ChatGPT

Supercharging AppSec Threat Modeling Services with GenAI and Semantic Graphs

Executive Summary

Application security (AppSec) consulting is on the cusp of a transformation driven by Generative AI (GenAI) and semantic knowledge graphs. By integrating these technologies, AppSec services companies can **dramatically scale and enhance their Threat Modeling and Training offerings**. This white paper – co-authored by Dinis Cruz and ChatGPT Deep Research – outlines a strategic vision and concrete service proposals for next-generation AppSec consulting. Key opportunities include:

- **AI-Accelerated Threat Modeling at Scale:** Use GenAI to rapidly produce, customize, and maintain threat models. Large Language Models (LLMs) can enumerate threat scenarios, suggest mitigations, and even generate hundreds or thousands of threat models and security documents in hours [1], far beyond what manual efforts achieve.

- **Semantic Knowledge Graphs for Context-Rich Analysis:** Represent threats, assets, and mitigations as nodes in a **semantic graph** linked to business context, code, and industry frameworks. This turns static threat models into living knowledge bases that are **queryable and continuously updatable**. By mapping technical findings to business goals and compliance requirements, every threat is tied to the revenue streams, customer promises, or regulations it could impact. This "business context layer" drives data-driven risk conversations instead of spreadsheet archaeology.

- **Personalized, Multi-Stakeholder Reporting:** Leverage GenAI to generate multiple tailored deliverables from one analysis. Instead of a single generic report, consultants can provide **personalized outputs** for each stakeholder – e.g. an executive summary highlighting business impacts, a developer-focused issue list with code insights, and a technical deep-dive for security teams. This ensures that the right message reaches the right audience without extra manual effort.

- **AI-Assisted Code Analysis & Visualization:** Employ LLMs and graph technology to rapidly digest large codebases and architectures. GenAI can extract software design knowledge (components, data flows, dependencies) and represent it in diagrams or knowledge graphs. This capability yields up-to-date system models, attack surface mappings, and even visual threat maps, allowing consultants to **understand and document complex applications faster** than ever before.

- **Continuous and Collaborative Threat Modeling:** Evolve threat modeling from a one-off project deliverable into an ongoing, integrated practice. By combining automation and graphs, changes in the code or business (new features, infrastructure, or threat intel) can trigger automatic updates to threat models. The result is a continuous, collaborative risk management process where AppSec consultants help clients stay ahead of evolving threats in real time.

- **Upskilling and Training with GenAI:** Integrate these innovations into training services. AppSec teams can be taught to use GenAI-driven tools for self-service threat modeling, or to interpret semantic risk dashboards. Consultants can offer workshops where students **generate threat models in bulk, build threat taxonomies, and practice with AI-guided code reviews**, accelerating their learning. This not only adds value to clients but also amplifies the consultants' own productivity on engagements.

By embracing GenAI and semantic graphs, AppSec service companies can **scale their expertise, deliver more value with less effort, and differentiate themselves in a competitive market**. The following sections delve into the strategic rationale behind this approach and outline concrete service offerings ready for collaboration.

The Need for Evolution in AppSec Consulting

Traditional application security consulting relies heavily on expert effort – workshops, manual diagramming, and text-heavy reports – to communicate threats and mitigations. While effective, this approach struggles to keep up with today's fast-paced development and sprawling software architectures. Common pain points include:

- **Scalability Issues:** A human-centric threat modeling process doesn't scale well. Many consulting firms can only deliver a handful of threat models or training sessions at a time. For clients with dozens of applications or rapid release cycles, important systems may go un-modeled due to limited expert bandwidth.

- **Static Outputs:** The typical deliverable (documents, slide decks) represents a snapshot in time. As systems change, those artifacts quickly become outdated. Maintaining them is labor-intensive, leading to knowledge gaps over time.

- **Context Gaps:** Technical threat models often lack business context. Busy executives and product owners reading a security report may struggle to connect a listed threat to real business impact. This disconnect can reduce the report's influence on decision-making.

