I INTRODUCTION

Les pages de phishing sont l'un des problèmes majeurs de sécurité sur internet. La majorité des attaques utilisent des méthodes sophistiquées comme les fausses pages pour tromper les utilisateurs afin d'acquérir des informations sensibles.

La méthode la plus simple pour éviter la visite des sites frauduleux est l'utilisation des listes noires si ces listes sont maintenues a jour immédiatement âpres qu'un site frauduleux est créé ce qui est pratiquement impossible a mettre en place.

Dans ce chapitre nous allons étudier une méthode pratique d'anti-phishing, ce qui consiste à un système de classification automatique.

II APPROCHE

Pour l'application nous avons utilisé l'approche d'analyse d'URLs pour dire qu'un site est sans risque sans avoir a examiner son contenu, cette approche cherche a éviter de télécharger le contenu puis faire l'analyse ce qui crée une importante latence qui dérange l'utilisateur.

L'URL tout seul peut contenir un lot important d'informations qui nous donne la possibilité de juger d'une façon proactive son contenu.

Nous avons réalisé un système de classification qui peut analyser une base d'URLs étiquetés selon un nombre d'heuristiques qui peuvent détecter les techniques utilisées pour tromper les utilisateurs.

* La présence de formes : Le but d'un site de phishing est d'avoir les informations de la victime ce qui se fait par l'utilisation de formes d'input. Ce paramètre peut être utilisé comme un pré filtre « s'il y a des formes d'input nous faisons l'analyse sinon pas la peine de faire ».

* L'âge du domaine : En général les sites de phishing on une très courte durée de vie avant qu'ils soient fermés par leurs hébergeurs a cause des plaintes d'utilisateurs et des

autorités, alors on peut prendre une durée de vie minimum pour dire qu'un site est sur ou pas.

* La longueur d'URL : Les sites de phishing utilisent deux façons pour tromper les utilisateurs, des URLs avec une importante longueur pour cacher la vraie adresse, dans d'autres cas on utilise des URLs réduites (comme le fait TinyURL.com) pour cacher la vraie adresse et passer l'analyse de la longueur.

Nous avons prit la longueur comme paramètre entier.

* Adresses avec IP : pour détourner l'attention des utilisateurs on utilise des adresses avec juste un IP, ce qui n'est pas très évident pour une importante compagnie qui n'a pas un nom de domaine.

Un paramètre booléen pour designer si l'adresse contient un IP affecter par la valeur 1 sinon 0.

* Points dans l'URL : Les adresses de phishing contiennent généralement un nombre de sous-domaines pour que l'url apparaisse légitime.

Nous avons pris la valeur 3 comme paramètre de division entre phishing ou pas car une valeur importante de nombre de point indique un ou plusieurs sous-domaines.

* Niveau de sécurité : L'utilisation de protocoles sécurisés est un facteur important pour dire est ce sur de visiter un lien ou pas malgré qu'il y a des sites de phishing avec https.

Notre système cherche si l'adresse est sécurisée alors il affecte au paramètre la valeur 1 (pour https) sinon il l'affecte un 0 (en cas de http, ftp ou encas d'absence de protocole dans le lien).

* Caractères spéciaux ‘@’,’-’ ,’_’ : On peut faire une redirection directe d’une adresse a une autre a l’aide de ‘@’, et on utilise ’-’,’_’ pour faire faire une adresse ressemblante a la légitime.

Nous avons pris la valeur 1 comme paramètre de division (1 si le nombre est supérieur à 1, 0 sinon).

* Modification d’encodage d’URL: Pour détourner l’analyse on modifie les URLs en changeant des caractères par leurs codes html %20 pour un espace par exemple.

Nous avons pris la valeur 1 comme paramètre de division (1 si le nombre est supérieur à 2, 0 sinon).

III BASE D’URLS UTILISEE

Pour la base de d’URLs nous avons eu recours à une base publique utilisée par OPENDNS, cette base est publiée par leur site phishtank.com (phishtank.com est créé par OPENDNS) qui est un site ou les internautes peuvent signaler les sites suspects. La base rendue publique par le site est vérifiée par des experts qui disent que ces adresses sont effectivement des adresses de phishing. Nous avons utilisé deux versions avec des dates de sorties différentes pour plus de précision.

