DOE CyberForce Competition Anomaly Challenges

[redacted] – CCI Summer 2020
Strategic Security Sciences (S^3) Division

Through the course of the summer, work was put towards preparing anomaly challenges for the 2020 DOE CyberForce Competition. Originally, this was to be an in-person event, with university teams taking the stage against each other. Due to the challenge of the Covid-19 pandemic, however, the competition is now set to take place virtually and in individual teams. Nonetheless, the main project goals stayed largely the same. Anomalies for the competition were developed and mapped according to the seven categories of the National Institute of Cybersecurity Education (NICE) Framework, further categorized with Knowledge, Skill and Ability (KSA) alignments. Anomaly challenges utilized resources including Kali Linux, an advanced penetration testing platform that provided many cybersecurity tools, Wireshark, and others. With the guidance of the NICE Framework, anomaly challenges covered a broad range of areas within cybersecurity. These areas include reverse engineering, steganography, password cracking, SQL injections, packet analysis, and error log analysis. Anomalies were ranked according to their level of difficulty, with more points for a harder anomaly and less for an easier one to be added to a competitor's total during the competition in November. Additionally, upon a satisfactory completion of anomaly work, summer research turned towards web development of the sCOARboard website, the server to be used during the competition, during the last few weeks. The app uses the Django Framework, written in Python, as the core structure. Modifications made to the server included an upgrade to Django 3.0, defaulting time zone settings to create consistency, and fixing a submission grading counter issue. The overall work throughout the summer contributed to the development of the 2020 DOE CyberForce Competition.

Introduction

In April 2016, Argonne hosted its first annual Cyber Defense Competition, with Iowa State University coming out victorious. Since then, competitions have been held every year, the most recent national winner being the University of Maryland, Baltimore County (UMBC) in November 2019. This year, although virtual, looks to carry on the recent tradition and mark five years of the CyberForce Competition. The goal of this project was to contribute to the 2020 November virtual competition by creating anomaly challenges for competitors from universities. Challenges covered a wide range of topics, including steganography, reverse engineering, Structured Query Language (SQL) injections, sniffing & spoofing, and error log analysis. In addition, upon the completion of an ample amount of anomaly challenges, during the last few weeks, the summer research shifted towards web development of the sCOARboard website used for the competition. Some details of the competition remain uncertain, as transitioning to a virtual environment is a unique challenge. One major change that looks to be implemented is the elimination of university teams, as individual competitors are now set to take the stage against each other. This was kept in mind both in the designing of the anomalies as well as the overall quantity needed. Nonetheless, the goals of the summer research and work remained largely the same and looked to be met well. Eight anomaly challenges were developed, with some easier and some more challenging. These anomalies were mapped to the National Institute of Standards and Technology's (NIST) National Institute of Cybersecurity Education (NICE) Cybersecurity Workforce Framework, as well as their Knowledge, Skill, and Ability (KSA) alignment(s) [1]. All seven categories of the NICE framework were covered in the anomalies, one of the main goals of the summer project. During the closing weeks of the internship, several development goals for the website were met as well.

Process

In the beginning stages of the project, a considerable amount of time went into learning the necessary tools provided and how they could be used to create anomalies for the competitions. By creating a virtual environment through VirtualBox, the Kali Linux operating system was able to be implemented, and along with it the wide range of tools it provides in many areas of cybersecurity. These tools include, but are not limited to, steganography, password cracking, SQL injection testing, and reverse engineering. Additional software downloaded to assist in anomaly development included Nmap, Wireshark, and MySQL.

After familiarizing with the aforementioned software and environments, work on the first anomaly challenge began. It incorporated steganography and password cracking through Steghide and hashcat, both tools provided by the Kali Linux platform. The anomaly mapped to KSA K0305, emphasizing knowledge in data concealment. The challenge provides the competitor with an image and alludes to the possibility that there is an embedded message inside. It is the competitor's objective to recognize this and use the appropriate software (Steghide) to extract the text from the image. This is only half the battle, however, as the text inside is an MD5 hash of the true message. If the competitor promptly recognizes the 128-bit string as an MD5, they should then use hashcat (or another password cracking tool) to decrypt the message and retrieve the answer to the anomaly challenge. This anomaly proved to be fairly uncomplicated, as it was simply a matter of following the Steghide procedure to embed the text file in the image and finding an appropriate MD5 hash

to embed. The only difficulty in the development of the challenge came in ensuring that the hash was decryptable for the competitor in a reasonable amount of time.

