LOCKHEED MARTIN

Cyber COBRA (**C**ontextual **OB**jective **RA**ting)

Govindra Ganesh; Shaun Walmsley; Sharif Hassan, PhD

Lockheed Martin - Red Team

Abstract

The Lockheed Martin (LM) - Red Team (RT) assesses the security posture of the corporation by emulating adversarial tactics, techniques, and procedures (TTPs) while reporting identified, and exploited vulnerabilities during mission execution. To effectively report vulnerabilities discovered during a cyber test, LM RT developed the **C**ontextual **OB**jective **RA**ting (COBRA). COBRA evaluates the overall severity of cyber exploitation based on the context *(circumstances, setting, conditions)* in which the discovered findings reside. This helps to reveal an environment's resiliency to a cyber-attack and provides countermeasures based on this data.

Keywords—finding, severity, rating, complexity, criticality, cyber, exploitation

1. Introduction

Throughout years of performing cyber security testing, the LM Red Team explored various methods to rate "findings". Findings are the core of cyber security testing as they document issues uncovered during engagements or mission execution. For cyber testing to be effective, these findings must be communicated clearly so they can be remediated appropriately. Target environment stakeholders need to understand the impact and likelihood of exploitation for each discovery.

Typically, findings have a common rating of High, Medium, or Low. In some cases, there are informational and critical categories. As testing is conducted and potential findings are discovered, they have a pre-assigned rating, regardless of the circumstances in which those findings were discovered. For example, using a SQL injection exploit and reading data from a backend database is considered in most scenarios a "High" regardless of whether it is Internet facing or isolated, with visibility of the affected system limited to only one other entity [7]. Internet facing systems are

exposed to a larger threat landscape. Therefore, increasing the likelihood of exploitation. In that case, a finding rating of "High" might truly be appropriate. In the event it is an isolated system behind multiple firewalls, it becomes unlikely that the vulnerability will be exploited. Despite the differing factors, the finding still receives a rating of "High", which does not accurately reflect the risk it presents. Pre-canned ratings associated with a vulnerability are a useful baseline; they provide insight to the potential maximum impact that could occur but should not be applied blindly in any situation. Ratings should consider the context in which the vulnerability resides. The use of a "canned" ratings regardless of the circumstances is a continuing problem in cyber testing [9].

Rating findings based on the surrounding criteria establishes a consistent approach that fits the context by following a defined empirical method to derive the calculated ratings. Numerous industry methodologies and frameworks exist today, such as the Common Vulnerability Scoring System (CVSS) which attempts to rate a finding based on answering generic questions [1]. In addition, there is the Common Configuration Scoring System (CCSS) which measures the severity by focusing on software configuration issues [8]. Utilizing these frameworks as designed proved to be challenging when applied to the Lockheed Martin ecosystem, a defense contractor that hosts national security systems (NSS), and processes controlled unclassified information (CUI) [6, 11, 12]. The CVSS, though generic, broad-based, and lacking the ability to adapt, provided a foundation for additional research to pinpoint the necessary modifications that would yield consistent, accurate results. The team created the **C**ontextual **OB**jective **RA**ting (COBRA) framework using authentic cyber test findings from past engagements to adjust questions, categories, and more. COBRA was designed to determine a finding criticality (severity rating) from a cyber test exploitation standpoint. *Note that throughout this paper, the terms criticality and severity, mission and engagement may be used interchangeably at times.* Ultimately, COBRA asks initial cyber testing questions supported by sub-questions:

How complex was the attack? *(Trivial, Moderate, Complex*)

What is the impact of what was uncovered via exploitation?

Severity factors in both the impact of a discovered finding and the attack complexity, which results in a final finding rating. COBRA enables a cyber tester by providing a set of succinct questions to identify the true impact and likelihood of a finding to be exploited, which ultimately drives the mitigation priority and severity rating. Capturing this data supports reporting for mitigation, which could be immediate or delayed based on scoring.

This paper will review the CVSS framework, examine the differences between COBRA and CVSS, discuss COBRA's framework, and conclude with case studies using real data from prior cyber testing engagements.

2. CVSS Background

The National Institute of Standards and Technology (NIST) published the Common Vulnerability Scoring System in 2006 to research the struggles organizations experience when assessing the relative importance of a vulnerability [4]. Many penetration testing vendors, vulnerability scanning products, and software scanning solutions have proprietary methods to assign impact scores to a vulnerability that cannot directly translate to the business customer. Developed as an open research initiative, the CVSS attempted to overcome this problem by leveraging a set of metrics and NIST-equations to assist analysts in scoring vulnerabilities for their organizations. This scoring system is useful when used to evaluate the generalized risk of a particular vulnerability but does not articulate the specific risk it poses to the organization.

