Automated Web Attack Detection Using Machine Learning Techniques

Rohini Hanchate[1], Kshitij Thakre[2], Rameshwari Khamkar[3],Priti Jadhav[4],Aditya Kotkar[5]

[1, 2, 3, 4, 5] Dr.D.Y.Patil Institute of Engineering And Technology,Ambi,Talegaon

Abstract- *The maintenance of web server security, availability, integrity and confidentiality has never been such an overbearing task as is today. With threats coming from hardware failures, software flaws, tentative probing and worst of all malicious attacks. Analysing the server logs to detect suspicious activities is regarded to as key form of defence. However, the sheer size of server logs makes human log analysis challenging. Additionally, the traditional intrusion detection systems rely on methods based on pattern-matching techniques which cannot developers cannot maintain based on the high rates at which new and never seen before attack techniques are launched each and every day.*

The aim of this project is to develop an intelligent log based intrusion detection system that can detect known and unknown intrusions automatically. Under a data mining framework, the intrusion detection system is trained with unsupervised learning algorithms specifically the k-means algorithm and the One Class SVM (Support Vector Machine) algorithm. The development of the system was time constrained and limited to machine generated logs due to lack of real access_log files. However, the system's development went smoothly and proved to be up to 85% accurate in detecting anomalous log patterns within the test logs. However, much work still needs to be done to make this a better Intrusion Detection System (IDS) in that a real-time analysis module could be incorporated and a function to retrieve logs from remote location.

I. INTRODUCTION

In an ever advancing and fast developing field, technology has become cheaper, easier to develop, and deploy. Unfortunately, this has also made probing, and attacking servers cheaper and easier to do. It is therefore vital to ensure that web servers are alert and hence secure against any form of attack.

Server logs have been used to confront failure either hardware or software, record notices, warnings and errors to ensure that system administrators can recover or at least know the cause the event of a system failure. Log recording has also acted as a form of defence against human attacks where predetermined techniques such as SQL injections can be easily identified.

On an average web server receiving traffic of at least 1000 unique visits a day generates a huge log that cannot be analysed manually. Now take the same web server and place it in a large company receiving over 10000 unique visits a day. The sheer size of the log file would turn down most system administrators as it is practically impossible to inspect as a human. Most log based intrusion detection systems on the market are pattern-matching technique based, that is, they compare the log entries to a set of predefined patterns that had been manually updated by security experts. Though this approach is effective in determining attacks of known patterns, the hassle is that for each new attack the system is defenceless and it takes the security experts a lot of time and effort to update the new patterns to the intrusion detection system.

From this point of view, current intrusion detection systems are far from intelligent in that they exclusively rely on human intervention to operate effectively. Therefore, more advanced intrusion detection systems are highly desirable. These systems should be capable of detecting known and unknown intrusions intelligently and automatically, distinguishing normal network activities from those abnormal and possibly malicious ones without or with minimum human intervention.

II. MOTIVATION

Recently, some researchers and programmers utilizing data mining algorithms applied to log based intrusion detection systems came up with an effective anomaly detection based intrusion detection system that relied on nothing more other than the inflowing stream of logs to determine what is normal and what is not (possibly an attack). Those algorithms are based on supervised learning. That is to say, they are trained, other than being explicitly programmed, on data sets with labels indicating whether the instances are pre-classified as attacks or not. However, the techniques seemed cumbersome as manually labelling the large volumes of server data mostly over 1GB log files was expensive and difficult. This is what aspired the approach of unsupervised machine learning where no labels are pre-set hence the system is left to determine what is an attack and what is not.

With no requirement for class labels, unsupervised learning algorithms seemed to solve this problem. A broad explanation of approach to intrusion detection systems is that when an intrusion detection system becomes “familiar” with the data through the unsupervised learning algorithms, it is likely to detect “abnormal” data when they come in. Many of which are malicious.

III. LITERATURE REVIEW

BACKGROUND

Designing an intelligent log based intrusion detection system involves with a broad range of knowledge, namely web server security, data mining techniques, learning algorithms and some novelty detection approaches.

In this chapter, initially an introduction will point out the weight of the web server security problems. Afterwards, some conventional intrusion detection methods are briefly discussed before data mining based approaches in the log analysis are introduced. In the next section the topic of novelty detection is covered, which links closely to the detection of intruder intrusions. Finally, some related work will be re-viewed.

