Classification of SQL injection, XSS and Path Traversal for Web Application Attack Detection

Ei Ei Han, Thae Nu Phyu
eieihan.ucsy@gmail.com, drthairnu@gmail.com
University of Computer Studies, Yangon

Abstract

Web application attack detection is one of the popular research areas during these years. SQL injection, XSS and path traversal attacks are the most commonly occurred types of web application attacks. The proposed system effectively classifies three attacks by random forest algorithm to ensure reasonable accuracy. Request length module is computed based on the certain length of the URL to analyze each record as normal or attack. Regular pattern analysis is emphasized on the content of URL and other features to analyze the certain attack patterns. ECML/PKDD standard web attack dataset is used in this system. Combination of random forest algorithm with request length and regex pattern analysis is proposed to outperform the accuracy.

1. Introduction

Web applications are becoming increasingly popular and complex in all sorts of environments, ranging from e-commerce applications to banking. The security of web applications has become increasingly important and a secure web environment has become a high priority for e-business communities. They are subject to all sorts of attacks. In today’s times, the most critical issue for any web application is security. Web servers and web-based applications are popular attack targets. To detect web-based attacks, intrusion detection systems are configured with a number of signatures that support the detection of known attacks.

There are two fundamentally different attack detection methods – rule-based detection (static rules) and anomaly-based detection (dynamic rules). Web server log analysis is a rule-based detection mode which concentrates on web attacks that are visible in default web server log files like Apache or IIS. This system combines traditional web usage mining system with security analysis. So, usage patterns of normal users and attack patterns of malicious users can be determined by this system.

Security for web application is necessary and it will be effective to study and analyze how malicious patterns occur in the web server log. If attacks occur, it is needed to analyze the attack patterns and certain features of attacks. Web based attacks need to be detected and types of attacks need to be classified.

This system specifies regular expression patterns for each attack type based on the web server log files with attacks. Random forest algorithm is an effective algorithm for attack classification. The proposed system intends to outperform accuracy of the random forest algorithm with request length module and regular expressions for attack patterns on the standard dataset ECML/PKDD. This is the name for ‘European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases’.

2. Related Works

Roger Meyer explains how to detect the most critical web application security flaws. The paper name is “Detecting Attacks on Web Applications from Log Files”, from SANS Institute 2008. This system explains how to detect the most critical web application security flaws. In this paper, various popular attack patterns are analyzed by their corresponding regex patterns. Rule-based attack detection technique is used in this paper [10].

Talasila Vamsidhar, Reddyboina Ashok and Rayala Venkat presented about the system “Intrusion Detection System For Web Applications With Attack Classification” [12]. In this system, web based attacks are detected by three modules: Request Length, Input Validation and Anomaly Detection. The authors prepare the web log dataset by themselves. Comparison results of three modules on each attack type in the dataset are shown is the paper.

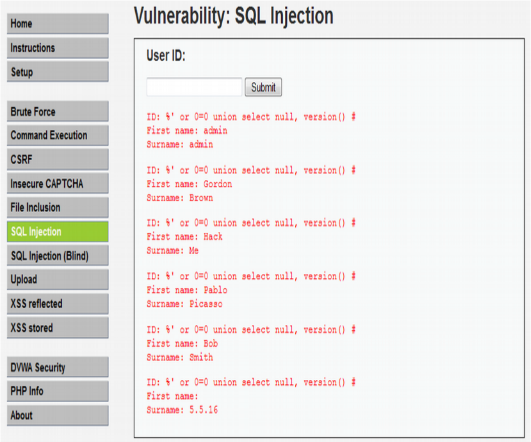
3. Web Application Attacks

Malicious users try to attack a web site or web server by using various attack patterns. Web application attacks are occurred by performing web application queries. They take the form of well-defined strings and parameters. These are recorded in the web server log file. By analyzing each record of server log file, malicious patterns can be detected. DVWA web server is tested for the specification of Regular Expression Patterns for the types of attacks.

SQL injection, XSS and path traversal attacks are contained in OWASP Top Ten Attacks. Because of limitation of testing with ECML/PKDD Dataset and DVWA web server. This system can handle on these three types of attacks.

