Turning AI Fear into Opportunity:

AI-Assisted Cybersecurity for Secure and Resilient Systems

Deepinder Sidhu
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
sidhu@umbc.edu

Abstract

Artificial Intelligence (AI) is increasingly capable of analyzing software, configurations, logs, policies, and other cybersecurity artifacts. Public discussion has focused largely on whether frontier AI systems can identify vulnerabilities and whether such capability could be misused. This paper reframes the issue as a defensive engineering opportunity. Organizations have long paid engineers, penetration testers, red teams, and ethical hackers to discover vulnerabilities before adversaries do; frontier AI changes how efficiently this mission can be performed, not why it matters. We introduce AI-assisted cybersecurity as a human-supervised engineering paradigm for identifying, explaining, prioritizing, validating, and eliminating weaknesses in existing systems. The paper discusses frontier AI reasoning capabilities, risks and governance requirements, critical infrastructure implications, and lifecycle integration from concept through operations. We argue that AI should not be viewed primarily as a source of cybersecurity fear, but as a continuously available defensive capability for building secure and resilient systems. The central question is no longer whether AI can identify vulnerabilities; it is whether we will use AI responsibly to eliminate them before adversaries exploit them.

Keywords— Artificial Intelligence, Cybersecurity, Frontier AI, Secure Systems, Critical Infrastructure, Vulnerability Management, Human-in-the-Loop, Post-Quantum Cryptography, Secure Engineering.

1 Introduction

"The only thing we have to fear is fear itself."—Franklin D. Roosevelt

Fear is often the first response to change. Bad news travels fast, especially when a transformative technology reveals uncomfortable weaknesses in systems that society already depends upon. Engineering progress, however, has never come from avoiding such discoveries; it comes from turning them into stronger designs.

Artificial Intelligence has rapidly evolved from an experimental research technology into an engineering capability capable of assisting professionals across numerous technical disciplines. Among its most significant applications is cybersecurity, where recent demonstrations have shown that frontier AI systems can analyze source code, identify software vulnerabilities, interpret malware artifacts, evaluate security configurations, and assist engineers in understanding complex cybersecurity problems [1–3]. These demonstrations have generated significant public discussion concerning the implications of AI for offensive cyber operations and have understandably heightened concerns regarding the potential misuse of increasingly capable AI systems.

The resulting public discourse, however, has largely focused on a single question: can AI identify vulnerabilities? From an engineering perspective, this is no longer the most important question. Organizations

have always sought to identify vulnerabilities before adversaries discover them. Software developers conduct peer reviews. Quality assurance engineers systematically test applications. Independent security researchers participate in responsible vulnerability disclosure programs. Enterprises employ penetration testers and red teams to simulate attacks against their own systems. Governments and commercial organizations invest in vulnerability assessments and bug bounty programs because discovering weaknesses before attackers do has long been recognized as a fundamental principle of defensive cybersecurity [4–7].

Frontier AI does not change this objective. Instead, it changes the efficiency, scale, and accessibility with which defensive cybersecurity analysis can be performed. Properly governed and operating under human supervision, AI enables organizations to continuously analyze software, network configurations, firewall policies, cloud infrastructures, access control policies, operational procedures, and numerous other cybersecurity artifacts that previously required substantial manual effort. AI not only identifies potential weaknesses but increasingly assists engineers by explaining security implications, recommending corrective actions, validating proposed improvements, and supporting continuous security engineering throughout the operational lifetime of complex systems.

The importance of this capability extends far beyond software development. Modern society depends upon highly interconnected digital infrastructures supporting electric power generation, telecommunications, transportation systems, healthcare delivery, financial services, water treatment, manufacturing, government operations, and national defense. These systems contain millions of lines of software, thousands of configuration files, continually evolving security policies, and complex operational environments. Despite decades of engineering improvements, vulnerabilities remain inevitable. The challenge is no longer whether vulnerabilities exist, but how rapidly responsible organizations can identify and eliminate them before malicious actors exploit them [8–10].

History demonstrates that transformative technologies frequently expose weaknesses in existing engineering practices. Quantum computing, for example, revealed fundamental limitations in widely deployed public-key cryptographic algorithms, motivating the development of Post-Quantum Cryptography (PQC) rather than discouraging quantum research [11–14]. Artificial Intelligence presents a similar opportunity. AI is revealing weaknesses that already exist within software, enterprise systems, and critical infrastructure. These discoveries should not be viewed primarily as reasons to fear AI; rather, they should be viewed as opportunities to strengthen the systems upon which society depends.

Accordingly, this paper proposes a shift in cybersecurity perspective. AI should become an integral component of cybersecurity engineering, assisting professionals throughout the lifecycle of software-intensive systems—from concept development and architecture through implementation, testing, deployment, operations, maintenance, and modernization. At each stage, AI addresses different cybersecurity optimization problems while remaining under human supervision and supporting engineering judgment rather than replacing it.

Central thesis. The question is no longer whether AI can identify vulnerabilities. The question is whether we will use AI responsibly to eliminate them before adversaries exploit them.

Every vulnerability identified through AI-assisted cybersecurity represents an opportunity to improve engineering quality, strengthen operational resilience, reduce organizational risk, and increase confidence in the software and infrastructure that underpin modern society.

2 From AI Fear to Engineering Opportunity

2.1 Frontier AI Systems: A New Cybersecurity Capability

Artificial Intelligence has undergone a remarkable transformation during the past several years. Earlier generations of AI systems primarily relied upon narrowly trained machine learning models designed to perform specific classification or prediction tasks. Modern frontier AI systems, by contrast, possess significantly broader reasoning and inferencing capabilities that enable them to analyze complex technical information, synthesize knowledge across multiple domains, explain their conclusions, and interact with users through natural language [1, 15, 16]. These capabilities represent a fundamental advancement beyond traditional cybersecurity tools.

It is important, however, not to treat all frontier AI systems as interchangeable. Different models vary in reasoning capability, context length, tool integration, hallucination behavior, explainability, deployment architecture, and security properties. The argument in this paper is therefore model-agnostic: it concerns the emerging class of frontier AI systems as engineering assistants, while recognizing that each operational deployment must be validated for the specific model, data boundary, toolchain, and mission context in which it is used.

Unlike conventional vulnerability scanners or rule-based expert systems, frontier AI systems can simultaneously evaluate multiple cybersecurity artifacts including source code, software documentation, firewall policies, router configurations, access control lists, security architectures, threat intelligence reports, vulnerability databases, operational logs, and configuration files. More importantly, these systems can reason about the relationships among these artifacts to produce engineering recommendations rather than simply reporting isolated findings.

Reasoning and inferencing capabilities distinguish frontier AI systems from previous generations of cybersecurity automation. Rather than searching for predefined signatures or applying static rule sets, AI systems can interpret programming logic, identify inconsistent security assumptions, recognize configuration conflicts, infer potential attack paths, evaluate alternative engineering solutions, and explain why a particular condition represents a cybersecurity concern. The ability to explain analytical conclusions improves human understanding and allows engineers to validate AI recommendations before implementing operational changes.