The CVSS scores are composed of three focus areas: Base, Temporal, and Environmental. The Base area represents the intrinsic qualities of a vulnerability. The Temporal group reflects the characteristics of a vulnerability that change over time. The Environmental metrics adjust the Base and Temporal severities to a specific computing environment

To be implemented as an effective tool, CVSS must be improved in several areas. [5]

3. COBRA and CVSS differences

CVSS served as a foundation for COBRA and was tailored to meet the LM RT's needs. COBRA is designed from an exploitation testing perspective, so the questions can be answered by a tester who may have incomplete insight into the specific system or device under test (e.g. gray box testing). COBRA incorporates concepts from CVSS base metrics as well as select temporal and environmental factors into a streamlined question set. The following table illuminates the elements COBRA leverages and the closest comparable items within the CVSS.

Table I: COBRA, CVSS Comparison

COBRA	Values	CVSS Comparable	Values
Attack Complexity			
Specialized Conditions	Y/N	Attack Complexity	Low, High
Discoverability	Easy, Med, Hard	-	-
Difficulty	Easy, Med, Hard	User Interaction	Y/N
Temporal Mod	-	Exploit Code Maturity	Unproven, PoC, Functional, High
Impact			
Point of Presence	IsoLAN, Intranet, Internet, Local	Attack Vector	Network, Adjacent, Local, Physical
Required Privileges	None, Low, High	-	None, Low, High
Confidentiality	Contextual	-	Application centric
Environmental Mod	-	-	Low, Med, High
Integrity	Contextual	-	Application centric
Environmental Mod	-	-	Low, Med, High
Availability	-	-	Application centric
Environmental Mod	-	-	Low, Med, High
Lateral Movement	Y/N	Scope Change	Y/N

Through trial and error using the CVSS framework in the LM ecosystem, the team identified deficiencies that motivated development of COBRA. For example, the CVSS scoring system relies on "reasonable worst-case assumptions" and does not account for context; root level access on a

printer is scored the same as SYSTEM level access on a Domain Controller. Environmental scoring adjusts this slightly, but not enough to account for the wide range of devices that the LM RT interrogates. Additionally, CVSS does not sufficiently account for attack execution difficulty. The single Low/High option in attack complexity does not provide enough variance and lacks objectivity. Finally, CVSS questions are complex, introducing subjectivity and requiring time to complete.

Section 4 articulates how COBRA values are calculated. Over two decades, the LM RT experimented with numerous methods of rating cyber findings. In most situations, the approach suited the general needs but lacked either objectivity, the ability to view trends in discoveries, or user friendliness.

4. COBRA Framework

The LM Red Team has a reputation of experience and excellence, and empirically identified several aspects of vulnerability scoring which could not be sufficiently addressed with CVSS. Red Team holds the advantage of full context: the attack builds on the team's previously discovered vectors and is carried out on a specific system with a known purpose. COBRA attempts to leverage that additional information to allow more precise and concise scoring.

The team also needed a scientific method to objectively determine the severity of findings. COBRA meets this objective with criteria that examine the complexity and impact of an attack to help determine a final severity rating. A high-level overview is presented in figure 1.

