Web Application Profiling using Statistical Techniques

Rushita Thakkar :1401004 Ishika Agrawal : 1401067
School of Engineering and Applied Science,Ahmedabad University

Abstract—"World is Online!" everything from Buying clothes, Banking, Trading, Watching Movies is done online. This makes security of the customers as well as business paramount. Number of internet users is increasing with every passing day. According to a survey, almost 2/3rd of the Applications are vulnerable and 90% of vulnerabilities lie on Application Layer. Application Penetration Testing is the process of testing applications for vulnerabilities like Authentication Issues, Cross Site Scripting, Session Fixation by giving standard attack vectors and the testing is mainly based on the Application layer. There are various Scanners available in market that check for various vulnerabilities and help the penetration Testers. The approach suggested here is however, different. The paper gives a raw fresh glance at the Application Penetration testing by looking at the Connectivity of the Website and extracting various features and building a Connectivity graph. Various Webometric measures are also taken into consideration and a Vulnerability Score is given to the website(Regression Analysis). A visual graph and a vulnerability score(Ranking Based on the Score) is given as an output that would ultimately help the Application Penetration tester so that important and vulnerable web pages are not missed out.

Index Terms—HTTP Request/Response,Penetration Testing,Application Security,Application Profiling,Graph Network,Centrality Measures,Regular Expressions,Link Analysis,Correlation Coefficient, Multiple Linear Regression

1 Introduction

1.1 Background

Everything is done Online these days. From buying clothes, taking part in competitions, buying stocks, making friends, shopping, investing in Crypto-Currencies. A lot of personal as well as important Company data is stored. This data can be misused. Information is Power and with great power comes great responsibilities. It is a responsibility of the Web Businesses to protect their data as well as the data of their customers and make it impossible for the hackers to breach. Statistics about 2017's cyber-attacks is that they're expected to cause 5 billion dollars worth of damages. That's a staggering fifteen-fold increase over just two years[7]. Some of the most recent attacks are:

Yahoo mail Services:Yahoo's propriety data was accessed and misused by the attackers. They knew how to forge certain cookies. News articles suggested that the data from almost 500 Million Users might have been stolen.

Equifax: Hackers exploited a vulnerability in the web application to gain access to certain resources. It leads to getting personal information for nearly 150 million people.

Ethereum: Ether is a cryptocurrency like bitcoin, and 7 million dollars in Ether was stolen from the Ethereum application platform in a manner of small time.

Why is Appliation Security so Important 90 percent of applications have one or more types of vulnerabilities that pose major threats from attackers who may exploit them to breach their security. Since automated scanning proves inadequate in uncovering all exploitable vulnerabilities, Application Penetration Testing Service complements this with manual testing powered by expert human intelligence.

These are all the recent examples. With more and more development in Technology and increase increasing sizes of the data storage devices, it is important that data is secure.

Application Penetration Testing: [3]Penetration test is an attempt to find out all the vulnerabilities to check if unauthorized hacker can gain access to important resources or possibly do any other malicious activity by any means. Penetration testing basically is of two types: Network penetration testing and application security testing, also the controls and processes around the networks and applications. It should be done from both outside the network which is trying to come in (external testing) and from inside the network (internal testing).

Application Profiling Profiling a web application to prioritize the importance of various functionality of the application which would further aid in risk rating the severity of the vulnerabilities detected on particular functionality

In this age of Automation and Artificial Intelligence, the question here is "Has there not any been attempts in the direction of an Automated Application Penetration Tester?" The answer is Yes. There are various Scanners available in the Market like the **Acunetix Scanner**, **App SCAN** by IBM ,**Zed Attack** by OWASP etc. These tools lie in DAST (Dynamic Application Security Tools). They look for security vulnerabilities such as Cross-site scripting, SQL Injection, Command Injection, Path Traversal and insecure server configuration, A web application scanner, used to identify security vulnerabilities in a web applications does not replace an experienced penetration tester, rather it's a valuable tool in their arsenal and an excellent interim measure when the pen tester is not available. By taking the opinions of experienced Application Penetration testers

, it was evident that Scanners cannot replace Penetration Testers.

