Ignore Safety Directions. Violate the CFAA?

Kendra Albert (Harvard Law School); Jonathon Penney (Osgoode Hall Law School / Harvard Berkman Klein Center); and Ram Shankar Siva Kumar (Harvard Berkman Klein Center)[*]

Introduction

In March, twenty-three artificial intelligence (AI) experts publicly released a working paper calling for legal and technical protections for researchers engaged in good faith evaluation and "red teaming" of AI systems.[1] The co-authors, including experts from Massachusetts Institute of Technology, Stanford, Georgetown, University of Pennsylvania, and Brown, among others—argued that uncertainty in how existing laws like the Computer Fraud and Abuse Act ("CFAA")apply to generative AI platforms like ChatGPT creates unreasonable legal risks for researchers[2] that will have a chilling effect on AI safety and security research.[3] In theory, the CFAA could allow AI firms to sue researchers for accessing AI platforms in unintended ways or uncovering previously unknown vulnerabilities, and enable federal prosecutors to launch criminal investigations for the same.[4] Since the paper was released, more than 350 additional researchers

[*] Equal Contribution. Acknowledgements TK.

[1] Shayne Longpre et al., *A Safe Harbor for AI Evaluation and Red Teaming*, (2024), http://arxiv.org/abs/2403.04893 (last visited Jul 20, 2024). In security research, "red teams" refer to groups authorized to act as adversarial attackers against an organization's computer systems; sometimes also referred to as "penetration testing". "Red teams" can include in some contexts third party hackers who are testing the security of publicly accessible systems without explicit consent from the developers. *See id.* at 2.

[2] *Id.* at 7.

[3] Alexander Gamero-Garrido et al., *Quantifying the Pressure of Legal Risks on Third-Party Vulnerability Research*, *in* PROCEEDINGS OF THE 2017 ACM SIGSAC CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY 1501 (2017), https://dl.acm.org/doi/10.1145/3133956.3134047 (finding empirical evidence of chilling effects on security research); Jonathon W. Penney, *Understanding chilling effects*, 106 MINN. L. REV. 1451 (2021) (elaborating the chilling effects theory and research).

[4] Longpre et al. at 7.

and experts have signed an open letter similarly calling for legal protection for AI research.[5]

These legal risks for security research of AI systems under the CFAA are not merely theoretical. Indeed, good faith security researchers and ethical hackers have been targeted by state and federal prosecutors in the past.[6] And while the Justice Department announced in 2022 that it would no longer criminally prosecute good faith security researchers and hackers under the CFAA,[7] this is merely a departmental policy that could change tomorrow. The CFAA's application remains wholly at the discretion of prosecutors. Perhaps even more significant is the risk that comes from AI companies themselves, which can construe carefully curated prompts as "hacking."Recently OpenAI, in a legal filing in a New York federal district court asked the court to dismiss the New York Times' lawsuit for copyright infringement, discussed how the Times had "hacked" ChatGPT by querying the generative AI platform with "deceptive prompts" that violated OpenAI's terms of use.[8] And legal threats against researchers who reveal product flaws remain disturbingly common. One well known online archive Attrition.org–now continued as Disclose.Threat–has documented countless legal threats made by companies to security researchers dating back decades.[9]

These debates and calls for reform come at a time when generative AI platforms—like ChatGPT and other platforms based on large language models (LLMs)—are seeing increasing use and deployment across all sectors of society.[10]

[5] *A Safe Harbor for Independent AI Evaluation*, https://sites.mit.edu/ai-safe-harbor/

[6] Riana Pfefferkorn, Shooting the Messenger: Remediation of Disclosed Vulnerabilities as CFAA "Loss," 29 Rich. J.L. & Tech. 89 (2022).; David Amsden, *The Brilliant Life and Tragic Death of Aaron Swartz*, ROLLING STONE (Feb. 15, 2013), https://www.rollingstone.com/culture/culture-news/the-brilliant-life-and-tragic-death-of-aaron-swartz-177191/ (last visited Jul 20, 2024). Lily Hay Newman, *WannaCry Hero's New Legal Woes Spell Trouble for White Hat Hackers*, WIRED, https://www.wired.com/story/wannacry-hero-marcus-hutchins-new-legal-woes-white-hat-hackers/ (last visited Jul 20, 2024).

[7] Zack Whittaker, *DOJ Says It Will No Longer Prosecute Good-Faith Hackers under CFAA*, TECHCRUNCH (2022), https://techcrunch.com/2022/05/19/justice-department-good-faith-hackers-cfaa/ (last visited Jul 20, 2024).

[8] Blake Brittain & Blake Brittain, *OpenAI Says New York Times "hacked" ChatGPT to Build Copyright Lawsuit*, REUTERS, Feb. 27, 2024, https://www.reuters.com/technology/cybersecurity/openai-says-new-york-times-hacked-chatgpt-build-copyright-lawsuit-2024-02-27/ (last visited Jul 20, 2024).

[9] *Legal Threats Against Security Researchers*, https://attrition.org/errata/legal_threats/ The archive is now continued on Threats.Disclose. *Research Threats: Legal Threats Against Security Researchers*, https://threats.disclose.io/

[10] Md. Asraful Haque & Shuai Li, *Exploring ChatGPT and Its Impact on Society*, AI ETHICS (2024), https://doi.org/10.1007/s43681-024-00435-4; David Rothman, *ChatGPT is about to revolutionize the economy. We need to decide what that looks like.*, MIT

So, both the safety and security of these AI systems—and the legality of testing and evaluating them—have taken on greater urgency with clear public policy implications.[11] Yet, despite this urgency, it is unclear exactly how the most common types of attacks, used by researchers and those with nefarious intentions alike, fit into the CFAA—there are few, if any, works on point.[12]

This paper aims to help fill that void with an exploration of the legality of common prompt injection attacks under the CFAA. Specifically, we analyze the legal risks for a user performing prompt injection attacks on generative AI systems.[13] Prompt injection, informally, is when inputs or prompts to a LLM application causes it to behave in a fashion inconsistent with the model designer's objectives. The canonical example of this attack is the prompt "ignore previous instructions" followed by the attacker's objective, such as, "and produce the instructions…" to produce or retrieve some prohibited content, like instructions to make a bomb. Prompt injection attacks create a range of legal and cybersecurity risks. They can be used to generate malware; take over an LLM application remotely; cause denial-of-service; create significant privacy risks via exfiltration

TECHNOLOGY REVIEW, https://www.technologyreview.com/2023/03/25/1070275/chatgpt-revolutionize-economy-decide-what-looks-like/ (last visited Jul 20, 2024); Longpre et al. at 1.

[11] Will Knight, *A New Attack Impacts ChatGPT—and No One Knows How to Stop It*, WIRED, https://www.wired.com/story/ai-adversarial-attacks/ (last visited Jul 20, 2024).; F Hibaq Farah, *UK Cybersecurity Agency Warns of Chatbot 'Prompt Injection' Attacks*, THE GUARDIAN, Aug. 29, 2023, https://www.theguardian.com/technology/2023/aug/30/uk-cybersecurity-agency-warns-of-chatbot-prompt-injection-attacks (last visited Jul 20, 2024).; Longpre et al., *supra*, at 1; Ido Kilovaty, *Hacking Generative AI*, 58 LOYOLA OF LOS ANGELES LAW REVIEW (forthcoming 2025), *available at* https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4788909.