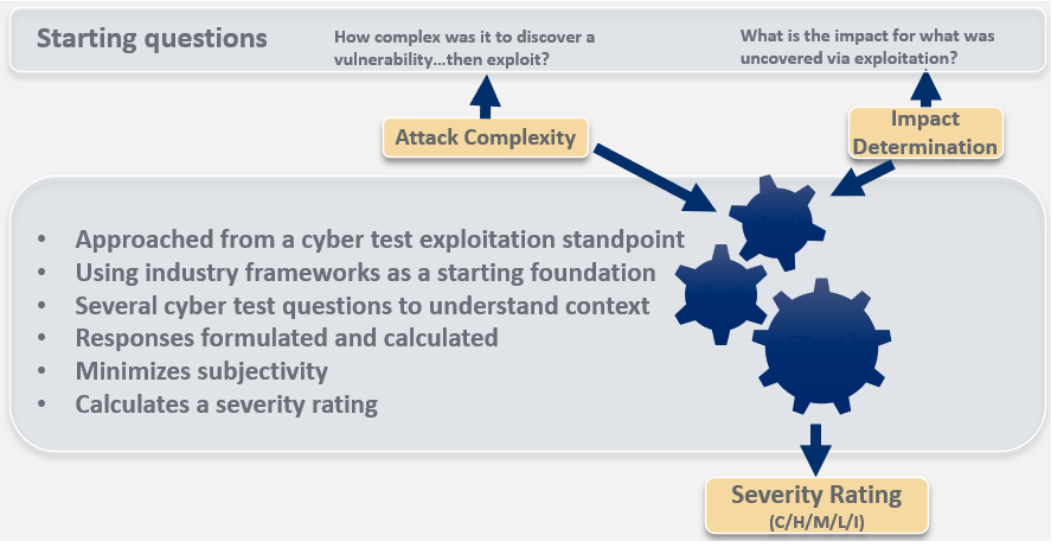

Figure 1: COBRA high level overview

Finally, it is important that the questions be framed from an emulated adversarial perspective. The tester may not know if a specific exploit resulted in a Scope Change as defined in CVSS, but they will know if lateral movement is successful. Similarly, it may be difficult to

distinguish between Low and High Confidentiality and Integrity ratings as defined in CVSS, but the tester will definitively know if they have accessed program data as opposed to a general system log.

*A. **Complexity, Impact, and Severity***

COBRA calculates severity or criticality based on attack complexity and impact. LM Red Team's definitions for each of these terms are listed below:

> *Attack Complexity: (Trivial, Moderate, Complex): A* combination of the expertise necessary to discover a vulnerability then execute an attack against it, tool accessibility to aid in the attack, and whether or not the attack relies on another attack or additional information. Contributes to the finding severity result.
>
> *Impact:* Determined by analyzing the initial point of presence, the compromised data as it pertains to the environment/system under test, and whether lateral movement is possible. Contributes to the finding severity result.
>
> *Finding Severity: (Informational, Low, Medium, High, Critical):* Determined based on attack complexity and impact. The scoring closely aligns with the CVSS severity rating model. If a finding is a newly discovered vulnerability not disclosed in the public domain, the team may pursue notification to a vendor for evaluation as a new potential Common Vulnerabilities and Exposure (CVE) [3, 10].

From a cyber security perspective, the ideal scenario is an attack requiring high complexity to succeed with minimal impact, resulting in a low severity rating.

*B. **Calculation***

LM Red Team rates finding severity using a modified CVSSv3[2] calculation which is customized to suit Lockheed Martin's corporate ecosystem and reduce the variance in input required to formulate a result.

To determine the attack complexity, weighted values are assigned to each Attack Complexity question, and static values are assigned to each response.

The value from each response is multiplied by the weighted value of the question, then the results are summed to generate an attack complexity score falling into one of the ranges in table III.

Values are assigned to impact-based responses and used in combination with the attack complexity score to determine the final COBRA score.

Table II: CVSS calculation

CVSS *(Without scope change)*	(6.42*(1-(1-C)*(1-I)*(1-A)) + (8.22*AttackVector* AttackComplexity*PrivlegeRequired * UserInteraction))

COBRA calculation

Attack Complexity
(ac1_wt * ac1_val) + (ac2_wt * ac2_val) + (ac3_wt * ac3_val) ac1 [Specialized Conditions - Reliant on another attack]: ac1_wt = 1 ac1_val = 0 (No), 9 (Yes) ac2 [Discoverability]: ac2_wt = 2 ac2_val = 1, 5, 10 ac3 [Difficulty]: ac3_wt = 10 ac3_val = 1, 5, 10
COBRA Score
((priv+lat_mv)*(1-(1-c)*(1-i)*(1-a)))+(pop*priv*ATTACK_COMPLEXITY) priv [Min required initial privileges] = 3.6, 5, 7 lat_mv [Lateral movement / pivoting] = 5, 225 c [Confidentiality] = 0, 0.04, 0.6, 1 i [Integrity] = 0, 0.45, 1 a [Availability] = 0 pop [Min required point of presence] = 0.2, 0.4, 0.45, 0.9

Some aspects of CVSS were intentionally excluded such as the *Temporal* values (Remediation Level and Report Confidence) as testing is performed in real time.

[Temporal] Exploit Code Maturity and User Interaction is instead incorporated into the Attack Complexity calculation.

Fixed Values were removed so that it is easier to modify the weighting.

