Offensive Machine Learning Methods and the Cyber Kill Chain

Hannu Turtiainen, Andrei Costin, Alex Polyakov, and Timo Hämäläinen

Abstract Cyberattacks are the "new normal" in the hyper-connected and all-digitized modern world, as breaches, denial-of-service, ransomware, and a myriad of other attacks occur every single day. As the attacks and breaches increase in complexity, diversity, and frequency, cybersecurity actors (both ethical and cybercrime) turn to automating these attacks in various ways and for a variety of reasons, including the development of effective and superior cybersecurity defenses. In this chapter, we address innovations in machine learning, deep learning, and artificial intelligence within the offensive cybersecurity fields. We structure this chapter inline with the Lockheed Martin's Cyber Kill Chain taxonomy in order to cover adequate grounds on this broad topic, and occasionally refer to the more granular MITRE ATT&CK taxonomy whenever relevant.

1 Introduction

Machine learning (ML), deep learning (DL), reinforcement learning (RL), and artificial intelligence (AI) technologies [1] have been among the most tackled and even controversial talking points in information technology for the past decade. Some view these technologies as a panacea for cybersecurity problems. Others are, how-

Hannu Turtiainen
University of Jyväskylä, Finland, e-mail: hannu.ht.turtiainen@jyu.fi

Andrei Costin
University of Jyväskylä, Finland, e-mail: ancostin@jyu.fi

Alex Polyakov
Adversa.AI, Israel, e-mail: alex@adversa.ai

Timo Hämäläinen
University of Jyväskylä, Finland, e-mail: timoh@jyu.fi

[1] Due to the breadth and depth of various definitions related to ML/DL/RL/AI and cybersecurity, throughout this paper we will refer to these technologies simply as MLsec.

ever, more sceptical yet, and see additional issues arising, mainly due to the bias and non-transparent nature of such technologies, which is unfortunately common and the norm in many present implementations. As these technologies are slowly being adopted within the cybersecurity tools space, the attacking side has also been productive and advancing in this matter.

MLsec technologies bring several crucial benefits for more efficient and productive (ethical) hacking: the increase of capable actors, extensive increase in the possible attack frequency, and an increase in the variety of possible targets. These feats are possible due to the considerable scalability of MLsec systems and the ease of deployment in a cost-efficient and time-saving manner when compared to the classical "army of skillful humans." In addition, target selection benefits greatly from MLsec technologies as larger numbers of diverse data can be gathered and analyzed in almost real-time [11].

Complex systems such as smart Cyber-Physical Systems (sCPS) are particularly vulnerable to MLsec attacks due to their complex inter-connectivity via smart sensor networks, clouds, and the Internet of Things (IoT) [44]. Kaloudi and Li [44] compare the possible attacks to these systems to the Butterfly Effect, in which small, seemingly inconsequential incidents may have large, catastrophic effects, such as the global effect of the infamous Mirai botnet [6].

As MLsec methods are being used more and more in the cybersecurity space, they may also cause further unforeseen issues. ML methods rely on predetermined features, which can be biased or otherwise compromised or unrepresentative of the current threat landscape. DL and AI methods, although better suited for constant learning, can be "poisoned" (i.e., undetectable alterations with malicious intents) during training with poisoning samples to render them ineffective or considerably "crippled." They can also be fooled at the inference stage with the help of so-called adversarial examples, which can bypass AI-driven systems [51]. Researchers agree that as MLsec models get increasingly more efficient in their function, and as more and more data are generated, the MLsec tools will play a vital role in future cybersecurity "wars." From the (ethical) attacker's perspective, the reconnaissance, target infiltration, data exfiltration, and privilege escalation stages seem particularly advantaged with the addition of MLsec tools and techniques [89].

At the same time, offensive MLsec can pose serious threats to organizations' cybersecurity. In comparison to human attackers, offensive MLsec improves coverage, as the MLsec methods can scale virtually without limits. The speed of execution of such attacks is also significantly increased, thus considerably reducing the time available to defenders to respond. The success rate of the attacks can increase with the use of MLsec models as more data can be processed with each step and therefore, more knowledge is available to the attacker [58].

This paper is organized as follows. In Section 2, we exhibit the Cyber Kill Chain phases and present relevant MLsec related works respectively. Section 3 introduces ideas and suggests works on how ML and AI can be used against MLsec models. Finally, we conclude in this chapter in Section 4.

2 Cyber Kill Chain phases

MLsec technologies are already entrenched in the cyberattack space in most of the phases of the Cyber Kill Chain. Today, MLsec methods are predominantly applied in the reconnaissance, infiltration, lateral movement, and act on objectives parts of cyberattacks. However, advances in the MLsec field bring the technologies to other phases and parts of the cyberattack taxonomy thanks to more use-cases being developed and showcased [12].

In this work, we try to map the current state-of-the-art in offensive MLsec by employing the seven steps of Lockheed Martin's original Cyber Kill Chain [54] (see Figure 1). We also highlight, whenever relevant, a more granular and differing version of the Cyber Kill Chain, presented as the MITRE ATT&CK framework [83]. Even though the MITRE ATT&CK kill chain framework presents 14 steps for more granularity, we feel that the work we present is not large enough in scope to warrant the use of the more granular frameworks at this time.

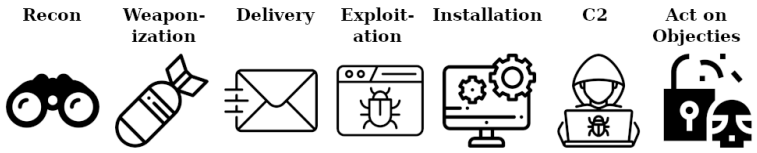

Fig. 1 The main phases of the traditional Cyber Kill Chain (as introduced by Lockheed Martin [54]).

2.1 Reconnaissance

Reconnaissance is the first step in infiltrating a target, and can be broken down into three steps: identifying, selecting, and profiling the target. Target identification is a passive reconnaissance method, meaning that the target is unaware of the proceedings. It involves checking the domain names and web assets of the potential target from Internet address registries. Target profiling can be done on system and social levels. Social profiling is also a passive method, and it means crawling through social networks and public websites and documents. System profiling, on the other hand, is usually an active approach, meaning that the target is most likely aware of it. It includes fingerprinting and scanning the open ports and services on the target systems [88]. MITRE ATT&CK [83] convey similar steps and techniques in the reconnaissance phase. Profiling the victim through all possible sources (closed/open sources, domains, websites, phishing, etc.) is crucial. Active scanning may also be necessary in order to adequately fingerprint and identify the network target.

In order to successfully infiltrate a target without a zero-day exploit, many data about the target should be gathered. However, these data are meaningless without proper analysis and the transformation from data to real actionable insights. Algorithmic data gathering such as web crawling and automatic social engineering are fast and produce an abundance of data to be analyzed.

