A Quantitative Analysis of Offensive Cyber Operation (OCO) Automation Tools

Samuel Zurowski*
University of New Haven,
Connecticut Institute of Technology
West Haven, CT, USA
szuro1@unh.newhaven.edu

George Lord*
University of New Haven,
Connecticut Institute of Technology
West Haven, CT, USA
georgelord@uchicago.edu

Ibrahim Baggili
University of New Haven,
Connecticut Institute of Technology
West Haven, CT, USA
ibaggili@newhaven.edu

ABSTRACT

The ecosystem for automated offensive security tools has grown in recent years. As more tools automate offensive security techniques via Artificial Intelligence (AI) and Machine Learning (ML), it may result in vulnerabilities due to adversarial attacks. Therefore, it is imperative that research is conducted to help understand the techniques used by these security tools. Our work explores the current state of the art in offensive security tools. First, we employ an abstract model that can be used to understand what phases of an Offensive Cyber Operation (OCO) can be automated. We then adopt a generalizable taxonomy, and apply it to automation tools (such as normal automation and the use of artificial intelligence in automation). We then curated a dataset of tools and research papers and quantitatively analyzed it. Our work resulted in a public dataset that includes analysis of (n=57) papers and OCO tools that are mapped to the the MITRE ATT&CK Framework enterprise techniques, applicable phases of our OCO model, and the details of the automation technique. The results show a need for a granular expansion on the ATT&CK Exploit Public-Facing application technique. A critical finding is that most OCO tools employed *Simple Rule Based* automation, hinting at a lucrative research opportunity for the use of Artificial Intelligence (AI) and Machine Learning (ML) in future OCO tooling.

KEYWORDS

Offensive Cyber Operations, Offensive Security, Tactics Techniques and Procedures, MITRE ATT&CK, Automation, Artificial Intelligence, Cybersecurity

ACM Reference Format:
Samuel Zurowski, George Lord*, and Ibrahim Baggili. 2022. A Quantitative Analysis of Offensive Cyber Operation (OCO) Automation Tools. In *The 17th International Conference on Availability, Reliability and Security (ARES 2022), August 23–26, 2022, Vienna, Austria.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3538969.3544414

*These authors contributed equally to this work

ARES 2022, August 23–26, 2022, Vienna, Austria

ACM ISBN 978-1-4503-9670-7/22/08…$15.00
https://doi.org/10.1145/3538969.3544414

1 INTRODUCTION AND MOTIVATION

Offensive Cyber Operations (OCO) can be defined as, "actions affecting the confidentiality, integrity, and availability of information "[23]. More specifically, OCOs are performed as organized attacks against specific entities' digital assets. Different actors may perform OCOs such as cyber operatives, threat actors, penetration testers, and attackers. An example of this is government-led operations performed in cyberspace with the intention of gaining access to an adversary's critical systems. A well known instance of a possible government OCO is the 2020 SolarWinds supply chain attack which resulted in tens of thousands of government computers being compromised. This attack was one of the worst government breaches in United States (U.S.) history [18].

The U.S. Department of Defense officially recognizes OCOs like Stuxnet as, "missions intended to project power in and through cyberspace"[1]. While government-led OCOs as such have displayed their importance in maintaining international relations and global security, the majority of what could be considered OCOs are performed by non-state actors at lower stakes. OCOs performed against non-consenting entities such as companies or individuals are considered to be a form of cybercrime, and are primarily conducted with the intention of exploiting the entity's systems for financial gain through the sale of stolen information or other criminal avenues [2]. To anticipate OCOs, institutions hire cybersecurity professionals to conduct penetration tests which uncover potential vulnerabilities that adversaries may exploit.

Artificial Intelligence (AI) and Machine Learning (ML) algorithms have been used to automate aspects of OCOs. Tools such as DeepExploit have automated penetration testing using the Asynchronous Advantage Actor Critic (A3C) algorithm. DeepExploit uses reinforcement learning to quickly learn various ways to pinpoint vulnerabilities and then exploit them [42]. Post-exploitation has been automated utilizing PowerShell Empire and trained using the Advantage Actor Critic (A2C) reinforcement learning algorithm [28]. Another example includes an ML-based automation tool containing an attack model that can automate spearphishing campaigns against social networks [38]. A more sophisticated example is an evasive AI-powered malware that emulates behavior of legitimate applications on a system [32, 35]. Because of the usage of automation in offensive security tools via AI and ML, tools may be vulnerable to possible adversarial attacks [24]. In the future, it may be possible that these tools' models could be poisoned to miss vulnerabilities or work in unintended ways. Therefore, critical analysis of how automation is implemented will aid in curbing future adversarial attacks against offensive security tools [33].

Despite the growing security concerns of OCOs by threat actors, in addition to the rise of sophisticated tools to automate such threats (or attacks, etc): academia has failed to systematize the research in this direction. We contend that there needs to be agreement on (1) An understanding of what an OCO is and (2) the phases of an OCO and their associated terminology and (3) An analysis of the state-of-the-art related to OCO tools and techniques. Thus, our work makes the following contributions:

(1) A generalized and widely applicable model of an OCO and its sequence of steps, including identification of the OCO's purpose/target, intelligence gathering, execution, and assessment, along with the phases potentially found in each step.
(2) A publicly available categorized dataset (n=57) of offensive security automation literature and tools.
(3) A quantitative analysis of our organized dataset.

The paper is organized as follows: First, the required prerequisites are discussed in Section 2 followed by our limitations in Section 3. We then present the OCO model in Section 4 and share our quantitative analysis of the curated dataset in Section 7. We finally future work in Section 10 and conclude our work in Section 12.

