KNOWLEDGE BASED APPROACH TO DETECT POTENTIALLY RISKY WEBSITES

Mr.T.Prasanth [1], Ms.M.Geetha[2], Ms.V.Haripriyanka[3], Ms.M.Rubha[4],Ms.P. Shalini[5]

Assistant Professor[1], Students [2 3 4 5], Department of Information Technology,

Erode Sengunthar Engineering college,

Perundurai, Erode, Tamil Nadu, India.

ABSTRACT

Unsupervised web page classification refers to the problem of clustering the pages in a web site so that each cluster includes a set of web pages that can be classified using a unique class. The existing proposals to perform web page classification do not fulfill a number of requirements that would make them suitable for enterprise web information integration, namely: to be based on a lightweight crawling, so as to avoid interfering with the normal operation of the web site, to be unsupervised, which avoids the need for a training set of pre-classified pages, or to use features from outside the page to be classified, which avoids having to download it. In this article, we propose CALA, a new automated proposal to generate URL-based web page classifiers. Our proposal builds a number of URL patterns that represent the different classes of pages in a web site, so further pages can be classified by matching their URLs to the patterns. Its salient features are that it fulfills all of the previous requirements, and it has been validated by a number of experiments using real-world, top-visited web sites. Our validation proves that CALA is very effective and efficient in practice

KNOWLEDGE BASED APPROACH TO DETECT POTENTIALLY RISKY WEBSITES

ABSTRACT

The challenge of grouping web pages inside a website so that each cluster contains a collection of web pages that can be categorised using a distinct class is known as unsupervised web page categorization. A number of requirements that would make the existing proposals for web page classification suitable for enterprise web information integration are not met, including the need to be unsupervised, which eliminates the need for a training set of pre-classified pages, to be based on lightweight crawling to prevent interfering with the website's normal operation, and to use features from outside the page to be classified to avoid downloading it. In this paper, we introduce CALA, a novel automated solution for creating URL-based classifiers for web pages. Our plan creates several URL patterns that correspond to the various page classes found in a website, allowing further pages to be categorised by comparing their URLs to the patterns. Its key characteristics are that it satisfies every one of the earlier prerequisites and has been verified by numerous tests on popular, real-world websites. Our validation demonstrates how effective and efficient CALA is in real-world scenarios.

Keywords: Machine Learning, Malicious URL Detection, Adversarial Attacks, Malicious Web Services

1. INTRODUCTION

1.1 MACHINE LEARNING

Artificial intelligence (AI) has a branch called machine learning that makes computers capable of learning without actual programming. Through training on data, computers can identify patterns and make predictions. Machine learning algorithms find applications in various domains, including spam filtering, fraud detection, product recommendation, and image recognition. The three categories of these algorithms are reinforcement learning, unsupervised learning, and supervised learning. Machine learning proves to be a powerful tool for solving diverse problems. Nonetheless, it is critical to recognize that the completeness and quality of the training data has a significant impact on how well machine learning algorithms perform. Biased or incomplete training data can lead to biased or inaccurate predictions.

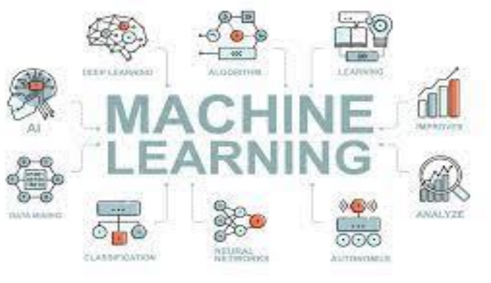

Figure 1. Machine learning

1.2 MALICIOUS URL DETECTION

Malicious URL detection involves the identification of URLs that direct users to malicious websites. These websites can distribute malware, steal personal information, or launch phishing attacks. Traditional methods of malicious URL detection, such as blacklists and heuristics, are progressively losing their effectiveness as attackers develop new evasion techniques. Machine learning presents a promising approach to address this issue. By training machine learning algorithms, patterns in malicious URLs that are difficult for humans to detect can be identified. These patterns may include the utilization of specific keywords or domains, the presence of suspicious characters in the URL, and the reputation of the hosting website.