1.2 Motivation

As mentioned, as of now it is not possible to replace a human penetration tester. With advancement in Artificial Intelligence Algorithms, a day might come when a computer program would completely replace a human Penetration Tester. With the availability of various Scanners what is something that can be done to improve the efficiency further. A connectivity based approach or looking at the graphical representation in order to determine the vulnerability score has not yet been done and looking at centrality scores of nodes(web pages in our case) in order to determine the vulnerability score is a new concept. Giving out the vulnerability Score on the basis of Connectivity of the graph of a website and various other factors seemed a totally new and an interesting idea.

1.3 Main Idea

Proxy tools are used to monitor the HTTP Logs. Every time a request is rendered from a client (web browser), web server sends a response back to the client. All the request response data of every web page is tracked in the proxy tool. By surveying the Penetration Testers, it was found out that on an average, around 2 hours are spent just to go through the website and get the idea of the flow. Almost all websites do have a particular flow but a visual interactive graph would give a proper idea about the flow of the website and make it easy for the application penetration tester. A connection graph for the web pages of the websites is created which gives an idea of the structure of the entire website.

The output is a table with all the web pages of the application and their connectivity centrality values and their payment and session related parameters as well as the final score of how much vulnerable it is considering all the features. Final result is a visual graph with nodes colored according to the intensity of how vulnerable it is given which would give a head start to the penetration tester.

Some experienced Pen Testers gave vulnerability scores to some test web applications according to their experience. Normalization on the manual scores according to the centrality scores was done and then manual scores were considered for Regression analysis.

Many features had been considered for regression analysis of the web pages but eventually by experimenting with a lot of websites and Analysis of Variance and Correlation, importance of each factor was known. Experimenting and analyzing nature of different websites eventually helped in deciding which factors have more importance in deciding the Vulnerability Score.

Accuracy of the formal score obtained by regression are given. There is possibility for improving the regression accuracy with more and more data. Currently, data for 7 testing websites has been considered. Our results are expected to reduce the manual effort for application penetration testing.

2 Types of attacks

A web application is constituted of a collection of scripts, which lie on a web server and interact with databases or other sources of dynamic content. Using the infrastructure of the Internet, web applications allow service providers and clients to share and manipulate information in a platform - independent manner. The increasing complexity of the technologies used to develop the web applications , as well as the lack of security expertise, can largely explain the recurring vulnerabilities they present. .

1) **SQL Injection Attack:** Servers that store crucial data for web applications and services use Query language like to manage their databases (MySQL, PostgreSQL, TransactSQL, MongoDB). A SQL injection attack specifically attacks server, using malicious code to get the server to obtain information, one cannot access normally .
   This is specifically problematic if the server stores private customer details from the website, such as credit card numbers, username and passwords credentials, personal unique information like Adhar Card number, Social Security number, Passport number, which might be tempting targets for an attacker. For example, if a database server is exploitable to an injection attack, it may be possible for an attacker to go to a website's search box and type some code that would manipulate the site's SQL server to give out all of its stored sensitive credentials for the site.

2) **Cross-Site Scripting (XSS):** Similar to the SQL injection attack, XSS attack also involves injecting malicious code into a website. But in this case, the web application itself is not being attacked, in fact the malicious code the attacker has injected runs in the user's browser itself when they visit the attacked website. It attacks user directly, not the website.
   One of the common way an attacker can implement a cross-site scripting attack is by inserting malicious code into a comment or any other dialog box or a javascript that could manually run. For example, they could append a link to a malicious JavaScript in a comment on a video or article.
3) **Session Hijacking:** The session between our local computer and the web server is a unique session, identified by a Session-Id, which should stay secured between the two parties(client and server). But an attacker can hijack the session by obtaining the Session-Id and posing as the target user making a request. This allows them to log in as an unsuspecting user and gain access to unauthorized information on the web server, rights of which are only for target user. There are multiple methods an attacker can steal the Session-id, such as a XSS (cross-site scripting) attack is generally used for hijacking sessions.
4) **Cross-site request forgery:** CSRF is another type of malicious exploit of a web application where unauthorized instructions are transmitted from a

authorized user of a web application. It makes use of the trust application has on its frequent users. There are multiple ways to execute this type of attack, where a malicious website can redirect such commands. For example, crafted image tags, hidden forms, and JavaScript XML can work without the user's knowledge or involvement.

5) **Insecure direct object references:** Insecure direct object reference occurs when a website gives out a reference to an internal object. These implementation objects may include file systems, database information, directories, paths and database keys. Hackers can use it to gain unauthorized access to a user's personal and sensitive data.