Another important characteristic of frontier AI systems is their ability to integrate diverse sources of information. A security engineer investigating a vulnerability may need to examine source code, software documentation, firewall rules, vulnerability databases, security policies, configuration files, threat intelligence, and operational logs before determining an appropriate corrective action. Frontier AI systems can rapidly correlate these heterogeneous information sources and present engineers with a coherent explanation of the underlying security problem together with recommended remediation strategies.

These capabilities do not eliminate the need for experienced cybersecurity professionals. Instead, they increase the productivity and effectiveness of those professionals. AI performs labor-intensive analytical tasks associated with reviewing large volumes of cybersecurity information, allowing engineers to concentrate on engineering judgment, operational tradeoffs, and risk management. Human expertise therefore remains central to the cybersecurity decision-making process while AI provides continuously available analytical support.

Figure 1 illustrates the role of frontier AI as a cybersecurity reasoning engine operating on diverse artifacts and producing candidate findings, explanations, and recommendations for human validation.

From a cybersecurity perspective, frontier AI should therefore be viewed not simply as another software tool but as a new engineering capability that changes how organizations analyze and improve the security of existing systems.

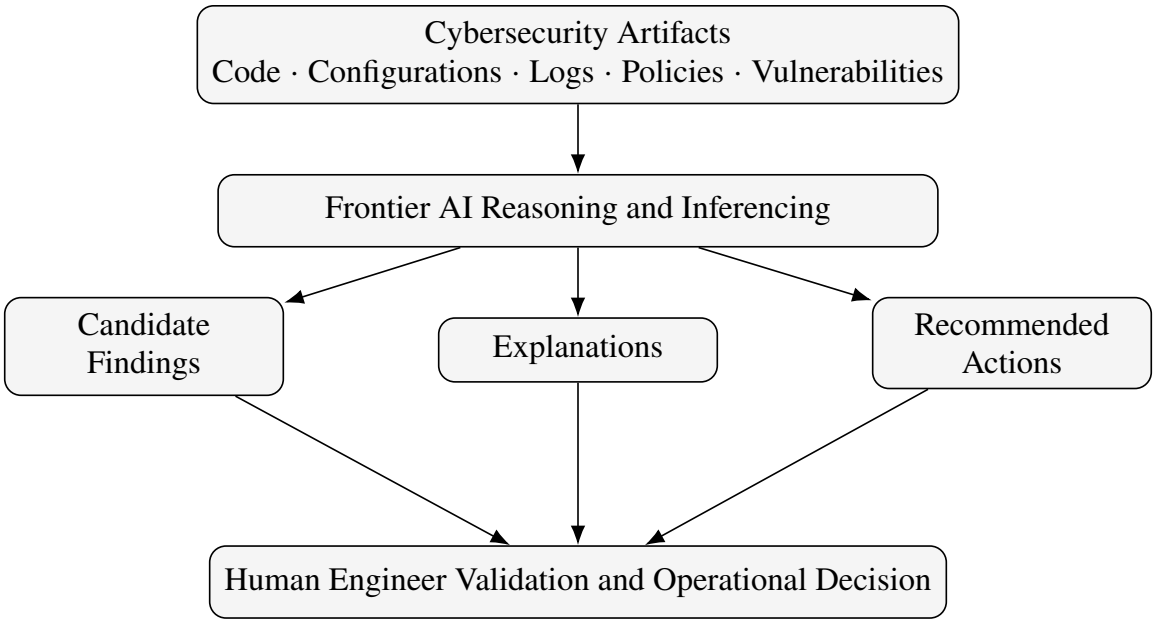

Figure 1: Frontier AI as a cybersecurity reasoning pipeline. AI analyzes cybersecurity artifacts, produces candidate findings and recommendations, and supports human validation rather than replacing human decision authority.

2.2 AI Fear Versus Engineering Opportunity

The emergence of frontier AI systems has understandably generated significant public concern. Recent demonstrations showing AI identifying software vulnerabilities, analyzing malware, interpreting source code, and assisting with cybersecurity tasks have raised questions regarding whether increasingly capable AI systems could also assist malicious actors. These concerns deserve careful consideration. Every transformative technology introduces new opportunities together with new risks, and Artificial Intelligence is no exception.

However, focusing exclusively on the offensive implications of AI presents only part of the picture. An equally important question has received far less attention: how can AI improve defensive cybersecurity? This distinction is important because AI does not create vulnerabilities within existing software or infrastructure. Those vulnerabilities already exist. AI simply provides a more effective means of discovering them. The engineering value of AI therefore lies not in exposing new weaknesses, but in enabling organizations to identify and eliminate existing weaknesses before adversaries exploit them.

The cybersecurity community has long accepted the principle that discovering vulnerabilities before attackers represents good engineering practice. Organizations routinely invest in secure software development, code reviews, quality assurance, penetration testing, red teams, vulnerability assessments, and bug bounty programs. Ethical hackers are compensated specifically to identify security weaknesses so they can be corrected before malicious actors discover them. This philosophy has become a cornerstone of modern cybersecurity engineering.

Frontier AI extends this same defensive philosophy. Rather than replacing ethical hackers or penetration testers, AI provides organizations with a continuously available analytical capability capable of reviewing software, evaluating configurations, interpreting vulnerabilities, and recommending corrective actions at a scale that would be impractical using manual methods alone. AI enables organizations to perform defensive cybersecurity continuously rather than periodically.

Consequently, the cybersecurity discussion should move beyond asking whether AI can identify vul-

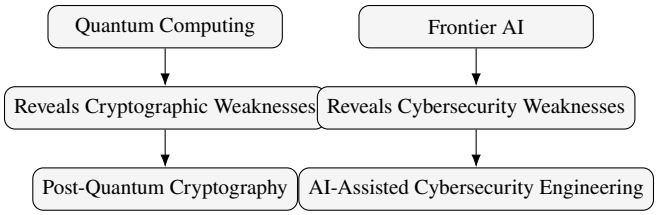

Figure 2: Transformative technologies reveal weaknesses and motivate better engineering. Quantum computing motivated PQC; frontier AI should motivate AI-assisted cybersecurity engineering.

nerabilities. The more important engineering question is whether organizations are prepared to use AI responsibly to improve the security and resilience of their systems.

2.3 Lessons from Transformative Technologies

History demonstrates that transformative technologies frequently reveal limitations in existing engineering practices. Rather than representing failures, these discoveries often become catalysts for significant technological advancement.

Quantum computing provides an instructive example. Shor's algorithm showed that a sufficiently capable quantum computer could efficiently solve mathematical problems underlying widely deployed public-key cryptographic systems, including RSA and elliptic-curve cryptography [11]. This discovery generated understandable concern regarding the future of secure communications. However, the engineering response was not to abandon quantum computing research. Instead, the cybersecurity community developed new cryptographic algorithms specifically designed to resist quantum attacks, culminating in NIST's standardization of Post-Quantum Cryptography algorithms [12–14].

The important lesson extends beyond cryptography. Transformative technologies frequently expose weaknesses that already exist but were previously difficult to recognize or address. Engineering advances occur when those discoveries are translated into improved designs, stronger architectures, and more resilient systems.