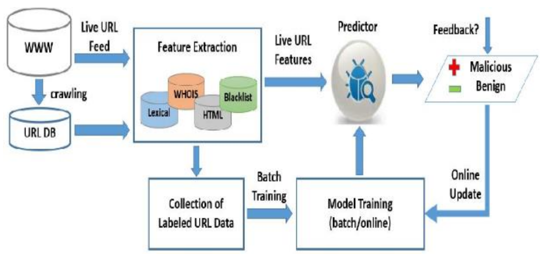

Figure 2. Malicious URL Detection

1.3 ADVERSARIAL ATTACKS

Adversarial attacks involve carefully crafted inputs that aim to deceive machine learning algorithms, causing them to make errors. These attacks can target various types of machine learning algorithms, including those used for classifying images and detecting objects. Adversarial attacks are typically created by making subtle changes to the input data, which may be imperceptible to humans. For example, adding a small amount of noise to an image could be an adversarial attack on an algorithm for classifying images. Although this noise may go unnoticed by humans, it can be sufficient to trick the algorithm into misclassifying the image. Adversarial attacks pose a significant threat to the security of machine learning systems. If successfully executed, an attacker could potentially gain control of the system or manipulate it to make harmful decisions.

1.4 MALICIOUS WEB SERVICES

Malicious web services refer to web services that are intentionally designed to carry out malicious activities, such as distributing malware, stealing personal information, or launching phishing attacks. These services often masquerade as legitimate ones, making them challenging to identify. Malicious web services can be utilized in various ways, including, Malware distribution: Malicious web services serve as a means to distribute malware, such as viruses, trojans, and worms. This can be achieved by embedding malware within web pages, scripts, or other downloadable files. Personal information theft: Additionally, credit card numbers, Social Security numbers, names, addresses, and other personal information are stolen through malicious web services.

2. LITERATURE REVIEW

Stefano Calzavara [1] et.al. Has proposed in this paper, in this article, we examine the most well-known attacks on web conferences, or attacks that go after legitimate web browsers setting up a confirmed meeting with a trusted online application. Next, we examine current security measures that prevent or lessen the different attacks by rating them based on four distinct criteria: security, ease of use, resemblance, and send ability. We also assess some secured arrangements that aim to provide robust defences against multiple attacks. With this overview in mind, we identify five guidelines that the authors of the different proposals we audited have taken into consideration, varying in degree.

Avinash Sudhodanan[2] et.al. Has proposed in this paper Cross-Webpage Solicitation Falsification (CSRF) assaults are a few of the main dangers facing web apps. Cross-site vulnerability research (CSRF) attacks that target site validation and board functionality are the main topic of this paper. Confirmation CSRF is the general term we will use to describe them. We began by gathering a few Auth-CSRF assaults revealed in the writing, then, at that point, examined their hidden systems and recognized 7 security testing techniques that can assist a manual analyser with uncovering weaknesses empowering Auth-CSRF. In order to evaluate the efficacy of our testing procedures and assess the rate of Auth-CSRF, we conducted an exploratory analysis involving 300 sites that fall into three distinct position ranges within the Alexa global top 1500. The results of our investigations are unsettling: of the 300 sites we looked at, 133 were eligible to guide our analyses, and 90 of these had at least one vulnerability that allowed Auth-CSRF to operate (for instance, 68% of them).

Stefano Calzavara[3] et.al. Has proposed in this framework, Web meetings are delicate and can be gone after at a wide range of levels. Exemplary assaults like meeting commandeering, meeting obsession and crosssite demand falsification are especially perilous for web meeting security, since they permit the assailant to break the respectability of genuine clients' meetings by manufacturing demands which get verified for the casualty's sake. In this article, we organize the available defenses against these attacks and their shortcomings, which could completely undermine security in the event that there are clear suspicions about the attacker's capabilities. Next, we build on our security analysis to familiarize black-box testing methods with identifying unreliable meeting execution practices on active sites, a task we accomplish in a Dredd program extension.