3.1. Damn Vulnerable Web Application

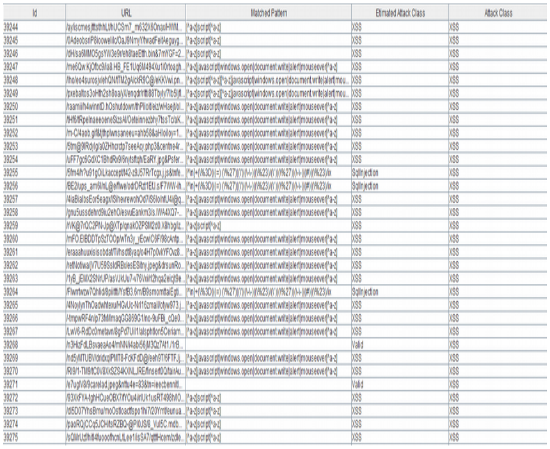
One of the popular web application attack tools is DVWA. Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications. It will be used for launching web application attacks and logging them. With this tool, popular web based attacks can be created and stored in the database.



```
ID: %' or 0=0 union select null, version() #
First name: admin
Surname: admin

ID: %' or 0=0 union select null, version() #
First name: Gordon
Surname: Brown

ID: %' or 0=0 union select null, version() #
First name: Hack
Surname: Me

ID: %' or 0=0 union select null, version() #
First name: Pablo
Surname: Picasso

ID: %' or 0=0 union select null, version() #
First name: Bob
Surname: Smith

ID: %' or 0=0 union select null, version() #
First name:
Surname: 5.5.16
```

Figure 1: Damn Vulnerable Web Server

Figure 1 shows an example of SQL injection attack with DVWA web server. By this web server, an attacker can insert certain attack patterns and click submit button. The web access log regarding this attack pattern can be analyzed. These patterns for three types of attacks are tested on ECML/PKDD dataset and the testing is shown as Figure 2.



Id	URL	Matched Pattern	Etimated Attack Class	Attack Class
39244	/ayiiscmesjttsthhLt/hUCSm7_m632X6Onax/HWM...	[^a-z]script[^a-z]	XSS	XSS
39245	/0AdeobsriP8ioowelilc/OaJ9NmyY/twadFeitAeguyg...	[^a-z]script[^a-z]	XSS	XSS
39246	/dH/sa6MMO5gsYW3e9r/eh8taeEtth.bin&7mYGF=2...	[^a-z]script[^a-z]	XSS	XSS
39247	/me6Qw.KjOfbc9/la8.HB_FE1Uq6M494Xu1/0rtoagh...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39248	/lho/eo4surosjv/ehQN/tTM2gA/ckR9C@/eKKVwi.pn...	[^a-z]script[^a-z]|[^a-z]javascript|windows.open|document.write|alert|mou...	XSS	XSS
39249	/pxebaltos3oHth2sh8oa/yX/enqdr/rltti88Tbyly/7Ib5I/jfl...	[^a-z]script[^a-z]|[^a-z]javascript|windows.open|document.write|alert|mou...	XSS	XSS
39250	/raamii/h4winntD.hOshutdown/thPlioit/eiz/wHaej/iol...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39251	/tHf6/tRpelnaeeoeneSizsAl/Oeteinnezbhy7tssTc/aK...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39252	/m-C/4aob.gif&tjthplwnsaneeu=ahb58&aHloiloy=1...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39253	/5tm@9IRdy/g/a0ZHhcrdtp7seeAcy.php3&centne4r...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39254	/uFF7gc6GdXC1BhdRx9/6nytsftqh/EaRY.jpg&Psfer...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39255	/5fm4/h?u91gOiLkaccept/t42-z9J57RrTcgx.j.js&tnfe...	[^\n]+(\%3D)|(=) (\%27)|(\')|(\-\-)|(\%23)|(\')|(\%27)|(\-\-)|(#)|(\%23)/ix	SqlInjection	XSS
39256	/BE2/ups_am6l/nL@eftwe/odrDRzt1EtJ.s/F7WW-ih...	[^\n]+(\%3D)|(=) (\%27)|(\')|(\-\-)|(\%23)|(\')|(\%27)|(\-\-)|(#)|(\%23)/ix	SqlInjection	XSS
39257	/4iaBialbsEor5eagx/lSihevrewohOd7iS6lo/ntU4/@q...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39258	/gnu5ussdehrd9iu2ehO/esvuEankm3/s.IWA4XQ7-...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39259	/tVK@7rQC2PN-Jp@XTp/qnakOZPSM2d0.X8hbg/iz...	[^a-z]script[^a-z]	XSS	XSS
39260	/mFO.EIBDDTpSzTOOp/wTn3y_yEcwiC6F/98cAntp...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39261	/eraaahuuxisisobdat/Ti/hsdt8yaq/o4H7p0vKYFOvz8...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39262	/retNotiwa/jV7U59SsIdRBx/esESitny.jpeg&drsunRo...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39263	/1yB_jEMX2SNrUP/asVJkUu7-v76Vxi/rt2hqa2eicjt9e...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39264	/Ftwrrtwzw7Qhlid/8pilttti/YsfB3.6m/B9smomttaiEgti...	[^\n]+(\%3D)|(=) (\%27)|(\')|(\-\-)|(\%23)|(\')|(\%27)|(\-\-)|(#)|(\%23)/ix	SqlInjection	XSS
39265	/4NoylynThOadwhtexu/HGvUc-Nrl16zmail/otyw973.j...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39266	/-tmpwRF4n/p73tMmaqGG869G1/no-9uFBi_cQe0...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39267	/LwV6-RdDc0metavn/8gPd7Uil1/alsphtlon5Ceriam...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39268	/n3HizFdLBsvaeaAo4/mNN/i4abi56jM3Qz7At1./1rB...		Valid	XSS
39269	/nd5yMTUBV/dridxqlPMT8-FcKFdD@/eeh9T/6FTFJj...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39270	/RI9/1-TM9/tC0V8XkSZS4KXNLJRE/finsert0Q/tairAu...	[^a-z]javascript|windows.open|document.write|alert|mouseover[^a-z]	XSS	XSS
39271	/e7ugIV8/9carelad.jpeg&nttu4e=83&tn=ieecbennitl...		Valid	XSS
39272	/93XkFYA-tghHCueOBX7/fYOu4i/rlUk1usRT498h/IO...	[^a-z]script[^a-z]	XSS	XSS
39273	/dI5D07YhsBmu/moOstloadtspo1hi7/20Ymt/eunua...	[^a-z]script[^a-z]	XSS	XSS
39274	/paoRQjCCq5JCH/tsRZBQ-@PI0JS/8_Vul5C.mdb...	[^a-z]script[^a-z]	XSS	XSS
39275	/sQMrUzf/hitl4fuooofhcnLtLee1/isSA7/qtttHcem/zdle...	[^a-z]script[^a-z]	XSS	XSS

