Artificial Intelligence Within Agile Software Development: Projected Impacts to Cyber Offense-Defense Balance

Timothy Kokotajlo, Dr. David Long, Dr. Mark Reith
David.long@afit.edu, Mark.Reith@afit.edu

Abstract: The cyber offense-defense balance theory evaluates the relative effort offense and defense forces need to prevail in conflict. Cyber conflict is a large domain, and the offense-defense balance is influenced by many factors. Recent increased investments into Artificial Intelligence (AI) and Agile development can be used to make more accurate predictions using existing cyber offense-defense models. This paper predicts future shifts in the balance of cyber warfare caused by current and projected capabilities of AI-augmented Agile software development pipelines. Initially, increased investments and capabilities are predicted to give an advantage to aggressing forces. Over time, however, technology advances and more standardized processes will likely create a much greater advantage for defensive capabilities.

Keywords: Artificial Intelligence (AI), Agile Software Development, Offense-Defense Theory, Cyber Warfare

1. Introduction

Software development is critical to cyber warfare, not only just for securing code, but also to quickly produce tools (National Security Council, 2021). Artificial Intelligence (AI) has increasingly become more ubiquitous within modern Agile software development, including DevSecOps and more holistic views of developing, maintaining, operating, and securing software (Cearley, 2017). This conceptual paper uses modern offense-defense theory to argue that AI implementations within Agile software development cycles will initially increase offensive power, yet ultimately prove to be more useful in cyber defense than attack. Change management has proven to be a weak link for cyber security, and autonomous systems in particular have shown to create false senses of security during early adoption (Morales, 2020). While organizations work both on the technology and culture of AI in Agile, offensive cyber actions will be easier. Many of the advantages AI systems bring to software development can be summarized as providing increased return on investment (Ilieva, 2019). With AI systems, teams can potentially produce more and better software with the same or with less funding. Current scaling models of cyber offense-defense show that at high investment levels the balance is in favor of defenders (Garfinkel, 2019). Thus, as AI systems make higher investment into software development more affordable, cyber defenders' advantage grows.

The first section of this paper provides an overview of offense-defense theory and discusses applicable offense-defense models. It outlines some of the most relevant research, particularly the existing scaling impacts within cyber conflict. Next, the paper outlines the current and projected impacts AI has on software development. For organizations adopting more Agile mindsets, AI technologies give two main cyber benefits: faster delivery of code, and greater total investment capability. Lastly, this paper ties the two together, drawing conclusions about how the offense-defense balance will shift due to the observed and predicted usages of AI.

An increase in AI software development technologies will allow for more total investment into software utilized for both offense and defense. If the applied scaling model holds, this gives an advantage to defending forces. In addition, software pipelines building offensive cyber tools likely will not benefit as much from the increased production speed. However, in the short-term organizations should expect and prepare for a period of increased vulnerability where the offensive benefits granted from new and poorly understood technology adoption outweigh those defensive benefits.

This paper specifically looks at applications of AI within the development cycle. Research on the cyber warfare implications of AI within software and in the maintenance and sustainment of software has been more thoroughly studied and compiled. The conclusions drawn within the DoD's strategic assessment of "Military Investments in Autonomy and AI" also predict short-term security challenges raised by "mistrust and overtrust" of AI systems, and expect that AI programs will eventually prove to be key to cyber security (Konaev, 2020).

2. The Cyber Offense-Defense Balance

Offense-defense theory is the quantification of the relative ease of conducting an attack compared to the effort needed to defend against said attack. Traditionally, the Offense-Defensive balance is a ratio between the monetary cost of the attacking forces required to take territory to the defending forces required to repel the attack (Kaufmann, 1998). A conflict where a contested piece of territory required $1 million in manpower and equipment to defend but took $2 million to take would have a 0.5 ratio. This construct has also been successfully applied to cyber warfare; the system produced by research from the Naval Postgraduate School calculated an average offense-defense cost ratio of 2.55 (Malone, 2012). These calculations did not factor in post-attack costs and did not take scaling into consideration. If the total costs include damage to the target, the balance can be expected to tip significantly in favor of offense (Slayton, 2017).

