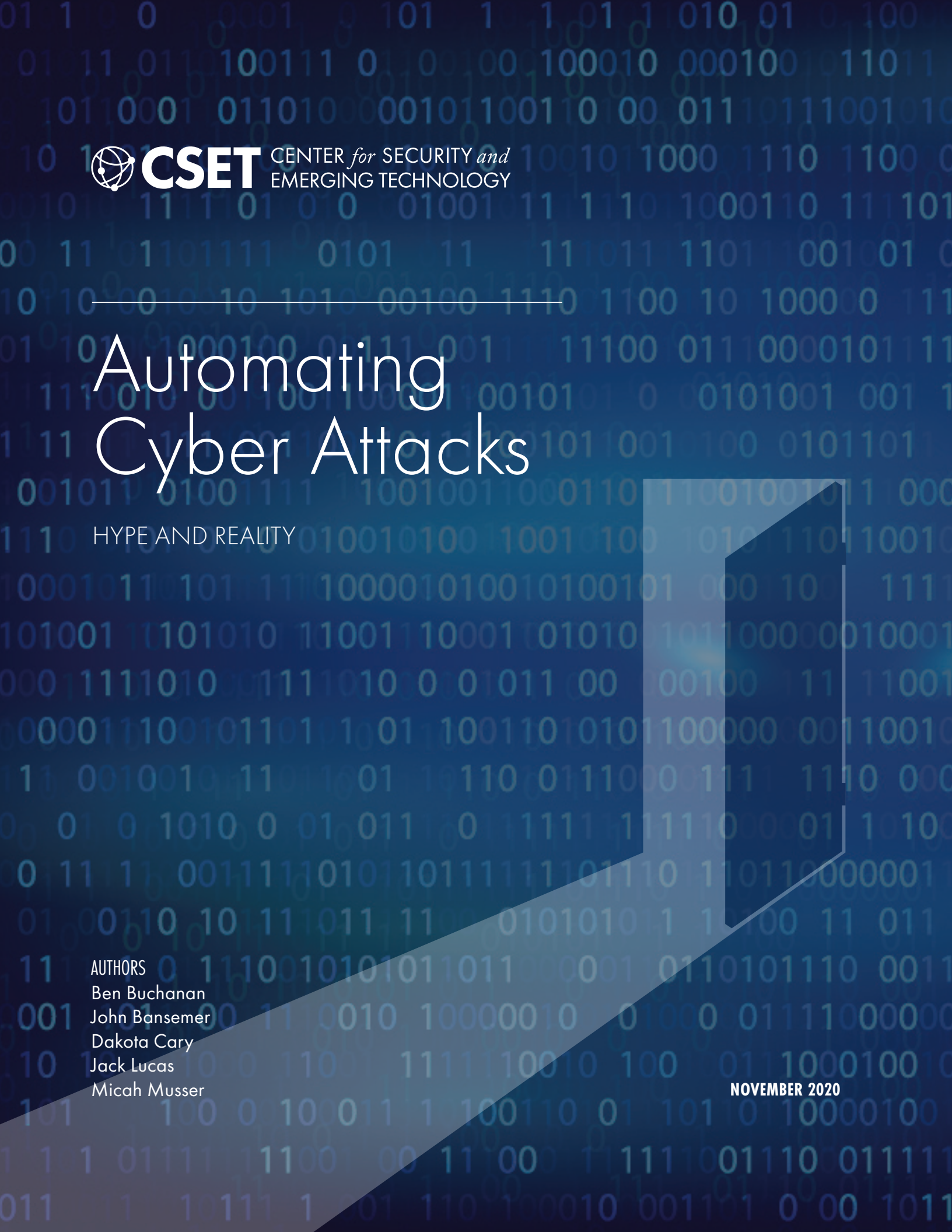

Automating Cyber Attacks

HYPE AND REALITY

AUTHORS
Ben Buchanan
John Bansemer
Dakota Cary
Jack Lucas
Micah Musser

NOVEMBER 2020

CENTER *for* SECURITY *and* EMERGING TECHNOLOGY

Established in January 2019, the Center for Security and Emerging Technology (CSET) at Georgetown's Walsh School of Foreign Service is a research organization focused on studying the security impacts of emerging technologies, supporting academic work in security and technology studies, and delivering nonpartisan analysis to the policy community. CSET aims to prepare a generation of policymakers, analysts, and diplomats to address the challenges and opportunities of emerging technologies. During its first two years, CSET will focus on the effects of progress in artificial intelligence and advanced computing.

CSET.GEORGETOWN.EDU | CSET@GEORGETOWN.EDU

NOVEMBER 2020

Automating Cyber Attacks

HYPE AND REALITY

AUTHORS
Ben Buchanan
John Bansemer
Dakota Cary
Jack Lucas
Micah Musser

ACKNOWLEDGMENTS

The authors would like to thank Perri Adams, Max Guise, Drew Lohn, Igor Mikolic-Torreira, Chris Rohlf, Lynne Weil, and Alexandra Vreeman for their comments on earlier versions of this manuscript.

PRINT AND ELECTRONIC DISTRIBUTION RIGHTS

Document Identifier: doi: 10.51593/2020CA002

Cover photo: KsanaGraphica/ShutterStock.

Contents

Executive Summary

Hacking is a well-established part of statecraft. Machine learning is rapidly becoming an arena of competition between nations as well. With the continued importance of computer hacking and the increasing drumbeat of AI advances due to machine learning, important questions emerge: what might machine learning do for cyber operations? How could machine learning improve on the techniques that already exist, ushering in faster, stealthier, and more potent attacks? On the other hand, how might its importance to cyber operations be misleadingly overhyped?

We examine how machine learning might—and might not—reshape the process of launching cyber attacks. We examine the cyber kill chain and consider how machine learning could enhance each phase of operations. We expect certain offensive techniques to benefit from machine learning, including spearphishing, vulnerability discovery, delivering malicious code into targeted networks, and evading cyber defenses. However, we caution that machine learning has notable limitations that are not reflected in much of the current hype. As a result of these constraints and flaws, attackers are less likely to apply machine learning techniques than many expect, and will likely do so only if they see unique benefits. Our core conclusions are:

- Current cyber automation techniques are powerful and meet the objectives of many attackers. For most attackers, they will not have an obvious need to augment their operations with machine learning, especially given the complexity of some machine learning techniques and their need for relevant data. If current methods of automation become less effective or machine learning techniques become more accessible, this may change.

- In the near term, machine learning has the potential to increase both the scale and success rate of spearphishing and social engineering attacks.
- Of the machine learning techniques reviewed in this paper, reinforcement learning promises the most operational impact over the medium-to-long term. Though its potential impact is speculative, it could reshape how attackers plan and execute cyber operations.
- Machine learning systems have substantial limitations, such as their reliance on salient data, their weakness to adversarial attacks, and their complexity in deployment.
- Like other cyber capabilities, many machine learning capabilities are inherently dual-use, with the advantage accruing to those who have the resources and expertise to use them best rather than always favoring attackers or defenders.

The paper proceeds in three parts. The first part covers the state of the art in cyber operations today, showing how attackers progress through the kill chain and taking care to demonstrate how traditional automation assists them in their efforts. The second part considers machine learning in more depth, exploring its differences from traditional automation and probing how those differences might—and might not—reshape key parts of the kill chain. Among other things, it highlights the way in which machine learning could improve discovery of the software vulnerabilities that enable cyber operations, grow the effectiveness of spearphishing emails that deliver malicious code, increase the stealthiness of cyber operations, and enable malicious code to function more independently of human operators. The conclusion takes stock, drawing out key themes of geopolitical and technical importance. It argues that machine learning is overhyped and yet still important, that structural factors will limit the relevance of machine learning in cyber operations for most attackers, that the dual-use nature of cyber operations will continue, and that great powers—including the United States—should be proactive in exploring how machine learning can improve their operations.

Introduction

The use of a computer at Lawrence Berkeley Laboratory in 1986 cost $300 per hour.[1] One day, when reviewing the accounting ledgers, a system administrator discovered a seventy-five-cent discrepancy. The administrator asked another staffer, Clifford Stoll, to investigate. What followed was one of the first and most well-documented hunts for a cyber criminal. Stoll, in his classic book *The Cuckoo's Egg*, details a case study in persistence on the part of both the attacker and the defenders that, from today's vantage point, seems to develop in slow motion.

This attack was not a highly automated quick strike. Instead, it unfolded over the course of months. The attack techniques, directed from a computer halfway around the world, were manual and relatively unsophisticated, yet effective. This slow pace and lack of automation is not surprising. The internet at the time had about 20,000 connected computers, transmission speeds were measured in kilobytes, and computing power was a fraction of what is available on today's mobile devices.

The attacker followed an operational process, or kill chain, that has largely endured: reconnaissance, initial entry, exploitation of known vulnerabilities, establishment of command and control channels, and lateral movement across networks. Each of these steps contributed to the ultimate objective of exfiltrating sensitive documents from defense contractors, universities, and the Pentagon.[2] With striking simplicity, the attacker attempted logging onto systems with known account names and commonly used passwords, such as "guest." Even with this rudimentary technique, the attacker gained unauthorized access upwards of 5 percent of the time.[3]

While the attack was largely manual, automation aided the defenders. Stoll and others established automated systems to alert them when the

attacker accessed key machines and networks, enabling the team to begin tracing the ultimate source of the attacks. Sometimes with the help of court orders and telephone companies, the defenders systematically worked back through the tangled network of infected computers toward the attacker. Stoll and his team baited the attacker with enticing (but fake) files related to the highly sensitive Strategic Defense Initiative—the rough equivalent of Cold War catnip.[4] The attacker spent so much time online examining the bogus files that technicians were able to trace the intruder's location: Hanover, Germany. Local authorities eventually charged Markus Hess and four of his German associates with espionage for their various roles in feeding pilfered documents and network details to the Soviet security agency then known as the KGB.

About a year later, Stoll received an alarming call about a new threat: an automated attack was cascading across the internet, digitally destroying everything in its path. Stoll and other computer security experts raced to stop the self-propagating code, which became known as the Morris Worm. They succeeded, but not before the worm disabled more than 2,000 computers in the span of 15 hours.[5] This attack stood in sharp contrast to the manual operations of the period and introduced the concept of automated cyber attacks.

The two cases neatly bookend the spectrum of conceptual possibilities when it comes to cyber operations. On one end are the plodding manual efforts, painstakingly carried out by attackers and thwarted by system administrators and their tools in a cat-and-mouse game that unfolds over months. On the other end are the automated attack sequences—often lacking nuance or control—that tear across the internet at high speed and destroy everything in their path. Operations at both ends of the spectrum continue today, though human-directed efforts benefit from more automation and automated attacks exhibit greater control than before.