- **One-Size-Fits-All Reporting:** A single report is often expected to serve multiple audiences – developers, security teams, and management. In practice, that usually means it's too high-level for engineers or too granular for executives, diminishing its effectiveness.

- **Skills Bottleneck:** Threat modeling is a specialized skill. Training up new consultants or client staff is time-consuming, and scaling the knowledge across an organization is hard. Traditional training materials and workshops have limitations in engagement and personalization.

In short, **there is a pressing need to amplify the reach and relevance of AppSec consulting services**. GenAI and semantic graphs offer a timely solution: they act as force-multipliers for human expertise, handling repetitive scale tasks and enriching analysis with broader context. The next sections explore how these technologies can be applied in practice.

Generative AI: A Force Multiplier for Threat Modeling

GenAI – typified by LLMs like OpenAI GPT-4, Google Gemini, etc. – can turbocharge how consultants perform threat modeling. Rather than replacing human experts, it automates tedious tasks and

provides creative suggestions, allowing consultants to focus on high-level analysis and client interaction. Key applications include:

- **Mass-Generating Threat Models:** Given a description of a system (architecture diagrams, user stories, or even code), an LLM can produce an initial threat model draft in minutes. It can enumerate threat scenarios, possible attacker paths, STRIDE categories, and propose mitigations in a structured format. Recent experiments have shown this at extreme scale – for example, using Google's Gemini 2.0 model to automatically generate **1,000 threat models along with security design documents, attack surface analyses, and attack trees** [1]. This showcases how GenAI can brute-force the creation of security documentation that would otherwise take an army of consultants.

- **Customization and Consistency:** AI-generated content can be **tailored to specific tech stacks or domains**. By feeding the model with context (e.g. "web application with microservices handling healthcare data"), it can apply relevant threat libraries (like OWASP Top 10 or HIPAA security rules). Unlike purely human-driven approaches, an LLM will reliably enumerate standard threat categories for each component – ensuring consistency and coverage across models. Consultants can then review and adjust these AI drafts, achieving high-quality results more efficiently.

- **Open Schema Outputs:** Critically, the outputs from GenAI can be structured (e.g. in JSON or YAML) following an open threat modeling schema. This allows the results to be machine-readable and easily imported into tools or databases. For example, an AI could output a list of threats with fields like "ThreatID, Description, AffectedAsset, Mitigation, Severity". Such outputs can feed directly into a **knowledge graph or client's GRC system**, removing the friction of parsing PDF reports. Using open formats ensures the consulting deliverables integrate with the client's workflows and remain accessible, rather than trapped in slides.

- **Speeding Up Iteration:** Because GenAI can generate content so quickly, consultants can iterate threat models multiple times within an engagement. They might generate a baseline model, discuss it with the client to gather corrections or missing info, feed back those insights, and regenerate a refined model – all within a single day. This agility contrasts with the traditional approach of spending weeks in interviews and writing. **Ultimately, GenAI lets AppSec teams do more threat modeling in less time**, covering more of the client's attack surface with the same resources.

However, it's important to guide these AI systems with strong prompts and oversight. Without care, LLMs may produce irrelevant or boilerplate threats. Consultants should use their expertise to **craft effective prompts and validate outputs**, combining the creativity of AI with the contextual judgment of human experts. Over time, the firm can develop a library of proven "prompt templates" for different architectures (cloud serverless vs. mobile app vs. IoT, etc.), continually improving the quality of AI suggestions. This human+AI symbiosis forms the core of next-gen threat modeling services.