Ayman Taha[4] et.al. Has proposed in this framework; Protection is an information rich area, facilitating huge volumes of client information that is broke down to assess risk. AI methods are progressively utilized in the viable administration of protection risk. However, protection datasets tend to be low quality, with noisy subsets of data (or highlights). Selecting appropriate information components is a critical pre-processing stage in the creation of AI models. It has been demonstrated that the inclusion of repetitive and immaterial elements affects how learning models are displayed. In this paper, we suggest a framework for advancing futuristic AI techniques in the field of protection through the selection of significant highlights. The exploratory outcomes, in view of five freely accessible genuine protection datasets, show the significance of applying highlight choice for the evacuation of boisterous elements prior to performing AI methods.

Martin Johns[5] et.al. Has proposed in this framework, Cross-Webpage Solicitation Fraud (CSRF) weaknesses are an extreme class of web weaknesses that definitely stand out enough to be noticed from the exploration and security testing networks. Even though a lot of time and effort has gone into finding countermeasures and locating SQLite and XSS, up until now, the identification of CSRF vulnerabilities is still done transcendentally physically. As far as we are aware, Deemon is the main computerized security testing system for identifying CSRF vulnerabilities, which we present in this paper. Our methodology is predicated on another illustrative worldview that captures various aspects of web applications, such as engineering levels, information owes, and execution follows, in an untied, comprehensive property chart.

3. EXISTING SYSTEM

This article presents a methodology to identify web application vulnerabilities using machine learning (ML). Because of their diversity and the extensive usage of custom programming techniques, web applications present unique challenges for analysis. As a result, by using manually labeled data, machine learning (ML) enhances web application security by allowing automated analysis tools to incorporate human understanding of web application semantics. In order to create Mitch, the first machine learning solution for the black-box detection of Cross-Site Request Forgery (CSRF) vulnerabilities, the suggested methodology was used. Mitch's use resulted in the discovery of 35 new CSRFs on 20 prominent websites and 3 new CSRFs on production software.

4. PROPOSED SYSTEM

With an emphasis on enterprise online information integration, the suggested system, CALA (URL-based online Page Classifier), provides a complete answer for unsupervised web page categorization. Fundamentally, CALA makes use of lightweight crawling techniques that allow users to enter entry point URLs with the least amount of interference with the target website's regular operations. By using this method, CALA groups gathered URLs into hub sets, which act as focal points for distinct online content categories. The system then makes use of pattern building algorithms to identify recurrent URL patterns that correspond to different categories of web pages on the website. After that, these patterns are tagged with semantic labels, which can be produced at random in the event that specifics are not easily obtained, or manually supplied by users. By ensuring that every pattern has a meaningful description, this important step makes correct classification possible. Lastly, new web pages input by users are classified by CALA's URL classification module, which uses the annotated patterns to classify them purely on the basis of URL structure.

