An Automated Post-Exploitation Model for Cyber Red Teaming

Ryan Benito, Alan Shaffer and Gurminder Singh
Naval Postgraduate School, Monterey, California, USA

ryan.m.benito.mil@us.navy.mil
alan.shaffer@nps.edu
gsingh@nps.edu

Abstract: Red teaming is a well-established methodology for ensuring and augmenting cyber system security; however, the training, expertise, and knowledge of appropriate tools and techniques required to perform effective red teaming come with a significant cost in time and resources. Large organizations such as the Department of Defense (DOD) use vulnerability assessment to identify software patches and other remediations for cyber systems to mitigate cyberspace exploitation. If a patch cannot be applied in a timely manner, for instance to minimize network downtime, measuring and identifying the impact of such unpatched vulnerabilities is left to scarce red teaming services. These services typically concentrate on initial access exploitation, which stops short of exploring the larger security impacts of cyber threats performing post-exploitation actions. This gap in post-exploitation red team analysis results in increased susceptibility to adversary offensive cyberspace operations (OCO) against DOD systems. This research extends the Cyber Automated Red Team Tool (CARTT), developed at the Naval Postgraduate School, by implementing automated red team post-exploitation analysis. The intent of this extended capability is to reduce the workload on limited DOD red teams and penetration testers by providing system administrators with the ability to perform deeper system analysis for the impacts of exploited vulnerabilities.

Keywords: Red teaming, automated cyber post-exploitation, defensive cyber operations, web-based assessment.

1. Introduction

The FY23 U.S. Department of Defense (DOD) budget increased funding for "cyberspace efforts" by 8% to $11.2 billion, indicating a greater need for cyberspace operations (CO) resources. Although this is a significant increase, CO resources like cyber protection teams (CPT), cyber mission teams (CMT), red teams, and national mission teams (NMT), all of which conduct offensive cyberspace operations (OCO), defensive cyberspace operations (DCO), and DOD information network (DODIN) operations, continue to be scarce throughout the DOD. Moreover, the DOD requires significant investment in time, skill, resources, and human capital to provide critical cyberspace insights, such as vulnerability assessments, threat emulation, and network hardening.

The scarcity of CO resources may leave low priority organizations without critical cybersecurity support, forcing them to wait for resources, and to potentially remain vulnerable to cyberspace attack or exploitation. DOD organizations do not currently have a tool that enables non-experts to self-assess their systems to triage and prioritize which vulnerabilities should be patched based on relative risk.

Prior to the work reported here, previous research and development on the Cyber Automated Red Team Tool (CARTT) at the Naval Postgraduate School in Monterey, CA, has automated the process for vulnerability assessment and some aspects of initial access exploitation, enabling users to assess the success or failure of a given initial access exploit against hosts on their systems. However, CARTT did not provide the ability to conduct post-exploitation assessment of these systems (Berrios 2020).

This paper summarizes research to develop and implement an automated post-exploitation capability for CARTT. We also summarize an impact-based post-exploitation taxonomy derived from the MITRE ATT&CK® framework to help organizations more accurately identify vulnerability impact and risk in implementing security controls (Benito 2022).

The rest of this paper is organized into five sections: related work, an impact-based post-exploitation taxonomy, system design, system implementation, and a summary of conclusions and future work.

2. Background

2.1 Cyber Post-Exploitation

Post-exploitation actions occur after a cyberspace attacker gains initial access onto an adversary's system. Without initial access exploitation, post-exploitation actions would not be feasible. Firewalls and intrusion detection and prevention systems can support perimeter defense strategies by keeping adversaries out of the network, however, a major vulnerability of a perimeter defense strategy is assuming that users within the

perimeter can always be trusted. Adversaries that can pierce perimeter defenses may exploit this assumed trust to perform post-exploitation actions.

The National Institute of Standards and Technology (NIST) and the DOD have created the Zero Trust Architecture (ZTA) and framework to thwart the implicit trust granted to assets or user accounts based solely on their physical or network location or based on asset ownership (DOD 2021; Rose et al 2020). Zero trust is not definitive though. Motivated adversaries will continue to find creative ways to conduct post-exploitation despite new security measures offered by a ZTA implementation. The tools, techniques, and procedures of post-exploitation actions can provide the necessary indicators of compromise to enable operators to identify and eradicate adversaries from the network. Some post-exploitation actions can include persistence, privilege escalation, discovery, lateral movement, command and control, and exfiltration (MITRE 2021).