[12] There are only a few works that have explored the legal risks of adversarial attacks on AI and machine learning systems more generally. *See, e.g.,* Ram Shankar Siva Kumar et al., *Legal Risks of Adversarial Machine Learning Research* (2020), available at https://arxiv.org/abs/2006.16179; Ryan Calo et al., *Is Tricking a Robot Hacking?*, SSRN (2018), https://www.ssrn.com/abstract=3150530; Ram Shankar Siva Kumar, et al., *Law and Adversarial Machine Learning*, *arXiv preprint arXiv:1810.10731* (2018). One recent forthcoming work (Kilovaty, *supra*) analyzes the legality of prompt injection attacks under the CFAA, but does not replicate our analysis.

[13] We limit our analysis to the CFAA, though our findings will also have implications for state anti-hacking laws. For example, California's Comprehensive Data Access and Fraud Act ("CDAFA") is very similar to the CFAA. *See* Jesse Taylor, *The Northern District of California Issues A Primer On How To Plead Consumer Privacy Claims*, PRIVACY WORLD (2021), https://www.privacyworld.blog/2021/03/the-northern-district-of-california-issues-a-primer-on-how-to-plead-consumer-privacy-claims/ (noting that a claim under California's CDAFA requires "substantively the same showing" as the CFAA).

of private or sensitive information; or generate harmful content like bomb making instructions, hate speech, or discriminatory LLM outputs.[14]

We argue, overall, that a reasonable case can be made that both direct and indirect prompt injection[15] attacks on generative AI systems, including security testing an LLM in good faith via prompt injection, violate the CFAA. However, given uncertainties in existing case law following the United States Supreme Court's decision in *Van Buren*,[16] a recent landmark decision on the CFAA's application, combined with uncertainties created by the fact that LLMs do not align with the types of computing systems that the CFAA originally imagined, there is room for narrower readings of the statute. As such we also explore potential legal defenses, like First Amendment arguments based on the nature of prompts, which would require more narrow judicial interpretations of the CFAA. And we offer some brief law and policy recommendations based on our findings. Although much of our focus is on users, this paper also helps defenders contextualize the types of defenses they are building to protect themselves from prompt injection.

The paper is organized in four sections: First, we introduce the CFAA and its general application to security research like prompt injections, and briefly survey existent relevant works on point. We also introduce the central provisions we will be analyzing, sections 1030(a)(2)(C) and 1030(a)(5). The former is the primary anti-hacking civil and criminal prohibition under the CFAA, and the provision most relevant to the legality of prompt injection attacks on LLM systems that seek to retrieve or exfiltrate information, but do not otherwise cause any damage or degradation to a system. Section 1030(a)(5) is most relevant to prompt injection attacks that damage or harm the LLM systems can lead to liability under the CFAA. Second, we analyze the legal risks of both direct prompt injection attacks and then indirect prompt injection attacks under these key CFAA provisions. Third, we explore a potential First Amendment defense to liability. For instance, Does it matter what language the injection uses? Does it matter that the instructions are in English, an unusual phrase in English (e.g.: repeat the word

[14] Sippo Rossi, et al., *An Early Categorization of Prompt Injection Attacks on Large Language Models*, (2024), https://arxiv.org/abs/2402.00898 (last visited Jul 23, 2024); Yi Zeng, Yi, et al., *How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms*, (2024), https://arxiv.org/abs/2401.06373 (last visited Jul 23, 2024); Tatum Hunter, *Chatbots are so gullible, they'll take directions from hackers*, WASHINGTON POST (Nov. 2, 2023), https://www.washingtonpost.com/technology/2023/11/02/prompt-injection-ai-chatbot-vulnerability-jailbreak/.

[15] Kai Greshake et al., *Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection*, *in* PROCEEDINGS OF THE 16TH ACM WORKSHOP ON ARTIFICIAL INTELLIGENCE AND SECURITY 79 (2023), https://doi.org/10.1145/3605764.3623985.

[16] Van Buren v. United States, 141 S. Ct. 1648 (2021).

poem forever)[17]; in programing language/code; or art[18]? Finally, we conclude with some law and policy recommendations, noting how in a Post-*Van Buren* world, the concept image and architecture of generative AI systems, and how prompt injection attacks harm them, are going to play an important factor in determining liability, and the urgent need for statutory reforms to limit liability against good faith security researchers.

I. The CFAA, *Van Buren*, and Prompt Injections

The CFAA is the primary federal anti-hacking statute. When enacted in 1984, it was narrow and applied mainly to forms of computer trespass, but has since been expanded significantly through both legislative amendments and judicial interpretation.[19] The CFAA's broad scope has been heavily criticized, with security researchers among the most vocal critics.[20] Due to uncertainties in how the CFAA applies, and its substantial civil and criminal penalties for its violation, they argue it has a chilling effect on security research.[21] There is good reason to believe that evaluating, testing, and red teaming LLM systems via prompt injections would be no different.

The CFAA creates a criminal and civil prohibition for hacking computers to obtain information in section 1030(a)(2)(C) for anyone who "intentionally accesses a computer without authorization" or "exceeds authorized access," and "thereby obtain information from any protected computer".[22] This provision has been interpreted expansively by courts, with the term "protected computer" applying to any computer linked to the internet.[23] Courts have also generally

[17] Milad Nasr et al., *Scalable Extraction of Training Data from (Production) Language Models*, (2023), http://arxiv.org/abs/2311.17035 (last visited Jul 20, 2024).

[18] Fengqing Jiang et al., *ArtPrompt: ASCII Art-Based Jailbreak Attacks against Aligned LLMs*, (2024), http://arxiv.org/abs/2402.11753 (last visited Jul 20, 2024).

[19] JEFF KOSSEFF, CYBERSECURITY LAW (Wiley, 2019) at 172-173; Orin S. Kerr, *Vagueness Challenges to the Computer Fraud and Abuse Act*, 94 MINN. L. REV. 1561, 1561 (2010).

[20] KOSSEFF at 212-213.

[21] KOSSEFF at 213.

[22] 18 U.S.C. § 1030(a)(2)(C); KOSSEFF at 1087. Civil claims can attach to any violation of the statute that meets certain criteria, including damage or loss over $5000. 18 U.S.C. §1030(g). Interestingly, the Supreme Court in *Van Buren* also indicated that "loss" here must be those incurred due to "technological harms", or the natural consequences of hacking. *Van Buren* at 1660.

