ChatGPT

The New Generation of OWASP Leaders: Why OWASP's Mission Is More Vital Than Ever

Introduction and Personal Perspective

As someone who has been involved with the Open Web Application Security Project (OWASP) for about twenty years – including time as a chapter leader and even a board member – I've witnessed the ebbs and flows of our community. Today, I see a *new generation of OWASP leaders* stepping up with unprecedented energy and purpose. This is happening at a time when OWASP's mission has never been more important. In this document, I want to share my observations on how the landscape of application security is changing and why the rise of new leadership within OWASP is perfectly timed to meet these challenges. My perspective is a personal one, born of decades of involvement in OWASP, and it is **reflective** of where we've come from and **optimistic** about where we are headed.

OWASP's Evolution: From Maturity to Re-Energized Mission

For much of the last decade, application security (AppSec) had reached a certain level of maturity and steady rhythm. Many of the classic web vulnerabilities became well-understood, and the industry developed effective guidance and tools to handle them. We knew, for example, how to build safer web applications – at least in theory. OWASP played a key role in this, producing influential guides, frameworks, and tools that helped developers avoid common pitfalls. Security best practices (like input validation, output encoding, authentication controls, etc.) were codified and widely taught. The result was a vibrant AppSec industry and community that had solutions (or at least mitigations) for most fundamental problems.

Yet, this very maturity led to a *plateau* of sorts. The field wasn't seeing a lot of brand-new vulnerability categories; instead, it was dealing with known issues at greater scales. In fact, the core OWASP Top Ten web application risks remained surprisingly consistent over the years – the same basic mistakes and vulnerabilities persisted for a long time [1]. This slow evolution meant that although security programs grew, the *excitement* around groundbreaking new AppSec problems had lessened. Some of the challenge became more about scaling up execution (e.g. how to perform security testing and fixes across thousands of apps) rather than inventing new approaches. As a community, we focused on how to integrate security into the development lifecycle and how to get organizations to prioritize and **scale** their application security efforts.

Another factor contributing to the lull was that security often remained a "non-functional requirement." Businesses would sometimes treat security as an add-on or a box to check, rather than a core feature. Many companies got away with shipping applications that were far from secure, yet suffered no immediate consequences. Why? In many cases, attackers didn't heavily target those weaknesses – or at least, not in ways that caused visible damage. Those of us performing security assessments frequently found far more vulnerabilities than actual exploits in the wild. In short, a lot of software was (and still is) insecure, but due to lack of widespread exploitation, organizations felt *comfortable enough* to continue business as usual. This dynamic made it challenging to push for significant improvements or investments in security; if nothing blew up, it was easy to ignore the ticking time bombs.

However, I believe this period of relative stability was the calm before the storm. The foundational work done by OWASP and the AppSec community in the past decades laid **critical groundwork** – we established what "good security hygiene" looks like, and we trained a generation of developers and security professionals. That groundwork is now proving invaluable as we enter a dramatically new phase of technology and risk.

The GenAI Revolution: A New Threat Landscape Emerges

In the past couple of years, we have witnessed an explosion of *Generative AI* and autonomous agents being integrated into applications and workflows. Large Language Models (LLMs) and so-called "agentic AI" systems are now capable of making decisions, taking actions, and dynamically interacting with other software and services. This is not just incremental change – it's a paradigm shift. We are effectively introducing a whole new **insider** in our systems: one that is non-deterministic, incredibly powerful in its capabilities, and not fully understood even by its creators.

To put it plainly, an AI agent (for example, an LLM-based assistant with tools) is just software – it calls APIs, it parses input and produces output. In that sense, it should be subject to the same kinds of security considerations as any other software component or microservice. But there's a twist: these agents exhibit *autonomy and adaptability*. Unlike traditional code that follows predetermined logic, an AI agent can rewrite its approach on the fly (within the constraints of its programming and training) and it can be influenced in real-time by **natural language inputs**. That extra autonomy means it can introduce risks beyond the usual concerns of a typical app – the agent might make an unpredictable decision or take an action the original developers never explicitly anticipated [2]. In effect, it's as if we've deployed a highly capable **black box** into the heart of our applications, one that learns and acts in ways we might not fully foresee.