Persistence should be the first post-exploitation action utilized once the cyberspace operator is able to gain initial access to a system. Persistence allows the operator to maintain long-term presence within a dynamic cyberspace environment. Once persistence is achieved, the number of malicious activities is bounded by the objective of the attacker and the defensive posture of the system.

Once persistence has been achieved, the attacker can conduct discovery actions under the disguise of a trusted user. The goal of discovery is to understand the targeted environment to support CO (MITRE 2021). Discovery can encompass accounts on the user's machines, internet bookmarks, directory services, and system information. Although user privileges can provide an abundance of information, the cyber operator may still need to perform privilege escalation for subsequent cyber actions. To conduct lateral movement, the attacker may need to conduct initial access actions, discovery, and privilege escalation post-exploitation actions within the network to achieve the desired cyber objective(s). This may include establishing an encrypted channel for command and control from the endpoint, followed by exfiltration of information (MITRE 2021). An alternate objective could be to conduct a cyberspace attack that could harm systems or data. Post-exploitation offers the attacker many different avenues to achieve desired objectives.

2.2 Post-Exploitation Frameworks

There are numerous open-source exploitation frameworks currently used to perform initial and post-exploitation actions. Although there are many niche frameworks that address specific use cases, frameworks such as Penetration Testing Execution Standard (PTES) and ATT&CK framework are agnostic of specific use cases, which can be tailored for post-exploitation actions.

PTES was created in 2009 to "provide both businesses and security providers with a common language and scope for performing penetration testing" (PTES 2014). PTES comprises two major sections: the standard and the technical guidelines. The standard has seven tailorable sections that explain how to plan, test, and report the results of a penetration test. The technical guide provides a baseline process of how to carry out different aspects of the penetration test. The post-exploitation portion consists of five tailorable sections. Although the PTES provides the base understanding of penetration testing, it requires an experienced operator to implement processes to connect the technical to the standard aspects of PTES.

The main concept of ATT&CK is to prioritize cyberspace defense based on "documented threat behavior" by understanding an adversary's tactics and techniques (Strom 2018). A tactic answers the "why" or the reason an operator performs an action. A technique is the "how" an attacker will achieve the tactic. The menu of tactics and techniques can be aggregated to create an attack scenario to match an Advanced Persistent Threat (APT), such that it can be simulated by red teamers to test and help network defenders identify the specific techniques utilized to understand where to search for compromise.

Unlike the PTES, the ATT&CK framework does not have a post-exploitation section. This is because an attacker can use both initial access and post-exploitation actions to achieve an objective. The lack of boundaries in the ATT&CK framework provides the flexibility required to track adversarial behavior from initial access to post-exploitation.

2.3 Post-Exploitation Tools

There are a variety of post-exploitation tools that are utilized by penetration testers and red teamers. Unfortunately, the DOD lacks resources to meet the required need of exploitation services. To make matters worse, the process to certify and accredit red teams "does not evaluate a red team's ability to portray validated

threat actors," so red teams "gravitate to low hanging fruit" such as known initial access exploits, which network defenders can readily identify (Schab 2021).

We researched three open-source tools that automate aspects of post-exploitation actions: Automated Network Exploitation Through Penetration Testing (ANEX), Scalable Automated Vulnerability Scanning & Exploitation Tool (SAVE-T), and Automated Deep Learning Post-Exploitation (Booz 2020; Dazet 2016; Maeda & Mamoru 2020).

These automated post-exploitation tools have a variety of limitations. First, each tool only tests a certain subset of the foundational ATT&CK cyber actions, which are discovery, persistence, privilege escalation, and lateral movement. The post-exploitation actions covered in ANEX, SAVE-T, and Automated Deep Learning tools are lateral movement and privilege escalation. SAVE-T delves into command and control and exfiltration actions, but agents are placed on target computers prior to the start of red teaming or penetration testing, circumventing the need to identify the risk associated with initial access vulnerabilities. To give network defenders a true assessment of external and internal cyberspace defenses, many threat scenarios require assessing initial access vulnerabilities along with post-exploitation actions.

Second, each implementation suffers from user misuse error. ANEX could crash the targeted system based on the type of exploit utilized. SAVE-T requires a remote access trojan (RAT) to be installed on each workstation such that if the SAVE-T server is compromised, all workstations are compromised. The deep learning tool requires the user to choose probability thresholds to train an agent, resulting in mixed assessments. Lastly, each implementation requires in-depth knowledge of the system architecture, moderate user interaction, and risk analysis of performing an automated assessment.