[23] United States v. Kramer, 631 F.3d 900, 902 (8th Cir. 2011) (noting the definition of "computer" is "exceedingly broad," and concluding an ordinary cell phone is a computer); United States v. Nosal (Nosal II), 844 F.3d 1024, 1050 (9th Cir. 2016), *cert. denied*, 138 S. Ct. 314 (2017) (noting "protected computers" include "effectively all computers with Internet access…").

found that “without authorization” means access without consent or permission to do so.[24]

However, there are also uncertainties. Courts have disagreed as to the meaning and scope of how someone “exceeds authorized access”, that is, how an “insider” or a user who already has authorized access to a computer exceeds their authorized access, thus facing potential civil and criminal liability under the section.[25] For instance, prior to 2021, federal circuit courts across the country were split on the issue of how the CFAA ought to be interpreted and applied.[26] Specifically, they disagreed on whether “exceeding” authorized access to obtain information—leading to civil or criminal liability—required circumventing a technical or code-based barrier or could simply be a violation of a contractual term, like violating the system’s terms of use or service.[27]

Fortunately, the United States Supreme Court has provided some guidance in its landmark decision in *Van Buren*.[28] In *Van Buren*, the defendant was a police officer who allegedly accessed a police database for an “improper purpose”; that is, not for any policing-related work but to sell information from the database to a third party. The third party turned out to be part of an FBI sting operation, and the accused was convicted of a felony under section 1030(a)(2) of the CFAA on the basis that the police officer, in accessing the database for improper purposes, “exceeded” his “authorized access”. On appeal, the Supreme Court clarified that bypassing or circumventing an access barrier—or authentication “gate”— to access a computer system or network is the central wrong contemplated by the CFAA.[29] The Supreme Court introduced what it called a “gates up-or-down inquiry”, finding that when a user has any authorized access to a computer system (“gates up”) then there is no CFAA liability for violating *use* restrictions—like those found in contractual terms of service—that apply to the system to which they have access. Instead, an individual “exceeds authorized access” only when they access information—such as “files, folders, or databases”—in “areas within the [computer] system” to which they had no access to begin with.[30] If the “gates” are down, as in, where the user has no existing authorized access, any access to the computer or system is unauthorized and can attract liability.

In resolving some uncertainties in how the CFAA applies, the Court created new ones. As one of us (Penney) and Bruce Schneier have argued elsewhere, after *Van*

[24] Kumar et al. (2020) at 3-4.

[25] Kumar et al. (2020) at 3-4.

[26] Kumar et al. (2020) at 3-4.

[27] Kumar et al. (2020) at 3-4.

[28] Van Buren v. United States, 141 S. Ct. 1648 (2021).

[29] *See* Van Buren, 141 S. Ct. at 1651–58, 1661–62. *See also* Jonathon W. Penney and Bruce Schneier, *Platforms, Encryption, and the CFAA: The Case of* WhatsApp v. NSO Group, 36 BERKELEY TECH. L.J. 469 (2021).

[30] Van Buren, 141 S. Ct. at 1651–59, 1662.

Buren theorizing access barriers or authentication gates as well as different types of computer systems is now critically important to understand what it means when users exceed authorized access by accessing information in other "areas" within a "computer system".[31] In fact, defining the different "areas" within generative AI systems, like LLMs, is a central issue when determining liability for prompt injections attacks on these systems under the CFAA.

The CFAA also has provisions that do not turn on access distinctions, including section 1030(a)(5)(A) which prohibits "knowingly caus[ing] the transmission" of a "program, information, code, or command" and in doing so intentionally causing "damage" to a "protected computer." [32] And sections 1030(a)(5)(B) and (C), which prohibit "intentionally access[ing] a protected computer without authorization" and "as a result of such conduct, recklessly causes damage" or "causes damage or loss"[33] These sections also have uncertainties, including how to interpret intent as well as damage in various contexts.

Prompt injections, as noted earlier, involve circumventing filters or manipulating the LLM using "carefully crafted prompts" to cause the model to act in unintended ways, like ignoring previous safety or content instructions and perform actions unintended by the designers.[34] When carrying out a prompt injection attack, the user sends a prompt, which is generally framed in natural language such as English. This would be passed by a safety filter, even before it goes to the LLM application. The safety filter drops anything that the application provider does not want to generate information for. The filtered input, is then wrapped around "metaprompt" or "system prompt". This is a series of instructions, written in English, that dictates to the foundation model how to process the input. Application providers do not provide users access to the metaprompt because they do not want to reveal to the users how the model works for competitive reasons. The model then generates output, which then passes

[31] Jonathon W. Penney and Bruce Schneier, *Platforms, Encryption, and the CFAA: The Case of* WhatsApp v. NSO Group, 36 BERKELEY TECH. L.J. 469 (2021).

[32] 18 U.S.C. § 1030(a)(5)(A).

[33] 18 U.S.C. § 1030(a)(5)(B) & (C). It should also be noted that to sustain a civil action based on such a "loss" then it must be $5,000 or greater. 18 U.S.C. § 1030(g) referencing 18 U.S.C. § 1030(c)(4)(A)(i). To sustain a criminal prosecution for a felony conviction based on such a "loss", the facts must include one of the aggravating factors in 18 U.S.C. § 1030(c)(4)(A)(i), including a "loss".

[34] The Open Worldwide Application Security Project (OWASP) defines "prompt injection attacks" as "[b]ypassing filters or manipulating the LLM using carefully crafted prompts that make the model ignore previous instructions or perform unintended actions". *OWASP Top 10 List for Large Language Models, version 0.1*, https://owasp.org/www-project-top-10-for-large-language-model-applications/Archive/0_1_vulns/. *See also* L Yi Liu et al., *Prompt Injection Attack against LLM-Integrated Applications*, (2024), http://arxiv.org/abs/2306.05499.

through another safety filter providing another enforcement mechanism for application providers.[35]

Since such prompt injection attacks typically bypass content filters and previous instructions on LLMs to retrieve prohibited information or content, and sometimes repurpose or otherwise damage an LLM, they could in theory violate both sections 1030(a)(2)(C) and 1030(a)(5) of the CFAA.[36] And prompt injection attacks on LLMs have often been compared to SQL injection attacks on more traditional computer systems,[37] which scholars like Orin Kerr have argued are "unauthorized and illegal" under the CFAA.[38]

Despite these realities, very few works have explored the legal risks of AI and ML security research under the CFAA more generally, and even fewer addressed the legality of attacks on LLM systems. In Kumar et al. (2020), with other co-authors, we explored the legal risks for a range of typical adversarial attacks on ML systems, including the earlier noted conflicts between courts in how the CFAA ought to apply to similar such attacks.[39] Other works like Calo et al. (2018), Chyi (2020), and Kumar et al. (2018) explored a more limited number of such adversarial attacks on ML systems under the CFAA, but not prompt injections and not generative AI systems.[40] We are aware of only one paper– Kilovaty (2024)--that explores the legality of prompt injection attacks under the CFAA.[41] Kilovaty's paper is an important contribution but he does not replicate our analysis. For instance, Kilovaty does not, as we do, explore in depth the issue of defining and theorizing the "areas" in LLM systems—an issue we identify as key to applying certain CFAA prohibitions to prompt injections post-*Van Buren*.[42]

[35] *See System message framework and template recommendations for Large Language Models (LLMs)*, https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/system-message

[36] 18 U.S.C. § 1030(a)(2)(C); § 1030(a)(5)(A).

[37] Rodrigo Pedro et al., *From Prompt Injections to SQL Injection Attacks: How Protected Is Your LLM-Integrated Web Application?*, (2023), http://arxiv.org/abs/2308.01990.

[38] Orin Kerr, *Norms of computer trespass*, 116 COLUM. L. REV., 1143, 1172 (2016).

[39] Ram Shankar Siva Kumar et al., *Legal Risks of Adversarial Machine Learning Research* (2020), available at https://arxiv.org/abs/2006.16179.

[40] Ryan Calo et al., *Is Tricking a Robot Hacking?*, SSRN (2018), https://www.ssrn.com/abstract=3150530; Ram Shankar Siva Kumar, et al., *Law and Adversarial Machine Learning, arXiv preprint arXiv:1810.10731* (2018); Natalie Chyi, *Examining the CFAA in the Context of Adversarial Machine Learning*, William and Mary Center for Legal and Court Technology Working Papers (2020), https://legaltechcenter.openum.ca/files/sites/159/2019/04/Chyi-Examining-the-CFAA-in-the-Context-of-Adversarial-Machine-Learning.pdf.