This new threat landscape is both *old and new*. It's "old" in that many of the risks resemble what we've seen before in AppSec, just manifesting in a new form. It's "new" in that the scale and nature of what these AI systems can do is beyond what traditional software typically would. Consider the issue of injection attacks, for example. We used to worry about SQL injection or command injection, where an attacker tricks an application into running unintended commands by mixing data with code. Now, we face **prompt injection** – where an attacker supplies crafted input that causes an AI model to follow malicious instructions it was not meant to follow. In both cases, the root cause is the same: failing to separate untrusted data from the code/commands the system executes. As one observer aptly put it, *we're making the same mistake of treating data and instructions as the same thing – that mistake gave us SQL injection in the Web 2.0 era, and now it's giving us prompt injection in the AI era* [3]. The OWASP community has spent decades advocating for strict separation of code and data, input validation, and output encoding; those principles are as relevant as ever, perhaps even more so now.

Let's look at why these AI-driven systems introduce a heightened sense of urgency for security:

- **Autonomous Agents with Unpredictable Behavior:** Unlike a conventional app, an AI agent can take initiative. By design, agents can be proactive – they don't just respond to a single request, they can *plan*, chain together actions, and pursue goals. In the context of a business workflow, that might mean an AI agent that reads incoming emails, summarizes them, takes action like creating support tickets, or even executing transactions. If such an agent goes awry (whether through malice or error), it could cause a lot of damage very quickly. We now have to worry about scenarios like an agent deciding to exfiltrate data or make unauthorized changes, potentially without a human directly triggering that specific action.

- **New Kinds of Malicious Abuse:** There are multiple ways an AI system can become dangerous. First, an AI agent could be **malicious by design** – imagine an attacker creating an agent specifically to infiltrate or harm systems (a sort of AI malware). Second, an initially well-intentioned agent could be **manipulated by an attacker** through something like prompt injection or by feeding it poisoned data, effectively turning it into a malicious actor from within. Third, even absent external attackers, an agent might **cause harm by pursuing misguided goals** – the classic "paperclip maximizer" thought experiment, where an AI single-mindedly pursues a goal to the detriment of everything else, comes to mind. In summary, an AI agent might do the wrong thing either because it *wants* to (malicious intent), because someone *tricked* it into misbehaving, or simply because it *doesn't know any better* and lacks alignment with human values or business intent.

- **Unsolved Problems (Prompt Injection and Beyond):** One of the stark realities we face is that **prompt injection is still an open problem**. We have not yet found a foolproof method to prevent an attacker from inserting malicious instructions into an AI's input or context. In traditional apps, we eventually learned to parameterize queries and encode outputs to mostly solve things like SQL injection and XSS. In AI systems, finding an analogous solution is harder – the model's "instructions" and "data" often co-mingle in the same input context. There isn't a clear boundary or sandbox between the AI's operational commands and the user-supplied data. This lack of isolation means an attacker can potentially smuggle commands via any channel the AI can read (a user query, a document the AI is asked to summarize, etc.). The AI might then confidently execute those commands (or take those malicious instructions to heart) as if they were legitimate. It's a perfect example of an old weakness wearing new clothes. And it's not just prompt injection; issues like data poisoning (contaminating training data to influence model behavior) or model extraction attacks (stealing the model) are new fronts where we're still developing defenses.

- **Agents as the "Insider" Threat:** Perhaps the most worrying aspect for companies is that AI agents collapse the distance between *vulnerability* and *action*. Traditionally, an attacker would exploit a vulnerability in a web app and then have to manually or via scripts carry out malicious actions. Now, if an attacker can compromise an AI agent (say, by prompt injection), the agent *itself* might carry out the nefarious actions automatically. We've essentially given some of these agents the keys to operate various tools: they can send emails, write code, execute transactions, or spin up infrastructure – all based on their programming and input. If compromised, they become a tireless insider threat operating at machine speed. This significantly changes the risk calculus. A single successful injection could cascade into a full-blown incident much faster than a traditional breach might.