AI technology in totality has been predicted to aid defensive cyber forces for a variety of reasons. Current AI research is being directed more towards defensive applications of AIs, such as malware investigation, network intrusion detection, and phishing identification, rather than in pursuit of offensive capabilities (Truong, 2020). As the weak link in a cyber defense strategy is often the human components, the Chief Technology Officer of IBM Resilient expects that AI technologies will "bring defense up to par with offense" (Schneier, 2018).

Despite AI technologies bringing many significant defensive benefits, several models make arguments assuming AI technology will give aggressors an advantage. AI applications currently are effective as "a potential power force multiplier and enabler for several high-tech domains – including cyberspace," but strategic planners are forced to assume AI technology will benefit offensive operations because the future capabilities of AI are unknown (Johnson, 2019). This opinion, that AI technology will first benefit aggression in cyberspace, is widely held by experts in the field. A Carbon Black survey of cybersecurity researchers reveals an extremely low level of confidence in organizations' ability to defend against AI-enabled attacks; even with AI-enabled defense (Michael Viscuso, 2017).

In 2019 Garfinkel and Dafoe proposed an offense-defense model of cyber warfare that specifically analyzes scaling effects (Garfinkel, 2019). Their model favors offense when investment levels on both offense and defense are low or middling, and gives a large advantage to defense when both become very high. Their model assumes a finite number of 'exploitable' attack vectors. Small investments do not enable defenders to discover and fix all vulnerabilities. Massive investments allow defender efforts to saturate the defensive surface area, discovering and covering all potential attack vectors. Thus, even if the offensive forces have equivalent or higher investment levels, at high levels of defender investment it is incredibly unlikely or impossible for an offensive force to discover a vulnerability that hasn't been secured against. *Figure 1* outlines this relationship, using the term 'Zero days' to describe a vulnerability that both sides are racing to discover first.

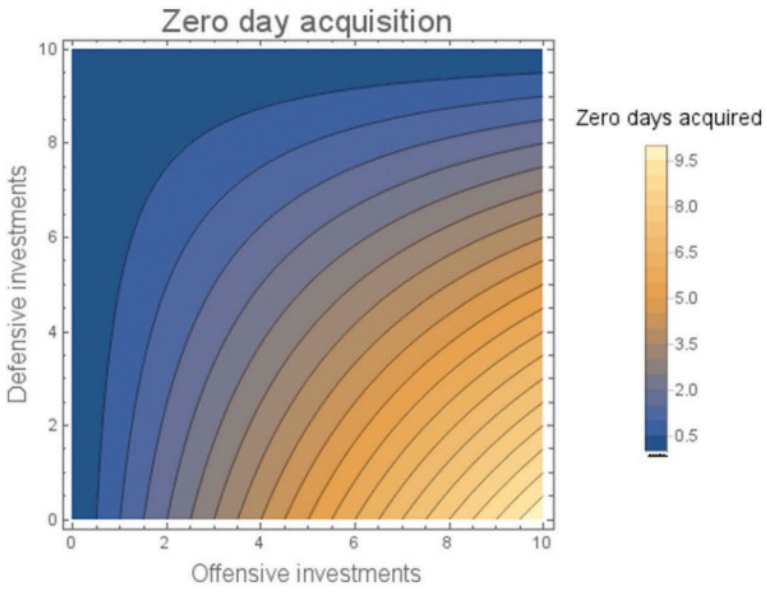

Figure 1. *Idealized model for a case with 10 zero day vulnerabilities (Garfinkel, 2019)*

The Garfinkle paper outlines 3 possible methods that more advanced AI might impact the offense-defense balance: a *new form* of conflict (like cyber-warfare), a changed *character* of conflict that necessitates a new model (automatic weapons drastically changed the value of manpower), or AI could support a massive change in the *quantity* of investments fielded by both sides (Garfinkel, 2019). Their paper argues that this third pathway is the area of greatest growth and least attention. The cyber impacts of AI-enhanced software development will likely be felt mainly as increased effective investment.