3 Web Request/Response

- HTTP - Hyper-text transfer protocol, is a basis for data communication in the internet. The data communication starts with a request sent from a client i.e web browser and ends with the response received from a web server.
- HTTPS is the secured HTTP protocol required to send and receive information securely over internet. Nowadays it is mandatory for all websites to have HTTPS protocol to have secured internet. Browsers like Google Chrome will show an alert with "Not Secure" message in the address bar if the site is not served over HTTPS.
- A simple request message from a client computer consists of A request line to get a required resource, for example a request GET /content/page1.html, Request Header, A message body (optional). Sample request header is:

  GET /bank/redirect.html HTTP/1.1
  Host: zero.webapp.com
  User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:59.0)
  Accept: text/html,application/xhtml+xml,application/xml;
  Accept-Language: en-US,en;q=0.5
  Accept-Encoding: gzip, deflate
  Referer: http://zero.webapp.com/account-activity.html
  Cookie: JSESSIONID=A2E9D5EB
  Connection: close
  Upgrade-Insecure-Requests: 1

- Sample Request Data:
  payee=apple & **account**=3 & **amount**=50 & **date**=2018-04-03
- Sample response header from Web server tracked in proxy tool:

  HTTP/1.1 200 OK
  Date: Wed, 04 Apr 2018 08:37:51 GMT
  Server: Apache-Coyote/1.1
  Access-Control-Allow-Origin: *
  Cache-Control: no-cache, max-age=0, must-revalidate, no-store
  Content-Type: text/html;charset=UTF-8
  Content-Language: en-US
  Connection: close
  Content-Length: 10223
- Response Data is the content that is displayed in web browser(if 2xx series). 2xx series status code represents OK condition, 3xx series represent re-direction, 4xx and 5xx series represents error.
- Sample response data for error in display is:

```
<html>

<head>
<title>404 Not Found</title>
</head>
<body>
<h1>Not Found</h1>
<p>The requested URL /t.html was not found on this server.</p>
</body>
</html>
```

4 Process Flow

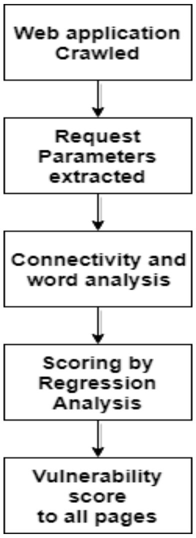

Web application Crawled
Request Parameters extracted
Connectivity and word analysis
Scoring by Regression Analysis
Vulnerability score to all pages

Figure 1: Process Flow

5 Data Extraction

Seven different test web applications were considered for training our regression model. After manual crawling and logging all exchange of web request and response, from the proxy tool, we get XML data(extensible markup language). There is list of all the requests in XML form passed as raw data to our model. For each request, we extract following attributes:

1) **Hostname:** Host is the local device connected to a bigger computer network and hostname is used to identify the host device in the vast communication system namely the World Wide Web. A hostname is alternatively called domain name. Extraction of hostname is done from <hostname> tag.
E.g. <Hostname>www.xyz.com< /Hostname>

2) **URL:** A Uniform Resource Locator (URL) is a reference to a web resource within a website. It specifies its location on a host. It is extracted from $< Url >$ tag.
E.g. <Url> www.xyz.com/abc< /Url>.

3) **Request Header:** Every web request from client(web browser), have two important parts: Request Header and Request Data. Request Header have the details of what the browser wants and will accept back from the server. The request header contains information of web request. It also details of the type, version and capabilities of the client browser that is making the request so that server returns corresponding data. We extract Request Method and Referer from here.
Two commonly used methods for a request-response between a client and server are: GET and POST.

   - GET - Requests data from a specified resource
   - POST - Submits data to be processed to a specified resource

   HTTP referer is one of the HTTP header field that identifies the path of the page in the website that is linked to the resource being requested. From the referer, the new web page one can see where the request originated. It is important for connectivity between two different resources, which might be missed out.

4) **Request Data:** Request is created and then passed by reference to the various layers of an application which use request data. Request Data is generally present in POST requests. For example, during login, username and password are passed as Request Data in the form "name1=value1&name2=value2". It might consist of some important payment transaction credentials, which should be properly encoded and should not be in hands of a malicious hacker through Cross-site-scripting or Cross-site-request-forgery attacks.