Semantic Knowledge Graphs: Adding a Living Context Layer

While GenAI accelerates content creation, **semantic knowledge graphs** ensure that all this information remains organized, queryable, and context-rich. A semantic graph is essentially a database of facts

represented as nodes and relationships, augmented with meaning (ontologies) that computers can reason about. Applying this to threat modeling yields huge benefits:

- **Unified View of Security Knowledge:** In a semantic graph, each entity – e.g. a software component, a threat scenario, a vulnerability, a mitigation, a business asset – is a node connected to others by meaningful relationships. For instance, a "SQL Injection" threat node can link to the "Web Portal Module" node it affects, which links to a "Customer Data" asset node, which links to a "GDPR Compliance" requirement node. By **storing threats and context as an interconnected graph, we create a living map of the risk landscape**. This goes beyond flat spreadsheets by capturing not just *what* the threats are, but *why they matter* and *how things relate*.

- **Multi-Framework Alignment:** Graphs make it easy to overlay multiple security frameworks and taxonomies simultaneously. An AppSec consultant can incorporate data from STRIDE, MITRE ATT&CK, OWASP Top 10, CWE, etc., all into the client's knowledge graph. The threats discovered by GenAI can be tagged with standard categories (e.g. STRIDE: Spoofing, Tampering, etc.), attacker techniques (from ATT&CK), compliance controls, and more. Because it's all in one graph, one can query, for example: *"Show me all discovered threats mapped to OWASP Top 10 categories, and list any categories with no findings yet"* – a great way to spot gaps. This kind of cross-reference is cumbersome with traditional documents but **becomes trivial with a graph database query**.

- **Linking to Business Goals:** Crucially, semantic graphs enable the blending of technical and business data. As noted, we can attach business context to threats – mapping each technical risk to the potential business impact. For example, a threat to "Payment Processing Service" links to the revenue stream it supports, or a threat impacting "Patient Data" links to regulatory penalties under healthcare law. By **mapping threats to goals, KPIs, org charts, and regulations in the graph, threat modeling is elevated from a technical silo to a business-critical capability**. Stakeholders can then clearly see *which* critical asset or objective is endangered by a given threat, lending weight to remediation efforts.

- **Continuous Updates and What-If Analysis:** A living knowledge graph can be updated in real-time as information changes. If the client adds a new microservice, the graph can be extended with its nodes, and GenAI can be invoked to analyze its threats. If a new vulnerability (e.g. zero-day in a library) appears, a query can reveal which systems in the graph use that library, flagging those threat models for review. In the future, such changes might even trigger automated LLM analysis – for example, on detecting a major business move (like a new product launch), the system could auto-generate an updated threat model reflecting the new scenario. This **continuous approach turns threat modeling into a cycle of constant risk awareness**, instead of a one-time engagement.

- **Graph-of-Graphs: Federating Knowledge:** As AppSec firms work with multiple clients, each with their own graph, there is an opportunity to connect graphs or maintain a meta-graph of generalized knowledge (carefully abstracted for confidentiality). This concept, referred to as "Graphs of Graphs of Graphs (G3)" in research, envisions an ecosystem where an application's threat graph links to an industry threat graph, which links to global attack technique graphs, and so on. In practical terms, this means **less reinventing the wheel** – if a threat is identified at one client, its pattern (node relationships) could be matched against another client's graph to see if a similar risk exists. Semantic standards make these kinds of knowledge transfers and comparisons feasible, amplifying the value of each threat model beyond its immediate context.

Implementing semantic knowledge graphs does introduce technical considerations. Firms will need a graph database (potentially cloud-based or even something lightweight like storing graphs in Jira as issues [2]). Ensuring **data provenance and determinism** is also key – each entry in the graph should trace back to its source (the code analysis, the consultant's input, a specific LLM query). This is vital for trust: both consultants and clients must be able to ask "why is this node/edge here?" and get a clear answer (e.g. *"this threat was added because component X handles sensitive data Y"*). As demonstrated in the MyFeeds.ai project for news, capturing a provenance trail can make an AI-driven system transparent and trustworthy [3]. The good news is that by having structured outputs and graphs, **every piece of information can carry metadata about its origin**, mitigating the "black box" concern of AI.