Pour la base des adresses sures nous avons utilisé, plusieurs sources comme ALEXA.com, et Google ranking spécialisés dans les statistiques du trafique sur internet et qui donnent le classement des sites les plus populaire périodiquement.

Pour le nombre d’instances de la base nous avons

6069 : adresses de phishing. 535 : adresses.

III-I STRUCTURE DE LA BASE DE PHISHING

La base de données est présente sous forme d’un fichier XML, avec les informations suivantes (un exemple d’instance) :

```
<url>http://aerospecialties.aerospecialties.com/osc22/mastercard.number/account.php</url>

<phish_id>1278385</phish_id>

<phish_detail_url>http://www.phishtank.com/phish_detail.php?phish_id=1278385</phish_detail_url>

<ip_address>209.161.24.98</ip_address>

<submission_time>2011-05-20T01:05:01+00:00</submission_time>

<verified>yes</verified>

<verification_time>2011-05-20T01:40:58+00:00</verification_time>

<online>yes</online>

<target>Mastercard</target>
```

Figure3.1 : Base de phishing sous forme XML.

phish_id : ID ou référence du site du phishing.
phish_detail_url : details sur phishtank du site en question.
url : L'url du site
submission_time : la date et l'heure de la déposition d'alarme sur le site.
Verified : site vérifier ou pas mais comme phishtank n'ajoute que les sites vérifies alors tous on une valeur :yes
verification_time : date et temps de vérification
online : statut du site
target : la compagnie ou la marque visée par l'attaque (visa, mastercard…)

IV ENVIRONNEMENT DU TRAVAIL

Pour le coté application de cette recherche nous avons travaillé avec DELPHI 7 Entreprise ce qui est un EDI édité par BORLAND, pour sa rapidité et la facilité de prendre en main, nous l'avons utilisé pour calculer les métriques prisent en considération.

Nous avons utilisé aussi WEKA qui est un ensemble d'outils permettant de manipuler et d'analyser des fichiers de données, et qui implémente beaucoup d'algorithmes d'intelligence artificielle. Il est écrit en Java.

V FONCTIONNALITES DE L'APPLICATION

Notre application se constitue de deux parties majeures :

* La première partie (Partie développeur) : contient tout se qui a un lien avec le développement, ainsi nous trouvons l'interface pour le calcul des différents paramètres que nous avons vu précédemment.

* La deuxième partie (Partie utilisateur) : Une partie utilisée par l'utilisateur final, qui peut décider pour lui si c'est le site est légitime ou c'est un phish, son implémentation est une barre d'un navigateur web qui bloque l'accès aux sites sensibles

