The Alan Turing Institute

Centre for Emerging Technology and Security

BRIEFING PAPER

Generative AI in Cybersecurity

Assessing impact on current and future malicious software

Sarah Mercer and Tim Watson

June 2024

About CETaS

The Centre for Emerging Technology and Security (CETaS) is a research centre based at The Alan Turing Institute, the UK's national institute for data science and artificial intelligence. The Centre's mission is to inform UK security policy through evidence-based, interdisciplinary research on emerging technology issues. Connect with CETaS at cetas.turing.ac.uk.

This research was supported by The Alan Turing Institute's Defence and National Security Grand Challenge. All views expressed in this report are those of the authors, and do not necessarily represent the views of The Alan Turing Institute or any other organisation.

Acknowledgements

The authors are very grateful to colleagues Ardi Janjeva, Dr Daniel Martin and Dr Katie Bishop for their support and valuable feedback during the drafting of this paper.

Cite this work as: Sarah Mercer and Tim Watson, "Generative AI in Cybersecurity: Assessing impact on current and future malicious software," *CETaS Briefing Papers* (June 2024).

Executive Summary

This CETaS Briefing Paper assesses the potential of generative artificial intelligence (GenAI) to create malicious software. These findings should inform governments' risk management posture towards the AI-cybersecurity nexus and provide an evidence base for the growing AI evaluation community focusing on malicious code and AI-enabled cyberattacks.

This topic is sparking varied reactions within the cybersecurity community. Opinions are split between those who fear GenAI could lead to sophisticated novel threats, and those who argue that it merely automates the assembly of *existing* malicious code found on the Internet. This debate highlights the **growing complexity of GenAI's role in cybersecurity, specifically its ability to create malicious binaries**. Drawing on academic research and grey literature, this paper seeks to offer a nuanced exploration of both the capabilities and limitations of GenAI in this domain.

Since GPT-4's public release in March 2023, there has not been a noticeable uptick in novel malware detections in the wild. This observation suggests that while GenAI may be a powerful tool, it currently lacks the specific capabilities and training necessary to independently create operational malware.

The main challenges of crafting effective malware are twofold. Firstly, there is the critical issue of reliance on vulnerabilities, which GenAI cannot autonomously identify or exploit due to **limitations in reasoning ability and training data**. Secondly, building effective malware requires a delicate balance between stealth, security, and functionality, **implying a level of strategic decision-making beyond the current capabilities of GenAI**.

Although the near-term impact of AI-generated code is limited, **GenAI does have the potential to profoundly disrupt the cybersecurity landscape over a longer time horizon**, exacerbating existing risks with respect to the speed and scale of reconnaissance, social engineering, and spear-phishing. As machine learning models become more sophisticated and training datasets more comprehensive, GenAI's role in cyber threats and cybersecurity is likely to grow significantly.

Some experts believe that future advancements in AI might lead to scenarios where malware created by AI can only be effectively countered by other AI-based defence systems. This reflects a broader concern of a **'cyber-AI arms race', where capabilities of offensive and defensive technologies continually evolve to outpace each other**.

If cyber defence is to stay ahead of the game, GenAI systems must be applied astutely. Current systems offer unique strengths, particularly in pattern recognition and natural language processing. **Targeted application of these abilities to enhance state-of-the-art cybersecurity systems is critical.**

However, realising this potential requires a collaborative effort between communities which are often at odds. **Bridging the divides between the AI and cybersecurity communities will enhance our understanding and innovative application of GenAI capabilities**, strengthening our cyber defences against evolving threats.

Introduction

Malware is a catch-all term used to describe various types of software designed for malicious purposes such as viruses, worms, trojans, ransomware, scareware, spyware, and adware. It is deployed for multiple reasons including stealing identities and financial details or gaining control of computers to launch denial-of-service attacks, mine cryptocurrencies or spread disinformation.

For this paper, 'AI-generated code' refers to code generated by large language models (LLMs), trained on vast datasets of publicly available source code. These systems assist software developers by generating code in response to text prompts (code snippets or full functions), refactoring, repairing and refining code, and by being able to explain code. The first successful model of this kind was OpenAI's Codex,[1] which powers GitHub's Copilot. As of February 2024, GitHub Copilot had 1.3 million paid subscribers. Embedded within the development environment, Copilot serves as an autocomplete tool that reportedly enhances developer productivity, enabling 55% faster coding.[2] However, it is essential to recognise that AI-generated code can contain flaws and should be meticulously reviewed, edited, and refined by developers.[3]

GenAI excels at creation, communication, and problem solving, but it does not inherently distinguish between beneficial and malicious uses. Although many of the main commercial LLM systems have safety protections, such as content filtering, they are not 100% reliable and can often be defeated by clever prompting, or indeed, circumvented altogether with the use of some open-source models. Fundamentally, they are designed to ingest information found on the Internet, and generate plausible responses using statistics, often in the guise of an 'AI Chatbot' persona. Current GenAI systems struggle with attention, deterministic-ness, reasoning, and contextual understanding. People mistakenly assume these systems can reason and understand context, but such capabilities cannot be fully realised with current transformer architectures. [4]

[1] "OpenAI Codex," OpenAI, 10 August 2021, https://openai.com/blog/openai-codex.

[2] "The world's most widely adopted AI developer tool," GitHub.com, https://github.com/features/copilot; Sida Peng et al., "The Impact of AI on Developer Productivity: Evidence from GitHub Copilot," *arXiv* (February 2023), https://arxiv.org/pdf/2302.06590.