2 PREREQUISITES

There were several prerequisites required to construct a comprehensive dataset composed of offensive security tools:

- **A standardized model for identifying the phases of an OCO**: For this, we chose the MITRE ATT&CK matrix [9], because it was the most recent, extensive, and well-structured model that provided a wide array of identifiable OCO processes (or techniques, as the model calls them).
- **A generalizable OCO model**: For this, we modified an existing OCO model, which we present in Section 4. The model is used to identify the specific phases conducted during an OCO. Each tool can be categorized by multiple phases included in the model.
- **An automation taxonomy for offensive security tools**: To identify tool automation types, an expansive taxonomy differentiating between general, AI, and "Hybrid" automation must be established. We used an existing taxonomy [22], which is presented in Section 5.

3 LIMITATIONS

Throughout the creation of the dataset, our results focused on the enterprise matrix in ATT&CK. This resulted in limitations of the scope for the types of offensive security tools and papers added to the dataset because ATT&CK has additional matrices for mobile and industrial control systems. Additionally, during the creation of the dataset, ATT&CK had an update that merged both PRE-ATT&CK and ATT&CK into one, which resulted in new techniques. The techniques added were reconnaissance and resource development, which could have been used in our analysis [34]. Lastly, only techniques were identified, not subtechniques.

Additionally, we mapped papers and various tools solely using publicly available information. This could result in potential discrepancies because tools or papers may not list every technique, process, and automation used. Our dataset is only a small subset of the offensive security tools that are publicly available. Because the ecosystem of available tools changes constantly, it is challenging to determine the entire scope of automated offensive security tools.

4 THE OFFENSIVE CYBER OPERATION PROCESS

In this section, we present, model, and analyze the OCO process.

4.1 Background Information on the OCO Process

The purpose of the OCO model is to allow entities to understand the approach of an OCO before, during, and after it is conducted. Both the MITRE ATT&CK Framework and PRE-ATT&CK focus mostly on the technical details of how an adversary approaches an enterprise target [12]. By no means is this work attempting to replace the MITRE ATT&CK Framework, rather it aims to understand the current state of automation in OCOs. Our OCO process model is an abstract combination of both ATT&CK and PRE-ATT&CK. PRE-ATT&CK focuses on the preceding phases before an attack is conducted, whereas ATT&CK is the operation which utilizes data from PRE-ATT&CK.

Additionally, our OCO model focuses more on the higher organizational level of planning and not only Advanced Persistence Threats (APTs). This grants a better interpretation of what any entity can do to automate in some way both organizational and technical aspects of an OCO. By combining our OCO process model with the Tactics, Techniques, and Procedures (TTPs) of ATT&CK, we are able to uncover areas areas that lack automation [7].

4.2 Modeling the OCO Process

Our OCO process model was based on [20]. The model is broken into five phases that describe the steps in conducting an OCO from both the organizational and technical level (see Figure 1). The OCO process model is broken into the following steps:

(1) **Purpose**: *Who is being targeted and why*? Because of this, our OCO model has the sole purpose of targeting an entity. The purpose can be broken down into specific reasons the entity is being targeted.
(2) **Target Identification**: *The intelligence required to attack a target.* It is important to note that the phases are iterative. Therefore, if new targets are found, the purpose can change.
(3) **Plan Attack**: *How will we attack*? This phase focuses on preparing and testing the tools and operators. This phase focuses on both rehearsing and planning the execution.
(4) **Execution**: *Was the attack successful*? This is when the operation is conducted. The execution should not include rehearsing because this is when the OCO is being conducted on the actual target.
(5) **Assessment**: *What have we learned*? In this phase of the OCO, an actor determines the results and assesses the success or failure of the attack.

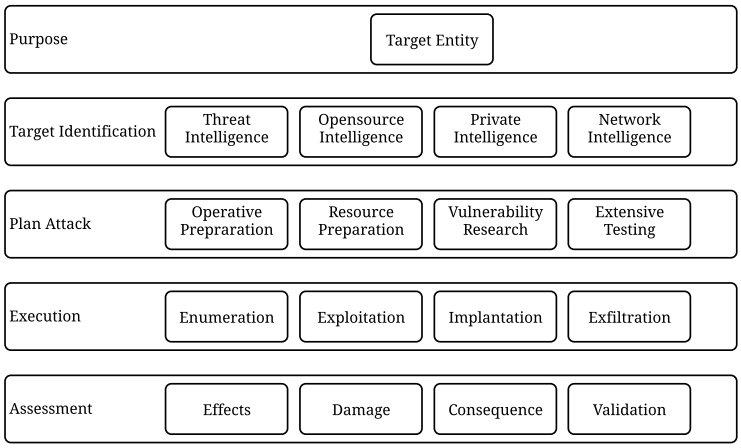

Figure 1: Diagram of the OCO Process Model

4.3 OCO Process Analysis

4.3.1 Purpose. The OCO process is broken down into five phases: (1) Purpose, (2) Target Identification, (3) Plan Attack, (4) Execution and (5) Assessment. Each of these phases are outlined in the following sections.

The first phase, *Purpose*, defines that an organization or individual always has the goal to target some sort of entity. This includes individuals, governments, corporations, and more. Additionally, the purpose entails why the OCO is being conducted, including reasons such as political motivation, penetration test, conjunction with military operations, retaliation, and more. Overall, the goal of purpose gives an overarching understanding of why the OCO is being conducted.

4.3.2 Target Identification. Target Identification is intelligence required to be able to plan the attack against an entity. Without any intelligence available, an OCO cannot be conducted. In some scenarios, attacks cannot be planned until critical information about the cyber-related infrastructure of the target. This can vary; in simpler cases, a spearphishing attack may provide a sufficient level of intelligence.