In summary, by adopting semantic graphs, AppSec consultancies turn their deliverables into a dynamic resource for clients. Instead of static PDFs, the output is a **living model of the client's security posture** that both parties can query, update, and learn from on an ongoing basis. This deepens the consultant's engagement (potentially leading to longer-term advisory roles) and provides continuous value to the client.

AI-Assisted Code Understanding and Visualization

A core part of threat modeling is understanding "what are we dealing with?" in the target system. GenAI and graphs can significantly reduce the time needed to gain this understanding and present it in insightful ways:

- **Automated Architecture Mapping:** Traditionally, consultants spend hours interviewing developers and reading docs to map out an architecture or data flow diagram. Now, **LLMs can take source code or configuration as input and describe the system architecture in natural language**. For instance, an AI given access to a code repository could produce a summary: "The system consists of a front-end React app calling a Node.js API, which in turn uses an AWS RDS database. The API has modules A, B, C… Module B handles authentication," and so on. This initial map can then be converted into a diagram automatically. There are already AI tools that turn code into UML or entity relationship diagrams. By integrating these, a consultant can quickly obtain *draft visuals* to validate with the client, instead of drawing from scratch.

- **Code-to-Graph Pipelines:** We can push this further by converting code relationships directly into a graph structure. Imagine each class, function, and data store in the code becomes a node in a graph, with relationships like "calls", "reads", "writes". This forms a **"source code knowledge graph"**. Using semantic enrichment, we can layer threat-relevant info onto it – e.g. tag which functions expose web APIs (attack surface), which data flows involve sensitive information, etc. Researchers and practitioners have begun exploring this space, showing that LLMs can assist in extracting such structured knowledge from code. For AppSec services, this means **rapid identification of high-risk areas** (the graph might show, for example, that a certain module has multiple external inputs and touches critical data – a likely hot-spot for threats).

- **Interactive Exploration:** Once code and threats are in a graph, consultants and clients can explore them with natural language queries or visual tools. For example, a security architect could query, "Show me all microservices without authentication threats" or "Which components would be affected if the user identity service is compromised?". The graph can answer these by tracing connections. This capability turns threat modeling into an interactive experience rather than a static report review. It also helps developers *see* the rationale behind security findings, aiding buy-in – they can trace from a threat node to the exact function in code it concerns.

- **Visualization and Reporting:** AI can also help tailor how information is presented. For a given threat model graph, a GenAI agent could create an **executive-friendly slide** highlighting key points (e.g. *"Out of 12 critical assets, 3 lack encryption at rest, potentially impacting ~$5M in revenue if breached"*), complete with a simple chart or graph. Conversely, it could generate a **developer checklist** or ticket descriptions for each issue (e.g. *"Implement input validation in Module X to mitigate SQL injection (High priority)"*). By automating these translations of the raw analysis to various formats, consultants ensure no stakeholder is left with "homework" to repackage the findings – it's already done.

- **Agentic Workflows:** Advanced "AI agents" can be configured to perform multi-step tasks on the code. For example, one could create a workflow where an agent identifies a risky code section, modifies it to a more secure pattern, and then explains the change. While still experimental, such capabilities could one day be part of consulting deliverables (offering not just advice but actual code fixes, with human review). Even today, tools like OpenAI's Codex can suggest code improvements for security. AppSec firms can position themselves to leverage these tools, offering **AI-augmented code review services** alongside threat modeling.

By using AI for deep code insight, AppSec consultants can tackle even unfamiliar or large codebases with confidence. The combination of **speed (AI scanning) and expertise (human validation)** means engagements start delivering value from day one. Clients often remark that the initial phase of a project – just figuring out the system – can take weeks; with AI assistance, that phase shortens dramatically, freeing time to focus on threat mitigation strategies and client discussions.