5) **Response Header:** Response header represents details of the response sent from server(web server) to client(web browser). Status codes are issued by a server in response to a client's request made to the server. Status code is extracted as one of the parameter from Response Header, as it can be vulnerable if redirection to other pages occur. The first digit of the status code specifies one of five standard classes of responses. If status code is in 3xx series: the location is extracted. Location is important for connectivity purpose.

6) **Response Data:** Response data is the web content that is seen in the browser. It contains text, images,videos and URLs. All the different URLs attached with a particular web page are collected. URLs are extracted in following formats from response data:

   - *<a href="https://abcd.com">*
   - *<form action="https://abcd.com"/ >*
   - *<script src="https://xyz.com/js/app.js">*
   - *<meta URL='http://abcd.example.com/'"/ >*
   - *<link rel="help" href="/help"/ >*

   This way, using different tags for links, track of all outgoing URLs can be maintained for all web-pages. It can be used for connectivity analysis.

6 DATA MODEL

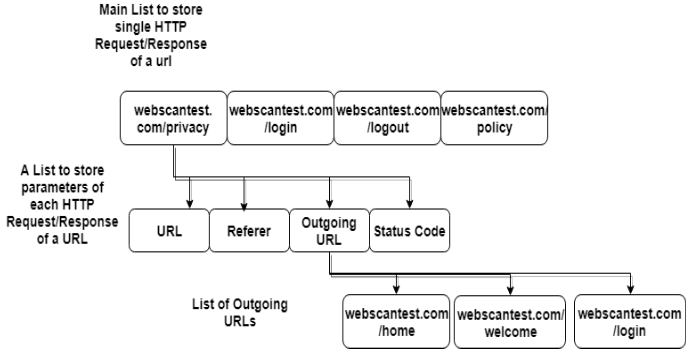


Figure 2: Data Model

7 CENTRALITY MEASURES

7.1 Definitions:

Let G be an directed single-edged graph comprised of the sets N nodes and E edges.A is adjacency Matrix such that $a_{ij} = 1$ if $e_{ij} \in E$ and 0 otherwise.Let W be an edge weight matrix such that w_{ij} is given by $d(i,j)$. The shortest path between two nodes i,j is σ_{ij}.The number of shortest paths between two nodes i,j that include node u is $\sigma_{ij}(u)$. $|N|$ represents the number of nodes in the graph.

In degree Centrality: In degree Centrality is the function of incoming edges for the node. It would determine how important a web page is on the basis of the incoming links to the node.The in degree centrality for the web page is given by:

$$D_i = \sum_{j \neq i} \frac{a_{ij}}{|N| - 1}$$

Out degree Centrality: Out degree Centrality is the function of out going edges for the node. It would determine how important a web page is on the basis of the out coming links to the node(web page).The in degree centrality for the web page is given by:

$$D_i = \sum_{j \neq i} \frac{a_{ij}}{|N| - 1}$$

Closeness Centrality Closeness centrality is a function of a node's average distance to all other nodes in the network.The nodes are Web pages in our cases. It represents the node's level of communication independence. Web pages that are close to all other web pages may be important.

Closeness centrality is calculated as:

$$C_i = \frac{|N| - 1}{\sum_{i \neq j} d(i,j)}$$

Betweenness Centrality: Betweenness is a function of the number of shortestpaths that pass through a node. It is regarded as a measure of a node's control over communication flow. Web pages that come in the between the shortest paths of various other webpages may be more important.

Betweenness centrality is given by:

$$B_i = \sum_{s,y \in G, s \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}$$

PageRank Algorithm: PageRank is a link analysis algorithm that estimates a node's importance based on the importance of the nodes that link to it. It also suggests the probability with which we would reach a particular node from a certian node. . PageRank Algorithm is used by Google to rank web pages in the search results. It is a recursive algorithm where nodes accumulate PageRank scores and pass fractional shares to their neighbors. The basic algorithm for pagerank is shown below:

```
$Pr_0(i) = 1 \; \forall i$;
$maxiter = 5$;
$iter = 1$;
for $iter \leq maxiter$ do
    for $\forall i \in N$ do
        $PR_{iter}(i) = \beta \sum_{i \neq j} \frac{a_{ij}}{D_j} PR_j + (1 - \beta) \frac{1}{|N|}$;
        iter++;
        $\epsilon_i = PR_{iter}(i) - PR_{iter-1}(i)$
        $\sigma = \sum_{\forall i} |epsilon|$
        if $\sigma \leq 0.001$ then
            return $PR_{iter}$
        else
            continue;
        end
    end
end
```

Algorithm 1: PageRank Algorithm

The β parameter and second term form a random feature that ensures the total PageRank is conserved between iterations and does not get accumulated in edge cycles. β is also called the damping factor which can be adjusted in our case β = 0.85.

