Détection du pharming côté client vers un remplacement du nom de domaine

Sommaire

Le Chapitre 3 montre l'efficacité des tests heuristiques pour la détection des sites de phishing, ainsi que leurs limitations. Il fait notamment apparaître la prédominance de certains heuristiques dans la différenciation des sites légitimes et contrefaits. Plus particulièrement, il met en évidence l'intérêt de l'analyse du code source HTML des pages webs.

En complément, le Chapitre 4 détaille les différentes techniques de comparaisons de contenus pouvant s'appliquer aux codes sources HTML. Par ailleurs, alors que les zones de vulnérabilités sont nombreuses côté client, ce même chapitre démontre la difficulté de détection des attaques de pharming pour l'Internaute.

Dans les Chapitres 5 et 6, nous nous intéressons à la portabilité de cette analyse du code source HTML à l'identification de sites webs, et plus particulièrement au cadre de la détection des attaques de pharming auxquelles le client est encore trop vulnérable.

Notre proposition est étayée via deux approches - visant à détecter le pharming réalisé côté client - basées sur l'étude du code source HTML du site web visité, combinée à des requêtes DNS. Par souci de clarté et afin d'apporter un maximum de détails sur chacune de ces deux approches, nous les détaillons de manière successive. Il est important de noter que la seconde approche (développée dans le Chapitre 6) est issue de l'évolution de la première (détaillée ci-après), avec pour principaux objectifs la correction et l'amélioration des défauts majeurs rencontrés.

Nous démarrons donc ce chapitre par une introduction à l'approche développée (cf. section 5.1). Nous y expliquons notamment l'origine de notre proposition et nous en profitons pour la positionner par rapport aux travaux existants.

Puis, nous détaillons une étude préalable que nous avons réalisée sur les hypothèses de travail (cf. section 5.2) avec le double objectif : 1/ de tester si la vérification de l'adresse IP du domaine visité peut s'avérer un critère de détection pertinent, et 2/ d'évaluer s'il est possible de différencier des pages légitimes de pages contrefaites au travers de l'analyse du contenu global des codes sources HTML.

Enfin, nous développons la première approche proposée (cf. section 5.3) qui combine à la fois une vérification de l'adresse IP du domaine visité et l'analyse du contenu de la page web visitée vs. des éléments de référence. Ces éléments sont récupérés à partir des informations fournies par un serveur DNS, différent de celui auquel est connecté le client (c.-à-d. tel que proposé par son FAI). Ils sont constitués d'une adresse IP (liée au domaine visité) ainsi que du code source d'une page web, tous deux dits de référence. La page web de référence est récupérée grâce à la génération d'une nouvelle requête HTTP. Cette requête est basée sur l'URL initialement visitée par l'utilisateur, au sein de laquelle la zone de domaine est remplacée par l'adresse IP de référence. Les tests réalisés dans cette proposition portent sur 108 URLs de login légitimes, évaluées depuis 11 localisations géographiques réparties sur 5 continents, ainsi que sur 37 couples de pages légitimes-contrefaites.

Ce chapitre fait partie de nos contributions : cette première proposition a été publiée et présentée à la conférence *New Technologies, Mobility and Security* (NTMS) en Février 2011 [GGL11b].

5.1 Introduction à l'approche développée

Notre proposition tire son origine de 2 éléments :

- La volonté de se focaliser côté client, afin de détecter les attaques de pharming pour lesquelles les efforts déployés côté réseau sont encore inefficaces.
- Une étude menée par Stamm et al. [SRM07], que nous avons considérée comme déterminante, puisqu'elle met en avant la vulnérabilité essentielle des routeurs personnels déployés à large échelle, à savoir : leurs utilisateurs. En effet, la méconnaissance des Internautes, qui utilisent les routeurs personnels dans leur pré-configuration d'origine, induit des points d'entrée providentiels pour les attaquants (cf. section 4.1.2.1).

A l'image des mécanismes de protection couramment utilisés contre les attaques de phishing (ie. les barres d'outils intégrées dans les navigateurs), nous avons élaboré une proposition de détection des attaques visant à s'intégrer dans le navigateur de l'Internaute. D'où les développements menés en langage Java.

Notre proposition vise à s'intégrer de façon complémentaire dans une barre d'outils anti-phishing. Un indicateur visuel très simple et binaire a pour rôle d'indiquer le niveau de confiance envers le site visité. En complément, en cas de site suspicieux, l'Internaute est alerté via une notification active (p.ex. un message d'alerte affiché sous forme de pop-up) (cf. section 6.2.3).

Le cœur de notre proposition repose sur la combinaison de deux analyses menées grâce aux informations fournies par plusieurs serveurs DNS :
- La vérification de l'adresse IP du domaine visité
- L'analyse et la comparaison du code source de la page web

Les attaques ciblées : Dans l'absolu, notre proposition de détection du pharming côté client vise les attaques mentionnées en figure 5.1 (cf. section 4.1.2.1 pour plus de détails sur ces attaques). Ainsi, elle cible aussi bien les attaques qui visent à éviter l'interrogation du serveur DNS configuré (par ajout de fausses entrées IP/FQDN), que les attaques qui visent à rediriger le trafic (par ajout de routes statiques[1] ou de proxy HTTP qui orientent le trafic vers les machines de l'attaquant), ou encore les attaques qui modifient l'adresse IP du serveur DNS habituellement interrogé.
Précisons toutefois que seule la seconde approche développée dans le Chapitre 6 permet de cibler l'ensemble de ces attaques.

Cible / Zone de vulnérabilités		Ajout de fausse association	Modification de l'adresse IP du serveur DNS interrogé	Autre
Poste client	Fichier HOSTS	**X** (IP/FQDN)		
	Configuration Réseaux		**X**	
	Cache DNS	**X** (IP/FQDN)		
	Navigateur web	**X** (IP/FQDN)		Proxy HTTP *DNS rebinding*
Réseau local			**X**	
Routeur personnel		**X** (routage)	**X**	

FIGURE 5.1 – Les attaques ciblées par notre proposition de détection du pharming côté client

Positionnement par rapport aux travaux similaires : Les travaux précédents, qui s'apparentent aux deux approches proposées dans notre étude, sont détaillés en sections 4.1.3.3.2 et 4.2.3. Les quatre études les plus proches sont discutées ci-après :
- La méthode développée par Cao et al. [CHL08] base sa détection du pharming sur une comparaison de l'adresse IP du domaine visité auprès d'une liste blanche d'adresses IP légitimes. Elle repose sur un principe de staticité des adresses IP associées aux pages de login. L'étude que nous menons dans ce chapitre va démontrer que la prise en compte de l'adresse IP du domaine visité s'avère un critère parfois insuffisant pour déterminer la légitimité d'un site. A contrario de cette étude, nous avons également exclu ici toute méthode de détection nécessitant le maintien d'une liste blanche ou liste noire (pour les faiblesses associées à cette approche, cf. explications détaillées dans les Chapitres 2 et 3).

- La méthode de détection proposée par Bin et al. [BQX10] est certainement celle qui s'approche le plus de notre travail de par l'étude combinée de deux types de contenus (c.-à-d. une donnée saisie dans une page web et l'adresse IP associée au domaine visité). Leur technique s'appuie sur une liste blanche pré-établie de noms de banques, adresses IP et plages de numéros de cartes bancaires associés. De notre point de vue, elle présente deux inconvénients majeurs : 1/ Elle nécessite un temps de latence proche de zéro. En effet, leur méthode implique obligatoirement un début de

1. Cette attaque n'est détectée que si l'adresse IP retournée par le serveur de référence - interrogé dans notre proposition - est différente de celle configurée dans la route statique.

saisie d'une information critique de l'utilisateur, à savoir son numéro de carte bancaire, en amont du lancement de la détection. 2/ De plus - tel qu'évoqué précédemment -, nous considérons que la nécessité du maintien d'une liste blanche est à éviter, de par les vulnérabilités supplémentaires qu'elle introduit.

- L'approche développée par Bilge et al. [BKKB11], publiée très récemment, est celle qui s'apparente le plus à notre étude en terme de techniques de détection utilisées, à savoir les tests heuristiques basés sur les informations DNS. Nos travaux et les leurs se basent sur une vérification de l'adresse IP du domaine visité. Néanmoins, leur étude se destine plutôt au phishing, de par le contenu de la majorité des critères étudiés : âge du site visité, nombre de caractères alphanumériques du nom de domaine, etc.

- L'étude de Reddy et al. [RRJ11], récemment publiée, propose deux scénarios de détection des sites contrefaits côté client. L'un des deux scénarios exposé se rapproche de notre proposition, bien qu'il se destine plutôt à la détection des sites de phishing. Celui-ci s'appuie sur une vérification de l'adresse IP du site visité, combinée à une analyse de l'URL en utilisant la Distance de Levenshtein. Les éléments de référence utilisés pour leurs vérification/comparaison sont issus d'une liste blanche pré-établie. Comme exposé précédemment, nous pensons que l'utilisation d'une liste blanche est un point de vulnérabilité important de la solution. De plus, ce scénario ne peut s'appliquer aux sites de pharming puisqu'il ne s'intéresse qu'à l'étude de l'URL visitée.

5.2 Etude préalable sur les hypothèses de travail

Plusieurs pistes ont été étudiées avant d'aboutir aux deux approches décrites en sections 5.3 et 6.1, l'idée essentielle étant de trouver un moyen d'utiliser un serveur DNS et une page web dits de référence.

Divers scénarios intermédiaires ont été envisagés pour la définition d'un serveur de référence. Toutefois, parce que ces scénarios impliquaient l'échange d'une adresse de référence entre le client et le FAI (ou une tierce partie), nous avons jugé préférable de les abandonner. De plus, ils restaient vulnérables à une corruption de l'adresse de référence en amont, dans la chaîne de résolution DNS.

Nous avons également un temps envisagé une comparaison "délocalisée", réalisée en dehors du réseau client, tant pour la vérification de l'adresse IP que pour la comparaison de pages webs. Néanmoins, les résultats de notre étude préalable nous ont indiqué que cela s'avérait difficilement exploitable (cf. sections 5.2.1 et 5.2.2).

Cette section s'intéresse donc à expliquer les études préalables réalisées, ainsi que les choix auxquels elles ont abouti dans la conception de notre approche.

5.2.1 Variabilité de l'adresse IP du domaine visité

Un des fondements de notre approche repose sur la possibilité de vérifier l'adresse IP du site visité.

Nous avons donc conduit un premier set d'expérimentations depuis 9 localisations géographiques réparties sur 5 continents (Amérique du Nord, Europe, Afrique, Asie et Australie). Pour chaque localisation, nous avons récupéré les adresses IP retournées par le serveur DNS par défaut ainsi que 2 serveurs DNS de référence (OpenDNS [ope] et GoogleDNS [gooa]), afin d'analyser les variations d'adresses IP des domaines visités.

5.2.1.1 Échantillon d'URLs

Nous avons testé 226 domaines (plus exactement des FQDN) de sites webs HTTP, sélectionnés de la manière suivante :

- 100 domaines issus des sites webs les plus populaires au niveau mondial[1],
- 100 domaines issus des sites webs les plus populaires en France[1],
- et 26 domaines issus de sites bancaires

1. Les domaines ont été récupérés depuis le Top 1000 Google des sites les plus visités [Gooc], le Top 500 Alexa des sites webs [Al] ainsi que la base de sites publiée par NetCraft [Net].

