International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 – 6435(Online), Volume 1, Issue 2, November 2013)

A Plug in Implementation for Phishing Attacks Using C4.5 Algorithm

Akanksha Upadhyaya[1], Jitendra Dangra[2], Dr. M. K Rawat[3]
[1,2,3]*Computer Science and Engineering , LNCT, Indore, India*

***Abstract*--Rapid increase in the size of web users. Users enter sensitive information such as passwords, their personal and professional information into scam web sites. Phishing is the criminally fraudulent process of attempting to acquire sensitive information such as usernames, passwords and credit card details, for some illegitimate purpose. Such scam sites cause substantial damages to individuals and corporations. These attacks can be analyzed through this work, and a plug in is designed which provide security from the fake websites . This work is improved by using decision tree c4.5 over id3 and a comparison is drawn.**

The objective of this work is to optimize the work ,done by ID3 algorithm before, using C4.5 for designing anti phishing toolbar by designing a new toolbar with improved accuracy, search time and less memory consumption and then give comparitive results for ID3 and C4.5.

***Keywords*--ID3, C4.5, phishing**

I. Introduction

Phishing is attempting to get information (and sometimes, indirectly, money) such as usernames, passwords, and credit card details by impersonating as a trustworthy entity in an electronic communication. Communications maintaining to be from popular social web sites, auction websites, online payment processors or IT administrators are commonly used to lure the unsuspecting public. Phishing emails may contain links to web sites that are infected with malware [1]. Phishing is typically carried out by e-mail spoofing or instant messaging and it often directs users to enter details at a fake website whose look and feel are almost identical to the legitimate one. Phishing is a model of social engineering techniques used to deceive users [2]. And exploit the poor usability of current web security technologies. Attempts to deal with the rising number of reported phishing incidents include legislation, user training, public awareness and technological security measures.

In the domain of internet and web access phishing is more a kind of attack but due to this attack to a large amount of money and reputation is compromised by the victim.

This proposed study work is a study around various kinds of anti-phishing toolbars available over internet and promises to provide the security against phishing attack. In addition of that document includes different aspects of phishing attack, detection techniques, and their solutions and attack types. The proposed work is motivated by toolbars and here designed technique is for internet explorer and with the help of machine learning algorithm (decision tree).

II. Literature Survey

The initial work which is done in this area is to make people aware about phishing attacks ,by giving them training. this is done by Jason Hong, Justin Crenshawand found that after training, people stop inserting their credential information into phishing websites up to 40% but did not stop visiting them. [1]

Then a analysis is made for determining that who falls for phish more by Steve Shang, Lorrie Carnor and found that people between age group of 18-25 falls for phish more and another comparison shows that females more falls to phish as compared to males. These researches shows that training prevent less from phishing attacks. [3]

Some technical ways can be used for avoiding phishing. To design an anti phishing toolbar is one of them suggested by Amir Herzberg and Ahmad Jbara.anti phishing toolbars are the software packages which can be installed on users device and either give him warning every time the phishing URL has been visited or block that website. [4]

Now a days there were lots of anti phishing toolbars are available, but with some deficiencies, an analysis of top ten toolbars were made in and found that most of these toolbars were fully automatic, not user interactive or dedicated one and suffering from the problem of false positives and false negatives.[2]

III. Proposed Solution

As phishing causes lots of damage to the individual , hence some security measure must be there which will provide security to the individual.

There were lots of anti phishing toolbars were available in the market for this purpose but according to the research these toolbars were not providing full security, because they were having some constraints:

1. Fully automatic.
2. Not user interactive.
3. Having false positives.

To overcome these deficiencies a new toolbar must be designed, which is followed in this work.

Proposed Algorithm: this module contains algorithms and user select an algorithm for consuming phish tank database and develop a data model for navigation.

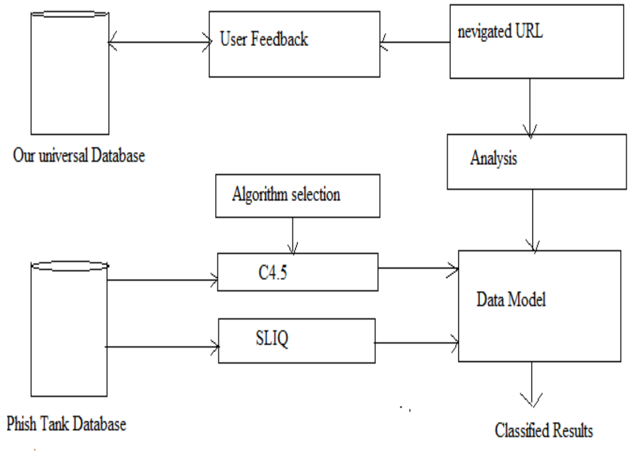


Fig 1 shows system architecture

3.1 Implemented Algorithm

This section of the document contains the algorithms available for implementation for URL classification, in addition of the algorithm here we provide brief introduction of the data used for experiments. The input data is in form of URL which is processed first than this data can be utilized for classification.