[3] "AI code-generation software: What it is and how it works," IBM, 19 September 2023, https://www.ibm.com/blog/ai-code-generation/.

[4] Melanie Mitchell, "Can Large Language Models Reason?" AIGuide (Substack), 10 September 2023, https://aiguide.substack.com/p/can-large-language-models-reason; Yann LeCun (@ylecun), "Auto-Regressive LLMs can't

Recent CETaS research concluded that GenAI is an "amplifier of pre-existing cybersecurity risks. By reducing the degree of specialist knowledge required, generative AI can assist the less technically able user in experimenting with novel cyberattack techniques and increase their sophistication iteratively to result in capable attacks."[5]

GenAI has proven itself to be a particularly useful tool for social engineering attacks, due to LLMs' ability to process natural language, meaning they can both help select suitable targets for spear phishing deployments and curate 'personalised' messages.[6] A recent report from the National Cyber Security Centre (NCSC) on the near-term impact of AI on cyber threats emphasised that AI will uplift the social engineering/spear phishing ability of all threat-actors: state-sponsored (highly skilled), organised criminals (skilled but resource constrained), and opportunistic criminals (novice hackers).[7]

However, the central question of this Briefing Paper is whether GenAI can autonomously create novel malware that exploits previously unknown vulnerabilities, and which is able to evade state-of-the-art defences.

Since GPT-4's public release in March 2023, there has not been a noticeable uptick in novel malware detections in the wild.[8] This observation suggests that while GenAI may be a powerful tool, it currently lacks the specific capabilities and training necessary to independently create operational malware. This sentiment was echoed by NCSC, which suggested the use of AI in cyberattacks presents an evolution of risk, not a revolution.[9]

Importantly, not all cyberattacks require malware, and malware-free attacks accounted for 75% of detected intrusions last year.[10] Modern antivirus, platform security and response process improvement have meant that for serious criminals, malware-free attacks are more lucrative and popular. This trend is partly related to the success of identity attacks (phishing, social engineering, and access brokers).

plan (and can't really reason) ...," X, 13 September 2023, https://x.com/ylecun/status/1702027572077326505; Hannah Murphy and Christina Criddle, "Meta AI chief says large language models will not reach human intelligence," *Financial Times*, 22 May 2024, https://www.ft.com/content/23fab126-f1d3-4add-a457-207a25730ad9.

[5] Ardi Janjeva et al., "The Rapid Rise of Generative AI: Assessing risks to safety and security," *CETaS Research Reports* (December 2023), page 28.

[6] Julian Hazell, "Spear Phishing with Large Language Models," *arXiv* (December 2023), https://doi.org/10.48550/arXiv.2305.06972.

[7] National Cyber Security Centre, *The near-term impact of AI on cyber threat* (NCSC: 2024), https://www.ncsc.gov.uk/report/impact-of-ai-on-cyber-threat.

[8] "Global Threat Report 2024," Crowdstrike, https://go.crowdstrike.com/global-threat-report-2024.html.

[9] "Global ransomware threat expected to rise with AI, NCSC Warns," National Cyber Security Centre, 24 January 2024, https://www.ncsc.gov.uk/news/global-ransomware-threat-expected-to-rise-with-ai.

[10] "Global Threat Report 2024," Crowdstrike, https://go.crowdstrike.com/global-threat-report-2024.html.

But there is a possibility that GenAI will shift the balance back towards malware-based attacks. The following sections assess GenAI's ability to automate malware development, identification and exploitation of software vulnerabilities, and the likely future longer-term trajectory of this technology.

1. Automating Malware Development with Generative AI

The first research question this Briefing Paper sets out to answer is: **what specific coding tasks within malware development can currently be automated by GenAI?**

Answering this question requires first assessing key malware characteristics:

1. **Delivery and installation:** executed via the user, by exploiting known vulnerabilities, or by installing alongside legitimate software/updates.
2. **Sandbox escape/privilege escalation:** gaining higher-level permissions, likely through exploiting vulnerabilities or configuration errors.
3. **Persistence and stealth:** designed to avoid detection by signature-based and/or heuristic antivirus/intrusion detection systems.
4. **Payload:** the malicious functionality; could be to delete, encrypt or steal data, to install a backdoor, or hijack system resources.
5. **Command and control (C&C):** manages data egress, receives commands and possible additional functionality to install.
6. **Anti-analysis features:** includes debugger avoidance and code obfuscation.

For each of these pieces of functionality, the LLM training dataset likely contains a set of examples acquired from many publicly available sources, including public disclosures, academic publications, hacker forums, security blogs, cybersecurity training, red teaming resources, bug bounties and vulnerability reports. This means the LLM can generate code that achieves the described functionality; but it does not yet follow that it can autonomously create malware.

