Google Cloud

Securing AI: Similar or Different?

Anton Chuvakin, John Stone, Tanya Popova-Jones at Office of the CISO, Google Cloud

Office of the CISO

A quick paper for CISOs and their teams on securing AI

chuvakin@ tpop@ thestone@ and OCISO team

Introduction and definitions

Artificial intelligence (AI) is rapidly becoming a ubiquitous part of our lives. From powering our smartphones to driving our cars, AI is already having a major impact on the way we live and work. As AI continues to develop, it's important as a security practitioner to consider the security implications of these technologies. Along with security, there are also concerns over important issues like safety and privacy. This paper will focus primarily on security, and we will cover other topics in the future.

There are many ways AI systems can be vulnerable to attack. For example, AI systems can be tricked into making incorrect decisions by feeding them malicious data. Additionally, AI systems can be hacked to gain access to sensitive data or to take control of the system itself. Attacks on AI systems can impact the security of the system, but could also cause harm or privacy issues.

It's therefore essential to take steps to secure AI systems. This includes ensuring that AI systems are fully part of cyber governance, that they're protected from malicious attacks, and that their security is regularly reviewed. In addition to securing AI systems, it's also important to ensure that AI is used in a secure way, as even the securely developed platform may be used in a less secure manner (public cloud being a great example here).

Naturally, the approach to securing AI heavily depends on the type of AI (generative AI has its own threat scenarios), your AI use cases (AI writing marketing copy vs. AI writing production code has different risks), and your role in the AI ecosystem (using consumer-grade AI or developing your own AI applications calls for different security safeguards). As a result, the discussion should be on specific AI use cases, and each new use case should merit a new risk assessment.

For an overall framework for thinking about AI security, review Google's Secure AI Framework document.

Scope

What we cover in this paper:

- Cybersecurity topics related to developing, deploying, and utilizing AI systems by business and other organizations

What we do not cover in this paper:

- Use of consumer-focused AI delivered via the web
- Security products powered by AI-based features or using AI for security
- Use of AI by attackers
- The ethical implications of AI for humanity (these are broadly covered in Responsible AI principles)

AI development process

The AI development life cycle is standardized and consists of the following steps:

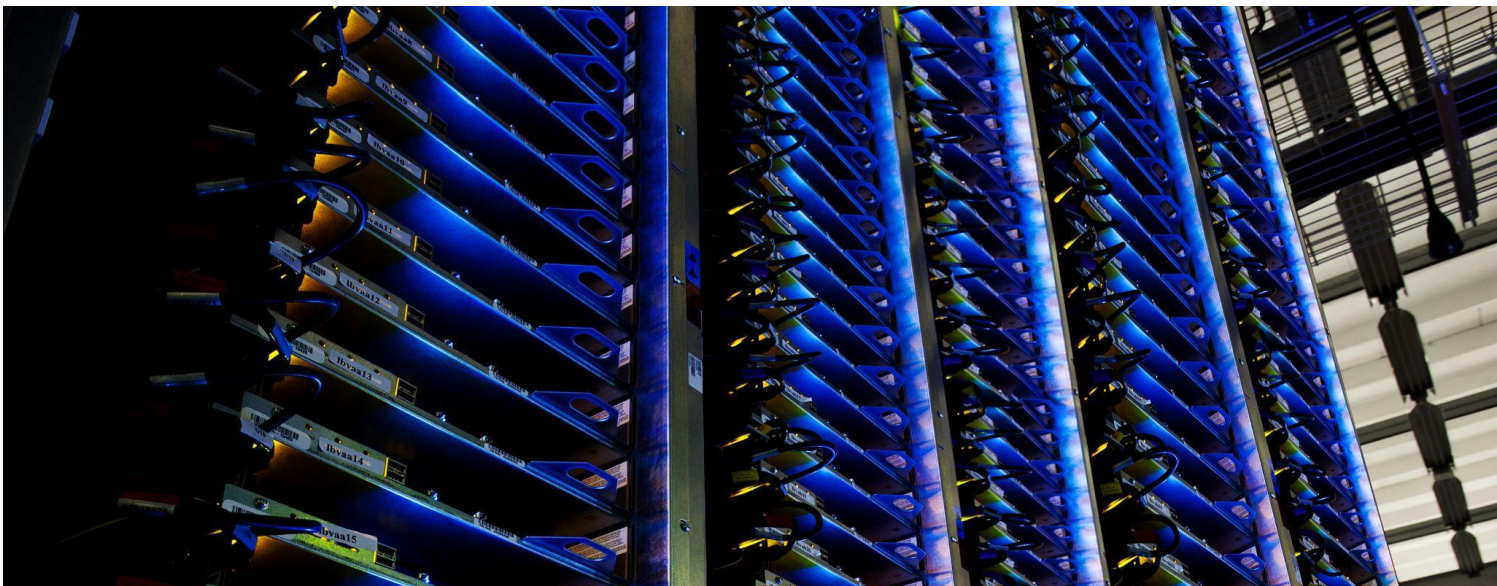
01	Opportunity discovery and problem definition	• Identify the business problem or opportunity that you want to solve with AI. • Evaluate for regulatory relevance, possible bias, etc. • Choose the right data, algorithms, and evaluation metrics
02	Data collection and preparation	• Collect (or generate) the data that you will need to train your AI model. • Analyze, label, transform, and ingest the data. • Establish end-to-end data governance based on regular risk assessment and threat modeling.
03	Model design and development	• Design and develop the AI model that will address the business opportunity that you have defined. • Build in mechanisms to assess and mitigate potential risks. • Audit model performance and screen output • Build in the facilities to explainability and human intervention
04	Model training, fine-tuning, and testing	• Train the AI model to production. • Test the model to make sure that it is performing as expected. • Analyze outcomes to compare with expected results • Implement security measures throughout
05	Model development and integration with end-product	• Deploy the AI model production • Make the model available to users so that they can solve the problem that you have defined. • Implement runtime security safeguards.
06	Model behavior/outcome monitoring and adjustment	• Monito the behavior and outcomes of the AI model to make sure that it is performing as expected. • Understand how users may be using the model to identify signs of badness. • Adjust the model over time to account for changes in the data or the environment. • Implement output filtering measures

Opportunity discovery and problem definition:

This is the first and most important step in the AI development life cycle. You need to be clear on the problem and business opportunity you want to solve with AI, as well as start evaluating for regulatory relevance, possible bias, and so on. Advances in generative AI (GenAI) are driving new transformative opportunities to create more complex and valuable products and services, improve customer experience, and maximize employee productivity. Building AI responsibly, however, also requires answering hard questions across a product's life cycle. It's critical to adopt a principled approach and identify, assess, and mitigate potential harmful impacts before AI products are developed and deployed. This will help you to choose the right data, algorithms, and evaluation metrics.

Data collection and preparation:

Once you have defined the problem, you need to collect (or generate, if using synthetic data), analyze, label, transform, and ingest the data that you will need to train your AI model. This step has compliance, security, and possibly privacy implications as the data may be corrupt, affected by malefactors, stolen, or what the AI model will produce at its output. It's important to establish end-to-end data governance based on regular risk assessment and threat modeling. This includes: data management processes (including data quality, data provenance, data acquisition, and data preparation); processes and controls for proper data handling and use (including data redaction and deletion); and data security controls, including access control, data encryption, and differential privacy controls.



Model design and development:

In this step, you will design and develop the AI model that will address the business opportunity that you have defined. Build in mechanisms to assess and mitigate potential risks, audit model performance, screen output, and build in the facilities for explainability and human intervention. Security implications here cover both model sensitivity and broader "supply chain" aspects.

Model training, fine-tuning, and testing:

Once you have designed and developed your AI model, you need to train it on the data that you have collected. Once the model is trained, you need to test it to make sure that it's performing as expected, analyzing outcomes to compare with expected results (such as via model benchmarks and evaluations). Naturally, security aspects encompass training process governance, and various types of security testing, including both formal evaluations and manual/automated testing.