As AI technology becomes more utilized, especially within Agile development, the above relationship graph will likely stay accurate. The key question for applying the Garfinkel model is "whether or not [AI in software Development] displays certain features that are sufficient for offense-defense scaling" (Garfinkel). The critical features used by Garfinkel all are indicators of if defending forces can 'saturate' the attack surface at high investment levels.

The next section will make the argument that AI technology will act primarily as a force multiplier within cyber warfare. With this assumption, cyber principles will not change, but AI systems will simply increase the capabilities of cyber actors. Therefore, investments into software development with AI systems will give a much greater return as the technology improves, pushing both offensive and defensive cyber warfare efforts into the top right 'high investment' area of *Figure 1*. If low and moderate investment levels less likely, for both offense and defense, the average cyber offense-defense ratio would shift heavily towards defenders.

3. AI Effects on Software Development

3.1 AI in Agile Methodologies

The rise of Agile Software Development has created opportunities for AI systems to integrate into the development pipeline. Agile methods (e.g. Scrum or SAFe) break software project management down to its component parts and focus heavily on developing the *culture* of creating code almost as much as the code itself. Task lists are broken down into product backlogs, developers and planners are broken down into groups of small teams consisting of only the essential members to fully complete a given deliverable (Richard Knaster, 2018). Agile

methodologies focus heavily on transparency and quantifiable progress. As a direct effect, this produces large quantities of data both about the software, and of meta-data about how the software was produced. Current Agile practices create billions of pieces of meta-data, including bug-fix notes, code reviews, and authorship tags (Allamanis, 2018). This surge of 'Big Code' lends itself extremely well to AI, as any system running off an Artificial Intelligence model needs massive amounts of data (Kantarcioglu, 2019).

AI usages within software development can loosely be clustered into three main categories. AIs can help program management, providing analytical capabilities lacking in current toolsets. AI systems can aid developers directly, or even bypass the highly technical skill sets by acting as an intelligent translator from human ideas into working code. Lastly, AI can drastically improve the Quality Assurance and security of developed software through conducting automatic code analysis. A 2018 study on AI impact found that 75% of participants believe AI will have a "moderate to significant impact within the next two years," either directly to their organization, or to software production in general (Rowe, 2018).

3.1.1 Program management

To support Agile methodologies, many tools have been created, such as Jira, Confluence, Trello, and Kabanchi. These tools, however, limit their support to "creating, managing, and tracking project artifacts, and visualizing historical project data" and do not provide "advanced analytical methods that are capable of harvesting valuable insights from project data" (Dam, 2019). AI researcher Hoa Khanh Dam proposes that AI can find an ideal place within Agile program management with three types of roles: an Analytics engine, a Planning engine, and an Optimization engine (Dam, 2019). Adding engines such as these to software programs aids in the Agile methodology--they are designed not to investigate and improve upon the code directly, but to take a look at the processes and performance metrics of the program *creating* the code (Dam H. K., 2019). Any improvements there would necessarily improve the quality or quantity of deliverables.

3.1.2 Human/Machine translation

AI systems are being designed that can convert human language and ideas into working software. Natural language parsing is close to reaching a critical point. AI systems are not just able to read and reply to human prompts but can create novel ideas and extrapolate from simple human-provided prompts (Dale, 2021). The website 'analysticsindiamag.com' has compiled a list of 10 separate AI applications that already generate code themselves (Choudhury, 2019). The project 'debuild.co' (another program not yet commercially marketed) has published simple proof of concepts where an extremely simple user interface can generate HTML code within seconds (See *Figure 2*). Advanced versions of software such as these might create as large a difference in coding experience as modern integrated development environments are to old-school punched cards.

