Application Profiling using Statistical Techniques

From Application Security Perspective

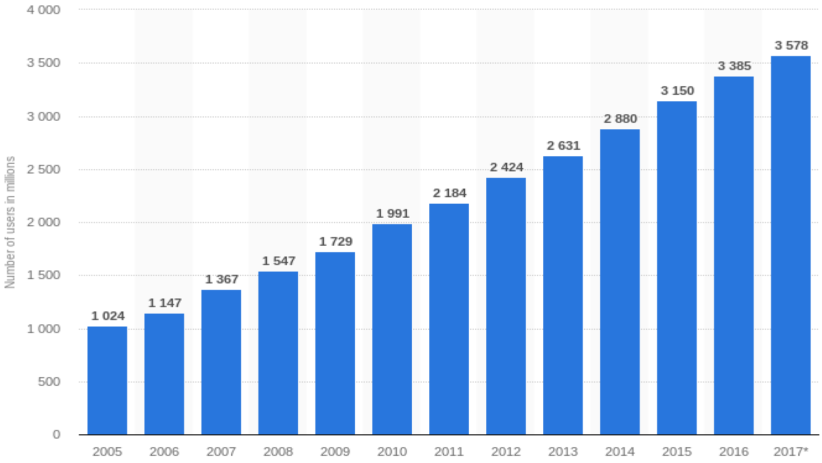
Rushita Thakkar(1401004) Ishika Agarwal(1401069)

School Of Engineering and Applied Science Ahmedabad University

7/5/2018

Outline

Global Internet Users



Why Application Security is important?

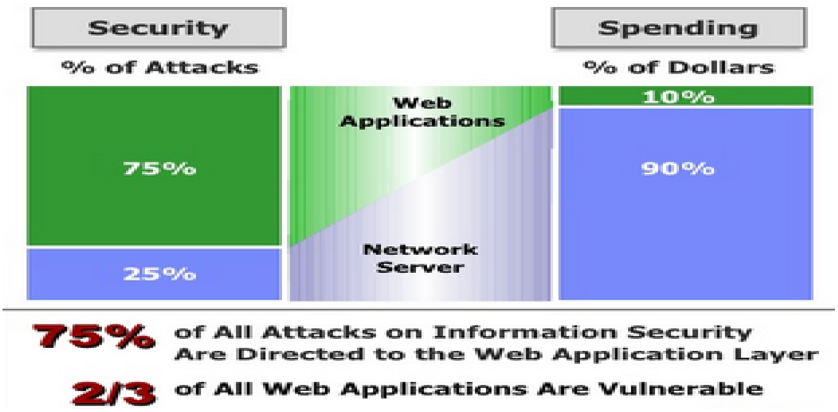
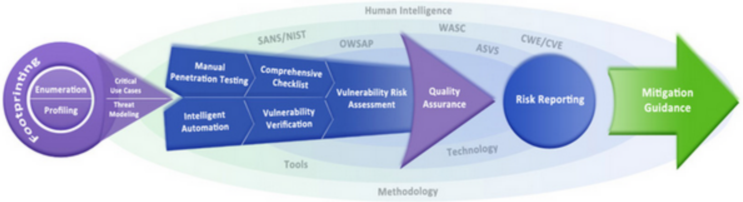


Figure: Src: Blueinfy Solutions

Why Manual Penetration Security is important?

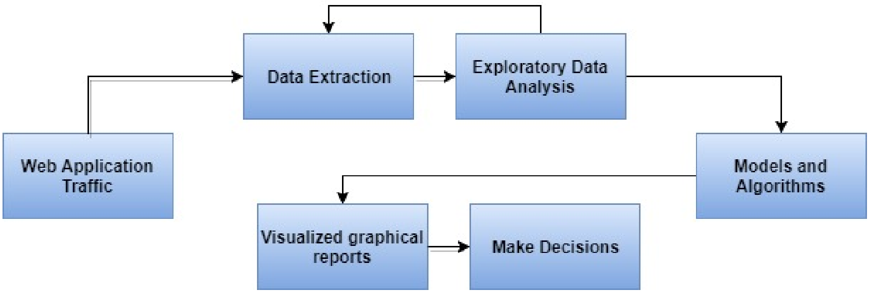
- 90 percent of applications have one or more types of vulnerabilities that pose major threats from attackers who may exploit them to breach their security. Since automated scanning proves inadequate in uncovering all exploitable vulnerabilities, Blueinfy solutions' Application Penetration Testing Service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



