Étude de l'influence de la biodiversité logicielle d'un système sur sa probabilité d'infection

Philippe Troclet @
Alexandre Mao @
Paul Berthier paul.berthier@polymtl.ca
Thomas Luinaud thomas.luinaud@polymtl.ca

***Abstract*—**

I. Introduction

En janvier 2015, à la suite des attentats à Charlie Hebdo, une vague d'attaques a ciblé les sites des municipalités françaises [1]. Sur les 36658 communes que comptait la France en 2015 d'après l'INSEE [2], plusieurs centaines ont vu leur site web piraté. Doit-on en déduire que les sites web des communes françaises sont mal protégés ? Si oui, quelle en est la cause ? Ou au contraire, la vague d'attaques aurait-elle pu être encore plus vaste ? Si oui, quel élément l'a-t-elle freinée ?
Pour répondre à ces questions, le magazine *La gazette des communes* a effectué en février 2015 un audit des sites webs municipaux français [3]. On y découvre qu'environ 15000 communes disposent de leur propre site, mais surtout que ceux-ci présentent une très grande biodiversité logicielle, tant au niveau des logiciels utilisés que de leurs versions. Nous allons donc par la suite évaluer l'impact de ce facteur environnemental sur les critères de performance d'un attaquant et d'un défenseur (les communes). Nous considérerons que le système de défense est inexistant. Le défenseur pourra seulement influer sur l'environnement moyennant un certain coût de migration. L'attaquant, lui, pourra adapter son modèle d'attaque à l'environnement dans une certaine mesure : celui-ci n'est pas toujours connu (les sites web ont la possibilité de masquer leur configuration), et cela a un certain coût (recherche ou achat de nouveaux *exploits*).
Dans un premier temps, nous développerons un modèle mathématique. Nous introduirons une entropie logicielle caractérisant la "biodiversité" de notre système, puis nous introduirons des fonctions de changement d'état, qui impliquent une certaine énergie et donc un certain coût pour le défenseur. Nous introduirons également les équations du côté de l'attaquant : l'efficacité de son attaque dépendra, tout comme en thermodynamique pour l'efficacité d'une machine thermique, de l'entropie du système. Dans un second temps, nous appliquerons notre modèle aux données obtenus à partir de l'article de *La gazette*, qui on été publiées en Open Data sur le site gouvernemental *data.gouv.fr* [4].

II. Revue de littérature

L'études des vulnérabilités au sein des logicielles n'est pas chose nouvelle. Des études quantitatives cherchant à modéliser le nombre de failles de sécurités découvertes au cours du temps ont déjà été menées [5]. Et ce, avec des résultats plus que satisfaisants. Il s'agissait dans cet article de parvenir à modéliser le taux de découvertes de vulnérabilités afin de pouvoir l'anticiper et prévoir à l'avance de consacrer un certain nombre de personnes à la résolution des failles qui serraient découvertes. Il s'agissait donc plutôt de mieux gérer ses effectifs. Une autre étude, plus centrée sur l'anticipation du nombre de failles dans un programme, a vu le jour [6]. Cet article introduit également des modèles permettant de caractériser le nombre de vulnérabilités que l'on trouvera si l'on consacre un certain "effort" à leur recherche. Par ailleurs, l'existence de malware utilisant plusieurs vulnérabilités pour se propager est déjà connu. On se souviendra ainsi du vers *Nimda*, qui utilisait 5 vulnérabilités différentes. Du point de vue de la défense, l'idée d'utiliser différents logicielles, afin de varier les vulnérabilités et donc de limiter les possibilités de l'attaquant n'est pas nouvelle, et plusieurs travaux ont tenté de caractériser le gain induit par une telle approche. En particulier, [7] applique la notion d'entropie à des graphes bipartis afin de mesurer la diversité d'un système.

III. Méthodologie

Afin de réaliser notre analyse, nous avons procédé en deux étapes. Tout d'abord nous avons défini un modèle mathématique permettant d'estimer le coût d'une attaque et de la défense. Puis nous avons appliqué notre modèle en analysant des jeux de données. Dans cette section, nous expliquerons tout d'abord notre modèle mathématiques puis nous expliquerons comment nous avons appliqué ce modèle à nos données.