Tableau 5.1 – Quelques résultats OpenDNS pour des FQDN génériques, sur 4 localisations géographiques

	France	Tunisie	Mexique	Turquie
www.facebook.com	66.220.146.25	66.220.153.19	66.220.146.11	66.220.153.11
www.apple.com	92.123.129.15	92.123.193.15	184.50.237.15	92.123.233.15
www.ask.com	62.41.85.83	62.41.85.49	63.97.94.41	194.221.37.139
	62.41.85.49	62.41.85.83	63.97.94.80	194.221.37.163
www.amazon.com	72.21.210.250	72.21.210.250	72.21.207.65	207.171.166.252
www.commentcamarche.net	62.41.85.40	62.41.85.40	63.97.94.35	194.221.37.163
	62.41.85.48	62.41.85.48	63.97.94.49	194.221.37.176
www.comcast.net	67.215.65.132	213.155.157.49	63.97.94.10	93.158.110.107
		213.155.157.16	63.97.94.58	93.158.110.144

Tableau 5.2 – Quelques résultats OpenDNS pour des FQDN plus précis, sur 4 localisations géographiques

	France	Tunisie	Mexique	Turquie
webmail.laposte.net	193.251.214.117	193.251.214.117	193.251.214.117	193.251.214.117
portail.free.fr	212.27.48.10	212.27.48.10	212.27.48.10	212.27.48.10
images.google.fr	66.102.9.99	66.102.9.105	209.85.225.106	74.125.39.105
	66.102.9.103	66.102.9.99	209.85.225.105	74.125.39.147
	66.102.9.147	66.102.9.106	209.85.225.103	74.125.39.104
	66.102.9.106	66.102.9.103	209.85.225.147	74.125.39.99
	66.102.9.105	66.102.9.147	209.85.225.99	74.125.39.106
	66.102.9.104	66.102.9.104	209.85.225.104	74.125.39.103
news.bbc.co.uk	212.58.226.77	212.58.226.138	212.58.246.83	212.58.226.140
login.yahoo.com	69.147.112.160	209.191.92.114	217.146.187.123	69.147.112.160
			217.12.8.76	
particuliers.societegenerale.fr	193.178.154.165	193.178.154.167	193.178.154.165	193.178.154.166
	193.178.154.166	193.178.154.164	193.178.154.166	193.178.154.167
	193.178.154.167	193.178.154.165	193.178.154.167	193.178.154.164
	193.178.154.164	193.178.154.166	193.178.154.164	193.178.154.165

Nous avons pris soin de multiplier les secteurs d'activité (p.ex. e-commerce, réseaux sociaux, annuaires téléphoniques, banques, forums de discussion, jeux en ligne, etc.), les langages des pages webs ainsi que leurs TLDs.

5.2.1.2 Variabilité IP entre localisations géographiques

Les résultats de tests démontrent que, pour les domaines évalués, les adresses IP varient notablement selon la localisation, et ce quel que soit le serveur DNS interrogé. A titre d'exemple, le tableau 5.1 reprend quelques résultats retournés par OpenDNS depuis 4 localisations différentes.

Nous constatons par exemple que pour Facebook, Apple et Comcast - malgré quelques similitudes -, il n'y a aucun recoupement d'adresses entre les 4 localisations. Tandis que pour Ask, Amazon et Commentcamarche, nous avons quelques recoupements d'adresses, entre la France et la Tunisie.

En utilisant des FQDN plus précis (p.ex. images.google.fr, portail.free.fr, webmail.laposte.net, etc.), nous obtenons davantage de résultats partiellement convergents, entre les différentes localisations (cf. tableau 5.2). Toutefois, nous notons que des exceptions subsistent (p.ex. login.yahoo.com, news.bbc.co.uk).

5.2.1.3 Variabilité IP depuis une même localisation géographique

Nous avons également évalué la staticité des adresses IP sur 4 localisations. Pour chacune d'entre elles, nous avons comparé - en local - les adresses IP retournées par le serveur DNS par défaut ainsi que celles retournées par nos 2 serveurs de référence (GoogleDNS et OpenDNS).

Pour les FQDN génériques, selon la localisation étudiée et pour une même liste d'URLs, nous constatons une grande variabilité des résultats (cf. tableau 5.3) entre les adresses IP retournées par le DNS par défaut et les adresses IP de référence.

Néanmoins, lorsque nous analysons les résultats portant sur les FQDN plus précis, nous constatons à nouveau une nette amélioration (c.-à-d. les adresses IP des serveurs DNS Défaut et Référence sont

Tableau 5.3 – Convergence des adresses IP Défaut et Référence pour 4 localisations géographiques

	Taux de convergence avec les adresses IP retournées par le serveur DNS par défaut (min ≤ moyenne ≤ max)	Écart-Type[1]
OpenDNS	62.83% ≤ **71.24%** ≤ 92.92%	14.55%
GoogleDNS	62.39% ≤ **73.34%** ≤ 84.51%	10.82%

[1] entre localisations

plus convergentes).

5.2.1.4 Problématique des informations WHOIS

Nous avons envisagé d'exploiter les données WHOIS en vue de vérifier l'identité des domaines visités. Néanmoins, plusieurs problèmes se posent :

- La RFC 3912 qui définit le protocole WHOIS ne spécifie pas de norme concernant les données stockées. Elle n'indique pas non plus de spécification sur le formatage/contenu des réponses WHOIS, envoyées en mode texte. Chaque registrar fournit donc ce service dans le format qu'il désire, ce qui rend les données difficilement exploitables de manière automatisée.
- On peut alors imaginer se focaliser sur les cc-TLD, qui sont généralement gérés par une même autorité gouvernementale imposant un format standard pour l'ensemble de ses domaines. Néanmoins, le service WHOIS n'étant pas dimensionné pour le traitement automatique, l'interrogation automatique s'avère difficile. En effet, lorsque nous avons essayé d'automatiser des requêtes WHOIS, nous constatons qu'après une dizaine de requêtes depuis une même machine, l'adresse IP source est bloquée.
- Il n'y a pas non plus d'obligation de maintenir des informations WHOIS à jour. D'ailleurs, ces informations accessibles au public, posent des problèmes liés à la protection (au sens protection de la vie privée). De nombreuses données stockées sont donc obsolètes et peu fiables (En 2010, une étude de l'ICANN - menée sur 5 g-TLD et 1419 enregistrements - a montré que seulement 23% des domaines interrogés ont des données WHOIS à jour [atUoCfI10]).
- Enfin, lorsqu'une entreprise utilise des services de GSLB (Global Server Load Balancing) pour une meilleure disponibilité, les informations retournées pour une adresse IP peuvent s'avérer fausses ou inexploitables pour l'identification d'un domaine. Considérons un exemple : Une interrogation DNS sur le FQDN `www.bouyguestelecom.fr` retourne les adresses IP 62.41.70.12 et 62.41.70.138. Une interrogation de la base WHOIS pour l'adresse IP 62.41.70.12 retourne des informations sur la société Akamai International. Il n'y alors aucun moyen de faire le lien entre l'entreprise BouyguesTelecom et la société Akamai, spécialisée dans la mise en cache de contenus web.

5.2.1.5 Synthèse de l'étude préalable sur la variabilité des adresses IP : choix des pages de login

Les données WHOIS peuvent générer un taux de faux-positif important, tant par leur obsolescence que par l'externalisation de la gestion des sites webs. Elles s'avèrent donc inexploitables pour la vérification de l'adresse IP d'un FQDN.

L'interrogation DNS sur des FQDN génériques démontre une trop grande variabilité des réponses (c.-à-d. adresses IP retournées) selon la localisation géographique. Notre approche visant à s'adresser à tout client, quelle que soit sa localisation, il s'avère donc impossible de délocaliser la vérification d'adresse IP sur un serveur déporté.

Enfin, des interrogations DNS sur des FQDN "plus précis" semblent être davantage propices à des résultats convergents entre serveur par défaut et serveur de référence, depuis une même localisation. Une étude menée par Cao et al. [CHL08] semble d'ailleurs indiquer une stabilité des adresses IP associées aux sites de login.

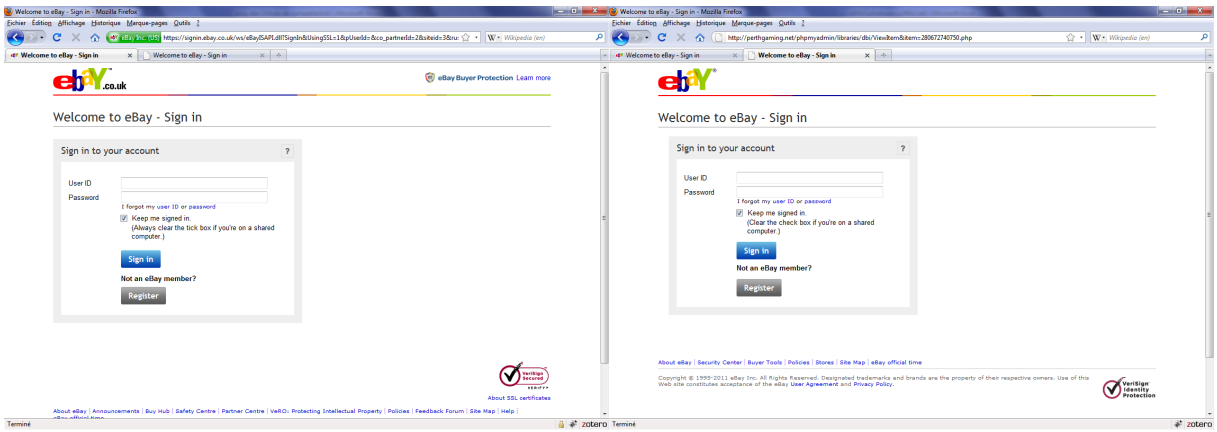
Pour les raisons évoquées précédemment, **nous focalisons donc la suite de notre étude sur des FQDN "plus précis", et plus spécifiquement sur des pages de login, cibles des attaques de pharming**.

5.2.2 Contenu des pages webs

5.2.2.1 Analyse des caractéristiques des pages webs légitimes et contrefaites

Nos travaux menés sur l'étude de pages webs légitimes et contrefaites, détaillés dans le Chapitre 2, ainsi que des analyses de pages récupérées depuis différents navigateurs webs et différentes localisations, font apparaître les caractéristiques et difficultés suivantes :

- **Le contenu des pages webs est de plus en plus dynamique**, incluant des flux RSS, des images renouvelées fréquemment, des publicités, etc.
- **Les attaquants créent des sites contrefaits qui, visuellement, sont de plus en plus ressemblants aux sites légitimes** (cf. exemple en figure 5.2). En effet, l'utilisation d'aspirateurs de sites webs et le choix délibéré de conserver un maximum de liens du site web légitime, permettent de minimiser les possibilités de détection de la contrefaçon et de leurrer un plus grand nombre d'utilisateurs.
- **Les sites légitimes et contrefaits utilisent tous deux des chemins absolus et relatifs** pour les références aux liens, images, etc.
- **Le code source des pages webs peut être modifié selon le navigateur utilisé par le client**. En effet, des morceaux de script peuvent être ajoutés aux codes sources des pages webs, selon le navigateur utilisé (p.ex. Internet Explorer, Firefox, Opera, etc.)
- **La structure HTML d'une page web est parfois modifiée d'une localisation à une autre**. Nous avons en effet constaté que l'ordonnancement du code source d'une même page web récupérée - à partir de la même URL - depuis différentes localisations, se retrouve parfois totalement bouleversé, bien que les deux pages apparaissent totalement identiques à l'affichage dans le navigateur.
- **Le contenu d'une page HTML est adapté aux préférences et/ou à la localisation de l'utilisateur**. En effet, l'affichage se fait selon la langue choisie, ou la localisation depuis laquelle la requête est effectuée.



(a) Site légitime (b) Site contrefait

FIGURE 5.2 – Captures d'écran du site légitime eBay (`https://signin.ebay.co.uk/ws/eBayISAPI.dll?SignIn&UsingSSL...`) et d'une contrefaçon (`http://perthgaming.net/phpmyadmin/libraries/dbi/ViewItem&item=280672740750.php`), récupérées le 12 Juin 2011

Nous en déduisons alors que baser l'analyse et la comparaison des pages webs uniquement sur la structure, la page complète (c.-à-d. incluant les contenus dynamiques) ou le type de liens, peut générer des taux de faux-positifs élevés.

La variabilité du contenu des pages, selon le navigateur utilisé, les paramètres de l'utilisateur et la localisation géographique, nous amène également à exclure toute possibilité de comparaison déportée. En effet, le contenu de la page web étant assujetti au contexte client (c.-à-d. le type de navigateur, sa version, les éventuels plug-in associés, etc.), il semble difficile de disposer d'une page web de référence personnalisée avec ces mêmes paramètres depuis une autre machine. En complément, de par la haute qualité de bon nombres de pages contrefaites qui répliquent (quasi-)parfaitement les sites légitimes usurpés et le développement des contenus dynamiques (qui peuvent conduire à des détections erronées [MKK08]), nous choisissons également d'exclure toutes techniques de comparaison par "images" de la page web.

Pour l'ensemble de ces raisons, **nous décidons donc de concentrer notre comparaison de pages webs sur l'étude du code source HTML**.