5. Data, Discussion and Definitions

In practice, the team uses a custom developed web-based workflow system designed to track all aspects of a cyber testing mission, such as test cases, artifacts, findings, and results. Within the workflow system, findings are not assigned a severity rating by default. If a finding is determined

to be an “Informational” finding, a check box is selected. Informational findings aligns to the “none” CVSS severity rating. Answering additional COBRA questions will generate non-informational type ratings for findings. Despite COBRA’s relative accuracy, the implementation should allow for manual overrides based on attributes or factors not easily accounted for using the framework questions. In many testing scenarios, an item might be discovered that warrants documentation as an observational concern, although not a candidate for exploitation. In such cases, raising awareness of the issue with an informational rating will suffice.

COBRA is modified iteratively to coincide with evolving cyber testing capabilities and demands, with the understanding that the framework is not applicable in every scenario. The initial implementation of COBRA within LM RT demonstrated consistent output, indicating sufficient minimization of subjectivity. This established a reliable baseline from which COBRA has continued to evolve. Although values and weightings continue to be adjusted as new situations are encountered, the current state of COBRA provides excellent coverage for a wide range of testing scenarios. LM initially designed the COBRA framework to determine finding severity from an exploitation perspective; however, Red Team adjusted COBRA for compatibility with overt testing operations. The current version of COBRA can be used for collaborative test cases that examine visibility of an attack as well as threat hunting activities.

A. Attack Complexity

Three questions establish an attack complexity score, as seen in table III.

The score determines if the attack is trivial, moderate, or complex. A numerical value further defines each of these designations.

Table III: COBRA Attack Complexity Scoring

Attack complexity	**Score range**	**Definition**
Trivial	0-30	Low effort, minimal skills and readily available tools required for the attack to succeed
Moderate	31-60	Medium effort, intermediate skills and some custom tooling required for the attack to succeed
Complex	61-129	High effort, specialized skills and custom tooling required for the attack to succeed, 0day or similar level

The following three questions calculate the above scores:

1. Did this attack rely on another attack or on information from another attack? [**Yes/No** response]
    a. *Yes – this attack could not be executed without first executing the reliant attack*
    b. *No – this attack can be directly executed from the defined Point of Presence and Initial Privileges*

2. How difficult was it to **discover** the vulnerability?
    a. An automated tool identified the potential vulnerability with minimal effort *(e.g., dirbuster reveals accessible admin page)*
    b. An automated tool identified the service, and additional manual effort was required to identify the vulnerability *(e.g., nmap identified Tomcat on port 8081, research confirmed a vulnerable version of Tomcat)*
    c. Primarily a manual effort used to discover the vulnerability, using custom techniques *(e.g., nmap identified web service on port 443, manual efforts reveal a SQLi vulnerability)*

3. How difficult was it to **execute** an attack leveraging the discovered vulnerability?
    a. Readily available tools or MSF module
    b. Public Proof of Concept (PoC) with minor modifications; trivial but manual effort *(e.g., single quote SQLi that displays results)*
    c. Non-trivial manual effort; heavily modified public PoC; custom code

B. Severity Determination

A series of questions pinpoints the impact of what was uncovered. In this case, the severity rating is based on the CVSSv3 severity ratings in table IV.

Table IV: CVSSv3 severity ratings

Finding Severity	**Score range**	**Definition**
Informational	0.0	Observed during cyber testing but had no adversarial impact, or was not appropriate for exploitation
Low	0.1-3.9	Minimal adversarial impact, but may help an adversary further an attack
Medium	4.0-6.9	Moderate adversarial impact, plausible to help an adversary further an attack
High	7.0-8.9	High adversarial impact, highly probable to help further an adversarial attack

Critical	9.0-10.0	Severe adversarial impact, proven and used as a vector of attack to compromise system/data, remediation should be prioritized

A series of six questions determine the impact of a finding:

1. Minimum required point of presence
   Rather than the CVSS Attack Vector, this calculation uses a Point of Presence:

   a) **Internet** - An attack from the Internet carries the most weight
   b) **Intranet** - Includes the general Intranet as well as program networks where e-mail, internet browsing, chat etc. DO take place
   c) **Isolated environment** - Network/system where enterprise user activity *(e.g., e-mail, internet browsing, chat)* does NOT take place
   d) **Local Host Access** - Starting point where local access to a system is used as the origin for testing

2. Minimum required initial privileges
   There are three levels of privilege – Anonymous, General domain user, Authorized resource user.