Figure 2: Regex Pattern Analysis on ECML/PKDD Dataset by Attack Patterns Obtained from DVWA

3.2. Web Usage Mining

Web usage mining is the process of extracting useful information by analyzing web usage data from server logs. It is defined as an application of data mining techniques on the navigational traces of the users to extract knowledge about their preferences and behavior. Web usage mining involves three major phases namely, pre-processing, pattern discovery and pattern analysis. Some of the techniques used in Pattern discovery are Association rules, Classification, Clustering etc. Pattern Analysis filters out uninteresting rules or patterns found in the pattern discovery phase.

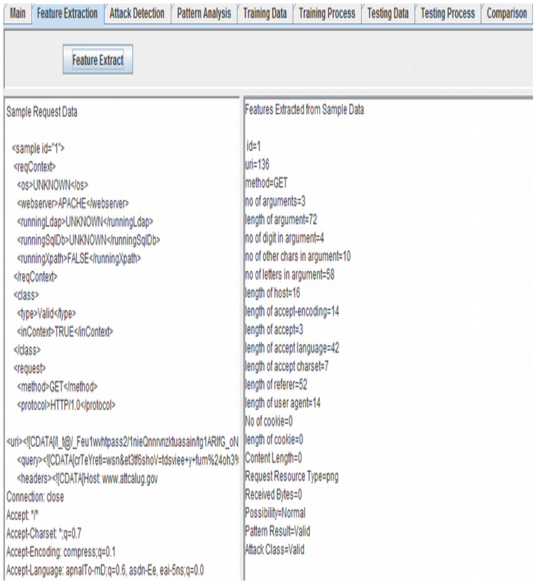
In the web usage mining system, analysis on web server log with attack features become a problem area. This system differentiates normal access patterns from malicious access patterns. It can detect how malicious users try to attack the web site. The system can know which pages or links are most accessed and are tried by malicious users. It also describes successful attacked (attack gained) web pages and links. This system will be effective for the security of web application system and analysis on web server log.