In this context arrives machine learning, a technology at the core of almost all the hype surrounding AI today. Within the last decade, machine learning has achieved technical feats that were not too long ago thought to be decades or even centuries away. Machine learning algorithms have beaten world champion players at fiendishly complex board and video games, demonstrating something akin to intuition. These algorithms have devised convincing photos and videos of people who never existed, painted compelling portraits, and written music and stories so good that they seem humanlike in their creativity. They have done so with rapidly increasing speed and quality, charting a growth curve in capabilities that seems to point ever upward.

Against this backdrop of advances, important questions emerge: what might machine learning do for cyber operations? How could the improved automation

technology improve on the techniques that already exist, ushering in faster, stealthier, and more potent attacks? On the other hand, how might it be misleadingly overhyped?

In this paper, we tackle these questions. To do so, we proceed in three parts. The first part covers the state of the art in cyber operations today, showing how attackers progress through the kill chain and taking care to demonstrate how traditional automation assists them in their efforts. The second part considers machine learning in more depth, exploring its differences from traditional automation and probing how those differences might—and might not—reshape key parts of the kill chain. Among other things, it highlights the way in which machine learning could improve discovery of the software vulnerabilities that enable cyber operations, grow the effectiveness of spearphishing emails that deliver malicious code, increase the stealthiness of cyber operations, and enable malicious code to function more independently of human operators.

The conclusion takes stock, drawing out key geopolitical and technical judgments. It argues that machine learning is overhyped and yet still important, that structural factors will limit the relevance of machine learning in cyber operations for most attackers, that reinforcement learning techniques show promise in the medium-to-long term, that the dual-use nature of cyber operations will continue, and that great powers—including the United States—should be proactive in exploring how machine learning can improve their operations.

1 The Cyber Kill Chain

The kill chain is an established method of conceptualizing cyber operations by presenting a checklist of tasks that attackers work through on their way to their objective. Lockheed Martin researchers published a canonical paper outlining the idea in 2010.[6] Other organizations, such as MITRE, have introduced more complex versions.[7] While the kill chain model has limitations—such as portraying cyber operations as overly linear—it is a common and useful way to begin to understand cyber attacks. We therefore use it as a foundation to explore how cyber operations work and how automation that does not use machine learning aids attackers; this is the status quo that machine learning-enabled automation seeks to advance.

Attackers will perform some or all of the kill chain's steps. Depending upon their overall objective, attackers may merge several steps by employing commonly used exploit tools or techniques. Each step they execute represents an opportunity for a defender to stop an attack. We discuss six steps widely agreed to be important: reconnaissance, weaponization, delivery, command and control, pivoting, and actions on objective. Each step constitutes its own processes, challenges, and techniques—all of which continue to evolve, including with greater automation.

This section illustrates well-known cases of each step of the kill chain and current state-of-the-art techniques. Readers familiar with the cyber kill chain and how automation helped enable major operations—especially NotPetya, CRASHOVERRIDE, Agent.BTZ, Conficker, and the 2015 Ukraine blackout—should feel free to skip ahead to our discussion of machine learning in the following section.

RECONNAISSANCE

Attackers must first pick their target. Their process of reconnaissance and target selection depends on the objectives. Some attackers will be interested in infecting broad categories of users and will largely forego this process. Others are more selective in their choice of victims, requiring a more extensive reconnaissance effort. In this phase, attackers first identify humans and machines that are worth targeting and then gather information about the technical vulnerabilities of those targets.

To inform their search for human targets, attackers can gather important details about an organization and its personnel through internet searches, social media analysis, and scraping technical online forums. These passive techniques have the added advantage of being largely undetectable. Traditional techniques of automation offer a means to collect, sort, and analyze data collected in this way, significantly shortening time spent in this phase and helping attackers plot their next move. Such techniques may be augmented with fairly simple machine learning-enabled methods to identify the victims most susceptible to a variety of social engineering techniques.

To inform their search for machine targets, attackers can use more active techniques, such as automated scanners that probe target networks for details on their connected systems, network defenses, and associated software configurations. Available since the late 1990s, nmap is a popular, freely available, automated tool that has evolved to include new functionalities and user interfaces, enabling attackers to remotely gain more information about their potential targets and more easily interpret the results.[8] This kind of active reconnaissance is extremely common, and most devices on the internet are constantly being scanned by a wide variety of malicious actors, many of whom are looking for vulnerabilities to exploit.

WEAPONIZATION

With their targets identified, attackers have to discover and exploit technical weaknesses in their target's software to gain illicit access. Attackers must then couple their malware with a vulnerability to create a payload that is later delivered to their target.[9] This process is called weaponization. The right exploit code takes advantage of the newly discovered weaknesses and grants the attackers the freedom to act in the target's network, often while remaining undetected. If a cyber operation were a bank heist movie, the malicious code would be the robber with just the right set of skills for the specific job.

Automated weaponization tools can rapidly identify vulnerabilities and assemble code to exploit them. These tools often feature databases of exploits that attackers can search through to find ones that suit their target's apparent vulnerabilities. Some tools, such as Metasploit, list thousands of freely available exploits, each

ranked from "Low" to "Excellent" based on reliability, impact, and likelihood of crashing the targeted system. Metasploit's automated capabilities help determine if a machine is vulnerable to one or more previously designed exploits. Its automated payload generator can combine bespoke malware and known vulnerabilities to aid the weaponization process.[10] Its auto-exploit feature takes this a step further, giving attackers the ability to point Metasploit at their target, provide details on what they learned during the reconnaissance phase, and then allow the tool to take the attack from there.[11] Another tool, AutoSploit, takes automation a step further by combining Metasploit with Shodan, which allows users to quickly search the internet for vulnerable systems.[12]

However, these tools, at least at the present time, do not develop new exploits autonomously. The hard work of writing code that exploits a previously unknown vulnerability is still largely a human-directed endeavor. To find new vulnerabilities, attackers often begin by investigating the code that runs on their target's system, again using information obtained during the reconnaissance phase. Tools called fuzzers may aid in this process. Fuzzers seek out bugs and vulnerabilities by bombarding a selected piece of software with many inputs and monitoring the results. These inputs can be entirely random or tailored to the software being tested. For example, attackers may seek to exploit commonly used software, such as the Chrome browser. They might use fuzzers to enter thousands of inputs into the URL bar with the knowledge that only a handful may cause a program to crash. The attackers can then study each of these crashes to investigate why it occurred, as such crashes often hint at software vulnerabilities. From there, they can begin to develop an exploit that, once delivered, will grant them illicit access.[13]

DELIVERY

The delivery phase of the operation is what most people envision when they imagine someone hacking into a system: the attackers typing at a computer and exclaiming "we're in!" After conducting reconnaissance and weaponizing a piece of software, the attackers must now complete the sometimes-trivial and sometimes-daunting task of getting that code onto their target system. Making entry into the targeted system can happen through machine or human vulnerabilities.

Some malicious code can be delivered via a watering hole attack, in which attackers compromise a legitimate website and infect all of its visitors with an exploit targeted at their browser.[14] Other operations spread via USB drives infected with malicious code.[15] Still other operations are carried out via third parties with which the target interacts. These operational techniques rely on the attacker moving "upstream" to a trusted party over which the victim has no control, such as a company that provides IT services or other software to the target. NotPetya, the 2017 Russian cyber-attack that caused billions of dollars in damages to the computer networks

of companies and nations globally, illustrates this method: the vector of infection for the first tranche of victims was through an automatic update function in a piece of tax software ubiquitous in Ukraine.[16]

When targeting human weaknesses, attackers often use social engineering to induce behavior that compromises an organization's security. The attack methods deployed against people are as varied as our emotions. Attackers have faked phone calls from IT departments, claiming an emergency is underway and that the organization needs their password immediately to stop an attack. Spearphishing attempts exhibit much creativity, too; an attacker may spoof an email address to resemble the HR department and send an email with a subject line "2020 Salary Scale - Confidential" with a weaponized document attached, after which the attacker might immediately send another email with the subject line "DO NOT OPEN PREVIOUS EMAIL"—a warning that only makes users more curious and entices them to download the malicious code in hopes of seeing confidential compensation documents. A 2019 Verizon study found that for the median company, more than 90 percent of all detected malicious code was initially delivered via email, and that spearphishing—which remains a largely manual process—was used by 78 percent of attackers conducting cyber-espionage operations.[17]

Attackers can use spearphishing at many points in the attack cycle, and at a large scale.[18] For example, in 2016, Russian military intelligence operatives targeted key members of the Democratic National Committee and John Podesta, chair of Hillary Clinton's presidential campaign.[19] The scale of the effort, which featured more than 9,000 spearphishing links, illustrates both the perceived value of the technique as well as the decision calculus of phishing with so many spears: send out a large number of targeted emails and hope a few unsuspecting users take the bait.[20] This is an area ripe for more automation in the future.

COMMAND AND CONTROL

After attackers finally infiltrate their targeted system, the next step is to establish a secure line of communication to the code they have placed. Through this channel, known as command-and-control (C2), attackers can pilot their malicious code and execute commands as if they were sitting at the infected computer. Attackers create and design their C2 infrastructure based on the victim's network security posture and configuration, the objective of the malicious code, and the frequency with which they need to communicate instructions. Variations in C2 structure present trade-offs between speed, stealth, and resilience. In some instances, attackers prioritize speed and the ability to exfiltrate large amounts of data. Other cyber operations prioritize stealth and use delay-tolerant C2, transmitting information through circuitous channels to avoid detection by defenders. As attackers' objectives change with each hacking campaign, so too do their tools and tactics.

Early cyber operations often hard-coded their C2 communication channels and offered no flexibility in their malicious code. Like a child dropped off at school with a list of numbers to call in case of emergency, these explicit instructions gave the malicious code clear direction and little discretion. For example, in the Moonlight Maze case from the late 1990s—in which Russian government hackers infiltrated the United States Air Force Research Lab, Sandia National Laboratory, NASA, and the Department of Energy—the attackers used two common networking protocols as their C2 channels.[21] Hard-coded channels are easy to block once discovered, and so attackers have since evolved ways to obfuscate their C2 methods.

In 2008, Conficker, a virulent computer worm, signaled the beginning of a new era. It was the first well-publicized instance of malicious code utilizing C2 infrastructure that was not hard-coded with a preset directory of domains to check.[22] The first version of Conficker used an algorithm to generate pseudo-random domain names for C2, essentially expanding and changing the list of numbers it would call. By using such an algorithm, the attackers determined which channels Conficker would call out to at any given time in a way that defenders had a hard time predicting and blocking. Conficker originally generated 250 new possible C2 channels every single day.

Later versions of Conficker took this further, increasing the number of daily generated domains and, more significantly, incorporating a peer-to-peer C2 option. With the peer-to-peer upgrade, computers already infected with Conficker could connect to one another for updates and relay commands between versions with access to the internet and those without. This feature represented an important increase in the capability of C2 infrastructure, one that frustrated defenders who still believed that blocking the malicious code's C2 domains was the best solution. The practice of using various automated techniques to avoid preset C2 infrastructure is now quite common.*

Sometimes the target of attacks is air-gapped or logically separated from the internet. In those cases, attackers may resort to different C2 mechanisms that are both delay tolerant and capable of bridging the air gap. For example, a top-tier Russian hacking group known as APT28 continues to use a wide array of tactics, including the USBstealer malware, to do this. This malware provides a mechanism to copy files from physically separated networks for later exfiltration, often through the same C2 network. It also allows commands to be connected across infected and potentially air-gapped devices. Attackers can add specific execution commands to infected USB drives; these commands are then automatically propagated onward as the USB drive makes its way to new victim systems.[23]

*For example, see the discussion of HAMMERTOSS below.

