Tony Bjerstedt, CISSP, CCSP

November 9, 2023

Enhancing Security in Generative AI Applications

with an Emphasis on OWASP LLM Top 10

Agenda

About me

What is Generative AI?

Security for AI or AI for Security

Securing Generative AI

About OWASP

OWASP Top 10 for LLM Applications

Summary

About Me

Email:
tony@bjerstedttechnologies.com
anthony.bjerstedt@sogeti.com

LinkedIn: linkedin.com/in/tbjerstedt
Twitter: @tbjerstedt

sogeti Part of Capgemini

AEA Association of Enterprise Architects Minnesota Chapter

CISSP® CCSP℠ CCSK v.4

aws certified Solutions Architect PROFESSIONAL

Microsoft CERTIFIED AZURE SOLUTIONS ARCHITECT EXPERT

TOASTMASTERS INTERNATIONAL

midwest architecture community collaboration

What is Generative AI?

Generative AI

Form of AI that "generates" various types of content

First introduced in 1960s

More awareness with deepfakes in 2014

ChatGPT introduced in 2018

https://www.brighttalk.com/resource/core/439892/webinar-deck_939501_939589.pdf

Large Language Models (LLMs)

Around for years

New chatbots

OpenAI, Anthropic, Google

Billions of parameters

Modern Chatbots

ChatGPT, Microsoft Bing API, Google Bard AI, GitHub Copilot X

https://www.brighttalk.com/resource/core/439892/webinar-deck_939501_939589.pdf

AI for Security or Security for AI

Good, Bad and Ugly

Good

- Better detection of advanced threats and vulnerabilities
- Improved security tools

Bad

- AI can help attackers
- Nefarious chatbots
  - WormGPT, FraudGPT
  - Custom malware and well-written spear-phising emails

Ugly

- Polymorphic malware, near-perfect deepfakes, supply chain attacks

Government Regulations

White House issued AI executive order (30 Oct 2023)

NIST AI Risk Management Framework (NIST AI 100-1)

Issued January 2023, Second draft 29 Sept 2023

Trustworthy and Responsible AI Resource Center

European Union AI Act

Passed on June 16, 2023 by the EU Parliament by overwhelming majority

Securing Generative AI

Avoid Public Tools for Proprietary Data

ChatGPT and other public sites are public

- Your data will be remembered and may be used
- ChatGPT on March 20, 2023 exposed subscriber data to other subscribers
  - Bleeping Computer: Over 100,000 ChatGPT accounts stolen via info-stealing malware

Microsoft and others provide solutions to build your own

Establish Some Principles

Example: Cisco Principles for Responsible Artificial Intelligence

- Transparency
- Fairness
- Accountability
- Privacy
- Security
- Reliability

Establish a Framework (Guardrails)

Example: Cisco Responsible AI Framework

- Guidance and Oversight
- Controls
- Incident Management
- Industry Leadership
- External Engagement
- Reliability

midwest architecture community collaboration

About OWASP

About OWASP

The Open Worldwide Application Security Project (OWASP) is a nonprofit foundation that works to improve the security of software

Vision

No more insecure software.

Mission

To be the global open community that powers secure software through education, tools, and collaboration.