It's no surprise, then, that I sometimes refer to the current state of AI integrations as a bit of a **"perfect storm"** or even a potential chaos scenario. We have extremely powerful new components (AI models and agents) being rapidly inserted into complex environments without full understanding of their failure modes. It's akin to placing a high-voltage experimental reactor at the center of your data center – it can do amazing things, but if it melts down, the blast radius is huge. The rush to adopt AI is understandable (the capabilities are game-changing), but it has outpaced our typical security caution. In many organizations, dozens of AI features and agent-based automations are getting bolted on to products and internal processes. These often come via third-party vendors or open-source tools, which means they might be introduced without rigorous security vetting. In some cases, even the security teams themselves aren't fully aware that an AI agent has been embedded in a business workflow. A recent commentary on the new OWASP Agentic AI Top 10 noted that *agents are slipping into real-world workflows faster than most teams expect* [4] – in other words, this is happening under our noses. The idea of putting a partially-understood autonomous agent in the middle of, say, your customer support

system or your CI/CD pipeline would have sounded crazy a few years ago, yet that's essentially what's occurring now across industries.

OWASP's Timeless Principles Confront Cutting-Edge Threats

The good news is that OWASP – as a community and an institution – was built for moments like this. We've always existed to identify emerging risks, distill them into actionable advice, and rally the community to build solutions (whether that be documentation, open-source tools, or training) to address those risks. In the face of the GenAI revolution, OWASP's principles and approach are not just relevant; they are absolutely **essential**.

Everything that OWASP has been teaching for two decades about building secure software applies to AI systems, arguably *even more so*. Consider some core OWASP tenets in light of AI:

- **Understand Your Attack Surface:** Just as OWASP guides developers to inventory their endpoints and third-party components, we now must inventory where and how AI models are integrated into our systems. What can they do? What do they connect to? What privileges do they have? Many organizations haven't fully mapped this, and that's dangerous.

- **Least Privilege and Sandboxing:** OWASP has long recommended limiting what each part of a system can do – don't give your web server root access if it doesn't need it, for example. The same goes for AI agents: if an agent is supposed to generate a report, maybe it doesn't need the ability to send emails or execute code on production servers. If we treat AI components with healthy distrust (assuming they *could* be compromised or simply make mistakes), we would isolate them in secure sandboxes and give them only the minimum necessary permissions. This mindset is exactly what is needed to prevent an AI from doing harm [5]. It's encouraging to see that some security folks are advocating for approaches like "assume the model is already compromised" and apply classic controls – network isolation, monitoring, failsafes – rather than hoping the AI will police itself [6]. In essence, **the solution is not to invent a whole new discipline of AI security from scratch, but to apply known security engineering practices to AI**.

- **Input Validation and Output Handling:** Every OWASP talk or cheat-sheet on web security will mention validating inputs and sanitizing outputs. In the AI context, "inputs" include user prompts and any data fed into the model, and "outputs" include the model's responses or actions. The OWASP community quickly recognized this parallel. For instance, the OWASP Top 10 for Large Language Model Applications (LLM Top 10) lists **Prompt Injection** as the number one risk (LLM01) and **Insecure Output Handling** as another major risk (LLM02) [7]. These items map almost directly to the concepts of injection attacks and output encoding/data handling issues that we've dealt with in traditional apps. Neglecting to sanitize what goes into or comes out of an AI system can lead to the AI being manipulated or its results being misused, just as failing to handle inputs/outputs in a web app could lead to SQL injection or XSS in the past.

- **Secure Development Lifecycle & Training:** OWASP has always emphasized that security is not a one-time effort; it's a process. We need threat modeling, secure design, code review, testing, and developer education. Now, with AI, we have new items to add to that process – e.g., *AI model threat modeling* (what if someone tries to poison our training data? what if someone abuses the model's API?), secure prompt engineering guidelines, and so on. But the framework of thinking in a structured way about security during design and development is directly applicable. In fact, many OWASP veterans are now spearheading the creation of **AI security testing**

methodologies and **red-teaming exercises** for AI systems, expanding on the legacy of projects like OWASP ASVS (Application Security Verification Standard) or OWASP Testing Guide.

OWASP as an organization has responded quickly to the rise of AI-related threats. A new initiative – the **OWASP Generative AI Security Project** – was launched to tackle these challenges head on. This project started in 2023 as a small group of concerned security professionals and researchers who realized there was a gap in guidance for AI and LLM security. In a short time, it has grown into a global community effort, with over 600 contributing experts from more than 18 countries and nearly 8,000 active community members driving it forward [8]. The mission of the project is broad and ambitious: to identify, document, and help mitigate the security and safety risks associated with generative AI technologies (including LLMs and agentic AI) [9]. In true OWASP fashion, this initiative is open-source and community-driven – a collective effort to develop actionable knowledge for everyone's benefit.