4. Datasets for the System

To implement this system, I have analyzed on three different datasets, namely, (1) ECML/PKDD, (2) CSIC and (3) Web attack testing log from experts. In the CSIC dataset, there are only two class labels of normal and attack. So, it is needed to classify for the SQL injection, XSS and Path Traversal attacks. Third dataset has web server log file nature but it cannot produce attack class labels. Therefore, it is difficult to measure accuracy for this dataset.

In this system, I have tested on ECML/PKDD dataset. There are 50116 samples in this dataset. Because this system detects three types of attacks namely SQL injection, XSS and path traversal. Other types of attacks in the dataset are removed for the efficient classification. The filtered dataset has about 42128 log records.

ECML/PKDD Dataset which is in XML format is used as input to the system. The attack patterns may include some special and encoding characters. So, URL decoding is needed for preprocessing. Features in the dataset are extracted to get specific features necessary for attack detection. Figure 3 illustrates how to extract sample one record of ECML/PKDD.xml file to get certain features. In this way, all 42128 records are extracted and these are organized as .csv file for further steps of classification.



Sample Request Data

```
  <sample id="1">
   <reqContext>
    <os>UNKNOWN</os>
    <webserver>APACHE</webserver>
    <runningLdap>UNKNOWN</runningLdap>
    <runningSqlDb>UNKNOWN</runningSqlDb>
    <runningXpath>FALSE</runningXpath>
   </reqContext>
   <class>
    <type>Valid</type>
    <inContext>TRUE</inContext>
   </class>
   <request>
    <method>GET</method>
    <protocol>HTTP/1.0</protocol>

<uri><![CDATA[/l_t@/_Feu1wvhtpass2/1nieQnnrvnzktuasain/tg1ARlfG_oN
    <query><![CDATA[crTeYreti=wsn&et3tf6shoV=tdsviee+y+fum%24oh3%
    <headers><![CDATA[Host: www.attcalug.gov
Connection: close
Accept: */*
Accept-Charset: *;q=0.7
Accept-Encoding: compress;q=0.1
Accept-Language: apnalTo-mD;q=0.6, asdn-Ee, eai-5ns;q=0.0
```

Features Extracted from Sample Data

```
 id=1
uri=136
method=GET
no of arguments=3
length of argument=72
no of digit in argument=4
no of other chars in argument=10
no of letters in argument=58
length of host=16
length of accept-encoding=14
length of accept=3
length of accept language=42
length of accept charset=7
length of referer=52
length of user agent=14
No of cookie=0
length of cookie=0
Content Length=0
Request Resource Type=png
Received Bytes=0
Possibility=Normal
Pattern Result=Valid
Attack Class=Valid
```

Figure 3: Feature Extraction from Dataset

4.1. Request Length Module

Request Length Module in this system can be computed as follows. μ = average length or mean of n requests and is calculated by equation (1). L_1, L_2, L_3,…….,L_n where L_i =length of the received requests of i and num_valid = number of valid records. σ =variance of requests is calculated by equation (2). The possibility ρ of a request will be calculated by equation (3). If the possibility value ρ is higher than a threshold, the request will be considered as an anomaly request. This method can detect attacks like Directory Traversal and Buffer Overflow.

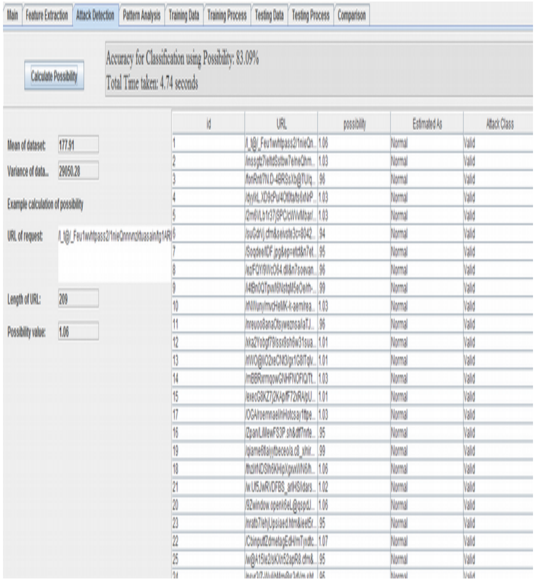
Equations used for computing request length module are as follows:

$$\mu = \frac{\sum L_i}{num_{valid}} \quad (1)$$

$$\sigma = \frac{\sum (L_i - \mu)^2}{num_{valid}} \quad (2)$$

$$\rho = \frac{\sigma}{(L_i - \mu)^2} \quad (3)$$

Figure 4 illustrates request length method calculation. Mean of dataset and variance of dataset is the same for all records. Based on the URL length of each record, the possibility value is calculated and if this value is more than the threshold, that record is estimated as Attack. Otherwise, it is estimated as normal record.