INTRUSION

Threats to web servers come typically from the malfunction of hardware or software, or through malicious behaviour by users of software. Promptly resolving incidents is vital, considering the huge costs of data loss and server down-time.

The abundance of computational resources makes lives of computer hackers easier. Without much effort, they can acquire detailed descriptions of system vulnerabilities and exploits to initiate attacks accordingly. According to statistics from CERT® Coordination Centre (CERT/CC), the most influential reporting centre for internet security problems, show that there was a dramatic increase of reported network incidents to CERT/CC (Ma, 2003).

LOGS

To protect servers from attacks, a common approach is to record server logs to monitor all those prominent activities. Each time a noticeable event happens in the server, an entry will be appended to a log file, in the form of plain text or binary format. Take web log files as an example. Every “hit” to a web site, including requests for HTML pages as well as images, is logged as one line of text in a log file. This records information about who is visiting, where they are from and what they are doing with the web server. Below is a sample of an apache log format,

"%h %u %t\"%r\" %>s %b\"%{Referrer}i\" \"%{User-Agent}i\"”

This translates to: -

%h –ip address
%u – Authenticated userID if http authenticated
%t – timestamp [day/month/year: hour: minute: second zone]
%r – request line (method_usedrequsted_resource protocol)
%>s– status code
%b size of returned obj
\"%{Referrer}i\” – http header referrer
\"%{User-Agent}i\"”– the user agent

Implementation: Regular expressions will be used to search through the requested URLs for known attack patterns such as;
XSS : Cross-Site Scripting

SQLI	: SQL Injection
CSRF	: Cross-Site Request Forgery
DOS	: Denial of Service
DT	: Directory Traversal
SPAM	: Spam
ID	: Information Disclosure
RFE	: Remote File Execution
LFI	: Local File Inclusion

This will be achieved through modified code forked from Scalp (a python library).

IV. METHODOLOGY

- Using the log analyzer that has powerfully trained and chosen classifier models capable to detect Web Application attacks, categorize and find the attack vector used into Cross-Site Scripting attack, SQL Injection, Path Traversal attack, LDAP Injection, XPath Injection, OS Command Injection, SSI, CRLF Injection, and to other Anomalous categories.
- The Intrusion Detection System is tailored using the Flask web framework, with the machine learning models at the core of the application, providing the following functionalities.
  - Real-time Intrusion Detection System: Set up over a Web Server it actively monitors the Web Application against Cyber Attacks.
  - Static Log Analysis: Can detect security attacks conducted on an Organization in past.
  - Web Application Security Scanner: Automated scanning of Web Application to detect any possible attacks or vulnerabilities.

– Attack Query Tester: Capable of figuring out the category of attack used by a user-supplied string

4.4.3 Machine Learning and Outlier Detection module

Input: Two Inputs

1. Vectorised logs to be checked
2. Vectorised logs to be trained with (Normal dataset)

Output: Detection Results

Function: Two major functions:

- Using the Normal Dataset to learn
- Detect outlier activities based on the Normal Dataset

Implementation: Two unsupervised learning algorithms are tried.