A separate sophisticated Russian hacking group, known as APT29 or Cozy-Bear, in 2015 started using its own automated techniques to obscure C2. It deployed a mechanism in its HAMMERTOSS malicious code that camouflaged C2 activity among normal network traffic.[24] HAMMERTOSS exploited the defender's trust of websites like Twitter, Github, and Microsoft Azure. HAMMERTOSS first checked a Twitter profile selected by an algorithm at a preset interval, from which it collected a decryption key hidden in that profile's latest tweet. The code then visited a GitHub account linked in that same tweet and downloaded a photo posted by the actor from that account. From there, HAMMERTOSS decrypted its instructions hidden inside the photo with the decryption key posted by the attackers' Twitter account. To blend in with normal office web traffic, HAMMERTOSS did all of this only during work hours. The attack demonstrates the benefits of camouflaging behavior by automatically hiding in the noise, making it hard for defenders to detect the C2 activity and track the attackers' operations.

If the attackers believe that every step of their attack can be automated in advance, they may forego the use of C2. The aforementioned Morris Worm, for instance, avoided using a C2 system because doing so would have slowed down the worm's spread and provided a means of more quickly identifying the attacker. The price of this decision was the attacker's total loss of control, which allowed the worm to cause far more damage than had apparently been intended.

Autonomy and C2 are thus related: most cyber operations will involve C2 infrastructure until attackers are able to automate reliably the essential elements of their operation. For example, it is reasonable to assume attackers will continue to include C2 for long-term data exfiltration campaigns because the operational objectives require the ability to send information back. Likewise, C2 may also persist as a fail-safe function for malicious attack code; in the event of an error or environment that the malicious code cannot process through its other automated functions, having the ability to phone home allows operational resilience in the face of unforeseen complications. On the other hand, for some future attack operations carried out by risk-tolerant adversaries, C2 may be less important if key parts of the kill chain can be automated, beginning with pivoting.

PIVOTING

Achieving an operational objective almost always requires compromising more than one device. After gaining access to an initial machine, attackers usually shift their attention to pivoting: the act of using a compromised system to infect other systems. Sometimes the primary goal of an operation is to spread to as many computers as possible. However, indiscriminate pivoting often increases the risk of detection, so many operations invest significant attention into pivoting strategically, identifying the most promising follow-on systems in a steady advance toward

the ultimate objective. Either method of pivoting includes two distinct components: privilege escalation, which involves gaining additional access and permissions to a compromised system, and lateral movement, which involves using credentials or software vulnerabilities to gain access to additional machines.[25]

Pivoting may make use of tools that exploit the same technical or human vulnerabilities attackers used to gain initial access to a network. For instance, attackers who have compromised trusted email accounts within a network—such as administrator accounts maintained by IT staff or accounts of senior employees—may use them to engage in further spearphishing, directly targeting accounts with still higher administrative privileges in order to get additional passwords and access.[26] Or a software exploit that granted illicit access to one machine on a network may work just as well against other machines.

As in the aforementioned Morris Worm, some code automates pivoting, propelling itself onto additional machines or networks. Although worms usually do not discriminate between different networks, they can be designed with an understanding of a target's network architecture. A famous example was a worm, known as Agent.BTZ, that infected both unclassified and classified United States military networks in 2008. Attackers used USB drives with malicious code that self-propagated and caused a significant number of infections.[27] In this attack, infected systems compromised USB drives connected to them, and then the drives infected additional systems when connected elsewhere. Malicious code on the devices enabled the attackers to exfiltrate files from these machines, even if they were not directly connected to the internet.

Since Agent.BTZ, auto-propagation systems have become more powerful. Until Windows 8.1, for instance, all Windows machines stored and automatically reused credentials so that users would only need to input them once. When the potential vulnerability introduced by this system was pointed out to Microsoft, the firm did nothing, arguing that the credentials were stored in memory that would only be accessible if an attacker had already obtained administrative control over a machine.[28] But a French information technology manager realized in 2011 that if an attacker could compromise one machine on a network, the attacker could find these stored credentials in memory and use them to compromise additional machines. His automated tool, Mimikatz, does exactly that: after first gaining access to one machine, Mimikatz can automatically scoop up a user's credentials and use them to access any other machine on which that user has an account. Mimikatz has become one of the most-used pieces of code deployed by attackers, and almost all Windows machines remain vulnerable to it today.*

*The reason: although all versions of Windows since 8.1 have let users disable the credential-storing functionality, an attacker with full administrative privileges can simply reenable the option.

While worms using these types of self-propagation techniques can be extremely powerful, automation can sometimes undermine an operation's strategic goals. The previously mentioned Agent.BTZ's auto-propagation system flung it far beyond Department of Defense networks, with infections occurring across the world years after its initial release.[29] More interesting still is the aforementioned case of Conficker. By the time the worm was discovered in November 2008, it had already infected millions of machines with code that instructed each machine to attempt to contact a well-known criminal website on December 1, 2008. But when the day came, the sudden surge of traffic brought the site itself down. In effect, the worm had become so large that it could not sustain its own weight.[30] Though Conficker continued to infect computers for years afterwards, it was never again used for any meaningful operations—possibly because it had already spread too far and attracted too much attention.[31]

ACTIONS ON OBJECTIVE

All of this offensive effort, from reconnaissance to pivoting, is a means to an end. If the attackers succeed at every step in the kill chain to this point, they can finally act against their target to fulfill their objective. These actions on objective can begin once the attacker has verified that they have reached their target machine—often confirmed by the computer name, files on the computer, or its position within the network. For indiscriminate campaigns, any computer will suffice.

Data exfiltration is the most common action on objective, and with good reason.[32] For example, the ability to save years of research and development by stealing the plans to a major weapons system is highly valuable to an attacker. Some states—most notably China—have engaged in long-term intellectual property theft in order to advance their strategic goals.[33] Cyber espionage can allow intruders to silently extract secrets for years on end.

One notable case of cyber espionage is worth discussing because of its marked *lack* of automation. In 2012, Federal Bureau of Investigation (FBI) officials uncovered emails between three Chinese agents that indicated they were attempting to steal detailed plans for the C-17, a workhorse American cargo aircraft. But the Chinese hackers had a problem: although they had gained access to a number of crucial networks, they could not exfiltrate every file without raising suspicion. Worse, they lacked the tactical knowledge to know which files were the most important. To solve this problem, they contracted with Su Bin, a Chinese aviation expert living in Canada. The hackers sent Su lists of thousands of files. He examined the names and manually indicated which files appeared the most worth copying. By November 2014, the Chinese military had created its own knock-off version of the C-17 using the stolen files—though unfortunately for Su, this success did not prevent him from being one of the few Chinese hackers arrested by Western governments.[34]

Of course, not all data exfiltration requires such manual work. Automated tools have been built to exfiltrate a variety of different types of data. For example, some tools automatically exfiltrate any filename that contains a specific extension or record and exfiltrate every keystroke made on an infected computer.[35] However, the larger the net that attackers cast when exfiltrating data, the more easily defenders can notice their activity, which makes indiscriminate grabs useful for attackers only if long-term data extraction is not the goal—or if no one is watching.

Other actions on objective are more aggressive than exfiltration. Attacks can be motivated by profit, including operations that encrypt critical files until a ransom is paid or compromise financial systems and make fraudulent transfers—a tactic that has been used by North Korea to enrich itself by tens of millions of dollars.[36] Other attacks are motivated by the goal of resource hijacking, in which a network is compromised in order to use its computational power for some other goal. Some Chinese actors, for instance, have been observed compromising networks in order to convert them into cryptocurrency mining tools.[37]

Other attackers seek to infect large numbers of vulnerable computers to create vast botnets that can then be used to attack unrelated targets. This is often part of a distributed denial of service attacks, in which a botnet sends so many useless requests to a target that the target becomes overwhelmed and legitimate users are prevented from accessing it. The process of compromising large numbers of systems for these operations is highly automated and often indiscriminate.

Still other attack objectives, however, are motivated by the pure desire to sabotage compromised systems or create chaos. This was the ultimate goal of the aforementioned NotPetya. The attack initially appeared to be a campaign motivated by profit because it repurposed ransomware to encrypt major files and displayed a ransom message on many machines. In actuality, after infecting a machine and automatically spreading to adjacent machines, NotPetya encrypted each computer's master boot record, a critical component that assists in loading the operating system. Because NotPetya contained no mechanisms to reverse this encryption process, the only motivation was destruction, pure and simple.

Even more troubling than the type of digital destruction caused by NotPetya is the possibility of physical destruction. Attacks can cause physical destruction by undermining computer systems that oversee physical infrastructure. Although such attacks are often theorized, very few operations over the last decade have specifically targeted industrial control systems.[38] In large part, this is due to the complexity of most industrial control systems infrastructure, which is typically so esoterically designed that it can only be manipulated by experts with extremely advanced technical knowledge of the system in question. However, it is also clear that these systems continue to be a focus of attackers. In 2017, the Department of Homeland Security

and FBI issued warnings that attackers had gained access to a wide variety of control systems as part of a longer-term campaign strategy to lay the groundwork for future disruptions.[39]

There is some evidence of increasing automation in attacks on industrial control systems. In 2015 and 2016, Russian attackers launched major operations against the power grid in Ukraine. The differences between the two attacks provide a powerful signal about the future of destructive cyber operations.

The 2015 outages used a number of automated systems to conduct reconnaissance and wipe data, but the actual attack was decidedly manual. Attackers gained control of infected industrial control systems and manually clicked through the controls that would open circuit breakers and halt the flow of power. Operators watched in horror as their computers started sabotaging the grid in front of them. But despite the terrifying visual of a computer cursor seemingly moving with a malicious will of its own, each manual attack at each substation required a distinct human operator. Video from the attack also suggests that the attackers were not exactly sure how to bring down the grid and had to do a bit of exploring in the middle of the attack—a telltale sign that they were in no position to adequately automate the attack ahead of time, despite their extensive reconnaissance.

A year later, Russian attackers launched a new piece of malicious code called CRASHOVERRIDE.* CRASHOVERRIDE was meant to substantially automate the attack process: its core module could automatically find circuit breaker controls and toggle them on and off, creating a blackout.[40] Analysts also noted that the malicious code could be easily adapted to other power grid systems in Europe, the Middle East, Asia, and the United States.[41] In effect, the creators of CRASHOVERRIDE had developed an automated weapon that they could easily adapt for electrical grids all over the world, and that they could use, in theory, to generate blackouts at the flip of a switch. Describing the complexity and power of the attack code, the security firm ESET wrote that, "any intrusion into an industrial network with systems using these protocols [targeted by CRASHOVERRIDE] should be considered as 'game over.'"[42] This attack is perhaps the most powerful sign yet of the role increased automation will play in the future of offensive cyber operations.