   a) **Anonymous** – no credentials were provided to obtain the initial Point of Presence
   b) **General domain user** – the functional equivalent of Active Directory's "Authenticated Users" group
   c) **Authorized resource user** – to obtain the initial Point of Presence, credentials were used for an account which has explicitly been granted access to the resource *(e.g., network file share, web application, jumpbox)*

3. Confidentiality Loss (e.g. Read Data)
   Loss of confidentiality indicates the attack resulted in the disclosure of information to an unauthorized party. This is generally a "read only" impact. There are currently four defined levels of confidentiality impact - None, general, sensitive non-program, and sensitive program.

   a) **None** – no loss of data confidentiality
   b) **General data disclosure (low)** –debug data, trace data, version data, verbose messages etc.
   c) **Sensitive non-program data (medium)** – considering the privileges required, general (non-program specific) data marked company proprietary, or Export controlled (ECI)
   d) **Sensitive program and/or PII data disclosure (high) -** disclosure of Personally Identifiable Information (PII) is considered a critical confidentiality impact. Personal data as defined by the EU would also be an example. Company sensitive program specific information was obtained (Proprietary and/or Export Controlled), restricted use only information.

4. Integrity Loss (e.g. Write Data)
   Loss of integrity indicates the attack resulted in the ability to write or modify data in an unauthorized manner. There are currently three defined levels of integrity impact – None, general and program.

   a) None – no ability to modify any files
   b) **Modify general files** – the ability to modify systems or data that are not specific to the program.
   c) **Modify program data** – the ability to modify data which impacts program systems, ability to modify program specific data

5. Availability Loss (e.g. Denial of Service, Delete Data)
   This is currently unused by the team. LM Red Team testing is not intended to prove disruption via exploitation vectors or to cause destruction to data.

   a) **None**

6. Did the finding allow for lateral movement and/or pivoting?
   Scope Change indicates if the attack allowed for either lateral movement or pivoting.

   a) **No** *lateral movement or pivoting achieved*
   b) **Lateral movement and/or pivoting –** leveraging the attack to gain access to another system using either system configurations or discovered credentials *(e.g., a firewall allows the compromised host access to a remote system or service) or discovered credentials.* Pivoting – leveraging the compromised host to reach additional resources which are otherwise inaccessible *(e.g., multiple network interfaces allowing access to an isolated network).*

6. Use Cases

While there are several use cases, the main motivation for LM Red Team usage of COBRA is to remove or reduce subjectivity when determining the severity of cyber test findings. This standardizes the rating of a threat, which ultimately drives the mitigation priority. Consistency is another benefit within the risk determination process. COBRA ensures similar findings are rated the same regardless of the tester's skills. In practice, the COBRA model can be built into a web workflow system shown in figure 2, increasing user friendliness so that testers can easily use the framework during engagements.