Artificial Intelligence presents a similar opportunity. AI is revealing weaknesses within software, network configurations, operational procedures, cloud infrastructures, and enterprise systems that may otherwise remain undiscovered for extended periods. These discoveries should be viewed as engineering opportunities rather than engineering failures. Every vulnerability identified by AI provides engineers with an opportunity to improve system security before adversaries exploit the same weakness.

This perspective suggests that AI-assisted cybersecurity represents the natural evolution of defensive engineering. Just as quantum computing motivated the development of Post-Quantum Cryptography, frontier AI has the potential to motivate a new generation of AI-assisted cybersecurity engineering practices focused on continuous security improvement rather than reactive vulnerability remediation.

2.4 Vulnerabilities in National Critical Infrastructure

Perhaps nowhere is the engineering opportunity presented by AI more significant than in the protection of national critical infrastructure. Modern society depends upon interconnected digital systems supporting electric power generation and distribution, water treatment facilities, telecommunications networks, transportation systems, healthcare delivery, financial services, emergency response, government operations, manufacturing, and national defense. These infrastructures increasingly rely upon complex software-intensive systems whose secure operation is essential to public safety, economic stability, and national security [8, 9].

Despite decades of engineering progress, these systems continue to contain software defects, configuration errors, inconsistent security policies, outdated software components, and operational vulnerabilities. Many infrastructures have evolved over decades, integrating legacy technologies with modern cloud services, industrial control systems, enterprise networks, and Internet-connected devices. The resulting complexity makes comprehensive manual cybersecurity analysis increasingly difficult.

Traditionally, organizations have relied upon periodic vulnerability assessments, penetration testing, compliance reviews, and manual engineering analysis to identify security weaknesses. While these practices remain essential, they are constrained by time, available expertise, and operational resources. Consequently, vulnerabilities may remain undiscovered long after deployment.

Frontier AI offers an opportunity to improve this engineering process. AI can continuously examine software, network configurations, firewall policies, routing protocols, access control rules, cloud infrastructures, and operational telemetry, identifying inconsistencies, explaining security implications, recommending corrective actions, and assisting engineers in validating proposed improvements before deployment into operational environments.

This capability raises an important engineering question:

Critical infrastructure question. If AI can help identify vulnerabilities within systems supporting national critical infrastructure before adversaries discover them, can responsible organizations justify choosing not to use that capability?

The objective is not autonomous cybersecurity. Human engineers remain responsible for evaluating AI recommendations, implementing corrective actions, and accepting operational risk. AI serves as a continuously available engineering assistant that enhances the effectiveness of cybersecurity professionals while maintaining human authority over operational decisions.

The opportunity is therefore not simply faster vulnerability discovery. The opportunity is continuous improvement of the security and resilience of the critical infrastructures upon which modern society depends.

Engineering takeaway. AI did not create the vulnerabilities; it created the opportunity to eliminate them before adversaries exploit them.

3 AI-Enabled Cybersecurity: Risks and Benefits

Artificial Intelligence represents one of the most significant advances in cybersecurity since the widespread adoption of automated vulnerability scanners and intrusion detection systems. However, like every transformative technology, AI introduces both opportunities and challenges. A balanced engineering assessment requires acknowledging the limitations of current frontier AI systems while recognizing their ability to improve defensive cybersecurity. The objective is not to determine whether AI is inherently beneficial or harmful, but rather to understand how it can be responsibly integrated into cybersecurity engineering to maximize its defensive value while minimizing operational risk.

3.1 Risks and Challenges

As frontier AI systems continue to evolve, several concerns must be addressed before widespread operational deployment.

Current AI systems may generate incorrect conclusions, overlook important contextual information, or recommend suboptimal engineering solutions. Although reasoning and inferencing capabilities have improved substantially, AI-generated recommendations should be viewed as engineering advice rather than

authoritative decisions. Human validation remains essential, particularly for mission-critical and safety-critical systems.

The protection of sensitive information also presents an important consideration. Organizations must ensure that proprietary source code, classified information, operational data, and critical infrastructure configurations are processed within appropriate security boundaries. Enterprise deployment of AI therefore requires careful consideration of data governance, access control, model security, and operational policies [3, 17].

Another concern involves adversarial manipulation of AI systems themselves. Prompt injection attacks, poisoned training data, malicious tool outputs, and supply chain compromises illustrate that AI systems, like all software, require appropriate security controls [17, 18]. AI-assisted cybersecurity must therefore include protections for the AI systems performing the analysis.

Finally, organizations must avoid excessive dependence upon automation. AI should augment engineering judgment rather than replace it. Cybersecurity remains an engineering discipline requiring technical expertise, operational understanding, and informed risk management. Human engineers remain responsible for approving recommendations, implementing corrective actions, and accepting operational risk.

These challenges are significant but manageable. Similar concerns accompanied earlier generations of automated security tools, cloud computing, virtualization, and software-defined networking. Engineering experience demonstrates that appropriate governance, validation, and operational oversight allow organizations to benefit from new technologies while managing associated risks.

3.2 AI-Assisted Cybersecurity

Engineering takeaway. AI did not create the vulnerabilities; it created the opportunity to identify and eliminate them before adversaries exploit them.

The benefits of frontier AI extend well beyond automated vulnerability discovery. Modern AI systems possess increasingly sophisticated reasoning capabilities that enable them to analyze a wide variety of cybersecurity artifacts and provide meaningful engineering assistance across multiple operational domains. Rather than replacing cybersecurity professionals, frontier AI augments human expertise by accelerating analysis, improving consistency, and providing engineering recommendations that remain subject to human review and approval.

Within software development, AI assists developers by reviewing source code, identifying insecure programming practices, explaining software defects, recommending secure implementations, and assisting with software maintenance. Rather than replacing software developers, AI functions as a continuously available engineering reviewer capable of examining large code bases rapidly and consistently.

Within enterprise networking, AI analyzes firewall policies, routing configurations, access control lists, network segmentation, and cloud security architectures to identify inconsistencies, conflicting rules, excessive privileges, and potential attack paths. Because enterprise configurations frequently evolve over many years, manually identifying subtle interactions among thousands of security rules can become exceptionally difficult. AI significantly improves the efficiency of this analysis.

Operational cybersecurity similarly benefits from AI-assisted reasoning. Security operation centers process enormous volumes of security alerts, event logs, vulnerability reports, and threat intelligence feeds each day. AI assists analysts by correlating related events, identifying likely root causes, prioritizing vulnerabilities according to operational risk, summarizing incident information, and recommending investigation strategies [9, 19]. These capabilities reduce analyst workload while improving consistency and response time.

Malware analysis represents another area where frontier AI has demonstrated value. AI can examine source code, scripts, macros, encoded payloads, and executable behavior to explain observed functionality,

identify suspicious patterns, distinguish benign artifacts from malicious software, and recommend appropriate defensive actions. Rather than replacing experienced malware analysts, AI accelerates routine analysis and allows experts to concentrate on the most complex investigations.