5.2.2.2 Décision de légitimité

Notre décision de légitimité d'une page web va s'appuyer sur un seuil de décision pré-défini, auquel sera comparé le taux de similitude obtenu pour les deux pages confrontées. Pour évaluer l'efficacité de notre proposition et déterminer ce seuil de décision, nous basons donc nos analyses sur deux types de comparaisons :

- des comparaisons de pages légitimes. L'idée est de déterminer si deux pages légitimes, potentiellement identiques - puisqu'elles correspondent à une même URL - mais récupérées par deux biais différents, apparaissent effectivement - du point de vue de notre comparaison - comme totalement ou très similaires.
- des comparaisons entre pages légitimes et contrefaites. L'idée est de déterminer si deux pages, visuellement très ressemblantes mais générées par deux personnes/organismes totalement différents, sont effectivement - du point de vue de notre comparaison - très différentes.

Plus la frontière basse des comparaisons de pages légitimes sera distincte et éloignée de la frontière haute des pages légitimes-contrefaites, plus il sera facile de déterminer un seuil de décision valable, amenant à des décisions fiables.

5.2.2.3 Application des méthodes de comparaison de documents HTML à la problématique des pages légitimes et contrefaites

Alors que le Chapitre 4 a détaillé les différentes techniques de comparaisons de contenus pouvant s'appliquer aux documents HTML, nous examinons à présent quelles sont celles qui pourraient s'appliquer à notre étude (c.-à-d. au cas des comparaisons de pages légitimes et contrefaites).

5.2.2.3.1 Pertinence de l'application des différentes méthodes de comparaison de textes à notre étude : Les algorithmes d'alignement de chaînes (cf. section 4.2.1.1) s'appliquent généralement au domaine de la bio-informatique où il est possible de définir une distance entre deux protéines. Dans ce contexte, les algorithmes d'alignement de chaînes reposent sur l'utilisation d'une matrice de substitution (p.ex. PAM, BLOSUM), dont le rôle est le suivant : pour une protéine donnée, elle indique sa probabilité de substitution par une autre protéine, suite à une mutation dans le temps [Jas].

Une application de ce type d'algorithmes à notre étude s'avère plus compliquée. En effet, il est difficile de prédire le caractère de remplacement pour un caractère donné, en cas de modification du code source de la page web (p.ex. un changement dans un lien hypertexte). Nous pourrions éventuellement nous contenter d'utiliser la pénalité de trou sans matrice de substitution. Ainsi un trou serait inséré jusqu'à ce que la correspondance reprenne. Cependant, au regard de la taille de l'alphabet et de la diversité des caractères pouvant être utilisés (c.-à-d. selon les langages), il se peut que la correspondance tarde à reprendre. Nous pourrions alors nous retrouver avec des zones de trous conséquentes, ce qui impacterait le degré de similitude des pages de manières conséquente (si on comptabilise le nombre de trous), pour un simple changement d'arborescence dans des liens par exemple. Par conséquent, notre choix ne s'est pas porté pas sur ce type d'algorithmes.

Les algorithmes de recherche de sous-chaînes (cf. section 4.2.1.2), qui permettent de recherche une chaîne de caractères au sein d'un texte, ne sont pas adaptés à notre problématique. Ils ne sont donc pas exploités dans notre étude.

Dans les algorithmes de mesure de similarité (cf. section 4.2.1.3), **seuls deux types d'algorithmes/approches retiennent notre attention : l'algorithme de Distance de Levenshtein et l'approche *N-gram***.

En effet, la Distance de Levenshtein prend en compte les 3 types d'opérations qui nous intéressent : la modification, l'ajout et la suppression. Elle nous semble donc pertinente pour l'étude développée.

La Distance de Hamming, quant à elle, ne s'applique que sur des chaînes de longueur identique. De plus, elle ne prend en compte que les substitutions de caractères. Elle ne peut donc être révélatrice des éventuelles insertions de script malveillants, ce qui ne répond pas à nos attentes.

L'algorithme de Distance de Jaro-Winkler délivre un score de type taux de similitude, représentatif des caractères identiques et déplacés. Partant du principe que le code source HTML d'une page contrefaite est fortement similaire à son pendant légitime, nous pourrions imaginer que cet algorithme puisse être intéressant pour notre problématique. Il faudrait pour cela, au préalable, découper les codes HTML en mots (c.-à-d. à raison d'un mot par ligne par exemple). Toutefois, dès l'apparition d'un décalage de mots (principalement en première moitié de code), le score délivré serait fortement impacté. En effet, cette Distance ne permet pas de distinguer l'ajout et/ou la suppression de mots. A titre d'exemple, si on prend le cas des 2 pages webs suivantes : la page légitime contient 500 mots, la page contrefaite les mêmes 500 mots auxquels ont été rajoutés 50 mots (p.ex. correspondant à l'insertion d'un script malveillant) au milieu du code. Ainsi, seuls les 250 premiers mots apparaissent strictement identiques et dans le même ordre. En conséquence, le taux de similitude retourné ne serait que de 69% alors que seulement 10% de mots ont été insérés, et que le reste du code est identique en terme de contenu.

Enfin, l'approche *N-gram*, qui est fortement recommandée pour le filtrage et le routage de documents textes [CT94], nous semble elle aussi intéressante pour notre étude. En effet, si on considère que le code source HTML légitime est notre document de référence, nous pouvons utiliser l'approche *N-gram* pour lui associer un score. Le code source HTML suspect (c.-à-d. le site visité par l'utilisateur) peut alors être traité selon le même processus. Puis, selon l'écart mesuré entre les deux scores obtenus, nous pourrions déterminer la légitimité du site visité.

5.2.2.3.2 Pertinence de l'application des différentes méthodes de comparaison de structures à notre étude : De par leur contenu, les pages HTML peuvent également être considérées selon leur structure. La condition préalable à l'utilisation des algorithmes de calcul d'édition basés sur les arbres est que les deux documents comparés doivent être de structure identique. Si on part de l'hypothèse que la page contrefaite est fortement ressemblante à son pendant légitime, c'est en grande partie parce que les attaquants utilisent des outils/techniques d'aspiration du site légitime comme base à l'élaboration de la contrefaçon. Ainsi, on peut donc supposer que bon nombre de sites contrefaits doivent avoir une structure très similaire à la page usurpée[1].

Les algorithmes de mesure de similarité sont typiquement indiqués pour la comparaison de documents XML [DT03] - un langage parallèle au HTML - qui utilisent un format de balisage sous forme d'arborescence. En comparaison avec le HTML qui est utilisé pour l'affichage des pages webs, le XML est dit "extensible" car il permet de définir ses propres balises en fonction des données traitées. Il est donc utilisable sur différents types de plateformes (p.ex. les mobiles, etc). Cependant, le XML est un langage strict dont l'écriture se doit d'être rigoureuse. A contrario, le HTML permet plus d'erreurs syntaxiques automatiquement corrigées par les navigateurs webs. Il est ainsi désormais devenu fréquent de rencontrer des codes source de pages webs truffés d'erreurs syntaxiques (p.ex. oublis de fermeture de balises, guillemets manquants, changements de l'ordre de balises imbriquées lors de leur fermeture, etc.), particulièrement en ce qui concerne les sites contrefaits (cf. section 6.1.5.6). Certes l'utilisation du XHTML - un HTML 4.0 amélioré selon les règles de syntaxe du XML - pour l'élaboration des pages webs limite ces dérives syntaxiques. Toutefois, force est de constater qu'une grande majorité des pages webs analysées dans notre étude sont encore en HTML. Pour cette raison, il est par conséquent difficilement envisageable d'utiliser une méthode de comparaison utilisant des algorithmes de mesure de similarité basés sur les arbres. Nous n'avons donc pas retenu cette piste, ni celle des algorithmes d'alignement basés sur les arbres qui ne répondent pas à nos attentes (pour les mêmes raisons que celles évoquées en section 5.2.2.3.1). Reste à voir si l'arrivée du HTML5 - actuellement en cours de développement [W3Cc] -, confirmera ou infirmera cette tendance.

5.2.2.4 Les méthodes de comparaison retenues

Nous choisissons donc de focaliser notre comparaison de pages webs sur l'étude du code source HTML en utilisant deux approches : l'une par caractères, l'autre par mots. L'idée première est de refléter les modifications globales apportées à l'intégralité du code source. Les méthodes d'analyse visent ainsi à détecter les changements de structure (balises) et de contenu (texte affiché, liens, etc.) de manière globale, sans pour autant préciser les zones affectées.

1. Notons ici que nous considérons le cas le plus défavorable. En effet, les travaux de Medvet et al. [MKK08] indiquent que la structure DOM d'une page web n'est pas nécessairement un critère de comparaison fiable, puisqu'un attaquant peut utiliser une structure différente et aboutir malgré tout à une apparence visuelle très similaire à celle de la page légitime.

TABLEAU 5.4 – Échelle des valeurs Ascii attribuées aux caractères alphanumériques

Échelle des caractères alphanumériques	**Échelle des valeurs Ascii associées** en décimal
[a ;z]	[97 ;122]
[A ;Z]	[65 ;90]
[0 ;9]	[48 ;57]

TABLEAU 5.5 – Taux de similitude de 10 à 23 couples de pages de login légitimes-contrefaites, avec l'approche par caractères

	Taux de similitude entre page légitime et page contrefaite (min ≤ moyenne ≤ max)	Écart-Type[2]
10 couples de pages[1]	59.09% ≤ **80.54%** ≤ 93.12%	15.93%
23 couples de pages[1]	19.57% ≤ **77.41.%** ≤ 99.93%	26.05%

[1] Les deux séries de tests portent sur des couples de pages distincts
[2] entre couples de pages

5.2.2.4.1 Approche par caractères : Cette méthode de comparaison est basée sur la technique des *N-gram* (cf. section 4.2.1.3). Dans notre utilisation de cette approche, un score est assigné à chaque page web (p), fonction de l'occurrence (occ) de chaque caractère (i) :

$$Score(p) = \sum_{i=1}^{n} occ(i) \cdot valAscii(i)$$

où *valAscii(i)* représente la valeur Ascii de *(i)*

A noter que l'échelle des valeurs attribuées à chaque caractère alphanumérique dans la table Ascii, permet d'utiliser un score différent pour chacun d'entre eux, tout en limitant les écarts créés au sein d'une même classe (p.ex lettres minuscules, lettres majuscules, etc.). Ceci permet donc d'accorder un poids de niveau sensiblement similaire aux différents caractères (cf. tableau 5.4).

Puis, nous calculons le taux de similitude (exprimé en pourcentage) des deux pages (page défaut[1] vs. page de référence[2], ou page légitime vs. page contrefaite) en comparant leurs scores :

$$\text{Taux de similitude} = \frac{\min\{Score(p1), Score(p2)\}}{\max\{Score(p1), Score(p2)\}} \cdot 100$$

où *p1* représente la première page comparée
et *p2* la deuxième page comparée

Nous avons implémenté et testé cette méthode sur des couples de pages de login légitimes-contrefaits (p.ex. HSBC, Paypal, Bank of America, etc.) récupérés grâce au site Phishtank [phi] (pour plus de détails sur la méthode de récupération des pages légitimes-contrefaits, cf. section 5.3.2). Les premiers résultats (cf. tableau 5.5) indiquent un taux de similitude moyen entre 77 et 80% avec un écart-type variant de 15 à 26%.

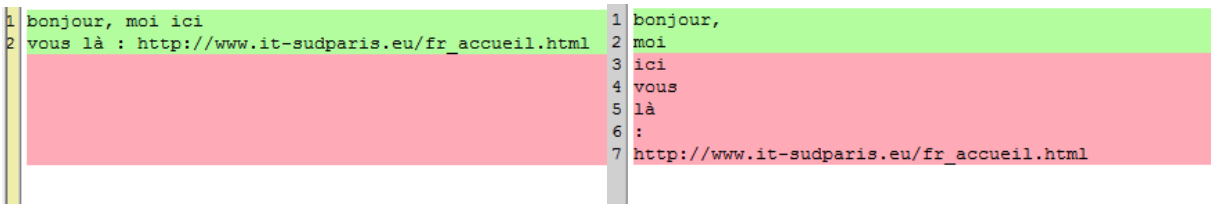
5.2.2.4.2 Approche par mots : Cette méthode de comparaison est basée sur l'approche *diff* qui appartient à la famille des algorithmes de mesure de similarité, traditionnellement utilisés pour la comparaison de textes (cf. section 4.2.1). La méthode de comparaison que nous utilisons ici - nommée *Diff* pour faciliter la lisibilité - implémente la Distance de Levenshtein.