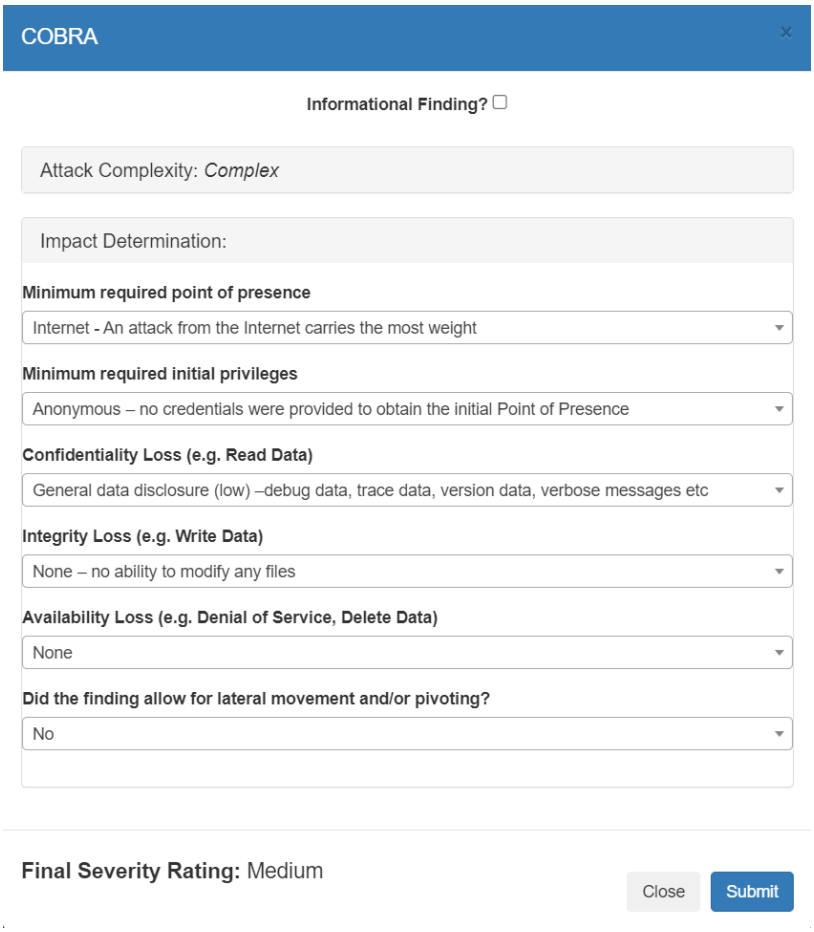

Figure 2: Web Workflow using COBRA

One example involved using COBRA and CVSS to determine the severity rating for a discovered vulnerability. The way the team answered both CVSS and COBRA questions were similar; however, due to understanding the LM environment and the type of data in play - the context of the attack and resulting data exposure - COBRA generated a critical finding vs. CVSS a high. This variance between the two frameworks helped determine the custom weighted values used in COBRA to produce a result with context specific to the LM ecosystem.

Tables V and VI compares COBRA and CVSS ratings for a theoretical misconfigured SMB file share in an environment that reveals sensitive program schematics to all Intranet based domain users. The final rating better reflects impact for the program specific data in COBRA as opposed to traditional CVSS.

Table V: Using COBRA to determine rating

	COBRA responses

Attack Complexity	Rely on another vector: No Ease of discovery: Easy Ease of execution: Easy Final attack complexity: Trivial
Point of Presence	Intranet
Initial Privileges Required	General domain user
Confidentiality (read data)	High (sensitive program data)
Integrity (write data)	None
Availability	None – placeholder / not used
Lateral Movement	No
Severity Rating	**Critical**

Table VI: Using CVSS to determine rating

	CVSS		
	Base Score	**Temporal Score**	**Environmental Score**
Attack Complexity	Low		
Attack Vector	Network		
User Interaction	None		
Privileges Required	Low		
Confidentiality	High		High
Integrity	None		
Availability	None		
Scope Change	Unchanged		
CVSS Score	**High**		

The data output by COBRA can be used in metrics to reveal trends and patterns in cyber testing results. Heatmaps convey results and inform risk- based decisions. Figure 3 illustrates high impact, low complexity issues in the red outlined box that should receive priority for triage efforts. In figure 4, the quad chart presents actionable intel by plotting findings according to their complexity (x-axis) and severity (y-axis). In figure 3, findings where the attack vectors are trivial with severe consequences appear in the upper right quadrant, while findings that are low-impact with complex attack vectors gather in the bottom left quadrant. These heatmap examples become powerful tools when communicating the results of a testing engagement; they provide discussion points for remediation planning. Calculating the average of the attack complexity (figure 5) and the average severity (figure 6), generates a high-level view of the cyber security posture from a cyber exploitation and testing standpoint.

Using the combination dial gauge and heatmap visual, it is possible to state that this example environment falls into a trivial range for the complexity in regards to the knowledge and level of effort to attack and exploit it, resulting in the average severity rating of medium