Throughout our own defensive experimentation using frontier AI systems, we observed that AI consistently demonstrated the ability to analyze software, identify programming errors, interpret malware-related artifacts, explain security implications, and recommend engineering improvements. Although the depth of reasoning varied among different frontier models, the overall pattern remained consistent. Different frontier models exhibit varying reasoning capabilities, context windows, tool integration, and explainability characteristics. Consequently, this paper advocates a model-agnostic engineering approach in which AI serves as an analytical assistant rather than relying on the capabilities of any particular implementation.

Importantly, these observations should not be interpreted as evidence that AI has solved cybersecurity. Instead, they demonstrate that AI has become a practical engineering capability capable of augmenting human cybersecurity expertise across a broad range of defensive activities.

3.3 AI as a Continuously Available Ethical Hacker

For decades, organizations have invested substantial resources in discovering vulnerabilities before adversaries exploit them. Secure software development practices, independent code reviews, penetration testing, vulnerability assessments, bug bounty programs, and ethical hacking all share a common objective: identify weaknesses early so they can be corrected before deployment or before malicious actors discover them [4, 5].

Artificial Intelligence represents the next evolution of this defensive philosophy. Rather than replacing ethical hackers, AI provides organizations with a continuously available analytical capability capable of performing many of the same defensive functions throughout the operational lifetime of software-intensive systems. AI reviews software, evaluates network configurations, analyzes security policies, interprets vulnerability reports, examines malware artifacts, recommends corrective actions, and assists engineers in validating proposed changes. Unlike traditional point-in-time assessments, AI enables these analyses to occur continuously as systems evolve.

The analogy to ethical hacking is useful because it emphasizes the defensive nature of AI-assisted cybersecurity. The objective is not to increase offensive capability. The objective is to provide defenders with better analytical tools than those available to potential adversaries. Organizations have long accepted the principle of paying trusted professionals to identify vulnerabilities before attackers do. Frontier AI extends this same principle by making advanced cybersecurity analysis continuously available, highly scalable, and economically accessible to organizations of all sizes.

Defensive objective. Find vulnerabilities first. Fix them first. Build better systems.

This shift in perspective represents the central message of this paper. AI should not be viewed primarily as a technology that creates cybersecurity risk because it can identify vulnerabilities. Instead, AI should be recognized as one of the most powerful defensive cybersecurity capabilities developed to date, enabling organizations to continuously improve the security and resilience of existing software, enterprise systems, and national critical infrastructure before adversaries exploit the same weaknesses.

3.4 Operational Governance and Evaluation

AI-assisted cybersecurity must be governed as an operational engineering capability, not treated as an informal advisory tool. Governance requires model validation, audit trails, reproducibility of findings, explainability of recommendations, secure deployment, access control, policy compliance, and clear assignment of human accountability. Security-sensitive deployments should record the artifact analyzed, the

Table 1: Three simple examples of human mistakes that AI-assisted cybersecurity can help identify and correct. The examples are illustrative and defensive; operational changes require human validation.

Artifact	Representative Human Mistake	AI-Assisted Finding	Engineering Action

C source code	`while (attempts < MAX);`
	`scanf("%s", buf);`
	`if (buf == password)`
		Empty loop caused by stray semicolon; unbounded input; pointer comparison instead of string comparison.	Replace with bounded input, explicit error handling, and `strcmp`; add regression tests for failed login and lockout behavior.

Firewall policy	Earlier broad allow rule precedes a later deny rule for a sensitive host or subnet.	Later deny rule is shadowed under first-match semantics; intended security boundary is not enforced.	Reorder or narrow rules, remove redundancy, document intended policy, and validate representative traffic flows before deployment.
Router configuration	ACL, routing, and management-plane settings are modified independently.	ACL permits unintended management access or routing change creates a path that bypasses segmentation.	Validate reachability, management-plane exposure, logging, and consistency with firewall policy before committing the change.

model or toolchain used, the prompt or task specification, the candidate findings produced, the human review decision, and the engineering action taken. This creates the auditability needed for regulated environments, critical infrastructure, and government acquisition programs, and aligns AI-assisted cybersecurity with secure-by-design and secure software development practices [20, 21].

The distinction between human-in-the-loop and human-on-the-loop operation is important. In human-in-the-loop use, AI recommendations do not produce operational changes until a qualified engineer explicitly approves the action. In human-on-the-loop use, AI may monitor, triage, or execute pre-approved low-risk workflows while humans supervise and retain the ability to intervene. For cybersecurity engineering, human-in-the-loop control is appropriate for code changes, firewall policy modifications, router configuration updates, access-control changes, and mission-impacting operational decisions. Human-on-the-loop approaches may be appropriate for summarization, alert grouping, enrichment, documentation, and other low-risk analytical workflows.

Effectiveness should be measured using engineering metrics rather than broad claims about intelligence. Useful metrics include vulnerability detection rate, false-positive rate, false-negative rate, review throughput, time to triage, time to remediation, configuration coverage, number of artifacts analyzed, human acceptance rate of AI recommendations, reproducibility of findings, and measured improvement in security posture after remediation. These metrics allow organizations to evaluate whether AI-assisted cybersecurity improves defensive operations without assuming that AI outputs are automatically correct.

3.5 Illustrative Examples of AI-Assisted Cybersecurity

The following examples illustrate the role of AI as a defensive engineering assistant. They are not presented as benchmark results. Rather, they show how AI-assisted cybersecurity can support different classes of engineering artifacts while preserving human validation. The examples are intentionally simple: if a few lines of code or configuration can contain consequential mistakes, the challenge becomes far more severe in operational systems containing millions of lines of code, thousands of devices, and many interacting security policies.

C code review. A few lines of C code can contain multiple defects with security consequences. In a defensive code-review task, AI can detect a stray semicolon after a loop condition that creates an unintended infinite loop, an unbounded `scanf` that may overflow a fixed-size buffer, and a pointer comparison that incorrectly checks password strings. Individually, these mistakes are easy to understand once identified. In practice, however, such errors may be embedded in larger authentication, logging, recovery, or access-control logic where their operational effect is less obvious. AI contributes by identifying the defect, explaining why

it matters, recommending safer code, and suggesting tests that a human engineer can validate.

Firewall policy analysis. Firewall rules are often order-dependent. A single broad allow rule placed before a narrower deny rule may unintentionally shadow the intended protection for a sensitive host or subnet. AI can reason over first-match semantics, detect shadowed or redundant rules, identify overly broad access, and explain which traffic classes are affected. The engineering value lies in converting a confusing policy interaction into a reviewable recommendation: narrow the broad rule, reorder the policy, or add explicit documentation and validation tests before deployment.

Router configuration review. Router and network-device changes can affect both reachability and security boundaries. A small access-control or routing change may expose management services, create asymmetric paths, bypass segmentation, or conflict with firewall policy. AI can assist by comparing the proposed configuration against intended security posture, identifying affected interfaces or prefixes, and recommending validation checks. The engineer remains responsible for determining whether the operational change is acceptable.

These examples illustrate the core workflow of AI-assisted cybersecurity: analyze the artifact, identify candidate weaknesses, explain the security implication, recommend corrective action, and preserve human engineering authority.

Engineering takeaway. Finding a local error is only the beginning; understanding its effect on the complete operational system is the cybersecurity challenge.