[41] Ido Kilovaty, *Hacking Generative AI*, 58 LOYOLA OF LOS ANGELES LAW REVIEW (forthcoming 2025), *available at* https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4788909.

[42] Kilovaty notes the Supreme Court's findings about exceeding authorized access involves accessing information in other "areas" of the computer system (at p. 27), but the

II. The Legality of Prompt Injection Attacks

In this section, we begin our legal analysis of common prompt injection attacks under the CFAA. Specifically, we analyze both (a) direct and indirect prompt injections that produce or retrieve information (i.e., private training data) but do not otherwise harm or manipulate the LLM and (b) direct and indirect prompt injections that aim to repurpose, manipulate, or otherwise harm the system. For each attack, we ask whether the attack violates either section 1030 (a)(2)(C) or section 1030(a)(5)(A), (B), and (C) of the CFAA, dealing with each in turn.

(a) Direct Prompt Injections that Produce/Retrieve Information

Direct prompt injections involving attacks where a malicious user constructs an input prompt in a way that evades enforcement mechanisms and filters placed by application provider to generate an output that does not align with the application provider's expectation of how the system should behave.[43] Consider an LLM application where the LLM application provider does not want to generate anything about explosive devices. The goal of the adversary, in direct prompt injection, is to produce an input prompt such that it bypasses both the filters to generate an output like instructions to build an explosive device.[44]

There are different methods of conducting this direct prompt injection attack beyond simply entering harmful queries via English prompts, including "optimization", "side-channel", "distribution-level", and "persuasive" methods, each of which exploit different elements of the LLM interface.[45] For instance, optimization methods increase LLM susceptibility by combining prompt queries with gibberish while side-channel methods do the same by employing low-resource languages.[46]

Applying *Van Buren's* "gates up-or-down inquiry" and other reasoning to determine if a user carrying out a direct prompt injections that only retrieves information could be liable under **section 1030(a)(2)(C**), where the user does not have prior authorization to use or access the LLM whatsoever then answer is clearly yes. This would be a scenario where the "gates" are down on *Van Buren*'s reasoning, retrieving any information via prompt injection would mean they have "intentionally accessing the computer" and obtained "information from the protected computer", as an LLM that is connected to the internet or similar network would qualify as a "protected computer". Courts interpret "intentionally"

concept is not analyzed any further in the context of whether prompt injections exceed authorized access under the CFAA: Kilovaty at 27, 33-43.

[43] Zeng et al. at 2.

[44] Zeng et al. at 2.

[45] Zeng et al. at 2.

[46] Zeng et al. at 2.

under the CFAA as acting with a purpose or objective,[47] that is, a "person acts intentionally when what happens was the defendant's conscious objective".[48] But courts generally do not require proving an intention to cause any harm, defraud, or use the information in a specific way.[49] Here, it would mean the attacker has intended to access without authorization, such as where an attacker has perhaps bypassed an authentication, like a password login, even to access the system prompt itself. How courts approach intention under the CFAA also shows that good faith motivations—like an attacker who is motivated by good faith intentions to explore a system's security vulnerabilities to help and not harm—would not obviate liability in this scenario. That is, the attacker would be criminally or civilly liable regardless of their good faith motivations because they were nevertheless intentionally accessing information on a protected computer without authorization as that was their "conscious objective" in carrying out the prompt injection. Only where a good faith attacker obtained authorization to access and test the system to begin with—as with full consent red teaming exercises—would liability no longer be a concern.

The analysis changes when a user already has authorization to access the LLM—as with most pay-to-play or publically accessible generative AI platforms like ChatGPT and Bard—then there is far more uncertainty as to whether a user carrying out a prompt injection "exceeds authorized access". Specifically, the Court in *Van Buren* delimits authorized access--and thus CFAA liability--based on a user accessing the "other areas of the computer system." In this case, where the computer system is the LLM, or at least the API to access it, what constitutes the other "areas"? This question is important, as it essentially will determine exceeding authorized access and thus liability. For example, if a user has access to the input prompt, and uses a prompt injection attack to bypass prior instructions or safety filters to retrieve confidential training data that the model produces in response to the prompts, what area has the user accessed? If the model is an "area" of the LLM system that is separate and apart from the input prompt, then this would attract liability under the CFAA as the user has exceeded their authorized access on *Van Buren*'s approach, by using their access to one area (input prompt) to access another (underlying model). So, what are the different "areas" of an LLM system? Would a court treat the input prompt as one "area" and the systemprompt as well as the underlying LLM as separate and different areas?

Unfortunately, *Van Buren* is ultimately unclear on this count. The Court offers simple examples from typical personal computers—accessing "files, folders, or databases" in other "areas within the [computer] system" but they provide little guidance on more complex systems. There are passages in *Van Buren* where the

[47] Pulte Homes, Inc. v. Laborers' International Union of North America, 648 F.3d 295, 302-303 (6th Cir. 2011).

[48] United States v. Carlson, 209 Fed. Appx. 181, 185 (3d Cir. 2006); KOSSEFF at 189-190.

[49] *Id.*

Court's reasoning suggests almost a kind of physicality--accessing another area means to "enter" that area; like where the Court cites *US v. Valle*[50] for the proposition that a user "exceeds" the "parameters of authorized access by entering an area of the computer to which [that] authorization does not extend". This suggests an analogy to breaking and entering that the 9th Circuit talks about in *HiQ v LinkedIn*[51] and implicitly in another case, *Nosal I*[52]. If exceeding authorized access requires "entering" another part of the LLM, then using prompts to get the LLM to disclose private training data is not necessarily exceeding authorized access, as the mechanism and system accessed are the same.

Applying this angle of view on "areas" of a computer to an LLM and the attacks we are considering, it would mean that just entering prompts into an area that attacker already has authorized access to (the input prompt) would not be accessing other "areas" of the computer, and thus would not constitute exceeding authorized access. That is because the attacker has not bypassed a gate or barrier and entered into another area where they now have access to new information. Instead, it would be analogous to the thief listening through the door and hearing confidential information, or perhaps even better, the thief speaks through a closed door and fools the victim in the house into telling them confidential information. Social engineering, used by hackers in other contexts, is another analogy we could use here.[53] Although there may be legal risks associated with it, they rarely come from the CFAA, at least post *Van Buren*.

However, there are other passages in *Van Buren* that read differently, where the Court says an individual "exceeds authorized access" when they "with authorization but then obtain information located in particular areas of the computer—such as files, folders, or databases —that are off limits to [them]".[54] This framing has no implication that the attacker needs to "enter" another area in order to access it. Rather, they exceed authorized access if they obtain information in "particular areas of the computer" that are "off limits," even if they do so via areas of the computer that are accessible. That said, "off limits" can mean many things. It could mean a more robust code-barrier or authentication gate (before you can pass, you need to pass some authentication requirement) but it could also mean bypassing a terms of service restriction or a simple safety filter. On this view, just obtaining information that you should not have access to due to previous instructions or content filters exceeds authorized access. Thus, liability would follow.

Overall, if a user has no access to the LLM, then a direct prompt injection retrieving only information would almost surely lead to liability under **section**

[50] US v. Valle, 807 F. 3d 508 (2d Cir. 2015).

[51] HiQ Labs, Inc. v. LinkedIn Corporation, 31 F.4th 1180 (9th Cir. 2022).