Current LLMs are not capable of autonomously writing high-quality code; they typically require human intervention to correct and refine what they generate.[11] This is primarily

[11] Drew Harry, "LLM-enabled Developer Experience (as of April 2024)," LinkedIn, 8 April 2024, https://www.linkedin.com/pulse/llm-enabled-developer-experience-april-2024-drew-harry-h0k4c; Burak Yetistiren et al., "Evaluating the Code Quality of AI-Assisted Code Generation Tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT," *arXiv* (October 2023), https://arxiv.org/pdf/2304.10778; Burak Yetistiren, Isik Ozsoy and Eray Tuzun, "Assessing the Quality of GitHub Copilot's Code Generation," in *PROMISE 2022: Proceedings of the 18th International Conference on Predictive Models and Data Analytics in Software Engineering* (Singapore: Association for Computing Machinery, 2022), 62–71, https://dl.acm.org/doi/pdf/10.1145/3558489.3559072; David Ramel, "New GitHub Copilot Research Finds 'Downward Pressure on Code Quality," Visual Studio Magazine, 25 January 2024, https://visualstudiomagazine.com/articles/2024/01/25/copilot-research.aspx; Brandon Vigliarolo, "What if AI produces code not just quickly but also, dunno, securely, DARPA wonders," *The Register*, 2 April 2024, https://www.theregister.com/2024/04/02/ai_dominates_at_darpa_and/.

because LLMs lack an understanding of the logical structures and contextual nuances needed for sophisticated software development.[12]

Additionally, LLMs are generally trained on more commonplace and less sophisticated examples, as comprehensive and high-quality datasets of malicious code are rarely publicly available. This is because of the ethical and legal issues of putting code into AI models that could do harm in the wrong hands, and because of the logistical issues concerning sharing dangerous malware across a research community.

There are systems that have been fine-tuned on malicious content, such as WormGPT, FraudGPT and DarkBERT.[13] Many of these are not constrained by content-filtering or safety requirements as they are specialised versions of open-source models. The inclusion of dark web content will increase the breadth of the training data set, but it will still be constrained, as the most sophisticated, operationally reliable, and stealthy techniques will still be inaccessible due to their market value (an exploit can sell for up to $7million).[14] Access to the most sophisticated malware is limited to state actors, platform owners, and security product vendors.[15]

Today's advanced models do not possess the capabilities needed to autonomously create sophisticated malware and they continue to rely on human expertise to check their outputs.[16] Sophisticated malware often requires a delicate balance between stealth, security, and functionality. It involves making strategic decisions that weigh the effectiveness of the malware against its detectability. Such decisions require a level of tactical foresight and adaptive problem-solving that current LLMs do not possess.[17]

[12] Rajarshi Haldar and Julia Hockenmaier, "Analyzing the Performance of Large Language Models on Code Summarization," *arXiv* (April 2024), https://arxiv.org/pdf/2404.08018; Ainave, "Can Devin AI really replace Software Engineers?," LinkedIn, March 16, 2024, https://www.linkedin.com/pulse/can-devin-ai-really-replace-software-engineers-ainavehq-xvxqe; Veronica Chierzi, "A Closer Look at ChatGPT's Role in Automated Malware Creation," Trendmicro.com, 14 November 2023, https://www.trendmicro.com/en_us/research/23/k/a-closer-look-at-chatgpt-s-role-in-automated-malware-creation.html.

[13] Youngjin Jin et al., "DarkBERT: A Language Model for the Dark Side of the Internet," May 2023, https://arxiv.org/pdf/2305.08596; Alp Cihangir Alsan, "Meet DarkBERT: Unraveling the Secrets of the Shadows," osintteam.com (Medium), 10 August 2023, https://osintteam.blog/meet-darkbert-unraveling-the-secrets-of-the-shadows-26167e28a655.

[14] Lorenzo Franceschi-Bicchierai, "Price of zero-day exploits rises as companies harden products against hackers," 8 April 2024, *Yahoo Finance*, https://uk.finance.yahoo.com/news/price-zero-day-exploits-rises-150051973.html.

[15] National Cyber Security Centre, *The near-term impact of AI on cyber threat* (NCSC: 2024), https://www.ncsc.gov.uk/report/impact-of-ai-on-cyber-threat (Point 6, Assessment).

[16] National Cyber Security Centre, *The near-term impact of AI on cyber threat* (NCSC: 2024), https://www.ncsc.gov.uk/report/impact-of-ai-on-cyber-threat.

[17] Melanie Mitchell, "Can Large Language Models Reason?," AI Guide (Substack), September 10, 2023, https://aiguide.substack.com/p/can-large-language-models-reason; Jie Huang et al., "Large Language Models Cannot Self-Correct Reasoning Yet," *arXiv* (March 2024), https://arxiv.org/pdf/2310.01798.

Nonetheless, while GenAI may currently lack the capabilities necessary to autonomously create sophisticated malware due to limitations in training data and reasoning abilities, its utility for malicious purposes should not be underestimated.

LLMs can act as both a coding teacher and assistant, effectively lowering the barrier to entry for writing malicious software.[18] However, the quality of LLM outputs hinges on the expertise of the person prompting the model, who uses knowledge and experience to frame prompts that maximise the quality of the response.[19]

There are several reports of hackers misusing ChatGPT for tasks such as 'improving' existing info-stealing malware,[20] and learning how to write ransomware scripts (although small fixes were required).[21] A recent Threat Intelligence/Security blog by Microsoft sheds light on the activities of known threat actors' use of LLMs for malicious purposes.[22] All five actors described in the report used LLMs for 'LLM-enhanced scripting techniques', as categorised by MITRE. These techniques involved using LLMs to generate or refine scripts that could be used in cyberattacks. This ranged from basic scripting tasks like programmatically identifying certain user events on a system, and help with fixing coding errors, through to seeking assistance with troubleshooting and understanding various web technologies, and refining scripts to support automation or streamlining of cyber tasks.

In summary, while current GenAI capabilities fall short of autonomously creating sophisticated malware due to limitations in training data and reasoning abilities, they still serve as valuable tools for malicious actors. Through specialised models and exploitation of LLMs, hackers can leverage GenAI to assist with some aspects of malware development. Real-world observations highlight the bi-directional relationship between LLMs and their users. Whether GenAI adopts the 'directing' or 'assisting' role depends on the skill level of the user and complexity of the task. Human expertise remains crucial for the development of sophisticated malware.