https://owasp.org/about/

OWASP Projects

OWASP Top Ten

OWASP Mobile Application Security

OWASP Software Assurance Maturity Model

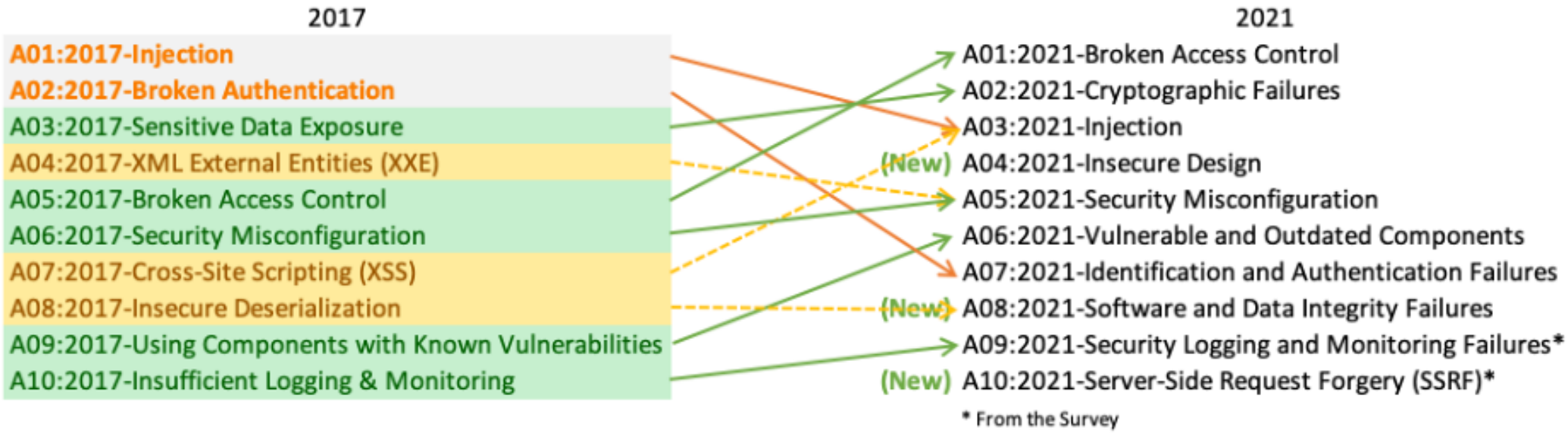
Web Security Testing Guide

OWASP Application Security Verification Standard

And more

OWASP Top 10

The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications.



https://owasp.org/www-project-top-ten/

midwest architecture community collaboration

OWASP Top 10 for LLM Applications

Introducing: OWASP Top 10 for LLM Applications

LLM01: Prompt Injection

LLM02: Insecure Output Handling

LLM03: Training Data Poisoning

LLM04: Model Denial of Service

LLM05: Supply Chain Vulnerabilities

LLM06: Sensitive Information Disclosure

LLM07: Insecure Plugin Design

LLM08: Excessive Agency

LLM09: Overreliance

LLM10: Model Theft

LLM01: Prompt Injection

Attackers can manipulate LLMs through crafted inputs, causing it to execute the attacker's intentions. This can be done directly by adversarially prompting the system prompt or indirectly through manipulated external inputs, potentially leading to data exfiltration, social engineering, and other issues.

LLM02: Insecure Output Handling

Insecure Output Handling is a vulnerability that arises when a downstream component blindly accepts large language model (LLM) output without proper scrutiny. This can lead to XSS and CSRF in web browsers as well as SSRF, privilege escalation, or remote code execution on backend systems.

LLM03: Training Data Poisoning

Training Data Poisoning refers to manipulating the data or fine-tuning process to introduce vulnerabilities, backdoors or biases that could compromise the model's security, effectiveness or ethical behavior. This risks performance degradation, downstream software exploitation and reputational damage.

LLM04: Model Denial of Service

Model Denial of Service occurs when an attacker interacts with a Large Language Model (LLM) in a way that consumes an exceptionally high amount of resources. This can result in a decline in the quality of service for them and other users, as well as potentially incurring high resource costs.

LLM05: Supply Chain Vulnerabilities

Supply chain vulnerabilities in LLMs can compromise training data, ML models, and deployment platforms, causing biased results, security breaches, or total system failures. Such vulnerabilities can stem from outdated software, susceptible pre-trained models, poisoned training data, and insecure plugin designs.

LLM06: Sensitive Information Disclosure

LLM applications can inadvertently disclose sensitive information, proprietary algorithms, or confidential data, leading to unauthorized access, intellectual property theft, and privacy breaches. To mitigate these risks, LLM applications should employ data sanitization, implement appropriate usage policies, and restrict the types of data returned by the LLM.

LLM07: Insecure Plugin Design

Plugins can be prone to malicious requests leading to harmful consequences like data exfiltration, remote code execution, and privilege escalation due to insufficient access controls and improper input validation. Developers must follow robust security measures to prevent exploitation, like strict parameterized inputs and secure access control guidelines.

LLM08: Excessive Agency

Excessive Agency in LLM-based systems is a vulnerability caused by over-functionality, excessive permissions, or too much autonomy. To prevent this, developers need to limit plugin functionality, permissions, and autonomy to what's absolutely necessary, track user authorization, require human approval for all actions, and implement authorization in downstream systems.

LLM09: Overreliance

Overreliance on LLMs can lead to serious consequences such as misinformation, legal issues, and security vulnerabilities. It occurs when an LLM is trusted to make critical decisions or generate content without adequate oversight or validation.

LLM10: Model Theft

LLM model theft involves unauthorized access to and exfiltration of LLM models, risking economic loss, reputation damage, and unauthorized access to sensitive data. Robust security measures are essential to protect these models.

Summary

Summary

Consider building you own Gen AI application

Establish Principles and a Framework with guard rails

Secure your application by addressing vulnerabilities

Questions?

midwest architecture community collaboration

Extras

References

OWASP

Organization: https://owasp.org/

OWASP Top 10 for LLM Applications, Version 1.1, Slides for Version 1.1

Webinars

AI Security: Practical Steps to Take Amidst the Hype (Cisco)

Strange Bedfellows: Software, Security and the Law (Mend.io)

The Impact of Artificial Intelligence on the Cybersecurity Industry and Part 2

Articles

Infosecurity Magazine: What the OWASP Top 10 for LLMs Means for the Future of AI Security

medium.com: The OWASP Top 10 for LLMs: A Light-hearted Look at Serious Security

Podcasts

Security Now Podcast

Unsecurity Podcast A weekly information security podcast (FR Secure)