Personalized Multi-Stakeholder Deliverables

One of the most powerful yet simple ways GenAI can enhance AppSec services is through the **personalization of outputs**. A single security assessment can have many "views", and tailoring the communication to each stakeholder multiplies its impact:

- **Executive Summaries with Business Impact:** For senior executives and boards, the concern is business risk. Using the semantic graph, we can automatically generate an **Executive Summary** that frames threats in terms of business outcomes. For example, if certain threats endanger customer data or uptime, the summary would state the potential business loss (reputation damage, financial penalties, downtime costs) in clear terms. GenAI can be prompted to keep the language high-level and avoid technical jargon, resulting in a one-page brief a CEO or board member can quickly grasp. This kind of tailored summary makes it far more likely that the security findings will translate into budget and support for remediation.

- **Developer-Focused Reports and Tickets:** Developers need actionable guidance. From the same master threat model, an AI can produce a **developer report** that filters to just the relevant issues for each team or component. It can even generate JIRA tickets or user stories for each recommended fix (complete with acceptance criteria). For instance: *"Ticket: Implement parameterized queries in the Orders API. Description: The Orders API is vulnerable to SQL Injection. Use prepared statements or an ORM to eliminate direct string concatenation in SQL queries. Impact: Prevents attackers from executing arbitrary queries on the database* [1] *."* Providing the output in the tools and format developers use day-to-day (like task tracking systems) bridges the gap from analysis to action.

- **Security Team Deep-Dive:** The client's security personnel may want the full details and even to query the data themselves. For them, consultants can deliver the complete knowledge graph or

a detailed compendium of findings. This could include all threat instances, model assumptions, and even the raw GenAI prompts and outputs for audit purposes. By having this depth available, the client's security team can validate and extend the work as needed. It also demonstrates rigor, showing that behind each high-level recommendation there's a wealth of supporting detail (down to which lines of code were considered, etc.).

- **Training and Awareness Materials:** Another "view" of the analysis can be educational content. For example, if a threat model reveals that many developers aren't aware of certain secure coding practices, the AI can generate a short training blurb or FAQ: *"Q: What is SQL injection and how do we prevent it? A: ..."*. These can be compiled into an **internal knowledge wiki** or included in an appendix of the report. Over time, as patterns repeat across assessments, a consulting firm can build a library of these Q&As, improving the client's security culture. GenAI can even produce analogies or stories to convey concepts to non-technical staff, enhancing security awareness organization-wide.

- **Client-Specific Customization:** Personalization also means reflecting the client's industry and internal language. If the client refers to certain systems with nicknames, or has a particular risk rating scheme, the AI-generated outputs can be adjusted to use those terms. The white paper content can thus feel "written by us, for us" from the client's perspective. Small touches like inserting the company's name, or aligning to their policy frameworks, make the recommendations more relatable. GenAI is adept at mimicking styles and integrating given terminology – consultants just need to feed it the right hints.

By delivering multiple tailored artifacts, AppSec consultants essentially **multiply the touchpoints of their work within the client's organization**. A single engagement might yield an exec presentation, a technical report, a set of tickets, and some training snippets – each of which travels further than a one-size-fits-all report. Importantly, producing these does *not* mean writing four different documents from scratch; it's the same content, restructured and rephrased by AI for each audience. This approach was demonstrated at the recent Threat Modeling Conference, where an analysis was packaged into different stakeholder reports with minimal extra effort, impressing attendees with its versatility. The net effect is a higher ROI for the client (they get more value), and for the consulting firm it can mean **broader exposure within the client (more stakeholders seeing your work) and increased follow-on opportunities**.

Upskilling AppSec Teams with GenAI and Graphs

In addition to direct client services, AppSec companies can leverage these ideas to improve their **training and internal skill development offerings**. Both their own consultants and their clients' security/development teams stand to gain:

- **GenAI-Powered Threat Modeling Training:** Traditional threat modeling training has students manually identify threats on sample systems. We can flip this model by introducing GenAI into the classroom. Trainees can be taught how to craft prompts to generate threat models, then critically analyze and refine the AI's output. This "human + AI" exercise teaches what the AI is good at and where expert insight is needed. It's an engaging way to cover a lot of ground – for instance, each student could generate a threat model for a different scenario (web app, IoT device, ML system, etc.), resulting in a pool of examples the whole class learns from. The **mass generation capability** means students aren't limited to one case study; they can explore dozens of systems in a short time, guided by AI.