- *Threat Intelligence* can be a challenge for governments to collect, such as when dealing with an APT to be able to combat against them, which would be information known as Cyber Threat Intelligence (CTI). CTI is "data that is collected, processed, and analyzed to understand a threat actor's motives, targets, and attack behaviors" [3]. CTI can be received from external organizations or individuals. Moreover, there could be cases where the OCO comes in contact with the APT and the operators of an organization contend. However, the definition of Threat Intelligence in our model also includes physical cyber-related threats. An example could be knowlege of plans of an organization physically attacking an electric grid.
- *Opensource Intelligence* (OSINT) is defined as "publicly available information appearing in print or electronic form including radio, television, newspapers, journals, the Internet, commercial databases, and videos, graphics, and drawings" [10]. OSINT includes all the layers of the web that are accessible. Opensource tools, such as the theHarvester, are used to "help determine a company's external threat landscape on the internet. The tool gathers emails, names, subdomains, Internet Protocol (IP) addresses and Uniform Resource Locators (URLs) using multiple public data sources" [27]. By using this information, one may identify various potential targets by email, previously unknown websites, and more.
- *Private Intelligence* is information that only an individual or an organization knows. This could be something as simple as knowing the hours when the entity heavily monitors and updates systems. Another example of private intelligence would be a zero-day vulnerability only known by a specific organization.
- *Network Intelligence* includes IP addresses, domain information, ports, services running on machines (e.g. File Transfer Protocol), URLs, and any information related to networking. In some scenarios, network intelligence may be gathered from previous OCOs. In other instances, network scans could occur before the execution.

4.3.3 Plan Attack. Using the intelligence gathered during the *Target Identification* stage, the information discovered is used to prepare conducting the OCO. *Plan Attack* goes over the preparation of security tools and operators prior to execution. This includes understanding how to exploit specific vulnerabilities or discovering zero-day vulnerabilities.

- *Operative Preparation* is training an intelligent computer-based agent or a human operator to prepare for execution. Training the attackers can require an extensive amount of time. This could also include exercises and briefings.

- *Resource Preparation* includes weaponizing systems that could automate discovery of a specific device, exploit vulnerabilities, create backdoors, and/or exfiltrate data.
- *Vulnerability Research* is the process of discovering and identifying how to exploit specific vulnerabilities. For example, Eternalblue (MS17-010) is a vulnerability that exploited Server Message Block Version 1 (SMBv1) and was only known by the NSA [13]. The NSA had to spend extensive time reverse engineering and developing a tool capable of exploiting SMBv1 [14].
- *Extensive Testing* is the last phase in *Plan Attack*. Execution entails the testing of developed capabilities and training the attacker. Extensive testing focuses on running through scenarios of the operation and testing tools to ensure they work.

4.3.4 Execution. This phase is where the planned attack is executed by the actor. Execution follows the kill chain [5] and our model specifies its goals. ATT&CK specifies many of the techniques potentially performed during execution [6].

- *Enumeration* is the first part in the *Execution* phase. It serves to identify beneficial information during execution.
- *Exploitation* uses known vulnerabilities to further the operation. For example, an actor could use stolen credentials to gain unauthorized access to target systems. Another example could be performing a technique such as a denial of service attack.
- *Implantation* involves modifying the target's environment to allow the actor future access. For example, an actor could use create a Reverse Transfer Protocol (TCP) Payload to act as a backdoor in target systems [26]. Planting false evidence on a target system (deception) is another example of implantation.

4.3.5 Assessment. The assessment phase analyzes the cumulative results of all previous phases.

- *Effects* is broken into two categories: cognitive and cyber. Cognitive effects are defined as using deception or the results of an attack to trigger human response to achieve goals. Cyber effects are defined by both physical and cyber-based effects on an entity. This include disturbances, data leakage, exposure, and more [31].
- *Damages* includes the destruction of digital and/or physical property during execution. In December 2015, Kyivoblenergo, an electrical distribution company in Ukraine, reported outages. Later, it was discovered that attackers targeted SCADA distribution management systems, which left approximately 225,000 customers without power [15].
- *Consequences* focuses on the operation's impact on the attacker. For example, when an OCO targets an APT, there can be some sort of retaliation. In 2019, Israel, "thwarted an attempted Hamas cyber offensive against Israeli targets. Following our successful cyber defensive operation, we targeted a building where the Hamas cyber operatives work" [16]. The consequence of Hamas' OCO included the destruction of the building in which their cyber capabilities were located. A breach of contract during a penetration test would also be considered a consequence.
- *Validation* includes verifying the effectiveness of the methods used in the operation.

5 OCO AUTOMATION AND TAXONOMY

To identify and study automation in OCO processes, we first need to define it. We adopt the definition for automation in the Springer Handbook of Automation, as the book already provides a widely applicable and clear definition [30]. In accordance with the text, we recognized automation as programs "developed by humans to perform a given set of activities without human involvement activities during those actitivities". In our case, "activities" are OCO processes.

New solutions are released constantly to automate parts of OCOs and organizations need to stay abreast over the state-of-the-art. If organizations are able to identify how these offensive security tools automate techniques, they can mitigate the potential threat they pose. To that end, a taxonomy is needed to allow researchers, attackers, and defenders to understand the methods of automation OCO tools implement.

The three types of automation used in OCO tools are "General Automation", "Hybrid Automation", and "Artificial Intelligence". The taxonomy was borrowed from Intrusion Detection Systems (IDS) research and then used to categorized each tool based on the type of automation employed [22].

There are three categories of automation in this taxonomy:

- **General Automation**: offensive security tools that follow predefined rules [17]. This includes repetitive tasks, such as manual configuration.
- **AI Automation**: a type of automation used to mimic human thinking. This allows offensive security tools to make its own decisions based on data.
- **Hybrid Automation**: a form of automation that has aspects of both general automation and AI. For example, an offensive security tool could use general automation to scan a network and then use AI to determine vulnerable systems.

6 METHODOLOGY

The purpose of our analysis was to explore the current state of offensive security tools. To analyze various offensive security tools that could be used in OCOs (Defined in Section 6.1), we reviewed research papers, opensource, and commercial tools. Details of these steps are discussed in Section 6.2.

6.1 Definition of Offensive Security Tools

Our dataset is a collection of tools that automate offensive security techniques. This includes but is not limited to SQL injection, active directory exploitation, privilege escalation, network scanning, intelligence gathering, and vulnerability scanning. In contrast, examples that are not considered to be an offensive security tool include: intrusion detection systems, firewalls, and antivirus software, as they play defensive roles.