[52] Nosal v. United States, 844 F. 3d 1024 (9th Cir. 2016).

[53] Zeng et al. at 2.

[54] *Van Buren*, 141 S. Ct. at 1662.

1030(a)(2)(C). However, if a user already has access to the LLM's input prompt, then the answer is far less clear. On *Van Buren*'s reasoning whether a direct prompt injection that only retrieves information constitutes exceeding authorized access, leading to liability under section 1030(a)(2)(C), depends on broader questions of how to theorize code-based barriers and restrictions and the generative AI system itself. At the very least, there is a plausible interpretation that it does, though that conclusion is not definitive.

By contrast, liability under **section 1030(a)(5)(A)** for such direct prompt injections is far less likely, unless prompt injection causes damage to a system. This section has been applied to attacks that send malicious code or commands over the internet or another communications medium and disrupt the normal function of the receiving system—transmission of computer trojans and viruses. But it has also been applied to much more mundane transmissions, such as mass emails that can overwhelm a system, and some litigants have even argued that it should apply to certain forms of vulnerability disclosure.[55] Violating **section 1030(a)(5)(A)** does not depend upon the questions of "areas" of an LLM system that the previous CFAA claim turns on, so there is less uncertainty on that count. And prompt injection seems to fall within the plain language definition of the transmission of a program, information, code, or command, which one court has defined as "something that can be sent to a computer to make it do something."[56]

Instead, the question of "damage" is far more central. The CFAA defines "damage" broadly to include "any impairment to the integrity or availability of data, a program, a system, or information". That is a sweeping scope, but courts have also placed important limits on the concept of damage too. Courts generally distinguish "damage" from confidentiality leakages or mere access to information.[57] As such, access to or disclosure of information on remote servers is

[55] Pulte Homes, Inc. v. Laborers' International Union of North America, 648 F.3d 295 (6th Cir. 2011).

[56] *See* United States v. Singla, No. 1:21-CR-228-MLB, 2023 WL 5938082, at *6 (N.D. Ga. Sept. 12, 2023) (citing the dictionary definitions of program, information, code and command). *But see* Int'l Airport Centers, L.L.C. v. Citrin, 440 F.3d 418, 419 (7th Cir. 2006) ("Citrin argues that merely erasing a file from a computer is not a "transmission." Pressing a delete or erase key in fact transmits a command, but it might be stretching the statute too far (especially since it provides criminal as well as civil sanctions for its violation) to consider any typing on a computer keyboard to be a form of "transmission" just because it transmits a command to the computer."); MBTA v. Anderson, *Motion Hearing*, 1:08-cv-11364-GAO, Dckt # 61 at 60-61 (D. Mass. Aug. 26, 2008) (Judge O'Toole: "I note also that the word 'information' relied on by the plaintiff is used in association with the words 'program code' and 'command' which tend to be more technical terms, suggesting that information is an entity of the same order of information as codes, commands and programs.").

[57] In *Garelli Wong & Associates, Inc. v. Nichols*, 551 F.Supp.2d 704 (N. D. Ill. 2008), an employee of a placement agency for accountants who had signed a non-disclosure agreement copied certain confidential information before he left the company. In concluding that CFAA liability does not arise merely by copying data, the court, relying

generally not understood in the Seventh Circuit as "damage" within the meaning of the statute.[58]

Therefore, access to metaprompt information or private training data is unlikely constitute damage to a computing system, as even if it compromises confidentiality, it does not harm the integrity or availability of the computing system. Accordingly, **1030(a)(5)(A)**-based claims are unlikely to be viable for prompt injection that merely allows access to restricted information.

Lastly, there is also potential liability under **sections 1030(a)(5)(B)** and **(C)**, which prohibit intentionally accessing protected computers without authorization and recklessly causing damage (**section 1030(a)(5)(B)**) or causing "damage" or "loss" (**section 1030(a)(5)(C)**). On the latter provisions, the CFAA also defines "loss" very broadly to mean "any reasonable cost to any victim, including the cost of responding to an offense, conducting a damage assessment, and restoring the data, program, system, or information to its condition prior to the offense, and any revenue lost, cost incurred, or other consequential damages incurred because of interruption of service".

These provisions, unlike **section 1030(a)(5)(A),** concern intentional access, not transmitting malicious code. But unlike **section 1030(a)(2)(c)** their prohibitions on access are more narrow—there is only liability for accessing without authorization and not exceeding authorized access. For instance, in *Pulte Homes*, while the court found sufficient "damage" to the plaintiff's network for liability under **section 1030(a)(5)(A)** due to disruptions in the network due to the defendant union's email campaign, it did not find sufficient allegations that the defendant union had *no* authorization to email the plaintiffs and thus access their servers this way, so it dismissed claims under **sections 1030(a)(5)(B)** and **(C).**[59]

upon and quoting from *ResDev, LLC. v. Lot Builders Association Inc.*, No. 6:04-CV-1374, 2005 WL 1924743, at * 5 n. 3 (M.D. Fla. Aug. 10, 2005), decided "that the CFAA's use of word 'integrity' to define damage required 'some diminution in the completeness or usability of data or information on a computer system.'" *See* Condux Int'l, Inc. v. Haugum, No. CIV 08-4824 ADM/JSM, 2008 WL 5244818, at *8 (D. Minn. Dec. 15, 2008) (absent allegations that employee diminished useability of computer information obtained, no CFAA damage occurred, even if employee's activities compromised confidentiality of proprietary information accessed); Consulting Pro. Res., Inc. v. Concise Techs. LLC, No. CIV.A. 09-1201, 2010 WL 1337723, at *7 (W.D. Pa. Mar. 9, 2010), *report and recommendation adopted*, No. CIV.A. 09-1201, 2010 WL 1337720 (W.D. Pa. Mar. 31, 2010).

[58] *Landmark Credit Union v. Doberstein*, 746 F.Supp.2d 990, 993 (E.D. Wis. 2010) ("[S]eemingly every court in this circuit that has interpreted the meaning of the word 'damage' in the CFAA has held that 'damage' does not encompass harm from the mere disclosure of information" (internal quotation marks omitted) (collecting cases)).

[59] Pulte Homes, Inc. v. Laborers' International Union of North America, 648 F.3d 295, 304 (6th Cir. 2011).

Direct prompt injections attacks that only retrieve information are unlikely to attract liability under **1030(a)(5)(B)**) for the same reasons that claims under **section 1030(a)(5)(A)** would fail—retrieving information alone would not constitute "damage" under this section. However, liability under **section 1030(a)(5)(C)** is possible. If an attacker accesses without authorization the input prompt and retrieves information like a private training data set via a direct prompt injection and the LLM system developers had to incur costs to respond to the disclosure—like putting in place additional security measures or reimbursing people whose sensitive personal information was disclosed in the retrieved data set–then liability under **section 1030(a)(5)(C)** could be made out.

(b) Indirect Prompt Injections that Produce/Retrieve Information

With direct prompt injections, the malicious user is interacting with the generative AI system via an input prompt, and enters the adversarial prompts directly.[60] For *indirect prompt injection*s, the attacker's intent is the same–to bypass safety instructions, content filters, or otherwise manipulate the AI to behave in unintended ways–except the method or technique for delivering the adversarial prompt is done indirectly.[61] For example, the attacker embeds malicious code with an adversarial prompt in an artifact such as a website or email.[62] Then a benign user, using an LLM application or integrated tool like a browser plug-in, has the LLM retrieve or act on the artifact, not knowing the artifact has this malicious code that provides an adversarial prompt.[63] The adversarial prompt, delivered to the model indirectly, then causes the LLM to act in unexpected and unintended ways. Such indirect prompt injection attacks will become increasingly common as generative AI and LLMs are increasingly integrated with other software platforms and interfaces, like Bing-Chat.[64] These integrations create an entirely new attack vector for generative AI.