Model deployment and integration with end product:

Once you are satisfied with the performance of your AI model, you need to deploy it to production. This means making the model available to users (most likely via product integration) so that they can use it to solve the problem that you have defined. Many types of runtime security safeguards apply here, from interface protection, input filtering, and access management to detection and response.

Model behavior/outcome monitoring and adjustment:

Once the model is in production, you need to monitor its behavior and outcomes to make sure that it's performing as expected. At this stage, you will also want to understand how users may be using the model as well to identify signs of misuse/abuse/attack. You may also need to adjust the model over time to account for changes in the data or the environment. Security implications cover output filtering of various types, for example

AI customer and AI developer roles

Securing AI covers the spectrum of activities with roles played by the AI tool developer, model developer, tool operator, data custodian, customer, and possibly partners and model providers of various types (for example, those who tuned the models for specific use cases or industries).

The role of tool and model vendors is to design, deploy, and operate appropriate security controls in their AI systems. If they are providing a foundational model, they should be able to describe the secure development life cycle they used in their model development. They should describe the controls they have applied to the data used for training, validation, and testing of the model and the methods they have for ongoing monitoring.

Customers are responsible for configuring and using these controls to protect their own data and systems. This varies drastically based on how the organization uses AI technologies and often includes things like:

- Choosing a vendor and partners with strong security track records
- Reviewing the vendor's security controls
- Configuring the controls to meet their specific needs
- Monitoring the controls to make sure that they are working properly

The AI development life cycle is a complex process, but it's essential for ensuring the success of AI projects. By following the steps in the life cycle and working with a trusted vendor, you can increase your chances of success.

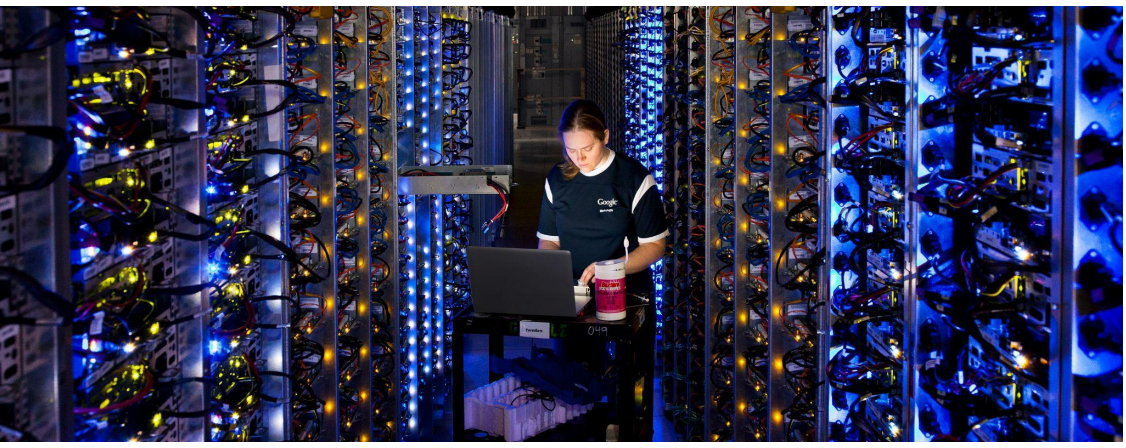
Security: What changes and what stays the same

AI systems introduce new security risks not present in traditional systems. For example, some AI systems can be more easily fooled by adversarial examples, which are carefully crafted inputs that can cause an AI system to make incorrect predictions. AI systems can also be used to generate synthetic data that can be used to attack other systems.

There are other risks – such as underfitting of the models, bias, and so on, – which also need to be considered, but these should form part of the use-case instantiation and would typically be handled by the team responsible for developing the model. A helpful hint is to have a good responsibility assignment matrix (RACI) to understand the responsibilities of the different stakeholders related to AI.

