1.2-3

"To support the process of preparing data, security auditing and incident response for an AI system, Developers shall document and create an audit trail in relation to the AI system. This shall include the operation, and life cycle management of models, datasets and prompts incorporated into the system." (ETSI TS 104 223 [i.1])

Related threats/risks:

A lack of audit trails can lead to untraceable changes or unauthorized adjustments, complicating incident response, forensic investigations, and regulatory compliance.

Example Measures/Controls:

Automated Audit Trails for ML Operations (MLOps) and System changes: Implement automated logging for all critical operations related to model training, dataset changes, prompts and parameter adjustments. For critical systems with compliance requirements, use WORM (write-once, read-many) storage to store logs, ensuring they remain tamperproof and accessible for audits.

1.2.5. Provision 5.1.2-4

"If a Developer or System Operator uses an external component they shall conduct an AI security risk assessment and due diligence process in line with their existing software development processes, that assesses AI specific risks." (ETSI TS 104 223 [i.1])

Related threats/risks:

Third-party components introduce risks through possible vulnerabilities in the external vendor's security practices which might not be to the same standard. This includes operating systems and libraries, container images, programming packages as well as models and datasets. Large models contain general purpose functionality, ensure that specific risks are mitigated or otherwise managed. In the context of product safety regulation, risks to AI security include risks that can lead to hazardous situations. Hazardous situations can include risks to fundamental rights, health and safety.

For further information about risk management in the context of the EU's AI Act, see JTC21024 [i.146].

Example Measures/Controls:

Security Due-Diligence for External Components: Mandate a risk assessment process before a component (including external models) can be used, covering provenance, known risks, and when personal data is used a DPIA. Safeguard provenance by mandating in internal standards that components can only be sourced by trusted and approved sources, documenting source, version, licencing, history, and other related artifacts (e.g. Model Card for models); use checksums to verify integrity.

1.2.6. Provision 5.1.2-5

"Data Custodians shall ensure that the intended usage of the system is appropriate to the sensitivity of the data it was trained on as well as the controls intended to ensure the security of the data." (ETSI TS 104 223 [i.1])

Related threats/risks:

Misalignment between the intended usage of the AI system and the sensitivity of the data it was trained on can result in inappropriate data exposure, inadequate security controls, and regulatory non-compliance, leading to potential data breaches and misuse of personal data or other confidential information.

Example Measures/Controls:

Ensure Data Custodian Assurance: Require Data Custodians to review system's intended usage and the data security controls to ensure compliance and balancing these risks with business needs.

1.2.7. Provision 5.1.2-5.1

"Organizations should ensure that employees are encouraged to proactively report and identify any potential security risks in AI systems and ensure appropriate safeguards are in place." (ETSI TS 104 223 [i.1])

Related threats/risks:

A lack of proactive reporting and identification of security risks in AI systems can lead to undetected vulnerabilities, increasing the likelihood of security breaches, data leaks, or misuse of AI, with potentially significant operational, financial, and reputational consequences.

Example Measures/Controls:

Support proactive reporting of security risks. Establish a clear, accessible process for employees to report potential security risks in AI systems, encourage a culture of proactive risk identification by providing training, communication channels, transparent handling, and recognition for reporting issues.

1.2.8. Provision 5.1.2-6

"Where the AI system will be interacting with other systems or data sources, (be they internal or external), Developers and System Operators shall ensure that the permissions granted to the AI system on other systems are only provided as required for functionality and are risk assessed." (ETSI TS 104 223 [i.1])

Related threats/risks:

There is huge potential and interest in "agentic systems", where an AI system can decide and conduct its own actions, typically through integrations with other systems. The actions taken by an AI system are not fully predictable, and can be coerced by an attacker. This risk introduces the potential for unauthorized access, data exfiltration, and privilege escalation.

Example Measures/Controls:

Least-privilege access to data and systems accessed by AI System. Mandate a risk assessment process before a component can be used covering provenance, known risks, and evaluations. Ensure that the assessment covers all possible model states, not just the designed or expected ones.

1.2.9. Provision 5.1.2-7

"If a Developer or System Operator chooses to work with an external provider, they shall undertake a due diligence assessment and should ensure that the provider is adhering to the present document." (ETSI TS 104 223 [i.1])

Related threats/risks:

Collaborating with external providers without assessing their adherence to ETSI TS 104 223 [i.1] can lead to increased vulnerabilities, lack of regulatory compliance, insecure systems, or inadequate response protocols, which could compromise the entire system's security.

Example Measures/Controls:

Security Review of External Providers: Verify the external provider's implementation of ETSI TS 104 223 [i.1] with the external provider and their overall regulatory compliance.

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. 5.2.2 - Enhance data quality - 2

A general method in pre-processing data as a means to minimize risks of data manipulations was proposed in [i.15]. It tries to estimate the quality of candidate training data and only uses data of sufficient quality for training the model. The authors consider the use-case of threat intelligence systems and use semi-supervised learning to judge the trustworthiness of data. The limitation of this approach consists in that it mostly considers increasing data quality with respect to benign problems (such as unintentional wrong labelling) and does not address targeted and optimized attacks.

1.3.2. 5.2.2 - Data sanitisation - 2

TRIM [i.12] iteratively estimates the model parameters and trains on the subset of best-fitting input points at the same time, until convergence is reached. Its results are much better than those of RONI. However, TRIM is devised for linear regression only and thus not applicable to (deep) neural networks.

1.3.3. 5.2.2 - Data sanitisation - 3

Provenance-based [i.13]: another extension of RONI uses (presumably correct) meta data about data provenance and first clusters data accordingly. Due to the additional information used, this approach achieves better results than RONI and is more efficient by a certain factor (essentially, the average cluster size), since the model does not need to be retrained for each individual point but only the cluster centroids. The intuition of [i.13] is that for data points of common provenance the probability of being poisoned is strongly correlated. Applying this technique assumes the existence and correctness of information on data provenance.

1.3.4. 5.2.2 - Data sanitisation - 4

Keyed Non-parametric Hypothesis Tests (KNHT) [i.17] assumes a set of clean training data which describe intended data distribution and inspect newly-collected ones. The rationale is to compare the two set of data and if their similarity is insufficient, the newly collected ones are further inspected. KNHT does the data distribution comparison after mapping the two data distributions into another space via a set of functions with secret keys.

1.3.5. 5.2.2 - Blocking poisoning

Gradient shaping [i.14] starts from the new insight that poisoning (and backdoor) attacks exhibit larger gradients with differing orientations for poisoned data as compared to clean data. Hence, gradient shaping aims to prevent these properties in the gradients from occurring during model training (e.g. by cropping large values). The gradient shaping technique does not use additional information about the training dataset, and its run-time overhead is negligible. The results presented in [i.14] look quite promising for several attacks, but like other approaches, the strategy cannot thwart strong state-of-the-art attacks.

1.3.6. 5.3.2 Model enhancement mitigations against backdoor attacks - 1

Spectral signature [i.18]: it is observed that spectral signatures can be used to identify and remove poisoned inputs, where spectral signatures are the spectrum of the covariance of a feature representation learned by the neural network.

1.3.7. 5.3.2 Model enhancement mitigations against backdoor attacks - 2

Activation clustering [i.19]: triggers can be detected from the neuron activation in the final hidden layer of a network. It is observed that input with triggers have different patterns of neuron activation from benign input. Hence, by using unsupervised clustering on neuron activation in the final hidden layer, input with triggers can be identified. The addressed model can then be retrained excluding triggers.

1.3.8. 5.3.2 Model enhancement mitigations against backdoor attacks - 3

Retraining [i.20]: retraining the model with a small subset of clean training data is explored for correcting backdoor-embedded neural networks.

1.3.9. 5.3.2 Model enhancement mitigations against backdoor attacks - 4

Fine-pruning [i.22]: pruning a neural network eliminates some less important neurons [i.21]. It is a means of raising the success bar of backdoor attacks at some cost of accuracy. Pruning technique is effective when attackers are unaware of pruning protection. Fine-pruning retrains the model after pruning to address pruning-aware backdoor attacks. By applying fine-pruning, a backdoor-embedded neural network can be restored.

1.3.10. 5.3.2 Model enhancement mitigations against backdoor attacks - 5

Neural-Cleanse [i.23]: small input perturbations are reverse engineered from the model. Those perturbations trigger backdoor behavior in the model and then a backdoored class can be identified. The addressed model can then be patched by retraining with purified data. This process does backdoor un-learning. Neural cleanse method relies on an assumption that the trigger for the backdoor-infected class is smaller than the median size of the reverse engineered trigger for all classes. This assumption fails when triggers have various sizes.

1.3.11. 5.3.2 Model enhancement mitigations against backdoor attacks - 6

TABOR [i.24]: the backdoor detection task is transformed to a non-convex optimization problem with a newly-designed objective function to narrow down search space. Hence, input with triggers can be efficiently reverse-engineered. After triggers are found, by retraining the model with purified data, the model can be restored.

1.3.12. 5.3.3 Model-agnostic mitigations against backdoor attacks - 1

STRIP [i.25]: a runtime trigger detection scheme is designed based on an assumption that an input with trigger is insensitive to perturbations. By applying a set of perturbations, a given inference input is then represented as a set of data samples. The classification entropy introduced by the set of data samples is a quantitative measure on how likely the given inference input contains a trigger. Inputs with low classification entropy even when applied strong perturbations are unlikely to be benign. The assumption is not universal. But this method does not require to know model parameters and can be performed at run-time.

1.3.13. 5.3.3 Model-agnostic mitigations against backdoor attacks - 2

SUTO-NEO [i.27]: it is a trigger detection and prediction restoration scheme for image classification task. With an assumption that only one trigger exists and the trigger location is fixed on the image, an input image can be analyzed to see if a trigger exists. Given an input image, the potential trigger location can be searched by comparing classification results of the intact image and the image with a dominant color block at a random position. If the results diverse, the position likely has a trigger. To further confirm about it, the image block covered by the color block is implanted to other known-results training images. If those known-result training images with the patched image block have different prediction results from before, the position and the trigger are confirmed. To restore the classification result, the image with a trigger is patched with a block of the dominate color and then the corresponding predication can be restored.

1.3.14. 5.3.3 Model-agnostic mitigations against backdoor attacks - 3

SentiNet [i.26]: by combining techniques of model interpretability and object detection, SentiNet is a mean of trigger detection at run-time against localized and universal backdoor attacks on image classifiers. It can detect more than one trigger with various sizes.

1.3.15. 5.3.3 Model-agnostic mitigations against backdoor attacks - 4

Februus [i.28]: researchers proposed a trigger deactivation method by input restoration technique against input-agnostic triggers. They analyzed which region of the input has the most impact on the classification result via visual explanation tool GradCAM [i.29] and then deactivate potential triggers. The most influential region is replaced by a neutralized color box and then the masked input is restored by generative adversarial networks. This method needs access to not only the model but also the training dataset.

1.3.16. 5.3.3 Model-agnostic mitigations against backdoor attacks - 5

Trigger reverse engineering: as described earlier, neural cleanse method [i.23] and TABOR [i.24] both consist of two steps. The first step is to reverse engineer triggers from the addressed model and the second step is to retrain the model by purified data. Although at deployment stage, the retraining step cannot be performed, the knowledge of triggers from the first step helps on detecting backdoored models.

1.3.17. 5.3.3 Model-agnostic mitigations against backdoor attacks - 6

Meta Neural Trojan Detection (MNTD) [i.30]: a meta neural analysis framework is proposed for classifying benign models and backdoored models. A meta classifier can be built to classify models by representing a model as a sequence of predictions from fine-tuned queries. This method needs to build shadow models from clean datasets and self-generated backdoored datasets. But the method only requires query access to the addressed model with zero knowledge of model parameters.

1.3.18. 5.3.3 Model-agnostic mitigations against backdoor attacks - 7

DeepInspect [i.31]: it detects backdoor without knowledge of model parameters and clean or training datasets. The method consists of three steps: model inversion to get a surrogate training dataset, trigger generation by conditional generative adversarial networks, and anomaly detection based on statistical hypothesis testing. DeepInspect detects backdoor in models with less prior knowledge at a cost of slightly lower detection rate.

1.3.19. 5.3.3 Model-agnostic mitigations against backdoor attacks - 8

Universal Litmus Patterns (ULPs) [i.32]: similar to MNTD, Universal Litmus Patterns fine-tunes a set of optimized input images for determining whether a model contains backdoors. Although ULPs cannot deactivate triggers or backdoors, it provides an efficient and effective backdoor detection technique against single trigger-based backdoor attacks.

1.3.20. 6.2.1 - Transferability

Transferability is a feature of adversarial examples. Some adversarial examples crafted for one model can fool a different model with a high probability. They are called transferable adversarial examples. The work in [i.33] demonstrates the transferability of adversarial examples among several machine learning models. A later work in [i.34] generates adversarial examples with better transferability.

The area of mitigations against evasion attacks has been rapidly evolving with new mitigations being proposed and broken at high speed. On the one hand, most mitigations empirically target very specific attack types, but are easily broken once the attacker can adapt to known mitigations, as has been repeatedly demonstrated [i.35], [i.36] and [i.37]. Such mitigations, even if they can be broken by adaptive attacks, can still be sufficient for certain use cases, since adaptive attacks take more efforts on the part of the attacker and are hence less likely to occur in practice. On the other hand, some mitigations are derived from theoretic analysis and formal reasoning in a certain threat model and hence remain effective against a wide range of attacks, including certain adaptive attacks. However, if the threat model does not capture the practical constraints of attacks sufficiently well, attackers can circumvent the threat model easily, and in this case these mitigations can be just as vulnerable as empirical ones. It is, for instance, well-known that the perturbation metrics used in threat models do not reflect human perception very well. In addition, there is a trade-off between the robustness and invariance to perturbations a model can have, which can likewise be exploited [i.38].

1.3.21. 6.2.2 - 1

Transformation: to narrow down the attack surface of manipulating inputs, some data transformation techniques are applied. Adversarial perturbations are expected to be diminished through data transformation. For images, researchers took data compression as defense, for example, using Joint Photographic Experts Group (JPEG) for input data transformation [i.39]. Later on more data transformation techniques are proposed [i.40] as defense, including low-pass filtering, JPEG compression, Principal Components Analysis (PCA), soft-thresholding, patchwise PCA, low resolution wavelet approximation, against gradient-based attacks and concluded that JPEG compression tends to outperform others in most experimental cases. However, it has been [i.41] shown how to generate adversarial examples which survive JPEG compression. The compression defense [i.42] is advanced by combining JPEG compression and random compression level and showed that the new technique is effective against Carlini-Wagner's (CW) attack and DeepFool attack. Researchers [i.43] also studied data transformation as defense, including cropping-rescaling, bit-depth reduction, JPEG compression, total variance minimization, and image quilting on ImageNet and show that total variance minimization and image quilting are effective.