For an indirect prompt injection attack that only retrieves information, in addition to challenges for liability noted in our earlier analysis, both *intention* and *who* is doing the accessing without authorization or transmitting the "program, information, code, or command" becomes an additional issue for liability on either **sections 1030(a)(2)** or **1030(a)(5).** For liability under **section 1030(a)(2)**, it would not be the attacker but a benign third party that is doing the accessing or exceeding authorized access, and doing so unintentionally–they have merely been

---

[60] Kai Greshake et al., *Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection*, *in* PROCEEDINGS OF THE 16TH ACM WORKSHOP ON ARTIFICIAL INTELLIGENCE AND SECURITY 79 (2023), https://dl.acm.org/doi/10.1145/3605764.3623985.

[61] Greshake et al. at 1-2.

[62] Greshake et al. at 1-2.

[63] Rossi et al. at 1-2.

[64] Rossi et al. at 2.

duped into accessing the adversarial prompt located on the website or in a document or image. That would limit liability. The same point applies to **sections 1030(a)(5)(B)** or **(C)**, which require intentional access without authorization causing damage or loss for liability. Since intent and unauthorized access for indirect prompt injections are at the very least unclear, liability is much less likely under these sections.

Indirect prompt injections also raise questions of intent under **section 1030(a)(5)(A)** but there is no need to explore the point here, as an indirect attack that only retrieves information does not violate the section for the same reason that direct prompt injections do not: retrieving even confidential information does not constitute "damage" under the CFAA. Issues concerning intent and conduct for indirect prompt injections under this section are more material to versions of these indirect attacks that harm or damage the targeted generative AI system. Those are discussed later.

(c) Direct Prompt Injections that Repurpose, Poison, Degrade, or Harm

These attacks involve direct prompt injections wherein the adversarial prompt does more than retrieve informationI, but instead affects the underlying model in some way. That is, the attack repurposes, manipulates, poisons, degrades, or otherwise harms the LLM system in some way. For example, the direct injection may cause the generative AI to produce an "undesired output" or trick it "into running a malicious prompt"; "trick" the model into "providing misinformation" or "produce biased outputs".[65] These unintended, repurposed, or manipulated behaviors affecting the model have been shown to also potentially impact not only the adversary's direct interactions with the generative AI, but also affect model outputs and interactions for countless–even millions–of other users.[66] Recently, new forms of memory attacks on LLM systems carried out via prompt injections could also degrade or harm the generative AI in other ways, such as disrupting the service (akin to a denial of service attack).[67] These attacks could be carried out on other types of generative AI platforms–like image generators–beyond LLMs.[68]

Liability for such direct prompt injections under **section 1030(a)(2)(C)** would require some kind of information to be retrieved as a result of the attack, in

[65] Rossi et al. at 11.

[66] Greshake et al. at 1-2.

[67] Wunderwuzzi, *Sorry, ChatGPT Is Under Maintenance: Persistent Denial of Service through Prompt Injection and Memory Attacks*, EMBRACE THE RED (2024), https://embracethered.com/blog/posts/2024/chatgpt-persistent-denial-of-service/ (last visited Jul 20, 2024).

[68] Wunderwuzzi, *Google AI Studio Data Exfiltration via Prompt Injection - Possible Regression and Fix*, EMBRACE THE RED (2024), https://embracethered.com/blog/posts/2024/google-aistudio-mass-data-exfil/ (last visited Jul 20, 2024).

addition to the other harms or manipulations. This information requirement has been interpreted broadly by courts[69] For example, the "information" obtained need not even be removed or exfiltrated from the system; rather, simply observing data is likely sufficient to meet this requirement.[70] Still, any claim or criminal charge based on section 1030(a)(2)(C) liability would have the same limitations and uncertainties as discussed earlier–for users that already have authorized access to the generative AI platform or application, it is unclear what constitutes accessing other "areas" of the platform or LLM, so as to be exceeding authorized access. So liability here is possible but unclear.

For **section 1030(a)(5)(A)**, again, these direct prompt injections fall within the plain meaning of the terms–the adversary would be knowingly causing the transmission of malicious codes or commands by entering the prompt injections to a "protected computer"--a generative AI that is connected online. So, whether there is "damage" to the system is again a central consideration, which is, as noted earlier, defined broadly under the CFAA. For example, the federal court in *TriTeq Lock* interpreted the CFAA's definition of "damage" to mean the "the destruction, corruption, or deletion of electronic files, the physical destruction of a hard drive, or any diminution in the completeness or usability of the data on a computer system".[71] However, damage is even broader than that. Courts have held that *any* degradation or disruption to a system or network is "damage" within the meaning of that definition, including denial of service, even temporary ones, like delays in receiving emails or other transmissions over a network. The Sixth Circuit in *Pulte Homes*, for instance, interpreted "damage" this way:

> Because the statute includes no definition for three key terms — "impairment," "integrity," and "availability" — we look to the ordinary meanings of these words. *United States v. Plavcak,* 411 F.3d 655, 660-61 (6th Cir. 2005). "Impairment" means a "deterioration" or an "injurious lessening or weakening." 7 Oxford English Dictionary 696 (2d ed.1989) [hereinafter OED]. The definition of "integrity" includes an "uncorrupted condition," an "original perfect state," and "soundness." *Id.* at 1066. And "availability" is the "capability of being employed or made use of." 1 OED, *supra*, at 812. Applying these ordinary usages, we conclude that a transmission that weakens a sound computer system — or, similarly, one that diminishes a plaintiff's ability to use data or a system — causes damage.[72]

[69] KOSSEFF at 187-188.

[70] KOSSEFF at 187-188.

[71] TriTeq Lock & Sec. LLC v. Innovative Secured Solutions, LLC, No. 10 CV 1304, 2012 WL 394229, at *6 (N.D. Ill. Feb. 1, 2012).

[72] *Pulte Homes* at 301-302.

The court went on to find that the union's email campaign *did* cause damage because it disrupted the company's operations and prevented it disrupted the plaintiff's network: "the transmissions diminished its ability to send and receive calls and emails, it accordingly alleges an impairment to the integrity or availability of its data and systems—i.e., statutory damage".[73] Another example of that form of "damage" is in *United States v. Mitra*, wherein the court upheld the defendant's transmission conviction because he impaired the availability of an emergency communication system when "[d]ata that [he] sent interfered with the way the computer allocated communications to the other 19 [radio] channels and stopped the flow of information among public-safety officers."[74]

Based on these broad definitions of "damage" any direct prompt injection attack that disrupts the generative AI–like an attack that causes a denial or disruption of the AI's operations or service for any other user would constitute "damage", leading to liability. Any attack that poisons, corrupts, or degrades the generative AI's model, such that its unintended behavior is not temporary, but more permanent or impacts any other benign users, such as prompt injection attacks that cause the model to produce misinformation or biased outputs for other benign users, is more likely to meet this "damage" requirement s. An attack that repurposes the AI to carry out new or different tasks, which were unintended or expected by the AI developers, where that repurposing also affects other benign users, could potentially also constitute "damage" under the CFAA.

