Automated Web Proxy-Data Leakage Analysis

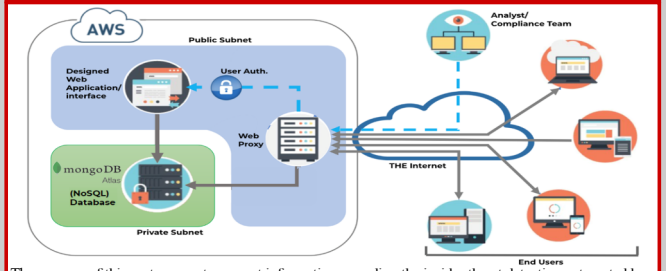
Akshith R. Bandam, Justin Luu, Kelvin K. Nguyen, Malay N. Patel, Harsimran K. Ruprai

Department of Cybersecurity Engineering, George Mason University

Background

As technology continues to progress rapidly, threat actors within small businesses, known as "insider threats", are becoming a growing problem for small businesses. According to the Insider Threat Report by Haystax, 60% of the small businesses have encountered one or more attacks from insiders within the past twelve months [1]. According to the Ponemon Institute's research, small businesses (less than 500 individuals) spent an average of $7.68 million combined for mitigating insider-threat related incidents [2]. We believe that many small businesses struggle to implement an insider threat program. Management requires data and statistics to validate that such a program is valuable before approving funding for such a program. Therefore, small businesses need a tool to effectively detect potential insider threats. Verizon assigned our team to develop an automated web proxy analysis tool that would collect and analyze potential insider threats from the web proxy logs.

Concept of Operations (CONOPS)



The purpose of this system was to present information regarding the insider threat detection automated by the tool. The automated web proxy analyzer is in the form of a web application and is hosted on the virtual web server. Analysts can access the tool using a web browser, as shown in the diagram above, these analysts must provide the proper credentials prior to being able to use the tool. The analysts are provided with several prepacked rules to filter the log data; the prepackaged rules contain a list of specific websites that was defined by our own criteria. Once the analyst begins the search, the prepackaged rules will be applied to the filter and sent to the backend to retrieve the correct web proxy log dataset. The filtered log entries will be returned and populated on the table.

Generating Threat Criteria

Metrics for Insider Threat

ID	Rank	Filter	Type	Allowed Occurance per Week	Pts Assigned per Occurance
a	1	After Work Hours	Time (GMT)	1	100
b	1	Foreign Sites	URL	1	100
c	3	Pirating Sites	URL	2	50
d	4	X-Rated Sites	URL	3	33.3
e	4	File Sharing Sites	URL/Byte > 1GB	3	33.3
f	6	Competitor Sites	URL	30	3.3
g	6	Social Media Site	URL	30	3.3
h	8	Streaming Site	URL	40	2.5

Not all businesses have the same policies that define the same insider threat criteria. Due to this variety, it is difficult to have a single standard that describes insider threat activities for all businesses. Therefore, our team felt it was necessary to cover both malicious and non-malicious users when defining the insider threat criteria. We looked at users visiting websites in the following insider threat categories: file sharing, pornography, foreign interaction, competitors, social media, and pirating websites. Any user visiting websites falling under our insider threat categories accumulates risk points. A risk point scale was used as a metric to determine which users are malicious or insiders. The table above displays different weights of each insider threat category used to calculate the risk score of each host. Each category was ranked from 1 to 8, where 1 represented the most compelling indicator of an insider threat and 8 represented the least compelling indicator of insider threat. There are four assigned risk levels: low (0-49 points), medium (50-99 points), high (100-199 points), and severe (200+ points). The users also accumulate points if they work during after-work hours; we defined the after-work hours criteria to be between 6 pm to 6 am EST.

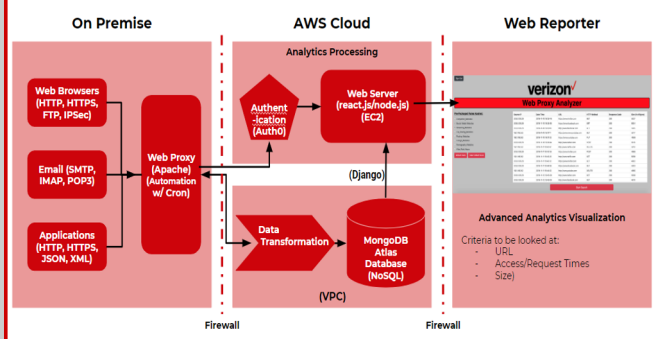
Data Generation Process

Insider Threats IP Breakdown

IP Address	URL Category	Risk Pts	IP Address	URL Category	Risk Pts	IP Address	URL Category	Risk Pts
192.168.24.8	Pirating Site, 5x Streaming Site	62.5	172.16.2.222	2x File Sharing Sites, 21x Competitor Sites, 10x Social Media Sites	168.9	10.1.7.2	X-Rated Site	33.3
192.168.24.23	10x Competitor Sites, 9x Social Media Sites	62.7	172.30.5.9		0	10.155.155.10	Streaming Site	2.5
192.168.67.117	5x Social Media Sites	16.5	172.25.33.125	2x File Sharing Sites, Competitor Site, Social Media Site, Streaming Site	75.7	10.248.10.7		0
192.168.50.60	Social Media Site	3.3	172.17.1.174	Social Media Site	3.3	10.100.76.212	Foreign Site, 2x Social Media Sites	106.6
192.168.183.84		0	172.22.172.49	2x X-Rated Sites	66.6	10.200.60.30		0
192.168.205.71		0	172.16.253.13		0	10.20.222.99	2x Streaming Sites, 21x Social Media Sites	74.3
192.168.111.11	Foreign Site, File Sharing Site, 10x Social Media Sites, 5x Streaming Sites	178.8	172.19.78.85	30x Streaming Sites, Social Media Site	78.3	10.174.84.185	Competitor Site	3.3
192.168.104.31	20x Social Media Sites, 30x Streaming Sites	141	172.24.88.22		0	10.92.234.148		0
192.168.212.19	File Sharing Site, Competitor Site	36.6	172.30.81.163	13x Social Media Sites	42.9	10.10.10.110	After Work Hours, Pirating Site, X-Rated Site, File Sharing Site, 2x Social Media Sites	223.2
192.168.156.2		0	172.30.83.55	After Work Hours	100	10.47.85.68	7x Streaming Sites	17.5