1.3.22. 6.2.2 - 2

NULL labelling [i.44]: for classifiers, instead of denoising adversarial inputs, NULL labelling technique lets the addressed model learn how likely an input is adversarial by adding a NULL label to the label space. As a result, adversarial examples are classified as NULL with a high probability while the model accuracy remains.

1.3.23. 6.2.2 - 3

Adversarial training is one of the most prominent techniques explored for mitigating evasion attacks. Adversarial training consists in training the model on a training data set augmented with correctly labelled adversarial examples, thus making sure the model will not be fooled by adversarial examples. The adversarial examples used for training can be computed using a variety of methods for mounting evasion attacks, such as the Fast Gradient Sign Method (FGSM) [i.45]. Some existing work [i.46] has shown that using the Projected Gradient Descent (PGD) method creates models with quite good robustness against evasion attacks, including adaptive attacks. This is probably due to the fact that PGD can be seen as a generalization of other attack methods, and can hence find the strongest adversarial examples for a given threat model with high probability. The comparatively good effectiveness of adversarial training using PGD compared to other defense methods is also confirmed by [i.47].

Still, adversarial training only guarantees a certain level of robustness against evasion attacks, which is not sufficient for some applications. Adversarial training also increases the run-time for training by a certain factor that depends on the level of robustness provided.

1.3.24. 6.2.2 - 4

Regularization builds on the idea that preventing small input perturbations from changing model output can defend against adversarial examples. In mathematical terms this translates to bounding the Lipschitz constant [i.48] of the function implemented by the neural network or its gradient [i.49] by adapting the training procedure. In fact, there is a strong relation between these two approaches, since every differentiable function with bounded gradient is Lipschitz continuous (although the implied constant can be large, and the function in question is not everywhere differentiable).

Regularization can certifiably improve robustness, but faces several limitations [i.50]. In many cases, the bounds on the Lipschitz constant or gradient, respectively, are not strictly enforced, but only encouraged by penalty terms. If the bounds are indeed enforced, this impacts the generalization performance of the network and thus its overall accuracy on complex tasks.

1.3.25. 6.2.2 - 5

Certifiable training: in the research line of robustness verification, certifiable training techniques attempt to train neural networks to improve the lower bound of robustness. Existing methods include designing specific regularizations, such as Lipschitz regularization, and randomized smoothing. Randomized smoothing technique [i.51] is derived from probabilistic robustness verification. For classifiers, a model is firstly trained by the training dataset augmented with Gaussian noises, where the output is a probability distribution over all classes. The final model is trained via the augmented dataset with labels indicating the class with the highest probability. Via randomized smoothing techniques, the derived model has probabilistic verifiable robustness.

1.3.26. 6.2.2 - 6

Gradient masking: to defend against gradient-based attacks, gradient masking is one approach. For example, shattered gradients introduce non-differentiable operations such that attackers cannot get correct gradients for crafting adversarial examples. Another example is stochastic gradients which introduce randomness on gradients. However, the results of [i.35] and [i.36] pointed out that gradient masking approach can be overcome effectively. To recover the gradient signals from shattered gradients, backward pass differentiable approximation techniques are proposed to approximate gradients [i.36]. To mitigate the uncertainty of stochastic gradients, expectation over transformation techniques are proposed to estimate gradients [i.35]. Hence, gradient masking approach is much less effective against gradient-based attacks.

1.3.27. 6.2.2 - 7

Distillation network [i.52]: it is also called defensive distillation. Distillation network is proposed as robust training technique for classifiers against adversarial examples. To train the distilled network, a teacher model is firstly trained on a training dataset. The distilled network is then trained on the same input and the corresponding output probabilities of the teacher network. The output probabilities are also called soft labels while the original labels of the training dataset are hard labels. Distillation network has been shown less effective against adaptive attackers [i.53].

1.3.28. 6.2.2 - 9

Model verification for certifiable robustness: Certifiable robustness is a conceptual theoretic description capturing model robustness against evasion attacks. An adversarial example is a crafted input close enough to an intact input where they are somewhat indistinguishable to human, yet the addressed model gives them different output. Figure 5 gives a conceptual illustration, where x is an input, x' is an adversarial example, y and y' are model output for x and x' respectively. The distance between x and x' is evaluated by certain metrics, such as L_p distance. The addressed model is the function mapping input space to output space. Certifiable robustness attempts to certify a lower bound of the perturbation range δ where no adversarial example exists. Existing examples include certifiable robustness for CNN [i.54], RNN [i.55] and GNN [i.56].

1.3.29. 6.2.3 - 1

Input transformation: for image classifiers, image transformations, such as rotation and shifting, are proposed to detect adversarial examples by constructing a detector from the training dataset with image transformations [i.69]. Experimental results show the effectiveness by the detection rate on MNIST and CIFAR-10 against FGSM and CW evasion attacks. Another example of image transformation is feature squeezing [i.70], such as reducing the color bit depth of each pixel and spatial smoothing. One more example is adding random perturbation. An adversarial example is close to an input within a certain bounded range, but the addressed model gives different outputs. When a given input is indeed an adversarial example, adding some perturbation on the given input can push the output back to the original class with a non-negligible probability. Based on this rationale, researchers [i.71] proposed adding random perturbations on input and analysing the resulting output to detect adversarial examples. Their experimental results show the effectiveness on MNIST, CIFAR10 and ImageNet against BIM, DeepFool and CW evasion attacks at the cost of generating a detector from the original training dataset.

1.3.30. 6.2.3 - 2

Statistics: to detect adversarial examples, some research seek for proper statistics. For example, for convolution neural networks, binary classifiers are constructed from the statistics of convolutional layer results from normal training dataset and adversarial examples [i.72]. Experimental results show the effectiveness on ILSVRC2012 (ImageNet-based) dataset with AlexNet and VGGNet network architectures against L-BFGS attacks at the cost of constructing a binary classifier. This method needs knowledge of training dataset. Another example of statistics method uses the expected perturbed log-odds over random perturbations to test if a given input x classified as class y is manipulated while the true class is z [i.73]. This method needs to access the results before the final layer at runtime. Experimental results show the effectiveness on CIFAR-10 and ImageNet against PGD attacks at the runtime cost of operating multiple inference for one input.

1.3.31. 6.2.3 - 4

Quantization methods [i.76] are proposed to diminish adversarial perturbations from input at inference time. The challenge is to find a proper quantization level such that perturbations are diminished while model accuracy remains. Trainable quantization method finds the suitable quantization level during training phase and applies it during inference phase. Experimental results show the effectiveness on MNIST and CIFAR-10 against FGSM, CW and JSMA attacks.

1.3.32. 6.2.3 - 5

Ensemble: for classifiers, ensemble methods are proposed to mitigate adversarial perturbations. The assumption is that some adversarial examples misleading one classifier do not mislead other classifiers. Ensemble methods construct a set of classifiers to classify a new data input by a combination of their predictions. Researchers [i.77] proposed four ensemble constructions for model robustness, including random initial model parameters, similar but different network architectures, bagging, and adding Gaussian noise on training dataset. Their experimental results on MNIST and CIFAR-10 show that ensemble methods are effective against FGSM and BIM attacks. Another ensemble construction [i.78] is proposed by using different loss functions in the set of classifiers. Experimental results show that the ensemble method has better robustness against FGSM-based attacks at the cost of model accuracy on clean dataset. Some adversarial examples [i.33] can transfer among models, which are thus immune to ensemble methods.

1.3.33. 6.3.3 - 2

Boundary differential privacy (ϵ -BDP) [i.83]: this method protects against model extraction attacks that use fine-tuned queries with differential property to infer the decision boundary of the addressed model. The proposed solution is to obfuscate the prediction responses near the decision boundary. Based on a perturbation algorithm called boundary randomized response, the ϵ -BDP method prevents adversaries from learning the decision boundary by a predefined precision regardless the amount of queries issued to the model prediction API. The technique is effective for both linear and non-linear models.

1.3.34. 6.3.3 - 4

Rounding confidences [i.85]: with the aim to limit the information provided by model query APIs, this technique proposes to round confidence scores of the target model to some fixed precision. For decision trees, rounding confidence scores increase the chance of node identifier collisions and thus reduces attacker's success rate. The defense has also shown its effectiveness against extraction attacks in decision trees, but not in some other models.

1.3.35. 6.3.3 - 5

Ensemble [i.85]: ensemble methods such as aggregation of predictions by a number of individual models, can be used to defend against model reverse engineering attack. In this research paper, the author found that ensemble methods' resilience to extraction attacks is higher, since attackers will only be able to obtain relatively coarse approximations of the target function.

1.3.36. 6.3.3 - 6

Fingerprinting based on conferrable adversarial examples [i.87]: this defense addresses DNN-based classifiers and it extracts a set of inputs from the source model so that only surrogate models agree with the source model on the classification of such inputs, whereas (benign) reference models relative to the source model do not. These inputs called conferrable adversarial examples are a subclass of transferable adversarial examples that exclusively transfer with a target label from a source model to its surrogates. While this fingerprinting scheme is robust to almost all derivation and extraction attacks, strong adaptive attacks can still remove the fingerprint. To implement this protection, the defender has the full knowledge of the source model parameters, query access to the target model deployed by the attacker and a limited number of queries to verify whether the target model is a surrogate.

1.3.37. 6.3.3 - 7

Fingerprinting the classification boundary [i.88]: this technique uses gradient descent method to find fingerprinting data points near the classification boundary of the addressed DNN-based classifier. These data points together with their predicted labels are used as the fingerprint for the addressed classifier. The model owner can identify that a suspect classifier is potentially pirated from the addressed model by querying its prediction API to get the labels of the fingerprinting data points. If most of the suspect classifier's predictions match those of the fingerprint for the owner's model, then the IP is compromised with a high probability. Authors claim that unlike watermarking, the technique does not incur in any accuracy loss for the addressed classifier, since it does not tamper its training or fine-tuning process at all.

1.3.38. 6.4.2 - 2

Adversarial learning with privacy: to provide data privacy, one approach is to train a data representation from the original training dataset for the model training at a later time. The trained data representation reveals less sensitive information of training data set while keeping model accuracy as much as possible. One example [i.90] is addressing medical records in text format. Medical records are converted to another representation with less identity information. The process is also called de-identification. The de-identification process is composed by three training stages, including training an initial representation, training a potential adversary, retraining the representation against such adversary. By using de-identified data representation, given two data representations, an adversary cannot distinguish if they are converted from the same input data. Another example [i.91] is addressing facial images. The representation generation of training data set is trained with two objectives of optimizing model accuracy and maximizing adversarial reconstruction loss. The adversarial reconstruction loss is defined by predictions of a discriminator and a perceptual distance. Hence, the obtained data representation of the training dataset reveals less information while keeping model accuracy.

1.3.39. 6.4.2 - 3

DP-SGD [i.93]: differential privacy [i.92] is a framework for quantitatively measuring the privacy guarantees provided by an algorithm. The basic idea is to add randomness to algorithm's behaviours. Learning with differential privacy provides provable privacy guarantees and mitigating the risks of exposing sensitive information about training data. For a model trained with differential privacy, the behaviours of the trained model is less affected by any single training data. Hence, it is difficult to tell which data record belongs to the training dataset by observing the model's behaviour. By using the framework of differential privacy, Differentially Private Stochastic Gradient Descent (DP-SGD) technique is proposed to train a model with provable membership privacy. DP-SGD adds noises to the gradients used in Stochastic Gradient Descent (SGD), which is the core of almost all deep learning algorithms. The parameters of the trained model are iteratively updated using random noises and gradients. The privacy comes from the added noises. However, DP-SGD imposes a significant classification accuracy loss for protecting large models on high dimensional data.

1.3.40. 6.4.2 - 4

Adversarial regularization [i.94] mitigates membership inference attacks through adversarial training. The model's predictions on the members of its training dataset are indistinguishable from its predictions on the nonmembers of its training dataset from the same distribution. This requirement is captured as a min-max game optimization problem. The adversarial training algorithm is designed to train a model that minimizes both the prediction error and the maximum gain of the membership inference attack against it.

1.3.41. 6.4.3 - 1

MemGuard [i.95] add noises to confidence scores to mitigate membership inference attacks. Assume attackers use a classifier to distinguish whether an input is in the training dataset. MemGuard crafts noises such that the manipulated confidence scores become adversarial examples to attacker's classifier. As a result, the attacker cannot successfully launch membership inference attack.

1.3.42. 6.4.3 - 2

Prediction purification [i.96] purifies the confidence score vectors predicted by the target classifier by reducing their dispersion. As a result, attackers get less information from purified confidence score vectors. Prediction purifications can be built to mitigate only membership inference attack, only model inversion attack and both attacks. Moreover, from experimental results, prediction purification against membership inference attack is effective against model inversion attack and vice versa. The generation of purifiers needs a subset of training dataset. The generated purifier is composed with the addressed classifier model to purify the confidence score vectors. Experimental results show prediction purification can reduce the membership inference accuracy and increase the model inversion error at the cost of little classification accuracy drop and little distortion to the confidence scores.

1.4. SAI 002 - Securing Artificial Intelligence (SAI); Data Supply Chain Security

Link: https://www.etsi.org/deliver/etsi_gr/SAI/001_099/002/01.01.01_60/gr_SAI002v010101p.pdf

Link: https://www.etsi.org/deliver/etsi_gr/SAI/001_099/002/01.01.01_60/gr_SAI002v010101p.pdf

1.4.1. 6.1.2 Cybersecurity hygiene - 4

Strong access controls should be in place, applying the principle of least privilege. These stand alongside limits to the number of queries allowed to be made against a model in a period of time.

1.4.2. 6.1.3 Supply chain security - 5

Using additional security (for example cryptographic protection of data) to protect the most critical functions.

1.4.3. 6.5 - Following standard cybersecurity good practice

Including following the principle of least privilege when accessing data.

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 - 5

When building agents, evaluate use of "intent capsule", an emerging pattern to bind the declared goal, constraints, and context to each execution cycle in a signed envelope, restricting run-time use.

2.1.2. ASI02: Tool Misuse and Exploitation - 1

Least Agency and Least Privilege for Tools. Define per-tool least-privilege profiles (scopes, maximum rate, and egress allowlists) and restrict agentic tool functionality and each tool's permissions and data scope to those profiles – e.g., read-only queries for databases, no send/delete rights for email summarizers, and minimal CRUD operations when exposing APIs. Where possible, express these profiles as IAM or authorization policy stanzas attached to each tool, rather than relying on ad-hoc conventions.

2.1.3. ASI03: Identity and Privilege Abuse - 1

Enforce Task-Scoped, Time-Bound Permissions: Issue short-lived, narrowly scoped tokens per task and cap rights with permission boundaries - using per-agent identities and short-lived credentials (e.g., mTLS certificates or scoped tokens) - to limit blast radius, block delegated-abuse and maintenance-window attacks, and mitigate un-scoped inheritance, orphaned privileges, and reflection-loop elevation.