*CRASHOVERRIDE is sometimes known as Industroyer.

2 How Machine Learning Can (and Can't) Change Offensive Operations

As the preceding section showed, automation is already part of cyber operations. Attackers create and use rule-based automated tools to power their operations, often with enormous effect. Outside of cyber operations, another kind of automation—machine learning—has proven enormously powerful in recent years. Advances in machine learning have transformed other disciplines; to what degree will they transform cyber operations? This is a vital question.

The key difference between traditional rule-based and machine learning systems is that machine learning systems have the capacity to learn and modify their own behavior to achieve some measurable objective. Machine learning is not by definition more powerful or more sophisticated than rule-based automation. It is simply different, and in ways that offer exciting, intriguing, and alarming possibilities.

Imagine trying to build a program to automatically guess passwords. Rather than simply guessing every possible combination of letters and numbers, engineers might use a long list of heuristic rules to create a program that could work far more efficiently. For instance, the program might try combining common words from a dictionary, or it might be instructed to replace "S" with "$." But this approach requires humans to identify and code each of these rules. By contrast, a machine learning program could be given a set of commonly used passwords and could learn which combinations of letters and numbers occur most frequently, thereby learning how to most efficiently crack passwords without any need for explicit human instruction. Moreover, a machine learning program could continue updat-

ing its behavior based on its own track record or the addition of new data (such as sets of stolen passwords from data breaches).

Machine learning systems require a single measurable objective in order to assess and improve their performance. In the case of the password-guessing program, that objective might be how many accounts are successfully cracked, possibly weighted by the importance of the account. In many other cases, however, it is much harder to distill operations to just a single objective, especially when major tradeoffs are at stake. Offensive cyber operations often involve such tradeoffs, making machine learning systems less applicable.

In the password-guessing example, we can imagine different scenarios in which either a rule-based or a machine learning-based system might perform better. For example, companies may track their customers' most common passwords and dissuade them from using certain ones that are becoming too common. In this context, a rules-based system would quickly become obsolete unless it was manually updated with new likely passwords, while a machine learning system could update itself as it observed customers gravitating toward new passwords. On the other hand, perhaps the attackers are only looking to crack a small number of accounts and do not have the capacity to make many guesses. In this context, a machine learning system would not have enough data to improve itself and a traditional approach would be better suited. This lack of data from which to learn is a recurring problem in offensive cyber operations, and limits the applicability of machine learning.

With these general abilities and frailties, machine learning systems are ripe to improve automation in some parts of the kill chain but are deeply unlikely to matter much for other tasks. To explore the impact of these systems, we focus on four types of machine learning: supervised learning, adversarial learning, generative learning, and reinforcement learning. These terms are somewhat abstract and not mutually exclusive, since some of the most compelling recent advances in AI capabilities have come from combining multiple machine learning techniques. Nonetheless, each of these four types of machine learning can help attackers with specific parts of the kill chain.*

SUPERVISED LEARNING

Supervised learning systems are perhaps the most straightforward kind of machine learning. The approach is often used for classification tasks, such as determining if an email is spam or not. To train such systems, engineers provide the machine with data sets of examples that include the proper classification, such

*Not all of these terms align perfectly with the practitioner's usage, especially the term "adversarial learning," which means something far more specific to practitioners than the meaning we attach to it in this paper. We use these terms in a less precise way in order to group multiple approaches together heuristically and allow the policymaker to understand some of the intuitions behind what makes machine learning successful at specific tasks.

as thousands of emails marked as spam or not spam. From these examples, the system learns which patterns—such as the text of the email—correspond to which classifications. Similarly, supervised learning systems can be trained to spot software running on a network that is likely to be malicious by learning what activity is normal and what activity is anomalous.[43]

Supervised learning can also help both defenders and attackers with an important question: which vulnerabilities are worth exploiting? Defenders will want to prioritize their security efforts to remediate these weaknesses. Attackers will want to exploit the most damaging of these flaws in the weaponization and delivery steps of the kill chain. Machine learning can help both sides determine where to focus their efforts.

Network defenders historically used crude scales to determine which vulnerabilities are the most severe. One scale, known as the Common Vulnerabilities Scoring System (CVSS), measures the severity of vulnerabilities based on three groups of characteristics, ranging from what privileges are required to whether the vulnerability requires user interaction.[44] More recent supervised learning approaches study and predict which exploits attackers are most likely to employ.[45] The Exploit Prediction Scoring System (EPSS) was trained on proprietary data from anti-virus vendors. The training data included information about a list of known vulnerabilities, including their software vendor, the relative severity of the vulnerabilities, and whether malicious code had already been written to exploit them. This data allowed the EPSS to determine which vulnerabilities most deserved defenders' attention. The result was a 76 percent reduction in the number of vulnerabilities that needed to be addressed to achieve the same level of security obtained using the old CVSS scale.[46] Tenable, a cybersecurity firm, advertises a similar product that uses machine learning to prioritize patching and performs similarly to the proposed EPSS.[47]

Attackers historically relied on experience and intuition to know which vulnerabilities to exploit, but supervised learning may reshape this part of the kill chain. The creators of the EPSS acknowledge that the same technology that can improve efficiency for defenders can act as a targeting tool for attackers. Their research may provide a blueprint for attackers to either conduct their own similar research or to exploit vulnerabilities that EPSS says are "low risk." Additionally, if defenders prioritize addressing vulnerabilities based on the EPSS, attackers may use the inverse approach to select vulnerabilities that are potent yet unlikely to be fixed.[48] In shaping which vulnerabilities get addressed and which get exploited, supervised learning will shape cyber operations.

Supervised learning can also help attackers once the operation begins. While defenders use historical network traffic data and anomaly detection tools to identify attacks, attackers may be able use the same data and methods to evade detection.

One particular cyber defense technique is to create honeypots that simulate actual users or machines and look like enticing targets to would-be attackers. Defenders can monitor the traffic launched against honeypots to gain insight into their adversaries' tactics and techniques. But a honeypot must be convincing to be effective, and defenders setting up honeypots often find it difficult to convincingly replicate typical user behavior and network traffic. Attackers can try to determine if the target is a honeypot, perhaps by looking for anomalies or determining if the target is a virtual machine (which are often used for that purpose). Some analysts speculate that the vast botnet known as Emotet uses machine learning-enabled methods to spot honeypots. The Emotet malicious code refuses to infect or quickly removes itself from machines that it determines are honeypots, thereby eluding the attempts of researchers to track its movements.[49]

ADVERSARIAL MACHINE LEARNING

In recent years, AI research has increasingly focused on machine learning techniques that learn from trial and error, rather than by interpreting historical data. Oftentimes, these learn-by-doing AI systems can be designed with the goal of "beating" some other system, whether a regular piece of software or another AI. These types of AI are often known as adversarial machine learning.[50] Adversarial machine learning systems can be useful at various stages of the kill chain, but attackers use them for two specific tasks: weaponization and defense evasion (which is helpful throughout the entire kill chain).

During the weaponization stage of the kill chain, an attacker may find vulnerabilities in some piece of software on a target network that can be exploited to gain entry. One means of pursuing this strategy—as described in the section on weaponization—is to use fuzzers to find inputs that cause a crash that may hint at underlying vulnerabilities.[51] In recent years, some fuzzers have turned to systems that rely on machine learning or techniques adjacent to machine learning. These algorithms first randomly produce thousands of inputs, then examine which inputs caused unexpected behavior in the target system and which did not. At this stage, the principle of survival of the fittest is applied: unsuccessful mutations are culled from the population, and successful variants are themselves randomly mutated to produce thousands of new inputs. This type of adversarial AI can both learn the expected input structure of a computer program while simultaneously finding subtle ways of breaking that structure to expose a vulnerability.[52] Because of their potential ability to better focus their search, machine learning fuzzers may enable attackers to find far more significant vulnerabilities in the weaponization stage of the kill chain.

After weaponization, the need for defense evasion arises because attackers will have to overcome their target's security measures as they progress through the kill chain. These barriers, such as anti-virus systems, intrusion detection systems,

spam filters, and other defenses often use machine learning. While the earliest spam detection rules-based systems were easily beaten once attackers learned the rules—such as rewriting "free" as "F*R*E*E" to avoid being classified as spam—modern machine learning cyber defense systems are more flexible and harder to evade.[53]

These newer defenses are, however, vulnerable to attacks by adversarial learning systems.[54] For example, researchers showed that alterations that to a human are imperceptible can fool supervised learning, such as those the separate benign and malicious activity to fail.[55] More generally, the rise of newer and more powerful adversarial techniques has created a significant worry that the best tools available to defenders may not be able to detect the next generation of cyber attacks.[56] These types of adversarial approaches will make defense evasion significantly easier for attackers: as early as 2016, researchers successfully used a genetic algorithm to create malicious code that could randomly mutate itself to evade antivirus software, and by doing so repeatedly, could learn what mutations to introduce to remain undetectable.[57]

GENERATIVE LEARNING

Supervised learning systems often focus on recognition, such as determining if activity is malicious or benign. Adversarial learning systems often focus on exploitation, finding weaknesses in other pieces of software. A third type of machine learning—which we call generative learning—focuses on something altogether different: producing new creations that fit within certain parameters. Drawing on some major technical breakthroughs in recent years, these systems mimic something like imagination.[58] These generative learning systems can create very realistic-seeming snippets of text, video, or audio. Among many other achievements, they can create "deepfake" images and video and write complex text on the fly.[59]

These developments have major implications for cyber operations, some of them direct and others less so. The largest impact will likely be on the delivery stage of the kill chain, when the goal is often to find ways of tricking unwitting humans within a network into installing or executing malicious code. More convincing fake text, audio, or video content is poised to substantially automate this goal, especially when it comes to impersonations of particular individuals. Four precedents suggest as much.

First, GPT-2 and GPT-3, natural language processing systems developed by leading research lab OpenAI, can write cogent and convincing text on their own using a generative system known as a transformer. GPT-2 has already demonstrated the capability to produce propaganda based on specific extremist ideologies.[60] GPT-3 can go much further and generate text that not only sounds realistic, but that also mimics the style of a particular author—and it can do this after seeing only just a few examples of that author's writing.[61] These capabilities have clear uses for attackers hoping to automate the crafting of spearphishing emails that could impersonate trusted accounts.

Second, generative learning techniques often improve machine translation, which itself enables more effective social engineering. For example, attackers often write phishing emails in one language and then translate them into another. As a result of poor translation, these emails often contain errors easily spotted by a native language speaker. The more machine translation improves, the more effective translated spearphishing emails will become. This could enable attackers to write lures once and translate quickly across a large number of languages.