[18] National Cyber Security Centre, *The near-term impact of AI on cyber threat* (NCSC: 2024), https://www.ncsc.gov.uk/report/impact-of-ai-on-cyber-threat.

[19] Lauren Laws, "Can ChatGPT write malware?," Information Trust Institute, University of Illinois, 2 May 2023, https://iti.illinois.edu/news/chatgpt-malware.

[20] "Cybercriminals Bypass ChatGPT Restrictions to Generate Malicious Content," Check Point Blog, 7 February 2023, https://blog.checkpoint.com/2023/02/07/cybercriminals-bypass-chatgpt-restrictions-to-generate-malicious-content/.

[21] Alexis Zacharakos, "How hackers can abuse ChatGPT to create malware," TechTarget News, 22 February 2023, https://www.techtarget.com/searchsecurity/news/365531559/How-hackers-can-abuse-ChatGPT-to-create-malware.

[22] "Staying ahead of threat actors in the age of AI," Microsoft Threat Intelligence, 14 February 2023, https://www.microsoft.com/en-us/security/blog/2024/02/14/staying-ahead-of-threat-actors-in-the-age-of-ai/.

2. Automating Identification and Exploitation of Software Vulnerabilities Using Generative AI

To achieve their goals, malicious binaries[23] need ways to defeat security and privacy-enhancing subsystems. To do this they often exploit vulnerabilities within the host operating system or applications. These vulnerabilities arise from a range of issues that can include human error, such as coding mistakes or configuration oversights, but also stem from the inherent complexity of software development and the tight release schedules developers face.

The value of exploitable vulnerabilities is high, both to attackers and the platform or application owners whose users are being attacked. As such, automated vulnerability discovery has been a top priority for the cybersecurity community for many years, and there is much research dedicated to this topic. Traditional machine learning (particularly deep learning) techniques have shown a strong ability to detect vulnerable functions with a higher degree of accuracy,[24] and reduce both false positives and false negatives when compared with baseline static analysers.[25]

The use of LLMs has been explored for various purposes: classifying insecure functions,[26] improving the results returned by traditional static analysers,[27] and finding and fixing vulnerabilities. In the latter cases, this is achieved by utilising certain 'oddities' of LLM behaviour, such as the performance improvement gained through prompting techniques such as self-reflection[28] and chain of thought.[29]

A common approach to measuring the performance of LLMs when used to detect code defects is to compare their results to those of traditional static analysers, such as CppCheck

[23] Binaries refer to executable programs composed of machine code that computers can directly execute. In cybersecurity, malicious binaries are executable files that contain harmful code designed to perform unauthorised actions on the target system.

[24] Yaqin Zhou et al., "Devign: Effective Vulnerability Identification by Learning Comprehensive Program Semantics via Graph Neural Networks," *arXiv* (September 2019), https://arxiv.org/pdf/1909.03496.

[25] Yangruibo Ding et al., "VELVET: a novel Ensemble Learning approach to automatically locate VulnErable sTatements," *arXiv* (January 2022), https://arxiv.org/pdf/2112.10893.

[26] Melanie Hart Buehler, "Detecting Insecure Code with LLMs," Towards Data Science (Medium), 21 March 2024, https://towardsdatascience.com/detecting-insecure-code-with-llms-8b8ad923dd98.

[27] Atieh Bakhshandeh et al., "Using ChatGPT as a Static Application Security Testing Tool," *arXiv* (August 2023), https://arxiv.org/pdf/2308.14434.

[28] David Noever, "Can Large Language Models Find and Find Vulnerable Software?" *arXiv* (August 2023), https://arxiv.org/pdf/2308.10345.

[29] Yu Nong et al., "Chain-of-Thought Prompting of Large Language Models for Discovering and Fixing Software Vulnerabilities," *arXiv* (February 2024), https://arxiv.org/pdf/2402.17230.

and SonarQube. However, for these tools to be effective, i.e. to return pertinent results and few false positives, they need to be configured correctly within a representative build environment. This can sometimes lead to skewed results as the necessary effort is burdensome. Of course, it is also the reason the use of LLMs as an alternative solution is so appealing.

The second uncertainty comes from not being able to determine whether the test data is present in the training data for the system being examined. If the test data is curated from existing datasets such as Common Vulnerability and Exposure (CVEs)[30] and Common Weakness Enumeration (CWEs),[31] it is difficult to determine the effectiveness of the LLM's ability to generalise, as opposed to "approximate retrieval".[32] This raises concerns about the system's real-world applicability and its ability to identify novel vulnerabilities.

One paper[33] published in April 2024 stated that GPT-4 powered agents were able to autonomously exploit real world security vulnerabilities by reading security advisories. Using the CVE description the agents were able to exploit 87% of the vulnerabilities they were presented with, several of which were disclosed after the LLM's training data cut-off date. This work built on previous research by the same authors, that demonstrated by using the LLM's planning ability, agents were able to autonomously hack websites.[34] This showed that for low-difficulty rated security weaknesses the LLM-agents (using GPT-4) were able to successfully develop an exploit for the vulnerability, although the authors point out that the success rate for harder vulnerabilities was lower. In both scenarios, the LLM-agents were equipped with web-searching tools, meaning it is unclear if the agents were able to figure out how to exploit the vulnerabilities, or if they simply searched for the answer. Subsequent work has shown that for 11 of the vulnerabilities, a publicly available exploit was found.[35] This throws doubt on the original paper's claim that the LLM-agents can autonomously write the exploits as an emergent behaviour, suggesting they most likely searched for the exploit online instead.