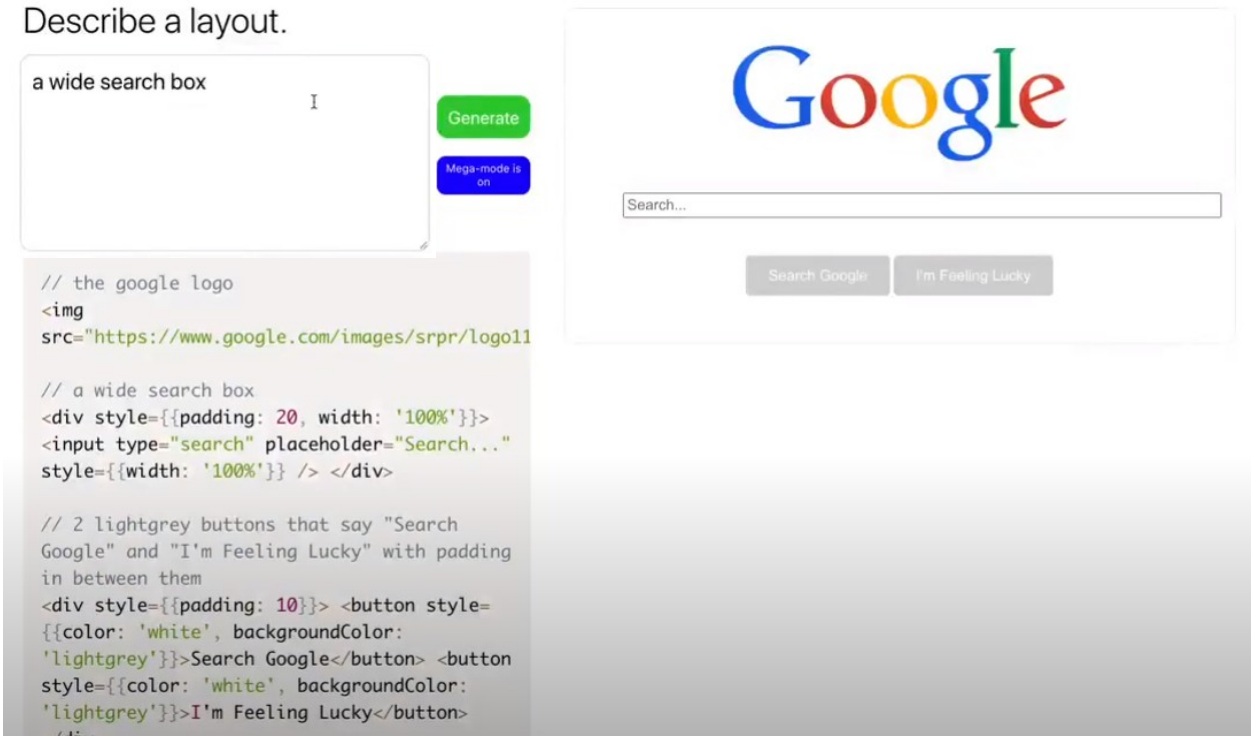

Figure 2: *User-submitted text from the top left box turns into code comments, and the AI determines what is meant by the human language, and writes the appropriate code (Shameem, twitter.com, 2020)*

3.1.3 Quality and Security Assurance

Lastly, the most popular AI application within Agile systems is for software analysis. Amazon's CodeGuru is one such example. CodeGuru is an AI program designed to be implemented within Agile pipelines and provide remediation suggestions not just for performance, efficiency issues, and Amazon Web Services (AWS) pricing recommendations (AWS, 2020). Furthermore, AIs can (and are) being used to augment the collaboration and modernization within Agile software pipelines. Recent research demonstrates how the massive amount of 'Big Code' created by Agile software development can be thrown into analysis bots to "provide an interface for developers to "subscribe" to new research techniques without needing to trust the implementation, the developer of the new tool, or to install the analysis tool locally" (Beschastnikh, 2017). AI programs bring security benefits too. AI-based algorithms can predict vulnerable code snippets much more efficiently than hard-coded rules in an Integrated Development Environment (IDE) could (Ferenc, 2019). Deep learning tools designed to secure software have been able to show significant improvements over traditional analysis approaches (Dam H. K., 2021). Right now AI systems are not able to replace a human quality or security check, but as the software gets more advanced, the benefits can only be expected to grow.