Third, in the months preceding the 2018 United States general election, a researcher at University of Copenhagen conducted an experiment in which a machine learning algorithm produced tweets targeting politically active Twitter accounts. The algorithm, SNAP_R, used a collection of human-identified Twitter accounts as the "inspiration" for writing its own tweets, replicating the substance and sentiment of the original content. By the time Twitter suspended the researcher's account, 20 percent of people who received a direct tweet had clicked on the link; SNAP_R was originally tested and published in 2016, when researchers had an even greater success rate (66 percent) with a smaller sample size.[62] While the links in this experiment were benign, an attacker could use such technology to dupe users into visiting websites that deliver malicious code.

Fourth, in late 2019, a criminal used a generative machine learning system to replicate the voice of an UK-based energy company's CEO. Using this realistic-sounding voice, the criminal convinced an employee to wire transfer $243,000 to the attacker's bank account.[63] As technology for creating deep fakes proliferates online, attackers are sure to make use of similar tactics—impersonating information technology staff, co-workers, direct reports, and CEOs to steal passwords and deliver malicious code.

Even after delivery, the capacity for generative systems to come up with new creations that fit a particular form has implications for the command and control, pivoting, and actions on objective stages of the kill chain. At each of these stages, attackers are often most focused on accomplishing their activities within a network without arousing suspicion. Generative systems can help them do so.

Past cyber operations have attempted to blend their command and control messages into background network traffic by using rule-based automation. For example, HAMMERTOSS, as previously discussed, only sent such messages during working hours on weekdays. Generative systems can improve on this approach, more realistically disguising command and control messages in network traffic.[64] As organizations continue to move to large-scale adoption of end-to-end encryption that hides the data content, the need for obfuscation may lessen, but in esoteric environments where traffic is not encrypted—such as industrial control systems that control the power grid and much else—generative systems can help evade defenses.[65]

REINFORCEMENT LEARNING

Creating fakes that fool a human or an AI is one thing, but creating a system that can plan strategically is another. AI is increasingly showing its potential at this task; recent research demonstrates that machine learning can beat humans at increasingly complex strategic games, from board games like Go to online video games like StarCraft 2 and Dota 2. Underpinning these strategic advances is a kind of machine learning known as reinforcement learning. The strategic capability of these systems may eventually allow cyber attackers to more fully automate the reconnaissance, delivery, and pivoting stages, and may also make it possible to tailor far more destructive attacks.

A reinforcement learning system utilizes an "agent" with a reward function that it seeks to maximize. The reward function is set by the agent's programmers and corresponds to the desired outcomes. For example, when reinforcement learning was used to train an agent to play the old video game Pong, the reward function was the game's built-in scoring system. In addition to a reward function, in reinforcement learning, programmers create an environment with certain rules and give the agent the freedom to act within that environment; in the case of Pong, this was the game itself. As the agent interacts with the environment and monitors which decisions result in rewards, it learns how to achieve success. By assigning rewards to a specific outcome, but not a certain way of achieving that outcome, programmers allow the machine learning agent to experiment within the bounded environment to find the best strategy.

Reinforcement learning has two major advantages over other machine learning methods. First, it allows an agent to consider a greater set of possible actions, unconstrained by the need for human-curated training data or rules. Second, it can enable savvier strategic thinking. To better understand these advantages, consider the problem of game-playing. In a simple game like tic-tac-toe, each player has a limited set of possible moves and it is easy to list all possible states of play. Programmers can write a winning program by expressly writing a list of rules (e.g. "if X begins in the top left, place O in the center"). In a more complex game like chess, supervised learning systems can beat humans by learning the winning moves from games played by expert humans. In this context, the supervised learning algorithm is trained on thousands of moves made by grandmasters and learns to play the game by imitating the kinds of moves most likely to result in victories.

But for even more complex games like Go, these methods are not enough. It is better to create an effective AI by simply teaching the AI the parameters of the game and allowing it to play games against itself, experimenting with new strategies rather than relying on past examples. For example, AlphaZero, a reinforcement learning program developed by Google's DeepMind, was able to achieve super-

human performance at the games of chess, shogi, and Go by deploying strategies never before seen in the history of these games.[66] The importance of reinforcement learning becomes even more pronounced in video games unconstrained by a defined board and a sequential, turn-based structure. In these environments, other machine learning techniques quickly become overwhelmed with the vast number of possible choices available to them. In contrast, reinforcement learning can often allow agents to navigate even highly complex game structures, especially if used in conjunction with other supervised learning methods as a foundation.[67]

Cyber operations share some aspects of complex video games, such as an imperfect information environment. There has been considerable research, dating back to at least 2005, on how a reinforcement learning agent armed with a set of exploits could be taught to pivot strategically across a network, one of the most essential parts of the kill chain.[68] Reinforcement learning could someday allow the type of on-the-fly reasoning that only a human on the keyboard can do right now, identifying the users, systems, and data that an intruder can access. The machine learning systems showcased in this sort of research thus far, however, often have many major limitations, such as requiring complete knowledge of the target network or exact knowledge of the success rate of different exploits.[69]

Advances in reinforcement learning architectures can help overcome these problems. Even without knowledge of a target network, newer reinforcement learning systems can learn optimal paths to take to reach target machines.[70] Deep Exploit, a machine learning-based penetration testing system, combines reinforcement learning with other pieces of software to automatically perform reconnaissance, deliver malicious code to the targeted server, and pivot within the network. Reinforcement learning allows Deep Exploit to deliver malicious code to a vulnerable server on the first try, rather than deploying many vulnerabilities in rapid succession to see what works.[71]

These approaches still have notable limitations. The most significant is scalability. As the number of machines on a network or the number of possible exploits given to an AI increase, the computational cost of making the right decisions also grows. There are already some proposals meant to address this problem; one possibility is to train two reinforcement learning systems that would work in tandem, where one identifies the best next target on a network and the other identifies the best exploit to use against it, thereby dividing the computational labor required and focusing the decision-making power of each network.[72] In broad terms, however, reinforcement learning systems have displayed a greater ability to generalize from less complicated tasks to more complex ones than previous machine learning approaches, so there is little reason to think that the computational cost will prove to be insurmountable as algorithms get better and computers get faster.[73]

Although current research in reinforcement learning systems has focused on pivoting, it is possible that more advanced reinforcement learning agents may be able to automate other parts of the kill chain in the future. In complicated networks, the optimal pivoting strategy will depend on which machine attackers first compromise. A more advanced reinforcement learning system might select the ideal foothold based on the attackers' reconnaissance. Armed with traditionally-automated exploitation tools like Metasploit or Autosploit, a reinforcement learning system could in theory also iterate through a number of different attack scenarios to find the one that it judges to be most efficient, thereby strengthening the delivery and weaponization steps of the kill chain. If the reinforcement learning system had reconnaissance capabilities, it might even learn over time that certain types of reconnaissance data were more useful to mapping out an attack strategy than other types, thereby becoming capable of improving reconnaissance efforts.

Reinforcement learning systems may also prove particularly adept at sabotaging industrial control systems, a key part of the actions on objectives phase for some operations. In previous attacks on industrial control systems, attackers programmed malicious code to take certain actions on objectives, carefully crafting their attack code to have particular kinetic effects. Reinforcement learning applied to industrial control system attacks may increase the attacker's ability to fine tune the impact of their attacks, particularly with regards to severity and timing. Trained in a well-modeled environment and given the correct reward function, reinforcement learning could help attackers achieve both long-term stealthy degradation of industrial machines or incredibly destructive attacks with unfathomed speed.

In addition, reinforcement learning systems can develop the capacity to coordinate their actions, either with other machine learning systems or with humans.[74] This result has significant implications for the use of botnets. Although attackers have begun to incorporate some forms of coordination into their botnets—for instance, the aforementioned peer-to-peer C2 structure used in later versions of Conficker—there is so far no evidence that botnets are capable of making strategically coordinated decisions. However, some analysts suggest that in the future, botnets may be able to learn collectively about different types of attack strategies or delegate tasks internally without explicit instructions.[75] If such coordination is ever attained, it will likely rely on some form of reinforcement learning.

All told, reinforcement learning systems have recently been incredibly successful and improved more quickly than expert observers predicted. After DeepMind put reinforcement learning on the map with a series of early successes, many experts still doubted that computer victory over top humans at Go was possible, with some predicting that it was a decade away. DeepMind's reinforcement learning-based AlphaGo accomplished the feat less than two years later.[76] Even after this success,

observers predicted that it would be five years before machine learning systems could win at the more complex video game StarCraft II; eighteen months after that prediction, DeepMind's reinforcement learning system AlphaStar attained grandmaster status.[77] Based on this rapid improvement in capability, it seems imprudent to rule out the possibility that a usable reinforcement learning system for cyber attackers might be developed within the next few years. Such a system might find novel strategies for key parts of the kill chain that even the most advanced operators have missed.

Conclusion

We approach our conclusions with a healthy respect for what we don't know. Will current machine learning progress stall? Are other significant AI breakthroughs on the horizon? How will the underlying landscape of cyber operations change, if at all? To what degree will geopolitical factors shape state behavior in cyber operations? Even with these persistent questions, the preceding examination of the cyber kill chain allows for some preliminary assessments.

First, while machine learning advances have occurred in rapid succession in recent years, their application to offensive cyber operations remains narrow. Attackers, especially states, are generally rational and will only turn to machine learning techniques if these techniques are simpler, cheaper, or more effective than the automated tools that are already available and easy to use. Traditional methods of automation will remain in vogue for many aspects of the kill chain, though they will be supplemented in some areas by machine learning techniques. Second, the structural weaknesses of machine learning systems, including their heavy reliance on data, their complexity, and their susceptibility to adversarial attacks, will constrain their impact on cyber operations in important ways. Third, reinforcement learning offers the medium- to long-term possibility of overcoming some of these fundamental constraints, and could someday be a game-changer for offensive cyber operations. Fourth, machine learning advances will likely be dual-use, plausibly benefitting both attackers and defenders; which side benefits most will depend in large part on organizational, not technological, factors. Such organizational factors include both the level of resourcing and expertise available.

MACHINE LEARNING IN CYBER OPERATIONS IS OVERHYPED YET IMPORTANT

Machine learning currently rides a wave of ever-increasing hype. But traditional methods of automation already work extremely well for many parts of the kill chain. Even if machine learning technology continues to advance at a rapid pace in other areas, it does not follow that it will also immediately transform offensive cyber operations. For some parts of cyber operations, machine learning techniques may never matter.