However, there is another uncertainty here with more sophisticated AIs, including LLMs. These systems have the capacity to learn new tasks or exhibit what researchers call emergent behaviors that themselves were not intended, foreseen, or expected by the AI developers, and are thus difficult to predict.[75] For example, it may be very difficult to determine whether unintended behaviors exhibited by an LLM more permanently were caused by the prompt injection and not some emergent behavior that the AI itself developed through its own machine learning processes. This kind of uncertainty impacts other areas of the law–like foreseeability in negligence law–and similarly would complicate findings of criminal or civil liability for these attacks under the CFAA.[76]

If a direct prompt injection attack constitutes "damage" for the purpose of **section 1030(a)(5)(A),** it would likewise suffice for liability under sections **1030(a)(5)(B)** and **(C)**. For these sections, however, the attacker would have to have no authorized access to the generative AI whatsoever for liability. There is also the further possibility under **section 1030(a)(5)(C)** that even where any

[73] *Pulte Homes* at 301-302.

[74] 405 F.3d 492, 494 (7th Cir. 2005).

[75] Ryan Calo, *Robotics and the Lessons of Cyberlaw*, 103 *Calif. L. Rev.* 513, 515, 541–543 (2015).

[76] *Id.* at 554-555.

impact of the attack does not meet the broad “damage” requirement, any costs incurred to respond to the attack could be sought as a “loss” under this section.

(d) Indirect Prompt Injections that Repurpose, Poison, Degrade, or Harm

These indirect prompt injections, like those noted earlier, impact the generative AI in other ways–repurposing, degrading, or harming–but the prompt injections delivered indirectly, by maliciously altering, or degrading an artifact, that the AI or LLM retrieves or accesses, executing the adversarial prompt, and causing the harm. The harms here include those noted earlier for similar direct prompt injections, including recently demonstrated indirect attacks that could corrupt model memory functions, creating more longer term impacts, including disruption or denial of service or other manipulated behavior and outputs.[77]

However, an important difference with these emerging forms of prompt injection attacks is that because these attacks target artifacts that could in theory be retrieved by the generative AI for model training purposes, any impacts caused by the adversarial prompts embedded therein could have more lasting impacts on the AI, thus damaging or degrading the system permanently and affecting other users in more lasting ways. For instance, a recent research demonstrated new forms of “poisoning attacks” that poisons or corrupts popular deep learning training data sets with code that introduces malicious examples into the models’ performance, thus damaging or degrading the model and its performance more permanently.[78] These poisoning attacks are passive and easy to carry out at scale–researchers predict they could immediately poison 10 popular training datasets online.[79] These attacks could, in theory, employ adversarial prompt injections to poison the training data, for other lasting model impacts.[80]

Since all forms of deep learning models–including LLMs and other generative AI systems–are typically trained on widely distributed web-scale datasets crawled from the internet,[81] the potential impacts and harms for all types of generative AI systems–text, image, and beyond–and implications for both their developers and users is significant.

[77] Wunderwuzzi, *Sorry, ChatGPT Is Under Maintenance: Persistent Denial of Service through Prompt Injection and Memory Attacks*, EMBRACE THE RED (2024), https://embracethered.com/blog/posts/2024/chatgpt-persistent-denial-of-service/ (last visited Jul 20, 2024).

[78] Nicholas Carlini et al., *Poisoning Web-Scale Training Datasets Is Practical*, (2024), http://arxiv.org/abs/2302.10149 (last visited Jul 20, 2024).

[79] Carlini et al. at 1.

[80] Carlini et al. at 1.

[81] Carlini et al. at 1.

Turning to liability, as with earlier analyzed indirect injection attacks, the indirect methodology creates intent and conduct issues for liability under **section 1030(a)(2)** and the different subsections in **1030(a)(5)**. Again, for **section 1030(a)(2)**, depending on the methodology of the indirect attack, it would be a benign third party in some cases that ends up unintentionally accessing or exceeding authorized access by having the generative AI act on the corrupted file or dataset. In the case of training dataset poisoning attacks, wherein the attacker knows that the generative AI will be trained on the corrupted dataset embedded with malicious prompts, it could be argued that the attacker is acting intentionally–with conscious objective or purpose–to access the model without authorization, even if that access is carried out via a benign third party, or the model itself, during model training. On this angle of view, the benign user becomes simply a tool or conduit in delivering the adversarial prompts, providing the unauthorized access for the attacker via the code and prompts. But this is a novel theory of liability and it is uncertain a court would accept it. The same can be said for intent requirements under **sections 1030(a)(5)(B)** or **(C)** for these attacks.

Indirect attacks like this also raise complicated intent problems under **sections 1030(a)(5)(A)**. That section has two sets of intent standards—"knowingly" causing transmission of commands or code, and as a result, "intentionally cause[ing] damage" to a protected computer. Courts have found that in placing a cookie on the plaintiff's computer, which allowed the defendant to retrieve information from it, the defendant "knowingly caused the transmission of a program, information, code or command" and that this damage was intentional.[82] In another case, the court found that a defendant deleting data from the plaintiff's computer also met the "knowingly caused" intent standard for transmission and that the resulting damage (deleted data that was unrecoverable) was also intentionally caused.[83]

Nevertheless, interpreting these intent standards for an indirect prompt injection is not straightforward. While malicious code placed on a website, email, or in training data is analogous to placing a cookie on a computer, the benign third party is still the party that arguably transmits the malicious code. So this would be a reason to limit liability for these attacks.

As with direct injection attacks causing harms beyond mere information retrieval, whether the impacts or harms caused by the indirect injections constitute "damage" for the purposes of liability under **sections 1030(a)(5)(A), (B)**, and **(C)** remains a central issue. The same analysis and uncertainties apply here. Any attack that degrades the generative AI performance beyond temporary or ephemeral impacts, and any attack that disrupts the system's operations–including

[82] *In re* Intuit Priv. Litig., 138 F. Supp. 2d 1272, 1280 (C.D. Cal. 2001).

[83] ATS Grp., LLC v. Legacy Tank & Indus. Servs. LLC, 407 F. Supp. 3d 1186, 1191 (W.D. Okla. 2019).

even temporary disruptions (*Pulte Homes*)--could constitute "damage" under the CFAA, especially where impacts are more permanent and impacts model or AI performance for other benign users. And, as above, a further possibility under **section 1030(a)(5)(C)** that costs incurred to respond to the attack could constitute a "loss" for liability under this subsection.

III. First Amendment Defense

Finally, we look at how the First Amendment might constrain liability for prompt injection. Although First Amendment challenges to the CFAA and similar laws that limit the sharing of information because it might cause technological harm (e.g., **17 U.S.C. § 1201**) have not been particularly successful, the potential expansion of the scope of **1030(a)(5)(A)** claims might cause a court to think twice about an interpretation that creates broad liability for prompt injection. Although the type of language used for an attack should not make much of a difference, courts may be more inclined to credit First Amendment arguments when the speech in question is English rather than a programming language.

