CPRE 492 Bi-Weekly Report #3

Report Weeks: Feb 22nd - March 1st

Group Number: 16

Project Title: Cloudflare WAF AI

Client/Advisor: Cylosoft; Dr. Yong Guan, Dr. Berk Gulmezoglu

Team Members:

Ryan Burgett: Meeting Leader
Giovanni Mejia: Report Manager
Jordan Heim: Documentation Manager
Eric Reuss: Documentation Manager
Presiian Iskrenov: Meeting Scribe
Benjamin Fedderson: Stakeholder Correspondent

---

***Summary:***
With a shortened period, we have still been working on some of the same tasks as we progress throughout the semester. Ben has delivered another finalized C# version of the console application, and we have confirmed it works with Cylosoft servers. Ryan has been implementing more hard-coded scans to supply for our database. We just want to ensure we account for any scans that can be supplied with our script. Giovanni and Eric have still been working on Analytic Extensions with Visual Studio. They are one step closer to implementing it with the database and testing various log files from there. Presiian and Jordan have been further researching attacks that we can encounter. They have also been implementing mitigation strategies to test with our database. Overall, this has been a successful week as are digging deeper into our finalized product.

***Past weekly Accomplishments:***

Our client requested another version of our Console Application. It was only small changes compared to our previous version, but we did confirm that it works with the Cylosoft servers and processes more efficiently. We are closer to implementing our analytic extensions to our own actual database. We still have been trying Visual Studio extensions on sample data. We want to finalize any testing before we carry it over to our actual database. As research has increased, we have been able to implement mitigation strategies as well so we can have everything finalized once we add our VS extension to our database. This research still includes the types of attacks we may encounter.

***Pending Issues:***
Currently, we are not experiencing any pending issues.

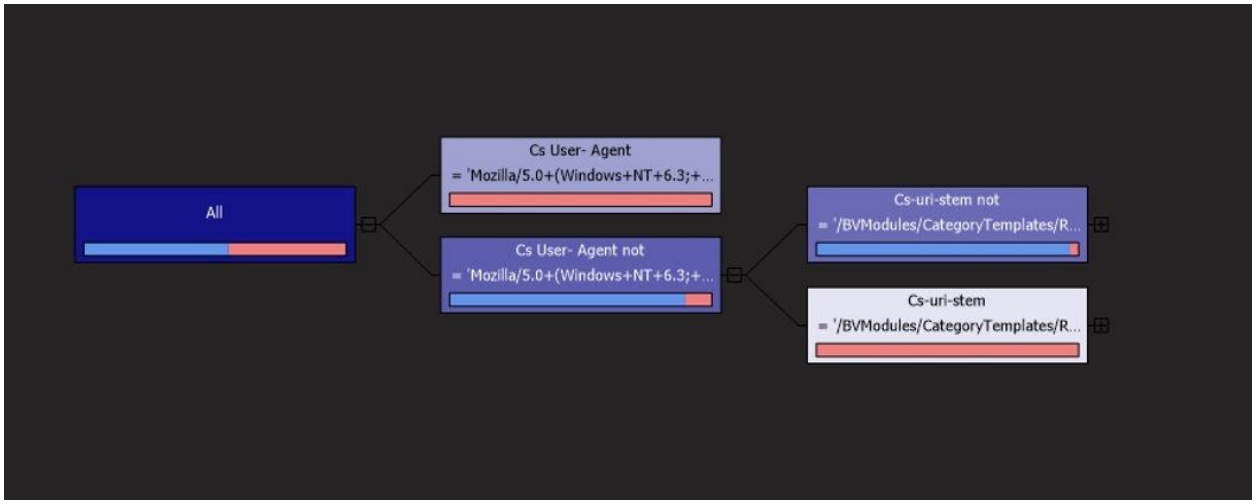
***Individual Contributions:***

| ***Team Member:*** | ***Individual Contributions*** | ***Hours this week*** | ***Hours Cumulative*** |
|---|---|---|---|
| Ryan Burgett | Worked on Database Scanner V2. Added another hard-coded version of SQL injection. | 6 hours | 21 hours |
| Giovanni Mejia | Used analytic extensions from Visual Studio on various sample data testing | 7 hours | 20 hours |
| Jordan Heim | Researched various attack methods. Started implementation with various mitigation strategies. | 6 hours | 18 hours |
| Eric Reuss | Used analytic extensions from Visual Studio to try out on a Sample Database | 7 hours | 18 hours |
| Presiian Iskrenov | Started up an implementation with mitigation strategies | 6 hours | 18 hours |
| Benjamin Fedderson | Worked on the second version of the C# version of Console Application. | 7 hours | 22 hours |

***Plans for the upcoming week:***

Since we are getting closer to our goal of implementing analytical tools within our database, we would like to start finalizing all research with our mitigation and types of attacks. Hopefully, we can keep the console application updated with our current version and not need any more updates. We would also like to meet with our client about his final thoughts in regards to whether we need AI software within our database.

***A screenshot of sample data and VS analytical extensions***




Is diagram shows the various user agents and corresponding OS and internet applications that were used to connect to Cylosoft servers. It all depends on the type of application that was being used to connect to the server and what OS so it can be recorded in our log files.

***Version 2 implementation of SQL injection***

```
#Function returns all IP addresses that are suspected of SQL Injection on a specific date
def detect_sql_injection(date):
    parameters = ["1=1", "or=", "DROP TABLE", "#", "CONCAT", "UNION", "Hex", "HAVING", "INSERT"]
    output = []
    for substring in parameters:
        query = " Select [c-ip], [CF-Connecting-IP] FROM LogTable WHERE [cs-uri-stem] LIKE '%" + substring + "%' AND [date] = '"
        cursor.execute(query)
        for row in cursor:
            if(row[1] != None):
                if(row[1] not in output):
                    output.append(row[1])
            else:
                if(row[0] != None and row[0] not in output):
                    output.append(row[0])
    return output
```

Our way of thinking was researching the usual SQL injections that hackers may try. The parameters we have above are some various things that were learned in previous classes.