The U.S. Navy uses a tool called Vulnerability Remediation Asset Manager (VRAM) that gives enterprise visibility into a network's baseline configuration protecting it from known initial access vulnerabilities (Barnett 2020). VRAM splits vulnerabilities into various categories, a program-of-record owned (POR-owned) vulnerability is the most severe (SPAWAR 2017). A POR-owned vulnerability requires the owning program office to investigate and provide remediation procedures to the on-site network defenders. On-site network defenders are either left to operate with POR-owned vulnerabilities or be forced to take the associated system or component offline. To make matters worse, there could be many POR-owned vulnerabilities that need to be analyzed for their potential impact on operations. If the organization decides to operate with POR-owned vulnerabilities, dwell time is increased by the adversary executing persistence post-exploitation actions.

FireEye defines dwell time as "the number of days an attacker is present in a victim environment before they are detected." In 2021, that dwell time was 21 days (FireEye 2022). From the time a vulnerability is identified, it could take on-site network defenders 21 days to detect an adversary. The time between notification, investigation, and implementing remediation procedures allows adversaries to conduct post-exploitation actions.

If the network defenders are experienced penetration testers, they can employ post-exploitation actions to understand the impact of the exploited POR-owned initial access vulnerability. Typically, network defenders are not trained to employ post-exploitation actions and they must compete for scarce post-exploitation services. These services require a request for resources from U.S. Fleet Cyber Command, which is adjudicated with U.S. Cyber Command. The further down the chain a command is from the combatant command, the less likely the request will be fulfilled. Once a red team or penetration tester performs post-exploitation, network defenders are given a technical report outlining successful post-exploitation actions. Oftentimes these actions are performed using proprietary tools which cannot be re-enacted or reemployed to understand network effects.

An automated post-exploitation tool fills the gap between normal operations and awaiting manual penetration and red teaming services. Network defenders do not have a reliable automated tool or taxonomy that requires both minimal CO expertise and interaction that provides an insightful assessment.

3. Impact-Based Post-Exploitation Taxonomy

We implemented the most important post-exploitation actions for OCO: discovery, persistence, privilege escalation, and lateral movement. Without these foundational actions, other cyberspace post-exploitation actions like cyberspace attack, command and control, and exfiltration would not be possible. Cyberspace attack and command and control require, at a minimum, the actions of discovery to ensure that the proper system is being targeted, and persistence to maintain access to the target.

In addition to discovery and persistence, exfiltration will also require privilege escalation to use privileged user services like file transfer protocol for data movement. If the data to exfiltrate must traverse multiple networks, lateral movement will be required to establish the exfiltration path.

To understand the impact of these foundational actions, a network defender could employ an ATT&CK inspired impact-based post-exploitation taxonomy based on red team, penetration testing, and CARTT reports to assess the qualitative impact of post-exploitation actions depicted by Figure 1.