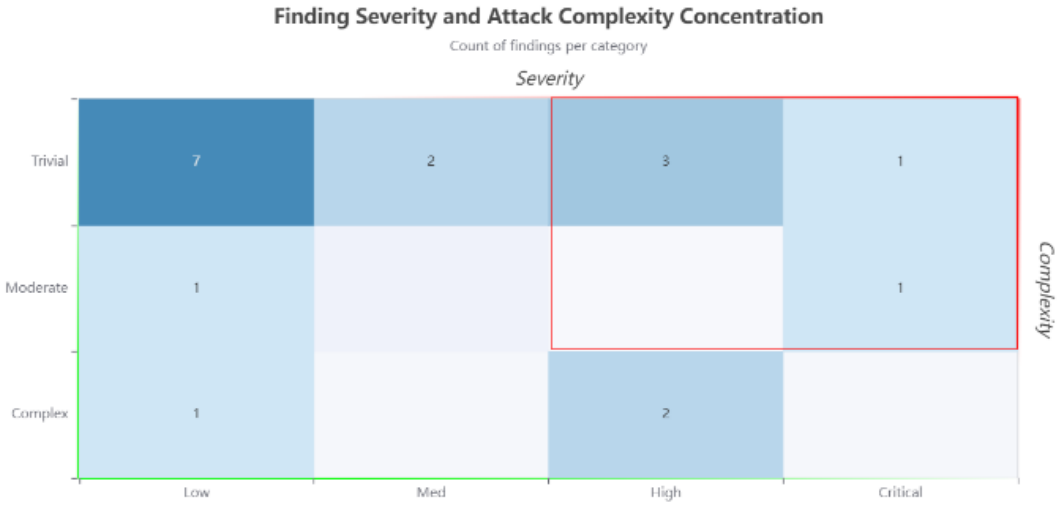

Figure 3: Heatmap

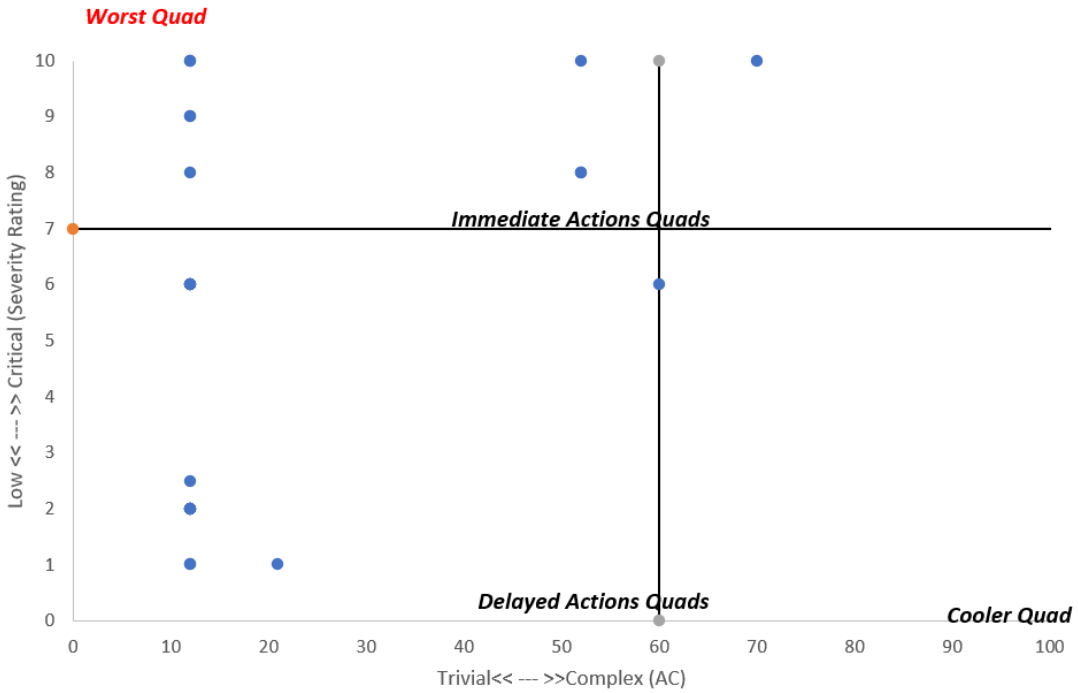

Figure 4: Heatmap Quad Chart

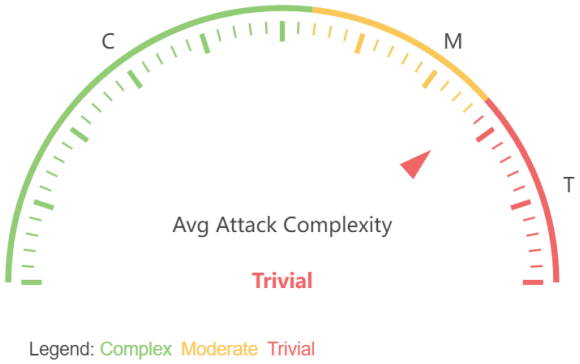

Figure 5: Average Attack Complexity

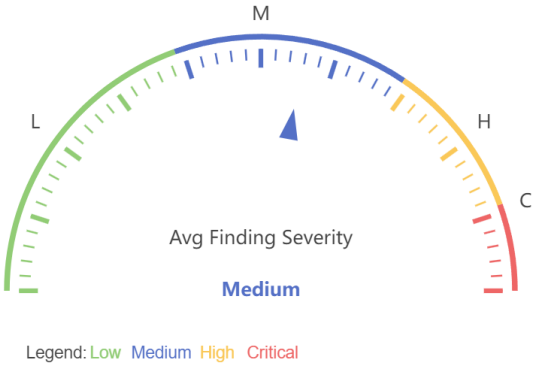

Figure 6: Average Finding Severity

7. Conclusion

The LM Red Team identified a problem where ambiguous ratings assigned to cyber testing findings were negatively impacting the prioritization of remediation efforts. Further, the subjective nature by which findings were rated limited the substantiality of a given rating. To address this problem, the team leveraged existing severity rating frameworks as a foundation to develop COBRA. Through research, practical testing, and analysis, the team has developed a framework which appropriately accounts for context, minimizes subjectivity, and is streamlined to reduce tester overhead.

CVSS was used as a base, and modifications were implemented to better aligns with the type of testing LM RT conducts and account for data types encountered in the defense industrial base. A

succinct set of questions for analysts was created to quickly ascertain the overall context of discoveries, allowing for automatic scoring (testers typically are not fans of documentation).

The team created custom mathematical equations that were adjusted during research (and are still being slightly refined as more is learned) that provide distinct levels of severity for each finding by accounting for impact, complexity, and the overall environment/system under test. Heatmaps were created by plotting the understood findings into a quadrant graph (figure3) visually depicting the more concerning discoveries based on testing. Along with the overall severity, average severity across all findings is depicted on a heatmap, allowing for the comparison of one assessed environment against another.

In the future, the team plans to explore scoring how well an environment performed from a resiliency perspective. As testing is a point in time activity, how an environment fared during the test could be compared year over year during re-testing periods and also opens possibilities to allow resiliency comparisons across domains tested.