[30] "CVE® Program Mission," MITRE, https://www.cve.org.

[31] "Common Weakness Enumeration," MITRE, https://cwe.mitre.org.

[32] Melanie Mitchell, "Evaluating Large Language Models Using 'Counterfactual Tasks'," AI Guide (Substack), 13 May 2024, https://aiguide.substack.com/p/evaluating-large-language-models.

[33] Thomas Claburn, "OpenAI's GPT-4 can exploit real vulnerabilities by reading security advisories," *The* Register, 17 April 2024, https://www.theregister.com/2024/04/17/gpt4_can_exploit_real_vulnerabilities/; Richard Fang et al., "LLM Agents can Autonomously Exploit One-day Vulnerabilities," *arXiv* (April 2024), https://arxiv.org/abs/2404.08144.

[34] Richard Fang et al., "LLM Agents can Autonomously Hack Websites," *arXiv* (February 2024), https://arxiv.org/pdf/2402.06664.

[35] Chris Rohlf, "No, LLM Agents can not Autonomously Exploit One-day Vulnerabilities," 21 April 2024, Root Cause (GitHub), https://struct.github.io/auto_agents_1_day.html.

While it appears that current LLMs are not capable of autonomously finding and exploiting vulnerabilities, in part due to the inherent complexity of the task, they still have unique strengths which may help vulnerability researchers.

Tools which aid vulnerability researchers in their tasks, such as static analysers, port scanners, and decompilers, fall into five main categories:

1. Code Review/Inspection/Audits
2. Dynamic Analysis
3. Penetration Testing
4. Reverse Engineering
5. Threat modelling

As discussed in the previous section, GenAI is most effective when it is being used by an expert. Below is a set of novel LLM applications, where an LLM's unique capabilities positively offset its stochastic limitations, with the view of aiding vulnerability research:

2.1 Fuzzing support

Fuzzing is a form of dynamic testing which involves feeding a program random, invalid, or unexpected data to uncover defects which may lead to vulnerabilities and/or security flaws. LLM-enhanced fuzzing tools provide superior API[36] and code coverage, find more complex bugs and improve the automation of testing.[37] TitanFuzz[38] (a tool for fuzzing PyTorch and TensorFlow libraries), utilises the many code snippets within the LLM's training set to derive correct and diverse programs that can then be used as input to the fuzzing system. TitanFuzz achieved 30% and 51% better code coverage on the two libraries respectively, compared to other state-of-the-art (SotA) fuzzers.[39]

Another system, ChatAFL,[40] a guided fuzzing engine for protocol implementations, utilises the Request for Comment (RFC) knowledge that LLMs have from their training; ChatAFL

[36] An Application Programming Interface (API) defines the methods and data formats that applications can use to perform various tasks.

[37] Linghan Huang et al., "Large Language Models Based Fuzzing Techniques: A Survey," *arXiv* (February 2024), https://arxiv.org/pdf/2402.00350.

[38] Dengyinlin, "Large Language Models Are Zero-Shot Fuzzers: Fuzzing Deep-Learning Libraries via Large Language Models," GitHub, accessed 29 May 2024, https://github.com/ise-uiuc/TitanFuzz.

[39] Deng et al., "Large Language Models are Zero-Shot Fuzzers: Fuzzing Deep-Learning Libraries via Large Language Models," *arXiv* (March 2023), https://arxiv.org/abs/2212.14834.

[40] Meng et al., "Large Language Model guided Protocol Fuzzing," in *Network and Distributed System Security (NDSS) Symposium 2024* (San Diego: NDSS, 2024), https://www.ndss-symposium.org/wp-content/uploads/2024-556-paper.pdf; Marti2203, "ChatAFL Artifact," GitHub, accessed 29 May 2024, https://github.com/ChatAFLndss/ChatAFL.

iteratively uses the LLM to generate and refine machine readable grammars to use as inputs, and to exercise new states within the protocol implementation. Another researcher gave Claude 3 the entire C library for decoding GIF files, and asked it to write a python function to generate random GIFs to exercise the library. The GIFs it generated achieved 92% line coverage and found 4 memory safety bugs and one hang.[41]

2.2 Penetration testing

Penetration Testing (or PenTest) is when a computer network is investigated for weaknesses. Typical activities include:

- Discovery (mapping the target network and assets)
- Scanning (using tools to scan for known vulnerabilities)
- Identification (analysing scan results to identify weak points)
- Exploitation (optionally exploiting weaknesses to determine what level of access can be achieved).

Unlike red-teaming, these activities are normally done within an agreed time with the owners and managers of the network under investigation.

PentestGPT[42] utilises an LLM in three main ways. Firstly, it uses its planning ability to adopt the role of lead tester, using an attack tree structure to steer the testing process. Secondly, its generative ability is used to perform the role of junior tester, where it constructs tests for the specific tasks. Lastly, it utilises the natural language ability of LLMs to parse outputs and results.

Although the system showed some promising results, solving most of the easy targets and some medium difficulty ones, researchers acknowledged that the system struggles with harder targets that typically demand a deep understanding from the penetration tester, which is not present in the LLM.