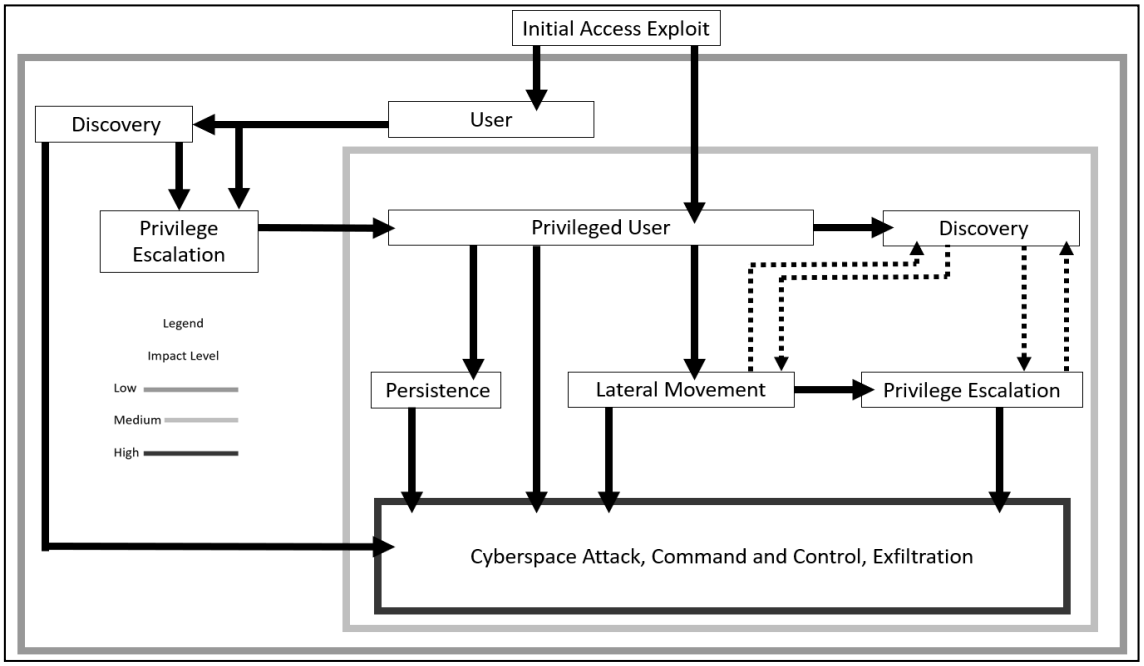

Figure 1: Impact-based post-exploitation taxonomy

The initial access exploit is the first step into either a user or privileged user system. A user system could support various business objectives which could threaten organizational security. A privileged user system provides greater access and authorization in an enterprise. Based on business impact analysis, the impact of exploiting a user or privileged user system can be defined. The impact taxonomy relies on successfully completed post-exploitation actions adapted from the ATT&CK framework. A successful completion of a post-exploitation action results in a low, medium, or high impact level. Impact level determination guidelines that could be used are the Federal Information Processing Standards Publication 199 or the Chairman Joint Chiefs of Staff Instruction 6510.01B (CJCS 2012). The impact levels can be adapted to fit any organization's security concerns of availability, integrity, and confidentiality.

If a user workstation becomes the victim of an initial access exploit, it could be considered a low impact on business operations availability if that workstation is lost. Although an adversary could disrupt business operations by taking many user workstations offline, it may or may not amount to a high-impact cyberspace attack based on business impact analysis. A user could easily switch to another workstation, or the security operations center could enact business continuity procedures to shift operations to another site, if the business impact reaches the established threshold for a cyberspace attack. If the adversary executes discovery actions to find that the workstation belongs to a high value target, they can conduct privilege escalation to either laterally move to a user system, persist on the exploited user system, or continue discovery actions at the privileged user level. If the adversary is a privileged user, they could potentially shut down the workstation denying access to business services, or laterally move into and exploit other user systems and perform shutdowns. The impact level of the adversary is raised to medium due to the scale and potential effect they could have on availability. If the adversary gained access to a privileged user system, a medium or high impact would immediately be assessed due to the ease of gaining access which could result in a cyberspace attack that results in a loss of availability for a privileged user system.

Based on the report of successful post-exploitation actions employed, the user can understand, assess impact, and mitigate risk using the impact-based post-exploitation taxonomy.

4. Design

CARTT utilizes a client-server architecture to enable users to complete cyber actions as shown in Figure 2. In the previous version of CARTT, the Operator user role was limited to vulnerability scanning and initial access exploitation, but it has now been expanded to include post-exploitation actions. CARTT leverages the Greenbone Vulnerability Management system (GVM) – formerly called OpenVAS – and the Metasploit Framework (MSF) to enable access to its expanded functionality. These tools were chosen primarily due to their open-source nature, which provides free access to the tools, and ensures real-time improvements from the open-source community. The CARTT server provides all the components required to conduct OCO.

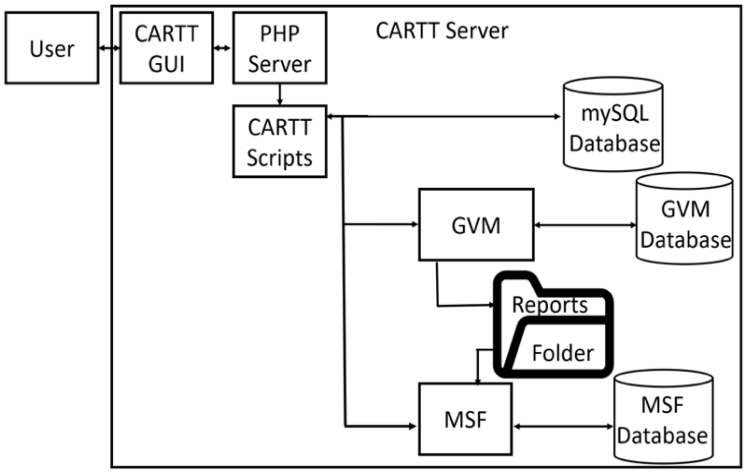

Figure 2: CARTT server architecture

The user interacts with the CARTT server via its graphical user interface (GUI) to perform cyber actions (Gomandakoye 2021). Based on the user interaction, the PHP server parses user input to create CARTT script interaction. The CARTT scripts enable communication between the following three systems: the mySQL database, GVM, and MSF. The mySQL database is used for the messaging system between users. GVM, a public vulnerability scanner maintained by Greenbone Networks, identifies vulnerabilities by conducting vulnerability scans on a system based on feeds comprised of Common Vulnerabilities and Exposures (CVEs), security communities, Greenbone labs, and input from technology partners (Greenbone 2021). The feeds are stored in the GVM database. Once a vulnerability scan is complete, its results are stored in the reports folder. From the reports folder, a scan report is imported by the CARTT Operator into MSF for exploitation actions, as shown in Figure 3, the CARTT Server process architecture.