1. Kmeans
2. One Class Support Vector Machine (One class SVM)

Screenshots

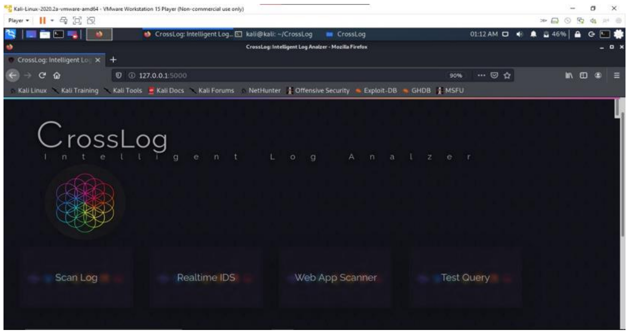


Fig-Front-end

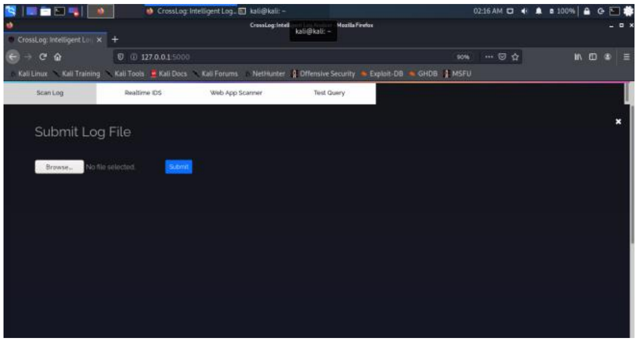
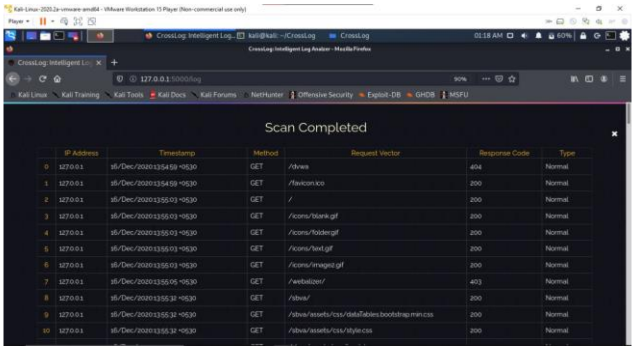


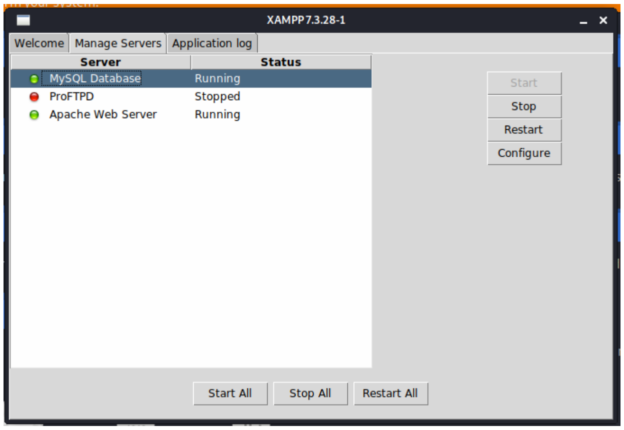
Fig-Scan log file



Scan Completed

	IP Address	Timestamp	Method	Request Vector	Response Code	Type
0	127.0.0.1	16/Dec/2020:13:54:59 +0530	GET	/dvwa	404	Normal
1	127.0.0.1	16/Dec/2020:13:54:59 +0530	GET	/favicon.ico	200	Normal
2	127.0.0.1	16/Dec/2020:13:55:03 +0530	GET	/	200	Normal
3	127.0.0.1	16/Dec/2020:13:55:03 +0530	GET	/icons/blank.gif	200	Normal
4	127.0.0.1	16/Dec/2020:13:55:03 +0530	GET	/icons/folder.gif	200	Normal
5	127.0.0.1	16/Dec/2020:13:55:03 +0530	GET	/icons/text.gif	200	Normal
6	127.0.0.1	16/Dec/2020:13:55:03 +0530	GET	/icons/image2.gif	200	Normal
7	127.0.0.1	16/Dec/2020:13:55:05 +0530	GET	/webalizer/	403	Normal
8	127.0.0.1	16/Dec/2020:13:55:32 +0530	GET	/sbva/	200	Normal
9	127.0.0.1	16/Dec/2020:13:55:32 +0530	GET	/sbva/assets/css/dataTables.bootstrap.min.css	200	Normal
10	127.0.0.1	16/Dec/2020:13:55:32 +0530	GET	/sbva/assets/css/style.css	200	Normal

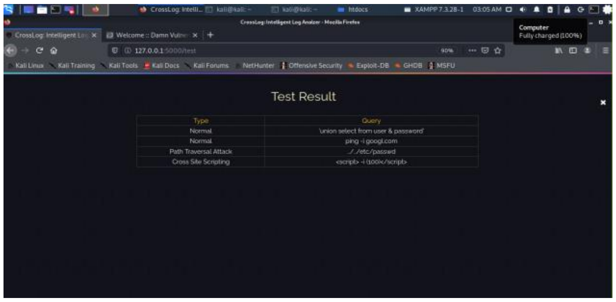
Fig-Log file scan



Server	Status
MySQL Database	Running
ProFTPD	Stopped
Apache Web Server	Running

Fig-Xamp

Fig-Web application for scanning vulnerabilities



Test Result

Type	Query
Normal	'union select from user & password'
Normal	ping -i googl.com
Path Traversal Attack	../../etc/passwd
Cross Site Scripting	<script> -i (100)</script>