The next few challenges shifted away from the expected software-based challenges, intending to test the competitor on more unique and specific KSA's within the NICE framework. The first of these focuses on KSA K0042 [2]. The National Institute of Standards and Technology (NIST) Cybersecurity Framework provides guidance for organizations to follow in pursuit of mitigating cybersecurity threats. Subcategories are identified in a AA.BB-C format, where AA designates the function, BB the category, and C the subcategory. To base an anomaly challenge around this, two situations were presented to the competitor, and to successfully complete the challenge, they must correctly categorize the situation within the NIST framework. For example, one prompt says: "In response to a phishing attack, a company releases a statement reaffirming the security roles and responsibilities for the employees." This corresponds to ID.AM-6 within the NIST framework, where ID represents Identify, AM represents Asset Management, and 6 represents the 6th subcategory within AM, focusing on establishing cybersecurity roles and responsibilities for the entire workforce. Another anomaly challenged narrowed towards KSA K0052, which focuses on a knowledge of mathematics, including linear algebra, calculus, and statistics. The anomaly for this case focused on linear algebra and supplied the competitor two vectors of one basis and two vectors of another, ultimately asking for the change of basis matrix between the two bases.

Turning back towards more traditional cybersecurity situations, the next anomaly challenge prioritizes principles in reverse engineering. Mapping to KSA S0270 in the Collect and Operate Category for skill in reverse engineering, the challenge provides the competitor with an executable file compiled from code written in the C programming language. The executable simply asks the user for a key to obtain the answer to the anomaly. Upon seeing this, the competitor should recognize there is no avenue for obtaining said key, and instead look for other methods to solve the anomaly. Using radare2, a reverse engineering tool found in Kali Linux, it is possible to decompile the binary code and access the source code of the executable file. First, the user must use the r2 command to load the executable file through radare2. Then, pressing V twice will load a visual view of the functions that will allow the competitor to use arrow keys to browse and search the functions as they please. In this case, searching the main function will reveal two scenarios for user attempts at guessing the key: one where the proper key is entered, one where it isn't.

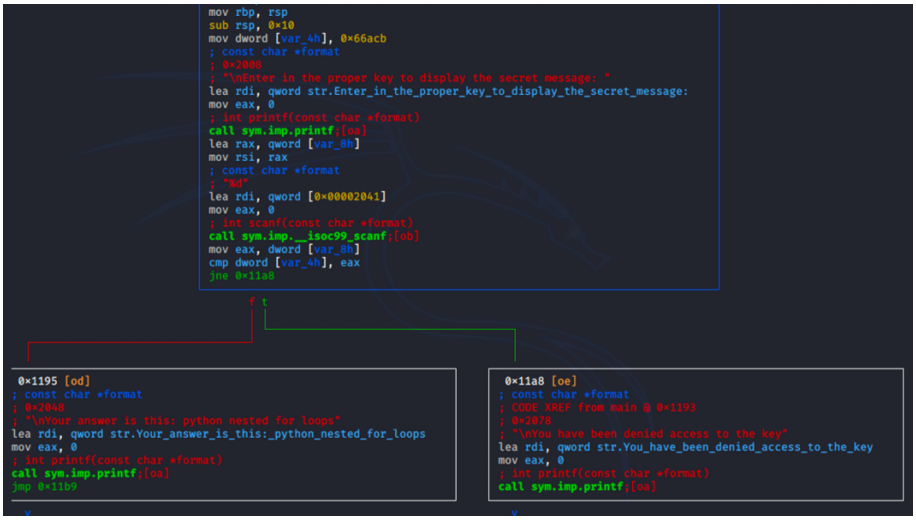

Figure 1
To the left is the result of using the radare2 software to gain access to the main function. On top is the prompt for the key, and on the bottom are two possible results.