Project Description

- 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
- Assigning basic scores to the web pages based on connectivity analysis, words analysis and other HTTP parameters. Training our model using regression analysis, by learning importance of different features on final scoring

Project Flow



Extracting Data

- 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.
- There is list of all the requests in XML form passed as raw data to our model.

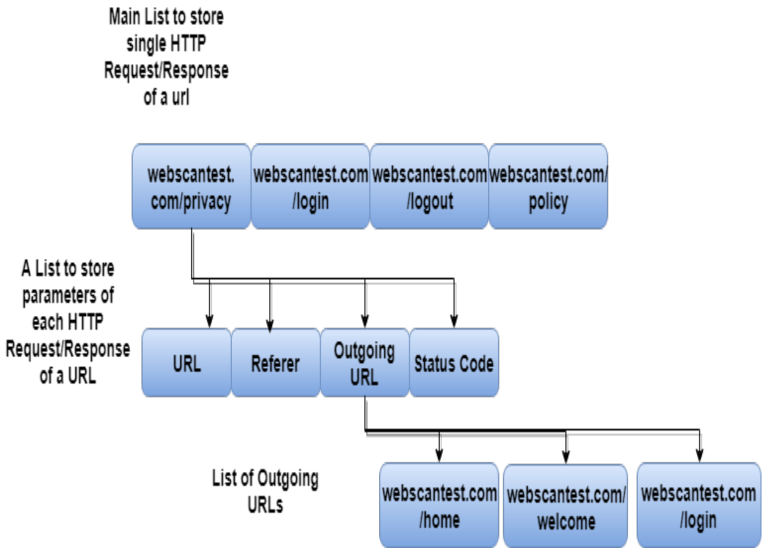
Parameters for HTTP Request

- Request Method (GET, POST, PUT, PATCH)
- URL (webscantest.com/privacy.php)
- Host (www.webscantest.com)
- Referer - identifies the path of the page in the website that is linked to the resource being requested.
- Sensitive Parameters in the Request Data For e.g. $username = adam\&password = 12345$

Parameters for HTTP Response

- Response Status - The first digit of the status code specifies one of five standard classes of responses (1xx, 2xx, 3xx, 4xx, 5xx)
- Location - Represents the resource the page will be redirected to.
- List of Outgoing URLs
- Check if any third party scripts are used
- Content-Length

Data Model


Main List to store single HTTP Request/Response of a url
webscantest.com/privacy
webscantest.com/login
webscantest.com/logout
webscantest.com/policy
A List to store parameters of each HTTP Request/Response of a URL
URL
Referer
Outgoing URL
Status Code
List of Outgoing URLs
webscantest.com/home
webscantest.com/welcome
webscantest.com/login

Connectivity

- Extracting all URLs from: -
  - $< a >$ "href"
  - $< form >$ "action"
  - <script> "src"
  - <meta> "URL"
  - <link> "href"
- For cases where the application navigates to another URLs using window.location etc. (using JavaScript) or through redirects (using 300 family response codes and Location response header) - Made nodes and edges using Referer request header and Request URL.

Regression Analysis

Various Parameters had been considered for scoring:

- Indegree Centrality
- Outdegree Centrality
- PageRank Score
- Betweeness Centrality
- Closeness Centrality
- No of Payment Related Words(Count)
- No of Session Related Words(Count)
- Method used by the page(1 if method is POST, 0 otherwise)
- Does the page have third party connection(Boolean)

Manual Scoring by Experienced Application Penetration Tester

- Senior Security Consultant from Blueinfy Solutions was asked to give a Manual Score to all the web pages of several websites.
- A security Score was given to the websites on the scale of 1 to 10 based on:
- How connected it is to other important pages
- Is there a query string in request data?
- Is there a query string in URL(XSS vulnerability)
- Status Code for checking redirections
- Request Method (GET/POST/PUT/PATCH)
- Payment related parameters or not?
- Session related parameters or not? (login/logout)

Regression Analysis

- Consider the equation: where f_θ is the final formula score $x_1, x_2, x_3, x_4....$ are the values for the parameters.