4 AI Enables Secure and Resilient Systems Development

Artificial Intelligence has the potential to fundamentally change how secure systems are engineered. Historically, cybersecurity has often been treated as a specialized activity performed after software development or infrastructure deployment. Security reviews, vulnerability assessments, penetration testing, compliance evaluations, and operational monitoring frequently occur after engineering decisions have already been made. While these activities remain essential, they are inherently reactive. Vulnerabilities introduced during earlier lifecycle phases may remain undetected until systems are operational, increasing remediation costs and organizational risk.

Frontier AI provides an opportunity to shift cybersecurity from a predominantly reactive discipline toward continuous engineering support throughout the lifecycle of software-intensive systems. Rather than functioning solely as a post-deployment security tool, AI becomes an engineering capability integrated from concept development through operational sustainment. At each stage of the lifecycle, AI addresses different optimization questions while maintaining the same overall objective: improving the security, resilience, reliability, and operational effectiveness of the resulting system.

4.1 AI Throughout the Engineering Lifecycle

The engineering questions asked during system development evolve as the system matures. Consequently, AI should not perform a single cybersecurity function but rather provide lifecycle-specific analytical support.

During concept development and requirements analysis, AI assists engineers by identifying security requirements, evaluating mission objectives, analyzing alternative concepts, detecting conflicting requirements, and recommending security considerations before system architectures are established. Early identification of security issues reduces downstream engineering costs and improves overall system quality.

During system architecture and design, AI evaluates alternative architectures, analyzes trust boundaries, identifies potential attack surfaces, reviews security assumptions, recommends security controls, and assists

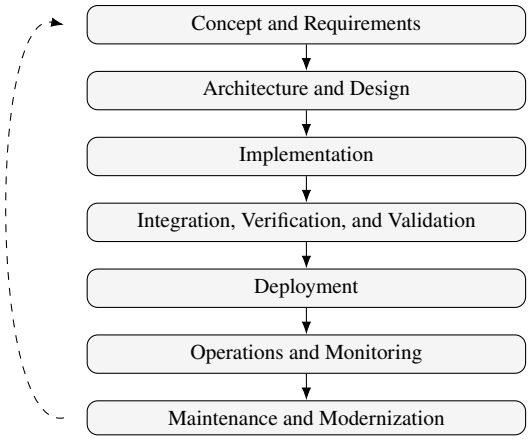

Figure 3: AI-assisted cybersecurity across the lifecycle. AI supports different cybersecurity optimization questions at each phase while preserving human engineering authority.

architects in balancing security, performance, cost, maintainability, and operational effectiveness. Security therefore becomes an integral component of architecture rather than an afterthought.

During implementation, AI reviews source code, identifies programming errors, detects insecure coding practices, explains software vulnerabilities, recommends secure implementations, and assists developers in improving software quality. AI becomes a continuously available peer reviewer capable of examining software throughout the development process.

During integration, verification, and validation, AI assists engineers by evaluating test coverage, reviewing configuration files, analyzing firewall policies, validating access control rules, interpreting test results, identifying inconsistent security assumptions, and recommending additional validation scenarios. Security validation therefore becomes more comprehensive while reducing manual engineering effort.

Following deployment, AI continues to provide operational value by analyzing logs, monitoring security events, correlating threat intelligence, evaluating configuration changes, identifying emerging vulnerabilities, recommending corrective actions, and assisting security operations centers in maintaining continuous situational awareness.

Finally, during maintenance and modernization, AI assists engineers in evaluating software updates, infrastructure modifications, cloud migrations, security policy revisions, technology refresh decisions, and system modernization activities. Security improvements therefore become continuous rather than periodic.

At every lifecycle phase, AI addresses different optimization questions. Early phases emphasize architecture, requirements, and design optimization. Middle phases emphasize software quality and validation. Operational phases emphasize resilience, monitoring, incident response, and continuous improvement. This lifecycle perspective represents one of the most significant opportunities presented by frontier AI.

4.2 AI and the Complexity of Modern Interconnected Systems

4.3 Why Software Vulnerabilities Persist

Despite decades of advances in programming languages, software engineering, testing, verification, and cybersecurity, software vulnerabilities continue to appear in operational systems. The reasons extend well beyond simple programming mistakes.

First, modern systems must satisfy increasingly complex and often competing functional, performance, safety, security, privacy, resilience, regulatory, and mission requirements. These requirements interact in subtle ways, making it difficult to reason about the complete behavior of the system throughout its lifecycle.

Second, software-intensive systems have grown beyond unaided human comprehension. Millions of lines of code, distributed services, cloud platforms, operational technologies, dynamic configurations, and thousands of interacting components create an enormous state space that engineers cannot exhaustively analyze manually.

Third, existing engineering tools have important limitations. Static analysis, testing, formal verification, model checking, symbolic execution, and theorem proving each provide valuable assurance, but none offers a complete solution for large, continuously evolving operational systems. Testing can demonstrate the presence of defects but rarely their absence. Formal methods often require abstractions that become difficult to maintain as systems evolve, while static analysis tools may generate false positives or fail to capture dynamic operational behavior.

Finally, software is never static. Every software update, configuration change, cloud deployment, library upgrade, routing modification, or security policy revision introduces new interactions that may create unintended consequences elsewhere in the system. Maintaining assurance therefore becomes a continuous engineering challenge rather than a one-time verification activity.

Frontier AI should be viewed within this broader engineering context. AI does not replace established engineering methods; rather, it complements them by helping engineers manage complexity, reason across heterogeneous artifacts, identify hidden dependencies, prioritize engineering effort, and determine where deeper verification and validation should be applied.

Engineering takeaway. The challenge is no longer simply writing correct software; it is understanding and assuring the behavior of increasingly complex, continuously evolving systems. AI represents the next major step in extending human engineering capability to meet that challenge.

AI is maturing at a time when the complexity of modern software-intensive systems increasingly exceeds what engineering teams can comprehend manually. Contemporary systems are no longer isolated programs. They are distributed compositions of software services, cloud platforms, identity systems, databases, routers, firewalls, endpoint agents, operational technology, sensors, real-time data streams, and external dependencies. A change intended to correct a local vulnerability can alter timing behavior, break interoperability, weaken segmentation, change trust relationships, or create unintended operational consequences elsewhere in the system.

Lines of code provide only a rough proxy for complexity, but they illustrate the scale of modern engineering environments. The Space Shuttle primary flight software contained approximately 400,000 lines of code, a large and carefully controlled software system for its era [22]. The F-35 has required more than 8 million lines of software code across flight controls, radar, communications, electronic attack, sensor fusion, weapons deployment, and logistics [23,24]. Porsche Engineering notes that a modern car may contain roughly 100 million lines of code, while citing 14 million lines for the Boeing 787 as a comparison point [25]. The U.S. Army Future Combat Systems program provides an even more sobering example: GAO reported that the projected software effort had grown to nearly 95 million lines of code [26]. These figures are not directly comparable because they count different kinds of software in different acquisition and operational contexts, but they make the same point: modern mission and infrastructure systems are software-intensive at a scale that challenges conventional review practices.