The next anomaly challenge was classified as one of the more difficult of the eight created and covers KSA K0023/K0024 within the Securely Provision/Operate and Maintain categories, with the focus of database systems and query languages. The anomaly supplies the competitor with a self-made login environment created with PHP: Hypertext Preprocessor (PHP) , Hypertext Markup Language (HTML) and Structured Query Language (SQL), and it prompts them to login to retrieve the anomaly answer. However, they have no login credentials for the site, nor do they have any way to create them. They must instead perform a Boolean-based SQL injection, manipulating the php code into granting them access to a successful login. Today, there are measures against this style of injection, so the environment created for the challenge was purposefully made vulnerable to allow the competitor the chance to perform the injection.

```
if($_SERVER["REQUEST_METHOD"] == "POST") {
        $uname = $_POST['username'];
        $pass= $_POST['password'];

        $sql="select * from test.users where username = '$uname' and password
            = '$pass'";
        $result=mysqli_query($db,$sql) or die(mysqli_error($db));
        $row=mysqli_fetch_array($result,MYSQLI_ASSOC);
        $active=$row['active'];

        if(mysqli_num_rows($result)!=0) {
                echo "Your answer is this: NEVER_GONNA_GIVE_YOU_UP";
                exit();
        }
        else {
            error_log("Failed attempt. Username:  ");
            error_log($uname);
            error_log(" Password: ");
            error_log($pass);
            echo "Not quite - try again!";
            exit();
        }
}
```

Figure 2
On the left is a code snippet that shows the program reading the user input (white arrow). The three lines below the white line process the true-false statement that checks if valid strings were entered for username or password. The orange arrow points to a line that evaluates a course of action based off the results of the login attempt.

Making the login environment vulnerable to the SQL injection was particularly challenging, as it required an in-depth knowledge of what each line of the php code did, and protections around certain functions. For example, the SQL query line, pointed to by the white arrow in Figure 2, originally used a function to read user input in a protected manner such that it did not directly impact the code. This functionality was eventually recognized and removed from the code, with the username and password being read simply by grabbing the string and inserting it into the code. In addition to this, the line pointed to by the orange arrow had also originally prevented vulnerability. Initially, the latter part of the line read ($result) == 1, meaning, if there was exactly one successful username and password match from the user input and the database, then access would be granted. For this type of SQL injection, however, every comparison to the database with the injection input will return a match, since the hack is Boolean based. Making the change to what is shown in Figure 2 still allowed for normal login, but also opened the gateway for a SQL injection, the intended goal of the anomaly challenge.

Staying on a similar track, the next anomaly covers KSA K0024, pertaining to knowledge of query languages. In addition, this challenge includes error log analysis by asking the competitor to analyze a log file of a SQL injection attack. The task of the competitor is to correctly identify the type of SQL injection that corresponds to the activity in the log file.

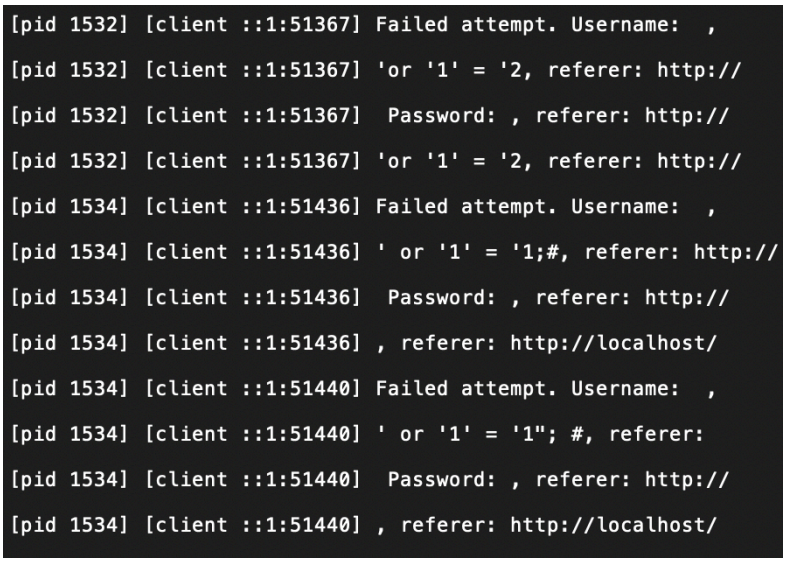

Figure 3
An excerpt of the log analysis to be provided to the competitor for one of the anomalies during the CyberForce competition. The 'or '1' = '1 related syntax should be a clear clue to the competitor that the user of the website is attempting to enter a malicious string into the username and password textboxes to use a Boolean-based SQL injection.