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

- The goal here is to minimize the mean squared error and weights are calculated via Gradient Descent Algorithm
- The final formula for finding out the weights is given by:

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

Training Data Websites

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)

Training Data Distribution

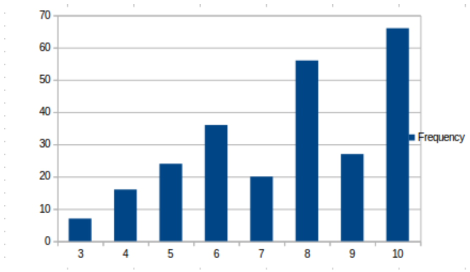


Figure: Frequencies of different Scores

Pre-processing the features

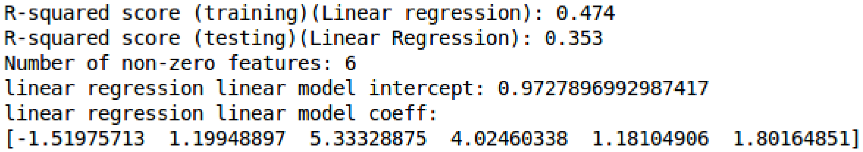
- Centrality Scores lie between 0 and 1 and they sum to 1 for a particular website.
- Payment Words and Session related words is a count.
- Method and checking weather there is a third party connection or not is a Boolean Value.
- This difference in scale would lead to unexpected outcomes in the coefficients for the regression line and that is why this pre-processing is done.

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

Correlation of different features with Manual Score

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**

Results



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

Results

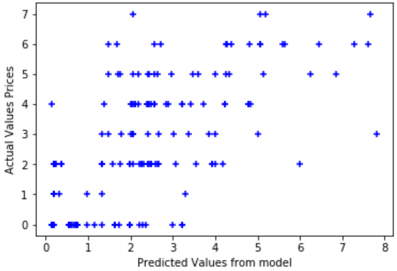


Figure: Graph Showing correlation between Manual Score and Predicted Value for Training Data

Results

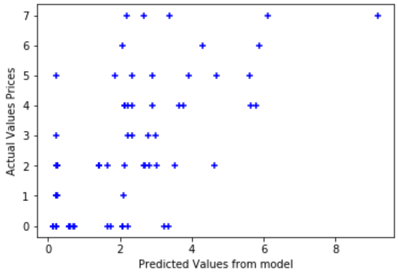
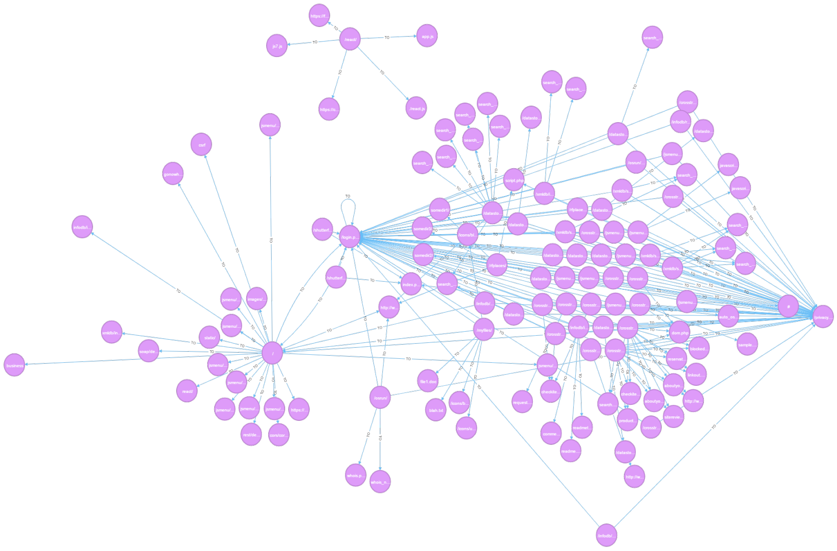