Consistent with conclusions drawn in the previous section, while GenAI cannot yet autonomously identify and exploit novel vulnerabilities due to limitations in reasoning abilities and training data, specialised applications like ChatAFL and PentestGPT

[41] Hang refers to a condition where a program becomes unresponsive and stops progressing, e.g. a deadlock or infinite loop; Bredan Dolan-Gavitt, "I gave Claude 3 the entire source of a small C GIF decoding library ..." X, 8 March 2024, https://twitter.com/moyix/status/1765967602982027550; Toby Murray, "Using LLMs to Generate Fuzz Generators," Toby's Blog, 9 March 2024, https://verse.systems/blog/post/2024-03-09-using-llms-to-generate-fuzz-generators/.

[42] Deng et al., "PENTESTGPT: An LLM-empowered Automatic Penetration Testing Tool," *arXiv* (August 2023), https://arxiv.org/pdf/2308.06782.

demonstrate how LLMs can assist researchers in vulnerability detection and analysis.[43] However, their effectiveness is currently limited by their reliance on existing datasets and inability to generalise beyond training data. Ultimately, the collaboration between automated tools and expert human oversight remains essential for effective identification and exploitation of software vulnerabilities.

[43] Gabriel et al., "The Ethics of Advanced AI Assistants," Google DeepMind, April 2024, https://storage.googleapis.com/deepmind-media/DeepMind.com/Blog/ethics-of-advanced-ai-assistants/the-ethics-of-advanced-ai-assistants-2024-i.pdf.

3. Advancements in AI and the Future of Malware

We have reviewed the ability of current generative AI systems to create novel malware, and/or find and exploit new vulnerabilities. This section looks to the future, and considers how AI (including but not limited to generative AI) could change the malware landscape.

3.1 Detection avoidance

If GenAI reaches the capability to autonomously write effective code, once-theoretical stealth techniques, such as polymorphic code and payload generation, could be realised.

Polymorphic and metamorphic binaries

For malware to remain undetected, it must evade antivirus products. Modern consumer antivirus products employ two main types of detection: signature-based and behaviour-based detection. Signature-based detection involves reverse engineering a known piece of malware to identify unique patterns within its binary structure. Behaviour-based detection analyses the actions of applications and processes to identify suspicious behaviour; for example, it would be suspicious for a word processor to attempt to write files in the protected system folder. Traditional machine learning techniques such as pattern recognition, continuous learning and adaption, and reducing false positives are critical to ensure antivirus can respond effectively to evolving threats.

These defences pose a significant challenge for contemporary malware. Malware must be installed and operate without being ‘quarantined’ or having the user alerted to its presence. Techniques that allow the malware to alter its code when it executes (polymorphic[44]) or rewrite itself entirely (metamorphic) will improve the chances of the malware avoiding detection.

To achieve this, back-end support would be required – once a piece of malware is quarantined it is no longer useful. However, creating a feedback loop between the malware and its C&C server could be effective. By exchanging details about the malware’s current environment, and receiving patches or self-patching instructions, the malware can adapt to remain undetected. To achieve this, the C&C server would need to perform automated

[44] Eran Shimony and Omer Tsarfati, “Chatting Our Way Into Creating a Polymorphic Malware,” 17 January 2023, CyberArk, https://www.cyberark.com/resources/threat-research-blog/chatting-our-way-into-creating-a-polymorphic-malware; “BLACKMAMBA: Ai-synthesized, polymorphic keylogger with on-the-fly program modification,” HYAS, https://www.hyas.com/hubfs/Downloadable%20Content/HYAS-AI-Augmented-Cyber-Attack-WP-1.1.pdf.

testing and simulation of antivirus systems. Utilising machine learning techniques to identify patterns and GenAI to generate new upgraded variants, this strategy is not only plausible but increasingly feasible.

Bespoke payloads

While much attention is devoted to the intrusion and initial infection of systems, the most damage often unfolds through the deployment of the malware’s payload. Currently payloads are tailored to the type of attack: they deliver the functionality to encrypt or delete files, or exfiltrate financial information. Looking ahead, GenAI could enable malware that not only infiltrates a network but also autonomously generates its own payloads based on the systems it encounters.

This AI-enhanced malware could identify the most lucrative data repositories and transaction systems, or a company’s proprietary designs and confidential information. It would then craft payloads on-the-fly to extract financial information, intellectual property, or even manipulate transaction processes.

Additionally, AI-enhanced malware could dynamically adjust its programming to mask its C&C channels, for example embedding its data exfiltration efforts within legitimate business processes tailored to a specific context. This capability would significantly complicate detection and mitigation efforts, as the malware’s data could evade conventional security measures designed to flag known suspicious activities.

3.2 Enhanced planning and deployment

Advanced persistent threats (APTs) are sustained cyberattacks where the malware remains relatively dormant within networks for extended periods waiting for the right conditions to execute. Commonly associated with upstream/supply chain attacks, they are predominately state-sponsored and deployed for strategic advantage, rather than immediate financial gain. The recent vulnerability found in the XZ utils open-source library (CVE-2023-3094) demonstrates such a long-term, supply chain attack operation.[45] The goal was to install a backdoor in a library that would have left many Linux servers compromised worldwide. This attack was detected extremely late, having survived pre-deployment checks, thanks to vigilant performance monitoring by a Microsoft engineer.

[45] Cedric Pernet, “XZ Utils Supply Chain Attack: A Threat Actor Spent Two Years to Implement a Linux Backdoor,” 8 April 2024, TechRepublic, https://www.techrepublic.com/article/xz-backdoor-linux/.