6.2 Tool Analysis

The analyzed offensive security tools were collected from sources such as GitHub, commercial vendors, and academic works. For each

source that developed or theorized a security tool, we asked the following questions:

- **OCO Phase(s)**: What parts of an OCO does this automate?
- **Techniques used**: What specific ATT&CK techniques does it automate? If so, which techniques and how many?
- **Automation Implemented**: What type of automation is used? Is it using multiple forms of automation? How much AI is used versus general automation?
- **What is lacking**: In the domain of OCO, what other things are currently missing or lacking?

Our dataset contains the following information (when possible): name of publication or software, location (URL), operating systems used, country of origin, developer(s), organization types, publicly available (yes/no), release date, latest update, license(s), actively maintained (yes/no), programming language(s), and a citation.

6.3 Curation Approach

To construct a dataset that included a diverse number of tools that automate various OCOs, we followed a curation methodology to create the dataset. For example, some offensive security tools may have developers who maintain the tool who are from various countries. Therefore, when selecting the country we analyzed the creator of the tool and listed in the dataset the top contributors on the project. If it was a commercial organization or a group, the country is based on the location of the organization's headquarters. Also, many tools are able to run on multiple platforms, therefore we explored that. Of note is that the operating system listed in our dataset is the platforms the tools run on and not the operating system that the tool may exploit against.

Next, the organizational type in the dataset is either commercial, individual(s), or a group. Commercial is a company who is attempting to make profit off the offensive security tool or has other tools that can be purchased. Individual(s) can be considered a single developer or several developers who are creating a security tool. Lastly, the other organizational type is group. Group is different from a commercial in that the organization that is not for profit (e.g. OWASP). We also include the first year of release and last year release in the dataset. The first year release is either the first version (or commit) publicly available. If it is academic work it would be the release of the paper and if any other papers were published related to the work, those would be the year of latest release.

Some opensource tools have licenses putting restrictions on what is allowed in terms of modification or redistribution. Therefore, any licenses found from tools were included in the dataset. Because we are including a year of first release and year of last release, we define currently maintained as being updated at least once per year. Therefore, current tools in this dataset are "yes" if they were updated at least once in 2020.

Moreover, many tools are written in one or more programming languages. The reason we include programming languages is that it can allow researchers in this domain to understand if a specific language is used as a standard for developing tools or even how portable certain languages are for tools. Also, some tools may use several programming languages and thus, we include all of them for that specific tool being analyzed.

Additionally, all tools are mapped to the techniques (not subtechniques) of the MITRE ATT&CK Framework. When analyzing the tool we examine documentation, publications from that tool, and source code to determine which techniques fit the description of a tool. We include all the that the tool is able to automate. All tools are mapped to ATT&CK based upon publicly available knowledge. Therefore, some tools may not include more techniques because it is not known from opensource information and only known by an individual, group, or organization. Similarly, the applicable OCO phase(s) and automation type(s) are mapped the same way; we check if the opensource information matches the description of the categories, and if so it is mapped to that tool in the dataset.

The security tools we identified were collected through extensive web searching. This included and was not limited to general web searches, opensource repositories, and academic search engines. To ensure that the dataset of tools created includes all phases of an OCO, we identified tools for each area within the OCO model as shown in Figure 1.

Keywords used in searches included "network enumeration tools", "penetration testing automation", "vulnerability scanning automation", "exploitation system", "command and control", and similar terms. Additionally, databases and articles were scraped for common penetration testing tools that could be deemed relevant (e.g. best penetration testing tools). These keywords were intended to find a wide variety of commonly referenced tools that would be used in automating various techniques performed in OCOs. Tools that could be consider also include emulation such as Network Attack Simulator are included [37]. This is because these tools aid in planning a OCO.

The dataset can be found and downloaded from **BLINDED FOR REVIEW** to benefit members of the cybersecurity community. Certain parameters (such as year of last release) may become outdated as many of the tools are subject to change.

7 RESULTS & QUANTITATIVE ANALYSIS

The tools in our dataset automated a total of 161 ATT&CK techniques, and were mostly from the United States. This number is almost three times greater than the number if tools in the dataset because most tools are able to automate more than one technique. We analyzed the 25 most frequently automated techniques in our data set, shown in Figure 2. The most common technique, Exploit Public-Facing Application (T1190), is used two to four times more frequently than the 24 other most common techniques. The second most used technique is Command and Scripting Interpreter (T1059).

While the majority of the 25 techniques are automated by five to ten of the tools gathered, Exploit Public-Facing Application (T1190) is automated by 21 tools, or more than a third of those in the dataset. This may indicate that the technique is used more commonly in cyber operations, creating an apparent need for and thus supply of tools automating this process. This figure may also be disproportionately high as there is a large quantity and variety of public-facing applications that exist (including websites, databases, network devices, and other servers/services), leading to the development of more tools specializing in exploiting certain types of these applications. A larger analysis of tools is required to confirm this theory.