2.1.4. ASI03: Identity and Privilege Abuse - 2

Isolate Agent Identities and Contexts: Run per-session sandboxes with separated permissions and memory, wiping state between tasks to prevent Memory-Based Escalation and reduce Cross-Repository Data Exfiltration.

2.1.5. ASI03: Identity and Privilege Abuse - 3

Mandate Per-Action Authorization: Re-verify each privileged step with a centralized policy engine that checks external data, stopping Cross-Agent Trust Exploitation and Reflection Loop Elevation.

2.1.6. ASI03: Identity and Privilege Abuse - 5

Define Intent: Bind OAuth tokens to a signed intent that includes subject, audience, purpose, and session. Reject any token use where the bound intent doesn't match the current request.

2.1.7. ASI03: Identity and Privilege Abuse - 6

Evaluate Agentic Identity Management Platforms. Major platforms integrate agents into their identity and access management systems, treating them as managed non-human identities with scoped credentials, audit trails, and lifecycle controls. Examples include Microsoft Entra, AWS Bedrock Agents, Salesforce Agentforce, Workday's Agentic System of Record (ASOR) model, and similar emerging patterns in Google Vertex AI.

2.1.8. ASI03: Identity and Privilege Abuse - 7

Bind permissions to subject, resource, purpose, and duration. Require re-authentication on context switch. Prevent privilege inheritance across agents unless the original intent is re-validated. Include automated revocation on idle or anomaly.

2.1.9. ASI03: Identity and Privilege Abuse - 8

Detect Delegated and Transitive Permissions: Monitor when an agent gains new permissions indirectly through delegation chains. Flag cases where a low-privilege agent inherits or is handed higher-privilege scopes during multi-agent workflows.

2.1.10. ASI04: Agentic Supply Chain Vulnerabilities - 9

Zero-trust security model in application design: design system with security fault tolerance that assumes failure or exploitation of LLM or agentic function components.

2.1.11. ASI06: Memory & Context Poisoning - 6

Prevent automatic re-ingestion of an agent's own generated outputs into trusted memory to avoid self-reinforcing contamination or "bootstrap poisoning."

2.1.12. ASI06: Memory & Context Poisoning - 8

Expire unverified memory to limit poison persistence.

2.1.13. ASI06: Memory & Context Poisoning - 9

Weight retrieval by trust and tenancy: Require two factors to surface high-impact memory (e.g., provenance score plus human-verified tag) and decay low-trust entries over time.

2.1.14. ASI07: Insecure Inter-Agent Communication - 9

Typed contracts and schema validation: Use versioned, typed message schemas with explicit per-message audiences. Reject messages that fail validation or attempt schema down-conversion without declared compatibility. Typed contracts help with structure, but semantic divergence across agents remains an inherent challenge; mitigations therefore focus on integrity, provenance, and controlled communication patterns rather than attempting full semantic alignment.

2.1.15. ASI08: Cascading Failures - 1

Zero-trust model in application design: design system with fault tolerance that assumes availability failure of LLM:2025, agentic function components and external sources.

2.1.16. ASI08: Cascading Failures - 4

Independent policy enforcement: Separate planning and execution via an external policy engine to prevent corrupt planning from triggering harmful actions.

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 - 1

Provide specific instructions about the model's role, capabilities, and limitations within the system prompt. Enforce strict context adherence, limit responses to specific tasks or topics, and instruct the model to ignore attempts to modify core instructions.

2.2.2. LLM01: Prompt Injection - 4

Provide the application with its own API tokens for extensible functionality, and handle these functions in code rather than providing them to the model. Restrict the model's access privileges to the minimum necessary for its intended operations.

2.2.3. LLM03: Supply Chain - 2

Understand and apply the mitigations found in the OWASP Top Ten's "A06:2021 – Vulnerable and Outdated Components." This includes vulnerability scanning, management, and patching components. For development environments with access to sensitive data, apply these controls in those environments, too.

2.2.4. LLM03: Supply Chain - 10

Encrypt models deployed at AI edge with integrity checks and use vendor attestation APIs to prevent tampered apps and models and terminate applications of unrecognized firmware.

2.2.5. LLM04: Data and Model Poisoning - 4

Tailor models for different use cases by using specific datasets for fine-tuning. This helps produce more accurate outputs based on defined goals.

2.2.6. LLM06: Excessive Agency - 1

Limit the extensions that LLM agents are allowed to call to only the minimum necessary. For example, if an LLM-based system does not require the ability to fetch the contents of a URL then such an extension should not be offered to the LLM agent.

2.2.7. LLM06: Excessive Agency - 2

Limit the functions that are implemented in LLM extensions to the minimum necessary. For example, an extension that accesses a user's mailbox to summarise emails may only require the ability to read emails, so the extension should not contain other functionality such as deleting or sending messages.

2.2.8. LLM06: Excessive Agency - 3

Avoid the use of open-ended extensions where possible (e.g., run a shell command, fetch a URL, etc.) and use extensions with more granular functionality. For example, an LLM-based app may need to write some output to a file. If this were implemented using an extension to run a shell function then the scope for undesirable actions is very large (any other shell command could be executed). A more secure alternative would be to build a specific filewriting extension that only implements that specific functionality.

2.2.9. LLM06: Excessive Agency - 4

Limit the permissions that LLM extensions are granted to other systems to the minimum necessary in order to limit the scope of undesirable actions. For example, an LLM agent that uses a product database in order to make purchase recommendations to a customer might only need read access to a 'products' table; it should not have access to other tables, nor the ability to insert, update or delete records. This should be enforced by applying appropriate database permissions for the identity that the LLM extension uses to connect to the database.

2.2.10. LLM06: Excessive Agency - 5

Track user authorization and security scope to ensure actions taken on behalf of a user are executed on downstream systems in the context of that specific user, and with the minimum privileges necessary. For example, an LLM extension that reads a user's code repo should require the user to authenticate via OAuth and with the minimum scope required.

2.2.11. LLM06: Excessive Agency - 7

Implement authorization in downstream systems rather than relying on an LLM to decide if an action is allowed or not. Enforce the complete mediation principle so that all requests made to downstream systems via extensions are validated against security policies.

2.2.12. LLM09: Misinformation - 6

Establish secure coding practices to prevent the integration of vulnerabilities due to incorrect code suggestions.

2.2.13. LLM10: Unbounded Consumption - 6

Restrict the LLM's access to network resources, internal services, and APIs. This is particularly significant for all common scenarios as it encompasses insider risks and threats. Furthermore, it governs the extent of access the LLM application has to data and resources, thereby serving as a crucial control mechanism to mitigate or prevent side-channel attacks.

2.2.14. LLM10: Unbounded Consumption - 9

Design the system to degrade gracefully under heavy load, maintaining partial functionality rather than complete failure.

2.2.15. LLM10: Unbounded Consumption - 11

Train models to detect and mitigate adversarial queries and extraction attempts

2.2.16. LLM10: Unbounded Consumption - 12

Build lists of known glitch tokens and scan output before adding it to the model's context window

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. MCP01:2025 - Token Mismanagement and Secret Exposure - 2

Limit Token Lifetime and Scope

- Issue short-lived, scoped tokens aligned with least privilege principles.
- Require token renewal for every new MCP session.
- Bind tokens to the specific agent, tool, or session context.

2.3.2. MCP01:2025 - Token Mismanagement and Secret Exposure - 3

Enforce Context Isolation

- Prevent sensitive data persistence in model memory or context windows.
- Redact or sanitize inputs and outputs before logging.
- Use ephemeral contexts for operations involving credentials.

2.3.3. MCP02:2025 - Privilege Escalation via Scope Creep - 1

Least Privilege by Design Define minimal permissions required per agent before deployment. Document intended actions and map them to explicit scopes. Use fine-grained scopes (e.g., `repo:write:branch=feature/*` rather than `repo:write`).

2.3.4. MCP02:2025 - Privilege Escalation via Scope Creep - 3

Expiry-Based & Just-in-Time (JIT) Access Issue time-limited scopes/tokens for sessions. Require revalidation for long-running or recurring tasks. Use JIT elevation workflows with approval gates for any higher-risk action.

2.3.5. MCP03:2025 - Tool Poisoning - 3

Strong Access Controls & Separation of Duties

- Apply least-privilege RBAC to the schema registry; separate the role that can propose a change from the role that approves and publishes it.
- Use short-lived tokens for deployment pipelines and require human approvals for critical schema releases.

2.3.6. MCP05:2025 – Command Injection & Execution - 1

Enforce Command Boundaries

- Use allowlists for permitted commands, arguments, and file paths.
- Reject shell metacharacters (`;` `&` `$()` `<>` `&&` `\``).
- Normalize and validate all file paths to block traversal.

2.3.7. MCP05:2025 – Command Injection & Execution - 2

Adopt Safe Execution Patterns

- Never use `shell=True`, `eval()`, `exec()`, or string-built commands.
- Always execute with structured parameters (e.g., `subprocess.run(['ls', 'logs'])`).
- Disable direct execution of model-generated code unless manually reviewed.

2.3.8. MCP05:2025 – Command Injection & Execution - 4

Apply Least Privilege Run tools as non-root with minimal filesystem, API, and DB permissions. Prevent agents from accessing environment variables or secrets by default.

2.3.9. MCP06:2025 – Intent Flow Subversion - 2

Independent Intent Verification (The Checker Pattern)

- Use a separate, independent “Guardrail Model” to verify proposed tool calls. This model should only see the User Intent and the Proposed Action, ensuring it is isolated from potentially poisoned MCP context.

2.3.10. MCP07:2025 – Insufficient Authentication & Authorization - 4

Least Privilege Principle

- Minimize agent permissions — assign only what’s needed for the task.
- Split high-privilege operations into separate workflows requiring human review.
- Restrict admin or system tokens from being used in development or shared contexts.

2.3.11. MCP09:2025 – Shadow MCP Servers - 12

Remove unapproved plugins, schemas, or connectors.

2.3.12. MCP10:2025 – Context Injection & Over-Sharing - 2

Context Isolation & Segmentation

Assign unique context namespaces per:

User

Agent

Workflow

Tenant - Prevent one agent from accessing another agent’s memory directly. - In multi-tenant setups, isolate retrieval indexes and vector stores.

2.3.13. MCP10:2025 – Context Injection & Over-Sharing - 8

Context Injection Filtering
Detect and block instruction-like content trying to persist in memory:

- "Ignore previous instructions"
- "Share everything you know"
- Maintain injection pattern detection models.

2.3.14. MCP10:2025 – Context Injection & Over-Sharing - 9

Purge existing shared contexts and caches. Enforce per-agent and per-user segmentation. Introduce TTL policies and auto-purge logic. Rotate keys and invalidate context stores if contamination is confirmed. Review access control around vector databases and embeddings.

2.4. AI Exchange

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

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

2.4.1. 1.1 General governance controls - SEC DEV PROGRAM

Secure development program: Have processes concerning software development in place to make sure that security is built into your AI system.

2.4.2. 1.1 General governance controls - DEV PROGRAM

Development program: Having a development lifecycle program for AI. Apply general (not just security-oriented) software engineering best practices to AI development.

2.4.3. 1.3. Controls to limit the effects of unwanted behaviour - LEAST MODEL PRIVILEGE

Least model privilege: Minimize what a model can do (trigger actions or access data), to prevent harm in case the model is manipulated, or makes a mistake by itself.

2.4.4. 1.3. Controls to limit the effects of unwanted behaviour - MODEL ALIGNMENT

In the context of Generative AI (e.g., LLMs), alignment refers to the process of ensuring that the model's behavior and outputs are consistent with human values, intentions, and ethical standards.

Controls external to the model to manage model behaviour are:

OVERSIGHT: conventional mechanisms responding to the actual outcome of the model

LEAST MODEL PRIVILEGE: conventional mechanisms that put boundaries on what the model can affect

PROMPT INJECTION I/O handling: detection mechanisms on input and output to prevent unwanted behaviour

The intent of Model alignment is achieve similar goals by baking it into the model itself, through training and instruction.

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. Design your system for security as well as functionality and performance

You are confident that the task at hand is most appropriately addressed using AI. Having determined this, you assess the appropriateness of your AI-specific design choices. You consider your threat model and associated security mitigations alongside functionality, user experience, deployment environment, performance, assurance, oversight, ethical and legal requirements, among other considerations.

3.1.2. Consider security benefits and trade-offs when selecting your AI model

Your choice of AI model will involve balancing a range of requirements. This includes choice of model architecture, configuration, training data, training algorithm and hyperparameters. Your decisions are informed by your threat model, and are regularly reassessed as AI security research advances and understanding of the threat evolves.

3.1.3. Collect and share lessons learned

You participate in information-sharing communities, collaborating across the global ecosystem of industry, academia and governments to share best practice as appropriate. You maintain open lines of communication for feedback regarding system security, both internally and externally to your organisation, including providing consent to security researchers to research and report vulnerabilities. When needed, you escalate issues to the wider community, for example publishing bulletins responding to vulnerability disclosures, including detailed and complete common vulnerability enumeration. You take action to mitigate and remediate issues quickly and appropriately.

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. Repeatability - c)

Ensuring exception handling is in line with organisations' policies;

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. 3.1.1 Data Poisoning: Threats and Mitigations in AI Supply Chains - Unauthorized Data

Training on copyrighted materials or using user data without consent:
Complete data provenance with source verification

5.1.2. 3.2.3 Application - Insecure Function Calling

Security weaknesses in the interface between language models and external function capabilities that can be exploited through control flow manipulation, allowing attackers to execute unauthorized actions when models are integrated with plugins, tools, or service APIs:
Restrict model-accessible functions to read-only operations where possible. Apply granular least-privilege principles to all model function access permissions. Implement role-based access control (RBAC) frameworks to limit model interaction capabilities based on authentication context.

5.2. Model Context Protocol (MCP) Security

Link: <https://github.com/cosai-oasis/ws4-secure-design-agentic-systems/blob/main/model-context-protocol-security.md>

Link: <https://github.com/cosai-oasis/ws4-secure-design-agentic-systems/blob/main/model-context-protocol-security.md>

5.2.1. 3.2.2 Secure Delegation and Access Control

To mitigate against privilege escalation, MCP servers should operate with the minimum privileges necessary. OAuth provides a widely adopted framework for secure delegation, with extensions that support fine-grained scope control and secure token flows (see MCP Authorization).