Cette méthode *Diff* permet de calculer un taux de similitude via la comparaison des lignes contenues dans chaque page web. Elle délivre 4 indicateurs : le nombre de lignes ajoutées, modifiées, inchangées ou supprimées.

1. récupérée à partir des adresses IP retournées par le serveur DNS configuré par défaut, c.-à-d. typiquement celui du FAI auquel le client est connecté.
2. récupérée à partir des adresses IP retournées par le serveur DNS de référence utilisé dans notre solution (cf. section 5.3.1 pour plus de détails).

Dans notre approche, suite à nos premiers tests, nous avons appliqué un traitement de découpage des pages en mots (où les délimiteurs de mots sont les espaces), en amont de l'application de la méthode *Diff*. Ainsi, nous obtenons une meilleure précision du calcul de similitude. Nous pallions également tout problème de lisibilité du code source de la page (c.-à-d. le contenu de la page ne risque pas d'être interprété comme écrit sur 1 seule et même ligne par *Diff*).

A noter toutefois que ce découpage par mots ne reflète pas le nombre réel de mots contenus dans la page web (c.-à-d. tel qu'interprété par l'esprit humain). En effet, par exemple, un lien hypertexte n'est vu que comme un seul mot par la méthode de comparaison (cf. figure 5.3).



(a) Texte vu par Diff

(b) Texte vu par Diff, après découpage par mots

FIGURE 5.3 – Exemple de textes sous Diff

Dans notre utilisation de *Diff* incluant le découpage par mots, le score (ou taux de similitude) est exprimé en pourcentage. Il est calculé à partir du nombre de mots ajoutés (ajout), modifiés (modif) et supprimés (suppr). Un poids (c.-à-d. une pénalité) identique est attribué à chaque changement apporté.

$$\text{Taux de similitude (ou Score)} = \frac{(longueur(ref) - modif - suppr - ajout) \cdot 100}{longueur(ref)}$$

où *ref* représente la page de référence,

et *longueur* le nombre de mots total de l'objet spécifié

Pour être précis, il est important de noter que l'appellation "taux de similitude" que nous utilisons pour désigner le score obtenu ici est une appellation erronée. Nous l'utilisons uniquement afin d'améliorer la lisibilité et la comparaison des résultats obtenus avec l'approche par caractères.

En effet, dans le calcul de score obtenu établi ici, nous sommes en mesure d'identifier les zones de changements (ce qui n'est pas le cas avec l'approche par caractères) de manière globale. Nous avons donc délibérément choisi de privilégier le degré de divergence des deux pages comparées, plutôt que leur degré de convergence. Considérons l'exemple montré en figure 5.4 : La partie gauche affiche la page légitime (c.-à-d. la page de référence), tandis que la partie droite affiche la page contrefaite. On constate que par rapport à un total de seulement 8 mots dans le fichier légitime, il y a : 4 mots ajoutés, 1 mot modifié et 1 mot supprimé dans le fichier contrefait. Si on basait notre calcul de score sur une simple mesure de similitude, on obtiendrait un score de 75% (c.-à-d. 6 mots inchangés sur un total de 8 mots). Or, nous considérons que ce calcul ne reflète pas réellement le danger auquel est exposé l'utilisateur. En effet, 50% de code supplémentaire, potentiellement malveillant, ont été ajoutés (c.-à-d. les 4 lignes "script"). Nous préférons donc profiter de la connaissance des changements réalisés et ainsi, grâce au calcul de score que nous avons défini, le résultat de la comparaison est de seulement 25%.

Il est également important de noter que nous avons borné le taux de similitude minimum à 0%, afin d'éviter des scores négatifs aberrants.

Les premiers résultats obtenus, avec cette approche par mots appliquée aux pages légitimes-contrefaites (cf. tableau 5.6), indiquent un taux de similitude moyen variant de 30 à 39 %, avec un écart-type de l'ordre de 30%, sur pages légitimes-contrefaites.

5.2.2.5 Synthèse de l'étude préalable sur le contenu des pages webs

Les caractéristiques des pages webs légitimes et contrefaites, ainsi que les contraintes imposées par une comparaison effectuée depuis le poste client, nous conduisent à une analyse de contenu des codes sources HTML, basée sur des méthodes de comparaison de texte de type mesure de similarité.

Les premiers résultats obtenus sur des couples de pages légitimes-contrefaites semblent indiquer que l'approche par mots donne des résultats plus intéressants concernant le taux de similitude (c.-à-d. les scores les plus bas), tandis que l'approche par caractères donne des écart-types jusqu'à deux fois

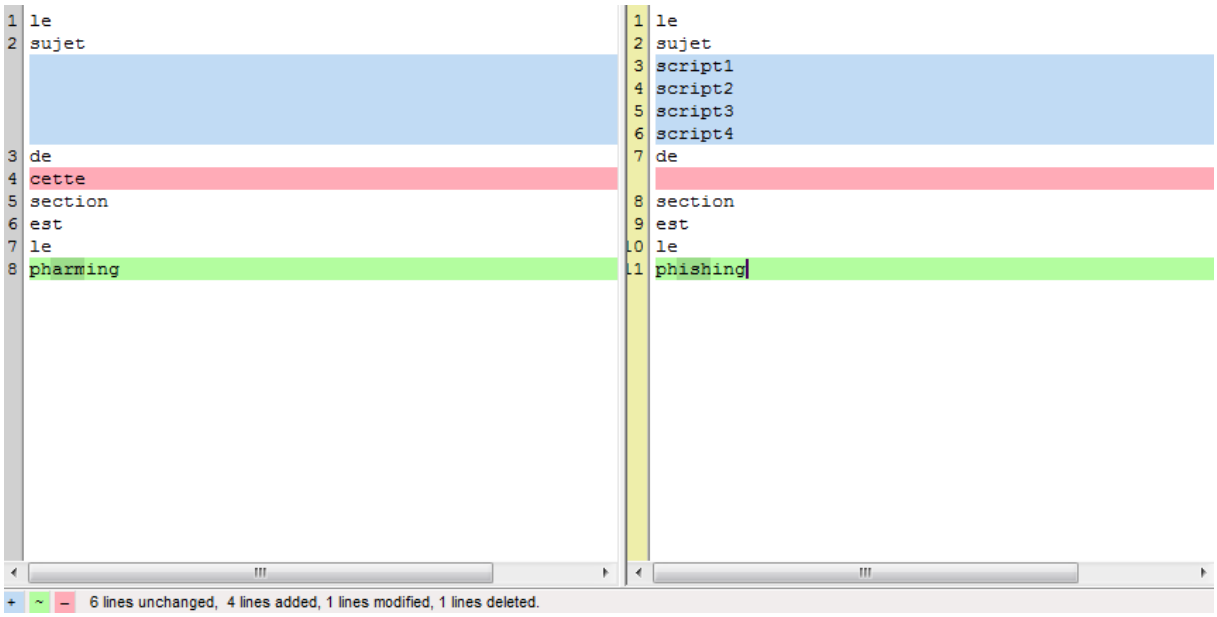


FIGURE 5.4 – Impact des modifications apportées dans le calcul de score, avec l'approche par mots

TABLEAU 5.6 – Taux de similitude de 14 à 23 couples de pages de login légitimes-contrefaites avec l'approche par mots

	Taux de similitude entre page légitime et page contrefaite (min ≤ moyenne ≤ max)	Écart-Type[2]
14 couples de pages[1]	3.16% ≤ **30.78%** ≤ 89.41%	30.20%
23 couples de pages[1]	0% ≤ **39.35%** ≤ 84%	31.63%

[1] Les deux séries de tests portent sur des couples de pages distincts
[2] entre couples de pages

moins élevés. A ce stade, il s'avère impossible de privilégier l'une des deux approches par rapport à l'autre : les valeurs minimales et maximales sont quasiment aussi extrêmes.

Les prochains tests doivent donc vérifier les premiers résultats sur une plus grande base de pages légitimes-contrefaites. Ils doivent également s'intéresser aux performances des deux méthodes de comparaison sur des couples de pages légitimes récupérées depuis une même localisation géographique.

5.3 Première proposition : vers un remplacement du nom de domaine

5.3.1 Fonctionnement général

La première étape de notre proposition (représentée par les points (1) à (4b) sur la figure 5.5) consiste à comparer l'adresse IP du domaine visité à une adresse IP (ou un pool d'adresses IP) dite de référence.

A chaque fois que l'utilisateur accède à une page web de login, le FQDN est extrait de l'URL visitée. Une requête DNS est alors envoyée à deux serveurs DNS : le serveur par défaut (nommé *DNSdef*) et un serveur de référence (nommé *DNSref*), en vue de comparer les adresses IP retournées pour le FQDN concerné. Le serveur DNS par défaut retourne plusieurs adresses IP, dont celle utilisée pour l'affichage de la page web dans le navigateur (nommée(s) *IPdef*). DNSref retourne, quant à lui, une ou plusieurs adresses IP (nommée(s) *IPref*), incluant ou excluant IPdef.

Dans le cas où l'adresse IPdef est incluse dans la réponse IPref, le site est considéré comme légitime. Dans le cas contraire, nous passons à la seconde étape de notre proposition : l'analyse de la page web.

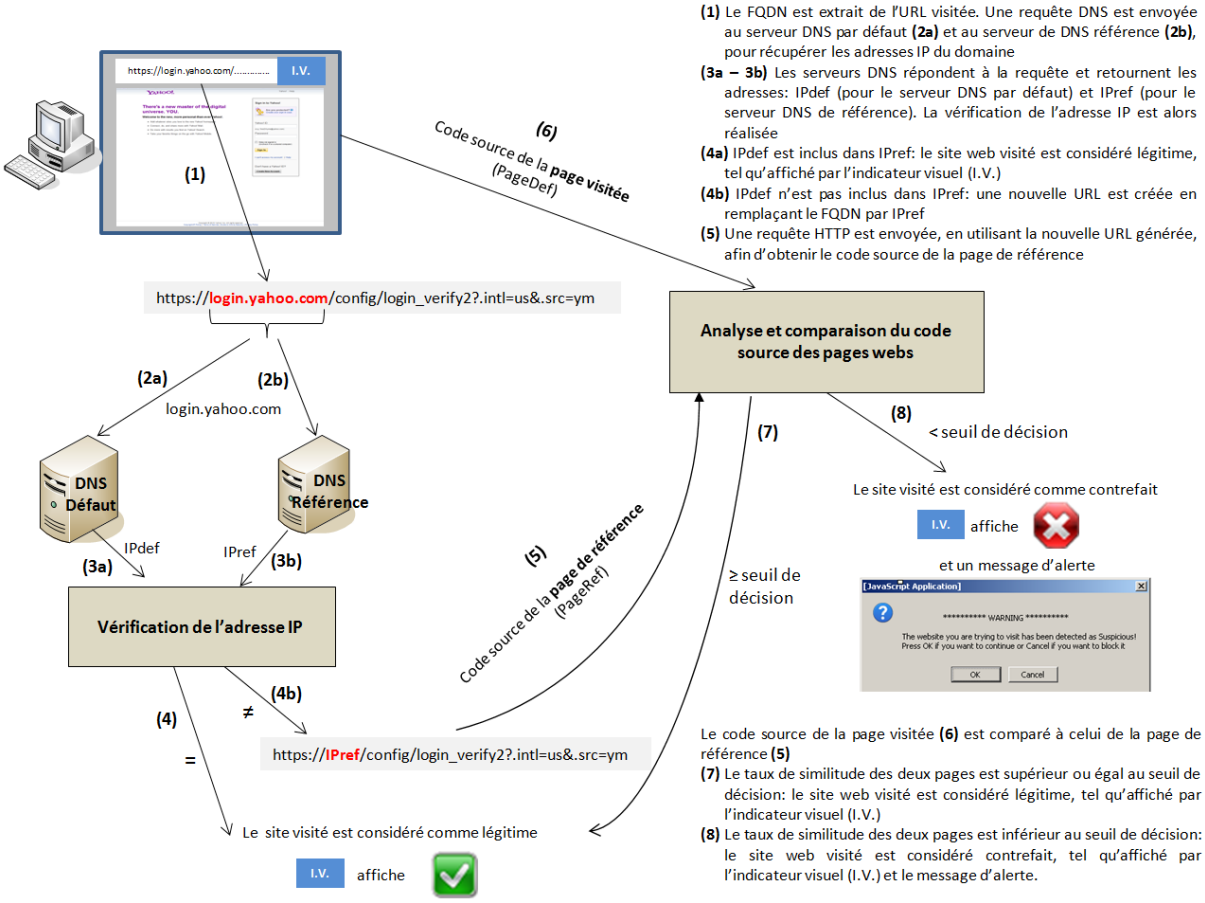


FIGURE 5.5 – Première approche : fonctionnement général

Définition du serveur de référence : Dans le schéma d'intégration envisagé pour notre proposition, nous prévoyons d'inclure une étape de configuration du serveur DNS de référence. Ce choix, laissé à ce stade à l'appréciation de l'utilisateur, se fait parmi une liste de serveurs DNS ouverts alternatifs proposés (p.ex. OpenDNS, GoogleDNS, DNSAdvantage [dns]).