MLsec opportunities in the reconnaissance phases can be summarised as targeting AI. The tasks of MLsec models in this phase are identifying the best targets and learning their standard behavior [44]. As MLsec methods work well in pattern detection, common vulnerabilities in, for example, front-end PHP code can be found by tokenizing the entire available front-end code and feeding them through a model, therefore extracting matching patterns. Database enumeration through numerous database function executions can also be a suitable task for MLsec technologies [80].

As ML is a great tool for handling large nubmers of data, ML for information gathering/reconnaissance is a great fit. Nowadays, people expose and post a lot of information about themselves online, and the reconnaissance towards people (and not just towards information technology systems) has become more feasible. Classifiers can be used to scout out potential victims from social media. For example, one can root out any IT professionals from the potential victim list as one could make an assumption that they might not be as gullible targets as some other average or unaware users. Online service vendors such as PimEyes [65] offer this type of service. Other clustering and classification methods such as K-means, random forests, and neural networks can be employed on top of Natural Language Processing (NLP) analysis for analyzing social media posts to learn about targets. Image and object recognition can be of use, as well as social media is full of pictures. Certain objects and types of images can be of use in identifying and classifying potential targets [66].

Tools for network scanning and sniffing have been around for a long time. However, modern Software-Defined Network (SDN) implementations can prove to be too tricky for simple tools as targets may not be as clearly defined; therefore, more complex tools are required. Machine learning aids in this process in several ways. The attacker must probe the network, trigger arbitrary flows, and gather information about the network policies, rules, and filters in place. This is traditionally an arduous task; however, with ML methods, it can be made more manageable and feasible. Know Your Enemy (KYE) [15] is an attack method that provides a stealthy way of gathering valuable information about a target SDN. This intelligence can vary from network policies to security tool configuration options.

Phishing, and its variations such as vishing and smishing, are fraud attacks in which the target is contacted by the conman via some means of communication, such as email, voice phone calls, or SMS. The attackers represent themselves as a reputable entity in order to gain trust or induce some sort of reaction in the target to make them give up sensitive information, such as passwords or banking information about themselves [45]. Phishing is a common way to gain access to a target system, and protecting against it can be so resource-heavy that affects day-to-day proceedings. DeepPhish is an experimental MLsec-driven phishing URL creator. It learns effective patterns from previous attacks and adapts the URL generation strategy, which greatly increases the chances of false negatives from phishing detectors [8].

Social media is also a good venue for phishing because of its wide-spread use, although the public nature of it could prove problematic. Still, using automated and intelligent target-aware spear phishing frameworks is proving to be more productive than large-scale phishing attacks [72]. SNAP_R [71] is an AI spear phishing framework that enables target discovery and message customisation in relation to the target for Twitter phishing. Using SNAP_R, the attacker can launch automated spear phishing campaigns that seem to be comparable to large-scale manual phishing efforts [71].

Phishing detection is a common research topic in the MLsec field. Attackers can benefit from defensive research as well, because ML for detection avoidance is one of the key features of a good and successful phishing campaign. Some efforts, such as that of Khan et al. [45], experiment with a variety of algorithms and datasets, exposing strengths and weaknesses in them. Thus, this information can be used to further evolve and improve the phishing campaigns as well as the detection algorithms. All in all, it is an arms race between the attackers and the defenders.

2.2 Weaponizing

After the reconnaissance phase, the target has been selected and possible vulnerabilities, security holes, backdoors, etc. have been identified. The information gathered from the reconnaissance phase is now used to construct the payload containing a malware or a Remote Access Trojan (RAT) and the exploit in order to infiltrate the target [54, 43]. The steps during this phase in MITRE ATT&CK [83] focus on attack preparation through capability and infrastructure development but also compromising target infrastructure and accounts. Kaloudi et al. [44] sets the most important MLsec tasks for this phase as acquiring information about the standard behavior of the target and generating abnormal behavior based on that information, as well as new vulnerability discovery and payload generation. They refer to these steps as "AI-aided."

A common way to discover vulnerabilities in software is to use fuzzing. It is a method in which the input to the target (e.g., software, system) is tested with a huge variety of uncommon and carefully manipulated inputs, while the target is monitored for crashes and anomalous behavior. Machine learning can be beneficial for fuzzing in two ways: generation of new samples, and ranking of resulting crashes. If an algorithm can create smarter input for the fuzzer, it means valuable time can be saved in trying to crash the target under test. At the same time, not all crashes lead to a vulnerability discovery. Therefore, an MLsec-driven ranking algorithm can sort out the most exploitable-relevant crashes for researchers to look at first [66].

For malware creation, the way that ML is used can be thought of as counter-intuitive. As malware detection heavily uses ML models, these models can be used as testbeds for new malware. For example, the attacker takes a known malware, introduces changes, and then tests the malware against malware detectors such as VirusTotal. On the other hand, many MLsec methods can be used to create initiation

triggers for malware. For example, a sleeper malware in a system can wait for a certain facial recognition event to occur until it initializes [66], or other similar "secret knock" computer vision signals [16].

Initial access and malware delivery methods can be categorized in two types: "attacker-controlled" and "attacker-released" delivery methods. The "attacker-controlled" methods involve directly attacking the web services of the target, while the "attacker-released" methods make use of the users who have access to the target to deliver their payload. This can happen via email, USB sticks, social media interactions, and other means [54]. MITRE ATT&CK [83] proposes several methods for initial access, such as exploiting public-facing applications and services, compromised account usage, supply-chain compromise, phishing, hardware additions, insider access, etc. For MLsec methods, phishing, compromised accounts, and public services are the venues to look into.

One example of a stealthy spyware was presented by Zhang et al. [93]. They developed a proof-of-concept app that recorded users' phone calls after infiltrating an Android phone. The voice data were then used to synthesise Android voice assistant activation keyword ("OK Google") via Natural Language Processing (NLP). As the voice assistant is naturally granted high privileges in the phone, gaining such an access level is of high value. The attack might be detected by the user if it was launched in an arbitrary time; therefore, more data from the phone's sensors are collected and a more appropriate noisy attack time is selected [93]. This approach could be used to further gather information about some target, or to gain more privileges within the phone or the networks it connects to.

Spamming and phishing are old techniques for information gathering. They have become such an everyday phenomenon that today, spam and phishing filters in services are common ground. To circumvent them, the attackers must create better examples that mimic human activity. In order to train an algorithm to mimic authority, training samples are needed. As social media is public, the amount of training samples one can acquire can be vast, and therefore, the results can be surprisingly good (see Section 2.1). For emails, however, if the attacker does not have an insider within the target organization, getting examples of private email correspondence can be challenging [66].

Spoofing has seen huge leaps of improvement with MLsec contributions over the last decade. Fake pictures, voice, and videos are a real threat. Lybebird [20] showcase their audio-mimicking capabilities, while Van den Oord et al. [62] propose WaveNet, a waveform-level audio data model built around GANs. With MLsec technologies, the Internet could be filled with fakes in the not-too-distant future. As an example, voice cloning was used in a bank heist resulting in $35 million in losses [10]. Spoofing related to AI is discussed separately in Section 3.