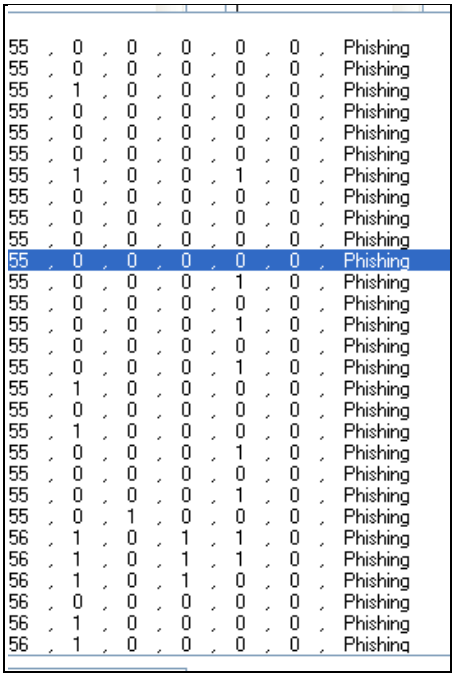


Figure 3.2 : aperçu des résultats

VI EVALUATION

Pour la partie d'évaluation nous avons utilisé la suite de logiciel d'apprentissage automatique et d'exploration de données « WEKA », et ce pour avoir plusieurs classifieurs en main, faire la comparaison, et avoir une analyse claire des résultats.
Pour cela nous avons transformé le fichier de sortie de notre application en fichier ARFF de ce format

```
@RELATION phishing    % Nom de l'ensemble de données

@ATTRIBUTE longueur                numeric        % déclaration des
@ATTRIBUTE nombredepoints          {0,1}          % attributs et leurs
@ATTRIBUTE securiteduprotocol      {0,1}          % types
@ATTRIBUTE sansnomdudomaine        {0,1}
@ATTRIBUTE caracteresspeciaux      {0,1}
@ATTRIBUTE caracteresunicodes      {0,1}
@ATTRIBUTE class                   {Phishing, SUR}

@DATA                  % la partie data
55 , 0 , 0 , 0 , 0 , 0 , Phishing        % les instances avec leurs
55 , 0 , 0 , 0 , 0 , 0 , Phishing        % classes
55 , 1 , 0 , 0 , 0 , 0 , Phishing
55 , 0 , 0 , 0 , 0 , 0 , Phishing
55 , 0 , 0 , 0 , 0 , 0 , Phishing
55 , 0 , 0 , 0 , 0 , 0 , Phishing
31 , 0 , 0 , 0 , 0 , 0 , SUR
31 , 0 , 1 , 0 , 0 , 0 , SUR
31 , 0 , 1 , 0 , 0 , 0 , SUR
31 , 1 , 1 , 0 , 0 , 0 , SUR
32 , 0 , 0 , 0 , 0 , 0 , SUR
32 , 0 , 0 , 0 , 0 , 0 , SUR
32 , 0 , 0 , 0 , 0 , 0 , SUR
```

Figure 3.3 : fichier au format ARFF

VI-I RESULTATS SOUS WEKA

Nous avons utilisé plusieurs algorithmes d'apprentissage

* Arbres de décision.
* Classification bayesienne probabiliste (naïve bayes, réseaux bayesiens).

* Machine à vecteur de support(SVM).

Le tableau suivant résume les résultats obtenus avec ces classifieurs

	SVM(SMO)	Arbres (j48)	Randomtree	Réseaux bayesiens	Naïve Bayes
Instances correctement classées	6218 (94.1551 %)	6485 (98.1981 %)	6500 (98.4252 %)	6445 (97.5924 %)	6360 (96.3053 %)
Instances mal classées	386 (5.8449 %)	119 (1.8019 %)	104 (1.5748 %)	159 (2.4076 %)	244 (3.6947 %)

Tableau 3.1: Résultats de classification

Le graphe suivant compare le nombre d'instances correctement classées par rapport au nombre total d'instances.

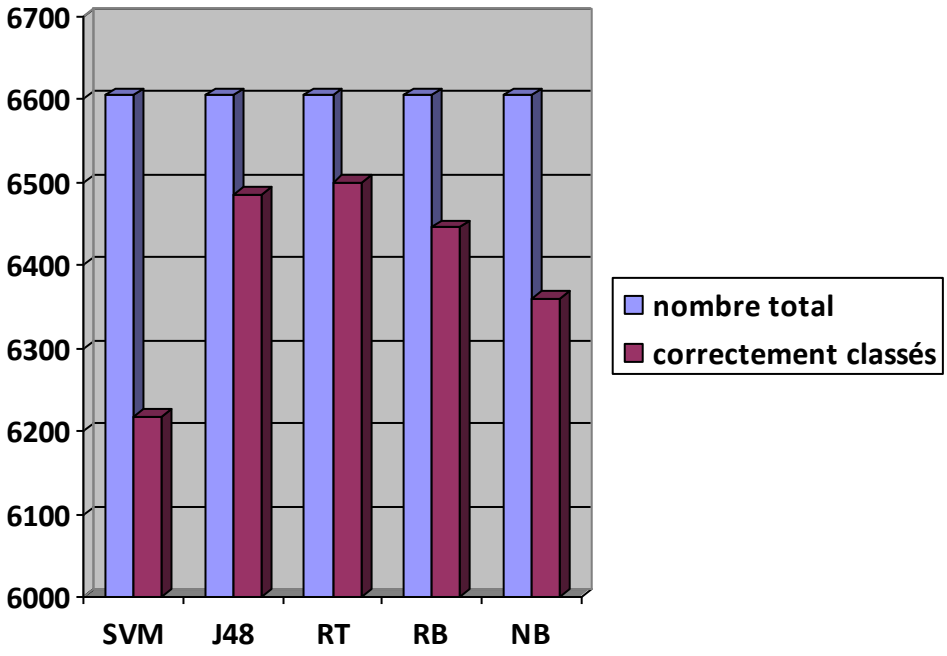


Figure 3.4: nombre d'instances correctement classifiées.