- **Interactive Workshops with Knowledge Graphs:** Instead of static slide presentations, imagine a training session where participants interact with a live threat model graph. The instructor might show a pre-built semantic graph for a fictitious company and ask, "What threats do you see if we add a new API here?" Students can visually see the connections and even query the graph for answers (with an AI helping interpret queries). This hands-on approach demystifies the concept of knowledge graphs. It trains people to think in terms of connections and systems, which is invaluable for threat modeling. Over time, graduates of such training will naturally incorporate graph thinking and maybe set up similar models in their own projects.

- **On-the-Job Aids:** For consulting teams internally, having a centralized AI assistant can vastly speed up preparation and research. For example, a junior consultant about to analyze a Kubernetes deployment could ask the team's AI knowledge base: "What are common threat scenarios for Kubernetes?" and get a quick primer with references. Dinis Cruz's work on *deterministic AI outputs with provenance* [4] is relevant here – by curating a library of Q&A and ensuring each answer has traceable sources, consultants can trust the AI's guidance and even share it with clients. Investing in such a system effectively **captures the collective knowledge of the firm** and makes it queryable 24/7.

- **Collaboration with Universities and Communities:** AppSec companies can also lead the charge in spreading these advanced practices by sponsoring workshops or hackathons. For instance, a "Threat Model Jam" where teams use an LLM to crank out threat models for well-known open-source projects, with prizes for the most comprehensive ones. This not only is good PR but also helps refine the techniques and pipelines in a public setting. It establishes the firm as a thought leader in AI-assisted security – attracting talent and clients who see that the company is ahead of the curve.

In essence, training offerings infused with GenAI and semantic graphs become more **engaging, scalable, and reflective of real-world augmented workflows**. They prepare the next generation of security professionals to work alongside AI and manage knowledge in graph forms. For the consulting firm, this is both a new revenue stream (training services) and a way to ensure their own staff continuously grow in proficiency with these cutting-edge tools.

New Service Offerings and Collaboration Proposals

By combining the above capabilities, AppSec service providers can craft **entirely new offerings** that differentiate them in the market. Below are concrete service packages that could be offered to clients (and potentially executed in partnership with GenAI/graph experts like the authors of this paper):

1. **AI-Augmented Threat Modeling Service** – A consulting engagement where the team uses GenAI to perform a comprehensive threat modeling exercise in a fraction of the usual time. The service deliverables include a semantic threat model graph of the target system, an executive risk briefing, and a developer remediation plan. The value proposition to clients is a deeper and faster analysis, with evidence of broad coverage (e.g. "we enumerated 5x more threat scenarios than a manual approach, covering not just known risks but creative 'what-ifs' courtesy of the AI"). This service could be sold on a per-application or per-release basis, encouraging clients to engage periodically for continuous updates.

2. **Knowledge Graph Integration & Dashboards** – Here, the consulting firm offers to build and maintain a **custom security knowledge graph** for the client's environment. Over a series of workshops and using automated data ingestion (from code, cloud config, etc.), the consultants

populate the graph with the organization's assets, threat models, controls, and relevant business metadata. They then provide a dashboard or portal for the client to visualize and query this graph at will (for example, a web UI showing the graph with filters for different frameworks). This essentially productizes the earlier concept of a living threat model repository. It's a high-touch engagement with recurring value, possibly delivered as a subscription or managed service. As part of this, the firm can also integrate external threat intelligence feeds or vulnerability scanners into the graph, making it the one-stop shop for contextual security knowledge.