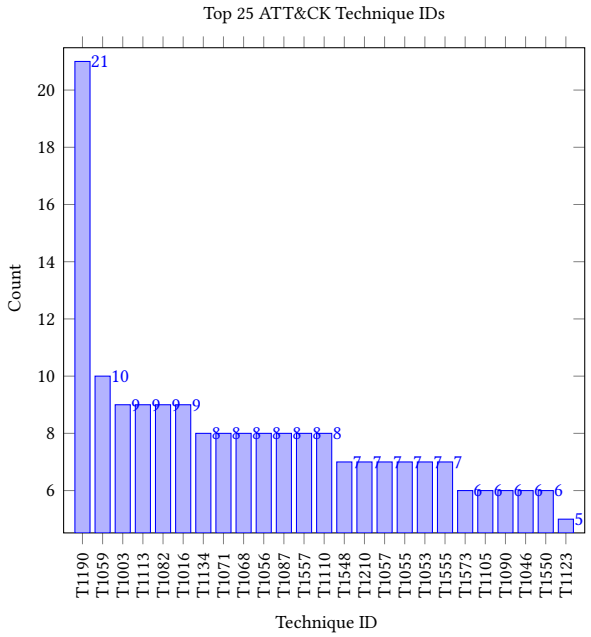

Figure 2: Top 25 ATT&CK Technique IDs

The following techniques show the second and third-highest prevalence of automation in the dataset (10 tools): Command and Scripting Interpreter (T1059), OS Credential Dumping (T1003), Screen Capture (T1113), System Information Discovery (T1082), and System Network Configuration Discovery (T1016). Two of these techniques provide remote access to adversarial machines, through remote execution of scripts/commands (T1059) or screen capture functionality such as those found in RATs (Remote Access Trojans) (T1113). This prevalence could be due to the importance of manipulating adversarial machines in OCOs (such as the remote execution of scripts to ruin centrifuges seen in Stuxnet), as well as the use of remote access tools used by criminals during attacks on individuals/organization's personal computers (such as during tech support scamming and ransomware attacks). The other three techniques are focused on providing intelligence to the attacker, particularly regarding system configuration information, indicating their possible use in the Target Identification and Enumeration stages of our OCO process model.

Table 1 shows the name of the correlating technique IDs. Table 2 displays all the OCO techniques that the tools automate. A notable finding is that no tools in our dataset automated: *Damages*, *Consequence*, or *Private Intelligence*. Automation plays a key role in offensive security tools and our results show that a majority of automation is *Simple Rule Based*. Many of the tools considered expert systems were able to automate multiple phases. As an example, there were Command and Control (C2) systems that perform multiple ATT&CK techniques and applied several OCO phases. Offensive security tools that use AI were more frequently created in the last couple of years. Other than *Simple Rule Based* and *Expert Systems* most forms of automation are underused. However, offensive security tools that automate a larger number of ATT&CK techniques frequently used AI.

Figure 3 displays the programming languages used by offensive security tools. Python is the most commonly used programming language amongst the collected offensive security tools. Python is used more than double the time than the second most used language, C. There is a wide variety of programming languages that were used to make at least one of the tools. It is also important to note that many tools used multiple programming languages.

Most of the tools in our dataset are publicly available (see Table 2), and approximately 80% of the tools are available for public download. 60% of the tools are actively maintained and were updated within the last year. This indicates that a large portion of tools are possibly outdated. Linux is the most frequently compatible operating system for the offensive security tools. The second most used operating system is Windows, followed by macOS. It is also important to note many tools were operable on all three of these operating systems.

8 DISCUSSION

The technique automated by the most tools was Exploit Public-Facing Application (T1190). T1190 is defined as "Adversaries may attempt to take advantage of a weakness in an Internet-facing computer or program using software, data, or commands in order to cause unintended or unanticipated behavior. The weakness in the system can be a bug, a glitch, or a design vulnerability" [8]. T1190 is mainly automated by tools that attempt to take advantage of vulnerabilities in network based protocol applications such as web applications. ATT&CK should create new techniques and subtechniques to highlight the specific service or exploitation techniques used on applications in T1190. This will allow researchers, attackers, and defenders to understand how they can exploit weaknesses in similar applications. Many organizations use ATT&CK for analytical purposes, suggesting that potentially useful data may not be captured [41]. We suggest that the ATT&CK techniques are broken down into more specific techniques such as Exploit Public-Facing Web Application, Exploit-Public Facing Virtual Private Network (VPN) application, and more. Exploit-Public Facing Web Application could also include subtechniques from the Open Web Application Security Project (OWASP) Top ten, such as Insecure Deserialization [4, 11].

In our dataset, most phases in the OCO model are automated by at least one offensive security tool as shown in Figure 4. However, only three tools automated *Target Entities*. This could be because actors usually have a specific target in mind. There is the possibility that future offensive security tools could be used to help organizations find employees or devices most vulnerable to an attack.

No tool was able to automate all of the phases of an OCO. This could be because it is difficult to automate all phases with one tool. None of the tools in our dataset automated the *Consequences* phase. This could be that the opensource and commercial community possibly do not need to conduct this phase. One area where the automation of *Consequences* may be useful is in recording the public unrest on sites such as social media, news articles, and more after an attack.