Last but not least, numerous papers have been published over the last decade on variations of Automatic Exploit Generators (AEG) [7, 41, 42, 86, 3]. These developed tools are essentially signature-based detectors for exploits from source code and binaries. These tools employ static and dynamic analysis (some including their extensive vulnerability databases) to root out issues in the execution of the code or in the source code itself. Some tools also apply fuzzing tools to increase the bug

yield by possibly triggering unknown vulnerabilities. Grieco et al. [31] resorted to ML algorithms in order to solve the AEG task. Their goal was to make vulnerability discovery easily scalable by using light-weight analysis components and automated by utilizing MLsec capabilities in having plenty of features from a large dataset instead of hand-picked features specialized for a certain task.

2.3 Delivery

In the delivery phase, the attacker launches their operation towards the target. The intent is to deliver the malware to the target system or endpoint. In controlled delivery, the attacker is attacking directly towards the target's web services; however, in released delivery, the attacker uses an auxiliary to deliver the malware. The auxiliary can be a phishing email, USB stick, or some other means of malware transportation not via the network [54]. In the delivery phase, "AI-concealed" steps are the ability to remain undetected and conceal the attacker's malicious intent [44]. MITRE ATT&CK [83] defines delivery as the initial access phase and it features methods such as phishing, supply chain compromise, hardware additions, etc.

Although denial-of-service (DoS) attacks are thought to be quite simple attacks; common detection and prevention systems are very effective against them [49, 92]. To fool these prevention mechanisms, ML algorithms can create arbitrary differing data packets, in turn, making detection much harder to differentiate from the real traffic. An even better approach is to sniff real traffic and train a neural network to generate legitimate packets to the target, albeit in a malicious manner.

Crowd-turfing, the malicious use of crowd-sourcing services, also benefits heavily on ML. The creation of fake reviews, news, accounts, etc, is tedious work for people and can be too complex for simple algorithms. There is an abundance of training material available online for creation of high-performing crowd-turfing ML models [66].

Botnets are the backbone of some of the most well known cyberattacks such as the Mirai-botnet [6]. The use of vast resources, even unsofisticated ones such as DVRs and CCTV cameras, can be technologically devastating. Although, as the name "botnet" suggests, the features of attacks are not sophisticated, as individually utilizing system resources of the bots is an impossible task for humans. Smart botnets, such as Hivenets [56], can identify the individual nodes of the botnet and put its resources to full use, making the threat of botnets even greater [66].

The "Completely Automated Public Turing test to tell Computers and Humans Apart" (CAPTCHA) is intended to restrict malicious attack attempts on websites and other graphical user interface services by disabling the possibility for automation. However, the advancements in MLsec technology have proven standard CAPTCHAs essentially broken [17, 78]. For example, Yu et al. [91] showcased their automated approach to cracking popular open-source Python CAPTCHAs using an MLsec-based chosen-plaintext attack. Further, Alqahtani and Alsulaiman [4] set several MLsec models (especially their CNN model) against Google's reCAPTCHA [29]

with over 50% of the challenges successfully solved. With these developments, newer types of CAPTCHAs are being developed [30, 63].

2.4 Exploitation

Exploitation (or the attack and exploit execution) stage means gaining access to the target system using the vulnerabilities and weaknesses found earlier [54]. There are numerous ways this can be achieved in classical scenarios. MITRE ATT&CK [83] proposes ways such as using valid user credentials, API abuse, malicious scheduled tasks, container deployment, system services, etc. MLsec methods can aid in this step, for example, by ensuring proper environment, selecting proper execution time, and automating execution ("AI-automated") [44]. However, currently this topic has not gained much academic attention as there has been plenty of work done in more influential topics in the Weaponizing (see Section 2.2), Delivery (see Section 2.3), and Installation (see Section 2.5) stages, where the exploits and payload are in the spotlight.

This phase can also be a representative example of how tools meant for security may have implications against it. Bleeding edge technology such as Darktrace [19], a real-time security incident AI analyst modeled on the human immune system, is one example of this. It is supposed to detect deviation from the norm in an ICS environment. However, such tools, if plausible to enable in the target system, could benefit the attackers greatly in defining the norm of the system operation and possibly being used as an alert system for testing the limits; for example, data exfiltration methods (see Section 2.7).

2.5 Installation

Installation phase means that the attacker has reached a state where they are in control of some part of the system. Typically, this stage involves installing a persistent backdoor to maintain the access to the server for an extended period of time. Methods can vary from webshells, system backdoors, adding services and making malware appear as standard system files [54]. This stage is referred to as "AI evolved" by Kaloudi et al. [44] and the MLsec task is the (ideally stealth) self-propagation of the malware and persistence. MITRE ATT&CK [83] has this step divided into several more specific steps. Persistence requires the attacker to establish some ground on the target system by acquiring accounts or setting up some services for further use. Privilege escalation means gaining elevated access in the system, endpoint, or service, where the attack has originated. This can be achieved by using exploits or by the means of gaining superior credential access in the system. Defence evasion implies obfuscating or masquerading the malicious behavior in the target entity and fooling defence systems. After the initial installation, the attackers might require

access to other systems or endpoints in the network; thus, network discovery and lateral movement in the network are required.

Installation can be a difficult phase for the attacker. Operating and security systems usually have systems in place where unwanted installations can be detected and prevented. MLsec-based evasive malware, such as DeepLocker [79], make this task easier by hiding behind benign, everyday software and only activating the payload once the conditions are met (e.g., precise target are reached). These conditions are determined by the deep neural network that is trained for the target specifications, as deep neural networks are essentially black boxes, and DeepLocker is able to hide the trigger conditions behind such smokescreens [79].

Another way to bypass MLsec-based malware detectors is to create malware examples that are benign to the detector itself. MalGAN [40] is an algorithm that generates adversarial malware examples, using the black-box detection algorithm employed in the defending end, in order to bypass it undetected [40]. The author's algorithm shows great promise and further fuels the MLsec technology war between the offensive and defensive actors. Vidal et al. [55] presented two MLsec models to masquerade detection bypassing. The models were based on action pruning and noise generation, and they were highly successful.

The accessed entity in the target's system is usually not the intended target, or if it originally was, a more attractive or valuable target may present itself when the first target is penetrated. Therefore, privilege escalation and lateral movement are required. Brute-force password guessing via generated password rules is resource intensive and time consuming and with proper security policies in place, it might be redundant as well. To address these issues, Hitaj et al. [38] presented PassGAN, a MLsec-based method for password guessing. The idea behind it is to generate better samples, with GAN models, based on password characteristics and structures, thus, being able to guess between 51% and 73% additional unique passwords in comparison to HashCat run with standard password generation rules and dictionary attacks.