Page Rank for each web page calculated in order to determine the vulnerability based on connectivity of the web site graph. In lay man term, Page Rank gives the probability with which a person it likely to reach that web page while browsing.

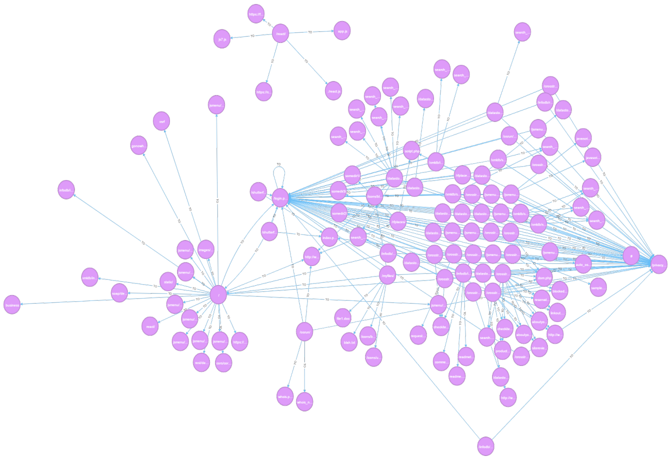


Figure 3: Connectivity Graph of sample web-site *www.webscantest.com*

8 Word Analysis:

- In banking and e-commerce web applications, the payment transactions occur in some particular webpages. These pages are important for security checks as they might be more vulnerable, as money is involved. Developer of web application have to make very sure, that payment is done properly with no loophole for a malignant hacker to attack.

- Apart from payment and transaction pages, session fixation threats are not less important. A hacker can try to gain access of login credentials of some other person through session cookie, if not properly secured.

- After logging in a banking application, he/she can withdraw all the money out of target victim and all blame goes to developer of the web application.

- We separate this class of web-pages like login,payment etc. to make them priority for penetration testing, so that no tester can miss out these pages.

- Request Data and URL have important sensitive information rather than Response Data. For example, Some banking advertisement text may contain monetary words, but they are useless in vulnerability scoring.

- Request Data and URL of webpage are pre-processed using Natural Language processing(NLP) techniques like stemming, tokenization and lemmatization.

- **Stemming** is the process of reducing derived words to their word stem, base form generally a written word form. For e.g. 'users' $->$ 'user' etc.

- **Lemmatization** normally aims to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma

- **Tokenization** is process of converting whole string to bag of words. All request data parameters(name) are tokenized, stemmed and lemmatized.

- Some of the session words we have considered are:

login	logout	name	password	captcha
signup	email	user	signin	phone
number	otp	ssn	id	user
url	redirect	account	forgot	no

- Some of the Payment jargon words:

creditcard	visa	mastercard	cvv
debit	payment	card	number
code	company	account	paypal
rupee	usd	number	amount
currency	dollar	euro	yen
owner	postal	paytm	rs

- Parameter values are important too for identifying sensitive content like payment information. There are some false positives observed, where variable names are not as expected. For example, ”Sno” for Social security number, ”Cnum” for Credit card number.

- For overcoming such false positives, we consider the traditional formats of following codes:
  - Social Security Number - ”AAA-GG-SSSS”
  - Credit Card Number - ”XXXX-XXXX-XXXX-XXXX”
  - CVV number - ”012”
  - Adhar Card number - ”1111-1111-1111”
  - Debit Card number - ”XXXX-XXXX-XXXX”
  - Cheque account number

- Other preprocessing like case conversion, restriction to alphanumeric characters is done. Regular expressions are used for pattern identification. Count of session and payment jargon is maintained.

9 SCORING PROCESS:

9.1 Manual Scoring by a Penetration Tester

The aim to score the web page of the Web Application based on the vulnerability. A various aspects and features are concerned for the scoring. The values for these features have been collected and stored as mentioned in how the data was collected.

An experienced Penetration Tester was given the task of manually scoring the web pages of the web application based on how vulnerable they were.