For example, it is not clear that machine learning has much more to offer than traditional automation when it comes to enabling indiscriminate propagation. NotPetya amounted to little more than a well-crafted worm that relied on a common tool for stealing credentials and a powerful but known vulnerability, and yet it spread with great success across thousands of networks around the world. Given that NotPetya caused billions of dollars of damage before most IT departments were even aware they were under attack, it seems unlikely that any state seeking to create similar unbridled destruction would feel the need to develop a more complicated AI-enabled method of propagation. Similarly, although machine learning may give botnets new capabilities to coordinate strategically or learn from shared experience, the technology will not greatly impact the speed or scale at which sophisticated actors can create botnets.[78]

Even if machine learning-enabled techniques become a reality, they will still depend upon many traditional concepts and tools. New technologies seldom displace older ones entirely. For example, while helicopters and mechanized forces replaced traditional horse-mounted cavalry, bullets and rifles remain an essential part of modern warfare. Likewise, even the most advanced offensive machine learning tools will not change the underlying cyber operations environment of network protocols, operating systems, and applications.

Despite these caveats, there remains little question that attackers will turn to machine learning-enabled techniques if they provide an advantage. For this reason, machine learning advances will likely partially change the practice of cyber operations, offering new ways to exploit digital environments and the humans that use them.

Some attack techniques, such as spearphishing, are well-suited for the very near-term application of machine learning techniques. There are already natural language processing systems capable of writing convincing articles in a wide variety of styles. This technology, when trained on a corpus of organizational emails or other relevant texts, could someday be able to create convincing spearphishing campaigns. A mass spearphishing campaign empowered by machine learning would accelerate operations, increase the number of potential targets, and allow

attackers to focus on other aspects of the kill chain.[79] Even if machine learning text generation can eliminate only 80 percent of the work that crafting a spearphishing email requires, attackers are likely to see it as a major gain worth the cost of adopting new tactics.

Machine learning also seems poised to help with the process of discovering exploitable vulnerabilities, though this is a matter of continued debate. Recent advances in fuzzing technology suggest as much. While there will always be a significant role for humans in finding and exploiting software weaknesses and while autonomous fuzzers that do not use machine learning also show significant promise, the task is data-intensive and depends on the ability to recognize patterns—exactly the areas where machine learning generally excels.

The timeline of these and other advances will be governed by how fast machine learning progresses and how much attackers invest in deploying the technology. Many of the recent breakthroughs in general machine learning—including generative systems, natural language processing, and reinforcement learning—have direct applications for cyber operations. As these broader technologies continue to mature, they are poised to further alter the cyber kill chain in much the same way that traditional automation did previously, though important structural barriers remain.

STRUCTURAL FACTORS WILL CONSTRAIN MACHINE LEARNING IN CYBER OPERATIONS

In addition to the continued utility of traditional automation, three additional factors may constrain machine learning applications within cyber operations. First, many machine learning techniques, most especially supervised learning, require plentiful and well-structured training data from which to learn. For offensive cyber operations, much of that data comes from the reconnaissance portion of the kill chain. If defensive systems deny access to large amounts of the necessary data, the ability for the attackers to use machine learning diminishes.

For example, a large-scale spearphishing campaign aided by machine learning would seem to need access to emails that show patterns of usage and communication between various members of the target organization. Absent this data, attackers looking to exploit human error may fall back to more traditional spearphishing tactics. Likewise, if attackers are attempting to blend into the background noise of the victim's network, understanding the types of network traffic, volume, and usage patterns is critical. Armed with this data, attackers can build stronger command and control and exfiltration systems if stealth is a priority. But without this data, it may be infeasible to build a machine learning system that can effectively blend into the background traffic without triggering alarms.

This need for data provides an advantage for network defenders who have greater access to network traffic flows, network-connected devices, and user activi-

ty logs. Armed with more information, defenders can apply machine learning techniques to enable enhanced defenses. To have the equivalent level of insight, attackers would have to first compromise the defender's network and associated systems. Attackers thus face a classic chicken vs egg dilemma: to get the necessary data to apply some machine learning techniques, they first must compromise a network, but by compromising a network, they often gain enough of a foothold such that many machine learning techniques would be redundant or unnecessary.

Second, both defensive and offensive machine learning systems themselves remain susceptible to a range of adversarial attacks during their training phase and after deployment.[80] These attacks can cause machine learning systems to fail in unpredictable ways. During the training phase of a machine learning system, attackers can alter the training data or key features of the fully trained system by subtly changing key parameters. Attackers can also apply techniques similar to the ones used to fool image classifiers; for intruders, this technique could cause machine learning-based intrusion detection systems to misclassify malicious activity, while for network defenders it might be useful in thwarting machine learning-enabled attacks.[81]

Third, even if more effective than traditional rule-based automation, machine learning systems might be harder to implement due to the level of expertise required. Machine learning expertise is rarer than offensive cyber expertise, and the combination is rarer still. Presently, many of the significant machine learning advances have come from a relatively small pool of talented researchers and developers and, even (or perhaps especially) for governments, demand for talent far outstrips supply. China explicitly recognizes this in its Thousand Talents program, and, like several other countries, is actively recruiting proven machine learning researchers with generous compensation packages and research facilities.[82] For their part, major tech companies have successfully recruited top academic machine learning researchers, creating pockets of deep expertise. The limited talent pool further restricts the number of machine learning investment areas that can receive significant attention and resources. These commercial and government incentives to produce more general machine learning applications may mean relatively less effort will be spent on narrower offensive cyber applications, at least in the near term.[83]

As machine learning capabilities become more capable and easier to use, however, intricate expertise may be less essential. Machine learning-enabled tools for cyber operations may follow well-established technology adoption arcs in which the fruits of a major discovery are initially shared by a few before the technology is quickly commoditized and requires limited expertise for general purpose use. Many mainstream machine learning capabilities, such as image classifiers and some generative algorithms, have already been commoditized in this way.

The same may occur in the cyber realm. For example, open source machine learning-based fuzzers may hold wide appeal, much like early ready-to-use exploitation kits opened up hacking to individuals with only limited skills. This is not a hypothetical; there are already strong open-source fuzzers available, such as American Fuzzy Lop, which uses a genetic algorithm-based approach, though one that is not typically considered to be machine learning.[84] If machine learning-enabled hacking tools become widely available, are easy to use, and have high success, then attackers will have the requisite incentives and capacity to upgrade their capabilities.

REINFORCEMENT LEARNING TECHNIQUES MAY BE A FUTURE GAME CHANGER

Our conclusions thus far have been mostly short-term in their approach. Over the medium- and long-term, the situation might be radically different. There is some notable evidence that reinforcement learning will reshape many stages of the kill chain, though we cannot predict when or how quickly this transition will occur. Of the variety of machine learning techniques surveyed in this report, reinforcement learning has proven to be the most capable of devising winning strategies in many complex environments. While most research has focused on games, the underlying technology has proven to be transferable to constrained network simulations. It is unclear, however, how much data is required to construct an accurate simulation; if a large amount of data on the target is required, then to gain the required information an attacker will need direct access to the target network itself, lessening the value of the simulation.

In general, two distinct developments might accelerate reinforcement learning's impact on cyber operations. First, if in fact large real-world networks can be successfully simulated, then reinforcement learning techniques could train within these environments. Success would require simulating only key aspects of an actual network environment such as vulnerable systems, trust relationships, and network traffic. Second, if reinforcement learning techniques continue to improve their performance in very complex environments, they will be more capable of tackling challenges in offensive cyber operations. There is good evidence for the continued rapid advancement of reinforcement learning, including its success in dynamic and complex video games like StarCraft II.[85] If successful, these more broadly capable reinforcement learning systems could adjust to changing network conditions in real-time and reshape more of the kill chain than other machine learning techniques.

THE DUAL-USE NATURE OF CYBER TOOLS WILL CONTINUE

Many cyber operation tools can benefit both attackers and defenders. For example, fuzzers have long been a classic dual-use capability. Software developers and cybersecurity researchers use them to discover vulnerabilities before they can be exploited in the wild, while attackers employ them to look for coveted new

vulnerabilities that can be turned into reliable exploits. Likewise, network mapping tools provide attackers an easy way to search for would-be victims while at the same time providing network defenders a means to routinely assess the security posture of their network. Though there are theoretical debates about who these tools benefit more, in practice the advantage usually goes to the side that uses the tool most effectively. Defenders with more highly trained and resourced personnel may be able to gain advantages over attackers because of these automated tools, but less capable and well-resourced organizations may struggle to defend themselves against the same attacks.

This complexity and nuance mean that one of the most common questions at the intersection of AI and cybersecurity—will machine learning benefit attackers or defenders?—is almost certainly far too broad to be useful. If tools like fuzzers begin to rely more on machine learning, a competition between attackers and defenders will likely occur. The winner of the competition is likely to depend in large part on an individual offensive or defensive organization's ability to deploy the tools, just as is the case with traditionally automated capabilities.

The net impact of more speculative future capabilities, such as reinforcement learning techniques, will likely also depend on the comparative resourcing and expertise of attackers and defenders. For example, robust computing infrastructure is important to enable realistic network simulations critical for reinforcement learning. This computing power does not come cheap but—as it becomes more commoditized via Google Cloud, Amazon Web Services, and other offerings—it will become more readily available to a wider array of attackers. Similarly, the reinforcement learning techniques discussed earlier in this paper require a great deal of expertise to develop, even as they offer significant operational capabilities for both attackers and defenders.

Whether on offense or defense, the dual use nature of these tools creates a competition for strategic advantage. If, as we theorize, machine learning is likely to improve specific aspects of the kill chain in a way that could benefit either side, then reaping those advantages becomes an imperative for success. Neither side is likely to sit idly by and cede ground to the other. For attackers, inaction risks the possibility that machine learning tools will help find and remediate vulnerabilities, secure networks, and detect intrusions. For defenders, a failure to use the best tools available gives attackers more room to maneuver, more capabilities to deploy, and more scale to their operations.

GREAT POWERS HAVE THE MOST TO GAIN, IF THEY ADAPT PROACTIVELY

Ultimately, policymakers can only craft effective measures if they understand what machine learning can—and can't—do. In the near- to mid-term, machine learning will not fundamentally alter the cyber kill chain. The heavy data requirements of most cutting-edge machine learning systems and the requisite expertise needed to create them will make it difficult to develop useful tools. In addition, open source datasets and benchmarks have been enormously helpful for machine learning researchers in other subfields, but few similarly large representative and public datasets exist for researchers hoping to explore machine learning use cases in cyber operations. States can build up sizable datasets of their own, but doing so is a costly and time-intensive process that may further slow the utilization of machine learning systems for cyber attacks.

As a result of these substantial requirements, deploying machine learning in cyber operations means overcoming barriers to entry. If only some states have the resources and expertise to overcome these barriers, those states will become simultaneously better defended and more capable of attacking their rivals.[86] In other words, machine learning in cyber operations may be less biased toward either attackers or defenders than it will be biased toward already powerful states and organizations.

Even so, the barriers that exist today may lower before long. Just as many tools of traditional automation were eventually distributed in easily accessible packages, it seems likely that some future machine learning-enabled hacking tools will someday be widely available for even novices to use.[87] Other states, powerful or weak, may pursue asymmetric advantage using these tools to target less well-defended networks, such as state and municipal governments or critical infrastructure systems. Well-resourced states will have to compete by continuously developing better machine learning and traditionally automated systems and strengthening their defensive posture.