Fig-Real time attack

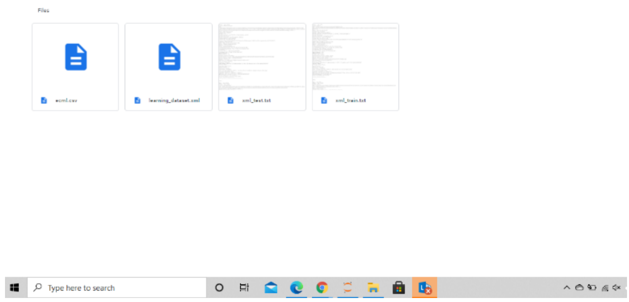


Fig-Dataset

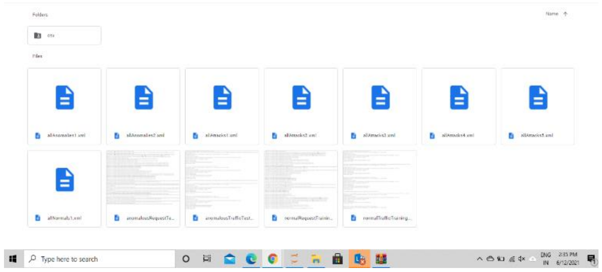
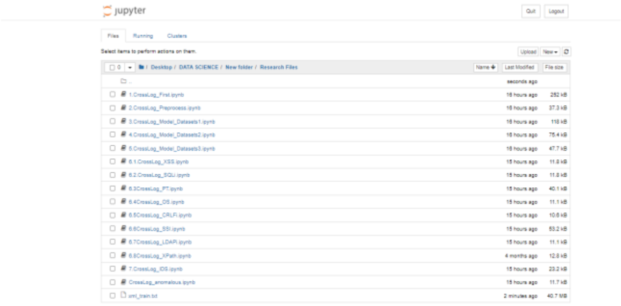


Fig-CSIV

Fig-SVM Graph Analysis



Select items to perform actions on them.

/ Desktop / DATA SCIENCE / New folder / Research Files

Name	Last Modified	File size
..	seconds ago	
1.CrossLog_First.ipynb	16 hours ago	252 kB
2.CrossLog_Preprocess.ipynb	16 hours ago	37.3 kB
3.CrossLog_Model_Datasets1.ipynb	16 hours ago	118 kB
4.CrossLog_Model_Datasets2.ipynb	16 hours ago	75.4 kB
5.CrossLog_Model_Datasets3.ipynb	16 hours ago	47.7 kB
6.1.CrossLog_XSS.ipynb	15 hours ago	11.9 kB
6.2.CrossLog_SQLi.ipynb	15 hours ago	11.8 kB
6.3CrossLog_PT.ipynb	15 hours ago	40.1 kB
6.4CrossLog_OS.ipynb	15 hours ago	11.1 kB
6.5CrossLog_CRLFi.ipynb	15 hours ago	10.6 kB
6.6CrossLog_SSI.ipynb	15 hours ago	53.2 kB
6.7CrossLog_LDAPi.ipynb	15 hours ago	11.1 kB
6.8CrossLog_XPath.ipynb	4 months ago	12.8 kB
7.CrossLog_IDS.ipynb	15 hours ago	23.2 kB
CrossLog_anomalous.ipynb	15 hours ago	11.7 kB
xml_train.txt	2 minutes ago	40.7 MB