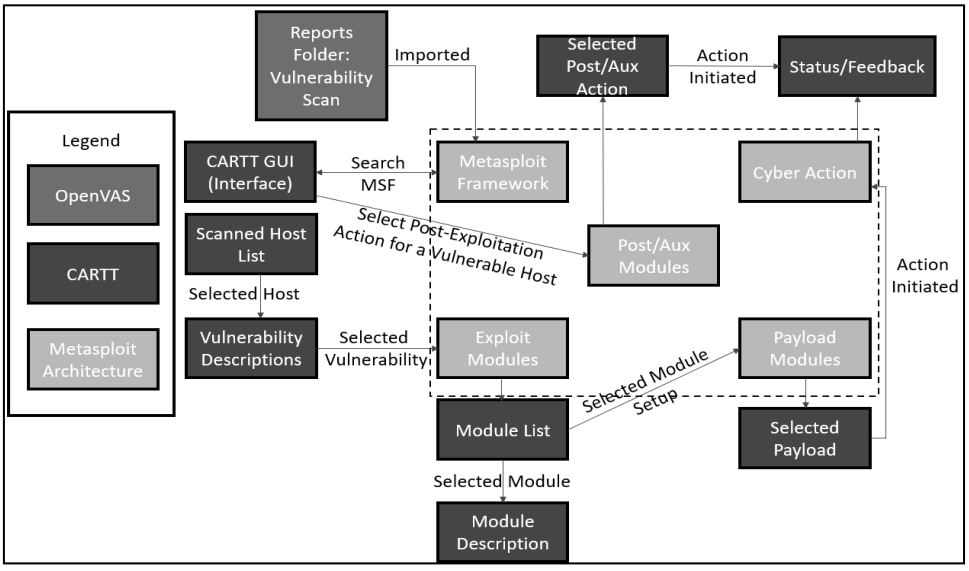

Figure 3: CARTT server processes

Each vulnerability scan is placed into a reports folder and then imported into MSF. The CARTT Operator is presented with the scanned host list in the CARTT GUI to select a host and view the related vulnerability descriptions (shown in purple in Figure 3). The Operator then selects a vulnerability, which populates a list of applicable exploit modules. The Operator can view a module description to aid with selection. Once a module has been selected, the Operator will select a payload, then verify the exploit and payload to initiate the initial access exploit cyber action. Once the initial access exploit is complete, the Operator will receive its status and feedback from the CARTT GUI. The Operator can now select and configure post-exploitation actions, as shown in Figure 3. After a post-exploitation action has been completed, the Operator can view its status and any system feedback on its success or failure.

When a user logs into CARTT as an Operator, they are directed to a webpage to enter the target IP and subnet to conduct a new vulnerability scan. Once the scan is completed, the Operator can import the completed scan and then begin target exploitation from a list of available target hosts. The Operator will then submit a selected Target Host IP, which prompts CARTT to provide a list of vulnerabilities for that target. After selecting an available vulnerability by its CVE, the Operator will be provided a list of MSF exploit modules from which to launch an initial access exploit. CARTT presents the results of this exploitation to the Operator, who then has the option to use the same or a different exploit to conduct post-exploitation actions.

In post-exploitation, the CARTT Operator can choose from among several ATT&CK post-exploitation actions or test all actions. Upon selecting a post-exploitation action, the Operator is directed through various workflows that require minimal input and interaction to complete the cyber action. Once a post-exploitation workflow has completed, the Operator will be presented its results, as in Figure 4.

Figure 4: Post-exploitation results display

In Figure 4, the results show that the initial access exploit was successful since a Meterpreter payload session was opened. In this example, the Operator chose to use the privilege escalation post-exploitation action. The ATT&CK technique for conducting privilege escalation leveraged *process injection* using the MSF ***post/windows/escalate/getsystem*** module (highlighted in Figure 4), which exploits a race condition on the target system.