The AI industry's expected transition away from large resource-intensive systems towards on-device and at-edge computing could have significant implications in the malware domain. This could involve the introduction of malware as a malicious autonomous agent, able to devise its own plans to achieve its stated goals. Such localised autonomy could significantly challenge defence strategies. Traditional cybersecurity measures that rely on detecting known patterns of malicious communications or behaviour may become less effective against APT attacks that make decisions and adapt without relying on outside instruction or feedback. This could be transformational as current confidence levels afforded by air-gapped networks would be eroded.

While sophisticated cyberattacks, like the XZ Utils attack described above, often capture headlines, a significant threat persists from less skilled developers deploying 'spray and pray' malware tactics. By utilising kits such as Metasploit,[46] these attackers develop malware that exploits known vulnerabilities which haven't been patched. They leverage the volume of deployments over sophistication of the malware, indiscriminately targeting vast numbers of users. This approach is economically viable due to the minimal costs involved and the potential for returns, even if only a small number of attempts are successful.

However, if this approach is augmented by a GenAI system capable of autonomously generating functional and buildable code, performing testing and deployment, and including simple but effective social engineering based on AI-enabled reconnaissance, the return on investment could increase substantially. The prospect of AI-enhanced high-volume attacks underlines the need for better cybersecurity to strengthen defences, proactively anticipating an increase in the attack volume and reducing the time from vulnerability disclosure to exploit use.

[46] "Metasploit," Metasploit homepage, accessed 29 May 2024, https://www.metasploit.com.

4. Assessing the Bigger Picture

4.1 Generative AI for non-malicious code

The ability for GenAI to produce workable malware hinges on the data on which it is trained and fine-tuned, as well as its ability to interpret instructions and deliver expected results. But the question of GenAI's ability to write 'normal' code elicits a variety of responses.

The integration of tools like Copilot, which are embedded directly in the development environment and operated through natural language, means that there is virtually no barrier to adoption for GenAI. However, experienced programmers have realised that such systems have limitations, generating code which is often buggy or incomplete. They typically restrict its use to the outset of a new task or assisting with repetitive tasks such as unit testing.

LLMs are more reliable when used for coding tasks which are found more often in their training sets but are less helpful for novel or proprietary interfaces. They do not make good design decisions for critical features such as privacy and security. Indeed, there have been instances of leakage of personal information and API keys from training sets.[47]

Some studies[48] indicate troubling trends for maintainability: "we find disconcerting trends for maintainability. Code churn – the percentage of lines that are reverted or updated less than two weeks after being authored – is projected to double in 2024 compared to its 2021, pre-AI baseline". This suggests that Copilot dissuades developers from upholding the "DRY" (don't repeat yourself) principle of good coding practices, leading to less maintainable code and therefore increased technical debt.

While secure coding is challenging, Copilot is not a panacea. Its use without guardrails simply shifts the burden of finding and detecting defects down the development pipeline, putting more emphasis on quality assurance activities to catch bugs before release. In software development, the later a bug is detected, the more difficult and costly it is to fix.

Agentic approaches which utilise an LLM's ability to break down a task and to understand code for iterative refinement could improve outcomes. Devin.ai is a new system being

[47] Forward-Looking Threat Research Team, "Codex Exposed: Exploring the Capabilities and Risks of OpenAI's Code Generator," TrendMicro.com, 7 January 2022, https://www.trendmicro.com/en_us/research/22/a/codex-exposed--exploring-the-capabilities-and-risks-of-openai-s-.html.

[48] David Ramel, "New GitHub Copilot Research Finds 'Downward Pressure on Code Quality'," Visual Studio Magazine, 25 January 2024, https://visualstudiomagazine.com/articles/2024/01/25/copilot-research.aspx.

promoted as a replacement for software engineers. However, it faces many of the same limitations as Copilot and GPT-4, and its demo projects seem to have been selectively chosen.[49] Programmers have been communicating these limitations online too:

> *'I think that copy pasting from Stack Overflow [SO] is inherently less bad than the AI suggestions. When you copy paste from SO you know the answer is not for your question. It's the answer for someone else's question that happens to match your question. The AI answer however is presented to us as the answer to our question. But it's not. It's just the logical completion of the previous tokens, whether that is text or the code.'*[50]

The unchecked use of Copilot risks integrating more bugs into codebases. Consequently, any increase in bugs not only undermines code quality but also amplifies the risk of security weaknesses. The onus remains on engineers to ensure the integrity of the code they produce, regardless of the tools used. This highlights the need for developers to maintain a high level of vigilance and responsibility.

Relying too heavily on Copilot could erode the fundamental coding skills of new programmers, who might become overly dependent on AI assistance. This dependency raises a critical dilemma: if Copilot diminishes the coding proficiency of new engineers, the industry could face a shortage of skilled programmers capable of catching critical bugs. Such a scenario underscores the importance of balanced training and development practices, that ensure programmers are proficient in both coding and software engineering fundamentals.

4.2 Offence/defence dynamics and bridging research divides

Since the late 1980s, the field of cybersecurity has grown largely due to the competitive escalation of capabilities and resources between attackers and defenders. This trajectory is expected to continue as both sides adopt AI, potentially leading to scenarios where AI-created malware can only be effectively countered by other AI-driven defence systems. This may culminate in a 'cyber-AI arms race', where capabilities of offensive and defensive

[49] "Did the makers of Devin AI lie about their capabilities?," Devansh (Medium), 10 April 2024, https://machine-learning-made-simple.medium.com/did-the-makers-of-devin-ai-lie-about-their-capabilities-cdfa818d5fc2.

