Mesurer et prévenir l'évolution de la menace dans un cloud d'infrastructure

Clément Elbaz, Louis Rilling, Christine Morin

Compas'2018 : Parallélisme/ Architecture / Système
ENSEEIHT - Toulouse, France, du 03 au 06 juillet 2018

Mesurer et prévenir l'évolution de la menace dans un cloud d'infrastructure

Clément Elbaz, Louis Rilling (DGA), Christine Morin

Univ Rennes, Inria, CNRS, IRISA,
Campus de Beaulieu, 263 Avenue Général Leclerc
35042 Rennes - France
clement.elbaz@inria.fr, louis.rilling@irisa.fr, christine.morin@inria.fr

---

Résumé
Le cloud computing a permis le passage d'un paradigme d'obligation de moyen à celui d'obligation de résultat, mais cette transition montre ses limites dans le domaine de la sécurité. En effet, la sécurité effective d'un système informatique est notoirement difficile à mesurer. L'évolution constante de la menace, par exemple via la publication de nouvelles vulnérabilités, peut profondément changer le risque auquel est soumis un système d'information, sans que la moindre modification ait été apportée à ce système. Dans ces travaux, nous souhaitons évaluer la pertinence d'un modèle économique où un fournisseur de cloud d'infrastructure garantit, dans une fenêtre de temps contractualisée, le déploiement de contre-mesures pour toutes les nouvelles vulnérabilités publiques pouvant impacter ses clients, assorti d'une pénalité financière en cas de manquement. Nous montrons ainsi qu'il est possible pour un fournisseur de cloud de quantifier la probabilité d'avoir à payer une pénalité, et ainsi garantir un bénéfice économique sur le long terme.

Mots-clés : Cloud, Sécurité, Contrat de Service (SLA), CVE, IDS

---

1. Introduction

Un aspect essentiel du cloud computing est la transition d'une obligation de moyen vers une obligation de résultat. Le client ne se soucie plus de la façon dont le service est mis en œuvre (machines physiques, datacentres, etc.), mais seulement de la qualité effective du service.
Cette qualité de service est estimée à travers un certain nombre de mesures observables par le client et le fournisseur et un contrat de service (SLA) est mis en place à partir de ces mesures.
Le cloud a démocratisé les contrats de service liés à la disponibilité et la performance des systèmes d'information mais pas à leur sécurité, qui reste difficile à mesurer. En plus des facteurs intrinsèques au système, comme son architecture et son implémentation, la sécurité dépend également de facteurs extrinsèques au système, comme la sécurité des briques logicielles tierces utilisées, ou encore les menaces présentes dans l'environnement.
De nos jours, la plupart des systèmes d'information s'appuient sur des briques logicielles tierces. Cette évolution, bénéfique sur bien des aspects, entraîne toutefois un effet de bord néfaste : la menace pesant sur un système peut profondément évoluer via la publication de vulnérabilités dans les briques logicielles tierces dont il dépend. Le système d'information est alors exposé jusqu'au déploiement d'une contre-mesure, la plus classique étant un correctif applicatif. Une autre contre-mesure possible est une règle par signature dans un système de détection d'intrusion (IDS) afin de détecter ou bloquer les tentatives d'un attaquant de faire usage de la vulnérabilité.

Aucune	Debian (exclusif)	Talos (exclusif)	Debian et Talos	Total
28338	3028	2691	181	34238

TABLE 1 – Distribution des contre-mesures dans l'échantillon de vulnérabilités.

Notre contribution est double. En premier lieu, nous présentons une étude du cycle de vie des vulnérabilités, à notre connaissance la première à intégrer le délai de publication de règles de signature IDS en plus de la distribution d'un correctif applicatif. En second lieu, nous utilisons cette étude pour proposer un modèle économique dans le cloud fondé sur la garantie de déploiement de contre-mesure dans un délai contractualisé par le fournisseur, pour toute nouvelle vulnérabilité publique menaçant un client. Nous montrons que le délai de disponibilité d'une contre-mesure peut être quantifié et permettre de garantir au fournisseur de cloud un bénéfice sur le long terme.
Le paragraphe 2 présente notre étude du cycle de vie des vulnérabilités et de leurs contre-mesures. Le paragraphe 3 établit notre modèle économique, propose une analyse probabiliste de celui-ci, et le calcul de son espérance de gain. Nous évaluons notre approche en paragraphe 4. En paragraphe 5, nous présentons les travaux apparentés à notre contribution. Enfin, nous concluons en paragraphe 6.

2. Cycle de vie des vulnérabilités et distribution de leurs contre-mesures dans le cloud

Afin de proposer un modèle économique réaliste, il est nécessaire d'étudier le cycle de vie des vulnérabilités et de leurs contre-mesures. Notre objectif est d'évaluer la fréquence des vulnérabilités et le délai de réponse par les distributeurs de contre-mesures. En particulier, nous comparons un distributeur de correctifs typique (un dépôt de sécurité d'une distribution Linux) avec un distributeur de règles de signature IDS typique. Les vulnérabilités traitées par ces distributeurs sont généralement référencées via une base de vulnérabilités publique normalisée, ce qui permet de corréler ces différentes sources.