After execution of the MSF module ***post/windows/escalate/getsystem*** the results are provided to the CARTT Operator, showing the successful execution of privilege escalation. Based on this, one could refer to the impact-based post-exploitation taxonomy discussed in Section 3 to determine whether the initial access vulnerability, if exploited, would result in a medium-level impact to the organization. The organization could then implement appropriate security controls to mitigate the risk impact.

5. Implementation

This section discusses the virtualized environment used for implementing the new post-exploitation capabilities in CARTT. It also describes the resource scripting and workflows used for implementing specific post-exploitation actions.

5.1 Environment

The implementation environment used for CARTT is shown in Figure 5. The machines used for implementation were virtualized through VMware Workstation Pro. The environment was divided into two networks: Network-A depicting an operational environment, and Network-B a developmental environment. The operational environment is used to perform business tasks, while the development environment is used to update the products that users utilize to complete business tasks. A CARTT operator could move laterally between the two environments to complete a CO objective.

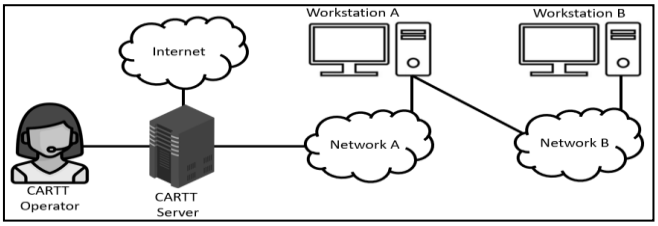

Figure 5: Implementation environment

The CARTT server is connected to Network-A and to the internet, whose connectivity is required to utilize GVM capabilities. Workstation-A in Network-A is dual-homed, since it has an additional connection to Network-B. Workstation-B is connected only to Network-B. The additional workstation was used to implement and test automation of the lateral movement post-exploitation action. For testing privilege escalation, which was the post-exploitation action example discussed in Section 4, CARTT gained access to Workstation-A using an initial access exploit, and then executed the privilege escalation post-exploitation action. All post-exploitation actions of discovery, persistence, privilege escalation, and lateral movement were developed, implemented, and successfully tested using this implementation environment.

5.2 Resource Scripting

CARTT leverages Metasploit resource scripting to allow MSF interoperability with the PHP web-based CARTT GUI. To implement post-exploitation actions, MSF post modules were appropriately added and logically chained in various Metasploit resource scripts and PHP webpages. The resource script enables automation by aggregating multiple Metasploit commands. The code for the initial access exploit resource script is shown in Figure 6.

Code

```
$fd_rc = fopen("user_data/exploit_$user.rc", 'w+');

$script =
"$exploit\n
set RHOST $host\n
set ExitOnSession false\n
exploit -z\n
exit -y\n";

fwrite($fd_rc, $script);
fclose($fd_rc);

$cmd = "msfconsole -q -r
user_data/exploit_$user.rc";
exec($cmd);
```

Figure 6: MSF initial access resource script

A file named *exploit_$user.rc* is the resource script file created using the C library function *fopen*, which is stored in the variable named *$fd_rc*. The *$fd_rc* will store the *$script* variable. The *$script* variable holds the set of Metasploit commands to complete the initial access exploit action. The *$exploit* is the initial access exploit that is set for use. The *set RHOST $host* is the IP address of the target. The *ExitOnSession* determines if session state will be maintained across multiple instances of Metasploit (Rapid7 2021). If the *ExitOnSession* option is set to true, and a Meterpreter session is obtained, the Meterpreter session will be terminated upon exit. If *ExitOnSession* is set to false, the Meterpreter session state is maintained upon exit.

To implement post-exploitation, *ExitOnSession* is set to false to maintain the open Meterpreter session to allow CARTT to execute post-exploitation actions. The *exploit -z* command runs the exploit and if a Meterpreter session is obtained, the session is placed in the background and returns to the MSF console environment. The *exit -y*

MSF command will exit the MSF environment. The *$script*, which holds the set of Metasploit commands to complete initial access exploitation is written to $fd_rc. The exploit_$user.rc resource script file is then executed through the Metasploit command line interface using the command *exec($cmd)* where *$cmd = msfconsole -q -r user_data/exploit_$user.rc*.

To implement post-exploitation actions, MSF post modules were appropriately added to the Metasploit resource script. It is important to note that the initial access exploit must succeed prior to performing any post-exploitation actions.

5.3 Workflows

Each post-exploitation menu option directs the CARTT Operator to a different webpage based on a PHP script that runs the menu. The webpage will prompt the Operator for a set of inputs to progress through a series of webpages that will eventually lead to the results of the post-exploitation action. The inputs are captured by the PHP POST method that stores the information in a PHP SESSION variable used across post-exploitation webpages (Chan 2020).