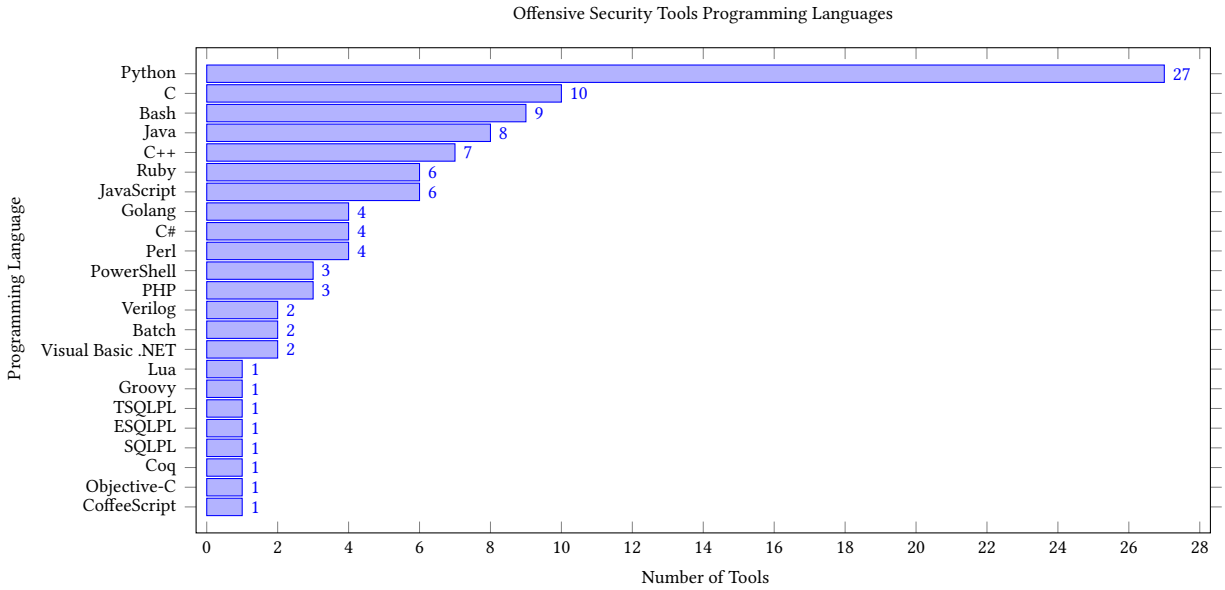

Figure 3: Offensive Security Tools' Programming Language

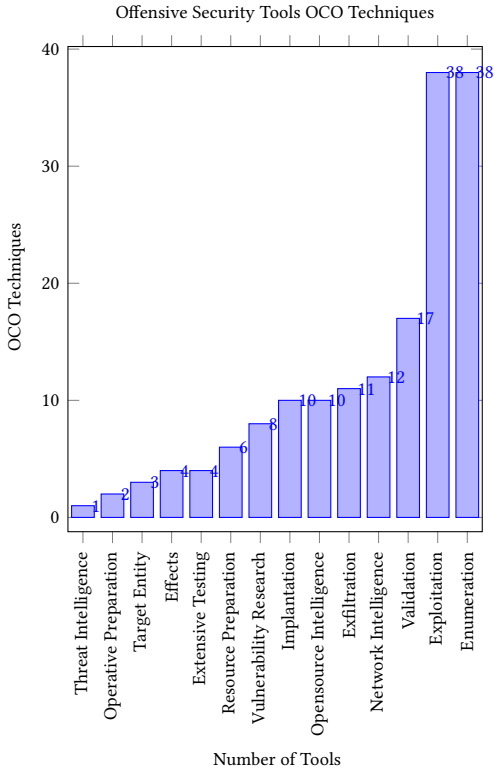

Figure 4: Offensive Security Tools OCO Techniques

Out of all the automation types, *Simple Rule Based* is used by the most tools, and in most cases it is able to automate multiple phases. Our data suggests that the best approach for future tools to automate more techniques would be to use AI. This is because an actor typically uses their own experience and understanding when conducting multiple techniques, which could be supplemented by AI. Also, the increased use of AI in newer tools suggests that AI may be used increasingly in upcoming years. This would imply that more offensive security tools may be susceptible to adversarial attacks in the future. Tools use various forms of automation, and as expected, General Automation is the most common of the three. It is notable that we had one tool in our dataset which was closed sourced therefore it resulted in not knowing the inner workings to be able to classify it.

9 KEY FINDINGS

The key findings from our analysis are as follows:

- Exploit Public-Facing Application (T1190) is the most used ATT&CK technique and may be broken down into more application-specific techniques. This would help gain a better understanding how applications are exploited from the outside.
- 161 MITRE ATT&CK Framework techniques are automated by the tools in our dataset.
- AI automation is found mostly in newer offensive security tools.
- Most tools do not use AI. This could be because recent advancements in AI are just starting to enable AI OCO tooling.
- Most tools use *Simple Rule Based* automation. This makes sense because most tools do not use AI.
- No tools automate the following OCO phases:
  - Private Intelligence
  - Damage
  - Consequences

- Most tools automate *Enumeration* and *Exploitation*. OCO tools may focus on gaining networking information to be able to exploit a system. Additionally, because AI is not used much, the purpose most tools may have is gaining an initial foothold.

10 FUTURE WORK

Future work may adopt newer versions of ATT&CK to identify more techniques automated by offensive security tools. Additionally, offensive security tools related to mobile and industrial control systems could be mapped to the correlating matrices. On the other side of an OCO, there are defenders who have to actively learn about defensive techniques in order to mitigate offensive techniques. This community would benefit from a curated list or repository that expands our work. Also, MITRE recently released Shield, an active defense knowledge base. Shield could be used to find methods to mitigate various techniques and offensive security tools [19]. Finally, the curated list could include additional data sources, such as fingerprints of offensive security tools, etc.

In the future, we have plan to construct a automated penetration testing tool using reinforcement learning. While there is research in this domain, we believe that more automation can be applied to offensive security based on the tools analyzed. Creating a tool capable of automating more phases of an OCO could help inspire research in this direction. It will also allow a better understanding of the strengths and weaknesses of automation in various OCO phases.

11 RELATED WORK

The following section presents several works that provided significant contributions or information related to our work. Works related to our OCO process model and/or our taxonomy of the algorithms implemented to automate OCO processes are described in Section 11.1. Works related to our taxonomy of the algorithms implemented to automate OCO processes are described in Section 11.2.

11.1 Work Related to our OCO Process Model

The MITRE ATT&CK Framework is an opensource knowledge base and model of APTs' behavior [9]. ATT&CK includes adversary tactics, techniques, and procedures (TTPs) from real-world observations [40]. ATT&CK's purpose is to improve post-compromise detection of adversaries operating within enterprise networks by using analytics [41]. Numerous attackers and defenders in the cybersecurity community use the ATT&CK to gain a better understanding of attack vectors, and allow organizations to prioritize those risks [39].

[20] presents a canonical model of Nation-State intervention in impending and incoming OCOs. This model is based on real-life experience in cyber operations and information from additional case studies. It contained outdated and inconsistent use of OCO-related terminology, such as when referring to the person conducting the operation as an operator, attacker, or intruder with no specific differentiation between the terms [25]. The model only encompasses government-led OCOs, and was created by SADT composition.

[36] was intended to model penetration tests and allow for the creation of systematized attack graphs. These attack graphs would diagram the possible sequence of events during a penetration test. The models includes the goals, assets, actions, and agents of the attack. The presentation also included a tool developed to automate the creation of an attack graph for penetration tests when given a target and additional related information.