3.1.1 Data

The data available for evaluation in the format of URL which contains some prefix and suffix in addition of that in real world applications data is never found in clean format it is pre-processed to develop a data model. To pre-process given data URL is read first and braked using the symbols (.),(/),(&) and converted into a table format. In next not all URL having the same length thus some instance contains the null values in training dataset.

3.1.2 ID3 algorithm

ID3 is a simple decision tree learning algorithm developed by Ross Quinlan. The basic idea of ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node. In order to select the attribute that is most useful for classifying a given sets, we introduce a metric ---information gain.

To find an optimal way to classify a learning set, what we need to do is to minimize the questions asked (i.e. minimizing the depth of the tree). Thus, we need some function which can measure which questions provide the most balanced splitting. The information gain metric is such a function.

3.1.2.1 Entropy

In order to define information gain exactly, we require discussing entropy first. Let's assume, without loss of simplification, that the resultant decision tree classifies instances into two categories, we'll call them P (positive) and N (negative)

Given a set S, containing these positive and negative targets, the entropy of S related to this Boolean classification is:

Entropy(S) = - P (positive) log2P (positive) -P (negative) log2P (negative)

P (positive): proportion of positive examples in S

P (negative): proportion of negative examples in S

3.1.2.2 Information Gain

As we mentioned before, to minimize the decision tree depth, when we traverse the tree path, we need to select the optimal attribute for splitting the tree node, which we can easily imply that the attribute with the most entropy reduction is the best choice.

We define information gain as the expected reduction of entropy related to specified attribute when splitting a decision tree node.

The information gain, Gain(S, A) of an attribute A,

Gain(S,A)= Entropy(S) -Sum for v from 1 to n of (|Sv|/|S|) * Entropy(Sv)

We can use this notion of gain to rank attributes and to build decision trees where at each node is located the attribute with greatest gain among the attributes not yet considered in the path from the root.

The intention of this ordering is:

1. To create small decision trees so that records can be identified after only a few decision tree splitting.
2. To match a hoped for minimalism of the process of decision making

The ID3 algorithm can be summarized as follows:

1. receive all unused attributes and count their entropy concerning test samples
2. Choose attribute for which entropy is minimum (or, equivalently, information gain is maximum)
3. Make node containing that attribute

The algorithm is as follows:

```
ID3 (Examples, Target_Attribute, Attributes)
  • Create a root node for the tree
  • If all examples are positive, Return the single-node tree Root, with label = +.
  • If all examples are negative, Return the single-node tree Root, with label = -.
  • If number of predicting attributes is empty, then Return the single node tree Root, with label = most common value of the target attribute in the examples.
  • Otherwise Begin
      o A = The Attribute that best classifies examples.
      o Decision Tree attribute for Root = A.
      o For each possible value, $v_i$, of A,
          ▪ Add a new tree branch below Root, corresponding to the test A = $v_i$.
          ▪ Let Examples($v_i$) be the subset of examples that have the value $v_i$ for A
          ▪ If Examples($v_i$) is empty
              ▪ Then below this new branch add a leaf node with label = most common target value in the examples
          ▪ Else below this new branch add the subtree ID3 (Examples($v_i$), Target_Attribute, Attributes – {A})
  • End
• Return Root .
```

3.1.3 C4.5 Algorithm

C4.5 (Quinlan, 1993) is one such system that learns decision-tree classifiers, several authors have recently noted that C4.5s performance is weaker in domains with a pre-penetrance of continuous attributes than for learning tasks that have mainly discrete attributes. For example, Auer, Holte, and Maass (1995) describe Two a system that searches for good two. Level decision trees and comment:

"The accuracy of T2's trees rivalled or surpassed C4.5's on 8 of the datasets, including all but one of the datasets having only continuous attribute."

INPUT: Tentative data set D which is showed by discrete value attributes.

OUTPUT: decision tree algorithm T which is created by giving experimental dataset.

i) Create the node N;

ii) If instance is related to the same class

iii) Then return node N as leaf node and marked with CLASS C;

iv) IF attribute List is null, THEN

v) Return node N as the leaf node and signed with the most common CLASS;

vi) Selecting the attribute with highest information gain in the attribute List, and signing the test_attribute;

vii) Validation the node N as the test_attribute;

viii) FOR the well-known value of each test_attribute to divide the samples;

ix) Producing a new branch which is fit for the test_attribute = ai from node N;