The CARTT post-exploitation actions of discovery, persistence, and privilege escalation follow a common workflow to arrive at providing results to the Operator, as shown in Figure 7 below. Each box in the workflow represents a different PHP webpage, and based on the CARTT Operator input, will either provide feedback or validate input prior to transition to the next webpage. The Operator must provide the required input to receive the results of the post-exploitation action.

The CARTT Operator will then click one of the post-exploitation actions of discovery, persistence, or privilege escalation, which will direct them to the "Configure Initial Access Exploit" page to provide the Vulnerability, Module, and Local Host IP Address. Upon submission, the action is completed, and results are displayed to the CARTT Operator (refer to Figure 4).

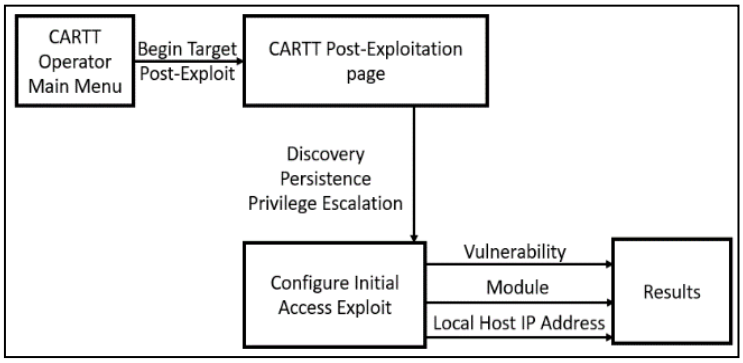

Figure 7: Post-exploitation workflow

The post-exploitation actions to perform lateral movement, or to test all post-exploitation actions, require two additional webpages to gather the required input from the Operator, as shown in Figure 8 below. If *Lateral Movement* is selected, the Initial Access Exploit Results will contain the results from executing a custom resource script CARTT MSF discovery module. If *Test All Actions* is selected, the Initial Access Exploit Results will also include the results of the *Persistence* and *Privilege Escalation* actions.

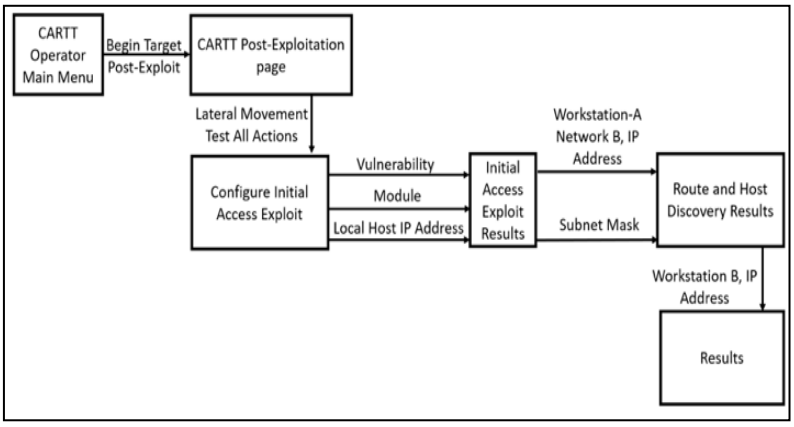

Figure 8: Lateral Movement and Test All Actions workflows

The CARTT Operator will use the custom resource script CARTT MSF discovery module results to provide the Workstation-A, Network-B IP address as well as the subnet mask. The Operator will then be presented the Route and Host Discovery Results and will be prompted to input the Workstation-B IP address. Upon submission the Operator will be presented with the results.

6. Conclusions and Future Work

This research expanded CARTT to support post-exploitation cyber actions by adding appropriate MSF modules to Metasploit resource scripts to perform discovery, persistence, privilege escalation, and lateral movement on a target host. CARTT uses MSF resource scripting coupled with PHP operability to automate the new post-exploitation actions. This capability provides the CARTT Operator with valuable information beyond initial access exploitation. CARTT also provides organizations with the capability to conduct self-assessment and impact-based analysis when higher-level cyberspace analysis resources are not available. Once organizations understand both external and internal impacts, they can harden their systems to increase their cybersecurity posture.