3. **Multi-Stakeholder Reporting Bundle** – This is an offering focused on communication. After any security assessment or testing engagement (whether done by the firm or by the client's internal team), the consulting company uses its GenAI toolkit to generate the full spectrum of reports: exec summary, technical deep-dive, compliance impact report, developer tickets, etc. Think of it as a report augmentation service. Often, companies have raw results (from a pen test, or a security review) but struggle to *communicate* them upward or outward. Here the AppSec firm steps in to take the findings and, using AI, rapidly churn out the polished artifacts for each audience. This could be especially valuable for large enterprises where different departments (legal, engineering, C-suite) all need to understand a security issue in their own language.

4. **AI-Driven Secure Code Review** – Pairing code analysis LLMs with human expertise, this service targets the SDLC (Software Development Life Cycle) directly. Consultants will set up an AI to scan a significant codebase for potential flaws and generate a report of suspect areas (with reasoning). The human experts then validate and prioritize these findings, and deliver a combined output. The twist is that, alongside the usual review report, the client also gets the **code knowledge graph** produced during analysis, which they can use for future development reference. This service can find issues that static analysis tools might miss (like design-level problems) and do so faster than a purely manual review. It helps development teams tackle security *during development* with AI as an ever-watchful assistant.

5. **GenAI AppSec Training Programs** – As discussed, the firm can offer modern training to clients who want to skill up their developers or security champions. This could be a multi-day workshop or e-learning package titled *"Threat Modeling and Secure Coding with AI Assistance"*. Participants learn not only classic threat modeling but also how to leverage AI tools (some provided by the consulting firm) to automate parts of the process. Each trainee might receive access to a sandboxed AI system where they can practice generating threat models or fixing vulnerable code. The consulting firm thereby positions itself not just as advisors but enablers, transferring these advanced capabilities to the client's personnel. This often deepens client relationships and can lead to follow-on consulting when those trained teams start new initiatives and seek expert guidance.

6. **Strategic GenAI Security Partnership** – In some cases, an AppSec company might pursue an alliance or co-development effort with a technology provider (cloud platforms, dev tooling companies) to embed these ideas at a larger scale. For example, a partnership with a cloud provider to offer built-in threat modeling-as-a-service for their customers, powered by the consulting firm's expertise and AI workflows [5]. Or a collaboration with a graph database vendor (like Neo4j) to create a special AppSec knowledge graph solution. Such strategic moves can create new revenue streams and industry visibility, though they require commitment. This white paper itself is a form of outreach to initiate these conversations across the AppSec ecosystem.

Each of these offerings can be further refined and customized, but together they illustrate how GenAI and semantic graphs enable **concrete, marketable services**. They solve real client problems – from scaling analysis, to maintaining continuous visibility, to improving communications and training. Importantly, these services also create ongoing engagement opportunities (managed platforms, subscriptions, training follow-ups) rather than one-off projects, contributing to more stable and predictable business for the consulting company.

Implementation Roadmap

Adopting GenAI and semantic graph capabilities is a strategic journey. Based on our research and experimentation, we propose a high-level roadmap for AppSec firms ready to take this leap:

1. **Pilot Phase – "Quick Win" Project:** Start with a small-scale pilot on an internal project or a friendly client. For example, pick one application and attempt an AI-generated threat model and knowledge graph. Measure effort vs. traditional methods and gather feedback. This phase builds confidence and uncovers practical issues (prompt tuning, tool configuration) in a low-risk setting.

2. **Build the Toolkit:** Invest in assembling the right tools for GenAI and graph workflows. This might include obtaining API access to LLMs (OpenAI, Azure, Google, etc.), setting up a graph database (Neo4j, GraphDB, or even leveraging Jira as a graph store [2]), and scripting glue code to connect them. Open-source libraries and cloud services can accelerate this – for instance, using existing prompts from community projects or graph schemas from standards like the Open Threat Modeling schema.

3. **Team Training and Culture:** Ensure the consulting team is on board and trained. Run internal workshops similar to what we'd offer clients. Encourage consultants to use the AI assistant for day-to-day tasks (with proper guidelines) and to share successful techniques. Adjust performance metrics to value the *outcomes* (quality of threat coverage, client satisfaction) rather than hours spent – this will encourage adoption of efficiency tools without fear of "automating oneself out of a job." The culture should be that AI is an assistant, not a competitor, and using it effectively is a skill to be rewarded.