A. Modèle mathématique

Afin de définir notre modèle mathématique, nous avons basé notre approche sur un phénomène physique homologue : la thermodynamique. Nous allons donc définir une entropie logicielle qui sera liée à la "biodiversité" du système. Redéfinissons tout d'abord le premier principe de la thermodynamique : Il y a conservation de l'énergie.

$$\Delta E = \Delta U = W + Q$$

ΔE est la variation d'énergie du système, ΔU est la variation d'énergie interne du système, W est le travail reçu par le système et Q est la chaleur reçue par le système.

La seconde loi de la thermodynamique nous donne :

$$S_{creation} = \Delta S_{syst} + \Delta S_{ext} \geq 0$$

Cela se caractérise dans notre cas par un principe simple : tout changement de configuration dans notre système, quel qu'il soit, crée de l'entropie et donc augmente, au moins temporairement, la difficulté de l'attaque. Dans le cas ou l'entropie de notre système diminuerai, il y a tout de même une augmentation de l'entropie du côté de l'attaquant. Cela s'explique par le fait que l'état du nouveau système lui est inconnu. Il doit à nouveau refaire toute son analyse avant d'effectuer une nouvelle attaque. On notera également que sans action spécifique du défenseur, en laissant les administrateurs des sous-systèmes faire les mises à jour indépendamment, l'entropie du système global va augmenter de manière naturelle au cours du temps.

Le second principe peut également s'écrire :

$$\frac{Q}{T} \leq \Delta S_{syst}$$

À quoi correspondent toutes ces grandeurs physiques dans notre système ? On va associer Q à l'effort fourni pour effectuer des mises à jours (Si Q est nul, il n'y a pas d'entropie créée), et W sera le travail fourni par l'attaquant pour effectuer son attaque. Du côté du défenseur, Q va en réalité être négatif. En effet, on va associer l'énergie du système à la quantité d'attaques subies. On veut donc en permanence le "refroidir". De même, la température T va être associée à la vulnérabilité des versions des logiciels du système. Plus les versions sont anciennes, plus elles seront vulnérables et donc la température va augmenter en fonction du temps.

B. Application du modèle

Dans cette section, nous expliquons comment nous avons appliqué notre modèle mathématiques. Nous avons dans un premier temps récupérer les données des différentes versions de serveur Web utilisés par les communes françaises ainsi que des vulnérabilités associées. Finalement nous avons recoupé les différentes information et effectué le calcul mathématiques.

1) Récupération des données: Pour faire une études comparative, nous avons récupéré un jeu de donnée sur les serveurs web utilisés par les communes françaises datant de mars 2015. Par la suite, nous avons généré un jeu de donnée à partir du même script afin de connaître l'état actuel des systèmes. Finalement nous avons récupéré les différentes failles de sécurité connu pour les différentes version de ces logiciels.

Une fois les données trouvée, nous avons réalisé un recoupement des données. Pour cela, nous avons considéré que les versions de logiciel donnés par les serveurs sont les versions réellement utilisés et également que les serveur n'envoyant pas d'informations sont sécurisé de base.

2) Application du modèle mathématique:

IV. Conclusion

References

[1] "Des centaines de sites, dont des collectivités locales, victimes de cyber-attaques." [Online]. Available: http://www.courrierdesmaires.fr/44724/des-centaines-de-sites-dont-des-collectivites-locales-victimes-de-cyberattaques/

[2] INSEE. (2015) Liste des communes françaises. [Online]. Available: http://www.insee.fr/fr/methodes/nomenclatures/cog/telechargement.asp

[3] "Plusieurs milliers de sites Internet de communes mal sécurisés." [Online]. Available: http://www.lagazettedescommunes.com/337105/plusieurs-milliers-de-site-de-collectivites-mal-securises/

[4] "Sécurité des sites informatiques des communes françaises - Data.gouv.fr." [Online]. Available: https://www.data.gouv.fr/fr/datasets/securite-des-sites-informatiques-des-communes-francaises/

[5] I. R. Jinyoo Kim, Yashwant K. Malaiya, "Vulnerability discovery in multi-version software systems," in *10th IEEE High Assurance Systems Engineering Symposium*.

[6] O. H. A. Sung-Whan Woo and Y. K. Malaiya, "Assessing vulnerabilities in apache and iis http servers," in *Assessing Vulnerabilities in Apache and IIS HTTP Servers*.

[7] M. E. L. Saran Neti, Anil Somayaji, "Software diversity: Security, entropy and game theory," in *Software diversity: Security, Entropy and Game Theory*.