The cybersecurity implication is direct. As the number of components, interfaces, dependencies, and policy interactions increases, the probability that a software or configuration change will produce unintended consequences also increases. A patch may close one vulnerability while opening a new operational path. A firewall update may block a known exposure while disrupting mission traffic. A router configuration change may improve reachability while weakening segmentation. A seemingly local code change in an integrated defense system may affect data fusion, communication timing, operator displays, or downstream decision-support workflows.

A modern missile defense capability illustrates this problem at an operational level. Such a system may integrate distributed sensors, radar systems, command-and-control software, communications networks,

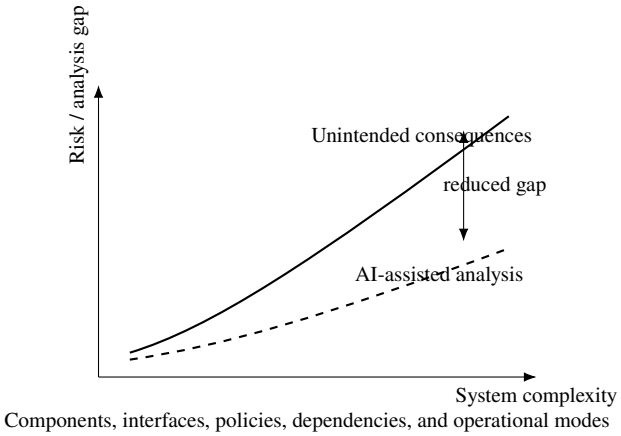

Figure 4: Notional relationship between system complexity and the risk of unintended consequences after software or configuration changes. The upper curve represents increasing analysis difficulty as systems become more interconnected. The lower dashed curve illustrates the intended role of AI-assisted cybersecurity: reducing, not eliminating, the engineering analysis gap through dependency analysis, change-impact assessment, and human-supervised validation.

satellite links, cyber defense services, operator interfaces, real-time data fusion, and time-constrained decision support. A modification to one subsystem may have effects that are difficult to anticipate without understanding dependencies across the broader architecture. In such environments, cybersecurity is inseparable from systems engineering: the question is not only whether a component is vulnerable, but whether the fix changes the behavior of the larger mission system.

Frontier AI can aid engineers by reasoning across software dependencies, configuration relationships, security policies, operational telemetry, and historical behavior. Before deployment, AI can identify affected components, estimate downstream impact, recommend additional regression tests, detect policy conflicts, and flag areas requiring human engineering review. The value of AI is therefore not merely that it identifies vulnerabilities; it helps engineers understand and manage the growing complexity of modern interconnected systems.

Table 2: Illustrative software scale in complex systems. Counts are approximate and should be interpreted as order-of-magnitude indicators rather than directly comparable measurements.

System	**Reported Scale**	**Source Context**
Space Shuttle primary flight software	~400,000 LOC	National Academies assessment
F-35 aircraft software	>8 million LOC	GAO / USAF software sustainment reports
Boeing 787	~14 million LOC	Porsche Engineering comparison
Modern automobile	~100 million LOC	Porsche Engineering discussion
Future Combat Systems	~95 million LOC	GAO defense acquisition report

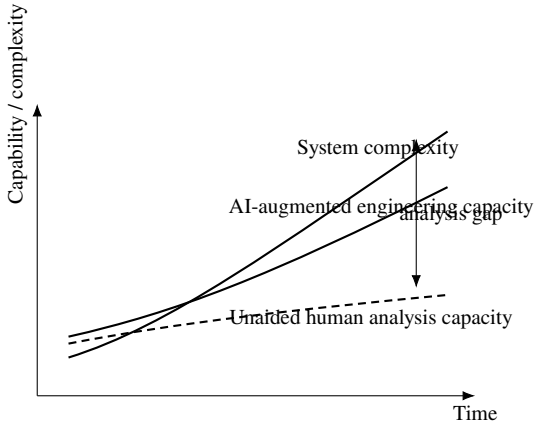

Figure 5: Notional relationship between increasing system complexity, unaided human analysis capacity, and AI-augmented engineering capacity. AI matured at a time when modern interconnected systems increasingly require computational assistance to reason about dependencies, configurations, and system-wide consequences of change.

4.4 Local Correctness and System-Level Assurance

The examples above also illustrate a deeper assurance problem. A few lines of code may be corrected locally, but the security and operational properties of the complete system may still remain uncertain. In small programs, engineers can often inspect the relevant control flow, reason about the affected state, and determine whether the fix preserves intended behavior. In large distributed systems, however, a local correction can interact with authentication services, message queues, caches, routing policies, monitoring agents, timing assumptions, retry logic, or failover behavior. The property of interest is no longer merely whether one function is correct, but whether the system as a whole continues to satisfy desired properties.

Formal methods, static analysis, model checking, theorem proving, and runtime verification provide powerful techniques for reasoning about system properties such as safety and liveness [27–29]. However, these techniques often require abstractions, models, invariants, and environmental assumptions that are difficult to maintain for continuously evolving operational systems with millions of lines of code and complex configuration state. Tools remain essential, but they do not automatically scale to prove all relevant properties of the system as deployed.

AI-assisted cybersecurity should therefore be viewed as complementary to, not a replacement for, formal assurance methods. Frontier AI can help engineers identify candidate invariants, map dependencies, summarize relevant code paths, compare intended policy with deployed configuration, generate test hypotheses, and flag situations where a proposed change may threaten safety, liveness, segmentation, availability, or mission assurance. In some cases, AI may help construct evidence supporting a limited assertion; in other cases, it may help disprove an assertion by identifying a counterexample, overlooked dependency, or conflicting configuration.

The appropriate claim is modest but important: AI does not magically prove arbitrary properties of complex systems. It can, however, help engineers determine where assurance arguments are weak, where additional tests are needed, where formal tools should be applied, and where a local fix may have system-level consequences.

Engineering takeaway. Finding a bug is a local problem; determining whether safety, liveness, and security still hold after the fix is a system-level assurance problem.

4.5 Continuous Engineering Improvement

Engineering takeaway. As software-intensive systems become increasingly complex, AI has matured at the right time to help engineers continuously improve system security while reducing unintended consequences.

One of the most significant advantages of AI-assisted cybersecurity is its ability to support continuous engineering improvement. Traditional cybersecurity practices often follow a repetitive cycle of deployment, vulnerability discovery, patch development, and corrective maintenance. While effective, this approach frequently allows vulnerabilities to remain operational until discovered through periodic assessments or real-world incidents.

AI enables a fundamentally different engineering model. Instead of periodically searching for vulnerabilities, organizations can continuously analyze evolving software, infrastructure, and operational environments. AI reviews proposed software changes before implementation, evaluates infrastructure modifications before deployment, analyzes the security implications of architectural decisions, validates firewall policies before activation, reviews router configurations before deployment, and assists engineers in identifying unintended consequences before operational impact occurs.

Rather than reacting to vulnerabilities after deployment, organizations increasingly have the opportunity to prevent vulnerabilities from entering operational systems in the first place. This shift represents an important change in cybersecurity philosophy. AI becomes an engineering capability focused on preventing defects rather than simply detecting them. The engineering objective therefore changes from reactive remediation toward proactive improvement.