La seconde étape de notre proposition se concentre sur l'analyse et la comparaison de la page web visitée.

L'étude de travaux précédents sur la comparaison de pages webs dans le cadre du phishing [WHX+05] [RKKF07] [MKK08] [HYM09], nous a montré que ces comparaisons sont essentiellement basées sur des bases de données de référence. Les inconvénients majeurs de ce type d'approche sont à la fois le maintien d'une base de données de référence à jour, ainsi que la préservation de sa confidentialité et de son intégrité.

Dans notre approche - de la même manière que nous avions écarté les listes noires dans l'analyse du phishing -, nous avons donc considéré comme essentiel que, bien qu'utilisant une comparaison de page web envers une page dite de référence, il ne devait y avoir aucun stockage de ces pages considérées comme légitimes.

La seconde étape de notre proposition se déroule donc de la manière suivante (représentée par les points (5) à (8) sur la figure 5.5) :

- la page web visitée (nommée *PageDef*), telle qu'affichée dans le navigateur, est récupérée à partir de IPdef.
- La page de référence (nommée *PageRef*) est, quant à elle, récupérée grâce à une nouvelle URL générée en utilisant IPref. En effet, le FQDN de l'URL d'origine est alors remplacé par IPref. Considérons l'exemple suivant : l'URL visitée par l'Internaute via son navigateur est `https://www.amazon.com/gp/yourstore?ie=UTF8&ref_=pd_irl_gw&signIn=1`. Le serveur DNS par défaut de

l'utilisateur retourne l'adresse IP : 72.21.210.250 (IPdef), tandis que le serveur DNS de référence interrogé retourne l'adresse IP : 207.171.166.252. La nouvelle URL utilisée pour la récupération de la page référence (PageRef) est alors : `https://207.171.166.252/gp/yourstore?ie=UTF8&ref_=pd_irl_gw&signIn=1`.

Les codes sources HTML des deux pages webs (PageDef et PageRef) sont ensuite comparés (cf. figure 5.5) en utilisant une des techniques énoncées en section 5.2.2. Enfin, la légitimité de la page web visitée est déterminée en comparant le pourcentage de similitude obtenu entre les deux pages (PageDef et PageRef) et un seuil de décision pré-défini.

Les sections suivantes de ce Chapitre 5.3 visent à détailler les expérimentations réalisées, ainsi qu'à exposer les résultats obtenus.

5.3.2 Conditions d'expérimentation

Pour évaluer l'efficacité de notre proposition, nous avons réalisé deux types de comparaison de pages webs :

- des comparaisons de couples de pages de login[1] légitimes récupérées, à partir des informations émises par plusieurs serveurs DNS, depuis différentes localisations géographiques.
- des comparaisons de couples de pages de login[1] légitimes-contrefaites, visuellement très similaires.

L'ensemble des pages utilisées pour ces tests ont été collectées entre Décembre 2010 et Janvier 2011.

5.3.2.1 Couples de pages légitimes

Les couples de pages de login légitimes ont été récupérées depuis 11 localisations géographiques différentes, réparties sur 5 continents.

Pour chaque localisation, nous avons récupéré les pages webs obtenues grâce aux adresses IP retournées par le serveur par défaut ainsi que par 3 serveurs de référence (OpenDNS, GoogleDNS et DNSAdvantage).

Sur chaque site géographique, nous avons testé un set de 108 URLs de login, dont 104 sont issues de FQDN distincts. Les principaux critères de sélection de ces URLs ont été : diversifier les secteurs d'activités, multiplier les TLD et les langues dans lesquelles les pages sont écrites (anglais, français, espagnol, arabe, russe, etc.). La majorité des URLs de sites bancaires ont été récupérées à partir du site *Levoyageur* [lev].

Classification des 108 URLs de login : Nous avons classifié les URLs utilisées en 5 secteurs d'activités : banques, réseaux sociaux, e-commerce, email, et autres (cf. Table 5.7). La catégorie *autres* comprend des sites provenant des domaines de l'assurance, l'administration, les jeux en ligne, les universités, les plateformes de partage vidéos et photos, des sites d'information, des sites de support produits logiciels ou divers sites issus de l'industrie (p.ex. constructeur automobile, panneaux solaires, etc.).

A noter que nous avons constaté que plusieurs sites, tels que celui de Facebook (réseaux sociaux) semblent utiliser une connexion non sécurisée pour leur page d'authentification (celle-ci s'affiche en *http*, p.ex. `http://fr-fr.facebook.com/`). Néanmoins, à l'aide d'un analyseur de protocoles, nous avons constaté que l'envoi des données d'authentification de l'abonné s'effectue pourtant bien de manière sécurisée. Cet envoi se fait de façon totalement transparente pour l'utilisateur, via l'URL `https://login.facebook.com/`. Dans ce type de cas, nous avons donc préféré intégrer l'URL réellement utilisée pour l'envoi des données (c.-à-d. celle en HTTPS) dans notre set de 108 URLs.

Les URLs sélectionnées sont issues de 20 TLDs différents, divisables en 2 catégories : les cc-TLD et les g-TLD. Pour une meilleure lisibilité, les TLDs sélectionnés ont été regroupés par continent (cf. Table 5.8).

Une large majorité des sites sélectionnés utilisent le TLD générique COM (pour *Commercial*), suivis par les TLDs d'Europe, d'Asie, d'Océanie et le TLD générique NET (pour *Network*). Enfin, quelques TLDs sont issus d'Amérique du Nord, d'Amérique du Sud, d'Afrique et du TLD générique ORG (pour *Organization*).

1. Les raisons de la focalisation de notre étude sur les pages de login sont expliquées en section 5.2.1.5

Tableau 5.7 – Répartition des 108 URLs de login légitimes par secteur d'activités

Catégories	Quantité	Pourcentage
Banques	53	49%
Autres (administration, assurances, logiciels, jeux, FAI, industrie, vidéos, photos, informations)	37	34%
e-commerce	8	7%
Réseaux sociaux	5	5%
email	5	5%

Tableau 5.8 – Répartition des 20 TLDs des 108 URLs de login légitimes

	Pourcentage	TLD
Europe	37.0%	AD, AT, BE, DE, DK, FR, IT, LV, UK
Asie	5.6%	IN, MV, TR
Océanie	3.7%	AU, NZ
Amérique du Nord	0.9%	CA
Amérique du Sud	0.9%	MX
Afrique	0.9%	ZA
Commercial	46.3%	COM
Network	3.7%	NET
Organization	1.0%	ORG

Répartition géographique des tests : Les tests ont été effectués depuis 11 localisations géographiques, réparties sur 5 continents (cf. figure 5.6) :

- Europe : France (4 tests en Île de France et 1 test dans le Sud de la France)
- Amérique du Nord : Mexique
- Amérique du Sud : Vénézuela
- Asie : Chine, Émirats Arabes Unis
- Afrique : Tunisie

Pour chaque localisation, nous nous sommes assurés que la configuration DNS par défaut était différente des 3 DNS de référence que nous avons sélectionnés.

La récupération des pages webs auprès des 4 serveurs DNS (c.-à-d. le serveur DNS par défaut et les 3 serveurs de référence) a été automatisée grâce au programme que nous avons développé en Java (cf. section 5.3.3). Il est important de noter que lors des tests, sur certaines localisations géographiques (p.ex. Sénégal, Vénézuela), nous avons rencontré des pertes de connectivité (rupture de courant électrique et/ou Internet) assez fréquentes. Par conséquent, les taux d'erreur de récupération des pages sur ces localisations sont notablement plus conséquents que pour les autres localisations testées.

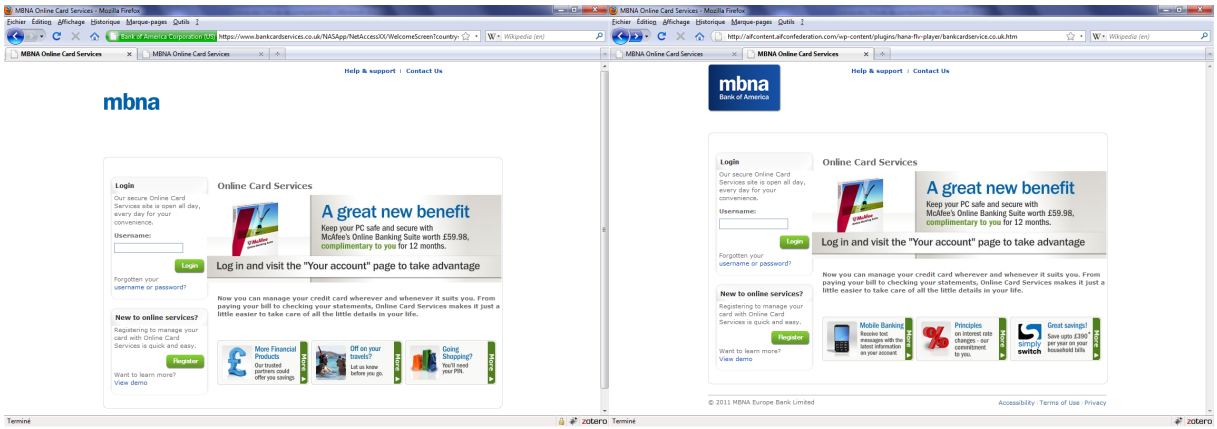
5.3.2.2 Couples de pages légitimes-contrefaites

Une des difficultés essentielles de notre expérimentation est de trouver des pages contrefaites réelles et en ligne (en effet leur durée de vie est très courte, cf. section 2.3) pouvant servir pour des attaques de pharming, ainsi que leur pendant légitime. Nous nous sommes donc appuyés sur les sites de l'APWG [apw] et de Phishtank [phi] qui délivrent des bases d'URLs de phishing pour récupérer nos pages contrefaites. Nous avons choisi de nous placer dans le cas le plus défavorable, en basant notre sélection sur ces 2 critères :

- Sélectionner des sites de phishing "validés" à 100%. En effet, les bases de données communiquées par l'APWG et Phishtank indiquent un niveau de confiance (c.-à-d. il est avéré et vérifié que l'URL présentée est un site de phishing), variant de 50 à 100% pour les URLs mises à disposition.
- Ne sélectionner que des sites de phishing pour lesquels nous trouvons le site légitime associé, visuellement très similaire (c.-à-d. même structure visuelle, cf. exemple en figure 5.7).

Nous avons ainsi collecté 37 couples de pages légitimes-contrefaites, chaque couple de page ayant été récupéré depuis une même machine et un même navigateur web.

FIGURE 5.6 – Première approche : répartition géographique des tests



(a) Site légitime (b) Site contrefait

FIGURE 5.7 – Captures d'écran du site légitime MBNA (`https://www.bankcardservices.co.uk/NASApp/NetAccessXX/...`) et d'une contrefaçon (`http://aifcontent.aifconfederation.com/wp-content/plugins/hana-flv-player/bankcardservice.co.uk.htm`), récupérées le 11 Juin 2011

Algorithme 1 Première approche : DNS par défaut

Entrées: le fichier *.txt* contenant la liste des n URLs de login.
Sorties: 1 fichier contenant les n réponses du serveur par défaut, 1 fichier contenant les scores des n pages par défaut, et n fichiers contenant les codes sources des pages webs par défaut.

1: **pour** $i = 0$ **à** n **faire**
2: Requête DNS auprès du serveur par défaut pour le FQDN(i).
3: Sauvegarde des adresses IP (*IPdef*) retournées pour le FQDN(i) (un même fichier pour les n URLs).
4: Récupération et sauvegarde du code source de la page web par défaut en utilisant l'URL `https://FQDN(i)/arborescence1/.../arborescenceX/fichier.html` (1 fichier par URL, nommé *date_heure_def_FQDN*(i)*.txt*).
5: Calcul du score de la page(i) récupérée avec l'approche par caractères, puis sauvegarde du résultat dans un fichier *.txt* (1 fichier pour les n URLs).
6: **fin pour**

L'ensemble des pages contrefaites ainsi collectées nous ramène à nos 5 catégories de secteurs d'activités (vus pour les pages légitimes) : majoritairement des banques (p.ex. Bank of America, Paypal, Chase, etc.), des sites d'e-commerce (p.ex. eBay, etc.), d'email (p.ex. Hotmail), de réseaux sociaux (p.ex. Facebook) ou autres (p.ex. jeux en ligne avec RuneScape).