The team expects enhancements to COBRA over time with some future work considering chaining of findings along a vector path. For example, a vector of attack with chained findings might each be rated differently but should also account for the initial vector of attack in the calculation. While this is partially accounted for in the existing iteration of COBRA, there is significant room for enhancement. Another area of future work surrounds the differences in testing overtly, compared to covertly. Assumptions are commonly leveraged in this situation, placing the testing in a specific point of presence, changing the overall initial vector. This allows for an unacceptable level of subjectivity and variance. Yet another planned future area will address physical penetration testing or close access testing (CAT), to determine severity ratings based on physical access obtained scenarios.

8. References

[1] NIST. National Vulnerable Database. Common Vulnerability Scoring System. https://nvd.nist.gov/vuln-metrics/cvss

[2] Common Vulnerability Scoring System v3.0: Specification Document. https://www.first.org/cvss/v3.0/specification-document

[3]https://www.cvedetails.com/cve-help.php

[4] Defense Counterintelligence And Security Agency. “Controlled Unclassified Information (CUI)”.https://www.nist.gov/publications/common-vulnerability-scoring-system

[5] Mell, P., Romanosky, S., Scarfone, K. “Common Vulnerability Scoring System”. IEEE Computer Society, Security & Privacy 1540-7993/06 pg. 85-89

[6] Defense Counterintelligence and Security Agency (DCSA). “Controlled Unclassified Information”. 2022. https://www.dcsa.mil/mc/ctp/cui/

[7]https://www.cvedetails.com/vulnerability-list/opsqli-1/sql-injection.html

[8] Scarfone, K. “The Common Configuration Scoring System (CCSS): Metrics for Software Security Configuration Vulnerabilities”. NIST. December 27, 2010. https://www.nist.gov/publications/common-configuration-scoring-system-ccss-metrics-software-security-configuration

[9] Common Weakness Enumeration (CWE). ” 2021 CWE Top 25 Most Dangerous Software Weaknesses” 2021. https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html

[10] Common Vulnerability Exposure (CVE). “Frequently Asked Questions (FAQ)”. MITRE. 2021. https://www.cve.org/ResourcesSupport/FAQs

[11] DOD INSTRUCTION 5200.48. “CONTROLLED UNCLASSIFIED INFORMATION (CUI)” . March 6, 2020. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/520048p.PDF

[12] Information Security Oversight Office, NARA. “Part 2002 – Controlled Unclassified Information (CUI). Pg. 527-547. https://www.govinfo.gov/content/pkg/CFR-2017-title32-vol6/pdf/CFR-2017-title32-vol6-part2002.pdf