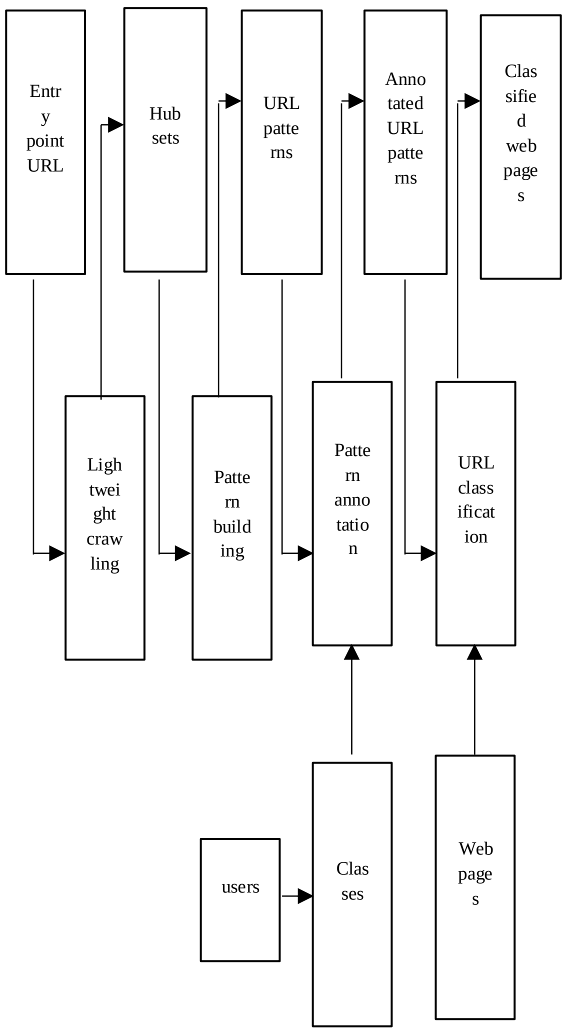

Figure 3 SYSTEM ARCHITECTURE

6.2.1 LIGHTWEIGHT CRAWLING

This module, which aims to reduce the strain on the web server from which data is being crawled, serves as the starting point for CALA's classification process. Users have the option to enter one or more URLs as the entry point URLs, which are where CALA starts its analysis. The module's design ensures that the system runs quietly and uses less bandwidth by extracting URL data without downloading the entire web page.

6.2.2 HUBSETS

This module uses CALA to group gathered URLs into hub sets, which are groupings of URLs that act as hubs or central locations for related kinds of data. Users can observe how URLs are organised before they are processed further for pattern development by viewing hub sets. This module aids in comprehending the website's structural arrangement and locating important nodal points that connect to different content categories.

6.2.3 PATTERN BUILDING

The functioning of CALA depends on pattern building. This module examines the collected and arranged URLs in the Hub sets to find recurring themes that correspond to various website page classes. CALA lays the groundwork for precise URL classification of new URLs without the need to download or analyse their content directly by comprehending and extracting these patterns.

6.2.4 PATTERN ANNOTATION

After patterns are found, they need to be labelled with semantic descriptions that explain what they mean or how they work on the website. This annotation is based on human input, where semantic labels are manually provided to each pattern of URLs. Each pattern must have a semantic class label entered. These labels can be supplied randomly or based on user knowledge if particular details are not needed at this time. In order to accurately classify new URLs in accordance with the existing patterns, this step is essential.

6.2.5 CLASSIFICATION OF URLS

The last module categorises fresh web pages based on the patterns and their annotations. URLs for inquiry webpages can be entered by users, and CALA will categorise them according to how well they follow the annotated patterns. This module allows web pages to be quickly and efficiently categorised based just on their URLs, without having to view the content of the page. It is the practical application of all the previous processes.

ALGORITHM DETAILS

Our programmer requires the URL of the website's entry point, or the URL of a page that has a keyword-based search form that users may fill out and submit to find hub pages. This condition is often met by the home page of the website. We present a working example of automatically classifying web pages using CALAB, which involves categorizing the pages on the Microsoft Academic Search website. This is an academic website that provides details about publications, writers, citations, journals, conferences, and other publishing hosts. As an illustration, the search form that serves as the <MSAS> website's entry point. Following the retrieval of a hub set, the programmer constructs several URL patterns that correspond to the various URL classes of that website. The user will likely need to annotate certain URL patterns after the fact. Keep in mind that the set of patterns is far less than the set of pages in a site, so there is very little expense involved in annotating them.

```
algorithm  $gatherHubsets$
input      $fp : URL$
output     $Result : Hubset$
variables  $wp : WebPage; n : \mathbb{N}; P, Q :$ set $Word$

$n$ := $0$
- Step 1: Download initial page
$wp$ := $download(fp)$
- Step 2: Compute keywords from page contents
$Q$ := $computeKeywords(wp)$
$P$ := $\emptyset$
while $n \leq T \wedge Q \neq \emptyset \wedge \#Result < M$ do
    while $Q \neq \emptyset \wedge \#Result < M$ do
        - Step 3: Submit form using keyword from Q
        $kw$ := get one keyword from $Q$
        $P$ := $P \cup \{kw\}$
        $wp$ := $submit(fp, kw)$
        - Step 4: Create hub using the URLs in $wp$
        $Result$ := $Result \cup \{computePatterns(wp)\}$
        - Step 5: Update the set of keywords
        $Q$ := $Q \cup computeKeywords(wp) \setminus P$
    end while
    - Step 6: Keep only non-empty hubs
    $Result$ := $getNonEmptyHubs(Result)$
    $n$ := $n + 1$
end while
```