3.2 Cyber Warfare Benefits Realized by AI-enhanced Software Pipelines

As software development transitions out of the traditional 'waterfall' approach and adopts Agile methodologies, AI usage for software pipelines also increases (DoD, 2018) (Wrubel, 2014). Organizations can expect to see Agile software programs utilizing AI to have faster release times and get more return out of their investment. However, adopting new technologies, especially ones that require organizational or procedural shifts, often causes new vulnerabilities or short-term weaknesses. Automated software in particular can lead to a trust in technology that is unwarranted. To avoid security pitfalls, adopting any AI technology would require very strict change management controls (Butavicius, 2020).

Commercial entities have already realized large benefits in delivery times directly due to implementing AI within Agile pipelines. American Family Insurance Group reported a 32% decrease in false-positive vulnerability alerts, a 45% reduction in security scan time, and 57% fewer alerts for human developers; accelerating application delivery times by 24% (Schwenger, 2020). While profit margins may not always be relevant, especially for state-funded cyber operations, the timeline benefits displayed by the commercial world are still important. The Symbiotic Design for Cyber Physical Systems (SDCPS) program currently in development at the United States Defense Advanced Research Projects Agency (DARPA) seeks to use AI-human partnerships to "reduce the time from CPS inception to deployment from years to months" (Neema, 2021). Other applications expected to come under cyber attack would likely also benefit from such partnerships.

These accelerated timelines benefit the defensive abilities granted by securing software as fast as possible, but do not aid offensive abilities in the same fashion. Customized attack software also requires a software development pipeline. Current scientific research investigates avenues of improving offensive cyber operations but lacks investigations focused specifically on the performance of "software development cycles to exploit vulnerabilities in target systems" (Huskaj, 2019). However, research shows that organizations are usually extremely quick developing software exploits (See Figure 3). Vulnerabilities usually have exploits written within days or weeks of disclosure, with sophisticated attackers taking advantage even faster (Arora, 2004) (Kathleen Metrick, 2020). Adding AI assistance to offensive development teams might not cause a significant increase in release times given the current release speed.

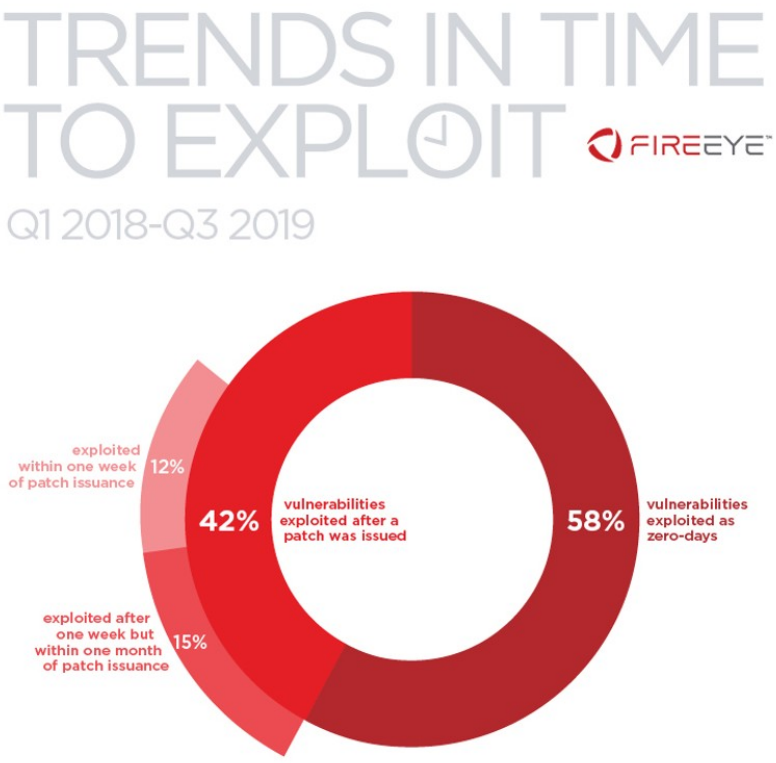

Figure 3: *Percentage of vulnerabilities exploited in relation to patch release (Kathleen Metrick, 2020)*