As mentioned, this paper focuses primarily on the security aspects. Many of the same security principles that apply to traditional systems also apply to AI systems. For example, it's still important to implement security controls such as network access control and threat detection, and data encryption. Additionally, it's important to train AI systems on data that's representative of the targeted use cases and to test them for vulnerabilities, misconfigurations, and misuse cases. It's important to recognize these risks and take steps to mitigate them.

Let's review common security domains and cover some similarities and differences.



Governance

The topic of governance can mean different things depending on the structure of your organization. It could take the form of security governance, overall governance (of which security governance can be a subset), or specific governance related to the selection and use of AI. For example, as part of your governance you have rules in place that disallows the creation of any model that can cause harm.

From all the categories we review in the paper, the topic of data and code (model) governance is extra critical for AI. When the organization was storing data, topics like lineage, provenance, accuracy, ownership, and integrity were important but perhaps not critical. When you train AI with this data, and the intent of it making decisions, writing code, and communicating, the stakes rise dramatically.

Similarities	Differences
Governance frameworks for AI and traditional systems can include similar elements, such as risk assessment, threat modeling, security controls, inventory, versioning, incident response, and so on	AI systems are often complex and opaque, which can make it difficult to understand how they make decisions. Explainability is a key topic for AI systems so that users can trust their decisions.
Both traditional enterprise software systems and AI systems store and process sensitive data	AI systems may be used to make decisions that have a significant impact on people's lives, which means significant risks that need to be considered and managed in a sound governance process.
Both types of systems need to have strong data security controls in place to protect this data from unauthorized access, use, disclosure, disruption, modification, or destruction	The composition of stakeholders expands to include other disciplines for judgment-based decisions on various use cases, such as the Responsible AI, Privacy, and Ethics teams[1]
Both have an approval mechanism with relevant stakeholders and clearly defined roles and responsibilities	Human oversight requirements and prohibition of particular use cases which may cause harm
	Transparency requirements to advise the end user that they're interacting with an AI, particularly for chatbots

Threats

Similarities	Differences
Both types of systems need to be protected from unauthorized access, modification, or destruction of data and other classic threats	AI systems are vulnerable to a variety of AI-specific threats, including adversarial examples, data poisoning, and other AI flaws like bias
Both systems must be protected from malware and other malicious software	Given the importance of data to programming AI, data-centric attacks are high on the threat list, adding to the list of digital supply-chain threats
Data theft is a concern with both AI and traditional systems	AI systems may be more vulnerable because they are more complex and rely on data for programming
Supply-chain attacks affect both AI and traditional systems	Online systems (those that are actively learning) that train on input potentially allow it to drift from its intended operational use
Threat model process and practice applies to both systems	AI systems may be used to create new types of threats, both attacks that target the AI system itself and attacks against other systems
	GenAI systems may suffer from hallucination problems (a response by an AI that does not seem to be justified by its training data, either because it's insufficient, biased, or too specialized)

Application/Product security

Similarities	Differences
Both traditional enterprise software and AI systems are susceptible to traditional application security vulnerabilities like input injection and various overflows. Security misconfigurations also remain an issue.	Threat models need to be updated to include new threats as they emerge.
These vulnerabilities can be exploited by attackers to gain unauthorized access to systems, steal data, or disrupt operations.	Product testing should include adversarial AI testing, a type of testing that traditional application security engineers may not be familiar with.
Threat modeling is still a good idea for both types of systems and should be part of a routine practice when these are built and deployed.	AI systems may be more vulnerable to security issues because they are more non-deterministic and thus complex, have a large number of moving parts, and rely on data for programming. They may also be built and maintained by staff that are not trained in security practices and the use of a software development life cycle (SDLC).
	AI systems are often developed using machine learning and other techniques that are not well understood by security professionals.
	AI systems are often trained on proprietary data or models. It's important to protect this data and models – not just software code – from unauthorized access or disclosure.
	AI systems are often trained on data that is collected from the real world, which can introduce security risks, uncontrolled inputs, and other issues.