One of the flagship results of this initiative has been the creation of **new OWASP Top 10 lists** tailored to AI. We now have the *OWASP Top 10 for LLM Applications*, which enumerates the most critical vulnerabilities and pitfalls when building applications that use large language models [10]. This list is updated as the field evolves (much like the classic Top 10) and serves as an entry point for organizations to understand where things can go wrong with AI. Building on that, in late 2025 the community released the inaugural *OWASP Top 10 for Agentic AI Systems* (sometimes referred to as the Top 10 for Agentic Applications) [11]. This is the first comprehensive framework to specifically address security challenges posed by autonomous AI agents – those systems that don't just generate text, but actually **take actions** and make decisions across tools and environments [2].

Cover of the OWASP Top 10 for Agentic Applications 2026 report, a new community-driven framework addressing the distinct security risks of autonomous AI agents. This effort highlights how deeply the OWASP community is diving into AI security. For example, one of the top risks identified is **Agent Goal Hijack**, which occurs when an attacker manipulates an agent's goals or instructions, causing it to act against the user's intent [12]. In essence, it's a form of injection or manipulation – very much in line with the kinds of issues OWASP has tackled in the past, but manifested in an AI context. Other issues in the agentic Top 10 include things like **Tool Misuse** (an agent being tricked into using its tools for harmful purposes) and **Identity & Privilege Abuse**, among others. These speak to scenarios where an AI agent might misuse its capabilities or escalate its privileges in unintended ways, again echoes of long-standing security concerns (like misuse of functionality or privilege escalation) in a new form. The fact

that OWASP is publishing these lists and guidance so quickly is a testament to how relevant and necessary our collective knowledge is right now. It's worth noting that industry and other security organizations are paying close attention to these OWASP efforts – we're effectively charting unknown waters, and many are looking to the OWASP community for leadership and direction in securing AI.

A New Generation of OWASP Leaders with Purpose

Perhaps the most inspiring development I've observed is the rise of a **new generation of OWASP leaders** who are galvanized by these emerging challenges. Many of these individuals are relatively new to OWASP – some drawn from the AI research community, some from software development backgrounds who found their way into security because of AI, and some are younger professionals who see the AI security problem as *the problem* to solve in this decade. They are joining long-time OWASP contributors to carry the mission forward, and they bring a renewed sense of purpose.

This new generation isn't content to just maintain the status quo; they are actively *building* the future. They have organized projects like the OWASP GenAI Security Project mentioned above, authored new guides and checklists, and led training sessions at major conferences. For instance, at the recent OWASP AI Security events (one of which coincided with Black Hat this year), I saw these leaders presenting novel research on prompt injection, demonstrating tools to detect AI vulnerabilities, and fostering community discussions on AI ethics and safety. The energy was palpable. It reminded me of the early days of OWASP, when everything was new and we were all figuring it out together – except now the scope is broader and the stakes in some ways are higher.

One key difference I notice in the new leaders is a strong *sense of mission*. They genuinely believe (and I agree) that OWASP's role is critical to ensuring that the AI revolution doesn't compromise security or safety. There's a feeling that **"if we don't do it, who will?"** – in other words, if the community doesn't come together to solve prompt injection, or to create standards for AI model security, then the gaps will remain and potentially lead to disaster. This sense of responsibility is driving an incredible amount of volunteer effort and innovation. Long-time OWASP members are finding themselves re-energized by working alongside these passionate newcomers. I've personally found it incredibly rewarding to mentor some of these rising stars, sharing lessons from past security battles, while at the same time learning from their fresh perspectives and cutting-edge technical insights.

It's also worth noting that this wave of leadership is *highly collaborative*. The complexities of AI security cut across many domains – software engineering, machine learning, policy, ethics, etc. I see OWASP folks teaming up with academics, with AI company engineers, and with other organizations to tackle problems. The OWASP ethos of openness and community collaboration is proving to be a great glue here, bringing people together under a common banner. For example, the OWASP Top 10 for LLMs project grew to hundreds of contributors precisely because it welcomed input from anyone with expertise, regardless of their title or affiliation [8]. This openness is attracting talent that might not have traditionally been involved in OWASP's web security projects. Our tent is expanding, and it's bringing in the kind of multidisciplinary brainpower we need to solve AI security problems.