- Leverage existing identity providers to provide user authentication using standards such as OIDC
- Register MCP server as clients with the IAM provider. If the registration cannot happen a priori, then use Dynamic Client Registration
- Do not passthrough the OAuth tokens provided by the user
- Perform token exchange with the authorization server to provide full accountability (RFC8693)
- Reduce scopes for least privilege, such as removing write scopes when only read access is required (SEP-835 adds native support to define scopes in 2025-11-25 MCP specification)
- User short-lived tokens and support proof-of-possession (DPoP) to prevent replay attacks (RFC9449)
- Fine grained authorizations, through Rich Authorization Requests (RFC9396), limit requests to specific resources or tool parameters

All endpoint services should implement robust access control models, such as role-based access control (RBAC) or attribute-based access control and evaluate against claims made by the identity provider, such as role membership, job title, or work location. Additionally, robust policy languages including Open Policy Agent (OPA), Cedar, or OpenFGA provide robust, flexible, and secure protections.

6. IBM

6.1. IBM Framework for Securing Generative AI

Link: <https://www.ibm.com/products/tutorials/ibm-framework-for-securing-generative-ai>

Link: <https://www.ibm.com/products/tutorials/ibm-framework-for-securing-generative-ai>

6.1.1. Secure the usage

During inferencing and live use, attackers can manipulate prompts to jailbreak guardrails and coax models into misbehaving by generating disallowed responses to harmful prompts that include biased, false and other toxic information. This can inflict reputational damage on the enterprise. Attackers might also seek to manipulate the model and analyze input/output pairs to train a surrogate model to mimic the behavior of the target model, effectively “stealing” its capabilities and costing an enterprise its competitive advantage.

Several types of attacks are concerning in this stage of the AI pipeline. First, prompt injections—where attackers use malicious prompts to jailbreak models and get unwarranted access, steal sensitive data or introduce bias into outputs. Another concern involves model denial of service, where attackers overwhelm the LLM with inputs that degrade the quality of service and incur high resource costs. Organizations should also prepare for and defend against model theft, where attackers craft inputs to collect model outputs to train a surrogate model that mimics the behavior of the target model.

Our best practices include monitoring for malicious inputs such as prompt injections and outputs containing sensitive data or inappropriate content, and implementing new defenses that can detect and respond to AI-specific attacks such as data poisoning, model evasion and model extraction. New AI-specific solutions have entered the market under the name of machine learning detection and response (MLDR). Alerts generated from these solutions can be integrated into security operations solutions, such as IBM Security® QRadar®, enabling security operations center (SOC) teams to quickly initiate response playbooks that deny access, quarantine or disconnect compromised models.

7. OpenAI

7.1. Preparedness Framework

Link: <https://cdn.openai.com/pdf/18a02b5d-6b67-4cec-ab64-68cdfbddebcd/preparedness-framework-v2.pdf>

Link: <https://cdn.openai.com/pdf/18a02b5d-6b67-4cec-ab64-68cdfbddebcd/preparedness-framework-v2.pdf>

7.1.1. Undermining Safeguards

Undermining Safeguards: ability and propensity for the model to act to undermine safeguards placed on it, including e.g., deception, colluding with oversight models, sabotaging safeguards over time such as by embedding vulnerabilities in safeguards code, etc.

If a model has High or Critical capabilities in any of the Tracked Categories, require the Safeguards case to be robust to the discovered capability and/or propensity.

7.1.2. Safeguards Against Malicious Users - Robustness

Malicious users cannot use the model to cause the severe harm because they cannot elicit the necessary capability, such as because the model is modified to refuse to provide assistance to harmful tasks and is robust to jailbreaks that would circumvent those refusals.

7.1.3. Safeguards Against Malicious Users - Trust-based Access

The actors who gain access to the model are not going to use it in a way that presents an associated risk of severe harm under our threat model.

7.1.4. Safeguards Against a Misaligned Model - Lack of Autonomous Capability

The model is not capable of carrying out tasks autonomously, including the risk of severe harm.

7.1.5. Safeguards Against a Misaligned Model - System Architecture

The model can't take actions that cause harm because it lacks access to output channels or mechanisms to persist sufficiently to execute the harm, due to system design and restricted permissions.

7.2. Safety Best Practices

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

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

7.2.1. Prompt engineering

“Prompt engineering” can help constrain the topic and tone of output text. This reduces the chance of producing undesired content, even if a user tries to produce it. Providing additional context to the model (such as by giving a few high-quality examples of desired behavior prior to the new input) can make it easier to steer model outputs in desired directions.

7.2.2. Constrain user input and limit output tokens

Limiting the amount of text a user can input into the prompt helps avoid prompt injection. Limiting the number of output tokens helps reduce the chance of misuse.

Narrowing the ranges of inputs or outputs, especially drawn from trusted sources, reduces the extent of misuse possible within an application.

Allowing user inputs through validated dropdown fields (e.g., a list of movies on Wikipedia) can be more secure than allowing open-ended text inputs.

Returning outputs from a validated set of materials on the backend, where possible, can be safer than returning novel generated content (for instance, routing a customer query to the best-matching existing customer support article, rather than attempting to answer the query from-scratch).

7.2.3. Understand and communicate limitations

From hallucinating inaccurate information, to offensive outputs, to bias, and much more, language models may not be suitable for every use case without significant modifications. Consider whether the model is fit for your purpose, and evaluate the performance of the API on a wide range of potential inputs in order to identify cases where the API's performance might drop. Consider your customer base and the range of inputs that they will be using, and ensure their expectations are calibrated appropriately.

8. NCSC/NSA/CISA etc

8.1. 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

8.1.1. 3.4 Ensemble Methods/ collaborative learning

Implement collaborative learning frameworks that combine an ensemble of multiple, distinct AI models to reach a consensus on output predictions. This approach can help counteract the impact of data poisoning, since malicious inputs may only affect a subset of the collaborative models, allowing the majority to maintain accuracy and reliability.

9. OECD

9.1. Due Diligence Guidance for Responsible AI

Link: https://www.oecd.org/content/dam/oecd/en/publications/reports/2026/02/oecd-due-diligence-guidance-for-responsible-ai_7831bb49/41671712-en.pdf

Link: https://www.oecd.org/content/dam/oecd/en/publications/reports/2026/02/oecd-due-diligence-guidance-for-responsible-ai_7831bb49/41671712-en.pdf

9.1.1. Step 1.1 – RBC policies

Devise, adopt and disseminate a combination of policies on RBC issues that articulates the enterprise's commitments to the principles and standards contained in the OECD AI Principles and the MNE Guidelines. Policies should include plans for implementing due diligence, which will be relevant to the enterprise's own operations and business relationships in the development and use of AI.

9.1.2. Step 3.2 – Addressing risks directly linked to the enterprise throughout the AI value chain

Based on the risk prioritisation, develop and implement plans to prevent or mitigate actual or potential adverse impacts directly linked to the enterprise by business relationships (e.g., temporary suspension of the relationship, continuation of the relationship throughout the course of risk mitigation effort, or disengagement).

Enterprises throughout the development and use of AI might be directly linked to adverse impacts caused by (1) other AI actors in the system lifecycle; or (2) business relationships outside of the AI system lifecycle, such as suppliers of AI inputs and users of the AI system.

Appropriate responses to risks associated with business relationships may at times include:

- continuation of the relationship throughout the course of risk mitigation efforts
- temporary suspension of the relationship while pursuing ongoing risk mitigation
- disengagement with the business relationship either after failed attempts at mitigation, or where the enterprise deems mitigation not feasible, or because of the severity of the adverse impact. A decision and subsequent plan to disengage should take into account potential social, environmental and economic adverse impacts and should include meaningful stakeholder engagement. These plans should detail the actions the organisation will take, as well as its expectations of its suppliers, buyers and other business relationships (see Box 2.15). Disengagement actions should be in line with applicable laws, including competition laws.

10. IMDA

10.1. 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>

10.1.1. 2.1.2 Bound risks through design by defining agents limits and permissions - Agent limits

Having selected an appropriate agent use case, organisations can further bound the risks by defining appropriate limits and permission policies for each agent.

Organisations should consider defining limits on:

- Agent's access to tools and systems: Define policies that give agents only the minimum tools and data access needed for it to complete its task. For example, a coding assistant may not require access to a web search tool, especially if it already has curated access to the latest software documentation.
- Agent's autonomy: For process-driven tasks, SOPs and protocols are frequently used to improve consistency and reduce unpredictability. Define similar SOPs for agentic workflows that an agent is constrained to follow, rather than giving the agent the freedom to define every step of the workflow.
- Agent's area of impact: Design mechanisms and procedures to take agents offline and limit their potential scope of impact when they malfunction. This can include running agents in self-contained environments with limited network and data access, particularly when they are carrying out high-risk tasks such as code execution.

10.1.2. 2.3.1 During design and development, use technical controls

Organisations should design and implement technical controls in the agentic AI system to mitigate identified risks. For agents specifically, in addition to baseline software and LLM controls, consider adding controls for:

- New agentic components, such as planning and reasoning and tools
- Increased security concerns from the larger attack surface and new protocols

For illustration, these are some sample controls for agents. For a more comprehensive list, organisations can refer to CSA's Draft Addendum on Securing Agentic AI and GovTech's Agentic Risk and Capability Framework.

11. Cyber Security Council (UAE)

11.1. 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

11.1.1. 2.1.1

The entity shall ensure secure design, development, and operation of AI/ML systems by establishing and maintaining comprehensive, enforceable policies and procedures.

11.1.2. 2.1.4

The entity shall control and manage changes to AI/ML systems in a way that maintains system security.

11.1.3. 2.2.2

The entity shall ensure AI/ML systems are securely and consistently configured in accordance with approved security baselines and configuration standards.

11.1.4. 2.2.5

The entity shall ensure that AI/ML applications are securely developed, tested, and maintained throughout their lifecycle in accordance with established security practices.

11.1.5. 2.3.1

The entity shall integrate cyber security into the design and development of AI/ML models from the earliest stages, to proactively mitigate security risks and avoid retrofitting controls after deployment.

11.1.6. 2.3.2

The entity shall establish and maintain security measures to protect AI/ML models from potential threats, ensuring their integrity and availability throughout the model lifecycle.

11.1.7. 2.4.1

The entity shall tailor cyber security measures for AI/ML systems according to their specific application domains, implementing context-appropriate safeguards.

11.1.8. 3.1.1 Cyber Security Policies & Procedures - 1

The entity should develop, implement, and maintain a set of comprehensive policies and procedures that specifically address the unique cyber security considerations associated with AI/ML systems.

11.1.9. 3.2.5 Application Security - 1

The entity should implement a secure software development lifecycle (SDLC) for AI/ML applications that includes security requirements definition, secure coding practices, security testing, and post-deployment security reviews.

11.1.10. 3.3.1 Security by Design for AI/ML Models - 1

The entity should ensure that cyber security processes are integrated into AI/ML models from the earliest stages of their design and development.

11.1.11. 3.3.1 Security by Design for AI/ML Models - 3

The entity should ensure that AI/ML models comply with established cyber security standards and best practices, including data minimization, least privilege, and separation of duties principles.

11.1.12. 3.3.2 AI/ML Model Security - 3

The entity should implement safeguards to protect against adversarial attacks, such as adversarial training, robust input validation, and continuous monitoring for anomalous model behavior indicative of potential cyber security incidents.

11.1.13. 3.3.3 Inference Security - 1

The entity should implement measures to secure the process of making predictions with AI/ML models, ensuring the integrity, confidentiality, and availability of inference data.

11.1.14. 3.3.3 Inference Security - 2

The entity should adopt strategies to mitigate threats against inference security, such as model inversion attacks, membership inference attacks, and adversarial attacks.

11.1.15. 3.3.3 Inference Security - 3

The entity should deploy defensive measures, including but not limited to robust input validation, to protect against inference attacks aiming to exploit or reverseengineer AI/ML models.

11.1.16. 3.4.1 Context-Specific AI/ML Security - 1

The entity should identify and address the unique cyber security requirements of AI/ML systems based on their specific application context (e.g. medical, automotive, industrial, customer service, etc.).

11.1.17. 3.4.1 Context-Specific AI/ML Security - 2

For safety-critical applications (e.g. self-driving cars, medical diagnostics), the entity should prioritize robustness, reliability, and fail-safe mechanisms in AI/ML design and deployment.

11.1.18. 3.4.1 Context-Specific AI/ML Security - 3

For privacy-sensitive applications (e.g. personal assistants, customer service bots), the entity should prioritize data protection, privacy, and transparency in AI/ML design and deployment.

11.1.19. 3.4.1 Context-Specific AI/ML Security - 4

The entity should follow industry-specific guidelines, best practices, and regulations for AI/ML cyber security, as applicable to their application context.

11.1.20. 3.4.2 Fail-Safe & Backup for AI/ML Systems - 1

The entity should implement robust fail-safe mechanisms for AI/ML systems to prevent failures and ensure system stability under unexpected conditions.

11.1.21. 3.5.2 Defending Against AI/ML Attacks - 2

Security measures should be integrated during the development and deployment of AI/ML models to minimize the potential impact of attacks.

12. Central Bank of the UAE

12.1. 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>

12.1.1. 5. Data Quality, Privacy and Security - c

Institutions should incorporate privacy-by-design and security-by-design principles into AI systems and maintain safeguards to protect data from unauthorised access or misuse.

12.1.2. 7. Human Oversight and Consumer Protection - d

The design and deployment of AI and ML systems should promote fair and equitable treatment. AI should not be used to target consumers with unsuitable products or to engage in pressure-selling or misleading marketing. Institutions should ensure that promotional materials and chatbots comply with disclosure requirements, all in line with the Consumer Protection Regulation's emphasis on good disclosure.

13. MIC/METI (Japan)

13.1. 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

13.1.1. Human-Centric - 3

Generative AI has enabled everyone to forge fake information that seems to be true and fair, so recognize the increasing risk of destabilizing and confusing the society through disinformation, misinformation, and biased information generated by AI, and take necessary countermeasures.

13.1.2. Safety - 2

Develop, provide, or use AI systems and services within the range in which the AI business actor can control, preventing damage due to a provision or use that deviates from the intended purpose.

13.1.3. Fairness - 1 (a)

There are a broad range of factors that can produce an inappropriate bias, so identify the factors that might produce biases that can be considered as problems from the viewpoint of fairness. Those factors may include technological elements (training data, AI model training process, prompts entered by AI business users or non-business users , and reference information and collaborating external services used by AI models for inference) and behaviors of AI business users.

13.1.4. Ensuring security - 2 (a)

To maintain the confidentiality, integrity, and availability of AI systems and services and ensure safe and secure AI use constantly, take reasonable measures based on the technological level at the time.

13.1.5. Ensuring security - 1 (b)

Understand the characteristics of AI systems and services, and examine whether the inter-system connections necessary for normal operations are properly established.

14. EU

14.1. 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

14.1.1. 10.1 Data and Data Governance

High-risk AI systems which make use of techniques involving the training of AI models with data shall be developed on the basis of training, validation and testing data sets that meet the quality criteria referred to in paragraphs 2 to 5 whenever such data sets are used.

