L-Protct: Textual Analysis of Web Content to Detect Possible Attempts of Exploitation

Shiju S.J.
Department of Computer Science and Engineering
DMI College of Engineering, Chennai
me@shijuleon.com

***Abstract*—Malware is delivered through various means to the end user. Other than local methods of propagation where the attacker has physical access to one of the connected computers, most malware are delivered through communication mediums. L-Protct is a textual analyzer which runs on top of the medium through which the user retrieves information. The web browser is selected as the program which the user relies upon for information. In this paper, email is selected as the communication medium through which the user is assumed to be targeted. L-Protct introduces a model of using existing data for detecting patterns on attempts of exploitation and learning from data which the user receives. L-Protct can be implemented in organizations to detect emails with content which may lead to the compromise of the host system.**

Keywords-content analysis, natural language processing, security

I. INTRODUCTION

Computers in organizations are attacked for compromising infrastructure and obtaining information by specifically targeting employees. Other than local or remote based system exploits [1], manipulation of employees through emails and other communication mediums is one of the most used methods of targeting [2]. The same can be said for individuals and personal computers. Currently protection from manipulative web pages is provided by certain web browsers by displaying a warning page for certain blacklisted URLs. The protection is limited in its extent since the process of blacklisting malicious URLs is slow and often contains a large number of false positives. To protect users from pages that contain content with manipulative intent, it is necessary for a system to be continuously monitoring the content running from the local system. L-Protct is a content analyzer which continuously monitors web pages as specified by user preferences. Web browser is one of the most important information transfer mediums today. Other than remotely exploiting a system the only way to deliver payloads to the user is through a communication or information medium. Web browser directly acts an information medium and indirectly acts a communication medium through web email clients. To target an individual or an employee of an organization it is plausible for the attacker to deliver the payload through email [3]. The payload can be attached to the email as a file. It is also possible for the attacker to deliver the payload by manipulating the user to perform an action which eventually leads to the delivery of the payload. Modern attacks contain workarounds to spoof URL patterns in the email such that hyperlinks may not be readily visible to the browsers but users could be manipulated to eventually turn it as intended.

II. SYSTEM OVERVIEW

L-Protct consists of varied components which work together to deliver the complete functionality. At the highest level, an extension is installed into the web browser. The extension uses the browser's Document Object Model (DOM) to retrieve the content from the email. In the L-Protct prototype, the extension is designed for the Google Chrome web browser. At the next level, a program which runs on the host PC known as the desktop client is present. After the extension receives the content of the email from web page, it parses the content and sends the content to the desktop client. In the current L-Protct prototype the learning engine is present itself in the desktop client. It is more reasonable to design an API to the learning engine present in a remote server and send a minified representation. It maybe considered a breach to privacy to send the content of the email to the remote server. Therefore, the desktop client should be able to process the content of the email to minify it to a pattern such that sensitive content can be discarded.

A. Browser Extension

Google Chrome provides an application programming interface to the browser through its extensions. A particular feature of the Chrome API allows us to inject a JavaScript file known as content script [4] into the web page. Using the content script it is possible to access the DOM [5] elements and its attributes.

To process the DOM elements and obtain content of the specific attributes, functions are defined in the content script. Based on the domain and web page the corresponding function is executed.

Google Chrome also provides a messaging API [6] which enables to send the DOM elements from the content script to the extension. Along with the content, the hyperlink for downloading the attachment is obtained and sent to the extension's page. The extension has a default page which receives the message and displays to the user. Along with the content of the email, the page also displays whether the email contains an attachment. In the L-Protct prototype, the ability to download the attachment and send to the desktop client are set to manual user control which can be modified to do it automatically without user intervention. The extension sends the attributes obtained from the email as a JSON representation through a HTTP POST request. XMLHttpRequest is used to send the parameters to the desktop client. Alternatively, the data can be send through the Chrome Native Messaging API and continuously reading data from STDOUT of the program through the desktop client.