5.3.3 Implémentation : aperçu général

L'implémentation de notre approche, visant à récupérer les couples de pages légitimes, a été réalisée en Java. Un certain nombre de fonctions pré-existantes, assez facilement utilisables, sont disponibles dans les librairies Java (p.ex. *java.net*), tant pour générer une requête DNS vers le serveur DNS par défaut que pour récupérer une page web en HTTP.

Une des premières difficultés rencontrées dans l'implémentation a été de pouvoir générer une requête DNS vers un serveur dit de référence. Pour ce faire, nous avons développé une fonction spécifique (cf. section 5.3.4.1). En outre, plusieurs mois d'essais ont été nécessaires pour éliminer toute erreur (ou presque) liée à cette seconde requête DNS de référence (cf. section 5.3.5.3).

Une deuxième étape importante a été de savoir quelle adresse avait été utilisée pour récupérer la page web. En effet, le DNS retourne les adresses connues pour le domaine interrogé, à un instant t. Mais au moment de la requête HTTP, ce n'est pas forcément la première adresse IP retournée par le serveur DNS qui est utilisée. Dans la seconde approche (développée en section 6.1), nous avons pu répondre avec certitude à cette question.

Une troisième difficulté rencontrée a été de pouvoir récupérer une page de login en HTTPS. Il fallait en effet pouvoir passer les étapes d'établissement de la connexion SSL/TLS avant de récupérer la page web requise.

Enfin, une quatrième étape a concerné la réécriture de l'URL d'origine, dans laquelle le FQDN est remplacé par l'adresse IP *IPref*.

Dans cette première approche, seul le calcul de score par caractères a été calculé automatiquement, au téléchargement des pages webs. L'approche par mots, ajoutée plus tardivement, a été appliquée aux pages séparément, après téléchargement.

La liste des URLs à récupérer est placée dans un fichier *.txt*, ordonnancé par secteur d'activités des URLs.

5.3.3.1 Traitement effectué durant les tests

Cette étape consiste en la récupération des adresses IP *IPdef* et *IPref*, des pages webs associées, et au calcul de score utilisant l'approche par caractères.

Suite à des problèmes d'implémentation, nous avons été obligés de scinder les traitements associés à DNSdef et DNSref en deux programmes séparés (cf. algorithmes 1 et 2), pour les raisons évoquées ultérieurement dans ce document (cf. section 5.3.4.3).

Algorithme 2 Première approche : DNS de référence

Entrées: le fichier *.txt* contenant la liste des n URLs de login.
Sorties: 1 fichier contenant les n réponses du serveur de référence, 1 fichier contenant les scores des n pages de référence, et n fichiers contenant les codes sources des pages webs de référence.

1: **pour** $i = 0$ **à** n **faire**
2: Requête DNS auprès du serveur de référence pour le FQDN(i).
3: Sauvegarde des adresses IP (*IPref*) retournées pour le FQDN(i) (un même fichier pour les n URLs).
4: Récupération et sauvegarde du code source de la page web de référence en utilisant l'URL `https://IPref/arborescence1/.../arborescenceX/fichier.html` (1 fichier par URL, nommé *date_heure_nomDNSref_FQDN*(i)*.txt*). A noter que l'adresse IPref utilisée est la première adresse IP retournée par le serveur DNS de référence.
5: Calcul du score de la page(i) récupérée avec l'approche par caractères, puis sauvegarde du résultat dans un fichier *.txt* (1 fichier pour les n URLs).
6: **fin pour**

Ensuite, le taux de similitude des deux pages est calculé sous Excel, à l'aide des deux fichiers de score générés précédemment (selon la méthode expliquée en section 5.2.2.4.1).

5.3.3.2 Traitement effectué après les tests

Cette étape consiste au traitement des pages webs récupérées précédemment (*PageDef* et *PageRef*) avec l'approche par mots.

Pour ce faire, nous avons été obligés de procéder au réordonnancement des fichiers de pages webs, afin de pouvoir procéder à des comparaisons justes et comparables aux scores obtenus avec l'approche par caractères. En effet, lors de la sauvegarde du code HTML, nous avions choisi d'horodater les fichiers, en incluant la date et l'heure de récupération, afin de faciliter l'archivage des pages récupérées (p.ex. *20110126_15.01.18_def_login.live.com.txt* est le nom de fichier pour l'URL contenant le FQDN *login.live.com* récupéré auprès du serveur par défaut *Def*, le 26/01/2011 à 15h01min18s.). Or, dans notre implémentation, nous récupérons l'ensemble des URLs auprès d'un même serveur DNS, avant de passer au DNS suivant. Les temps de réponse des serveurs webs interrogés pouvant être très variables, nous avons sauvegardé les pages webs dans un ordre parfois très différent de celui dans lequel les requêtes avaient été émises. Nous avons pallié ce problème en supprimant l'indication horaire dans les noms de fichiers sauvegardés. Ainsi l'ensemble des fichiers - quel que soit le serveur DNS interrogé - ont été classés, à l'identique, par ordre alphabétique des FQDNs. A noter que les développements menés lors de la seconde approche ont corrigé ce problème.

L'approche par mots telle que nous l'avons envisagée (cf. section 5.2.2.4.2) a été implémentée en trois parties :

- La première partie consiste en une méthode de découpage par mots des pages webs, dans laquelle les délimiteurs sont des espaces. Nous avons développé cette méthode à l'aide de la classe Java *TextIO*.
- La deuxième partie consiste en la comparaison des pages et l'identification des zones de changements (ajouts, modifications, suppressions). Pour ce faire, nous avons utilisé une implémentation Java de *Diff* pré-existante [Gat].
- Enfin, la troisième partie est le calcul de score réalisé grâce à l'extraction et à la réutilisation des zones de changements identifiées précédemment.

Contrairement à l'approche par caractères, un seul fichier de score est généré : celui contenant le taux de similitude des deux pages.

A noter que les analyses par mots et par caractères ont toutes deux été appliquées aux couples de pages légitimes-contrefaites selon les mêmes principes que décrits ici.

5.3.4 Vérification de l'adresse IP du domaine visité

Les résultats de tests de la vérification de l'adresse IP se concentrent sur les FQDN uniques, à savoir : 104 des 108 URLs légitimes sélectionnées (cf. section 5.3.2.1).

5.3.4.1 Implémentation : points spécifiques

Les requêtes vers le serveur DNS par défaut sont faites grâce à la fonction Java *getAllByName*, présente nativement dans la classe *InetAdress*.

Les requêtes vers le serveur DNS de référence sont quant à elles réalisées grâce à l'utilisation d'une librairie particulière, DNSJava [lib]. Cette bibliothèque, qui est une implémentation de DNS en Java, propose des fonctions similaires à la classe InetAddress. En supplément, elle apporte des fonctions additionnelles telles que le transfert de zone, DNSSEC, etc. Pour forger une requête DNS, on utilise un objet *SimpleResolver* qui représente le serveur DNS à contacter (il contient le nom du serveur, son adresse IP et le port destination (53)). Ensuite, on créé un *Lookup* qui correspond aux caractéristiques de la requête à envoyer. Elle contient le domaine interrogé, le type d'enregistrement demandé ("A" pour nom d'hôte/adresse IPv4) et sa classe ("IN" pour protocole Internet). Reste alors à affecter le *Lookup* au *SimpleResolver*, puis lancer la requête.

5.3.4.2 Résultats

5.3.4.2.1 Variabilité du nombre de réponses DNS : L'exploitation des résultats DNS s'est avéré difficile. En effet, selon les sites testés, les DNS interrogés, les heures et dates des tests, etc., une à plusieurs adresses IPs peuvent être retournées par chacun des serveurs DNS.

Difficile donc d'automatiser une comparaison, puisqu'aucune indication préalable ou certitude n'était acquise sur les tailles des espaces d'adresses IP récupérés. Néanmoins, dans la seconde approche développée en section 6.1, nous avons réussi à optimiser ce traitement.

5.3.4.2.2 Variabilité des adresses IP utilisées par les pages de login : Les résultats IP énoncés dans le tableau 5.9 sont obtenus en vérifiant si au moins 1 adresse *IPdef* figure dans les résultats *IPref* du serveur de référence. Prenons trois exemples, sur une même localisation donnée (France) :

- Pour le FQDN `login.yahoo.com`, DNSdef retourne les adresses IP 217.12.8.76 et 217.146.187.123, tandis que DNSref retourne l'adresse IP 69.147.112.160. Dans ce cas, nous considérons une convergence de 0.
- Pour le FQDN `www.paypal.com`, DNSdef retourne les adresses IP 66.211.169.65, 64.4.241.33, 64.4.241.49 et 66.211.169.2, tandis que DNSref retourne les adresses IP 64.4.241.49, 66.211.169.2, 66.211.169.65 et 64.4.241.33. Ici, la convergence est de 4.
- Pour le FQDN `www.halifax-online.co.uk`, DNSdef retourne l'adresse IP 212.140.245.71, tandis que DNSref retourne les adresses IP 62.172.43.199, 212.140.245.11, 62.172.43.131 et 212.140.245.71. Dans ce cas, la convergence est de 1.

Le taux de convergence global est ensuite obtenu de la manière suivante : pour l'ensemble des FQDN interrogés, nous additionnons l'ensemble des convergences ≥ 1, que nous divisons par le nombre total de FQDN pour lesquels nous avons des réponses DNS (c.-à-d. absence d'erreurs DNS).

Les résultats obtenus tendent à indiquer que, comme supposé en section 5.2.1.5, les adresses IP utilisées par des sites de login sont plus convergentes que pour les FQDN génériques. Pour rappel, la moyenne de convergence des adresses IP sur les FQDN génériques était de l'ordre de 71 à 73% avec des écart-types (entre localisations) oscillants entre 10 et 14% (cf. section 5.2.1). Ici, l'écart-type (entre localisations) est considérablement réduit (3 à 5%), les moyennes des taux de convergence oscillent autour de 81 à 82%. Enfin, l'intervalle de confiance à 95% (c.-à-d. l'incertitude d'estimation, entre localisations) varie entre 79 à 85%. Nous constatons que les 3 DNS de référence donnent le même type de résultats, avec un léger avantage de convergence pour OpenDNS.

Il est important de noter que ces résultats ne signifient par pour autant, qu'en moyenne, 81 à 82% des pages HTML ont été récupérées auprès du même serveur web (c.-à-d. même adresse IP). En effet, à ce stade, nous n'avons aucune certitude sur l'adresse IP utilisée pour la récupération de PageDef, de par la fonction utilisée (cf. section 5.3.5.3).

Tableau 5.9 – Comparaison des réponses du DNS par défaut vs. 3 serveurs DNS de référence, sur 11 localisations géographiques

	Taux de convergence avec les adresses IP retournées par le serveur DNS par défaut (min ≤ moyenne ≤ max)	Écart-Type [1]	Intervalle de confiance à 95% [1]
OpenDNS	76.47% ≤ **82.90%** ≤ 95.19%	5.00%	[79.95% ; 85.85%]
GoogleDNS	76.92% ≤ **81.61%** ≤ 91.35%	3.62%	[79.47% ; 83.75%]
DNSAdvantage	74.51% ≤ **81.22%** ≤ 89.42%	4.10%	[78.80% ; 83.64%]

[1] entre localisations

Tableau 5.10 – Taux d'échec des requêtes DNS auprès des 4 serveurs DNS, sur 11 localisations géographiques

	Taux d'échec des requêtes DNS (min ≤ moyenne ≤ max)	Écart-Type [1]
Défaut	0% ≤ **0.08%** ≤ 0.96%	0.29%
OpenDNS	0% ≤ **0.16%** ≤ 1.92%	0.58%
GoogleDNS	0% ≤ **0.80%** ≤ 9.62%	2.90%
DNSAdvantage	0% ≤ **0.24%** ≤ 0.96%	0.45%

[1] entre localisations

5.3.4.2.3 Taux d'échec des requêtes DNS : Sur les 11 localisations testées, nous obtenons des taux d'erreur des requêtes DNS très faibles (cf. tableau 5.10). En effet, nous obtenons au maximum 1 à 2 erreurs (sur les 104 URLs testées) par site géographique. Une exception est à noter toutefois : sur le site du Vénézuela, nous atteignons un taux d'échec de 9.62% avec GoogleDNS, ce qui représente 10 erreurs. Ceci s'explique par les pertes de connectivité rencontrées lors de la réalisation des tests (cf. explications en section 5.3.2.1).