14.1.2. 10.2 Data and Data Governance

Training, validation and testing data sets shall be subject to data governance and management practices appropriate for the intended purpose of the high-risk AI system. Those practices shall concern in particular:

- (a) the relevant design choices;
- (b) data collection processes and the origin of data, and in the case of personal data, the original purpose of the data collection;
- (c) relevant data-preparation processing operations, such as annotation, labelling, cleaning, updating, enrichment and aggregation;
- (d) the formulation of assumptions, in particular with respect to the information that the data are supposed to measure and represent;
- (e) an assessment of the availability, quantity and suitability of the data sets that are needed;
- (f) examination in view of possible biases that are likely to affect the health and safety of persons, have a negative impact on fundamental rights or lead to discrimination prohibited under Union law, especially where data outputs influence inputs for future operations;
- (g) appropriate measures to detect, prevent and mitigate possible biases identified according to point (f);
- (h) the identification of relevant data gaps or shortcomings that prevent compliance with this Regulation, and how those gaps and shortcomings can be addressed.

14.1.3. 14.3 Human Oversight

The oversight measures shall be commensurate with the risks, level of autonomy and context of use of the high-risk AI system, and shall be ensured through either one or both of the following types of measures:

- (a) measures identified and built, when technically feasible, into the high-risk AI system by the provider before it is placed on the market or put into service;
- (b) measures identified by the provider before placing the high-risk AI system on the market or putting it into service and that are appropriate to be implemented by the deployer.

14.1.4. 15.1 Accuracy, Robustness and Cybersecurity

High-risk AI systems shall be designed and developed in such a way that they achieve an appropriate level of accuracy, robustness, and cybersecurity, and that they perform consistently in those respects throughout their lifecycle.

14.1.5. 17.1 Quality Management Systems

Providers of high-risk AI systems shall put a quality management system in place that ensures compliance with this Regulation. That system shall be documented in a systematic and orderly manner in the form of written policies, procedures and instructions, and shall include at least the following aspects:

- (a) a strategy for regulatory compliance, including compliance with conformity assessment procedures and procedures for the management of modifications to the high-risk AI system;
- (b) techniques, procedures and systematic actions to be used for the design, design control and design verification of the high-risk AI system;
- (c) techniques, procedures and systematic actions to be used for the development, quality control and quality assurance of the high-risk AI system;
- (d) examination, test and validation procedures to be carried out before, during and after the development of the high-risk AI system, and the frequency with which they have to be carried out;
- (e) technical specifications, including standards, to be applied and, where the relevant harmonised standards are not applied in full or do not cover all of the relevant requirements set out in Section 2, the means to be used to ensure that the high-risk AI system complies with those requirements.
- (f) systems and procedures for data management, including data acquisition, data collection, data analysis, data labelling, data storage, data filtration, data mining, data aggregation, data retention and any other operation regarding the data that is performed before and for the purpose of the placing on the market or the putting into service of high-risk AI systems;
- (g) the risk management system referred to in Article 9;
- (h) the setting-up, implementation and maintenance of a post-market monitoring system, in accordance with Article 72;
- (i) procedures related to the reporting of a serious incident in accordance with Article 73;
- (j) the handling of communication with national competent authorities, other relevant authorities, including those providing or supporting the access to data, notified bodies, other operators, customers or other interested parties;
- (k) systems and procedures for record-keeping of all relevant documentation and information;
- (l) resource management, including security-of-supply related measures;
- (m) an accountability framework setting out the responsibilities of the management and other staff with regard to all the aspects listed in this paragraph.

14.1.6. 17.2 Quality Management Systems

The implementation of the aspects referred to in paragraph 1 shall be proportionate to the size of the provider's organisation. Providers shall, in any event, respect the degree of rigour and the level of protection required to ensure the compliance of their high-risk AI systems with this Regulation.

14.1.7. 17.4 Quality Management Systems

For providers that are financial institutions subject to requirements regarding their internal governance, arrangements or processes under Union financial services law, the obligation to put in place a quality management system, with the exception of paragraph 1, points (g) (h) and (i) of this Article, shall be deemed to be fulfilled by complying with the rules on internal governance arrangements or processes pursuant to the relevant Union financial services law. To that end, any harmonised standards referred to in Article 40 shall be taken into account.

14.1.8. 50.1 Transparency obligations for providers and deployers of certain AI systems

Providers shall ensure that AI systems intended to interact directly with natural persons are designed and developed in such a way that the natural persons concerned are informed that they are interacting with an AI system, unless this is obvious from the point of view of a natural person who is reasonably well-informed, observant and circumspect, taking into account the circumstances and the context of use. This obligation shall not apply to AI systems authorised by law to detect, prevent, investigate or prosecute criminal offences, subject to appropriate safeguards for the rights and freedoms of third parties, unless those systems are available for the public to report a criminal offence.

15. CEN/CENELEC

15.1. prEN 40000-1-1

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

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

15.1.1. product control

a measure on a product that modifies risk.

15.2. 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>

15.2.1. 5.3

Security by Design

15.2.2. 5.4

Secure by Default

15.2.3. 6.2

Product Context

15.2.4. 7.2

Product cybersecurity planning

15.2.5. 7.3

Product cybersecurity requirements

15.2.6. 7.11

Third-party component cybersecurity management

16. World Economic Forum

16.1. 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

16.1.1. Expanded AI life cycle

The expanded AI life cycle encompasses risks and guardrails with varying safety benefits and challenges throughout each phase. The expanded AI life cycle synthesizes elements from data management, foundation model design and development, release access, use of generative capabilities and adaptation to a use case.

16.1.2. Guardrails across the expanded AI life cycle

Implementation of known and novel guardrails is necessary for safe systems to ensure technical quality, consistency and control.

17. U.S. Department of Health & Human Services

17.1. Trustworthy AI (TAI) Playbook: Executive Summary

Link: <https://www.hhs.gov/sites/default/files/hhs-trustworthy-ai-playbook-executive-summary.pdf>

Link: <https://www.hhs.gov/sites/default/files/hhs-trustworthy-ai-playbook-executive-summary.pdf>

17.1.1. Robust / Reliable

AI systems should have the ability to learn from humans and other systems and produce accurate and reliable outputs consistent with the original design

18. ICO

18.1. 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>

18.1.1. What should we assess in our DPIA?

Your DPIA needs to describe the nature, scope, context and purposes of any processing of personal data - it needs to make clear how and why you are going to use AI to process the data. You need to detail:

- how you will collect, store and use data;
- the volume, variety and sensitivity of the data;
- the nature of your relationship with individuals; and
- the intended outcomes for individuals or wider society, as well as for you.

In the context of the AI lifecycle, a DPIA will best serve its purpose if you undertake it at the earliest stages of project development. It should feature, at a minimum, the following key components.

18.1.2. Why is controllership important for AI systems?

In many cases, the various processing operations involved in AI may be undertaken by a number of different organisations. It is therefore crucial that you determine who is a controller, joint controller or processor if your use of AI involves multiple organisations.

A first step to understanding these relationships is to identify the distinct sets of processing operations and their purposes (see 'What should we assess in our DPIA?'). For each of these, you and the organisations you work with then need to assess whether you are a controller, processor or joint controller. You should consult our guidance on controller/processor for assistance in this assessment, but in essence:

- if you decide on the purposes and means of processing, you are a controller;
- if you process personal data under the instruction of another organisation, you are a processor; and
- if you jointly determine the purposes and means of processing with another organisation, you are joint controllers.

18.1.3. What type of decisions may make us a controller?

The type of decisions that are likely to make you a controller include deciding:

- to collect personal data in the first place
- the purpose(s) for that processing
- which individuals to collect the data about, and what to tell them about the processing;
- how long you will retain the data for; and
- how to respond to requests made in line with individuals' rights.

For more specifics on the circumstances that may make you a controller, read our guidance on controllers and processors. In the context of AI, the type of decisions made by controllers can include:

- the source and nature of the data used to train an AI model;
- the target output of the model (ie what is being predicted or classified);
- the broad kinds of ML algorithms that will be used to create models from the data (eg regression models, decision trees, random forests, neural networks);
- feature selection – the features that may be used in each model;
- key model parameters (eg how complex a decision tree can be, or how many models will be included in an ensemble);
- key evaluation metrics and loss functions, such as the trade-off between false positives and false negatives; and
- how any models will be continuously tested and updated: how often, using what kinds of data, and how ongoing performance will be assessed.

18.1.4. What are our security requirements?

There is no "one-size-fits-all" approach to security. The appropriate security measures you should adopt depend on the level and type of risks that arise from specific processing activities.

Using AI to process any personal data has important implications for your security risk profile, and you need to assess and manage these carefully.

Some implications may be triggered by the introduction of new types of risks, eg adversarial attacks on machine learning models.

18.1.5. Preventative Controls - 2

Comply to or assess an AI system against external security certifications or schemes.

18.1.6. Preventative Controls - 3

Subject software to a quality review where one or more individuals view and read parts of its source code. At least one of the reviewers must not be the author of the code.

18.1.7. Preventative Controls - 11

Assess more secure implementations of the trained model, and implement them as appropriate, post development but pre-deployment.

19. ISO

19.1. 42001:2023 - Information technology — Artificial intelligence — Management system

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

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

19.1.1. 4.1

Understanding the organization and its context

19.1.2. 4.2

Understanding the needs and expectations of interested parties

19.1.3. 4.3

Determining the scope of the AI management system

19.1.4. 6.2

AI objectives and planning to achieve them

20. Federal Office for Information Security

20.1. 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

20.1.1. 3.3 Defending against Evasion Attacks - Adversarial Retraining

Adversarial retraining consists of iteratively generating adversarial examples and repeatedly training the model on them. As a result, the robustness of the model against the selected attack methods increases.

20.1.2. 5.3 Defending against Poisoning and Backdoor Attacks - Retraining

For network pruning, benign data samples are fed into the trained neural network, and their average activation is measured. Neurons without a high level of activation can be trimmed without substantially reducing the model's accuracy. In the process, potential backdoors can be removed as well. Similar to retraining, the complete success of the measure cannot be guaranteed [20].

20.1.3. 5.3 Defending against Poisoning and Backdoor Attacks - Autoencoder Detection

Regularization can lower the success rate of backdoor attacks without significantly degrading the baseline performance on benign inputs [18].

21. ENISA

21.1. 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>

21.1.1. Poisoning

Similarly, for poisoning attacks, processes that maintain the security levels of ML components over time should be implemented, the exposure level of the used model should be assessed, the training data set should be enlarged as much as possible to reduce its susceptibility to malicious samples, and pre-processing steps that clean the training data from such malicious samples must also be considered.

21.1.2. Networking 6

Have you defined/developed/used specific cyber measurements/metrics at the national level that AI stakeholders are required to use?

22. Databricks

22.1. 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>

22.1.1. DASF 25: Use retrieval augmented generation (RAG) with large language models (LLMs)

Generating relevant and accurate responses in large language models (LLMs) while avoiding hallucinations requires grounding them in domain-specific knowledge. Retrieval augmented generation (RAG) addresses this by breaking down extensive datasets into manageable segments ("chunks") that are "vector embedded." These vector embeddings are mathematical representations that help the model understand and quantify different data segments. As a result, LLMs produce responses that are contextually relevant and deeply rooted in the specific domain knowledge.

22.1.2. DASF 41: Platform security — secure SDLC

Databricks engineering integrates security throughout the software development lifecycle (SDLC), encompassing both technical and process-level controls under the oversight of our chief security officer (CSO). Activities within our SDLC include:

- Code peer reviews
- Static and dynamic scans for code and containers, including dependencies
- Feature-level security reviews
- Annual software engineering security training
- Cross-organizational collaborations between security, product management, product security and security champions
- Controls to prevent over-disclosure of technical information about AI systems and organizational details that could enable adversarial targeting

These development controls are augmented by internal and external penetration testing programs, with findings tracked for resolution and reported to our executive team. Databricks' processes undergo an independent annual review, the results of which are published in our SOC 2 Type 2 report, available upon request.

22.1.3. DASF 54: Implement AI guardrails

AI Guardrails allow users to configure and enforce data compliance at the model serving endpoint level and to reduce harmful content on any requests sent to the underlying model. Bad requests and responses are blocked, and a default message is returned to the user. You can configure and enforce data compliance at the model serving endpoint level and reduce harmful content on any requests sent to the underlying model with AI Guardrails. With AI Guardrails, you can configure the following controls on your AI system:

- Safety filtering prevents your model from interacting with unsafe and harmful content, like violent crime, self-harm, and hate speech.
- Personally identifiable information (PII) detection to detect any sensitive information (such as names, addresses, and credit card numbers) for users
- Topic moderation to list a set of allowed topics. Given a chat request, this guardrail flags the request if its topic is not one of the permitted topics.
- Keyword filtering will specify different sets of invalid keywords for both the input and the output. One potential use case for keyword filtering is so the model does not talk about competitors.

22.1.4. DASF 64: Limit access from AI models and agents

Allow AI models and agents access to enterprise resources based on the principle of least privilege. In Unity Catalog, a "securable object" is any object that can be permissioned to a principal (e.g., user, service principal, or group) and is organized hierarchically. Treat AI as a principal and assign permissions accordingly.

With on-behalf-of-user authentication, agents deployed via Mosaic AI model serving can access Databricks resources using the identity of the Databricks end user who queried the agent. This enables accessing sensitive information on a per-user basis, with fine-grained enforcement of data access control in Unity Catalog.

22.1.5. DASF 68: Use Securely Hosted Managed MCP Servers

Model Context Protocol (MCP) servers act as bridges that let AI agents access external data and tools. Instead of building these connections from scratch, you can use securely hosted Databricks managed MCP servers to instantly connect your agents to data stored in Unity Catalog, vector search indexes, and custom functions.

22.1.6. DASF 73: Register prompts

Although not quite as strong a control as a good safety filtering, prompt injection or jailbreak detection model, a good system prompt can sometimes be the difference between an attack being successful or not. MLflow prompt registry is a powerful tool that streamlines prompt engineering and management in your GenAI applications. It enables you to version, track, test and reuse prompts across your organization, helping maintain consistency and improving collaboration in effective prompt development.

23. ISO/IEC

23.1. DIS 27090

Link: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:27090:dis:ed-1:v1:en>

Link: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:27090:dis:ed-1:v1:en>

23.1.1. 7.2

Conflicting interactions of mitigations

23.2. TR 27563:2023

Link: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:tr:27563:ed-1:v1:en>

Link: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:tr:27563:ed-1:v1:en>

23.2.1. 7.2

Provide assessment of systems of interest

23.3. TR 27091

Link: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:27091:dis:ed-1:v1:en>

Link: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:27091:dis:ed-1:v1:en>

23.3.1. 6.2

Privacy threats to AI models

23.3.2. 7.2.1

Privacy engineering

24. METI (Japan)

24.1. 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

24.1.1. Action Target 3-4

Companies that develop and operate AI systems and those that provide data should, under the leadership of top management, reduce incident-related burdens on users by preventing incidents and through early response.