The **leadership** I refer to is not just about official titles or roles in the OWASP Foundation (though we do see new faces there too), but also about thought leadership. These are people publishing blog posts, giving conference talks, writing proof-of-concept code, and setting up community meetings to share knowledge – much like the OWASP pioneers did in the 2000s for web app security. It's a full-circle moment: OWASP made its name by being ahead of the curve on web vulnerabilities and training a generation of developers to care about security. Now, the next generation of OWASPers is doing the

same for AI. And just as before, this knowledge is being freely shared with the world, multiplying its impact.

Bridging Experience and Innovation: A United Community

With all this change, one might wonder: where does it leave those of us who have been around a long time? I actually think it leaves us in the best position possible – *working together across generations*. The challenges we face with AI are new enough to require fresh ideas, but they're also similar enough to past issues that the experience of seasoned AppSec professionals is incredibly valuable. This is a moment where the OWASP community is learning from its history to avoid "reinventing the wheel," while also bravely venturing into uncharted territory.

For the established OWASP leaders and veterans (myself included), our role is part cheerleader, part mentor, and part collaborator. We have to encourage and support these new initiatives, even if they're outside our old comfort zones. We have to provide guidance from our successes and mistakes of the past – for example, advising on how to structure an OWASP project, how to build consensus on a Top 10 list, or how to effectively advocate for change in industry. At the same time, we must stay humble and open to learning, because the truth is, no one is *truly* an expert in AI security yet – it's too new and evolving. I've been in security for decades and still find myself daily learning new things about machine learning models, vector databases, or AI prompt engineering quirks. That's exciting! It makes me feel like a student again, and that kind of continual learning is what drew many of us to security in the first place.

From a community perspective, I see OWASP as the big tent that can house all of this. We have the organizational framework – chapters, global conferences, project structure – that can empower the new generation to succeed. And importantly, we have a culture of **knowledge sharing** and **volunteerism** that is perfectly suited to tackle something as broad as AI security. No single company or government can solve these issues alone; it's going to take a community-driven approach, which is exactly OWASP's modus operandi.

I also want to emphasize how global and diverse this new wave is. OWASP has always been international, but the GenAI security push has shown that talent is everywhere: I've been on OWASP GenAI Project calls where there are experts dialing in from Nigeria, Brazil, India, Poland, Vietnam, you name it. Each brings a different perspective – whether it's how AI might be misused in their local financial apps, or insights into how translations by LLMs could introduce risks, etc. This diversity in thought is a huge strength. Security is a universal concern, and having a broad set of voices helps us foresee problems that a homogeneous group might miss. It also helps spread the OWASP mindset to new communities and industries that perhaps weren't deeply involved with OWASP before.

Why OWASP's Mission Has Never Mattered More

Bringing it all together, we are at a pivotal moment. The core mission of OWASP – *to improve the security of software* – is dramatically increasing in scope and impact. Software today is not just websites and mobile apps; it's AI models making decisions, it's complex supply chains of machine learning data and third-party APIs, it's software controlling physical systems via IoT and AI. The potential consequences of insecure software have always been serious (think data breaches, financial loss, etc.), but now they can be outright **dangerous** on a broader scale. An insecure AI system in a healthcare setting, for instance, could give harmful advice or leak sensitive patient data. An insecure AI agent in a critical infrastructure context could potentially disrupt services or damage equipment. These scenarios move software security from just an IT problem to a public safety and national security concern.

OWASP's role in all this is to be the beacon and the resource for anyone who wants to navigate this new landscape safely. The **OWASP Top 10** has been a de facto standard for web application security for years – now we're creating the analogous knowledge for AI (and doing so quickly, as seen with the LLM and Agentic Top 10 lists). Similarly, where OWASP provided cheat sheets and tools (like the OWASP ZAP proxy or dependency-check tools) for traditional apps, the community is now exploring and building tools to test AI systems (for example, automating prompt injection testing, scanning AI plugins for vulnerabilities, etc.).