The need for continuous engineering improvement is driven not only by the persistence of cybersecurity vulnerabilities, but also by the rapidly increasing complexity of modern software-intensive systems. Contemporary enterprise systems, cloud infrastructures, telecommunications networks, industrial control systems, and national critical infrastructure routinely consist of millions of lines of software executing across thousands of interconnected hardware and software components. A seemingly minor modification to a software module, firewall rule, router configuration, cloud policy, or access-control rule may unintentionally affect system behavior in ways that are difficult for engineers to predict through manual analysis alone.

Historically, engineering teams have relied upon peer reviews, testing, simulation, formal verification, and operational experience to reduce the likelihood of unintended consequences. These techniques remain indispensable, but their effectiveness is increasingly challenged by the scale, heterogeneity, and dynamic behavior of modern distributed systems. As system complexity increases, the number of interactions among software components, network services, security policies, and operational environments grows even more rapidly, making exhaustive human reasoning increasingly impractical.

Frontier AI provides an opportunity to significantly expand engineering analytical capacity. AI can examine proposed software modifications, firewall policy updates, router configurations, cloud deployments, and infrastructure changes before operational deployment, identifying potential inconsistencies, security implications, policy violations, and unintended interactions that warrant further engineering review. Rather than replacing established engineering practices, AI complements them by enabling continuous engineering analysis at a scale that would otherwise be impractical.

This capability is particularly valuable when evaluating system-level engineering properties. Individual software modules may be locally correct while the integrated system still exhibits undesirable emergent behavior. Demonstrating global properties such as safety, liveness, resilience, mission assurance, security policy compliance, and operational correctness remains one of the most challenging problems in systems engineering. Existing formal verification and validation techniques remain essential but often do not scale to the size and complexity of modern operational systems. Frontier AI cannot formally prove these properties in the general case, but it can assist engineers by analyzing system dependencies, identifying potential

violations, evaluating the impact of proposed modifications, and directing engineering attention toward portions of the system that merit deeper formal analysis and testing. AI therefore complements rather than replaces established verification, validation, testing, and formal methods.

A lightweight engineering model captures this idea. Let the security condition of a system be represented by

$$C_{\text{sec}} = f(S, N, P, O, H), \tag{1}$$

where S denotes software state, N denotes network and infrastructure state, P denotes policy and access-control state, O denotes operational telemetry and mission context, and H denotes human oversight and engineering judgment. AI-assisted cybersecurity acts on cybersecurity artifacts $\mathcal{X}$ and produces findings and recommendations:

$$(\mathcal{F}, \mathcal{R}) = \mathcal{A}(\mathcal{X}; S, N, P, O), \tag{2}$$

where $\mathcal{F}$ are candidate weaknesses and $\mathcal{R}$ are candidate engineering improvements. Human engineers validate these recommendations and determine whether the proposed change improves the security condition:

$$\Delta C_{\text{sec}} = C_{\text{sec}_{\text{after}}} - C_{\text{sec}_{\text{before}}}. \tag{3}$$

The objective is not autonomous optimization by AI, but human-supervised improvement in which AI accelerates discovery, explanation, recommendation, validation, and continuous engineering improvement while humans retain operational authority and responsibility for all operational decisions.

These equations are conceptual engineering models intended to illustrate AI-assisted cybersecurity workflows and continuous security-posture improvement. They are not presented as formal mathematical derivations.

4.6 Secure and Resilient Systems

Engineering takeaway. Secure and resilient systems are not achieved by eliminating every defect; they are achieved by continuously understanding, managing, and reducing system risk throughout the engineering lifecycle.

Cybersecurity extends beyond preventing unauthorized access. Modern systems must remain available, trustworthy, recoverable, and operational despite hardware failures, software defects, configuration errors, natural disasters, insider threats, and sophisticated cyber attacks. Security and resilience therefore represent complementary engineering objectives.

AI contributes to resilience by continuously evaluating the overall health of software-intensive systems. AI identifies architectural weaknesses, recommends improved segmentation strategies, evaluates redundancy mechanisms, analyzes configuration consistency, reviews operational procedures, identifies single points of failure, and assists engineers in understanding how local vulnerabilities may influence broader system behavior.

This capability becomes particularly important within large enterprise environments and national critical infrastructure where millions of software components, thousands of network devices, cloud services, operational technologies, and security policies interact continuously. Human engineers cannot realistically evaluate every possible interaction manually. Frontier AI significantly expands analytical capacity while allowing human experts to concentrate on engineering judgment and operational decision-making.

Security and resilience are ultimately system-level properties rather than properties of individual software modules or network devices. A software component may satisfy its local requirements while interactions

among independently correct components still produce unexpected behavior at the system level. Consequently, improving the security of complex systems requires understanding not only individual vulnerabilities but also their relationships, dependencies, and potential operational consequences. AI assists engineers by identifying these relationships, highlighting potential cascading effects, and recommending areas requiring deeper engineering investigation.

The effectiveness of AI-assisted cybersecurity should therefore be evaluated using engineering metrics rather than anecdotal examples alone. Representative measures include vulnerability discovery rate, configuration coverage, analysis throughput, false-positive and false-negative rates, time-to-remediation, reduction in engineering review effort, and the percentage of AI recommendations accepted after human review. Such metrics provide organizations with objective measures for evaluating the contribution of AI to secure systems engineering.

Importantly, AI should not replace engineers. Instead, AI becomes an engineering partner that augments human expertise. Engineers remain responsible for system architecture, operational risk acceptance, security policy, mission decisions, and final operational authority. AI provides analysis, explanation, recommendations, and validation; humans provide engineering judgment, accountability, and responsibility.

Ultimately, AI-assisted cybersecurity represents more than another security technology. It represents an opportunity to fundamentally improve how secure and resilient systems are engineered throughout their entire lifecycle.

5 Summary and Conclusions

Artificial Intelligence has rapidly become one of the most transformative technologies affecting cybersecurity. Public discussion has understandably focused on the ability of frontier AI systems to identify software vulnerabilities, analyze malware, interpret security configurations, and perform increasingly sophisticated cybersecurity tasks. While these capabilities deserve careful consideration, the broader engineering opportunity has received comparatively little attention.

Organizations have spent decades investing in defensive cybersecurity. Software quality assurance, independent code reviews, penetration testing, vulnerability assessments, red teams, bug bounty programs, and ethical hackers all pursue the same engineering objective: identify vulnerabilities before adversaries exploit them. Frontier AI fundamentally changes how this objective can be achieved by providing organizations with a continuously available analytical capability capable of examining software, network configurations, operational policies, cloud infrastructures, and cybersecurity artifacts at unprecedented speed and scale.

The engineering significance of AI extends well beyond vulnerability discovery. Frontier AI increasingly assists engineers by explaining vulnerabilities, identifying relationships among complex cybersecurity artifacts, recommending corrective actions, validating proposed improvements, and supporting continuous engineering throughout the lifecycle of software-intensive systems. AI therefore becomes an engineering capability that augments rather than replaces human expertise.