24.1.2. Action Target 5-1

Companies that develop and operate AI systems should, under the leadership of top management, have individuals independent of the design and operation of the AI management system verify whether an AI management system such as a gap analysis process is appropriately designed and operated in light of the AI governance goals, in other words, whether an AI management system appropriately works for the achievement of the AI governance goals through the implementation of Action Targets 3 and 4.

25. Qatar Central Bank

25.1. 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

25.1.1. 6.3

An Entity should ensure that its strategy provides a business case, addresses information and communication technology requirements, information security, and operational risk management including business continuity, disaster recovery, and resiliency framework.

25.1.2. 6.4

An Entity's AI strategy should define an implementation plan and architectural roadmap which covers the target IT environment, the transition from the current environment to the target environment and the operating model, including any organizational change or additional skillsets that may be necessary.

25.1.3. 6.5

An Entity must allocate sufficient resources to handle any AI projects and ongoing business needs.

25.1.4. 9.2

An Entity must evaluate the use of the AI System in critical organizational processes.

25.1.5. 12.9

An Entity must also have a contingency plan in the event that the arrangement with the outsourcing service provider is suddenly terminated.

25.1.6. 13.7.5

An Entity must ensure an AI System is subject to built-in operational constraints that cannot be overridden by the system itself.

25.1.7. 16.2

The implementation of aspects referred to in clause (16.1) must be proportionate to the size of the Provider's organization and use of High-Risk AI.

25.1.8. 17.3

An Entity's TRISM program must ensure that AI Providers adhere to their program which includes:

- Utilizing defined AI Model management processes.
- Building AI Models with defined controls against security breaches.

25.1.9. 17.4

An Entity must examine any AI Model's attack surface prior to deployment, and address any security findings.

25.1.10. 17.5

An Entity must ensure that its AI model is protected from integrity related attacks to prevent model manipulation.

25.1.11. 17.6

An Entity must ensure that the AI model is protected against query attacks that may lead to model manipulation or theft.

25.1.12. 17.7

An Entity must ensure that the AI model is protected against prompt injections that may lead to data poisoning or data drift.

26. Smart Dubai (UAE)

26.1. 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>

26.1.1. 1.1.1.2

AI developer organisations and AI operator organisations should refrain from training AI systems on data that is not likely to be representative of the affected AI subjects, or is not likely to be accurate, whether that be due to age, omission, method of collection, or other factors.

26.1.2. 1.2.2.1

AI operator organisations should only use AI systems that are backed by respected and evidence-based academic research, and AI developer organisations should base their development on such research.

27. SDAIA (Saudi Arabia)

27.1. 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>

27.1.1. Principle 2 – Privacy & Security - Plan and Design - 1

The planning and design of the AI system and its associated algorithm must be configured and modelled in a manner such that there is respect for the protection of the privacy of individuals, personal data is not misused and exploited, and the decision criteria of the automated technology is not based on personally identifying characteristics or information.

27.1.2. Principle 2 – Privacy & Security - Plan and Design - 2

The use of personal information should be limited only to that which is necessary for the proper functioning of the system. The design of AI systems resulting in the profiling of individuals or communities may only occur if approved by Chief Compliance and Ethics Officer, Compliance Officer or in compliance with a code of ethics and conduct developed by a national regulatory authority for the specific sector or industry

27.1.3. Principle 2 – Privacy & Security - Plan and Design - 3

The security and protection blueprint of the AI system, including the data to be processed and the algorithm to be used, should be aligned to best practices to be able to withstand cyberattacks and data breach attempts.

27.1.4. Principle 2 – Privacy & Security - Plan and Design - 4

Privacy and security legal frameworks and standards should be followed and customized for the particular use case or organization.

27.1.5. Principle 2 – Privacy & Security - Plan and Design - 5

An important aspect of privacy and security is data architecture; consequently, data classification and profiling should be planned to define the levels of protection and usage of personal data.

27.1.6. Principle 2 – Privacy & Security - Build and Validate - 1

Privacy and security by design should be implemented while building the AI system. The security mechanisms should include the protection of various architectural dimensions of an AI model from malicious attacks. The structure and modules of the AI system should be protected from unauthorized modification or damage to any of its components.

27.1.7. Principle 2 – Privacy & Security - Build and Validate - 2

The AI system should be secure to ensure and maintain the integrity of the information it processes. This ensures that the system remains continuously functional and accessible to authorized users. It is crucial that the system safeguards confidential and private information, even under hostile or adversarial conditions. Furthermore, appropriate measures should be in place to ensure that AI systems with automated decision-making capabilities uphold the necessary data privacy and security standards.

27.1.8. Principle 5 – Reliability & Safety - Plan and Design - 1

Designing and developing an AI system that can withstand the uncertainty, instability, and volatility that it might encounter is crucial.

27.1.9. Principle 5 – Reliability & Safety - Plan and Design - 3

Establishing a set of standards and protocols for assessing the reliability of an AI system is necessary to secure the safety of the system's algorithm and data output. It is essential to keep a sustainable technical outlay and outcomes generated from the system to maintain the public's trust and confidence in the AI system.

27.1.10. Principle 6 – Transparency & Explainability - Build and Validate - 1

Transparency in AI is thought about from two perspectives, the first is the process behind it (the design and implementation practices that lead to an algorithmically supported outcome) and the second is in terms of its product (the content and justification of that outcome). Algorithms should be developed in a transparent way to ensure that input transparency is evident and explainable to the end-users of the AI system to be able to provide evidence and information on the data used to process the decisions that have been processed.

27.2. Generative AI Guidelines

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

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

27.2.1. 4.5 Privacy & Security - 4

Privacy and security by design should be implemented while building the AI system. The security mechanisms should include the protection of various architectural dimensions of an AI model from malicious attacks. The structure and modules of the AI system should be protected from unauthorized modification or damage to any of its components.

27.3. 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>

27.3.1. 5.1.2 Privacy and Safety - Embedding Cybersecurity and Privacy

Designing modern intelligent systems requires a "Security & Privacy by Design" approach, integrating protection requirements into every development stage, from analysis to deployment. This includes data encryption, access management, and multi-factor authentication to safeguard the system from vulnerabilities and attacks before official launch.

28. IETF

28.1. Security Requirements for AI Agents

Link: <https://www.ietf.org/archive/id/draft-ni-a2a-ai-agent-security-requirements-00.html>

Link: <https://www.ietf.org/archive/id/draft-ni-a2a-ai-agent-security-requirements-00.html>

28.1.1. 5.3. Converting to Internal Workflow

Workflow Generation: Complex tasks often require multi-agent collaboration. The master agent receives, parses, and extracts the original job request from the external requesting agent, then create sequential workflows or parallel calls. This requires the master agent to have information of all callable internal API assets, agent capabilities, etc.

Downscoping: If the master agent intends to use a workflow, it extracts the original caller's identity and authorization context, and initiates a new internal workflow. It should follow the current least privilege best practice of downscoping-Transaction Tokens as specified in [I-D.draft-tulshibagwale-oauth-transaction-tokens-05]. The access rights to each downstream workload decrease.

29. SANS

29.1. Critical AI Security Guidelines

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

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

29.1.1. 3.3 Implement Access Controls Outside of the Model

Attempting to implement these types of controls within the model is error prone and can often be easily subverted. Instead, consider a RAG-style approach with access control lists (ACLs) applied in the vector retrieval system from which responses are generated. This eliminates the need to attempt to implement these guardrails in the LLM. This approach also has the not-so-subtle benefit of limiting the likelihood of so-called hallucinations in the responses from the LLM. In addition, organizations should pay close attention to the use of function calling, especially in agentic AI systems. If not properly scoped, function calls may allow models to invoke external tools or actions beyond their intended purpose. Limit access to critical functions and monitor usage.

29.1.2. 4.3 Sanitize, Validate, and Filter LLM Outputs/Responses

Adversaries employ prompt injection to get the LLM application to do or say something it should not. Although trying to prevent or detect the attempted inject should be considered necessary, the complexity and nuance of LLM applications make it obvious that merely controlling the input should not be considered sufficient. Additionally, prompt injection primarily focuses on intentional abuse or misuse, yet inputs could still result in undesirable LLM application responses or behaviors. Much as validation and filtering of inputs proves vital, so too is properly handling and assessing outputs. Keep in mind that, like inputs, multiple layers and levels of output might exist in a complex LLM application, such as one that employs web search, function calling, tool use, or downstream LLMs. Output should not be construed to refer only to what would be presented to an end user

29.1.3. 4.4 Employ the Principle of Focused Functionality (and Agency)

Models continuously evolve, acquiring tremendous new capabilities and achieving previously unthinkable milestones. Despite this, LLM applications should offer as limited functionality as is acceptable. Since the 1990s, Bruce Schneier has been offering some version of the mantra, "The worst enemy of security is complexity." In designing agents, it is advisable to explicitly define and limit the functions and tools (code interpreters, web search, and other external APIs) the agent requires access to in order to fulfill its tasks. Avoid assigning multiple tools to an agent and apply the principle of least privilege.

30. CISA

30.1. 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>

30.1.1. 1.2.1 Understand the Secure AI System Development Lifecycle

Design the AI system with security considerations in mind from its inception, including using robust coding, protocols, and data protection measures.

30.1.2. 1.3.2 Educate Personnel on AI - Developing Clear Standard Operating Procedures

Developing clear standard operating procedures (SOPs) for all operations (including AI-related operations), interventions, and incidents to support stakeholder awareness of their roles and responsibilities in managing AI-enabled OT systems.

30.1.3. 2.1 Consider the OT Business Case for OT

Before incorporating an AI system into their OT environment, critical infrastructure owners and operators should assess if AI technologies are the most appropriate solution for their specific needs and requirements compared to other technologies. Critical infrastructure owners and operators should further consider whether an established capability meets their needs before pursuing more complex and novel AI enabled solutions. While AI comes with unique benefits, it is an evolving technology that requires continuous evaluation of risks.

30.1.4. 2.2.7 - Prioritising OT Data Protection

Prioritize the protection of critical types of OT data, including the following: Engineering Configuration Data. These include network diagrams, asset inventories, documentation on operations sequences, safety-related information, logic diagrams, and schematics. These data points have enduring value and are highly valuable to cyber adversaries. Ephemeral OT Data. Data from industrial measurement technology, especially process measurement technology (e.g., voltage or temperature, pressure levels, mass/volume flow rates) can provide insight into organizational activities or system behavior. If that data is used to train or update an AI model, the data may become accessible or stored (statistically) for a longer period of time in the model. As such, securing these data points can be important for protecting intellectual property (IP) and patterns of activity

30.1.5. 2.4.2

Integrate AI systems into their overall security and cybersecurity framework (see 3.2 Integrating AI Into Existing Security and Cybersecurity Frameworks).

30.1.6. 2.4.3

Add AI security considerations to a comprehensive security strategy that also includes traditional cybersecurity considerations, such as data encryption, access controls, and intrusion detection systems. Critical infrastructure owners and operators should define and validate security clauses in cloud contracts, explicitly outlining any AI security responsibilities, compliance standards, and support provisions, including data protection, access controls, incident response, and audit capabilities. o Cloud providers should provide detailed documentation that outlines security obligations specific to AI capabilities, in addition to traditional cloud security shared responsibility models

30.1.7. 2.4.4

Consider existing OT infrastructure and assess and develop an integration plan for AI systems.
- Consider using test infrastructure before deployment to production systems, if possible (see Principle 3 – Establish AI Governance and Assurance Frameworks).

30.1.8. 2.4.5

Encourage push-based architectures where data is pushed out of the OT network for AI systems to use without persistent access into the OT network

30.1.9. 2.4.6

Prioritize the organization's control over critical functions that AI systems may integrate with or enable when hosting AI systems locally or in the cloud. Ensure there are failsafe mechanisms that revert to traditional automation or manual for any AI-enabled system processes.

30.1.10. 2.4.7

Integrate AI systems the same as any new OT systems: test AI systems for safety impacts (e.g., latency, interoperability) and verify they work within existing device management policy

30.1.11. 3.1.5 - Continuously validate and verify the performance of AI systems

Make sure they (AI systems) meet the organization's objectives and regulatory requirements.

30.1.12. 3.2 Integrating AI into Existing Security and Cyber Security Frameworks

When integrating AI into OT environments, critical infrastructure owners and operators should consider the existing security and cybersecurity frameworks that govern these systems and embed AI system assessments within existing risk evaluation, mitigation, and monitoring processes. This means that traditional cybersecurity requirements, vulnerability management, and critical infrastructure regulations must be factored in when integrating AI systems.

30.1.13. 3.4 Navigating Regulatory and Compliance Considerations for AI in OT

Critical infrastructure owners and operators should evaluate the applicability of current AI technical standards in their OT domain as AI technical standards are rapidly evolving.

30.1.14. 4.2.2 - Design functional safety procedures that account for the AI system

Each critical infrastructure sector has its own safety states and procedures. Per Principle 2 – Consider AI Use in the OT Domain, critical infrastructure owners and operators should review how they are integrating the AI system into their existing procedures and create new safe use and implementation procedures that focus on the AI system integration into the OT environment.

31. Cloud Security Alliance (CSA)

31.1. AI Controls Matrix

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

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

31.1.1. AIS-02

Establish, document and maintain baseline requirements for securing applications.

31.1.2. AIS-04

Define and implement a secure software development lifecycle (SDLC) process for application requirements analysis, planning, design, development, testing, deployment, and operation in accordance with security requirements defined by the organization.

31.1.3. AIS-11

Establish security boundaries for agents.

31.1.4. AIS-15

Implement mechanisms enabling the model to clearly distinguish user-provided input instructions from data and system instructions (e.g., system prompts).

31.1.5. DSP-07

Develop systems, products, and business practices based upon a principle of security by design and industry best practices.

31.1.6. DSP-08

Develop systems, products, and business practices based upon a principle of privacy by design and industry best practices. Ensure that systems' privacy settings are configured by default, according to all applicable laws and regulations.

31.1.1.7. DSP-24

Ensure training-data differentiation and relevance to the intended use of the AI Model.

31.1.1.8. IAM-11

Define, implement and evaluate processes and procedures for customers to participate, where applicable, in the granting of access for agreed, high risk (as defined by the organizational risk assessment) privileged access roles.

31.1.1.9. IAM-14

Define, implement and evaluate processes, procedures and technical measures for authenticating access to systems, application and data assets, including multifactor authentication for at least privileged user and sensitive data access. Adopt digital certificates or alternatives which achieve an equivalent level of security for system identities.