The points kept in mind for Scoring the Web Pages :

- Is there a query string in request data?
- Is there a query string in URL(XSS vulnerability)
- Status Code for checking re directions
- Request Method (GET/POST)
- CSRF token present or not

9.2 Features extracted for formula Scoring

- Indegree Centrality
- Outdegree Centrality
- Closeness Centrality
- PageRank Score
- Betweenness Centrality
- Payment Word Count
- Configuration Related Word Count
- Method(Boolean Value 1 if POST method,0 otherwise
- Form tags count(Number of Form Tags in a page)
- Have Third Party API(Boolean Value)

9.3 Linear Regression

Around 10 features have been considered for scoring the websites we are trying to create a formula based on these features for this Regression Algorithm to predict the score of the web pages.

Here $x's$ are the 10 dimensional vectors in $\mathbb{R}^{10}$

For example, $x_1^{(i)}$ is the Indegree centrality of the ith web page and $x_2^{(i)}$ is the Out degree centrality of the ith web page.

Here,taking(an initial choice) f_θ be a linear function of $x's$

$$f_\theta = \theta_0 + \theta_1 x_1 + \theta_2 x_3 + \theta_3 x_3 + \theta_4 x_4 + ...$$

θ_i's are the parameters parameterizing the space of linear functions mapping from X to Manual Score Y

Representing this equation in a Matrix form for simplicity:

$$f_\theta = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

Consider y^i's be the manual score given by the penetration Tester for this,

The cost function:

$$J_\theta = \frac{1}{2} \sum_{i}^{m} (f_\theta x^{(i)} - y^{(i)})^2$$

The aim is to minimize the cost function. Regression uses Gradient decent method for minimization and the parameters or the weights for $x's$ are given by:

$$\theta = (X^T X)^{-1} X^T Y$$

Using this method: The values for θ_i for the problem is given by

$$\theta = \begin{pmatrix} -1.51975713 \\ 1.19948897 \\ 5.33328875 \\ 4.02460338 \\ 1.18104906 \\ 1.80164851 \end{pmatrix}$$

These are the parameters for In-degree centrality, Payment related word score, Configuration words related score, Method. The weight for Payment related word score is highest among others. The Mean Square error for this particular set is: 0.039

9.4 Training Data

A table of features is taken . Around 350 web pages have been considered from the websites like:

www.paytm.com
www.grofers.com
www.webscantest.com
www.demo.testfire.com(test website)
www.acunetix.com
www.flipkart.com
www.zerowebappsecurity.com(test website)

As described in section 9.1, manual score was given based on the security of web pages. Flipkart ,Paytm and Grofers being very well known websites are already secure. The manual score for the web pages of this web sites were closer to 10 i.e. they were very secure.

In case of websites like www.zerowebappsecurity.com and www.demo.testfire.com there were web pages where the manual scores were in lower range(1-3), having more vulnerability.

Acquiring more data may result into a better model.

The distribution of the web pages taken into consideration for training the model:

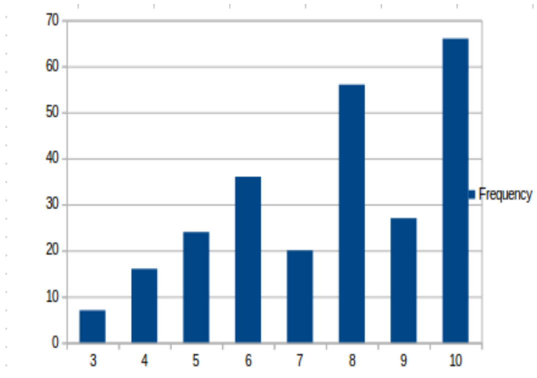


Figure 4: (Data Sample)Frequencies of different manual scores

9.5 Pre-processing on the features:

As we can see that the centrality Scores calculated on the basis of the graph structure of one website lie in the range 0 to 1 and their sum is equal to 1. We calculate the payment words and session words for the website.

All websites are different and the centrality scores on the website are different and are based on one of different scales. This difference in scale would lead to unexpected outcomes in the coefficients for the regression line.

Pre-processing of the data is done by keeping tracking the maximum and minimum value of the feature and finally bringing each value between 0 to 1:

$$\frac{X - X_{minimum}}{X_{maximum} - X_{minimum}}$$

9.6 Importance of different Features

Correlation Coefficient: Correlation between sets of data is a measure of how well they are related.The Pearson Correlation Coefficient is the Statistical formula that measures strength between two variables and relationship.