[1] Some of these may be included in traditional AI governance ad-hoc, but would be stateful participants in AI Governance Committees, for instance.

Data security and privacy

Similarities	Differences	
Both traditional enterprise software systems and AI systems store and process sensitive and regulated data, sometimes in large volumes. The data types may include personal information, financial data, and intellectual property.	If models are online, they may be learning from their inputs. Sanitization and screening of this training data extends traditional approaches to input validation.	Output filtering is also important, since generative AI may produce texts and data that may subject your organization to additional risks, such as perpetuating social biases or using toxic language
Both types of systems need to have strong data security controls in place to protect this data from unauthorized access, use, disclosure, disruption, modification, or destruction.	AI systems are often trained on large amounts of data, which makes them more interesting to an attacker and may have a greater impact if breached. It's important to protect this data from unauthorized access, modification, or destruction.	When personal data is used to train models, several additional issues may arise: the data may not be processed lawfully, fairly, or in a transparent manner in relation to the data subject, it may not have been collected for this specific purpose, it may not remain accurate over time, and so on.
Both AI and traditional systems require the same types of data security controls, such as access control, encryption, and data backups.	Input filtering is required, since data may be derived from multiple sources, not all of which are accurate and complete – particularly if sourced from the public domain.	The models may potentially memorize data that was provided as training input. An attacker can extract memorized data from the model without authorization.
	Use of unstructured data for training purposes (in addition to traditional text includes biometrics, videos, and images) heightens risk as traditional tools aren't typically calibrated to detect such use cases.	Because it involves data in the creation of and use of the service, it's bound by significant data regulations and privacy law, contractual limitations, and social norms.
	AI systems can also be used to extract sensitive data from other systems or to include malicious code when generative AI is used by developers to write code.	

Network and endpoint security

Similarities	Differences
Both AI and traditional systems are connected to the network, which makes them susceptible to the same types of network security threats, such as unauthorized access, denial-of-service (DoS) attacks, and data breaches.	AI systems are often more complex than traditional systems and access multiple other systems over the network, which can make them more difficult to secure
Both AI and traditional systems connected to the public internet via web access and APIs need network security controls	

Threat detection and response

Similarities	Differences	
Detections should be tuned according to the threat model, and that threat model should be developed by competent humans using an expanded library of risks specific to AI, ML, and Deep Learning	Since AI systems can be used to generate synthetic data that can be used to attack other systems, detecting this becomes a priority. For example, an AI system could be used to generate fake login credentials that can be used to gain unauthorized access to a system.	Furthermore, online AI systems that are constantly learning on the data they are processing are subject to drift that necessitates monitoring.
Both traditional enterprise software systems and AI systems are susceptible to a variety of threats and need to have strong threat detection and response capabilities in place to identify and mitigate these threats	AI systems can also be used to automate attacks. Detecting such abuse of the model should form part of your abuse-detection criteria.	Detecting threatening input and output becomes an issue with GenAI; input may threaten the AI system while the output from a AI system may threaten other systems or humans.
Both AI and traditional systems require a human element to detect and respond to threats, such as security analysts and incident responders	AI systems may be more vulnerable to threats that are difficult to detect, such as adversarial examples.and complete – particularly if sourced from the public domain.	Detection needs to cover the range of known malicious uses of the AI system – for example attacks against the AI safeguards or using AI to generate attacks against other systems – and be able to rapidly respond to newly discovered threats.