4. **Service Integration:** Gradually roll out the new capabilities as part of existing services. For instance, in the next threat modeling engagement, inform the client that "we will be using an AI-augmented approach which allows deeper analysis in the same timeframe." Use it to deliver extra findings or nicer reports as a bonus. As confidence grows, start packaging distinct offerings (like those listed above) and marketing them explicitly. Collect success stories – e.g. how much time was saved or how an AI-found issue prevented an incident – to build credibility.

5. **Feedback Loop and Improvement:** Set up a feedback loop where consultants report on AI suggestions that were wrong or graphs that were hard to query, etc. Use these to refine prompts, update the knowledge base, or adjust the graph schema. This continuous improvement will, over a few iterations, yield a very robust, proprietary capability that competitors (who are not doing the same) cannot easily replicate.

6. **Collaboration and Partnerships:** Finally, engage with the wider community. Partner with specialists (like Dinis Cruz's team, if we may humbly suggest) who have been pioneering these methods, to cross-pollinate ideas or even co-deliver projects initially. Sponsor or speak at industry events about your successes. Perhaps work with tool vendors to integrate your methodologies (for example, contribute to an open-source project or standard). By positioning

as a leader in AI-driven AppSec, the firm will attract clients that are forward-thinking and ready to invest in innovative security solutions.

Conclusion

The integration of Generative AI and semantic knowledge graphs represents a **paradigm shift for application security consulting**. It enables scalability, consistency, and context in threat modeling that were previously unattainable with purely manual methods. AppSec companies that embrace these techniques can deliver richer value to clients – uncovering more threats, connecting security to the business, and communicating insights in the language of each stakeholder. At the same time, they empower their own consultants to operate at a higher strategic level, supported by AI co-pilots handling repetitive analysis and documentation tasks.

This white paper has outlined both the high-level vision and the concrete steps to realize it, from specific service offerings to implementation milestones. The message is clear: **with GenAI and graphs, we can finally make AppSec work at the speed and scale of modern software development**. The authors – Dinis Cruz and ChatGPT Deep Research – invite forward-looking AppSec firms to collaborate on bringing this vision to life. Together, we can supercharge threat modeling and training services, turning them into continuous, context-aware, and business-aligned practices that redefine cybersecurity consulting for the years to come.

Co-authored by Dinis Cruz and ChatGPT Deep Research, 2025.

Sources:

- Dinis Cruz, *Advancing Threat Modeling with Semantic Knowledge Graphs*, 2025
- Dinis Cruz, *Linking Threat Models with Semantic Business Graphs*, 2025
- Marcin Niemiec, *Scaling Threat Modeling with AI: Generating 1000 Threat Models…*, Jan 2025 [1]
- Dinis Cruz, LinkedIn Post – *Jira as a GraphDB*, 2024 [2] [5]
- Dinis Cruz, *Graphs of Graphs (G3) in Threat Modeling*, 2025
- Dinis Cruz, *MyFeeds.ai – Provenance in AI-Powered News Feeds*, 2025 [3] [4]

---

[1] Scaling Threat Modeling with AI: Generating 1000 Threat Models Using... | Dinis Cruz
https://www.linkedin.com/posts/diniscruz_scaling-threat-modeling-with-ai-generating-activity-7281616297044393984-eZVs

[2] [5] Jira as Graph DB | Dinis Cruz | 10 comments
https://www.linkedin.com/posts/diniscruz_jira-as-graph-db-activity-7293099257547354112-z4xs

[3] [4] Establishing Provenance and Deterministic Behaviour in an LLM-Powered News Feed (first MyFeeds.ai MVP)
https://www.linkedin.com/pulse/establishing-provenance-deterministic-behaviour-llm-powered-cruz-dimhe