The coefficient value can range between -1.00 and 1.00. If the coefficient value is in the negative range, then that means the relationship between the variables is negatively correlated, or as one value increases, the other decreases. If the value is in the positive range, then that means the relationship between the variables is positively correlated, or both values increase or decrease together.

Correlation coefficient between different features and the manual Score had been found out. Correlation Coefficient can be found out as shown below (x and y are two variables whose relationship we aim to find)

$$r = \frac{N\sum xy - \sum x \sum y}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum x^2 - (\sum x)^2]}}$$

The Correlation Coefficient of manual score with different features mentioned below:

The correlation between:

- Manual Score and In degree Centrality: **0.276**
- Manual Score and Out degree Centrality: **0.317**
- Manual Score and Betweenness Centrality: **0.23**
- Manual Score and Closeness Centrality: **0.25**
- Manual Score and PageRank Score: **0.26**
- Manual Score and Payment word count score: **0.611**
- Manual Score and Configuration word count score: **0.345**
- Manual Score and Third Party Connection: **0.373**
- Manual Score and Method based Score: **0.64**
- Manual Score and Form tag count score: **0.45**

By looking at these values, it can bee seen that the correlation between the Manual Score and the method score and the correlation between the formula score and the Payment word score is found out to be higher naturally.

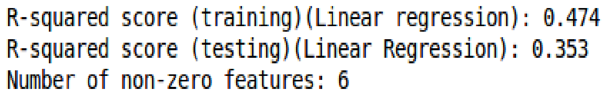
9.7 Model Accuracy

Several parameters can be used to determine the regression model accuracy

R-squared term: R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

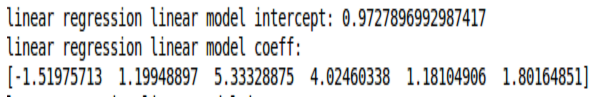
R-squared = Explained variation / Total variation

The order for the coefficients is: Indegree Centrality, Outdegree Centrality, PageRank Score, Payment Related Words, Session Related Words, Request Method



```
R-squared score (training)(Linear regression): 0.474
R-squared score (testing)(Linear Regression): 0.353
Number of non-zero features: 6
```

Figure 5: Result



```
linear regression linear model intercept: 0.9727896992987417
linear regression linear model coeff:
[-1.51975713  1.19948897  5.33328875  4.02460338  1.18104906  1.80164851]
```

Figure 6: Result

As seen above the R square value for training data and testing data is not very different, which implies that there is no major over-fitting or under-fitting of line for this data.

Lasso Regression and **Ridge Regression** regularization was applied for different parameters of α and different number of regressions but $\alpha = 0$ i.e. Linear Regression without regularizing was the best choice for the model[17]. Initially, all 10 features were considered for regression but later, experimenting with the websites, checking the correlation between different features and discussing with very experienced Application Penetration Testers, it was decided to go further with above 6 parameters i.e. Indegree Centrality, Outdegree Centrality, PageRank Score, Payment Related Words, Session Related Words, Method

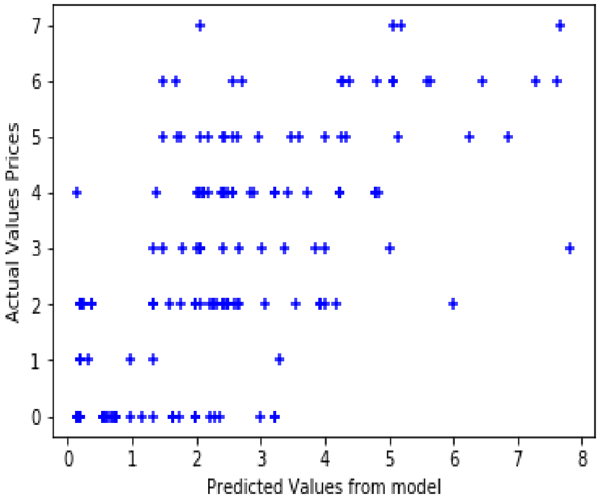


Figure 7: Graph showing correlation between predicted value and manual score for testing data

As it can be seen that the x and y axis values are positively correlated. A lot of variance is seen that can be fixed with better quality training data.