Along with gaining access through MLsec-driven credential cracking [47], privilege escalation can be established through exploit use as well. Individually checking all the services and software within the breached target is tedious and stagnant, thus, requiring automation. However, testing every possible option is again quite a noisy and resource intensive procedure. To further develop this strategy (for white hat penetration testing purposes), MLsec-based smarter penetration testing frameworks have been researched.

Network discovery within the target's network may be noisy and alert the defenders easily. On the other hand, passive network discovery may be slow and yield less results. Network traffic analysis can be used to accomplish this task and discover hosts and services within a network automatically, thus, generating knowledge about the network, potential additional targets, and possible data exfiltration channels as well as establishing information about the standard network activity. The attackers may use methods meant for defenders in their activities. Works from Pacheco et al. [64] and Shafiq et al. [73] are some of the few examples of network traffic classi-

fication by capturing traffic and using MLsec models to refine it and gaining higher understanding of it.

Automated exploitation is a less-researched topic in the MLsec field possibly due to the high interpretability and high number of impacting factors in the decision. Still, Valea and Oprisa [85] used data from the popular penetration testing training platform Hack The Box [37] to train their MLsec models. The intention of their framework was to obtain root shell on the breached machine by first scanning the target, then search for vulnerabilities for the found services, and lastly, exploit the vulnerability to obtain root shell. Within their tests on the Hack The Box platform, they were able to successfully exploit seven out of ten test machines, clearly showcasing the ability to automatically exploit machines with known exploits with great success when using MLsec approaches. On the other hand, Fang et al. [22] developed an automatic MLsec-based exploitability metrics model for determining the real-world usability of recent exploits, which can be used to minimize the time wasted on lesser exploits with poor success rate.

As network data are often encrypted even in local networks, Man-in-the-Middle (MITM) attacks by themselves are less useful. Tools such as SSLStrip and SSLStrip2 can be used to set plain text data transfer between the victim and the attacker while preserving the perceived encrypted data transfer between the victim and the server (between the attacker and the server in reality). However, this type of attack is noticeable and can be hindered by the use of HTTP Strict Transport Security (HSTS). Nonetheless, a MITM attack can be useful even with encrypted data. Al-Hababi and Tokgoz [2] demonstrated how MLsec models can be trained to classify encrypted network traffic and determine which applications or services are being used. The attackers can use this technique for example to establish standard network behavior, or to prospect possible C2 solutions (see Section 2.6).

Keylogging is an old technique that can be used legitimately, for example, as a parental control in a child's computer or as an attack software, for example, to steal credentials. Essentially, if the tool can be installed on a desktop endpoint, the keylogger saves all the keyboard inputs and sends them to the attacker. The attackers can use this information mostly to yield credentials from the user or to gather personal or company secrets. To combat this, random noise can be inserted to the internal memory of the keyboard in order to obfuscate real data [90]. However, MLsec models can be used to counter this defence. Lee and Yim [50] proposed MLsec models to filter out the noise generated by the defence tools and yield the actual keystrokes from the keylogger. Their best model resulted in up to 96.2% accuracy, making the attack significantly more feasible than without such filtering.

2.6 Command and Control

As the attack and the malware has established a beachhead in the target system, the intent is to establish a Command-and-Control (C2 or C&C) infrastructure to remotely observe and manipulate the target. The C2 communication channel can be

established in many different ways to mask it as standard traffic [54]. Kaloudi et al. [44] defines tasks for the C2 phase as "AI multilayered." MLsec is required to self-destruct the initial access processes and set up new layers for the campaign to move forward towards its end goal. MITRE ATT&CK [83] characterizes this step with means of creating the C2 network. There are many means to this end such as proxies, non-standard port use, communication through removable media, encrypted channels, protocol tunneling, etc.

Chung et al. [14] proposed a self-learning malware targeting Cyber-Physical Systems (CPS), as their security is often overlooked in the overall system. At first, their malware gathers data from the target CPS and learns patterns. The data are then used with statistical methods to select a strategy of attack that has the lowest probability of detection and the highest chance of success by emulating random incidents in the CPSes. The malware injects the CPS logic with their malicious code to start the attack. The malfunctioning part of the system in many cases causes a cascading effect that affects the system as a whole [14].

One way to establish an undetected C&C connection is to use Domain Generation Algorithms (DGAs). DGAs are typically used in many malware types; however, there are detectors set up by the network administrators in order to root out these fake domain names. In that vein, DeepDGA [5] is a DL architecture for creating hard-to-detect pseudo-random domains, for example, for C2 purposes. DeepDGA uses the GAN algorithm to generate domain names, which are sufficiently effective and efficient in confusing the MLsec-based DGA classifiers [5].

2.7 Actions on objectives

The attackers have reached the final stage of the operation and their goal is accomplished. The attackers can now choose what to do next. They can, for example, destroy systems, exfiltrate and collect data and credentials, overwrite or corrupt data, escalate privileges further, and move laterally across the environment [54]. In this stage, the real damage can be done. MLsec tasks for this can be described as "AI massive." These tasks can vary from data destruction to victim extortion [44]. In this context, the collection, exfiltration, and impact steps from the MITRE ATT&CK [83] fit here. Data collection can be as effective as the attackers' access and lateral movement allow. The data can be collected from databases, emails, microphones, and cloud servers, and via screen, keyboard, or clipboard capture, with Man-in-the-Middle attacks, etc. After data capture, they need to be exfiltrated safely and unnoticed from the victim's system. At present, there are many ways to exfiltrate data, including unconventional methods covering air-gapped networks [35, 34, 36, 33]. Data exfiltration can be achieved by obfuscating it, timing it to fit regular behavior, done through a physical medium, set up via a web service, conducted through the C2 channel, as well as employing other sophisticated steps. MITRE defines impact steps as the destructive phase in the scenario. Possible impact of the attack can vary greatly and can include storage wipes, defacement, account removal, data corruption, DoS,

data destruction and theft, and resource hijacking. At this time, we were unable to find any major work on MLsec data exfiltration methods. In theory, MLsec models could be a decidedly superior choice for selecting and generating data exfiltration traffic and steps based on the target's and network's standard behavior and current defence mechanisms.

Machine learning can be useful for the attack itself as well. When the attackers have breached their target, they may have one or more of the following goals: espionage, fraud, and sabotage. The attacks are performed with malicious programs such as malware, spyware, or ransomware. The most common way to infect a host nowadays is using phishing tactics to con legitimate users to download the malicious program [1]. Uploading the malicious program with the use of software vulnerabilities is also a prolific way to accomplish an attack. Denial-of-service attacks are also quite common mischievous attacks that could cause serious harm. An example of some of the less known attacks is crowd-turfing, which means the generation of fake news, fake reviews, mass following, and other exploits for social media and crowd-sourcing efforts. Machine learning is especially well adapted for crowd-turfing attempts [66].