As a result of these dynamics, the United States has much to gain from developing new machine learning tools for cyber operations early and much to lose if it waits. Maintaining a competitive edge will require constant work on both offense and defense. It is yet one more reason that, for as nuanced, complex, and overhyped as machine learning is, it remains too important to ignore.

Endnotes

1. Clifford Stoll, *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage* (Doubleday, 1989).
2. Clifford Stoll, "Stalking the Wily Hacker," Communications of the ACM 31, no. 5 (May 1, 1988): 487.
3. Stoll, "Stalking the Wily Hacker," 488.
4. Stoll, "Stalking the Wily Hacker," 488.
5. Stoll, *The Cuckoo's Egg*, 373.
6. Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," *Leading Issues in Information Warfare & Security Research* 1, no. 1 (2011): 80. For a fuller explanation see, Ben Buchanan, *The Cybersecurity Dilemma: Hacking, Trust, and Fear Between Nations* (Oxford University Press, 2016), ch. 2-3 and Matthew Monte, *Network Attacks and Exploitation: A Framework* (Wiley, 2015).
7. Blake E. Strom et al., "Finding Cyber Threats with ATT&CKTM-Based Analytics," MITRE, June 2017, https://www.mitre.org/sites/default/files/publications/16-3713-finding-cyber-threats%20with%20att%26ck-based-analytics.pdf.
8. "Chapter 15. Nmap Reference Guide," nmap.org, https://nmap.org/book/man.html.
9. "Gaining the Advantage: Applying the cyber kill chain methodology to network defense," Lockheed Martin, https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/Gaining_the_Advantage_Cyber_Kill_Chain.pdf.
10. "The Payload Generator," Rapid7, https://docs.rapid7.com/metasploit/the-payload-generator.
11. "Auto-Exploitation," Rapid7, https://metasploit.help.rapid7.com/docs/auto-exploitation.
12. Joseph Cox, "New Tool Automatically Finds and Hacks Vulnerable Internet-Connected Devices," *VICE*, January 31, 2018, https://www.vice.com/en_us/article/xw4emj/autosploit-automated-hacking-tool.; "Heartbleed Report," shodan.io, https://www.shodan.io/report/0Wew7Zq7.
13. Ari Takanen et al., *Fuzzing for Software Security Testing and Quality Assurance*, Second Edition (Artech House, 2018).
14. MITRE, "Initial Access," MITRE ATT&CK, October 17, 2018, https://attack.mitre.org/tactics/TA0001/.
15. MITRE, "Initial Access."
16. Ben Buchanan, *The Hacker and the State: Cyber Attacks and the New Normal of Geopolitics* (Harvard University Press, 2020), 290–93. Renee Dudley, "Innovations; Managed Service Providers Are Ransomware Hackers' New Gold Mine," *Houston Chronicle*, September 23, 2019.
17. "2019 Data Breach Investigations Report," Verizon, May 2019, 13, 25, https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf.
18. "Spearphishing for Information," MITRE ATT&CK, October 17, 2018, https://attack.mitre.org/techniques/T1397/.
19. Buchanan, *The Hacker and the State*, 218.
20. Buchanan, *The Hacker and the State*, 219.
21. Chris Doman, "The First Cyber Espionage Attacks: How Operation Moonlight Maze Made History," Medium, July 7, 2016, https://medium.com/@chris_doman/the-first-sophistiated-cyber-attacks-how-operation-moonlight-maze-made-history-2adb12cc43f7.
22. The Rendon Group, "Conficker Working Group: Lessons Learned" (Department of Homeland Security under Air Force Research Laboratory Contract No. FA8750-08-2-0141., January 2011), https://web.archive.org/web/20190205162902/http://www.confickerworkinggroup.org/wiki/uploads/Conficker_Working_Group_Lessons_Learned_17_June_2010_final.pdf.
23. "USBStealer," MITRE ATT&CK, https://attack.mitre.org/software/S0136/.

24. "HAMMERTOSS: Stealthy Tactics Define a Russian Cyber Threat Group," FireEye, July 2015, https://www2.fireeye.com/rs/848-DID-242/images/rpt-apt29-hammertoss.pdf.
25. It is possible to engage in privilege escalation without the goal of pivoting, so some versions of the kill chain treat privilege escalation as a distinct step. However, privilege escalation is often required to pivot to more valuable machines, and automated tools already exist that can perform privilege escalation very well, as discussed below. This report therefore links privilege escalation and pivoting together and treats them as one in order to focus on the ways in which machine learning will alter the kill chain in the future.
26. Tim MalcomVetter and Swetha Prabakaran, "Internal Spearphishing," MITRE ATT&CK, October 22, 2019, https://attack.mitre.org/techniques/T1534/.
27. William F. Lynn, "Defending a New Domain," *Foreign Affairs*, Sep-Oct 2010, p. 97.
28. Andy Greenberg, "He Perfected a Password-Hacking Tool—Then the Russians Came Calling," *WIRED*, November 9, 2017, https://www.wired.com/story/how-mimikatz-became-go-to-hacker-tool/.
29. Alexander Gostev, "Agent.Btz: A Source of Inspiration?," SecureList 12, no. 3 (2014).
30. Mark Bowden, "The Worm That Nearly Ate the Internet," *The New York Times*, June 29, 2019, https://www.nytimes.com/2019/06/29/opinion/sunday/conficker-worm-ukraine.html.
31. "The Odd, 8-year Legacy of the Conficker Worm," We Live Security (ESET, November 21, 2016), https://www.welivesecurity.com/2016/11/21/odd-8-year-legacy-conficker-worm/.
32. Thomas Rid, "Cyber War Will Not Take Place," *Journal of Strategic Studies* 35, no. 1 (February 1, 2012): 5–32.
33. "Update to the IP Commission Report, The Theft of American Intellectual Property: Reassessment of the Challenge and United States Policy" (The National Bureau of Asian Research, February 2017).
34. Garrett M. Graff, "How the US Forced China to Quit Stealing—Using a Chinese Spy," *WIRED*, October 11, 2018, https://www.wired.com/story/us-china-cybertheft-su-bin/.
35. Andy Settle, Nicholas Griffin, and Abel Toro, "Monsoon—Analysis of an APT Campaign," Forcepoint, 2016, https://www.forcepoint.com/sites/default/files/resources/files/forcepoint-security-labs-monsoon-analysis-report.pdf.; Vicky Ray and Kaoru Hayashi, "New Malware 'Rover' Targets Indian Ambassador to Afghanistan," Palo Alto Networks, February 29, 2016, https://unit42.paloaltonetworks.com/new-malware-rover-targets-indian-ambassador-to-afghanistan/.
36. Buchanan, *The Hacker and the State*, 271–78.
37. "Resource Hijacking," MITRE ATT&CK, October 10, 2019, https://attack.mitre.org/techniques/T1496/.
38. Most analysts consider only five operations to have targeted industrial control systems: Stuxnet, the 2015 and 2016 Ukrainian power grid outages, as well as two other pieces of software known as HAVEX and BLACKENERGY 2. Both of these final two were primarily espionage tools that could extract information about an industrial control system but were never used to actually cause any physical damage, meaning that only three operations have yet been observed actually causing kinetic destruction. See "CRASHOVERRIDE: Analysis of the Threat to Electric Grid Operations," Dragos, June 13, 2017, 8–11, https://dragos.com/wp-content/uploads/CrashOverride-01.pdf.
39. Kevin E. Hemsley and Ronald E. Fisher, "History of Industrial Control System Cyber Incidents," Department of Energy, December 2018, https://www.osti.gov/servlets/purl/1505628.
40. "CRASHOVERRIDE: Analysis of the Threat to Electric Grid Operations," 16–21.
41. "CRASHOVERRIDE: Analysis of the Threat to Electric Grid Operations," 4.; Anton Cherepanov, "WIN32/INDUSTROYER: A New Threat for Industrial Control Systems," ESET, June 12, 2017, 7, https://www.welivesecurity.com/wp-content/uploads/2017/06/Win32_Industroyer.pdf.
42. Cherepanov, "WIN32/INDUSTROYER," 15.
43. Ron Tolido, Anne-Laure Thieullent, Geert van der Linden, Allan Frank, Luis Delabarre, Jerome Buvat, Jeff Theisler, Sumit Cherian, Yashwardhan Khemka, "Reinventing Cybersecurity with Artificial Intelligence: The New Frontier in Digital Security" (Capgemini Research Institute, July 11, 2019), https://www.capgemini.com/wp-content/uploads/2019/07/AI-in-Cybersecurity_Report_20190711_V06.pdf; G. Apruzzese et al., "On the Effectiveness of Machine and Deep Learning for Cyber Security," in 10th International Conference on Cyber Conflict (CyCon), 2018, 371–90.
44. "Common Vulnerability Scoring System Version 3.1: User Guide," FIRST, https://www.first.org/cvss/user-guide.