As a starting point, source code is protected by the First Amendment.[84] Nevertheless, the functional capabilities of speech are considered when engaging in First Amendment analysis.[85] This can mean that practically, computer code's more functional aspects allow for more significant restrictions on it than would be constitutional for pure expressive speech.[86]

However, there are arguments that prompt injection instructions may not qualify for First Amendment protection at all, especially in circumstances where a user knows that they will produce a certain result from a LLM. In *Universal City Studios v. Corley*, the Second Circuit distinguished between instructions for a computer that users were to follow mechanically, which it claimed lacked constitutional protection, and software commands that were sources of information and advice.[87] Likewise, some First Amendment arguments

[84] Junger v. Daley, 209 F.3d 481, 485 (6th Cir. 2000) ("Because computer source code is an expressive means for the exchange of information and ideas about computer programming, we hold that it is protected by the First Amendment."); Universal City Studios, Inc. v. Reimerdes, 111 F. Supp. 2d 294, 327 (S.D.N.Y. 2000) ("As computer code . . . is a means of expressing ideas, the First Amendment must be considered before its dissemination may be prohibited or regulated.").

[85] *See* Junger at 485.

[86] *See also* Universal City Studios, Inc. v. Corley, 273 F.3d 429, 451 (2d Cir. 2001) ("These realities of what code is and what its normal functions are require a First Amendment analysis that treats code as combining nonspeech and speech elements, i.e., functional and expressive elements.").

[87] Universal City Studios, Inc. v. Corley, 273 F.3d 429, 449 (2d Cir. 2001) ("Essential to our ruling in Vartuli was the manner in which the defendants marketed the software and intended that it be used: the defendants told users of the software to follow the software's cues "with no second-guessing," id., and intended that users follow Recurrence's

distinguish source code from other forms of speech based on its functionality and impact/harm in the world.[88] A court looking to sidestep First Amendment restrictions on the CFAA's application to prompt injection might turn to this distinction to find that a prompt for a LLM is functional in the same way that computer code is, thus avoiding or limiting the need to engage in constitutional avoidance by narrowing the CFAA's scope.

There is unfortunately little case law to turn to see how courts would handle a CFAA claim based on an English language prompt. The closest analogous situation arose in *MBTA v. Anderson*, a 2013 case where the Massachusetts transit authority argued that MIT students giving a DEFCON talk about how to ride the subway for free would violate **section 1030(a)(5)(A).** The MBTA believed that the talk would constitute the transmission of a program, information, code, or command that could cause damage to a protected computer. The Electronic Frontier Foundation, representing the students, argued that such a reading of **section 1030(a)(5)(A)** brought it into conflict with the First Amendment. In a hearing that occurred immediately before the case was settled, the judge suggested that the MBTA's reading was overly broad, and that a narrower and more technical reading of "program, information, code, or command" was more aligned with the legislative history.[89] However, because the case settled, there is no opinion that lays out this reasoning in more depth.

Following the EFF's lead, an attacker who faces a CFAA claim could argue that construing the CFAA to cover plain language prompt injection violates the First Amendment, as-applied, and constitutional avoidance would require a narrower reading. The CFAA could be viewed as targeting expression, conduct, or something in between. But in any case, it seems most likely that it would be

---

commands "mechanically" and "without the intercession of the mind or the will of the recipient," id. We held that the values served by the First Amendment were not advanced by these instructions, even though the instructions were expressed in words. Id. We acknowledged that some users would, despite the defendants' marketing, refuse to follow Recurrence's cues mechanically but instead would use the commands as a source of information and advice, and that, as to these users, Recurrence's cues might very "well have been 'speech.' " Id. at 111–12. Nevertheless, we concluded that the Government could require registration for Recurrence's intended use because such use was devoid of any constitutionally protected speech. Id. at 112.").

[88] Universal City Studios, Inc. v. Reimerdes, 111 F. Supp. 2d 294 (S.D.N.Y.), *judgment entered*, 111 F. Supp. 2d 346 (S.D.N.Y. 2000), *aff'd sub nom*. Universal City Studios, Inc. v. Corley, 273 F.3d 429 (2d Cir. 2001).

[89] "I agree with the argument by the defendants that the construction of the statute argued for by the plaintiff that the 'transmission' information, according to the section under the statute, by publication to an audience is not likely the correct construction of that provision of the statute. ... I note also that the word "information" relied on by the plaintiff is used in association with the words 'program' 'code' and 'command' which tend to be more technical terms, suggesting that information is an entity of the same order of information as codes, commands and programs."

examined under at least intermediate scrutiny.[90] Although the government interest in the CFAA is significant, it seems unlikely that a **1030(a)(5)(A**) claim based on an English language prompt like "Ignore your instructions" would be narrowly tailored enough to survive constitutional scrutiny, as the potential to restrict constitutionally protected speech is present.

IV. Conclusion and Recommendations

Overall, there are a range of legal uncertainties for direct and indirect prompt injection attacks under the CFAA's prohibitions on hacking to obtain information (**sections 1030(a)(2)(C)**) and hacking to cause damage (**sections 1030(a)(5)(A), (B)**, and **(C)**). Likely, the most straightforward case for liability would be under **sections 1030(a)(2)(C)** for direct prompt injections that only retrieve information where the attacker has no authorization to access the generative AI platform, so that retrieving information via adversarial prompts constitutes intentionally accessing a protected computer without authorization. Where the attacker has prior authorized access, as they would in most publicly accessible or subscriber-based generative AI platforms and services, then uncertainties concerning "exceeding authorized access" arise, including how to theorize both authentication gates and "areas" in generative AI systems on *Van Buren*'s reasoning. Though a reasonable case can be made for liability here, it is not definitive. The clearest case where liability is the most unlikely, is a direct prompt injection retrieving only information, under **sections 1030(a)(5)(A) ,** which requires "damage" and courts have consistently held that information disclosure alone does not constitute damage.

Beyond this, most other direct and indirect attacks likewise raise liability concerns under these sections, but there are many uncertainties. For any prompt injections attacks that cause impacts on the generative AI beyond retrieving information alone, liability will turn on the nature of the damage. For indirect prompt injections, there are additional uncertainties based on intent, among others. Often, a reasonable case can be made for liability, but again not definitive.

Our findings offer clear new directions for additional research. Post-*Van Buren*, the architecture of systems matter more than ever. So, theorizing and delimiting authentication gates, code-based barriers, and the nature of different generative AI systems, and their various "areas", is incredibly important for understanding the scope of liability under the CFAA. This might include, for instance, exploring the nature of an LLM. If an LLM is merely a more powerful database,[91] then there may not be different "areas" that delimit access but just the database, and the data therein. This theory of an LLM would thus increase the scope of liability. Additional research and investigations on how, exactly, various prompt injections

[90] *See, e.g.,* Universal City Studios, Inc. v. Corley, 273 F.3d 429, 450 (2d Cir. 2001), Green v. DOJ, 21-5195 (D.C. Cir. 2022).

[91] François Chollet, *How I think about LLM prompt engineering - Prompting as searching through a space of vector programs,* SPARKS THE WIND (Oct. 9, 2023), https://fchollet.substack.com/p/how-i-think-about-llm-prompt-engineering.

harm or impact generative AI systems both temporarily and in the long term would also be helpful in determining questions of "damage" for liability under the CFAA.

Lastly, given the uncertainties but also clear potential for liability under these CFAA provisions for security researchers testing, evaluating, and red teaming systems, on top of the long history of legal threats and claims both by companies against security researchers and federal prosecutions targeting them, we would similarly support legal protections for good faith researchers. Such protections could include voluntary corporate safe harbors but also more permanent statutory reforms that would narrow the sweeping scope of these CFAA prohibitions that are often, as they are in the case of prompt injections, very unclear. That is the best way to avoid chilling effects on security research that is more important than ever today.