3 Adversarial Machine Learning

Unsurprisingly, the MLsec technology can be attacked, as with any technology developed at present. As the use of MLsec models increases, so does the desire to influence them. The models themselves are in many cases essentially black-boxes; therefore the attacker's influence on them can be left unseen by the defenders.

In the logistics industry, autonomous vehicles can be tricked into misinterpreting signs and other triggers for action. The method for spoofing image classification algorithms has been proven already [26], and will certainly cause issues in this field. For AI in the cybersecurity space, MLsec methods are abundant in email spam filters and malware detection systems. These systems are not impenetrable and can be bypassed [81]. Facial recognition is used in many places, such as smart devices and laptops, as a credential replacement (i.e., identification feature). As with the logistics example, face recognition uses similar technology and can be fooled in many, sometimes arbitrary, ways [74]. This spoofing logic can be applied to voice recognition as well. Research shows [13] that, for example, Amazon Echo devices can be tricked into executing commands by adding minimal, inaudible noises to a voice recording. These arbitrary commands can be used to hack the devices. In the finance industry, MLsec-driven fraud detection systems and user behavior analysis are in place; however, hackers can still manipulate and trick these systems as well [67].

MLsec-driven systems can be attacked in many ways, but the attacks can be boiled down to three distinct goals that the attackers have: espionage (Privacy), sabotage (Availability), and fraud (Integrity) [67].

3.1 MLsec Attack Goals

The attackers can weaponize a defensive MLsec model against itself, by creating well-crafted inputs to the system in order to gain information about it. There are examples of anonymous data in a dataset being identified, such as the Netflix dataset case [59]. This information can be used to further enhance later attacks or used directly as-is [67].

The defender's MLsec models can also be sabotaged. Some typical attacks, such as flooding, also work on MLsec models. However, some specialized attacks are also available. For example, the trustworthiness of a model can be deteriorated by submitting many misclassified data to it and even aiding in the retraining of the model with this intentionally bad data. Another attack is called adversarial reprogramming of a MLsec model, and it means using the model for one's own purposes to solve one's tasks instead of the model holder's intended use for the model [67]. Another example of sabotage is resource denial, which is effectively a DoS-attack. Hong et al. [39] proposed DeepSloth, a multi-exit architecture targeting adversarial slowdown attack. The authors showcased how their attack can essentially render popular multi-exit architecture models, such as MobileNet, unable to perform tasks properly. With DeepSloth, the attacker can create adversarial perturbations, which are indistinguishable to humans, but result in far more complex calculations than a regular sample, thus, creating slowdowns [39].

Fraud is intentional task misclassification in MLsec model terms. This means that the model interprets it's environment or inputs incorrectly and possibly does something inaccurate or harmful. This can be done using poisoning or evasion. Poisoning means interacting with the system during the training phase and poisoning the dataset in order to make the model behave unintentionally [46]. Evasion is a term used for model vulnerability exploitation in order to achieve misoperation [67].

3.2 Adversarial Attacks

The scope of attacks against MLsec is extensive. MLsec models require specific hardware and environments with ML libraries and frameworks and they all have their own vulnerabilities. Nevertheless, the most important and underestimated part of AI security is the algorithm-specific vulnerabilities unique to ML.

There are three main approaches an attacker can take to affect a MLsec model:

- **Manipulation.** The attacker corrupts the logic of the MLsec model with a specially generated input. This approach includes such attacks as adversarial examples, adversarial reprogramming, scaling, and others.
- **Extraction.** The attacker extracts data from the MLsec model. The data can be data about the model or the dataset itself. Attacks in this section are model inversion, membership inference, model extraction, and others.

- **Injection.** The attacker injects data into a training process aiming to corrupt the model. Attacks such as poisoning, backdoors, and Trojans are part of this approach.

Similarly to OWASP Top-10 for Application Security [87], below we summarize an unofficial Top 10 list of adversarial attacks on AI solutions.

1. **Evasion attacks** involve inputs for MLsec that result in an incorrect output and cause models to make false predictions or behave in an unexpected manner [82]. As an example from the real world, we can take an AI-driver malware detection engine, which can be bypassed using an adversarial attack by changing binary in such a way that it will not be recognized as a malware.
2. **Poisoning attacks** involve injecting data into the training dataset to make MLsec models produce an incorrect output [70]. Such an attack can be performed, for example, against a spam filter by poisoning the ML classifier with non-spam data, which looks like spam to retrain the classifier.
3. **Model inversion** relates to retrieving information about the samples used in the dataset [25]. This type of attack can be used against NLP algorithms, such as GPT-3 [24], to retrieve critical information on how the system was trained. These data can contain credit card information, passwords, or any other private information.
4. **Backdoor attacks** affect training models in such a way that they behave anomalously when fed a particular input but work as intended with other inputs [32]. For example, AI applications, especially devices, may be developed by multiple vendors. The outsourced AI algorithm developer can insert a backdoor into their AI model and sell it to an IoT vendor as a smart speaker model. These types of backdoors can, for example, have some undocumented voice commands for accessing the device.
5. **Model extraction** implies gaining knowledge about the ML model and its parameters with the help of model extraction attacks [61]. For example, if someone wants to steal an API-provider's model, the attackers could use legitimate API requests in a specially-crafted manner so that it leads to model extraction after a finite number of API calls.
6. **Membership inference** is a privacy attack that is intended to determine whether certain data points were in the training dataset or not [75]. The healthcare industry could be affected by these types of attacks as their datasets contain private medical data.
7. **Trojan attacks** are direct modifications to a trained model. The intention is to enable the model to perform additional tasks [53]. In comparison to the *backdoor attack* above, the attacker does not have access to the dataset; however, the attacker has access to the MLsec model stored or transmitted insecurely. Many AI applications were trained with the help of transfer learning, therefore, most of the time the model base is publicly available. Attackers may hack the model hosting and inject trojans into the base model and affect models derived from that as well.
8. **Presentation attacks** are methods where the attacker generates and constructs new objects to fool the MLsec model [21]. Face recognition applications are the most common target for such an attack.

9. **Attribute inference** is an attack that is performed to get information about dataset attributes [27]. With this attack, the hackers may find information about the dataset attributes and guess what parameters are favored by the model. Recommendation systems may be a good example of a target for attribute inference attack.
10. **Slowdown attack** implies performing various types of DoS attacks [39], as some ML-related operations are resource intensive (e.g., re-training a model, transfer learning, inference on complex models). Any publicly available AI-driven API can be vulnerable to this type of attack.

3.3 Attacks on various MLsec methods

Adversarial samples are slightly altered versions of originals that cause MLsec models to misclassify the sample [28]. Adversarial samples are few and far between when it comes to ML models. Every classification algorithm [18] and regression algorithm [84] that the MLsec models are based on is vulnerable to some sort of adversarial input. Generative models such as auto-encoders and GANs are also prone to attacks [48].