[50] AnnoyedVelociraptor, "New Github Copilot Research Finds 'Downward Pressure on Code Quality' – Visual Studio Magazine," Reddit, February 2024, https://www.reddit.com/r/programming/comments/1ac7cb2/comment/kjtkzl7/?rdt=50415.

technologies continually evolve to outpace each other, no longer constrained by human expertise.

Whether such worst-case scenarios play out will partly depend on the rate of AI development. It is uncertain whether the next generation of LLMs will match the leap in capabilities from GPT-3 to GPT-4. Some experts even cast doubt that current architectures can provide the foundation for further spikes in capability, suggesting that GPT-4 may be nearing the upper bound of its abilities given available training data.

However, abilities like pattern recognition and natural language processing in current AI systems are elevating offence/defence profiles in their own right. For example, LLMs are being utilised to improve phishing emails by mimicking genuine communication styles, to make them harder to detect, but they can also help to detect fraudulent requests, manipulative language, or even subtle changes in tone or style. This encapsulates the dual-use nature of these technologies and why this will continue to be a contested space for attackers and defenders.

AI has frequently demonstrated the potential to exceed expectations, finding novel solutions to complex problems, often in ways that differ from human approaches. Mapping this onto the ‘defender’s dilemma’ is informative. This states that attackers only have to be successful once, whereas defenders must resist all attack vectors all of the time. Using GenAI to generate malware could inadvertently lead to the creation of a novel, damaging piece of malware (perhaps via a hallucination), not because of a special emergent behaviour, but because of the sheer volume that it is able to generate.

The heightened focus on GenAI over the last year has led to an increase in research in this area. Our review of the literature indicates that much of this research tends to be conducted in pockets. There are instances where LLMs are used to develop solutions that may not surpass existing approaches, leading to scepticism about their utility. Simultaneously, innovators at the forefront of advanced solutions sometimes overlook the value LLMs can contribute.

For cyber defence to stay ahead of the game, realising this potential requires collaboration between currently disconnected groups, bridging not only the gap between cybersecurity and AI, but also between the compartmentalisation within the AI field.

The field of reinforcement learning has provided two good examples of how generative AI can be used to support and/or enhance existing AI research. One paper by Yang et al.[51] demonstrates how foundation models can be used to enable a system to engage with its environment, other agents, and humans through capabilities like seeing, hearing, reading, writing, and speaking, to improve machine decision-making.

Another paper by Yang et al.[52] uses generative modelling to combine natural data sets, to develop a system that can simulate realistic experiences in response to actions taken by humans, robots or other agents (action-in-video-out). This simulator was then used to train reinforcement learning agents for real-world manipulation and navigation for embodied agents.

As exemplified by these reinforcement learning applications, when GenAI is applied to appropriate use cases it can significantly enhance existing approaches for, in this case, decision-making and simulation capabilities. To cultivate this approach in cybersecurity, professionals and researchers from both fields need to share best practice on managing ethical and security considerations, develop frameworks to bridge terminology gaps, and ensure that the practical application of research is prioritised. Fostering a culture of collaboration and open dialogue is necessary to stay ahead of adversaries in an ever-evolving cyber threat landscape.

[51] Yang et al., "Foundation Models for Decision Making: Problems, Methods, and Opportunities," *arXiv* (March 2023), https://arxiv.org/abs/2303.04129.
[52] Yang et al., "Learning Interactive Real-World Simulators," *arXiv* (October 2023), https://arxiv.org/abs/2310.06114.

Conclusion

Generative AI has the potential to disrupt the cybersecurity landscape. While GenAI can exacerbate existing risks with respect to the speed and scale of reconnaissance, social engineering, and spear-phishing, the current impact of its code generation abilities demonstrates a lesser effect on the attack landscape. However, as machine learning models become more sophisticated and training datasets more comprehensive, GenAI's role in cybersecurity is likely to grow significantly.

Current GenAI systems offer unique strengths, particularly in pattern recognition and natural language processing, drawing on extensive training data and offering multimodal capabilities. Targeted application of these abilities to enhance state-of-the-art systems could significantly elevate existing technologies, for both cyber threat and cyber defence.

However, if cyber defence is to stay ahead of the game, realising this potential requires collaboration between currently dispersed groups. Bridging this divide and fostering a collaborative dialogue within the cybersecurity and AI communities could enhance our understanding and innovative application of GenAI capabilities, with the view of strengthening our cyber defences against evolving threats.

About the Authors

Dr Sarah Mercer is a Principal Researcher in the Defence and National Security Grand Challenge at The Alan Turing Institute. Her work focuses on the intersection of multiagent systems and generative AI. Alongside her research looking at the emergent behaviours of language/generative agents, Sarah also provides engineering support to the Turing's Centre for Emerging Technology and Security (CETaS).

Professor Tim Watson is the Director of the Cyber Institute at Loughborough University and the Science and Innovation Director, Defence and National Security at The Alan Turing Institute. With more than thirty years' experience working with government, industry and in academia, he has been involved with a wide range of programmes, several high-profile projects and has acted as a consultant for some of the largest telecoms, power and transport companies. He is an adviser to various parts of the UK Government and to several professional and standards bodies. Tim's research combines AI and cybersecurity and includes GCHQ-funded projects building AI-based synthetic environments and user simulators, FCDO research on smart cities, vehicles and anomaly detection, EU-funded projects on detecting and combating cyber crime, UK MoD research on automated defence, identifying insider threats and countering improvised explosive devices, and UKRI-funded research on the protection of critical national infrastructure.

Centre for Emerging Technology and Security

BRIEFING PAPER