[44] contributes both taxonomies and a formalized ontology for Network Attack Classification. Some of the defined network taxonomies include the Actor (e.g. Nation-State or Corporation), Attack Goals, Automation Level, Effects, Motivation, Phase, and more. Some of the sub-processes of this taxonomy are derived from the attack mechanism which, in essence, includes areas such as System Abuse, Information Gathering, Denial of Service, Exploit, and Malware. This work takes the defined taxonomies and develops an ontology that can be used to describe various OCOs.

[29] proposes a taxonomy of Distributed Denial of Service (DDoS) attack mechanisms in order to provide researchers with a greater understanding of the various methods of performing DDoS attacks. Their work provides an overview of the general DDoS attack process phase by phase. Additionally, their taxonomy includes the classification criterion "degree of automation", which can be used to identify automated DDoS attack mechanisms such as tools that automate DDoS attacks.

11.2 Work Related to our Quantitative Analysis

[43] presents a diagram of the malicious uses of AI in offensive cybersecurity. This diagram includes specific processes performed in OCOs that can be automated with AI such as developing malware and performing social engineering. They also contribute a table containing a collection of tools that use AI to automate processes performed in OCOs, identifying the specific type of AI algorithm(s) each one implements.

[21] systematically identifies various tools and technologies available for use in government-led OCOs. Many of these tools are not necessarily restricted use in government-led OCOs, such as the password cracking and network mapping tools. The tools were organized into a table showing which step(s) of a government-led OCO each type of tool could be used in. The steps were those established in [20].

12 CONCLUSION

The complexity of offensive security tools is expanding as new vulnerabilities are discovered and patched. Thus, the future of offensive security tool development may rely upon the use of various forms of automation. The constant evolution of offensive security tools allows actors to continuously adapt to evolving organizational defenses. Additionally, more phases in OCOs need to be automated to allow for fully autonomous OCOs. Our analysis shows that while the MITRE ATT&CK Framework is extremely robust and resourceful, some techniques are too broad. These techniques (such as Exploit Public-Facing Application T1190) could be expanded upon to provide a higher level of specificity to its users.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Number 1921813 Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Joint Chiefs of Staff 2018. *Cyberspace Operations.* Joint Chiefs of Staff. https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp3_12.pdf
[2] Kaspersky 2019. *Tips on how to protect yourself against cybercrime.* Kaspersky. https://www.kaspersky.com/resource-center/threats/what-is-cybercrime
[3] Crowdstrike 2020. *What is Cyber Threat Intelligence? [Beginner's Guide].* Crowdstrike. https://www.crowdstrike.com/epp-101/threat-intelligence/
[4] OWASP 2022. *A8:2017-Insecure Deserialization.* OWASP. https://owasp.org/www-project-top-ten/2017/A8_2017-Insecure_Deserialization
[5] Lockheed Martin 2022. *Cyber Kill Chain®.* Lockheed Martin. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
[6] The MITRE Corporation 2022. *Enterprise Matrix.* The MITRE Corporation. https://attack.mitre.org/matrices/enterprise/
[7] The MITRE Corporation 2022. *Enterprise Techniques.* The MITRE Corporation. Retrieved March 7, 2022 from https://attack.mitre.org/techniques/enterprise/
[8] The MITRE Corporation 2022. *Exploit Public-Facing Application.* The MITRE Corporation. https://attack.mitre.org/techniques/T1190/
[9] MITRE Corporation 2022. *MITRE ATT&CK®.* MITRE Corporation. Retrieved March, 7, 2022 from https://attack.mitre.org/
[10] Office of the Director of National Intelligence 2022. *ODNI Home.* Office of the Director of National Intelligence. https://www.dni.gov/index.php/what-we-do/what-is-intelligence
[11] OWASP 2022. *OWASP Top Ten.* OWASP. https://owasp.org/www-project-top-ten/
[12] The MITRE Corporation 2022. *PRE-ATT&CK Matrix.* The MITRE Corporation. https://attack.mitre.org/matrices/pre/
[13] BetaFred. 2020. *Microsoft Security Bulletin MS17-010 - Critical.* https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010
[14] Carly Burdova. 2020. . Avast. https://www.avast.com/c-eternalblue
[15] Defense Use Case. 2016. Analysis of the cyber attack on the Ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)* 388 (2016).
[16] Zak Doffman. 2019. *Israel Responds To Cyber Attack With Air Strike On Cyber Attackers In World First.* https://www.forbes.com/sites/zakdoffman/2019/05/06/israeli-military-strikes-and-destroys-hamas-cyber-hq-in-world-first
[17] Dave Evans. 2017. *So, What's the Real Difference Between AI and Automation?* https://medium.com/@daveevansap/so-whats-the-real-difference-between-ai-and-automation-3c8bbf6b8f4b
[18] FireEye. 2020. Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor. https://www.fireeye.com/blog/threat-research/2020/12/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.html
[19] Christina Fowler. 2020. *Welcome to the MITRE Shield Blog!* https://medium.com/mitre-shield/welcome-to-the-mitre-shield-blog-55d74f385bfc
[20] Tim Grant, Ivan Burke, and Renier Van Heerden. 2012. Comparing models of offensive cyber operations. *Proceedings of the 7th International Warfare and Security* (2012), 108–121.
[21] Tim Grant and R. Prins. 2013. Identifying tools and technologies for professional offensive cyber operations. *8th International Conference on Information Warfare and Security, ICIW 2013* (01 2013), 80–89.
[22] Elike Hodo, Xavier Bellekens, Andrew Hamilton, Christos Tachtatzis, and Robert Atkinson. 2017. Shallow and deep networks intrusion detection system: A taxonomy and survey. *arXiv preprint arXiv:1701.02145* (2017).
[23] Gazmend Huskaj. 2019. The Current State of Research in Offensive Cyberspace Operations. In *Proceedings of the 18th European Conference on Cyber Warfare and Security.* Academic Conferences and Publishing International Limited, 660–667.
[24] Olakunle Ibitoye, Rana Abou-Khamis, Ashraf Matrawy, and M Omair Shafiq. 2019. The Threat of Adversarial Attacks on Machine Learning in Network Security–A Survey. *arXiv preprint arXiv:1911.02621* (2019).
[25] L.J. Janczewski and A.M. Colarik. 2007. *Cyber warfare and cyber terrorism.* 1–532 pages. https://doi.org/10.4018/978-1-59140-991-5
[26] Y. Kolli, T. K. Mohd, and A. Y. Javaid. 2018. Remote Desktop Backdoor Implementation with Reverse TCP Payload using Open Source Tools for Instructional Use. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON).* 444–450.
[27] Laramies. 2022. *laramies/theHarvester.* https://github.com/laramies/theHarvester
[28] Ryusei Maeda and Mamoru Mimura. 2021. Automating post-exploitation with deep reinforcement learning. *Computers & Security* 100 (2021), 102108.
[29] Jelena Mirkovic and Peter Reiher. 2004. A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *ACM SIGCOMM Computer Communication Review (COMPUT COMMUN REV)* (2004).
[30] Shimon Y. Nof. 2009. *Automation: What It Means to Us Around the World.* Springer Berlin Heidelberg.
[31] Erwin Orye and Olaf Maennel. 2019. Recommendations for Enhancing the Results of Cyber Effects. 1–19. https://doi.org/10.23919/CYCON.2019.8756649
[32] Charlie Osborne. 2018. DeepLocker: When malware turns artificial intelligence into a weapon. https://www.zdnet.com/article/deeplocker-when-malware-turns-artificial-intelligence-into-a-weapon/
[33] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security.* 506–519.
[34] Adam Pennington. 2020. *Bringing PRE into Enterprise.* https://medium.com/mitre-attack/the-retirement-of-pre-attack-4b73ffecd3d3
[35] M. Rigaki and S. Garcia. 2018. Bringing a GAN to a Knife-Fight: Adapting Malware Communication to Avoid Detection. In *2018 IEEE Security and Privacy Workshops (SPW).* 70–75.
[36] Carlos Sarraute. 2010. Using AI Techniques to improve Pentesting Automation. (04 2010).
[37] Jonathon Schwartz and Hanna Kurniawati. 2019. Autonomous penetration testing using reinforcement learning. *arXiv preprint arXiv:1905.05965* (2019).
[38] John Seymour and Philip Tully. 2016. Weaponizing data science for social engineering: Automated E2E spear phishing on Twitter. *Black Hat USA* 37 (2016), 1–39.
[39] Blake Strom. 2018. *ATT&CK 101.* Retrieved March, 7, 2022 from https://medium.com/mitre-attack/att-ck-101-17074d3bc62
[40] Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington, and Cody B Thomas. 2018. Mitre att&ck: Design and philosophy. *Technical report* (2018).
[41] Blake E Strom, Joseph A Battaglia, Michael S Kemmerer, William Kupersanin, Douglas P Miller, Craig Wampler, Sean M Whitley, and Ross D Wolf. 2017. Finding cyber threats with ATT&CK-based analytics. *The MITRE Corporation, Bedford, MA, Technical Report No. MTR170202* (2017).
[42] Isao Takaesu. 2018. *Black Hat Europe 2018.* Black Hat. https://www.blackhat.com/eu-18/arsenal/schedule/#deep-exploit-fully-automatic-penetration-test-tool-using-machine-learning-13320
[43] Cong Truong, Diep Quoc Bao, and Ivan Zelinka. 2020. Artificial Intelligence in the Cyber Domain: Offense and Defense. *Symmetry* 12 (03 2020), 410. https://doi.org/10.3390/sym12030410
[44] Renier van Heerden. 2014. *A Formalised Ontology for Network Attack Classification.* Ph.D. Dissertation.