Modern software-intensive systems have reached a level of scale and complexity that increasingly exceeds unaided human analytical capacity. Enterprise systems, cloud infrastructures, industrial control systems, telecommunications networks, and national critical infrastructure contain millions of lines of software executing across thousands of interconnected components. Small engineering changes may produce system-wide consequences that are difficult to predict through manual analysis alone. Frontier AI has matured at precisely the time when engineers require new analytical capabilities to manage this complexity.

This capability is particularly important for protecting national critical infrastructure. Electric power systems, telecommunications networks, transportation systems, healthcare delivery, financial institutions, water treatment facilities, government operations, and defense systems increasingly depend upon complex digital infrastructures that cannot realistically be analyzed manually in their entirety. AI provides an

opportunity to continuously improve the security and resilience of these systems before adversaries exploit existing weaknesses.

The historical lesson is instructive. Quantum computing revealed limitations in classical public-key cryptography, motivating the development of Post-Quantum Cryptography rather than discouraging scientific progress. Artificial Intelligence presents a similar engineering opportunity. AI reveals vulnerabilities that already exist within software, enterprise systems, and critical infrastructure. These discoveries should motivate better engineering rather than greater fear.

Core message. The question is no longer whether AI can identify vulnerabilities. The question is whether we will use AI responsibly to eliminate them before adversaries exploit them.

Call to Action

The cybersecurity community should embrace AI as a defensive engineering capability and act decisively to integrate it into cybersecurity practice. Specifically, we recommend the actions in Table 3.

Table 3: Call to action for AI-assisted cybersecurity.

Recommended Actions
Integrate AI throughout the systems engineering lifecycle, from concept development through operations and modernization.
Use AI-assisted cybersecurity to continuously analyze software, network configurations, cloud infrastructures, and security policies.
Treat every AI-discovered vulnerability as an engineering opportunity to improve security and resilience.
Incorporate AI-assisted cybersecurity into government acquisition, independent verification and validation, and lifecycle sustainment activities.
Leverage AI to strengthen protection of national critical infrastructure through continuous defensive analysis.
Maintain meaningful human oversight through appropriate human-in-the-loop or human-on-the-loop operational models while allowing AI to augment engineering analysis and decision support.
Develop objective engineering metrics to continuously evaluate the effectiveness of AI-assisted cybersecurity throughout the system lifecycle.
Combine AI-assisted cybersecurity with Post-Quantum Cryptography, secure software engineering, and resilient systems engineering.

The future of cybersecurity should not be defined by fear of Artificial Intelligence. It should be defined by how effectively we use AI to engineer software, networks, enterprise systems, and critical infrastructure that are secure, resilient, and worthy of society's trust.

Organizations have spent decades paying software engineers, quality assurance teams, penetration testers, red teams, security consultants, and ethical hackers to identify vulnerabilities before adversaries exploit them. Frontier AI does not change that objective—it fundamentally changes how efficiently it can be achieved. AI did not create the vulnerabilities that exist in today's software, networks, and critical infrastructure; it created an unprecedented opportunity to identify and eliminate them.

The question before the engineering community is therefore no longer whether AI should participate in cybersecurity. The question is whether we can justify not using AI to help secure the software-intensive systems upon which modern society depends.

Future Work. Future work will investigate AI observability, runtime assurance, multi-agent cybersecurity workflows, AI-assisted secure software development, integration with formal verification techniques, and large-scale validation within operational enterprise and critical infrastructure environments.

References

[1] OpenAI, "GPT-4 technical report," OpenAI, Tech. Rep., 2023, arXiv:2303.08774.

[2] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon *et al.*, "Constitutional AI: Harmlessness from AI feedback," *arXiv preprint arXiv:2212.08073*, 2022.

[3] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST, Tech. Rep. NIST AI 100-1, 2023.

[4] *ISO/IEC 29147: Information Technology – Security Techniques – Vulnerability Disclosure*, International Organization for Standardization Std., 2018.

[5] *ISO/IEC 30111: Information Technology – Security Techniques – Vulnerability Handling Processes*, International Organization for Standardization Std., 2019.

[6] OWASP Foundation, "OWASP Top 10: The Ten Most Critical Web Application Security Risks," https://owasp.org/www-project-top-ten/, 2021.

[7] MITRE, "Common Weakness Enumeration (CWE)," https://cwe.mitre.org/, 2024.

[8] Cybersecurity and Infrastructure Security Agency, "Critical Infrastructure Sectors," https://www.cisa.gov/topics/critical-infrastructure-security-and-resilience/critical-infrastructure-sectors, 2024.

[9] National Institute of Standards and Technology, "The NIST Cybersecurity Framework (CSF) 2.0," NIST, Tech. Rep. NIST Cybersecurity White Paper 29, 2024.

[10] *IEC 62443: Industrial Communication Networks – Network and System Security*, International Electrotechnical Commission Std., 2018.

[11] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE, 1994, pp. 124–134.

[12] National Institute of Standards and Technology, "FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard," NIST, Tech. Rep., 2024.

[13] ——, "FIPS 204: Module-Lattice-Based Digital Signature Standard," NIST, Tech. Rep., 2024.

[14] ——, "FIPS 205: Stateless Hash-Based Digital Signature Standard," NIST, Tech. Rep., 2024.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[16] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[17] OWASP Foundation, "OWASP Top 10 for Large Language Model Applications," https://owasp.org/www-project-top-10-for-large-language-model-applications/, 2025.

[18] MITRE, "MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems," https://atlas.mitre.org/, 2024.

[19] ——, "MITRE ATT&CK Framework," https://attack.mitre.org/, 2024.

[20] Cybersecurity and Infrastructure Security Agency, "Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software," https://www.cisa.gov/resources-tools/resources/secure-by-design, 2023.

[21] National Institute of Standards and Technology, "Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities," https://doi.org/10.6028/NIST.SP.800-218, NIST, Tech. Rep. SP 800-218, 2022.

[22] National Research Council, *An Assessment of Space Shuttle Flight Software Development Processes*. National Academies Press, 1993.

[23] U.S. Government Accountability Office, "F-35 Aircraft: DOD and the Military Services Need to Reassess the Future Sustainment Strategy," https://www.gao.gov/assets/gao-23-105341.pdf, GAO, Tech. Rep. GAO-23-105341, 2023.

[24] 388th Fighter Wing Public Affairs, "Hill, Robins to Provide F-35 Software Sustainment," https://www.388fw.acc.af.mil/News/Article-Display/Article/1280043/hill-robins-to-provide-f-35-software-sustainment/, 2017.

[25] Porsche Engineering, "When Software Writes Software," https://newsroom.porsche.com/en_US/2021/technology/porsche-engineering-when-software-writes-software-25367.html, 2021.

[26] U.S. Government Accountability Office, "Defense Acquisitions: Significant Challenges Ahead in Developing and Demonstrating Future Combat System's Network and Software," https://www.gao.gov/assets/gao-08-409.pdf, GAO, Tech. Rep. GAO-08-409, 2008.

[27] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. MIT Press, 1999.

[28] L. Lamport, *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley, 2002.

[29] G. J. Holzmann, *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley, 2004.