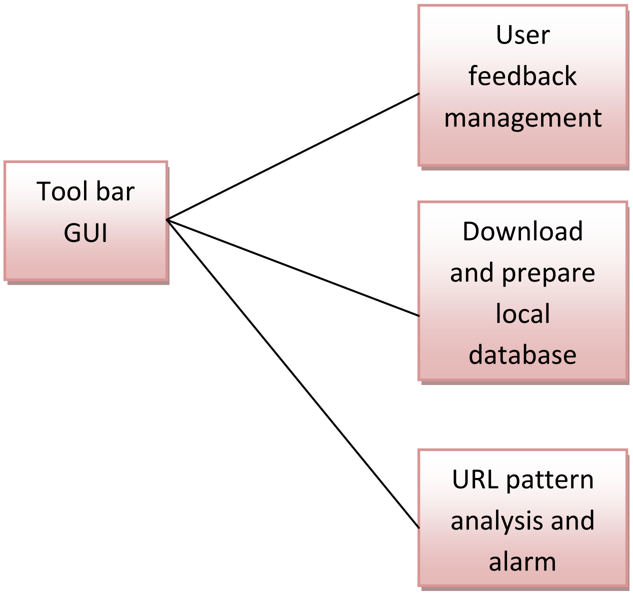
x) Suppose that Ci is the set of test_attribute=aiin the samples;

xi) IF Ci = null THEN

xii) Adding a leaf node and labeled with the most common CLASS;

xiii) ELSE we will add a leaf node return by the Generate_decision_tree.

IV. Proposed Architecture


Tool bar GUI
User feedback management
Download and prepare local database
URL pattern analysis and alarm

Above given diagram show the different level of security and its working with the URLS in this section of thesis we show how the above given model in being implemented.

Toolbar GUI: it is effort to demonstrate the user interface to show different notifications and there information generated by system. Actually it is individual software unit get the URL which is type over the internet explorer and it explore the information regarding the collected URL.

USER feedback Management: here user can add the positive or negative feedback for particular URL that is updated over a server. By which different toolbar users can access the user feedback from the server.

Prepare local database from phish tank: phish tank is a large database of phishing web sites and it contains different URLs which is found as fraud or phishing work.

URL Pattern Recognition: here a new concept implemented to find the URL patterns of phishing using a decision tree algorithm that help us to predict is URL is a phishing URL or not.

V. Conclusion

Following conclusions can be made:

1 Proposed architecture will provide an application with improved security.

2 As C4.5 algorithm will be used so the accuracy get improved over ID3.

3 The system will be more user interactive.

4 The system will provide feedback to the URL opened.

REFERENCES

[1] Amir Herzberg2 and Ahmad Jbara :Security and Identification Indicators for Browsers against Spoofing and Phishing Attacks1: Computer Science Department Bar Ilan University.(2006)

[2] Lorrie Carnor, Serge Egelman, Jason Hong ,Yue Zhang :Phinding phish-an evaluation of anti phishing toolbar: Carnege Melon University.(2006)

[3] Juan Chen,ChuanxiongGuo,"Online Detection and Prevention of Phishing Attacks",2006 - ieeexplore.ieee.org

[4] Ponnurangam Kumaraguru, Justin Cranshaw, Alessandro Acquisti, Lorrie Cranor, Jason Hong, Mary Ann Blair, Theodore Pham: School of Phish: A Real-Word Evaluation of Anti-Phishing Training March 9, 2009 CMU-CyLab-09-002.(2009)

[5] Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, and Suku Nair, A Comparison of Machine Learning Techniques for Phishing Detection, October 4-5, 2007, Pitts-burgh, PA, USA.

[6] Ian Fette, Norman Sadeh,Anthony Tomasic, "Learning to Detect Phishing Emails",2007, May 8–12, 2007, Banff, Alberta, Canada. ACM 978-1-59593-654-7/07/0005.

[7] Collin Jackson, Daniel R. Simon, Desney S. Tan, and Adam Barth,An Evaluation of Extended Validation and Picture-in-Picture Phishing Attacks,Microsoft Research, Redmond, WA,2007 – Springer, http://usablesecurity.org/papers/jackson.pdf

[8] SergeEgelman, Lorrie Faith Cranor, Jason Hong,"You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings",Copyright 2008 ACM 1-59593-178-3/07/0004.

[9] Steve Sheng, Mandy Holbrook Ponnurangam Kumaraguru. Lorrie Cranor, Julie Downs1 1Carnegie :Who Falls for Phish? A Demographic Analysis of Phishing Susceptibility and Effectiveness of Interventions Mellon University,2Indraprastha Institute of Information Technology xsheng@andrew.cmu.edu, holbrook@andrew.cmu.edu, pk@iiitd.ac.in, lorrie@cmu.edu, downs@cmu.edu.(2010)

[10] IndraneelMukhopadhyay, MohuyaChakraborty, SatyajitChakrabarti, "A Comparative Study of Related Technologies of Intrusion Detection & Prevention Systems",Journal of Information Security, 2011, 2, 28-38