5.3.4.3 Problème majeur rencontré à l'implémentation : la réinitialisation du cache DNS

Le problème majeur que nous avons rencontré à l'implémentation a concerné la possibilité de faire des requêtes vers différents serveurs DNS, au sein d'un même programme Java. En effet, les requêtes vers le DNS par défaut se déroulaient sans aucun problème mais, dès lors que nous faisions appel à un second serveur DNS, nous étions confrontés à des comportements aléatoires. Grâce à une analyse du trafic, nous nous sommes aperçus que les nouvelles requêtes DNS étaient aléatoirement générées. Ainsi, nous utilisions parfois les réponses DNS du serveur par défaut en tant qu'adresse *IPref*, ce qui faussait une partie de nos résultats.

Confrontés à un problème de réinitialisation du cache DNS, nous avons essayé de trouver son origine, à savoir : problème de configuration du système d'exploitation ou problème Java.

Dans un premier temps, nous avons identifié un paramètre susceptible de permettre la réinitialisation du cache DNS Java : *networkaddress.cache.ttl*.

Trois méthodes ont donc été testées pour permettre la réinitialisation du cache DNS Java :

Méthode N° 1 - Ajout d'une commande de réinitialisation du cache au sein du programme Java : Le cache *networkaddress.cache.ttl* peut être configuré selon trois types de valeur : 0 = absence de cache, -1 = cache infini, >0 = valeur de cache en secondes. Nous avons donc forcé la valeur du cache à 0. A l'exécution du programme nous avons cependant constaté que la valeur de cache restait inchangée à sa valeur par défaut : 30 s.

Méthode N° 2 - Modification du paramètre de cache dans le logiciel Java : La modification du cache *network.cache.ttl* est possible par manipulation du fichier *java.security*, accessible dans le réper-

Algorithme 3 Programme pour l'identification du Bug DNS Java

1: Récupération de la page web par défaut en utilisant l'URL `https://FQDN(i)/arborescence1/.../arborescenceX/fichier.html`.
2: Requête vers le DNS Référence afin de récupérer *IPref*.
3: Récupération et sauvegarde du code source de la page web de référence en utilisant l'URL `https://IPref/arborescence1/.../arborescenceX/fichier.html`.

toire d'installation %JRE_HOME%/lib/security/. A nouveau, malgré un cache paramétré à 0, nous avons constaté que celui demeurait toujours bloqué à 30 s.

Méthode N° 3 - Positionnement de la valeur de cache, en ligne de commande, à l'exécution du programme : Le cache peut être forcé à 0 via la commande *java -jar -Dsun.net.inetaddr.ttl=0 nom_programme.jar*. Là encore, aucune amélioration à l'exécution du programme.

Après investigation du côté des problèmes connus, nous avons identifié un bug Java susceptible de correspondre à notre problème [Ora05]. En effet, il met en avant l'inefficacité de la réinitialisation du cache DNS. Toutefois ce bug est mentionné comme "State 11-Closed, Not a Defect, bug". Il est ramené à un problème de paramétrage du cache DNS au niveau du système d'exploitation.

Toutefois, via une implémentation alternative, nous avons bel et bien constaté que ce problème est inhérent au Java.

En effet, théoriquement, une implémentation logique serait d'intégrer la totalité de notre programme en un seul et unique exécutable Java, constitué de 4 étapes successives : 1/ Requête DNSdef, 2/ Récupération de PageDef, 3/ Requête DNSref et 4/ Récupération de PageRef. Néanmoins, pour les raisons évoquées précédemment, cela s'avère impossible. Afin de faire la distinction entre bug Java ou paramétrage du cache DNS au sein du système d'exploitation, nous avons créé un programme alternatif (cf. algorithme 3). Dans celui-ci, la requête DNSdef n'est pas explicitement demandée par le programme Java, mais automatiquement générée par le fonctionnement normal de la pile TCP/IP induit par l'étape 1 de cet algorithme. Nous constatons alors que la requête DNSref, lancée à l'étape 2 de l'algorithme, fonctionne parfaitement.

Le problème rencontré est donc bien lié à un problème de réinitialisation du cache associé à la fonction *java.net.InetAddress.getByName()* utilisée pour la requête au DNS par défaut.

Il aurait alors paru simple d'appliquer la librairie DNSJava, utilisée pour la requête DNSref, à la requête DNSdef (cf. section 5.3.4.1). Toutefois, cela s'avère difficile. En effet, les fonctions proposées impliquent de spécifier l'adresse IP du DNSdef. Ceci parait difficilement utilisable en environnement réel, c.-à-d. demander à un utilisateur de spécifier l'adresse de son DNSdef. De plus, cela devient problématique en cas de mise à jour automatique de l'adresse DNSdef par le FAI.

Pour les deux approches développées dans notre contribution, nous avons donc été amenés à scinder le programme en plusieurs morceaux. Nous avons ainsi créé un exécutable par serveur DNS interrogé, afin d'éliminer tout problème de cache DNS Java.

5.3.4.4 Synthèse sur la vérification de l'adresse IP

Au travers des résultats obtenus dans cette première proposition, nous confortons les hypothèses de travail initiales, à savoir : une certaine staticité des adresses IP associées aux pages de login. Ceci peut notamment s'expliquer par la gestion de certificat associée à un FQDN, impliquant la pré-définition d'adresses IP réservées à cet usage.

Les trois serveurs DNS de référence interrogés donnent sensiblement le même type de résultats.

Les taux d'échec des requêtes DNS sont minimes et ce, quel que soit le serveur utilisé (les résultats GoogleDNS sont impactés par les pertes de connectivité rencontrées au Vénézuela. En effet, les 10 autres localisations ne présentent aucun échec).

Nous constatons également que le nombre de réponses DNS n'est pas forcément directement lié à l'idée que nous nous faisons de la taille du domaine interrogé.

Tableau 5.11 – Taux de similitude des 108 pages légitimes avec l'approche par caractères, sur 11 localisations géographiques

	APPROCHE PAR CARACTÈRES		
	Taux de similitude avec la page issue de IPdef (min ≤ moyenne ≤ max)	Écart-Type[1]	Intervalle de confiance à 95%[1]
OpenDNS	98.14% ≤ **98.63%** ≤ 98.96%	0.26%	[98,48% ; 98.78%]
GoogleDNS	98.14% ≤ **98.68%** ≤ 98.96%	0.27%	[98,52% ; 98.84%]
DNSAdvantage	98.22% ≤ **98.71%** ≤ 99.26%	0.32%	[98.52% ; 98.90%]

[1] entre localisations

Nous en déduisons donc que la vérification de l'adresse IP semble une étape valable dans notre processus d'identification des attaques de pharming. Néanmoins, les résultats obtenus ici doivent être vérifiés sur davantage d'URLs. De plus, cette étape ne peut être le seul indicateur de légitimité du site visité.

Enfin, dans l'intégration envisagée pour notre solution, le problème identifié pour la réinitialisation du cache DNS Java peut s'avérer problématique. Ce point devra donc être étudié avec une attention toute particulière.

5.3.5 Analyse et comparaison du code source des pages webs

Les tests de cette partie portent sur la totalité des 108 URLs légitimes sélectionnées (cf. section 5.3.2.1).

5.3.5.1 Implémentation : points spécifiques

Contrairement à l'étude préalable effectuée, qui portait sur des URLs HTTP, nous nous sommes focalisés ici sur la comparaison d'URLs de login en HTTPS. Ceci implique de passer avec succès les étapes d'établissement de la connexion SSL/TLS, avant de pouvoir récupérer la page web souhaitée.

En préambule du GET HTTP (qui récupère la page web), nous avons donc implémenté une fonction spécifique *https3.gethttps* qui réalise l'établissement de la connexion sécurisée grâce à la librairie Java *javax.net.ssl*. Les deux éléments essentiels de cette fonction sont : l'acceptation de tout certificat proposé, ainsi que la non-vérification du "hostname" (c.-à-d. le FQDN).

Une autre étape importante est la génération de la nouvelle URL utilisant IPref. Pour ce faire, nous avons utilisé la première adresse IP retournée par le serveur DNS de référence.

5.3.5.2 Résultats

5.3.5.2.1 Taux de similitude des pages légitimes : Les tableaux 5.11 et 5.12 indiquent les scores obtenus avec les deux approches, pour la comparaison des pages issues de IPdef vis-à-vis de celles fournies par IPref. Il apparaît que l'approche par caractères semble donner les meilleurs résultats : le taux de similitude y est le plus élevé (en moyenne 98 %) et les écart-types entre localisations les plus minimes (autour de 0.30%). Néanmoins, l'approche par mots donne également de très bon résultats avec un taux de similitude autour de 91% et des écart-types entre localisations assez faibles (autour de 3%).

Nous constatons également peu de variabilité des résultats associés à la provenance de l'adresse IPref. Autrement dit, les 3 serveurs DNS conduisent à des résultats très similaires.

5.3.5.2.2 Taux de similitude des pages légitimes-contrefaites : Concernant la différenciation des pages légitimes-contrefaites, il apparaît très clairement que la méthode par mots est la plus intéressante (cf. tableau 5.13). En effet, c'est l'approche qui donne les scores les plus intéressants (c.-à-d. les plus bas) avec un taux de similitude moyen de 56%, contre 80% avec l'approche par caractères. En comparaison avec les résultats de l'étude préalable, on constate également que l'approche par caractères perd en avantage sur l'écart-type : nous obtenons ici 23% pour cette dernière et 27% pour l'approche par mots.

Tableau 5.12 – Taux de similitude des 108 pages légitimes avec l'approche par mots, sur 11 localisations géographiques

	APPROCHE PAR MOTS		
	Taux de similitude avec la page issue de IPdef (min ≤ moyenne ≤ max)	Écart-Type[1]	Intervalle de confiance à 95%[1]
OpenDNS	81.96% ≤ **91.56%** ≤ 93.57%	3.24%	[89.65% ; 93,51%]
GoogleDNS	81.61% ≤ **91.50%** ≤ 94.53%	3.46%	[89.46% ; 91.50%]
DNSAdvantage	82.20% ≤ **91.38%** ≤ 92.76%	3.13%	[89.53% ; 93.23%]

[1] entre localisations

Tableau 5.13 – Taux de similitude des 37 couples de pages légitimes-contrefaites

	Taux de similitude entre pages légitimes et contrefaites (min ≤ moyenne ≤ max)	Écart-Type[1]	Intervalle de confiance à 95%[1]	Taux de faux-négatifs si le seuil de décision est 80%	90%	Approche déterminante pour % couples de sites
CARACTÈRES	19.57% ≤ **80.37%** ≤ 99.93%	23%	[72.95% ; 87.79%]	59.46%	24.32%	2.70%
MOTS	2% ≤ **56.76%** ≤ 92%	27.42%	[29.34% ; 65.59%]	2.70%	-	97.30%

[1] entre couples de pages

Les taux de similitude maximum obtenus avec les deux approches semblent assez voisins (92 ou 99%). Néanmoins, dans l'objectif de définir un seuil de décision, nous avons commencé à nous intéresser au taux de faux-négatifs (FNR - False Negative Rate) potentiel. Il apparaît alors qu'avec un seuil fixé à 80%, un fossé se creuse entre les deux méthodes. Avec l'approche par caractères, 59.46% des pages contrefaites seraient déclarées légitimes à tort, contre seulement 2.70% avec l'approche par mots. En positionnant le seuil à 90%, il n'y aurait plus aucun faux-négatif avec l'approche par mots, contre encore 24.32% avec l'approche par caractères.

Enfin, nous avons cherché à savoir quelle méthode donnait le score le plus bas, pour chaque couple de page légitime-contrefaite. Il apparaît alors que l'approche par mots est déterminante pour 97.30% des sites (soit 36 couples de sites sur 37), contre seulement 2.70% des sites avec l'approche par caractères (soit 1 couple de sites sur 37).