Figure: Graph Showing correlation between Manual Score and Predicted Value for Testing Data

Neo4j Graph

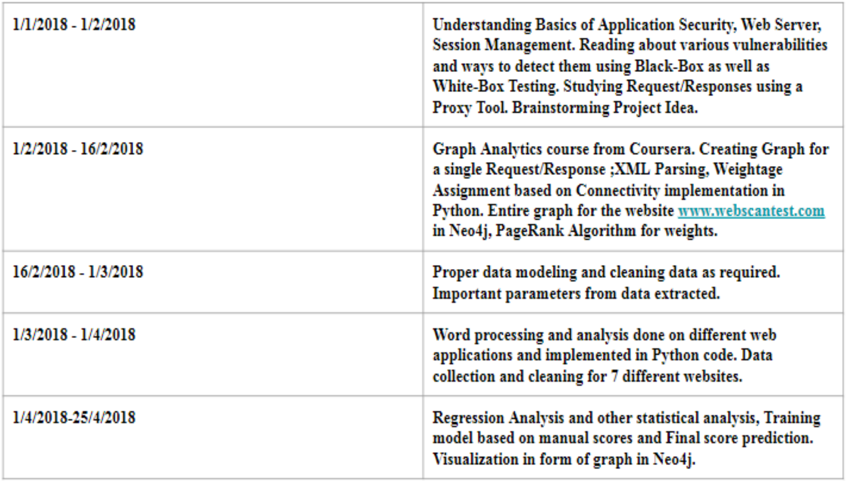


/react/
https://f...
js7.js
app.js
https://c...
/react.js
jsmenu/...
csrf
gonowh...
infodb/i...
/shutterf...
/login.p...
index.p...
http://w...
/
xmldb/in...
static/
soap/de...
business
react/
images/...
rest/de...
cors/cor...
https://...
/osrun/
whois.p...
whois_n...
file1.doc
blah.txt
/myfiles/
script.php
somedir1/
somedir2/
somedir3/
/rfplaces/
dom.php
blocked...
reservat...
linkout...
aboutyo...
checkite...
product...
sitereviе...
readme...
comme...
request...
sample...
auto_os...
#
/privacy...
/infodb/...
TO

Conclusion

- 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
- A General trend of linear relationship with particular feature is not there. In fact, result is a combination of different features.
- 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.

Timeline



1/1/2018 - 1/2/2018	Understanding Basics of Application Security, Web Server, Session Management. Reading about various vulnerabilities and ways to detect them using Black-Box as well as White-Box Testing. Studying Request/Responses using a Proxy Tool. Brainstorming Project Idea.
1/2/2018 - 16/2/2018	Graph Analytics course from Coursera. Creating Graph for a single Request/Response ;XML Parsing, Weightage Assignment based on Connectivity implementation in Python. Entire graph for the website www.webscantest.com in Neo4j, PageRank Algorithm for weights.
16/2/2018 - 1/3/2018	Proper data modeling and cleaning data as required. Important parameters from data extracted.
1/3/2018 - 1/4/2018	Word processing and analysis done on different web applications and implemented in Python code. Data collection and cleaning for 7 different websites.
1/4/2018-25/4/2018	Regression Analysis and other statistical analysis, Training model based on manual scores and Final score prediction. Visualization in form of graph in Neo4j.

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.
- Implementing UI for penetration testers to use our technique in real testing with justification of our scores.

References

[1] http://snap.stanford.edu/class/cs224w-2012/projects/cs224w-043-final.pdf
[2] HTTP Request and Response -
https://www.webnots.com/what-is-http/.
[3] Connectivity and Pagerank algorithm -
https://www.coursera.org/learn/big-data-graph-analytics/home/week/3.
[4] Application Penetration Testing:
https://www.owasp.org/index.php
[5] NLTK Parsing
https://www.nltk.org/modules/nltk/parse/stanford.html
Github link
https://github.com/RushitaThakkar/FinalYearProject

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 final year project mentors from Blueinfy Solutions whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project.