31.1.1.10. IAM-17

Define policy and procedure for "need to know" access to knowledge, information and data within the organization and in the context of the AI system to be applied when regulating access to resources.

31.1.1.11. IAM-18

When allowing model output modification of AI generated output, establish a role for this access and allow changes only by authorized identities.

31.1.1.12. IAM-19

Restrict agents' access to the tools and plugins necessary for the activity or use case at hand, ensuring adherence to the principles of need-to-know and least privilege.

31.1.1.13. IPY-01

Establish, document, approve, communicate, apply, evaluate and maintain policies and procedures for interoperability and portability including requirements for: a. Communications between application interfaces b. Information processing interoperability c. Application development portability d. Information/Data exchange, usage, portability, integrity, and persistence Review and update the policies and procedures at least annually or upon significant changes.

31.1.1.14. MDS-07

Define, implement, and evaluate processes, procedures, and technical measures for Model Hardening to mitigate relevant adversarial attacks as identified in the Threat Analysis and Adversarial Threat Analysis.

31.1.1.15. MDS-13

Adopt secure model formats and processes for AI model serialization where applicable.

31.1.1.16. TVM-02

Establish, document, approve, communicate, apply, evaluate and maintain policies and procedures to protect against malware and malicious instructions. Review and update the policies and procedures at least annually or upon significant changes.

32. Microsoft

32.1. 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>

32.1.1. Secure AI resources 3 - Apply platform-specific security controls

Different AI deployment models face distinct security threats based on their architecture and exposure points. Platform-tailored controls address the specific vulnerabilities present in each deployment type. Follow dedicated security guidance based on your deployment model:
Azure PaaS security
Azure IaaS security

32.2. 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>

32.2.1. A3.1

Document in the Impact Assessment how the system's use will solve the problem posed by each intended use, recognizing that there may be multiple valid ways in which to solve the problem.

32.2.2. A3.2

Define and document for each model in the AI system:

- 1) the model's proposed inputs and how well they represent the concepts they are intended to represent; include analysis of the limitations of this representation,
- 2) the model's proposed output and how well it represents the concept it is intended to represent; include analysis of the limitations of this representation, and
- 3) limitations to the generalizability of the resulting model based on the training and testing data that will be used.

32.2.3. A3.3

Define and document Responsible Release Criteria for this Goal. Include:

- 1) a concise definition of the problem being solved in the intended use,
- 2) performance metrics and their Responsible Release Criteria, and
- 3) error types and their Responsible Release Criteria.

32.2.4. A3.6

Provide documentation to customers which describes the system's:

- 1) intended uses, and
- 2) evidence that the system is fit for purpose for each intended use.

When the system is a platform service made available to external customers or partners, include this information in the required Transparency Note.

32.2.5. A4.1

Define and document data requirements with respect to the system's intended uses, stakeholders, and the geographic areas where the system will be deployed. Document these requirements in the Impact Assessment.

32.2.6. T1.6

If there are Responsible Release Criteria for metrics or rubrics that have not been met, consult with the reviewers named in the Impact Assessment, and in the case of Sensitive Uses, with the Office of Responsible AI, to develop a plan detailing how the gap will be managed until it can be closed. Document that plan.

32.2.7. F1.6

Reassess the system design, including the choice of training data, features, objective function, and training algorithm, to pursue the goals of:

- 1) improving performance for any identified demographic group that does not meet any target minimum performance level, and
 - 2) minimizing performance differences between identified demographic groups, paying particular attention to those that exceed the target maximum, while recognizing that doing so may appear to affect system performance and that it is seldom clear how to make such tradeoffs.
- Consult with your attorney to determine your approach to this, including how you will identify and document tradeoffs.

32.2.8. F1.7

Identify and document any justifiable factors, such as circumstantial and other operational factors (e.g., "background noise" for speech recognition systems or "image resolution" for facial recognition systems), that account for:

- 1) any inability to meet any target minimum performance level for any identified demographic group, and
- 2) any remaining performance differences between identified demographic groups.

32.2.9. F2.6

Reassess the system design, including the choice of training data, features, objective function, and training algorithm, to pursue the goal of minimizing differences between the rates at which resources and opportunities are allocated to identified demographic groups, paying particular attention to those that exceed the target maximum difference, while recognizing that doing so may appear to affect system performance and it is seldom clear how to make such trade-offs.

Consult with your attorney to determine your approach to this, including how you will identify and document trade-offs.

32.2.10. F3.5

Reassess the system design, including the choice of training data, features, objective function, and training algorithm, to pursue the goal of minimizing the potential for stereotyping, demeaning, and erasing the identified demographic groups.

32.2.11. RS1.5

Define intended uses, if any, where additional operational factors, more narrow or different acceptable ranges, or lower acceptable error rates (including false positive and false negative error rates), are advised to ensure reliability and safety. Document your conclusions.

33.1. Secure AI Framework

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

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

33.1.1. Privacy Enhancing Technologies

Use technologies that minimize, de-identify, or restrict use of PII data in training or evaluating models.

33.1.2. Input Validation and Sanitization

Block or restrict adversarial queries to AI models.

33.1.3. Adversarial Training and Testing

Use techniques to make AI models robust to adversarial inputs (i.e. prompts) in the context of their use in applications.

33.1.4. Agent Permissions

Use least-privilege principle as the upper bound on agentic system permissions to minimize the number of tools that an agent is permitted to interact with and the actions it is allowed to take. An agentic system's use of privileges should be contextual and dynamic, adapting to the specific user query and trusted contextual information. This design also applies to agents that have access to user information. For example, an agent asked to fill out a form or answer questions should share only contextually appropriate information and can be designed to dynamically minimize exposed data using reference monitors.

34. European Commission

34.1. Assessment List for Trustworthy Artificial Intelligence (ALTAI)

Link: <https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment>

Link: <https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment>

34.1.1. REQUIREMENT #2 Technical Robustness and Safety

A crucial requirement for achieving Trustworthy AI systems is their dependability (the ability to deliver services that can justifiably be trusted) and resilience (robustness when facing changes). Technical robustness requires that AI systems are developed with a preventative approach to risks and that they behave reliably and as intended while minimising unintentional and unexpected harm as well as preventing it where possible. This should also apply in the event of potential changes in their operating environment or the presence of other agents (human or artificial) that may interact with the AI system in an adversarial manner. The questions in this section address four main issues: 1) security; 2) safety; 3) accuracy; and 4) reliability, fall-back plans and reproducibility.

34.1.2. REQUIREMENT #3 Privacy and Data Governance

Closely linked to the principle of prevention of harm is privacy, a fundamental right particularly affected by AI systems. Prevention of harm to privacy also necessitates adequate data governance that covers the quality and integrity of the data used, its relevance in light of the domain in which the AI systems will be deployed, its access protocols and the capability to process data in a manner that protects privacy.

34.2. 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>

34.2.1. 1.2.4 Reliability and Reproducibility

It is critical that the results of AI systems are reproducible, as well as reliable. A reliable AI system is one that works properly with a range of inputs and in a range of situations. This is needed to scrutinise an AI system and to prevent unintended harms. Reproducibility describes whether an AI experiment exhibits the same behaviour when repeated under the same conditions. This enables scientists and policy makers to accurately describe what AI systems do. Replication files can facilitate the process of testing and reproducing behaviours.

34.2.2. 1.5.1 Avoidance of unfair bias

Identifiable and discriminatory bias should be removed in the collection phase where possible. The way in which AI systems are developed (e.g. algorithms' programming) may also suffer from unfair bias. This could be counteracted by putting in place oversight processes to analyse and address the system's purpose, constraints, requirements and decisions in a clear and transparent manner. Moreover, hiring from diverse backgrounds, cultures and disciplines can ensure diversity of opinions and should be encouraged.

34.2.3. 2.2.3 Standardisation

Standards, for example for design, manufacturing and business practices, can function as a quality management system for AI users, consumers, organisations, research institutions and governments by offering the ability to recognise and encourage ethical conduct through their purchasing decisions. Beyond conventional standards, coregulatory approaches exist: accreditation systems, professional codes of ethics or standards for fundamental rights compliant design. Current examples are e.g. ISO Standards or the IEEE P7000 standards series, but in the future a possible 'Trustworthy AI' label might be suitable, confirming by reference to specific technical standards that the system, for instance, adheres to safety, technical robustness and transparency.

35. MITRE

35.1. ATLAS Framework

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

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

35.1.1. AML.M0006 - Use Ensemble Methods

Use an ensemble of models for inference to increase robustness to adversarial inputs. Some attacks may effectively evade one model or model family but be ineffective against others.

35.1.2. AML.M0009 - Use Multi-Modal Sensors

Incorporate multiple sensors to integrate varying perspectives and modalities to avoid a single point of failure susceptible to physical attacks.

35.1.3. AML.M0020 - Generative AI Guardrails

Guardrails are safety controls that are placed between a generative AI model and the output shared with the user to prevent undesired inputs and outputs. Guardrails can take the form of validators such as filters, rule-based logic, or regular expressions, as well as AI-based approaches, such as classifiers and utilizing LLMs, or named entity recognition (NER) to evaluate the safety of the prompt or response. Domain specific methods can be employed to reduce risks in a variety of areas such as etiquette, brand damage, jailbreaking, false information, code exploits, SQL injections, and data leakage.

35.1.4. AML.M0021 - Generative AI Guidelines

Guidelines are safety controls that are placed between user-provided input and a generative AI model to help direct the model to produce desired outputs and prevent undesired outputs.

Guidelines can be implemented as instructions appended to all user prompts or as part of the instructions in the system prompt. They can define the goal(s), role, and voice of the system, as well as outline safety and security parameters.

35.1.5. AML.M0022 - Generative AI Model Alignment

When training or fine-tuning a generative AI model it is important to utilize techniques that improve model alignment with safety, security, and content policies.

The fine-tuning process can potentially remove built-in safety mechanisms in a generative AI model, but utilizing techniques such as Supervised Fine-Tuning, Reinforcement Learning from Human Feedback or AI Feedback, and Targeted Safety Context Distillation can improve the safety and alignment of the model.

35.1.6. AML.M0026 - Privileged AI Agent Permissions Configuration

AI agents may be granted elevated privileges above that of a normal user to enable desired workflows. When deploying a privileged AI agent, or an agent that interacts with multiple users, it is important to implement robust policies and controls on permissions of the privileged agent. These controls include Role-Based Access Controls (RBAC), Attribute-Based Access Controls (ABAC), and the principle of least privilege so that the agent is only granted the necessary permissions to access tools and resources required to accomplish its designated task(s).

35.1.7. AML.M0027 - Single-User AI Agent Permissions Configuration

When deploying an AI agent that acts as a representative of a user and performs actions on their behalf, it is important to implement robust policies and controls on permissions and lifecycle management of the agent. Lifecycle management involves establishing identity, protocols for access management, and decommissioning of the agent when its role is no longer needed. Controls should also include the principle of least privilege and delegated access from the user account. When acting as a representative of a user, the AI agent should not be granted permissions that the user would not be granted within the system or organization.

35.1.8. AML.M0028 - AI Agent Tools Permissions Configuration

When deploying tools that will be shared across multiple AI agents, it is important to implement robust policies and controls on permissions for the tools. These controls include applying the principle of least privilege along with delegated access, where the tools receive the permissions, identities, and restrictions of the AI agent calling them. These configurations may be implemented either in MCP servers which connect the agents to the tools calling them or, in more complex cases, directly in the configuration files of the tool.

35.1.9. AML.M0030 - Restrict AI Agent Tool Invocation on Untrusted Data

Untrusted data can contain prompt injections that invoke an AI agent's tools, potentially causing confidentiality, integrity or availability violations. It is recommended that tool invocation be restricted or limited when untrusted data enters the LLM's context.

The degree to which tool invocation is restricted may depend on the potential consequences of the action. Consider blocking the automatic invocation of tools or requiring user confirmation once untrusted data enters the LLM's context. For high consequence actions, consider always requiring user confirmation.

35.1.10. AML.M0031 - Memory Hardening

Memory Hardening involves developing trust boundaries and secure processes for how an AI agent stores and accesses memory and context. This may be implemented using a combination of strategies including restricting an agent's ability to store memories by requiring external authentication and validation for memory updates, performing semantic integrity checks on retrieved memories before agents execute actions, and implementing controls for monitoring of memory and remediation processes for poisoned memory.

35.2. SAFE-AI

Link: https://atlas.mitre.org/pdf-files/SAFEAI_Full_Report.pdf

Link: https://atlas.mitre.org/pdf-files/SAFEAI_Full_Report.pdf

35.2.1. Unauthorized access to environment, platform/tool

Malicious actors can exploit unauthorized access to perturb valid inputs to AI models, causing them to consistently generate incorrect decisions. If safeguards are not in place to validate inputs, AI-enabled systems may be vulnerable to attacks that inject instructions or commands to an AI model, causing it to execute unauthorized tasks or generate erroneous outputs. Also, be wary of "off-label use" where an AI component is developed outside of an organization's security safeguards, or a component has been lifted from one context or application and then "fine-tuned" to be used in a different setting.

35.2.2. Insecure deserialization – embedding and executing remote unapproved code or other malicious activities

Malicious users can sometimes exert control over an AI enabled system by finding a command sequence that abuses the logic of the system and causes it to execute unauthorized tasks or generate incorrect behavior. The most prominent recent examples of this are the prompt attacks that some large language models are vulnerable to. Additionally, some AI models use procedural representations of knowledge and models (e.g., as in a rule-based system). These representations might be compromised by an adversary to enable the execution of malicious code. All procedural data and input data must be rigorously sanitized and validated.

35.2.3. Backdoor and malware insertion

Given the tendency of AI-enabled systems to depend on massive data stores and somewhat complex models and decision logic, it can be difficult to identify all unsecured points of entry vulnerable to backdoor and malware insertion attacks. Attackers can manipulate data in all phases of the system lifecycle, exploit vulnerabilities in AI algorithms and models, or use a variety of other techniques to insert backdoors into AI systems that get triggered once the AI is deployed. It is important to anticipate potential threats and AI-related attack surfaces during the design phase, secure and verify data and software during development, and establish testing procedures to regularly monitor system components for data/model drift, changes in performance, or other AI system behavior issues once it is deployed. If a potential attack is detected, information about errors and attack patterns must be shared with incident databases.

35.2.4. Indirect Prompt Injection

Adversaries may devise malicious prompts that cause the AI component to act in unintended ways. The attacks may be designed to bypass defenses or allow the adversary to issue privileged commands. The attack is indirect when the AI component ingests the malicious prompt from a separate data source (e.g., text or multimedia from a website, chat plugins) as part of its normal operation. Plugins may be vulnerable to an indirect prompt injection attack that uses the AI component to exfiltrate the history of a user conversation with an external website. The user may never be aware of the prompt injection. This type of injection can be used by the adversary to target the PII of the user.