A APPENDIX - OCO TABLES

Table 1: Technique IDs Name

Technique ID	Name
T1190	Exploit Public-Facing Application
T1059	Command and Scripting Interpreter
T1003	OS Credential Dumping
T1113	Screen Capture
T1082	System Information Discovery
T1016	System Network Configuration Discovery
T1134	Access Token Manipulation
T1071	Application Layer Protocol
T1068	Exploitation for Privilege Escalation
T1056	Input Capture
T1087	Account Discovery
T1557	Man-in-the-Middle
T1110	Brute Force
T1548	Abuse Elevation Control Mechanism
T1210	Exploitation of Remote Services
T1057	Process Discovery
T1055	Process Injection
T1053	Scheduled Task/Job
T1555	Credentials from Password Stores
T1573	Encrypted Channel
T1105	Ingress Tool Transfer
T1090	Proxy
T1046	Network Service Scanning
T1550	Use Alternate Authentication Material
T1123	Audio Capture

Table 2: Quantitative Data for OCO Tools

Category	Percent
Operating System	
Linux	37%
Windows	31%
macOS	23%
WindowsXP	2%
FreeBSD	2%
OpenBSD	2%
Solaris	<1%
QNX	<1%
Country	
United States	56%
United Kingdom	8%
Australia	5%
Germany	5%
France	4%
Japan	4%
Nigeria	2%
Lithuania	2%
Argentina	2%
Netherlands	2%
China	2%
Austria	2%
Indonesia	2%
Canada	2%
Belgium	2%
Automation Type	
AI Automation	33%
Hybrid Automation	16%
General Automation	50%
Unknown	<1%
Algorithm Type	
Simple Rule based	45%
Expert System	31%
Fuzzy Logic	5%
Convolutional Neural Network	4%
State Modeling	4%
Decision Tree	4%
Deep Belief Network	3%
Bayesian Network	<1%
Recurrent Neural Network	<1%
Threshold	<1%
Adaptive Resonance Theory	<1%
Machine Learning	<1%
Mean and Standard Deviation	<1%
Markov Process	<1%
Genetic Algorithm	<1%
Clustering	<1%
Support Vector Machine	<1%
Multivariate	<1%
Publicly Available?	
Yes	80%
No	20%
Actively Maintained?	
Yes	60%
No	40%