45. Jay Jacobs et al., "Exploit Prediction Scoring System (EPSS)," ArXiv [Cs.CR] (August 13, 2019), arXiv, http://arxiv.org/abs/1908.04856.
46. Jay Jacobs et al., "Improving Vulnerability Remediation Through Better Exploit Prediction," 2019 Workshop on the Economics of Information Security (2019), https://weis2019.econinfosec.org/wp-content/uploads/sites/6/2019/05/WEIS_2019_paper_53.pdf.
47. Tenable Team, "Security Teams: What You Need to Know About Vulnerability Response," Tenable, Inc., July 3, 2019, https://www.tenable.com/blog/security-teams-what-you-need-to-know-about-vulnerability-response.
48. Jacobs et al., "Exploit Prediction Scoring System (EPSS)."
49. Ondrej Kubovič, Juraj Jánošík, and Peter Košinár, "Machine Learning Era in Cybersecurity: A Step Toward a Safer World or the Brink of Chaos?," ESET, 6, 9–11.
50. Those familiar with current trends in machine learning may associate the term "adversarial" with generative adversarial networks. These are a more precisely defined type of machine learning architecture, and while they are an important type of adversarial system, our definition is a broader one that is meant to include any AI created in order to thwart some other system, whether or not that other system is also an AI. See our discussion on GANs below.
51. For a discussion of other genetic algorithm-based fuzzers, see G. Liu et al., "Vulnerability Analysis for X86 Executables Using Genetic Algorithm and Fuzzing," in 2008 Third International Conference on Convergence and Hybrid Information Technology, vol. 2, 491–97. For an example of an AI fuzzer that instead relies on deep reinforcement learning, see Konstantin Böttinger, Patrice Godefroid, and Rishabh Singh, "Deep Reinforcement Fuzzing," ArXiv [Cs.AI] (January 14, 2018), arXiv, http://arxiv.org/abs/1801.04589. For an example of the high success rate of AI fuzzers, see William Blum, "Neural Fuzzing: Applying DNN to Software Security Testing," *Microsoft Research Blog*, November 13, 2017, https://www.microsoft.com/en-us/research/blog/neural-fuzzing/.
52. Patrice Godefroid, Hila Peleg, and Rishabh Singh, "Learn&Fuzz: Machine Learning for Input Fuzzing," ArXiv [Cs.AI] (January 25, 2017), arXiv, http://arxiv.org/abs/1701.07232.
53. Alexy Bhowmick and Shyamanta M. Hazarika, "Machine Learning for E-Mail Spam Filtering: Review, Techniques and Trends," ArXiv [Cs.LG] (June 3, 2016), arXiv, http://arxiv.org/abs/1606.01042.
54. Gary Marcus, "Deep Learning: A Critical Appraisal," ArXiv [Cs.AI] (January 2, 2018), arXiv, http://arxiv.org/abs/1801.00631.
55. Christian Szegedy et al., "Intriguing Properties of Neural Networks," ArXiv [Cs.CV] (December 21, 2013), arXiv, http://arxiv.org/abs/1312.6199.
56. Anirban Chakraborty et al., "Adversarial Attacks and Defences: A Survey," ArXiv [Cs.LG] (September 28, 2018), arXiv, http://arxiv.org/abs/1810.00069. Zilong Lin, Yong Shi, and Zhi Xue, "IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection," ArXiv [Cs.CR] (September 6, 2018), arXiv, http://arxiv.org/abs/1809.02077. Abhijeet Katte, "Here's The Truth, GANs Easily Can Fool Intrusion Detection Systems," Analytics India Magazine, September 13, 2018, https://analyticsindiamag.com/heres-the-truth-gans-easily-can-fool-intrusion-detection-systems/.
57. Weilin Xu, Yanjun Qi, and David Evans, "Automatically Evading Classifiers: A Case Study on PDF Malware Classifiers," in Proceedings 2016 Network and Distributed System Security Symposium (Network and Distributed System Security Symposium, Reston, VA: Internet Society, 2016), https://doi.org/10.14722/ndss.2016.23115. For a deep learning approach to antivirus detection, see also Hyrum S. Anderson et al., "Learning to Evade Static PE Machine Learning Malware Models via Reinforcement Learning," ArXiv [Cs.CR] (January 26, 2018), arXiv, http://arxiv.org/abs/1801.08917. for an approach that used a genetic algorithm but with the goal of getting antivirus to mislabel the type of attack (rather than to fail to notice the attack itself) in order to create a time-consuming red herring for defenders, see Alejandro Calleja et al., "Picking on the Family: Disrupting Android Malware Triage by Forcing Misclassification," *Expert Systems with Applications* 95 (April 1, 2018): 113–26.
58. A few such models include recurrent neural networks (a broad category which includes some of the other models on this list), long short-term memory models, gated recurrent units, sequence2sequence models, and transformers. Other older architectures, such as Markov chains, have been able to produce very limited sequential content for some decades, but have in recent years become far more powerful.

59. Alec Radford et al., "Better Language Models and Their Implications," OpenAI, February 14, 2019, https://openai.com/blog/better-language-models/.; Ye Jia and Ron Weiss, "Introducing Translatotron: An End-to-End Speech-to-Speech Translation Model," *Google AI Blog*, May 15, 2019, https://ai.googleblog.com/2019/05/introducing-translatotron-end-to-end.html.
60. Irene Solaiman, Jack Clark, and Miles Brundage, "GPT-2: 1.5B Release," OpenAI, November 5, 2019, https://openai.com/blog/gpt-2-1-5b-release/.
61. Tom B. Brown et al., "Language Models Are Few-Shot Learners," ArXiv [Cs.CL] (May 28, 2020), arXiv, http://arxiv.org/abs/2005.14165.
62. Michael Bossetta, "A Simulated Cyberattack on Twitter: Assessing Partisan Vulnerability to Spear Phishing and Disinformation Ahead of the 2018 U.S. Midterm Elections," ArXiv [Cs.SI] (November 14, 2018), arXiv, http://arxiv.org/abs/1811.05900. John Seymour and Philip Tully, "Weaponizing Data Science for Social Engineering: Automated E2E Spear Phishing on Twitter," Black Hat USA 37 (2016): 1–39.
63. Catherine Stupp, "Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case," *The Wall Street Journal*, August 30, 2019, https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402.
64. Adriel Cheng, "PAC-GAN: Packet Generation of Network Traffic Using Generative Adversarial Networks," in 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (ieeexplore.ieee.org, 2019), 0728–34.; Markus Ring et al., "Flow-Based Network Traffic Generation Using Generative Adversarial Networks," ArXiv [Cs.NI] (September 27, 2018), arXiv, http://arxiv.org/abs/1810.07795.
65. Some have posited that industrial control system traffic is harder to emulate. See, for instance, Matti Mantere, Mirko Sailio, and Sami Noponen, "Network Traffic Features for Anomaly Detection in Specific Industrial Control System Network," *Future Internet 5*, no. 4 (September 25, 2013): 460–73. Cheng Feng et al., "A Deep Learning-Based Framework for Conducting Stealthy Attacks in Industrial Control Systems," ArXiv [Cs.CR] (September 19, 2017), arXiv, http://arxiv.org/abs/1709.06397.
66. David Silbert et al., "AlphaZero: Shedding New Light on Chess, Shogi, and Go," DeepMind, December 6, 2018, https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go. AlphaZero is an evolution of an older program named AlphaGo, which successfully bested a top-ranked human player at Go while still relying on historical data and supervised learning techniques. Unlike AlphaGo, AlphaZero uses a strictly reinforcement-learning-based approach and has been able to reliably beat its predecessor AI.
67. Oriol Vinyals et al., "Grandmaster Level in StarCraft II Using Multi-Agent Reinforcement Learning," *Nature 575*, no. 7782 (November 2019): 350–54.
68. See, for instance, Mark Boddy et al., "Course of Action Generation for Cyber Security Using Classical Planning," in Proceedings of the Fifteenth International Conference on International Conference on Automated Planning and Scheduling, ICAPS'05 (AAAI Press, 2005), 12–21. Joerg Hoffmann, "Simulated Penetration Testing: From 'Dijkstra' to 'Turing Test++,'" in Twenty-Fifth International Conference on Automated Planning and Scheduling, 2015, https://www.aaai.org/ocs/index.php/ICAPS/ICAPS15/paper/viewPaper/10495.
69. See Jonathon Schwartz and Hanna Kurniawati, "Autonomous Penetration Testing Using Reinforcement Learning," ArXiv [Cs.CR] (May 15, 2019), 21–24, arXiv, http://arxiv.org/abs/1905.05965.
70. Schwartz and Kurniawati, "Autonomous Penetration Testing."
71. Isao Takaesu, "Deep Exploit," GitHub, February 17, 2018, https://github.com/13o-bbr-bbq/machine_learning_security/tree/master/DeepExploit.
72. This possibility is mentioned in Carlos Sarraute, Olivier Buffet, and Jörg Hoffmann, "POMDPs Make Better Hackers: Accounting for Uncertainty in Penetration Testing," in Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI12 (AAAI Press, 2012), 1816–24. as well as Schwartz and Kurniawati, "Autonomous Penetration Testing Using Reinforcement Learning," 70.
73. AlphaZero, for instance, was developed not for Go specifically but was created with the capability to learn and master any 2-person, zero-sum game with perfect information symmetry between players. See David Silver et al., "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm,"

ArXiv [Cs.AI] (December 5, 2017), arXiv, http://arxiv.org/abs/1712.01815. Note that this generalization ability is not limitless: cutting-edge reinforcement learning research tends to rely on far more specific types of reinforcement learning systems, which have only extremely narrow applications and do not generalize well (or do not yet generalize well).

74. See, for instance, Max Jaderberg et al., "Capture the Flag: The Emergence of Complex Cooperative Agents," DeepMind, May 30, 2019, https://deepmind.com/blog/article/capture-the-flag-science.
75. Kubovič, Jánošík, and Košinár, "Machine Learning Era in Cybersecurity: A Step Toward a Safer World or the Brink of Chaos?," 7.
76. Alan Levinovitz, "The Mystery of Go, the Ancient Game That Computers Still Can't Win," WIRED, May 12, 2014, https://www.wired.com/2014/05/the-world-of-computer-go/.
77. Tom Simonite, "Google's AI Declares Galactic War on StarCraft," *WIRED*, August 9, 2017, https://www.wired.com/story/googles-ai-declares-galactic-war-on-starcraft-/.
78. It is partially for this reason that this paper does not discuss the Internet of Things at any length. Though the proliferation of largely unsecured IoT devices does present a very real worry for cybersecurity, it is not clear that machine learning will significantly impact the way that attacks on IoT devices are carried out, and we therefore do not dwell much on the possibility here.
79. One major kink is that although state-of-the-art text generators can produce convincing text, they often struggle to terminate plausibly and have a tendency to ramble rather than elegantly concluding their messages.
80. Elham Tabassi et al., "A Taxonomy and Terminology of Adversarial Machine Learning" (National Institute of Standards and Technology, October 30, 2019), https://doi.org/10.6028/NIST.IR.8269-draft.
81. One promising line of research uses machine learning techniques to detect possible spearphishing attempts and natural language processing capabilities to engage autonomously with attackers. Adam Dalton et al., "The Panacea Threat Intelligence and Active Defense Platform," ArXiv [Cs.CL] (April 20, 2020), arXiv, http://arxiv.org/abs/2004.09662.
82. Wm C. Hannas and Huey-Meei Chang, "China's Access to Foreign AI Technology: An Assessment" (Center for Security and Emerging Technology, September 2019), 11, https://cset.georgetown.edu/wp-content/uploads/CSET-Chinas-Access-to-Foreign-AI-Technology-2.pdf.
83. Cindy Martinez and Micah Musser, "U.S. Demand for Talent at the Intersection of AI and Cybersecurity," (Center for Security and Emerging Technology, November 2020), https://cset.georgetown.edu/wp-content/uploads/CSET-U.S.-Demand-for-Talent-at-the-Intersection-of-AI-and-Cybersecurity-1.pdf.
84. Michal Zaleski, "American Fuzzy Lop," Github, https://github.com/google/AFL.
85. "AlphaStar: Grandmaster Level in StarCraft II Using Multi-Agent Reinforcement Learning," DeepMind, October 30, 2019, https://deepmind.com/blog/article/AlphaStar-Grandmaster-level-in-StarCraft-II-using-multi-agent-reinforcement-learning.
86. For more on this dynamic in cyber operations generally, see Ben Buchanan, "Nobody But Us: The Rise and Fall of the Golden Age of Signals Intelligence" (Hoover Institution, 2017), https://www.hoover.org/sites/default/files/research/docs/buchanan_webreadypdf.pdf.
87. Ben Buchanan, "The Life Cycles of Cyber Threats," *Survival* 58, no. 1 (February 2016), https://www.tandfonline.com/doi/abs/10.1080/00396338.2016.1142093.

CSET CENTER *for* SECURITY *and* EMERGING TECHNOLOGY

CSET.GEORGETOWN.EDU | CSET@GEORGETOWN.EDU