Ces résultats de meilleure qualité, obtenus avec l'approche par mots, s'expliquent notamment par la possibilité d'identifier les zones de modifications entre pages légitimes et contrefaites, ainsi que par le poids accordé aux changements.

A contrario, avec l'approche par caractères, il faut qu'une partie conséquente de script soit ajoutée - par rapport à la taille de la page légitime - afin qu'elle soit détectée. De plus, les modifications de script sont moins détectables si les changements apportés tournent toujours autour des mêmes caractères.

5.3.5.2.3 Taux d'erreur de récupération des pages Défaut et Référence : Au-delà de l'efficacité de la méthode de comparaison utilisée sur la comparaison de contenu des pages, nous avons voulu évaluer son efficacité en terme de récupération des pages. En effet, si nous trouvons une méthode de comparaison de contenu efficace mais que notre technique de récupération des pages ne fonctionne que pour une faible proportion des URLs, l'analyse perd tout son intérêt.

Nous avons donc cherché à évaluer le taux d'échec de récupération des pages, auprès de chaque adresse IP utilisée : IPdef ou les 3 adresses IPref (cf. tableau 5.14).

Avec l'adresse IPdef, le taux d'échec moyen est de 2.19%, variant de 0 à 13.68% selon la localisation. A noter que sur les 11 localisations, 9 d'entre elles présentent un taux d'échec inférieur à 2%. Seules deux localisations situées en Asie (Chine et Émirats Arabes Unis) présentent des taux d'échec plus élevés, respectivement de 3.85% et 13.68%.

Avec les adresses IPref, le taux d'échec moyen est de l'ordre de 22%, oscillant entre 20 et 31%. Pour 10 des 11 localisations testées, les résultats sont de même calibre (taux d'échec entre 20 et 24%). Une localisation sort du lot, avec les adresses IPref issues de OpenDNS et GoogleDNS, le Vénézuela. Sur

Tableau 5.14 – Taux d'échec de récupération des pages webs légitimes en utilisant IPdef ou IPref, sur 11 localisations géographiques

URL utilisant l'adresse IP fournie par le DNS	**Taux d'échec** de récupération des pages webs (min ≤ moyenne ≤ max)	Écart-Type[1]
Défaut	0% ≤ **2.19%** ≤ 13.68%	3.99%
OpenDNS	20.37% ≤ **22.73%** ≤ 28.70%	2.09%
GoogleDNS	20.37% ≤ **22.56%** ≤ 31.48%	3.11%
DNSAdvantage	21.30% ≤ **22.39%** ≤ 24.07%	0.81%

[1] entre localisations

ce site géographique, nous avons deux pics d'échec à 28 et 31%.

Nous notons donc ici une nette différence de résultats sur les taux d'échec obtenus, selon l'adresse IP utilisée : IPdef ou IPref (cf. explications sur les causes probables en section 5.3.5.3).

5.3.5.3 Problèmes rencontrés

5.3.5.3.1 Causes probables des échecs de récupération de la page PageRef : Nous avons constaté un taux d'échec de récupération des pages 10 fois plus élevés pour PageRef (vs. PageDef). La principale différence, entre les 2 techniques de récupération de page web, réside dans la zone "domaine" de l'URL demandée. Dans un cas il s'agit d'un FQDN, dans l'autre d'une adresse IP. Au travers de l'analyse des pages récupérées ou des erreurs rencontrées à l'exécution du programme, nous supposons que les causes probables de ces échecs sont : soit un manque de configuration (ou une configuration inappropriée) du reverse DNS, soit la virtualisation de plusieurs domaines derrière une seule et même adresse IP. En effet dans ce dernier cas, puisque nous n'apportons aucune précision sur le FQDN recherché dans l'URL demandée, nous ne pouvons obtenir de réponse.

Les causes probables d'échec rencontrées sont illustrées au travers de 4 exemples, pour lesquels nous n'avons pu récupérer la page PageRef :

- Une interrogation DNS sur le FQDN `www.lsn.com.au` retourne l'adresse IP 20.134.224.83. Néanmoins, une interrogation Reverse DNS sur l'adresse IP 20.134.224.83 retourne un message d'erreur (*"unable to resolve, Country IP Address : UNITED STATES"*). Nous sommes confrontés ici à une absence de configuration du reverse DNS.
- Une requête DNS sur le FQDN `www.allianz.fr` retourne l'adresse IP 194.98.39.11. Une interrogation Reverse DNS sur l'adresse IP 194.98.39.11 retourne le FQDN `www.agf.fr`. A savoir que AGF et Allianz ont fusionné. Via un navigateur web, nous constatons d'ailleurs qu'une redirection automatique est effectuée depuis `www.agf.fr` vers `www.allianz.fr`. Nous sommes donc ici en présence d'un double hébergement de FQDN derrière une même adresse IP.
- Une interrogation DNS sur le FQDN `moncompte.numericable.fr` retourne l'adresse IP 85.68.0.39. Une requête DNS inverse sur l'adresse IP 85.68.0.39 retourne le FQDN `ip-39.net-85-68-0.static.numericable.fr`. Dans ce cas, la configuration Reverse DNS n'est pas appropriée pour la récupération de la page web. En effet, elle est représentative d'un découpage en sous-domaines.
- Une requête DNS sur le FQDN `www.cisco.com` retourne les informations suivantes : 88.221.8.170 pour l'adresse IP et 4 alias : `www.cisco.com.akadns.net`, `geoprod.cisco.com.akadns.net`, `www.cisco.com.edgekey.net` et `www.cisco.com.edgekey.net.globalredir.akadns.net`. Une requête DNS inverse sur l'adresse IP 88.221.8.170 retourne le FQDN `a88-221-8-170.deploy.akamaitechnologies.com`. Nous sommes ici en présence d'une virtualisation de plusieurs domaines derrière une même adresse IP, effectuée par une société tierce : Akamai Technologies.

A contrario, un exemple pour lequel notre récupération de PageRef fonctionne est le FQDN `ib.swedbank.lv` dont l'adresse IP retournée est 193.203.196.143. Derrière cette adresse IP, il semblerait qu'il n'y ait pas de découpage, ni de virtualisation du domaine. En effet, une résolution DNS inverse sur l'adresse IP 193.203. 196.143 retourne le FQDN `ib.swedbank.lv`.

Un des objectifs essentiels de notre seconde approche, développée en section 6.1, sera donc de pallier ce taux d'échec anormalement élevé pour la récupération de la page PageRef.

5.3.5.3.2 Elimination des comparaisons de pages faussées : L'approche par mots, telle que nous l'avons conçue dans le calcul de score, nous retourne directement un taux de similitude (et non un score par page). Il a donc fallu être particulièrement attentif aux analyses de ce résultat. En effet, par exemple, il nous a fallu éliminer tous les cas où nous obtenions un score de 100% alors que les deux pages comparées étaient vides ou inappropriées (c.-à-d. deux pages d'erreur).

Autre cas qui concerne les deux approches (par caractères et par mots) : certaines pages récupérées ne sont pas vides mais erronées. En effet, il arrive qu'une des deux pages récupérées (voire les deux) soit une page d'erreur (p.ex. indisponibilité temporaire du site, déplacement de page, etc.). Le résultat de la comparaison étant faussé, le score obtenu doit également être éliminé des statistiques.

Dans cette première proposition, l'essentiel des éliminations de résultats erronés a été faite manuellement. Dans la seconde proposition portant sur davantage d'URLs, ce traitement a été automatisé.

5.3.5.3.3 Identification de l'adresse IPdef : De par l'implémentation réalisée, à partir de fonctions pré-existantes Java, il s'est avéré impossible d'identifier l'adresse IPdef utilisée pour la récupération de la page par défaut. En effet, rien n'indique que l'adresse IPdef utilisée est la première adresse IP retournée par DNSdef. Les fonctions utilisées pour la requête HTTP (p.ex. *url.openConnection()*) ne nous permettent pas d'identifier cette adresse. Ce verrou a été levé dans la seconde proposition.

5.3.5.3.4 Phase d'établissement de la connexion sécurisée : Une des difficultés rencontrées pour la récupération des pages webs a été la phase d'établissement SSL/TLS, préambule nécessaire à la récupération des pages webs de login. Divers essais ont été nécessaires avant d'aboutir à la solution exposée en section 5.3.5.1. Cette solution semble satisfaisante pour une très large majorité des URLs. Néanmoins, nous avons constaté quelques erreurs de certificats résiduelles et aléatoires. A ce jour, nous n'avons pas eu le temps d'investiguer plus avant ces erreurs ponctuelles.

5.3.5.4 Synthèse sur l'analyse du code source des pages webs

Les résultats de comparaison de pages webs obtenus dans cette première proposition sont assez convergents avec les résultats de l'étude préalable. Ils semblent toutefois indiquer que l'approche par mots doit être privilégiée, principalement de par les résultats obtenus sur les pages légitimes-contrefaites. Néanmoins, les résultats de la seconde proposition - portant sur davantage d'URLs - doivent confirmer ces résultats.

Un des freins essentiels exposés dans cette première proposition concerne le taux d'échec - trop élevé - relevé pour la récupération des pages de référence. Les développements menés dans la seconde approche viseront donc à tenter de remédier à ce problème, ainsi qu'à corroborer nos hypothèses sur les causes probables d'échec.

A l'issue de cette première proposition, deux autres verrous demeurent :

- la difficulté d'identification et d'élimination des comparaisons de pages erronées (c.-à-d. les pages d'erreur).
- l'identification de l'adresse IPdef utilisée pour le chargement de la page par défaut.

Enfin, même si l'approche par mots semble la plus appropriée pour déterminer la légitimité d'un site, les seuils minimum relevés sur les taux de similitude ne permettent pas - à ce stade - de dégager un seuil de décision. En effet, concernant les comparaisons de pages légitimes (cf. tableau 5.12), le seuil minimal moyen relevé est de 81.61% (tous DNSref et localisations confondus). Concernant les pages légitimes-contrefaites, le seuil maximal relevé est de 92%. La fenêtre de recouvrement est donc d'environ 11%. Certes, d'après les résultats des pages légitimes-contrefaites, ce chiffre peut être tempéré par deux éléments (cf. tableau 5.13) :

- la quantité de taux de similitude dépassant la barre des 80% n'est que de 2.70%.
- la fourchette haute de l'intervalle de confiance (à 95%) se situe à 65.59%.

Néanmoins, même si de futurs résultats confirment ces tendances, il semble peu probable que l'approche par mots telle qu'envisagée ici (c.-à-d. appliquée à l'ensemble du code source de la page) puisse être un critère suffisant dans la détermination de la légitimité de la page web visitée.

Notre seconde proposition visera donc également à explorer d'autres pistes pouvant aider à la décision de légitimité (p.ex. application de la méthode de score à des sous-parties de la page web, analyse focalisée sur les balises, etc.).

5.4 Synthèse du chapitre

Ce chapitre a présenté une première proposition visant à détecter les attaques de pharming réalisées côté client, focalisée sur les pages de login. Cette proposition est constituée de deux étapes : une vérification de l'adresse IP du domaine visité et une analyse/comparaison de la page web visitée, vs. des éléments de référence obtenus en utilisant un serveur DNS alternatif.

Les tests effectués sur la première solution proposée ont démontré que la vérification de l'adresse IP est une étape valable - mais pas toujours suffisante - dans la décision de légitimité d'un site de login. Ils ont également indiqué que les résultats issus de différents serveurs DNS de référence sont globalement convergents, ce qui laisse présager d'une certaine liberté dans le choix du serveur alternatif.

Ces résultats ont par ailleurs démontré que les méthodes actuellement utilisées pour la comparaison des pages webs - bien qu'intéressantes - se doivent d'être améliorées pour aboutir à un seuil de décision. En effet, à ce stade, aucune des deux approches ne se distingue avec certitude.

Par ailleurs, à l'issue de cette première proposition des doutes subsistent. Tout d'abord, la base d'URLs testées se doit d'être augmentée pour conforter les résultats obtenus et ce, pour les deux étapes de notre processus de décision.

Ensuite, certains éléments sont insuffisamment précis et/ou nécessitent une meilleure optimisation de traitement. On peut par exemple citer le problème d'identification de l'adresse IPdef ou le problème de l'élimination des pages d'erreur récupérées.

Enfin, le verrou majeur subsistant concerne les taux d'échec de récupération des pages webs de référence trop importants.

L'ensemble de ces éléments nous amène donc à envisager un nouveau scénario, qui fait l'objet de la seconde proposition développée dans le chapitre suivant.