35.2.5. Direct Prompt Injection

Adversaries may devise malicious prompts to an AI component causing it to act in unintended ways. Direct prompt injections are often an attempt to manipulate the AI component to generate harmful content or issue privileged commands to gain a foothold on the system, including placing the AI component in a state in which it will freely respond to any user input, bypassing controls or guardrails placed on the AI component.

35.2.6. Excessive Agency

Excessive Agency refers to situations where AI components have access to APIs, plugins, extensions, and tools with capabilities that go beyond what is necessary to support the AI component operations. Excessive permissions, unnecessary functionality, and unchecked authority to act autonomously are all examples of excessive agency that can result in unintended and unacceptable application behaviors with potentially damaging consequences. To mitigate these risks, developers need to limit extension capabilities (functionality, permissions, and autonomy) to only what is absolutely necessary, track user authorization, require human approval for all actions, and implement authorization in downstream systems.

35.2.7. Content Manipulation

Content manipulation poses a significant threat to Google DocumentAI, particularly in the context of OCR. Malicious actors can intentionally alter documents by changing numbers or formatting to deceive the AI, resulting in errors and misclassifications. Subtle content alterations can result in incorrect data extraction and faulty decision-making, compromising the integrity of the documents processed by the AI. By deceiving AI-enabled systems via document content manipulation, attackers can undermine trust in the automated document processing system.

35.2.8. Evade AI model

Adversaries can craft input data designed to prevent AI models from correctly identifying the contents of the data. For example, an adversary might introduce subtle perturbations that cause the model to misclassify or overlook meaningful information. This technique can be used to evade downstream tasks where machine learning is utilized by exploiting weaknesses in the AI model algorithms. Additionally, the adversary may evade machine learning-based virus/malware detection or network scanning tools towards the goal of a traditional cyber-attack.

35.2.9. Robotic Process Automation Permissions

Robotic Process Automation (RPA) bots are responsible for handling and manipulating sensitive data. If access controls and policies are not properly implemented, the bots can cause damage to systems and data due to errors. RPA bots are vulnerable to adversarial attacks, such as Evasion Attack, where malicious actors manipulate input data to deceive the bots, causing them to perform unintended actions, misclassify data, or corrupt data. Monitoring unauthorized access and insider threats is crucial, as bots with excessive permissions can be misused for malicious purposes. Ensuring robust security measures and continuous monitoring can mitigate these risks and protect sensitive data.

36. NIST

36.1. 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>

36.1.1. MAP 1.1

Intended purposes, potentially beneficial uses, context-specific laws, norms and expectations, and prospective settings in which the AI system will be deployed are understood and documented. Considerations include: the specific set or types of users along with their expectations; potential positive and negative impacts of system uses to individuals, communities, organizations, society, and the planet; assumptions and related limitations about AI system purposes, uses, and risks across the development or product AI lifecycle; and related TEVV and system metrics.

36.1.2. MAP 1.3

The organization's mission and relevant goals for AI technology are understood and documented.

36.1.3. MAP 1.4

The business value or context of business use has been clearly defined or – in the case of assessing existing AI systems – re-evaluated.

36.1.4. MAP 1.6

System requirements (e.g., "the system shall respect the privacy of its users") are elicited from and understood by relevant AI actors. Design decisions take socio-technical implications into account to address AI risks.

36.1.5. MAP 3.1

Potential benefits of intended AI system functionality and performance are examined and documented.

36.1.6. MAP 3.3

Targeted application scope is specified and documented based on the system's capability, established context, and AI system categorization.

36.1.7. MANAGE 1.1

A determination is made as to whether the AI system achieves its intended purposes and stated objectives and whether its development or deployment should proceed.

36.1.8. MANAGE 2.2

Mechanisms are in place and applied to sustain the value of deployed AI systems.

36.2. 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>

36.2.1. PO.1.1

Identify and document all security requirements for the organization's software development infrastructures and processes, and maintain the requirements over time.

Include AI model development in the security requirements for software development infrastructure and processes.

Identify and select appropriate AI model architectures and training techniques in accordance with recommended practices for cybersecurity, privacy, and reproducibility.

36.2.2. PO.1.2

Identify and document all security requirements for organization-developed software to meet, and maintain the requirements over time.

Organizational policies should support all current requirements specific to AI model development security for organization-developed software. These requirements should include the areas of AI model development, AI model operations, and data science. Requirements may come from many sources, including laws, regulations, contracts, and standards.

36.2.3. PO.2.3

Obtain upper management or authorizing official commitment to secure development, and convey that commitment to all with development-related roles and responsibilities.

Leadership should commit to secure development practices involving AI models.

36.3. 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>

36.3.1. Practice 1.1: Anticipate model capabilities - 2

Consider also collecting evidence from less similar models, such as those that are significantly more or less powerful than the model (e.g., if a less capable model presents a misuse risk, prioritize measuring for that risk).

36.3.2. Practice 1.1: Anticipate model capabilities - 3

Consider other factors beyond model characteristics, such as resources available to develop the model, when relating the proxy models' capabilities to anticipated ones.

36.3.3. Practice 1.1: Anticipate model capabilities - 4

Collect information about the prevalence of misuse, the usefulness of proxy models for potentially harmful or dual-use real-world tasks, and the efficacy of safeguards.

36.3.4. Practice 2.2 Establish an organizational plan to manage misuse risk - 1

Plan for information security and physical security practices necessary to manage misuse risk, when appropriate.

36.3.5. Practice 2.2 Establish an organizational plan to manage misuse risk - 2

Plan research needed to implement and evaluate adequate safeguards. If necessary, base these plans on the mapping established in Practice 2.1.

36.3.6. Practice 2.2 Establish an organizational plan to manage misuse risk - 3

Plan for decision-making about model deployment. This plan should include the organization's plan for pre-deployment testing and determining which appropriate risk mitigations are available.

36.3.7. Practice 3.2: Maintain security practices sufficient to prevent unauthorized access - 5

Only develop a model that relies on confidentiality to manage misuse risk when the risk of threat actors gaining unauthorized access to the model is sufficiently mitigated. When the organization is made aware of an increased risk of unauthorized access, adjust or halt further development until the risk of unauthorized access is adequately managed.

36.3.8. Practice 5.1: Implement safeguards proportionate to the model's misuse risk - 2

Consider implementing safeguards at various stages of model development, such as before, during, and after training and once the model has been integrated into a downstream system, as well as other mitigation measures. Some examples are included in Appendix B.

36.3.9. Practice 5.2: Assess misuse risk based on implemented safeguards - 1

Identify planned deployments that could impact misuse risk. For example, consider whether the planned deployment could impact the number of actors who have access to a model or the level of access that they would have.

36.3.10. Practice 5.3: Adopt appropriate deployment strategies based on misuse risk assessments - 2

Consider deployment strategies that provide additional real-world evidence to inform risk assessments without presenting significant misuse risks.

36.3.11. Practice 6.2: Respond to incidents of model misuse - 3

Plan for how identified instances of misuse will inform future development and deployment decisions, especially when reducing access to the model may not be possible.

36.4. 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>

36.4.1. GV.OC-01

The organizational mission is understood and informs cybersecurity risk management

36.4.2. GV.OC-04

Critical objectives, capabilities, and services that external stakeholders depend on or expect from the organization are understood and communicated

36.4.3. PR.AA-05

Access permissions, entitlements, and authorizations are defined in a policy, managed, enforced, and reviewed, and incorporate the principles of least privilege and separation of duties

36.5. 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>

36.5.1. 2.2.5 Adversarial training

Introduced by Goodfellow et al. [144] and further developed by Madry et al. [232], adversarial training is a general method that augments training data with adversarial examples generated iteratively during training using their correct labels. The stronger the adversarial attacks for generating adversarial examples are, the more resilient the trained model becomes. Adversarial training results in models with more semantic meaning than standard models[379], but this benefit usually comes at the cost of decreased model accuracy on clean data. Additionally, adversarial training is expensive due to the iterative generation of adversarial examples during training.

36.5.2. 2.2.5 Randomized smoothing

Proposed by Lecuyer et al. [207] and further improved by 17 NIST AI 100-2e2025 March 2025 Cohen et al. [94], randomized smoothing is a method that transforms any classifier into a certifiable robust smooth classifier by producing the most likely predictions under Gaussian noise perturbations. This method results in provable robustness for ℓ_2 evasion attacks, even for classifiers trained on large-scale datasets, such as ImageNet. Randomized smoothing typically provides certified prediction to a subset of testing samples, the exact number of which depends on factors such as the size of the potential perturbations or the characteristics of the training data and model. Recent results have extended the notion of certified adversarial robustness to ℓ_2 -norm bounded perturbations by combining a pretrained denoising diffusion probabilistic model and a standard high-accuracy classifier [62]. Li et al. [211] developed a taxonomy for the robustness verification and training of representative algorithms. They also revealed the characteristics, strengths, limitations, and fundamental connections among these approaches, along with theoretical barriers facing the field.

36.5.3. 2.2.5 Formal verification

Another method for certifying the adversarial robustness of a neural network is based on techniques from FORMAL METHODS. Reluplex uses satisfiability modulo theories (SMT) solvers to verify the robustness of small feedforward neural networks[191]. AI2 is the first verification method applicable to convolutional neural networks using abstract interpretation techniques [136]. These methods have been extended and scaled up to larger networks in follow-up verification systems, such as DeepPoly [346], ReluVal [394], and Fast Geometric Projections (FGP) [131]. Formal verification techniques have significant potential for certifying neural network robustness but are limited by their lack of scalability, computational cost, and restriction in the type of supported algebraic operations such as addition, multiplication, etc.

36.5.4. 2.3.1 Robust training

An alternative approach to mitigating availability poisoning attacks is to modify the ML training algorithm to increase the robustness of the resulting model. The defender can train an ensemble of multiple models and generate predictions via model voting [37, 209, 395]. Several papers apply techniques from robust optimization, such as using a trimmed loss function [109, 179]. Rosenfeld et al. [314] proposed the use of randomized smoothing to add noise during training to provide protection against label-flipping attacks.

36.5.5. 2.3.3 Trigger reconstruction

This class of mitigations aims to reconstruct the backdoor trigger, assuming that it is at a fixed location in the poisoned training samples. NeuralCleanse by Wang et al. [390] developed the first trigger reconstruction approach and used optimization to determine the most likely backdoor pattern that reliably misclassifies the test samples. The initial technique has been improved to reduce performance time on several classes and simultaneously support multiple triggers inserted into the model [163, 411]. A representative system in this class is Artificial Brain Simulation (ABS) by Liu et al. [221], which stimulates multiple neurons and measures the activations to reconstruct the trigger patterns. Khaddaj et al. [193] developed a new primitive for detecting backdoor attacks and a corresponding effective detection algorithm with theoretical guarantees.

36.5.6. 2.3.3 Certified defenses

Several methods for achieving certified defenses against data poisoning attacks have been proposed in the literature. BagFlip [440] is a modelagnostic defense that extends randomized smoothing [94] and combines training data bagging with adding noise to both training and testing samples. Deep Partition Aggregation [209] and Deep Finite Aggregation [396] are certified defenses that partition the training data into disjointed subsets and train an ensemble method on each partition to reduce the impact of poisoned samples. Recently, FCert [398] provides a certified defense against data poisoning in few-shot classification settings used for both vision and text data.

36.5.7. 2.4.5 Data Reconstruction

The discovery of reconstruction attacks against aggregate information motivated the rigorous definition of differential privacy (DP) [112, 113], an extremely strong definition of privacy that guarantees a bound on how much an attacker with access to the algorithm output can learn about each individual record in the dataset. The original pure definition of DP has a privacy parameter ϵ (i.e., privacy budget), which bounds the probability that the attacker with access to the algorithm's output can determine whether a particular record was included in the dataset. DP has been extended to the notions of approximate DP, which includes a second parameter δ that is interpreted as the probability of information accidentally being leaked in addition to ϵ and Rényi DP [246].

36.5.8. 3.2.3 Data Poisoning Attacks

GenAI poisoning mitigations largely overlap with PredAI poisoning mitigations (see Sec.2.3). For preventing data poisoning with web-scale data dependencies, this includes verifying web downloads as a basic integrity check to ensure that domain hijacking has not injected new sources of data into the training dataset [57]. That is, the provider publishes cryptographic hashes, and the downloader verifies the training data. Data filtering can also attempt to remove poisoned samples, though detecting poisoned data within a large training corpus may be very difficult. While traditional software supply chain risk management practices such as vulnerability scanning of model artifacts can help manage some kinds of AI supply chain risks, new approaches are required to detect vulnerabilities in models such as those introduced through model poisoning attacks. Current proposed approaches include using methods from the field of mechanistic interpretability to identify backdoor features [67] and detecting and counteracting triggers when they are seen at inference time. Beyond these mitigations, risks can be reduced by understanding models as untrusted system components and designing applications such that risks from attacker-controlled model outputs are reduced [266].

36.5.9. 3.3.3 Direct Prompting Attacks - Interventions during pre-training (2) and post-training (3)

A range of training strategies have been proposed to increase the difficulty of accessing harmful model capabilities through direct prompt injection, including safety training during pre-training [197] or post-training [147, 445], adversarial training methods [340], and other methods to make jailbreak attacks more difficult [447].

36.5.10. 3.3.3 Direct Prompting Attacks - Interventions during deployment (5) - Prompt instruction and formatting techniques

Model instructions can cue the model to treat user input carefully, such as by wrapping user input in XML tags, appending specific instructions to the prompt, or otherwise attempting to clearly separate system instructions from user prompts [14, 206, 219].

36.5.11. 3.4.4 Indirect Prompt Injection Attacks

Various techniques (see Sec. 3.3.3) can be used throughout the development and deployment life cycle (Fig. 7) to mitigate attacks, including:

- Several training techniques have been developed to mitigate against indirect prompt injection, including fine-tuning task-specific models [296] and training models to follow hierarchical trust relationships in prompts [387].
- Detection schemes have been proposed to detect indirect prompt injection, and many LLM-based defenses have been designed to mitigate both direct and indirect prompt injection [6, 18, 154, 204, 313].
- A range of input processing methods have been proposed to combat indirect prompt injection, including filtering out instructions from third-party data sources [146], designing prompts to help aid LLMs in separating trusted and untrusted data (i.e., spotlighting [160, 206]), or instructing models to disregard instructions in untrusted data [206].

Many of the defenses described in the context of direct prompt injection can also be adapted to mitigate indirect prompt injection. Because current mitigations do not offer full protection against all attacker techniques, application designers may design systems with the assumption that prompt injection attacks are possible if a model is exposed to untrusted input sources, such as by using multiple LLMs with different permissions [145, 405] or by allowing models to interact with potentially untrustworthy data sources only through well-defined interfaces [410]. Additionally, public education efforts can inform model users and application designers of the risks of indirect prompt injection [266].