AI software assists more than just release times, however. AI injected into the cyber production line "enhances innovation in design" by giving engineers better performance predictability and by assisting exploration

of “unconventional but highly performant solutions” (Neema, 2021). One such example of a successful AI integration is NASA’s usage of Evolutionary Algorithms. The development team was able to create a more optimized design for spacecrafts’ antenna structure using AI than a human design team could (Hornby, 2006). While not a software development pipeline, the same general evolutionary construct has been shown to work in optimizing, streamlining, and minimizing code in pieces of government and commercial software (Fogel, 2011). With the storage space and computing power of modern technology, code minimization has been replaced by readability and modularity. However, for cyber-attacks, rudimentary AI is already being used to create minuscule and polymorphic payloads for certain types of malware (Thanh & Zelinka, 2019). Software produced with AI systems is expected to improve not just in quantity but in quality; AI systems can assist developers in both mundane chores and high-creativity endeavors. Both software teams securing software or building offensive tools can utilize advanced AI systems to outperform teams working in traditional development environments.

The code produced by organizations utilizing AI within their Agile pipelines can also be expected to be more fluid and modular; allowing greater flexibility, but may also introduce novel types of security vulnerabilities. Perfectly secure AI models can still suffer from oracle problems, where bad data can cause inherent prediction errors in the learned classification model (Xie, 2018). Perhaps worse, specific well-designed inputs to AI systems can cause radical behavior in deep neural networks. These ‘adversarial examples’ are extremely hard to predict, and might not ever have a foolproof ‘defense’ against malicious exploitation (Yuan, He, Zhu, & Li, 2019). However, the realization of AI’s utilizing Agile software’s ‘Big Code’ might enable greater standardization and regimentation of software modules for stability and reliability (Beschastnikh, 2017).

Offensive abilities are going to benefit from AI systems augmenting Agile software engineering. Offensive forces will have faster deployment times and more optimized and/or creative tools. However, defensive forces will have the same benefits; and the deployment times benefits will have an even greater effect. Where offense gains an advantage is in the migration period, when organizations are adopting AI systems into their pipeline for the first time. AI software has been shown to cause over and under-confidence in the security both of development systems and the code produced by the pipelines (Rahkovsky, 2020). Coupled with the unique security vulnerabilities AI can bring to an environment, the offense-defense balance in the short term would likely tilt towards offense.

4. Conclusions and future work

It would be poor practice to look at the current state of AI in software pipelines and draw conclusions only from current technology. AI is quite possibly the single scientific field with the *most* potential for massive changes to civilization (Bostrom, 2019) (Nowak, 2018). Projections into the future of AI technology within the field of software development point towards being able to field software more securely and rapidly, but not necessarily with more accountability. The growing pains of adopting Agile software development and AI will be more challenging than other new technologies, and will probably continue for years to come (Morales, 2020). This will likely cause defensive gaps until the proper processes for safely utilizing AI to assist software pipelines is standardized and adopted across all software providers. Organizations may see an increase in the short-term offense-defense ratio, but mitigated if they are not adopting AI within their own development pipelines.

It is not advisable, however, for an organization to not adopt AI in their development pipelines, despite the short-term drawbacks.

Over time AI’s impact upon software development will swing the pendulum of offense-defense towards the side of the defenders for two reasons. All likely benefits of AI will assist in defensive efforts, creating more effective software production lines for creating defensible software, producing responsive software faster, and enabling more optimal solutions. However, offensive software production currently displays an already rapid pace of deployment, and will therefore not benefit to the same degree as the much slower development pace of ‘defending’ software. Secondly, Agile software teams working with AI systems will have more developer capability, more effective security checks, and much faster release times. This will vastly increase the effective investment

level for both sides of the offense-defense balance. If the scaling offense-defense model outlined in the first section holds, this predicts a lower investment threshold to reach a clear defenders' advantage.

If the production of software being used for offense and defense becomes more rapid and more secure, defenders benefit more.