Fast Gradient Sign Method (FGSM) [28] is one heavily utilized method for creating adversarial samples. In essence, FGSM calculates the gradients of a loss function in regards to the input sample and utilizes the sign of these gradients in order to generate a new adversarial input [28].

Clustering differs from classification, as the classes of the data are not known in advance. Clustering is heavily used in cybersecurity for malware detection, and commonly, new training data for it comes from the Internet. If this is somehow known, the attacker can manipulate the training data, effectively worsening the model from the start. Other attacks against clustering are available. For example, an attack against a common algorithm, k-nearest neighbor, was presented [77].

With complex systems and multi-featured data, dimensionality reduction is often required. Although this category is not nearly as popular as some of the others, there have been attacks performed for dimensionality reduction [68, 69]. For example, Rubistein et al. [69] showcased their attack on Principal Component Analysis (PCA), where they poisoned training data in order to decrease DoS attack detection rate significantly.

Reinforcement learning can also be tricked with adversarial attacks even though there is no training data gathered beforehand. Behzadan and Munir [9] demonstrated that the action-based and reward-based training of RL algorithms can be fooled, and thereby the policies can be affected by injecting adversarial inputs to the environment.

As MLsec-based Intrusion Detection Systems (IDSs) are one of the most implemented systems for MLsec in the cybersecurity space, it makes sense that those systems are being tested with new types of attacks. For example, Lin et al. [52] developed IDSGAN, a GAN-model-based adversarial malicious traffic generating system, to attack IDSs. The authors managed to reduce the detection rates of black-box IDS models to almost zero for several different attacks. Similarly, Shu et al. [76]

demonstrated their Generative Adversarial Active Learning (Gen-AAL) method for adversarial attacks against IDSs without previous knowledge about the target system. They also require minimal number queries to the target to gain adequate numbers of training data for their GAN model.

Martins et al. [57] conducted a review on adversarial ML applied to malware and system intrusion cases. They concluded that adversarial attacks can deteriorate the efficacy of all normal classifiers. Nonetheless, the most effective defence against such attacks is adversarial training (i.e., including adversarial samples in training).

Catching encrypted C2 traffic is essential when dealing with attackers who have previously established a beachhead into a system. However, detection avoidance of C2 traffic can be improved by applying MLsec techniques, for example, Novo and Morla [60] generated C2 traffic adversarial samples by employing FGSM and adding TCP/IP flow delays (padding). They set the samples against their own detector and reached highly successful avoidance when the detector was not trained with a dataset containing adversarial samples.

4 Conclusion

In this chapter, we addressed the innovations in ML, DL, RL, and AI within the offensive cybersecurity fields. We structured this chapter inline with the Lockheed Martin's Cyber Kill Chain taxonomy in order to cover adequate grounds on this broad, growing, and hot topic.

At the time of this writing, the offensive MLsec is quite well represented and mature. On the one hand, Reconnaissance, Delivery, and Installation phases of the Cyber Kill Chain are most advanced in terms of technology maturity, number of state of the art works, and the coverage of possible scenarios. However, Exploitation and Acting on objectives phases are in incipient stages, and require more intensive research and innovation efforts. In addition to the use of MLsec for cybersecurity offensive scenarios, we have also explored the offensive technologies against MLsec technologies themselves. As the presented research has shown, the MLsec systems have weaknesses that can (and most certainly will) be exploited by attackers, which sets an increasingly worrying trend for future attacks as well as the defensive capabilities based on MLsec.

In this sense, we also conclude that besides the defenders, the attackers most certainly can and will (ab)use the MLsec technologies for nefarious purposes, and there is evidence that the attackers are often one step ahead. Very likely, it will be impossible to completely embargo or restrict the MLsec use to ethical use only (similarly to kitchen knives being easily available in many stores). However, the research and practitioner communities can minimize the abuse of MLsec for offensive purposes by developing MLsec technologies and models in a way that is easy out of the box to inspect, verify, and explain such as "eXplainable AI" (XAI).

Acknowledgements The authors would like to thank the technical team of Adversa.AI for valuable feedback and insights throughout the draft stages of this chapter. Hannu Turtiainen would like to thank the Finnish Cultural Foundation / Suomen Kulttuurirahasto (https://skr.fi/en) for supporting his Ph.D. dissertation work and research (grant decision 00211119), and the Faculty of Information Technology of University of Jyvaskyla (JYU), in particular Prof. Timo Hämäläinen, for partly supporting his PhD supervision at JYU in 2021–2022.

The authors acknowledge icon authors Gregor Cresnar, Freepik, and Good Ware (courtesy of `https://flaticon.com` [23]) for their royalty-free icons that are used in Figure 1.

References

1. Abbate, P.: Internet Crime Report 2020. Tech. rep., Federal Bureau of Investigation (2021). URL `https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf`
2. Al-Hababi, A., Tokgoz, S.C.: Man-in-the-middle attacks to detect and identify services in encrypted network flows using machine learning. In: 3rd International Conference on Advanced Communication Technologies and Networking (CommNet). IEEE (2020)
3. Alhuzali, A., Gjomemo, R., Eshete, B., Venkatakrishnan, V.: {NAVEX}: Precise and scalable exploit generation for dynamic web applications. In: 27th {USENIX} Security Symposium (2018)
4. Alqahtani, F.H., Alsulaiman, F.A.: Is image-based CAPTCHA secure against attacks based on machine learning? An experimental study. Computers & Security **88** (2020)
5. Anderson, H.S., Woodbridge, J., Filar, B.: DeepDGA: Adversarially-tuned domain generation and detection. In: ACM Workshop on Artificial Intelligence and Security. ACM (2016)
6. Antonakakis et al., M.: Understanding the Mirai Botnet. In: 26th USENIX Security Symposium. USENIX Association (2017)
7. Avgerinos, T., Cha, S.K., Rebert, A., Schwartz, E.J., Woo, M., Brumley, D.: Automatic exploit generation. Communications of the ACM **57** (2014)
8. Bahnsen, A.C., Torroledo, I., Camacho, L.D., Villegas, S.: DeepPhish: simulating malicious AI. In: APWG symposium on electronic crime research (eCrime) (2018)
9. Behzadan, V., Munir, A.: Vulnerability of deep reinforcement learning to policy induction attacks (2017). URL `https://arxiv.org/abs/1701.04143`
10. Brewster, T.: Fraudsters Cloned Company Director's Voice In $35 Million Bank Heist, Police Find (2021). URL `https://www.forbes.com/sites/thomasbrewster/2021/10/14/huge-bank-fraud-uses-deep-fake-voice-tech-to-steal-millions/?sh=1456cb187559`
11. Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., et al.: The malicious use of artificial intelligence: Forecasting, prevention, and mitigation (2018). URL `https://arxiv.org/abs/1802.07228`
12. Chomiak-Orsa, I., Rot, A., Blaicke, B.: Artificial Intelligence in Cybersecurity: The Use of AI Along the Cyber Kill Chain. In: International Conference on Computational Collective Intelligence. Springer (2019)
13. Chung, H., Iorga, M., Voas, J., Lee, S.: Alexa, can I trust you? IEEE Computer Magazine **50** (2017)
14. Chung, K., Kalbarczyk, Z.T., Iyer, R.K.: Availability attacks on computing systems through alteration of environmental control: smart malware approach. In: 10th ACM/IEEE International Conference on Cyber-Physical Systems. ACM (2019)
15. Conti, M., De Gaspari, F., Mancini, L.V.: Know your enemy: Stealth configuration-information gathering in SDN. In: International Conference on Green, Pervasive, and Cloud Computing. Springer (2017)
16. Costin, A.: Security of CCTV and video surveillance systems: Threats, vulnerabilities, attacks, and mitigations. In: 6th International Workshop on Trustworthy Embedded Devices (TrustED) (2016)