In future work, CARTT can be made more effective by expanding it to model obfuscation, stealth, and non-attribution to better model threat behaviors. To demonstrate the new post-exploitation capabilities, CARTT was tested against workstations that were not fully hardened, however, most real-world target organizations will have host and network intrusion and prevention devices implemented behind firewalls or within a DMZ. Another aspect to improve upon is automating initial access exploitation. If a CARTT Operator is unfamiliar with specific target vulnerabilities, this may require them hours of research to pair a vulnerability with an exploit module to achieve initial access. Finally, CARTT reporting and user feedback could be improved. The current system reporting requires the user to manually parse the results of a vulnerability analysis to find the proper inputs to complete a post-exploitation workflow. Future work could parse and present information in a cleaner manner to provide improved feedback and action suggestions to the user.

References

Barnett, J. (2020) “New cybersecurity tools sought for Navy’s VRAM program”, [online], FEDSCOOP, https://www.fedscoop.com/navy-cybersecurity-vram-cots/

Booz, J. (2020) “Towards scalable automated vulnerability scanning & exploitation”, [online], M.S. Thesis, Dept. of Info. Sci., Carnegie Mellon University, USA, https://kilthub.cmu.edu/articles/thesis/Towards_Scalable_Automated_Vulnerability_Scanning_Exploitation/12728360/1?file=24095777

Benito, R. (2022) “An automated post-exploitation model for offensive cyberspace operations”, [online], M. S. thesis, Naval Postgraduate School, Monterey, CA, USA, https://calhoun.nps.edu/bitstream/handle/10945/70631/22Jun_Benito_Ryan.pdf.

Berrios, J. (2020) “A client/server model for automated red teaming”, [online], M. S. thesis, Naval Postgraduate School, Monterey, CA, USA, https://calhoun.nps.edu/bitstream/handle/10945/66584/ 20Dec_Berrios_Joseph.pdf.

Chairman Joint Chiefs of Staff (2012) “CJCSM 6510.01b: Cyber incident handling program”, [online], https://www.jcs.mil/Portals/36/Documents/Library/Manuals/m651001.pdf?ver=2016-02-05-175710-897.

Chan, J. (2020), *Learn PHP in One Day and Learn It Well*. Hedge End, Hampshire, UK: LCF Publishing.

Dazet, E. (2016) “ANEX: automated network exploitation through penetration testing”, [online], M.S. Thesis, Dept of Comp. Sci., California Polytechnic State University, San Luis Obispo, USA, https://digitalcommons.calpoly.edu/theses/1592/.

Department of Defense (2021) “Department of Defense (DOD) zero trust reference architecture,” [online], https://dodcio.defense.gov/Portals/0/Documents/Library/(U)ZT_RA_v1.1(U)_Mar21.pdf.

FireEye (2022) “M-trends 2022”, [online], https://www.mandiant.com/resources/m-trends-2022.

Gomandakoye, O.M. (2021) “An expanded graphical user interface for web-based cyber automated red teaming tool (CARTT)”, [online], M. S. thesis, Naval Postgraduate School, Monterey, CA, USA, https://calhoun.nps.edu/bitstream/handle/10945/68326/21Sep_Goumandakoye_Ousmane.pdf.

Greenbone, “Security feed: The daily vulnerability update.” Accessed May 5, 2021 [Online]. Available: https://www.greenbone.net/en/security-feed/

Maeda, R. and Mamoru, M. (2020) “Automating post-exploitation with deep reinforcement learning”, *Comp. & Sec.*, vol.100, pp. 1-13, January, [online], https://doi.org/10.1016/j.cose.2020.102108.

MITRE, “ATT&CK matrix for enterprise,” Accessed July 31, 2021 [Online]. Available: https://attack.mitre.org/

NIST (2021) “Federal information processing standards publication standards for security categorization of federal information systems”, [online], https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.199.pdf.

PTES (2014), “Penetration testing execution standard,” Accessed July 14, 2021 [Online]. Available: http://www.pentest-standard.org/ index.php/Main_Page

Rapid7, “Meterpreter HTTP/HTTPS communication,” Accessed April 28, 2021 [Online]. Available: https://www.rapid7.com/blog/post/2011/06/29/meterpreter-httphttps-communication/

Rose, S., Oliver, B., Stu, M., and Connelly, S. (2020) “NIST special publication 800-207 zero trust architecture”, [online], https://csrc.nist.gov/publications/detail/sp/800-207/final.

Schab, J. (2021) “Tackling DOD cyber red team deficiencies through systems engineering”, [online], SANS Institute, Bethesda, MD, USA, https://sansorg.egnyte.com/dl/Bx3K1e25qH/?.

Space and Naval Warfare Systems Center (2017) “Vulnerability remediation asset manager 2.0: VRAM quick start guide”.

Strom, B. (2018) “The philosophy of ATT&CK”, [online], https://medium.com/mitre-attack/the-philosophy-of-att-ck-9e8f81aba119.