Accuracy for Classification using Possibility: 83.09%
Total Time taken: 4.74 seconds

Mean of dataset: 177.91
Variance of data... 29050.28
Example calculation of possibility
URL of request: /l_t@/_Feu1whtpass2/1nieQnnnvnzltuasain/tg1AR
Length of URL: 209
Possibility value: 1.06

id	URL	possibility	Estimated As	Attack Class
1	/l_t@/_Feu1whtpass2/1nieQn...	1.06	Normal	Valid
2	/nssgtz7ieltdSstbw7eheQhm...	1.03	Normal	Valid
3	/fonRnt/7N.D-4BRSsXb@TUq...	.96	Normal	Valid
4	/dyylkL.XD9cPu/4Of0ta/ts6xNrP...	1.03	Normal	Valid
5	/2m6VLb1r37jSPC/cWWvMbar/...	1.03	Normal	Valid
6	/suGdrVj.cfm&selvste3c=8042...	.94	Normal	Valid
7	/SoqdeeIDF.jpg&ep=etdt&n7et...	.95	Normal	Valid
8	/ezFQY/9WcO64.dll&n7soevan...	.96	Normal	Valid
9	/4tBn0QTpvxl6NstqM5eOelrh-...	.99	Normal	Valid
10	/rNWuny/mvcHeMK-k-aem/rea...	1.03	Normal	Valid
11	/nreuoo8anaOtsyweznsa/aTJ...	.96	Normal	Valid
12	/xka2Yobgf79/ssx9sh6w31sua...	1.01	Normal	Valid
13	/rWO@lIO2xeCN3/gx1G8ITqlv...	1.01	Normal	Valid
14	/mBBRxmqowGNHFNOFIQ/Tt...	1.03	Normal	Valid
15	/execG8KZ7j2KApffF72rRAlpU...	1.01	Normal	Valid
17	/OGAlroemnaelInHotcsay1ttpe...	1.03	Normal	Valid
16	/ZpanLiMewFS3P.sh&dff7nfe...	.95	Normal	Valid
19	/qlame6llajytbeceola.c8_xlnr...	.99	Normal	Valid
18	/thzl/tNDSlh6KHipXpwWN6/h...	1.06	Normal	Valid
21	/w.Uf5JwRVDFBS_arlHS/dars...	1.02	Normal	Valid
20	/9Zwindow.openk6eL@qspdJ...	1.06	Normal	Valid
23	/nratb7/ehjUpsiaed.htm&leet5r...	.95	Normal	Valid
22	/CbinputfZdmetagEdH/mTyxdtc...	1.07	Normal	Valid
25	/w@A15ie2/sKXn52apR8.cfm&...	.95	Normal	Valid
24	[illegible]	.95	Normal	Valid

Figure 4: Request Length Method

4.2. Regular Expression Pattern Analysis

Regular expressions enable a powerful, flexible, and efficient text processing. The goal of a regular expression is to match a certain expression within a lump of text. A regular expression pattern is usually enclosed within slashes ('/'). This system can analyze how attack log file occurred by using DVWA web server. By inputting some attack patterns from input box and by POST method, the system can analyze how certain types of attacks occurred in web server log file [10].

5. Random Forest Algorithm

This algorithm consists of collection of decision trees and majority vote on these trees is used as the final result. It runs efficiently on large data and provides high accuracy. The algorithm can provide effective methods for estimating missing data. The analyst does not need to do any variable selection or data reduction. Data do not need to be rescaled, transformed, or modified. Growing a large number of random forest trees does not create a risk of over fitting.