.5. RESULT ANALYSIS

The suggested algorithm shows an improved accuracy of 85% over the current algorithm's 80%. By using lightweight crawling approaches, it minimises disruptions to target websites, groups URLs into hub sets for focal points within categories, and uses pattern building algorithms to find recurrent URL patterns for classification. Meaningful descriptions are guaranteed for precise categorization by semantic labelling. The process of categorising new web pages based only on their URL structures is then made more accurate and efficient by the URL classification module, which makes use of annotated patterns.

algorithm	accuracy
Existing	80
Proposed	85

Table 1. Comparison table

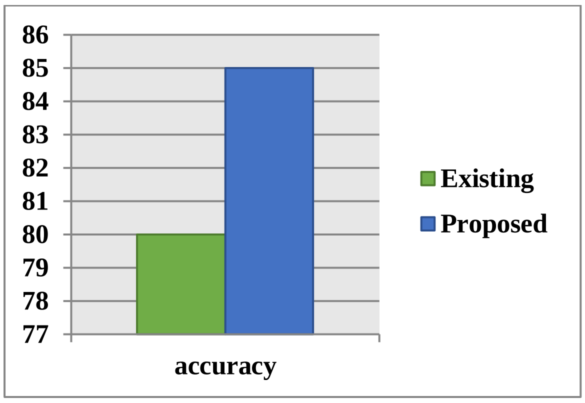

Figure 4. Comparison graph

6. CONCLUSION

Finally, we have introduced CALA, an automatic web page classifier generation proposal that may be applied to enterprise online information integration systems. Our proposed system receives as input the URL of a webpage including a keyword-based search form and returns a set of patterns that correspond to the URLs of pages in the same class. Three key needs for enterprise online information integration are satisfied by our system: It is unsupervised because it learns a classifier from a training set of automatically gathered, non-classified hubs from the website. It is based solely on URL features, allowing a page to be classified without having to be downloaded first. It gathers a sample of hubs automatically by performing a lightweight crawl.

7. FUTURE WORK

Future versions of CALA might investigate various directions for improvement and growth. Using machine learning techniques to automate the process of pattern annotation could be one way to go, since it would reduce the need for manual labour and improve scalability. Furthermore, the adaptability and accuracy of the system could be improved by improving the pattern forming algorithms to handle dynamic website structures and changing content. To further increase classification accuracy, future study could further expand CALA's capacity to classify web pages based on criteria other than URL structures, like page content or metadata.

8. REFERENCES

[1] Stefano Calzavara, Alvise Rabitti, Alessio Ragazzo, and Michele Bugliesi. Testing for respectability imperfections in web meetings. In PC Security - 24rd European Discussion on Exploration in PC Security, ESORICS 2019, Luxembourg, Luxembourg, September 23-27, 2019, pages 606-624, 2019.

[2] Stefano Calzavara, Mauro Conti, Riccardo Focardi, Alvise Rabitti, and Gabriele Tolomei. Mitch: An AI way to deal with the blackbox recognition of CSRF weaknesses. In IEEE European Conference on Security and Protection, EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019, pages 528-543, 2019.

[3] Stefano Calzavara, Riccardo Focardi, Marco Squarcina, and Mauro Tempesta. Getting through the web: An excursion into web meeting security. ACM Comput. Surv., 50(1):13:1-13:34, 2017.

[4] Avinash Sudhodanan, Roberto Carbone, Luca Compagna, Nicolas Dolgin, Alessandro Armando, and Umberto Morelli. Huge scope examination and discovery of verification cross-site demand fabrications. In 2017 IEEE European Conference on Security and Protection, EuroS&P 2017, Paris, France, April 26-28, 2017, pages 350-365, 2017

[5] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Component

determination: An information point of view," ACM Comput. Surv., vol. 50, no. 6, pp. 94:1-94:45, 2017