B. Desktop client

The desktop client is developed in Python with Flask [7] web framework for handling the HTTP POST requests. Email content, the presence of the attachment and the attachment link is sent from the extension to the desktop client which is displayed to the user in the client. From the extension the content is received in JSON which is parsed using the Flask's JSON parser. Due to privacy concerns the content of the email is never transmitted to the learning engine in case it is implemented in a remote server. If the learning engine is implemented in the desktop client then this condition may be relaxed. Depending on the user preference the content of the email can be stored in a database in the host PC. After constructing a representation from the content of the email from which the pattern of the email can be traced, the representation is sent to the remote client for training the learning engine. An option to download the attachment is provided with the L-Protct client. The L-Protct desktop client has an option to scan the attachment with its own interface which uses an online virus scanner API to scan the file with multiple malware scanners at once. By using multiple malware scanners, the probability of detecting if the file is malicious is effectively increased.

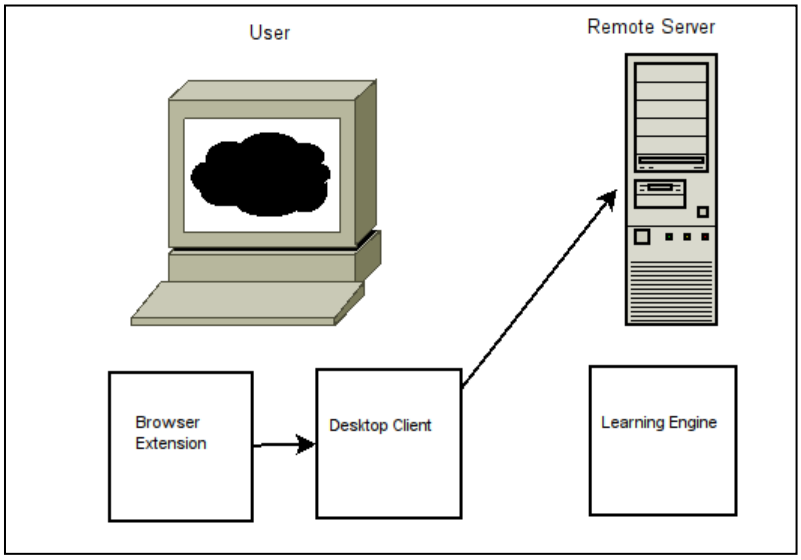

Figure 1. Components of L-Protct

C. Learning engine

Two approaches for the learning engine can be followed. The learning engine can be coupled with the desktop client or the learning engine can be coupled with a web application in a remote web server. Minified content is sent from the desktop client to the remote web server. The current L-Protct prototype contains the learning engine built into the desktop client itself. Data received from the extension is analyzed by using various machine learning techniques implemented in the learning engine. At first the learning model is trained using datasets available on email spam [8] and other like data. After that manually generated text files which contain possible instances of manipulation are used to train the model. Text files created manually are generated by using a vector of words which commonly in combination are found to be manipulative.

Supervised learning [9] is used at the initial stage of the training due to the major boost when compared with an unsupervised learning approach with a vector of common manipulative words. Aiming for the probability of the presence of words in the content of the email results in a large number of false positives. To train the model, CSV files with fields for the text and probability of it being manipulative are generated. Using a Naive Bayesian [10] classifier the probability of a single clause being manipulative is calculated. Every time a group of sentences is fed to the L-Protct it converts the given sentences into vectors of words. Frequency of words and the complexity of the sentence are considered as the major factors in determining the authenticity of the email.

Manipulative emails have the tendency to capture the attention of the user with short catchy phrases which get repeated often in the body of the email. Address of the email along with the subject provide more learning data to the engine. With multiple factors determining the authenticity of the email, a score rating is provided to the email and then displayed to the user.