Finally, the last two anomaly challenges were designed around packet capture and analysis using Wireshark. The first of these challenges covered KSA K0058 within the Analyze category, pertaining to knowledge of network traffic analysis methods. The anomaly puts the competitor in a scenario: they are working for a company, and, during a break, they investigate their team's network activity to make sure they are staying on task. The competitor is told to identify a website that is consistently used and take the link as the answer. It was created using Python within a Chrome browser to open the same website many times within a small window of time. This created a noticeable enough of a trend for the competitor, upon using Wireshark and applying a DNS filter, to obtain the answer to the challenge. The second packet-related challenge, narrowing on KSA K0116 within Protect and Defend, is a bit more detailed. It again puts the competitor in a scenario, this time where they suspect that two co-workers are plotting against them and using a random website as their secret source of communication. The competitor, to solve the anomaly, must load the network traffic capture history provided by the packet capture (pcap) file. They can use the DNS filter to investigate the website they are using, at that point recognizing that the only avenue of user input on the site is through the username and password textboxes. Since the website is insecure (http rather than https) the competitor will be able to find the username and password attempt history in the pcap file, upon which they will find the secret messages and the answer to the anomaly challenge.

Upon completion of the main summer research in the development of anomaly challenges, the last few weeks of research mainly consisted of web development of the sCOARboard website used during the competition. Using the Django framework for Python web development, several issues were worked to be completed during this time. Contributions made include identifying backwards incompatibility problems between Django 2.2 and 3.0, assisting in the upgrade to 3.0 after the issues were resolved, fixing an issue with a submission counter, and updating time zone functions to default to the Django settings rather than the settings of the computer to ensure consistency.

Future Work

As mentioned earlier, this year's CyberForce competition in November will take place virtually due to the Covid-19 pandemic. Consequentially, the competition has shifted from teams to individual competitors. While plans for the future certainly aren't concrete, change appears to be in store for years to come, depending on how certain aspects of the competition work in a virtual environment. Anomaly challenges may even be created for virtual and in person competitions in future years, as the possibility of hosting both formats of the competition annually is certainly not out of question, especially considering that experience with both settings will be obtained.

Impact on Laboratory or National Missions

Part of Argonne's vision and the vision of other Department of Energy (DOE) laboratories is providing resources and opportunities for developing skills in the next generation of professionals. Specifically, the CyberForce competition provides a platform for university students to showcase and test their cybersecurity capabilities in a competitive setting. The anomaly challenges allow for the competitors to familiarize themselves with situations that may be related to real world problems, as well as acquire additional knowledge in line with the NICE Framework. Additionally, the progress made on the sCOARboard website looks to refine the flow of the virtual competition and adjust towards this year's change in setting. Cybersecurity is a field that is rapidly accelerating in importance, and as the United States strives towards meeting cybersecurity demands, this year's competition looks to further inspire a new inspiration of professionals. This year's competition is funded by the Department of Energy's Office of Cybersecurity, Energy Security, and Emergency Response (DOE CESER) and the National Nuclear Security Administration.

Conclusions

Throughout the summer research, work was put in towards developing anomaly challenges for competition, covering categories of the NICE Framework and providing competitors with a broad range of challenges to visit during 2020 DOE CyberForce Competition. In addition, web development efforts went into ensuring the competition, the virtual, provided the best experience possible to participants. This effort aligns with the Argonne and the DOE's mission to provide inspiration and educational platforms for cybersecurity professionals to come. Though the Covid-19 pandemic has caused the usual nature the CyberForce competition to change, the goals remained the same, and the summer research aided in the effort to achieve them.

References

[1] NIST Special Publication (SP) 800-801, *National Initiative for Cybersecurity Education (NICE) Framework*, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-181.pdf

[2] Reference Spreadsheet for NIST Special Publication 800-801, https://www.nist.gov/document/supplementnicespecialtyareasandworkroleksasandtasksxlsx

Participant's Table

Name	Institution	Role
[redacted]	Argonne National Laboratory	CCI Intern – developed anomaly challenges for the CyberForce competition, aided in development of the sCOARboard website
[redacted]	Argonne National Laboratory	Cybersecurity Analyst II – anomaly development leader and intern mentor, managed and tracked summer progress
[redacted]	Argonne National Laboratory	Cybersecurity Analyst II – lead developer of the sCOARboard website, intern mentor

Scientific Facilities

No scientific facilities were incorporated during the summer research.

Notable Outcomes

As of now, there are no notable outcomes to acknowledge.