Crucially, OWASP also has an educational mission. Never has it been more important to educate developers, product managers, and even executives about software security risks. In the age of AI, this means teaching about things like the dangers of plugging an LLM directly into critical workflows without proper guardrails, or the importance of monitoring AI outputs for integrity. The new generation of OWASP leaders are already active here – I've seen them give talks to CISO forums explaining AI risks in plain terms, and workshops to developers on secure AI coding practices. This translation of security knowledge to a wider audience is something OWASP has always done, and it's needed now more than ever.

For the broader OWASP community and its leadership, my message is one of **optimism and urgency**. We should be optimistic because we are relevant – more than ever. The surge of interest and talent coming into OWASP via AI security is proof that our mission resonates and is adapting to the times. We are not just reminiscing about past glories; we are actively shaping the future of security. There is a real sense that what we do in the next few years (the guidance we produce, the standards we set, the awareness we raise) will heavily influence how safely society traverses the AI revolution. That's a heavy responsibility, but it's also a privilege to be at the forefront of something so impactful.

The urgency comes from the fact that technology will not slow down. Companies large and small are adopting AI at a breakneck pace because of the competitive advantage and efficiency it can bring. Security cannot afford to be an afterthought here – we have to embed ourselves into this wave of innovation *now*, or risk playing catch-up later with potentially disastrous results. OWASP is our vehicle to do that at scale. Through OWASP, we can provide the templates and tools for others to follow, so each organization doesn't have to figure it all out from scratch (just as we did with web security best practices).

Conclusion: Embracing the Future Together

Reflecting on my twenty years with OWASP, I honestly have never been more excited about the organization's future. We find ourselves at the nexus of profound technological change and significant risk – exactly the kind of environment where OWASP's community thrives and proves its worth. The **new generation of OWASP leaders** gives me confidence that we have the passion, creativity, and drive needed to tackle whatever comes next. They are carrying forward the torch that was lit long ago, but they're also lighting new torches and expanding the light to areas we hadn't considered before.

To my fellow long-time OWASP members and leaders: now is the time to re-engage if you've been less involved, and to lend your wisdom and support to these new initiatives. Your experience is invaluable, and combined with the fresh energy on hand, we truly have an unstoppable community.

To the new members and leaders: thank you for reinvigorating OWASP with your enthusiasm and ideas. You've chosen to attach yourselves to our core mission of making software (and now AI) safer for everyone, and in doing so, you're not only continuing OWASP's legacy but also ensuring its evolution. Keep pushing us older folks out of our comfort zones – it's a good thing!

Finally, to the broader OWASP community and the industry at large: let's recognize that OWASP's work is **far from done**. In fact, it may never have been as critical as it is now. Software is eating the world, as the saying goes, and now AI is poised to eat software. Security must be a fundamental ingredient in this transformation. OWASP, with its open, community-driven approach, is uniquely positioned to inject security DNA into the AI-enabled software ecosystems of tomorrow.

By embracing the challenges of today and those to come, OWASP can ensure that security is woven into the fabric of innovation – not as an obstacle, but as an enabler of trust. I am optimistic that with our collective effort, we will look back years from now and see that this period was one of growth and positive impact for OWASP and for the world. **The future is unwritten, but with this new generation of leaders at the helm, I firmly believe the best chapters of OWASP are yet to come.**

Sources:

1. OWASP GenAI Security Project – *Project Background and Growth* [8] [9]
2. Dave Wichers (co-founder of OWASP Top 10) interview – *22 Years of OWASP Top 10* [1]
3. Matt Asay, InfoWorld – *Building AI Agents the Safe Way (Opinion)* [3] [6]
4. ActiveFence Blog – *OWASP Top 10 for Agentic AI Explained* [2] [4] [12] [11]
5. OWASP Top 10 for LLM Applications – *OWASP Foundation Project* [7]

---

[1] What 22 Years of OWASP Top 10 Really Tells Us About AppSec
https://www.appsecai.io/blog/what-22-years-of-owasp-top-10-really-tells-us-about-appsec

[2] [4] [10] [11] [12] Understanding the new OWASP Top 10 for Agentic AI Security
https://www.activefence.com/blog/owasp-agentic-top-ten/

[3] [5] [6] Building AI agents the safe way | InfoWorld
https://www.infoworld.com/article/4110056/building-ai-agents-the-safe-way.html

[7] [8] [9] OWASP Top 10 for Large Language Model Applications | OWASP Foundation
https://owasp.org/www-project-top-10-for-large-language-model-applications/