17. Cruz-Perez, C., Starostenko, O., Uceda-Ponga, F., Alarcon-Aquino, V., Reyes-Cabrera, L.: Breaking reCAPTCHAs with unpredictable collapse: Heuristic character segmentation and recognition. In: Mexican Conference on Pattern Recognition. Springer (2012)
18. Dalvi, N., Domingos, P., Sanghai, S., Verma, D.: Adversarial classification. In: 10th ACM SIGKDD international conference on Knowledge discovery and data mining (2004)
19. Darktrace: Darktrace Cyber AI Analyst: Autonomous Investigations (White Paper)
20. Descript: Lyrebird AI. URL https://www.descript.com/lyrebird
21. Fang, M., Damer, N., Kirchbuchner, F., Kuijper, A.: Real Masks and Fake Faces: On the Masked Face Presentation Attack Detection (2021). URL https://arxiv.org/abs/2103.01546
22. Fang, Y., Liu, Y., Huang, C., Liu, L.: FastEmbed: Predicting vulnerability exploitation possibility based on ensemble machine learning algorithm. PLOS ONE (2020)
23. Flaticon.com: (2021). URL https://www.flaticon.com/
24. Floridi, L., Chiriatti, M.: GPT-3: Its nature, scope, limits, and consequences. Minds and Machines **30** (2020)
25. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security (2015)
26. Gitlin, J.M.: Hacking street signs with stickers could confuse self-driving cars (2017). URL https://arstechnica.com/cars/2017/09/hacking-street-signs-with-stickers-could-confuse-self-driving-cars/
27. Gong, N.Z., Liu, B.: You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In: 25th {USENIX} Security Symposium (2016)
28. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2014). URL https://arxiv.org/abs/1412.6572
29. Google: What is reCAPTCHA? URL https://www.google.com/recaptcha/about/
30. Gossweiler, R., Kamvar, M., Baluja, S.: What's up captcha? a captcha based on image orientation. In: 18th international conference on World wide web (2009)
31. Grieco, G., Grinblat, G.L., Uzal, L., Rawat, S., Feist, J., Mounier, L.: Toward large-scale vulnerability discovery using machine learning. In: 6th ACM Conference on Data and Application Security and Privacy (2016)
32. Gu, T., Dolan-Gavitt, B., Garg, S.: BadNets: Identifying vulnerabilities in the machine learning model supply chain (2017). URL https://arxiv.org/abs/1708.06733
33. Guri, M., Bykhovsky, D.: air-jumper: Covert air-gap exfiltration/infiltration via security cameras & infrared (IR). Computers & Security **82** (2019)
34. Guri, M., Kachlon, A., Hasson, O., Kedma, G., Mirsky, Y., Elovici, Y.: Gsmem: Data exfiltration from air-gapped computers over {GSM} frequencies. In: 24th {USENIX} Security Symposium (2015)
35. Guri, M., Kedma, G., Kachlon, A., Elovici, Y.: AirHopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies. In: 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE). IEEE (2014)
36. Guri, M., Zadov, B., Elovici, Y.: LED-it-GO: Leaking (a lot of) Data from Air-Gapped Computers via the (small) Hard Drive LED. In: International conference on detection of intrusions and malware, and vulnerability assessment. Springer (2017)
37. HackTheBox: A Massive Hacking Playground. URL https://www.hackthebox.eu/
38. Hitaj, B., Gasti, P., Ateniese, G., Perez-Cruz, F.: Passgan: A deep learning approach for password guessing. In: International Conference on Applied Cryptography and Network Security. Springer (2019)
39. Hong, S., Kaya, Y., Modoranu, I.V., Dumitraş, T.: A Panda? No, It's a Sloth: Slowdown Attacks on Adaptive Multi-Exit Neural Network Inference (2020). URL https://arxiv.org/abs/2010.02432
40. Hu, W., Tan, Y.: Generating adversarial malware examples for black-box attacks based on GAN (2017). URL https://arxiv.org/abs/1702.05983
41. Huang, S.K., Huang, M.H., Huang, P.Y., Lai, C.W., Lu, H.L., Leong, W.M.: Crax: Software crash analysis for automatic exploit generation by modeling attacks as symbolic continuations. In: IEEE 6th International Conference on Software Security and Reliability. IEEE (2012)