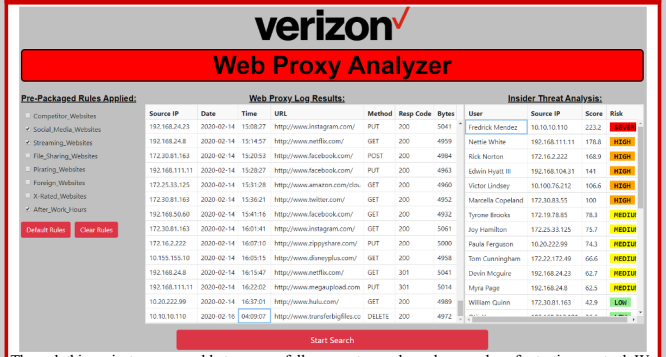
A custom set of criteria was developed and used as a foundation to build our own dataset of web proxy logs. This dataset was primarily generated to verify and test the accuracy of our automated tool for detecting insider threat activities. To generate a realistic web proxy log, we injected URLs that came from our predefined list of insider threat URLs into the web proxy logs of the non-malicious users. This mixture of URLs (benign/malicious) also plays a significant role in testing the accuracy of our tool. Our dataset contains 30 unique IP addresses to represent employees of a small business. Following this approach, the team crafted multiple personalities/intents for each employee and predetermined which employees were malicious or non-malicious as shown in the table figure above. Using a python script, a total of seven web proxy log files were created that represented each day of the week.

Implementation



In the figure above, the web application was divided into three sections. The first section of our web application is the web proxy server. This server was configured on an AWS Ubuntu EC2 instance and it was implemented to emulate a real-world proxy server. The proxy server consists of three scripts: a perl script which parses the log data into CSV format, a python script that generates the risk score for each distinct IP in the log file and outputs the result to a CSV file, and a bash script which automatically executes the perl and python script to export the generated CSV files into the database. The second section of our web application is the database server which runs on MongoDB Cloud Atlas. The database transforms the imported CSV log file and calculated scores obtained from the web proxy server into json format. The third section of our web application is the web server that hosts our web application/GUI on an AWS EC2 instance. The GUI was developed using the Node.js and React.js frameworks for the frontend. Express.js was installed and configured to allow our tool to make API and http calls. To prevent unauthorized individuals from accessing the web proxy tool, an authentication page was implemented using Auth0. Analysts must enter their credentials before being able to access the tool.

Results



Through this project, we were able to successfully generate apache web-proxy logs for testing our tool. We also completed our web application in the form of a GUI. The filtering mechanism of our tool resulted in an accurate filtered output of expected returned data. For example, in the figure above, we tested the filter by applying two predefined rules ("Social Media Websites" and "Streaming Websites") using the checkboxes on the left side of the web application which returns the filtered log data from the database. The next result displays our risk assessment of each user. On the right side of the figure above, the table displays all the distinct users with their associated IP address, risk score, and risk level. After completing the search, the table displayed 30 total distinct users as expected, along with the correct risk score and risk level associated with each user. In the results, we see that 19 users have "low" risk, 5 users have "medium" risk, 5 users have "high" risk, and 1 user with "severe" risk. The user with a "severe" risk level is a man named "Fredrick Mendez". Upon further investigation, Fredrick appears to have visited a website called "www.transferbigfiles.com" within the "after-work" hour range (6 pm - 6 am), with a timestamp of 4:09 AM on a Sunday morning. Fredrick is one of many examples of users or employees within a small business that conduct unusual behavior and should be monitored more closely to determine whether he is a potential insider or not.

Conclusion

With improved insider threat detection, an important long-term investment can be justified for many small businesses and organizations. Through various stages of research and careful implementation, the custom-built web proxy analyzer was delivered as a web application. This web application was able to automate the collection of web proxy log data and retrieve filtered data from our proxy analyzer web application for ease of analysis. Our web proxy analyzer, hosted on the cloud, does not require additional setup/installation. Our tool can be utilized at very little cost because the components within the tool are mainly open source; however, the tool does require periodic maintenance. Our tool is scalable since we used a cloud-based approach and users can increase the volume of data ingested, if necessary. Compared to the manual collection and analysis approach, our designed web proxy analyzer has many potential benefits to offer to small businesses in terms of affordability, reliability, and increased value in security through improved detection of insider threats. In the future, new technologies such as supervised and unsupervised machine learning algorithms could be incorporated into our tool with possible improvement in the accuracy of detecting insider threats.

Acknowledgements

We would like to thank Dr. Powell, David Grady, and Anastasia Hansen for their continued support throughout our project.

References

[1] The "Insider Threat Report," HayStax, Cybersecurity Insiders, 2019.

[2] J Ponemon Institute, "2020 Cost of Insider Threats Global Report," 2020.I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] Verizon. (n.d.). *Corporate Social Responsibility (CSR) | About Verizon*. Retrieved March 19, 2020, from https://www.verizon.com/about/responsibilityR. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.