Processing steps in the random forest algorithm are as follows:

1. Choose T- number of trees to grow.
2. Choose m- number of variables used to split each node m<<M, where M is the number of input variables. m hold constant while growing the forest.
3. Grow T trees. when growing each tree do the following:
   a. Construct a bootstrap sample of size n sampled from S_n with replacement and grow a tree from this bootstrap sample.
   b. When growing a tree at each node select m variables at random and use them to find the best split.
   c. Grow the tree to a maximal extent. There is no pruning.
4. To classify point X collect votes from every tree in the forest and then use majority voting to decide on the class label.

5.1. Classification Features to detect Web Application Attacks

Request general features, Request content features, Response features and Request history features are used for the detection of web application attacks. Request general features include request length, request method (GET, POST, etc.), request resource type, number of parameters and number of arguments, etc. Request content features include SQL command tricks, Directory Traversal tricks, Script injection, etc. Response features include response code, response time, etc. Request history features include analyzing malicious users' previous access paths.

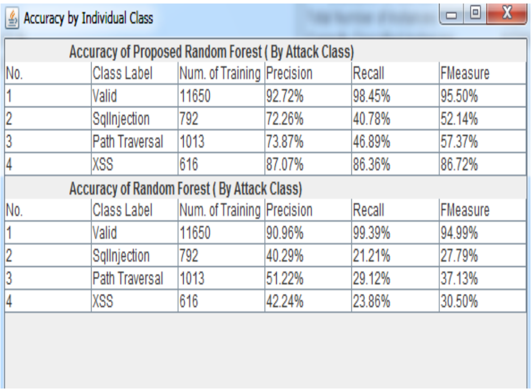
Features used in this system based on ECML/PKDD dataset are listed from 1 to 19. Because the proposed system first computes request length method, the result is as attack or normal. This possibility result is used as an additional feature to the random forest algorithm listed at 20. Results of regex pattern analysis are as XSS, SQL Injection, or Path Traversal. This pattern result is also used as an additional feature to the random forest algorithm listed at 21.

The intention of this system is to outperform the accuracy of mining algorithm Random Forest by combining request length and regex pattern analysis for attack classification. The use of regex pattern analysis is to secure attack detection system based on the certain attack patterns.

1. URI
2. Method identifier
3. Number of arguments
4. Length of the arguments
5. Number of digits in the arguments
6. Number of other char in the arguments
7. Number of letters in the arguments
8. Length of the Host
9. Length of the header "Accept-Encoding"
10. Length of the header "Accept"
11. Length of the header "Accept-Language"
12. Length of the header "Accept-Charset"
13. Length of the header "Referer"
14. Length of the header "User-Agent"
15. Number of cookies
16. Length of the header "Cookie"
17. Content Length
18. Request Resource Type
19. Received Bytes
20. Possibility
21. Pattern Result

6. Experimental Results

Performance of the system is measured by Precision, Recall and F-Measure. Figure 5 shows the results of these measures on each attack type and valid records.



Accuracy of Proposed Random Forest (By Attack Class)

No.	Class Label	Num. of Training	Precision	Recall	FMeasure
1	Valid	11650	92.72%	98.45%	95.50%
2	SqlInjection	792	72.26%	40.78%	52.14%
3	Path Traversal	1013	73.87%	46.89%	57.37%
4	XSS	616	87.07%	86.36%	86.72%

Accuracy of Random Forest (By Attack Class)

No.	Class Label	Num. of Training	Precision	Recall	FMeasure
1	Valid	11650	90.96%	99.39%	94.99%
2	SqlInjection	792	40.29%	21.21%	27.79%
3	Path Traversal	1013	51.22%	29.12%	37.13%
4	XSS	616	42.24%	23.86%	30.50%

Figure5: Accuracy Measures of the System

Figure 6 and Table 1 shows the percent of accuracy based on the number of trees by random forest and the proposed system. I have tested on Core i3 processor with 2 GB memory. When I set the tree number more than 30, there is out of memory error. Because the percent difference is not distinct, both chart and table are presented in this paper.