III. SIGNIFICANCE

L-Protct plays a huge role in protecting users with multilingual cultures or users who don't have the greatest grasp of the system language. They may not be fully aware of the intentions of the email with the limited exposure to the language. By not completely understanding the intentions of the email the employee or user may put the whole system into risk by downloading a malicious attachment or navigating to a harmful page. L-Protct provides an easy interface for the user to learn better about the intentions of the email. Thus by effectively stopping the attempt of manipulation L-Protct adds an extra layer of security to the user.

Operations performed by L-Protct are confined to the user's system and the system is open source. This enables the user to achieve a sense of privacy when compared to closed source systems which perform operations in a remote server. L-Protct continuously improves its rate of detection of manipulative content with the help of the learning engine present in it. Continuous learning enables the protection of user with no regards to the language skills of the user.

Huge organizations can benefit the most from L-Protct by incorporating it with their email clients and web browsers. Depending on the needs of the organization the communication mediums may be designed and developed. Combining a learning approach with a large amount of data can mitigate attacks for employees.

IV. FURTHER WORK

There is a lot of room for improvement in almost all the components of L-Protct. Functionality of the browser extension can be tweaked to allow the user to select websites to run L-Protct instead of the current option to allow it in all websites. Selection of DOM elements to process can be further optimized by considering the type of the page. Using a white list approach of websites based on factors like the rank of the page in reputed page ranking sites can reduce the number of sites L-Protct runs on an average browsing session. Currently a very small number of web email clients are supported by the L-Protct prototype. Desktop client can be improved by using more effective ways to minify text and to derive the pattern. A method of continuously monitoring the downloaded file for a given time even after resulting negative in the scanner can prevent zero days. Monitoring the file is done by tracking the changes the program makes to the computer in regards of file system. Network activity is also a major indicator of a file's trustworthiness. Learning engine of L-Protct can be improved by using better performing machine learning algorithms. Using new techniques in Natural Language Processing, specifically Sentiment Analysis [11], the content of the email can analysed with more precision.

V. CONCLUSION

L-Protct aims at protecting the end user and the infrastructure of an organization by continuously analyzing the content the user is subjected to. Rule based engines have a threshold of how much they can detect malicious content due to the amount of predefined rules. They also require continuous update by the developer. The amount of explicit programming done to detect the content being manipulative may result in a large number of instructions. Using a machine learning approach and continuously training the model, this issue is resolved. Work put into development of the system is reduced and the efficiency is continuously improved. Learning from data may have a slow rate of growth in efficiency but it keeps on improving with the right parameters when compared to predefined explicit programming. L-Protct thus has a significant impact on user security and protection from malware due to its nature of learning. It is essential to continuously monitor content one is exposed to while providing privacy and intelligence to the user. Ignorance should never take away one's right to privacy and security.

REFERENCES

[1] RobSeace Local vs remote exploits lwn.net/Articles/91280/
[2] Dancho Danchev, Which is the most popular malware propagation tactic? ZDNet, 2011.
[3] Bill Sweeney, Social Engineering: How an Email Becomes a Cyber Threat SecurityWeek.com, 2015
[4] Google Chrome, Content Scripts in Google Chrome developer.chrome.com/extensions/content scripts
[5] Mozilla Document Object Model developer.mozilla.org/en-US/docs/Web/API/Document Object Model/
[6] Google Chrome, Message Passing in Google Chrome developer.chrome.com/extensions/messaging
[7] Flask Flask, a Python Microframework flask.pocoo.org
[8] UCI UCI Machine Learning Repository: Spambase Data Set archive.ics.uci.edu/ml/datasets/Spambase
[9] Department of Computer Science, UCL COMPGI01 – Supervised Learning cs.ucl.ac.uk/students/syllabus/mscml/gi01 supervised learning/
[10] David Poole, Alan Mackworth Bayesian Classifiers Artificial Intelligence - foundations of computational agents
[11] Bing Liu, Sentiment Analysis: mining sentiments, opinions, and emotions Cambridge University Press, June 2015