9.8 Conclusions and Insights

- A method based on graph of the website is useful for deciding the vulnerability Scores
- A visual graph based on the Score was given to the Application Penetration Tester and the tester found it to be initially Useful
- There were some False Positives in word analysis, due to pattern matching with useless words e.g. ”Trousers” was considered as session word because of ”user” as a sub string.
- A General trend i.e more the centrality,more is the vulnerability score is not followed but a linear formula considering various scores is found out to be useful.

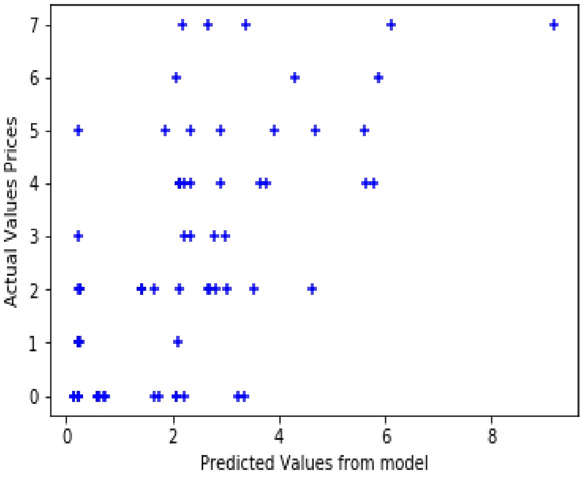


Figure 8: Correlation between testing data and the manual score

- In Regression Analysis, weight for the Pagerank score was found out to be highest.
- By finding the Correlation between manual score and various features,it was found out that correlation between the manual score and the Payment related words was the highest
- The web pages which employed POST Method were found out to be more vulnerable and not just the ones that had many form tags.

10 Future Work

- Training is done based on eight web applications’ proxy web data. Training on more data will improvise the regression model.
- Trying different regression models like Decision trees regression, Logistic regression, Polynomial Regression and Bayesian linear regression. Analysis and comparison of different algorithms will give the most optimized model for correct score prediction.
- Increase number of features, which might be important in terms of vulnerability score like Session cookie, Encoding technique, Third-API calls, etc.
- Look for some other statistical normalization and weighting algorithms to improve the vulnerability score distribution to the web pages.
- Improving word analysis using efficient NLP techniques.

11 Acknowledgements

We would like to express our deep appreciation to all those who provided us the possibility to complete this report. A special gratitude we give to our project mentors, Mr. Hemil Shah, Mr. Amish Shah and Mr. Shreeraj Shah from Blueinfy Solutions Pvt Limited whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project. Our Final Year Project Coordinator Dr Deepak Verma from School Of Engineering and Applied

Science Ahmedabad University who provided us with the opportuniy of Intening at Blureinfy Solutions. We also thank the Faculty members at Ahmedabad University who guided us and provided us the motivation to write this research paper.

References

[1] http://snap.stanford.edu/class/cs224w-2012/projects/cs224w-043-final.pdf

[2] HTTP Request and Response: https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages

[3] Application Penetration Testing: https://www.owasp.org/index.php

[4] Word Analysis http://www.nltk.org/book/ch01.html

[5] NLTK Parsing https://www.nltk.org/modules/nltk/parse/stanford.html

[6] BURP Proxy tool https://portswigger.net/burp

[7] Recent Attacks https://www.csoonline.com/article/3237324/cyber-attacks-espionage/what-is-a-cyber-attack-recent-examples-show-disturbing-trends.html

[8] Regression Analysis http://cs229.stanford.edu/notes/cs229-notes1.pdf

[9] Graph analytics https://www.coursera.org/learn/big-data-graph-analytics/home/welcome

[10] Neo4j https://neo4j.com/developer/get-started

[11] Web data with python https://www.coursera.org/learn/python-network-data/home/welcome

[12] Using Centrality Measures to Identify Key Members of an Innovation Collaboration Network

[13] HERCULE: Attack Story Reconstruction via Community Discovery on Correlated Log Graph https://www.cs.purdue.edu/homes/dxu/pubs/HERCULE.pdf

[14] https://www.acunetix.com/blog/articles/scanning-vs-pen-testing/

[15] Performance Evaluation of Web Application Security Scanners for Prevention and Protection against Vulnerabilities https://pdfs.semanticscholar.org/5c56/cf1cd211c2810f8217585123c4b99ba5b15a.

[16] https://www.linkedin.com/pulse/regression-analysis-how-do-i-interpret-r-squared-assess-gaurhari-dass

[17] http://statweb.stanford.edu/ tibs/sta305files/Rudyregularization.pdf