Fig-Back-end

inserted in place. The caption is under the figure. All reference to the figure use "Fig." followed by the figure number. Fig. is also used in the caption. J. Tables Tables must occupy a single column, if possible, and must be printed in place. The name is above the table. TABLE I TYPE SIZES FOR PAPERS Type Appearance Font size :16pt Font type: Times new roman Bold, Centred Initial Caps. Font size :12pt Font type: Times new roman Bold, Centred Font size :10pt Font type: Times new roman Font size :10pt Font type: Times new roman All caps 2 size (pts.) Regular Bold Italic 8 9 10 12 16 Table captions,a table superscripts Section titlesa , references, tables, table namesa , first letters in table captionsa , figure captions, footnotes, text subscripts, and superscripts, main text, equations, first letters in section titlesa Authors' affiliations Abstract Authors' names Paper title Subheading Times N. R. aUppercase K. Figures and Tables Place figures and tables at or near the top or bottom of columns where possible. Large figures and tables may span across both columns. Figure captions must be below the figures; table captions must be above the tables. Avoid placing figures and tables before their first mention in the text. Use the abbreviation "Fig. 1," even at the beginning of a sentence. Figure axis labels are often a source of confusion. Try to use words rather than symbols. As an example, write the quantity "Magnetization," or "Magnetization, M," not just "M." Put units in parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization (A/m-1)," not just "A/m." Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)," not "Temperature/K." Multipliers can be especially confusing. Write "Magnetization (kA/m)" or "Magnetization (103 A/m)." Do not write "Magnetization (A/m) x 1000" because the reader would not know whether the top axis label in Fig. 1 meant 15 000 A/m. or 0.015 A/m. Figure labels must be

CONCLUSION

The aims of this projects were to build an IDS that utilized machine learning to detect known and unknown attack patterns in apache logs. In this chapter the achievements, problems and limitations of the system are discussed.

FUTURE WORKS

plans are underway to modify the IntelliIDS into a real time scanner. However, many features need to be incorporated in the system to ensure better accuracy and hence higher detection rates. The features that miss in this version that would make this project even better are:

i. Online Learning

Incorporating the ability to access and analyze remote logs from say a hosted server. This will would increase the productivity of the system in that one system can be used from a stationary location to analyze and report on anomalies of remote systems in real-time.

ii. Real time analysis

This is a feature that is vital to any IDS, this will ensure that the system analyzes logs are they come in other than a post-analysis based approach that the current version works with.

REFERENCES

[1] Alspaugh, S., Chen, B., Lin, J., Ganapathi, A., Hearst, M.,& Katz, R. (2014). Analyzing log analysis: An empirical study of user log mining. In *28th Large Installation System Administration Conference (LISA14)* (pp. 62-77).

[2] Amoli, P. V., Hamalainen, T., David, G., Zolotukhin, M., &Mirzamohammad, M. (2016). Unsupervised Network Intrusion Detection Systems for Zero-Day Fast-Spreading Attacks and Botnets. *JDCTA (International Journal of Digital Content Technology and its Applications, Volume 10 Issue 2*, 1-13.

[3] Coates, A., Lee, H., & Ng, A. Y. (2010). An analysis of single-layer networks in unsupervised feature learning. *Ann Arbor*, *1001*(48109), 2.

[4] Computational Statistics and Predictive Analysis in Machine Learning. (2016). *International Journal Of Science And Research (IJSR)*, *5*(1), 1521-1524. http://dx.doi.org/10.21275/v5i1.nov152818

[5] Gardner, A. B., Krieger, A. M., Vachtsevanos, G., &Litt, B. (2006). One-class novelty detection for seizure analysis from intracranial EEG. *Journal of Machine Learning Research*, *7*(Jun), 1025-1044.

[6] Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised learning. In *The elements of statistical learning* (pp. 485-585). Springer New York.

[7] Kim, Y., Street, W. N., &Menczer, F. (2000, August). Feature selection in unsupervised learning via evolutionary search. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 365-369). ACM.

[8] Li, K. L., Huang, H. K., Tian, S. F., &Xu, W. (2003, November). Improving one-class SVM for anomaly detection. In *Machine Learning and Cybernetics, 2003 International Conference on* (Vol. 5, pp. 3077-3081). IEEE.

[9] Li, W. (2013). Automatic Log Analysis using Machine Learning: Awesome Automatic Log Analysis version 2.0.

[10] Ma, P. (2003). Log Analysis-Based Intrusion Detection via Unsupervised Learning. *Master of Science, School of Informatics, University of Edinburgh*.

[11] Manevitz, L. M., & Yousef, M. (2001). One-class SVMs for document classification. *Journal of Machine Learning Research*, *2*(Dec), 139-154.