42. Huang, S.K., Huang, M.H., Huang, P.Y., Lu, H.L., Lai, C.W.: Software crash analysis for automatic exploit generation on binary programs. IEEE Transactions on Reliability **63** (2014)
43. Hutchins, E.M., Cloppert, M.J., Amin, R.M., et al.: Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues in Information Warfare & Security Research **1** (2011)
44. Kaloudi, N., Li, J.: The AI-based cyber threat landscape: A survey. ACM Computing Surveys (CSUR) **53** (2020)
45. Khan, S.A., Khan, W., Hussain, A.: Phishing attacks and websites classification using machine learning and multiple datasets (a comparative analysis). In: International Conference on Intelligent Computing. Springer (2020). URL https://arxiv.org/abs/2101.02552
46. Khurana, N., Mittal, S., Piplai, A., Joshi, A.: Preventing poisoning attacks on AI based threat intelligence systems. In: IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE (2019)
47. Knabl, G.: Machine Learning-driven Password Lists. Ph.D. thesis (2018)
48. Kos, J., Fischer, I., Song, D.: Adversarial examples for generative models (2017). URL https://arxiv.org/abs/1702.06832
49. Kotey, S.D., Tchao, E.T., Gadze, J.D.: On distributed denial of service current defense schemes. Technologies **7** (2019)
50. Lee, K., Yim, K.: Cybersecurity threats based on machine learning-based offensive technique for password authentication. Applied Sciences **10** (2020)
51. Li, J.h.: Cyber security meets artificial intelligence: a survey. Frontiers of Information Technology & Electronic Engineering **19** (2018)
52. Lin, Z., Shi, Y., Xue, Z.: Idsgan: Generative adversarial networks for attack generation against intrusion detection (2018). URL https://arxiv.org/abs/1809.02077
53. Liu, Y., Ma, S., Aafer, Y., Lee, W.C., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks (2017)
54. Lockheed Martin Corporation: Gaining the advantageapplying cyber kill chain® methodology to network defense. URL https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/Gaining_the_Advantage_Cyber_Kill_Chain.pdf
55. Maestre Vidal, J., Sotelo Monge, M.A.: Obfuscation of malicious behaviors for thwarting masquerade detection systems based on locality features. Sensors **20** (2020)
56. Manky, D.: Rise of the 'Hivenet': Botnets That Think for Themselves (2018). URL https://www.darkreading.com/vulnerabilities-threats/rise-of-the-hivenet-botnets-that-think-for-themselves
57. Martins, N., Cruz, J.M., Cruz, T., Abreu, P.H.: Adversarial machine learning applied to intrusion and malware scenarios: a systematic review. IEEE Access **8** (2020)
58. Mirsky, Y., Demontis, A., Kotak, J., Shankar, R., Gelei, D., Yang, L., Zhang, X., Lee, W., Elovici, Y., Biggio, B.: The Threat of Offensive AI to Organizations (2021). URL https://arxiv.org/abs/2106.15764
59. Narayanan, A., Shmatikov, V.: How to break anonymity of the netflix prize dataset (2006). URL https://arxiv.org/abs/cs/0610105
60. Novo, C., Morla, R.: Flow-based detection and proxy-based evasion of encrypted malware c2 traffic. In: 13th ACM Workshop on Artificial Intelligence and Security (2020)
61. Oh, S.J., Schiele, B., Fritz, M.: Towards reverse-engineering black-box neural networks. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer (2019)
62. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: WaveNet: A Generative Model for Raw Audio (2016). URL https://arxiv.org/abs/1609.03499
63. Osadchy, M., Hernandez-Castro, J., Gibson, S., Dunkelman, O., Pérez-Cabo, D.: No bot expects the DeepCAPTCHA! Introducing immutable adversarial examples, with applications to CAPTCHA generation. IEEE Transactions on Information Forensics and Security **12** (2017)
64. Pacheco, F., Exposito, E., Gineste, M., Baudoin, C., Aguilar, J.: Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. IEEE Communications Surveys & Tutorials **21** (2018)

65. PimEyes: PimEyes: Face Recognition Search Engine and Reverse Image Search. URL https://pimeyes.com/en
66. Polyakov, A.: Machine learning for cybercriminals 101 (2018). URL https://towardsdatascience.com/machine-learning-for-cybercriminals-a46798a8c268
67. Polyakov, A.: Ai security and adversarial machine learning 101 (2019). URL https://towardsdatascience.com/ai-and-ml-security-101-6af8026675ff
68. Ringberg, H., Soule, A., Rexford, J., Diot, C.: Sensitivity of PCA for traffic anomaly detection. In: ACM SIGMETRICS international conference on Measurement and modeling of computer systems (2007)
69. Rubinstein, B.I., Nelson, B., Huang, L., Joseph, A.D., Lau, S.h., Rao, S., Taft, N., Tygar, J.: Stealthy poisoning attacks on PCA-based anomaly detectors. ACM SIGMETRICS Performance Evaluation Review **37** (2009)
70. Schwarzschild, A., Goldblum, M., Gupta, A., Dickerson, J.P., Goldstein, T.: Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In: International Conference on Machine Learning. PMLR (2021)
71. Seymour, J., Tully, P.: Weaponizing data science for social engineering: Automated e2e spear phishing on twitter. BlackHat USA **37** (2016)
72. Seymour, J., Tully, P.: Generative models for spear phishing posts on social media (2018). URL https://arxiv.org/abs/1802.05196
73. Shafiq, M., Yu, X., Laghari, A.A., Yao, L., Karn, N.K., Abdessamia, F.: Network traffic classification techniques and comparative analysis using machine learning algorithms. In: 2nd IEEE International Conference on Computer and Communications (ICCC). IEEE (2016)
74. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: ACM SIGSAC conference on computer and communications security (2016)
75. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP). IEEE (2017)
76. Shu, D., Leslie, N.O., Kamhoua, C.A., Tucker, C.S.: Generative adversarial attacks against intrusion detection systems using active learning. In: 2nd ACM Workshop on Wireless Security and Machine Learning (2020)
77. Sitawarin, C., Wagner, D.: On the robustness of deep k-nearest neighbors (2019). URL http://arxiv.org/abs/1903.08333v1
78. Sivakorn, S., Polakis, J., Keromytis, A.D.: I'm not a human: Breaking the Google reCAPTCHA. BlackHat (2016)
79. Stoecklin, M.P.: Deeplocker: How AI can power a stealthy new breed of malware. Security Intelligence **8** (2018)
80. Stone, G., Talbert, D., Eberle, W.: Using ai/machine learning for reconnaissance activities during network penetration testing. In: International Conference on Cyber Warfare and Security. Academic Conferences International Limited (2021)
81. Subramaniam, T., Jalab, H.A., Taqa, A.Y.: Overview of textual anti-spam filtering techniques. International Journal of Physical Sciences **5** (2010)
82. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks (2013). URL https://arxiv.org/abs/1312.6199
83. The MITRE Corporation: MITRE ATT&CK Matrix for Enterprise. URL https://attack.mitre.org/
84. Tong, L., Yu, S., Alfeld, S., Vorobeychik, Y.: Adversarial regression with multiple learners (2018). URL https://arxiv.org/abs/1806.02256
85. Valea, O., Oprişa, C.: Towards pentesting automation using the metasploit framework. In: IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE (2020)
86. Wang, M., Su, P., Li, Q., Ying, L., Yang, Y., Feng, D.: Automatic polymorphic exploit generation for software vulnerabilities. In: International Conference on Security and Privacy in Communication Systems. Springer (2013)

87. Wichers, D., Williams, J.: OWASP TOP-10 2017. OWASP Foundation (2017)
88. Yadav, T., Rao, A.M.: Technical aspects of cyber kill chain. In: International Symposium on Security in Computing and Communication. Springer (2015)
89. Yamin, M.M., Ullah, M., Ullah, H., Katt, B.: Weaponized AI for cyber attacks. Journal of Information Security and Applications **57** (2021)
90. Yim, K.: A new noise mingling approach to protect the authentication password. In: International Conference on Complex, Intelligent and Software Intensive Systems (2010)
91. Yu, N., Darling, K.: A low-cost approach to crack python CAPTCHAs using AI-based chosen-plaintext attack. Applied Sciences **9** (2019)
92. Zargar, S.T., Joshi, J., Tipper, D.: A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. IEEE communications surveys & tutorials **15** (2013)
93. Zhang, R., Chen, X., Lu, J., Wen, S., Nepal, S., Xiang, Y.: Using AI to hack IA: A new stealthy spyware against voice assistance functions in smart phones (2018). URL `https://arxiv.org/abs/1805.06187`