Le tableau suivant récapitule les valeurs de précision, les vrais positifs

	SVM(SMO)		Arbres (j48)		Randomtree		Réseaux bayesiens		Naive Bayes	
Classe	phishing	sur	phishing	sur	phishing	sur	phishing	sur	phishing	sur
Vrai positif	0.995	0.66	0.987	0.925	0.988	0.064	0.98	0.929	0.966	0.931
Faux positif	0.34	0.005	0.075	0.013	0.936	0.012	0.071	0.02	0.069	0.034

Tableau 3.2 : Tableau récapitulatif des précisions.

VI-II CHOIX DU MEILLEUR RESULTATS

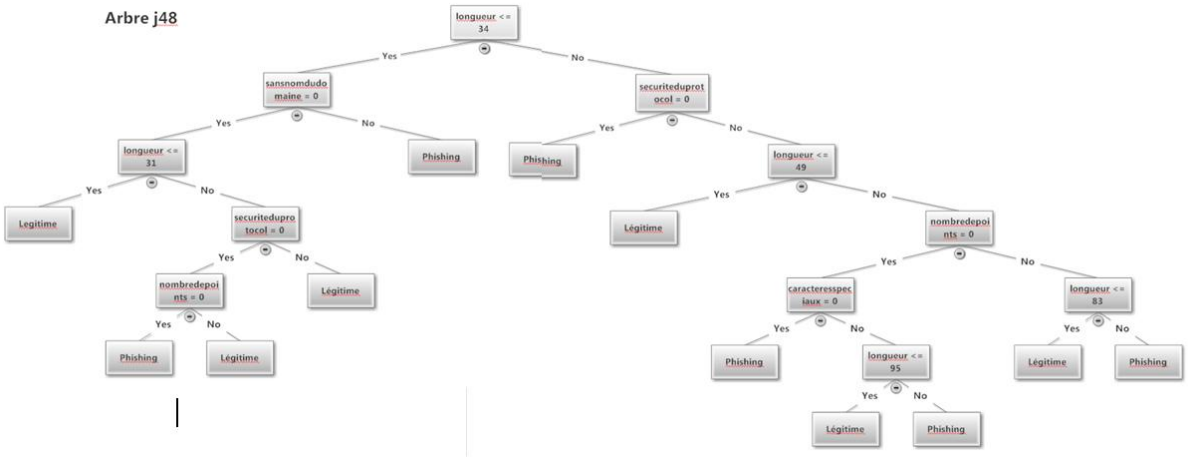
Le but de cette étude est de éliminer voir réduire les faux négatifs pour les deux classes (les adresses phishing classée comme valides et inversement).

Pour cela et après l'analyse de données nous avons trouvé que les réseaux bayesiens et la meilleure solution avec la matrice de confusion.

```
=== Confusion Matrix === (RANDOM TREE)
    a       b       <-- classified as
 5999      70 |     a = Phishing
   34     501|     b = SUR
```

Figure3.5 : Matrice de confusion pour les réseaux bayesiens

Voici l'arbre de décision de type J48 que Weka a généré



VI-III DISCUSSION DES RESULTATS

Apres avoir essayé plusieurs classifieurs nous avons trouvé que les réseaux bayesiens et les arbres de décision sont les mieux adaptés, les mieux performants avec une précision avoisinant les 98 %.

En se basant sur la dernière matrice de confusion nous pouvons dire deux choses

1- Le système détecte des faux positifs (fausses alarmes 34 instances), qui n'est pas un vrai problème vis-à-vis l'utilisateur, malgré que c'est un argument de confort pour l'utilisateur.

2- Le problème majeur est qu'il y a un nombre d'instances qui sont passés inaperçus (70 instances confirmées phishing) par le système alors qu'ils constituent un vrai danger pour l'utilisateur inexpérimenté.

VII CONCLUSION

Malgré les bons résultats obtenus par notre système, il y a beaucoup d'améliorations à faire comme l'ajout d'heuristiques, et l'intégration d'autres approches par prendre en considération le corps de l'email et le site en considération.

La base que nous avons utilisée est vraiment réduite (6604 instances) et ne reflète pas vraiment tout les cas possibles, les formes imaginables que peut avoir une adresse phishing. Alors pour faire un système fiable et applicable en réalité il faut avoir beaucoup plus d'instances dans la base d'apprentissage.