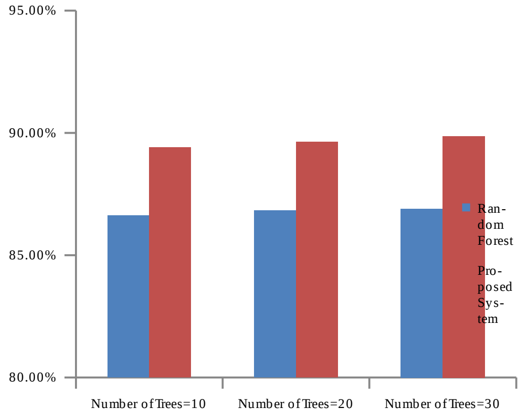


Figure 6: Percent of Accuracy on the Number of Trees

Table 1: Percent of Accuracy on the Number of Trees

	Number of Trees=10	Number of Trees=20	Number of Trees=30
Random Forest	86.62%	86.82%	86.90%
Proposed System	91.12%	89.64%	89.87%

Figure 7 shows the accuracy percentage of four methods on ECML/PKDD dataset. The four methods are Regex Pattern Analysis, Request Length Module, Random Forest and the Proposed System. The proposed system is the combination of request length, regex pattern analysis and random forest algorithm. In Figure 7, accuracy of regex pattern analysis is 84.21%, accuracy of request length method is 83.09%, accuracy of random forest is 86.62%, and accuracy of the proposed system is 91.12%.

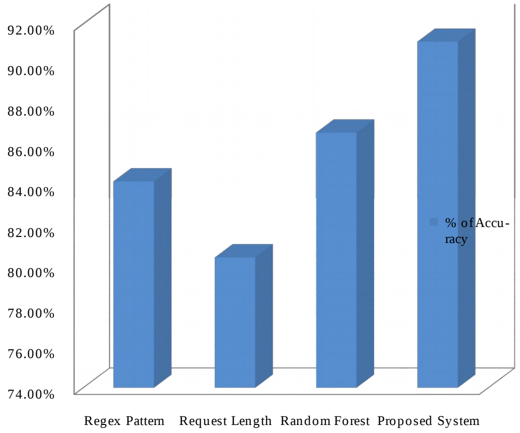


Figure 7: Accuracy Percent of Four Methods on ECML/PKDD Dataset

7. Conclusion

This system presents about analyzing and classifying web application attacks. Combination of request length module, regular expression patterns and Random Forest algorithms are used in this system. SQL injection, XSS, directory traversal attacks and valid records can be classified by this system. By computing request length module, each record is computed as normal or attacks. Predefined regex pattern analysis can classify as SQL injection, XSS and path traversal attacks. The proposed system intends to outperform the accuracy for the classification of web application attacks.

References

[1] AmrutaSurana, Shyam Gupta, "An Intrusion Detection Model for Detecting Types of Attacks Using Data Mining", Department of Computer Engineering, Siddhant College of Engineering, Pune, Maharashtra, India.

[2] Christopher Kruegel, Giovanni Vigna, William Robertson, "A multi-model approach to the detection of web-based attacks", Reliable Software Group, University of California, Santa Barbara, USA, 2005.

[3] F.Livingston, "Implementation of Breiman's Random Foreste Machine Learning Algorithm", ECE591Q Machine Learning Journal Paper, 2A0.

[4] J.Su and H.Zhang, " A Fast Decision Tree Learning Algorithm", Proceedings of the 21't national conference on Artificial Intelligence, 2006.

[5] L.Breiman, "Random Forest", January 2001.

[6] L. Breiman, "Random Forests", Machine Learning 45(1):5-32, 2001.

[7] L.Breiman, J.Friedman, R.Olshen and C.Stone. "Classification and Regression Trees", Wadsworth International Group, 1984.

[8] RistoPantev, "Analysis and Classification of Current Trends in Malicious HTTP Traffic", College of Engineering and Mineral Resources at West Virginia University.

[9] R. Kohavi, " A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection", International Joint Conference on Artificial Intelligence (IJCAI), vol.12, p.1137-1143, 1995.

[10] Roger Meyer, Carlos Cid, "Detecting Attacks on Web Applications from Log Files", SANS Institute 2008.

[11] S.Mukkamala and A. H. Sung, "Feature Ranking and Selection for Intrusion Detection System Using Support Vector Machines", IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.11, November 2007.

[12] Talasila Vamsidhar, Reddyboina Ashok and RayalaVenkat ,"Intrusion Detection System For Web Applications With Attack Classification", Journal of Global Research in Computer Science, 2012 .

[13] T. Lappas and K. Pelechrinis, "Data Mining Techniques for (Network) Intrusion Detection Systems", Department of Computer Science and Engineering UC Riverside, Riverside CA 92521.

[14] W.T.D.Aung, "Web Page Classification by Using Random Forest Classifier", Thesis dissertation, October 2010.