Purely offensive software development does not often display the extended deployment timelines that the software it targets does. Therefore, an improvement to all development timelines disproportionately benefits forces attempting to defend cyber assets. Furthermore, investment in software development creates far greater returns due to AI. Defending forces are more easily able to cover all attack vectors with high investments. In a theoretical environment, extreme investments allow defenders to uncover all possible vulnerabilities and defeat all attacks, regardless of any amount of offensive investment. AI in Agile development, acting as a force multiplier, will likely enable a shift towards a high-investment offense-defense balance. The same AI systems do have the potential of creating a period of offensive cyber advantage. However, this advantage for offense will only affect the short-term. Volatility and mistrust during the adoption of automation technologies can create weak spots an aggressor can exploit. Once AI systems have been fully adopted, both technologically and culturally, those defensive weaknesses disappear.

The current balance of cyber is difficult to model, let alone empirically test and evaluate. This work is only one small piece of the cyber offense-defense equation, and many more factors need to be at least conceptually considered. Cyber offense-defense models can be broken down into components, with individual technologies, cultures, and practices feeding into the total model. Unfortunately, there are only a few cases where cyber offense-defense models have been rigorously validated. Future studies could first attempt to quantify the current offense-defense balance of cyber in terms of controllable variables such as development environment and relevant technologies. Once parameters have been established, more research could be done identifying the weights and balances each variable brings to the total balance. To identify the actual shift to the cyber offense-defense balance caused by Ai adoption within Agile development would require a study comparing organizational environments equivalent in other cyber-relevant factors such as governmental oversight, internal security policies, type of software being produced, size of the development team, Agile methodology, and more. This could gather real data as to the actual cyber significance of AI systems for software development. Without such hard evidence, papers such as this can only ever be conceptual.

Bibliography

Allamanis, M. (2018). A survey of machine learning for big code and naturalness. *arXiv*, 36.

Arora, A. K. (2004). Impact of vulnerability disclosure and patch availability-an empirical analysis. *Third Workshop on the Economics of Information Security*, 1268-1287.

AWS. (2020). *codeguru*. Retrieved from aws.amazon.com: https://aws.amazon.com/codeguru/

Beschastnikh, I. (2017). Accelerating software engineering research adoption with analysis bots. *IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Results Track, ICSE-NIER*, 35-38.

Bostrom, N. (2019). The Vulnerable World Hypothesis. *Global Policy*, 455-476.

Butavicius, M. P. (2020). When believing in technology leads to poor cyber security: Development of a trust in technical controls scale. *Computers and Security, 98*, 11.

Cearley, D. W. (2017). Top 10 Strategic Technology Trends for. *Gartner*.

Choudhury, A. (2019, 11 25). *10-ai-applications-that-can-generate-code-themselves*. Retrieved from analyticsindiamag.com: https://analyticsindiamag.com/10-ai-applications-that-can-generate-code-themselves/

Dale, R. (2021). GPT-3: What's it good for? *Natural Language Engineering*, 113-118.

Dam, H. K. (2019). Towards effective AI-powered agile project management. *IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results, ICSE-NIER*, 41-44.

Dam, H. K. (2021). Automatic Feature Learning for Predicting Vulnerable Software Components. *IEEE Transactions on Software Engineering*, 67-85.

DoD, U. (2018). *Summary of the 2018 Department of Defense Artificial Intelligence Strategy*. United States Department of Defense.

Ferenc, R. (2019). Challenging machine learning algorithms in predicting vulnerable javascript functions. *IEEE/ACM 7th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, RAISE* , 8-14.

Fogel, G. F. (2011). Evolutionary programming. *scholarpedia 6(4)*.

Garfinkel, B. D. (2019). How does the offense-defense balance scale? *Journal of Strategic Studies 42(6)*, 736-763.

Hornby, G. S. (2006). Automated antenna design with evolutionary algorithms. *Collection of Technical Papers - Space 2006 Conference*, 445-452.

Huskaj, G. (2019). The current state of research in offensive cyberspace operations. *European Conference on Information Warfare and Security, ECCWS*, 660-667.

Ilieva, R. Y. (2019). The Impact of AI & ML in Agile Production. *10th National Conference with International Participation, ELECTRONICA* .

Johnson, J. (2019). Artificial intelligence & future warfare: Implications for international security. *Defense and Security Analysis 35(2)*, 147-169.