Security assessment and validation

Similarities	Differences
Both AI and traditional systems can benefit from the same types of security assessment and validation tools, such as vulnerability scanners and penetration testing tools	AI systems are often built on complex algorithms that are difficult to understand and analyze
Both AI and traditional systems require a human element to conduct security assessments and validations, such as security analysts and security engineers	AI systems are trained on large amounts of data, which can be a valuable target for attackers and may require new types of security validation approaches, not just code reviews and scanning
Assessments should be tuned according to the threat model, that itself was done by competent humans using an expanded library of risks specific to AI, ML, and Deep Learning	AI models must be validated to ensure that they are accurate, reliable, and unbiased by testing the models on a variety of data sets and by using statistical methods to assess their performance

Security assessment and validation

Similarities	Differences
Both types of systems are vulnerable to DoS attacks and availability risks	AI systems are often used in real-time applications, which means that a DoS attack or availability risk can have a significant impact on the performance of the system
	AI systems require substantial processing and memory capacities, and can therefore be more easily targeted by application-level DoS
	New types of availability risks and DoS may exist in intelligent and reasoning systems like AI

Overall, AI systems introduce new security risks not present in traditional systems. It's important to be aware of these risks and to take steps to mitigate them.

By understanding the differences between securing a traditional enterprise software system and an AI system, organizations can develop a more comprehensive security strategy to protect their AI systems from a variety of security threats. It's possible that potential for harm is exponentially higher with AI – that's one of the reasons for the global regulatory focus on bias assessments and explainability.

What to do about it?

Here are a few select recommendations:

Governance: Implement robust governance and security controls throughout the AI life cycle. Also, understand and document jurisdictional regulations as they emerge and evolve

Inventory: Understand the AI systems used in your organization. This includes understanding how they work, what data they use, and how they are used by employees or customers. The more you know about your AI systems, the better equipped you will be to identify and mitigate security risks

Data security: Implement robust security controls for data collection, data storage, data processing, and data use as well as related code and models.

Secure software development: Use secure development practices. This includes practices like code review, threat modeling, and penetration testing. SDLC practices must apply to both code and data.

Education: Educate users about security risks (this includes users, developers, and operators of AI systems). Educate more AI system designers about threat modeling and other security practices.

Secure deployment: Enable a set of tools and processes to filter and log inputs (such as prompts for GenAI) and outputs, control access, and deliver enterprise-grade security controls.

Threat detection and response: Monitor AI systems for security threats, including using security tools to monitor for malicious activity. The output of a generative model will need to be monitored: not only its state of deployment but the content of its output that may be an indication of a compromise.

Testing: Start the "AI red teaming" program using both security and AI experts. Review AI red teaming guidance from Google

Policy: If you don't develop AI systems but only use them, create an acceptable use policy (AUP) for AI, including GenAI.

Incident response: Respond to security incidents promptly and effectively. This includes having a plan for responding to security incidents involving AI systems.

For more recommendations and guidance on securing AI systems, be sure to check out Google's SAIF guide.

Appendix A Definitions

Artificial intelligence (AI) is a branch of computer science that deals with the creation of intelligent agents, which are systems that can reason, learn, and act autonomously. AI research has been highly successful in developing effective techniques for solving a wide range of problems, from game playing to assisting in medical diagnosis.

Specifically, generative AI (GenAI) is a type of AI that can create new data, such as text, images, or music. GenAI is under rapid development, and it has the potential to create new forms of art, entertainment, and communication.

Machine learning (ML) is a subset of AI that deals with the development of algorithms that can learn from data without being explicitly programmed. ML algorithms are trained on large datasets of examples, and they can then be used to make predictions about new data. ML has been used to achieve state-of-the-art results in a wide range of applications, including image recognition, natural language processing, and fraud detection.[2]

Deep learning is a type of machine learning that uses artificial neural networks to learn from data. Neural networks are inspired by the human brain, and they are able to learn complex patterns in data that would be difficult for traditional machine learning algorithms to learn.

Finally, automation is the use of technology to perform tasks that would otherwise be done by humans. Automation can be used to improve efficiency and productivity, and it can also be used to create new products and services.

Google Cloud offers Vertex AI and other AI services and tools for organizations and enterprises, in addition to consumer products like Bard.

[2] "While artificial intelligence encompasses the idea of a machine that can mimic human intelligence, machine learning does not. Machine learning aims to teach a machine how to perform a specific task and provide accurate results by identifying patterns.." (source)