Kantarcioglu, M. (2019). Securing big data in the age of AI. *1st IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA* , 218-220.

Kathleen Metrick, J. S. (2020, april 13). *Think Fast: Time Between Disclosure, Patch Release and Vulnerability Exploitation — Intelligence for Vulnerability Management, Part Two*. Retrieved from fireeye.com: https://www.fireeye.com/blog/threat-research/2020/04/time-between-disclosure-patch-release-and-vulnerability-exploitation.html

Kaufmann, C. L. (1998). What Is the OffenseDefense Balance. *International Security*.

Konaev, M. (2020). U.S. Military Investments. *CSET*, 58.

Malone. (2012). *Offense-Defense Balance in Cyberspace, a proposed model.* Monterey, California: NPS.

Michael Viscuso. (2017, March 28). *'Beyond the Hype: Security Experts Weigh in on Artificial Intelligence, Machine Learning, and Non- Malware Attacks*. Retrieved from Carbon Black: https://www. carbonblack.com/2017/03/28/beyond-hype-security-experts-weigh-artificial-intelligence- machine-learning-non-malware-attacks/

Morales, J. A. (2020). Security impacts of sub-optimal DevSecOps implementations in a highly regulated environment. *ACM International Conference Proceeding Series*.

National Security Council. (2021, Jan 5). *JOINT STATEMENT BY THE FEDERAL BUREAU OF INVESTIGATION (FBI), THE CYBERSECURITY AND INFRASTRUCTURE SECURITY AGENCY (CISA), THE OFFICE OF THE DIRECTOR OF NATIONAL INTELLIGENCE (ODNI), AND THE NATIONAL SECURITY AGENCY (NSA).* Retrieved from cisa.gov: https://www.cisa.gov/news/2021/01/05/joint-statement-federal-bureau-investigation-fbi-cybersecurity-and-infrastructure

Neema, S. (2021, March 16). *Symbiotic Design for Cyber Physical Systems*. Retrieved from darpa.mil: https://www.darpa.mil/program/symbiotic-design-for-cyber-physical-systems

Nowak, A. L. (2018). Assessing Artificial Intelligence for Humanity: Will AI be the Our Biggest Ever Advance ? or the Biggest Threat [Opinion]. *IEEE Technology and Society Magazine 37(4)*, 26-34.

Rahkovsky, M. K. (2020). U.S. Military Investments in Autonomy and AI, A Strategic Assessment. *CSET*.

Richard Knaster, D. L. (2018). *SAFe® 4.5 Distilled: Applying the Scaled Agile Framework® for Lean Enterprises, Second Edition.* Addison-Wesley Professional.

Rowe, C. (2018). *How, why, and when AI & machine learning will transform organizations.* 451Research.

Schneier, B. (2018). Artificial Intelligence and the Attack/Defense Balance. *IEEE Security and Privacy 16(2)*, 96.

Schwenger, M. N. (2020). DevSecOps - An Ideal Use Case For Applying AI. DevOps World - CloudBees.

Shameem, S. (2020). *debuild.co*. Retrieved from debuild.co: https://debuild.co/

Shameem, S. (2020). *twitter.com.* Retrieved from twitter.com: https://twitter.com/sharifshameem/status/1283322990625607681

Slayton, R. (2017). What is the Cyber Offense-Defense Balance? Conceptions, Causes, and Assessment. *International Security*.

Thanh, C. T., & Zelinka, I. (2019). A survey on artificial intelligence in malware as next–generation threats. *Mendel 25(2)*, 27-34.

Truong, T. C. (2020). Artificial intelligence in the cyber domain: Offense and defense. *Symmetry 12(3)*, 24.

Wrubel, E. (2014). Agile Software Teams: How they Engage with Systems Engineering on DoD Acquisition Programs. *IEEE*, 79.

Xie, T. (2018). *Intelligent Software Engineering: Synergy Between AI and Software Engineering*. Champaign, Urbana, IL: University of Illinois at Urbana.

Yuan, X., He, P., Zhu, Q., & Li, X. (2019). Adversarial examples: Attacks and defenses for deep learning. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS 30(9)*.