2.1. Sources de données

Notre étude croise quatre sources de données : la *National Vulnerability Database* (NVD) [6] fournie par le *National Institute of Standards and Technology* (NIST), le compte Twitter @CVENew [2] tenu par la Mitre Corporation [9], le dépôt de paquets debian-security de la distribution Linux Debian [3], et les règles de signature IDS fournies par Talos [8]. Nous en avons tiré un échantillon de 34000 vulnérabilités publiées entre juin 2014 et octobre 2017.
Pour chaque vulnérabilité de notre échantillon, nous avons cherché à isoler son score de sévérité calculé avec le Common *Vulnerability Scoring System* (CVSS) [1] ainsi que trois horodatages. Nous avons uniquement utilisé le score de sévérité de base, mais un fournisseur cloud pourrait intégrer les axes temporels et environnementaux de CVSS pour ajuster plus finement les scores CVSS à sa situation spécifique et celle de ses clients.
Le premier est celui de la publication initiale de la vulnérabilité. Le second est celui du premier correctif applicatif publié dans le dépôt debian-security pour corriger la vulnérabilité, et le troisième est celui de la première règle IDS publiée par Talos en rapport avec cette vulnérabilité. Ces deux derniers horodatages sont optionnels, puisque qu'il existe des vulnérabilités qui ne concernent ni Debian ni Talos.
Comme [17] avant nous, nous avons noté que l'horodatage de publication fourni par NVD n'est pas fiable. Nous l'avons donc corrélé avec les horodatages du compte Twitter @CVENew (qui publie un tweet par vulnérabilité publique) afin d'extrapoler un horodatage de confiance pour la publication initiale de la vulnérabilité.

2.2. Résultats et analyse

A partir des sources précédentes, nous présentons les résultats de notre étude dans les tables 1, 2 et 3. Nous en retirons plusieurs points d'attention.

Score CVSS	0	1	2	3	4	5	6	7	8	9
Debian (%)	0,19	12,68	10,47	4,99	11,18	10,68	12,56	14,29	5,92	3,99
Talos (%)	0,38	4,35	3,56	1,03	3,47	3,32	6,21	11,32	11,84	33,36

TABLE 2 – Taux de réponse par Debian ou Talos en fonction du score d'une vulnérabilité. Le taux de réponse indépendant du score est de 9,37% pour Debian et 8,39% pour Talos.

Délai depuis la publication	0h	24h	48h	7j	30j	60j
Vulnérabilités avec contre-mesures	45,22%	53,77%	56,03%	65,78%	82,64%	89,45%
Vulnérabilités en général	7,79%	9,26%	9,65%	11,33%	14,24%	15,41%

TABLE 3 – Évolution dans le temps des statistiques de publication d'une contre-mesure pour une vulnérabilité, sur le total des vulnérabilités avec contre-mesure (hypothèse exhaustive) ou sur le total des vulnérabilités en général.

Le premier est qu'une part importante des vulnérabilités ne sont prises en compte ni par Debian ni par Talos. Dans notre échantillon, 82,76% des vulnérabilités ne reçoivent aucune contre-mesure. Intuitivement, cela a du sens : il existe de nombreuses vulnérabilités qui ne concernent ni une distribution Linux ni un IDS. Toutefois, une question ouverte de notre étude est de savoir si Debian et Talos sont "exhaustifs" dans leur choix de réponse aux vulnérabilités.
Le second point d'attention est la très faible intersection entre les vulnérabilités prises en compte par Debian et par Talos. Dans notre échantillon, Debian propose un correctif pour 9,37% des vulnérabilités, et Talos une règle IDS pour 8,39% d'entre elles. Mais seulement 0,53% des vulnérabilités de l'échantillon ont reçu une réponse commune de Debian et Talos. Nous n'avons qu'une explication partielle à cela. Nous avons noté que le taux de réponse à une vulnérabilité par Talos est nettement plus corrélée au score CVSS de la vulnérabilité que le taux de réponse par Debian (table 2). En particulier, Talos se concentre largement sur les vulnérabilités critiques (score CVSS supérieur à 9,0). Notre conclusion est que Debian et Talos ont une approche différente dans leur processus de sélection des vulnérabilités auxquelles ils répondent.
Concernant le délai entre la publication de la vulnérabilité et la première contre-mesure (cf table 3), nous notons que 45,22% des vulnérabilités avec une contre-mesure l'ont reçue à la publication de la vulnérabilité ou avant. Ce chiffre monte à 65,78% 7 jours après la publication, et 89,45% après deux mois. A noter que le score d'une vulnérabilité n'influence pas significativement le délai de publication. Rappelons que seulement 17,23% des vulnérabilités reçoivent une contre-mesure de Debian ou Talos. La question de savoir si Debian et Talos sont exhaustifs dans leur processus de sélection de vulnérabilité est donc structurante. Sans avoir de réponse formelle à cette question, plusieurs expérimentations que nous avons mené vont dans le sens de la viabilité de cette hypothèse. En particulier nous avons réalisé une étude manuelle des vulnérabilités critiques de trois piles logicielles cloud typiques : LAMP [4], MEAN [5], et Java Spring [7]. Toutes ces vulnérabilités sont prises en compte par Debian ou Talos.
Dans ce paragraphe nous avons proposé une nouvelle étude du cycle de vie des vulnérabilités et leurs contre-mesures, nous permettant d'élaborer un modèle économique fondé sur des données réelles.

3. Un modèle économique probabiliste de protection dynamique pour un fournisseur de cloud

A partir des résultats de notre étude, nous pouvons formuler et modéliser un modèle économique pour le fournisseur de cloud.

3.1. Description du modèle économique

Le modèle économique que nous proposons est fondé sur un système d'abonnement.

À intervalle régulier (par exemple mensuellement), un client paye au fournisseur de cloud un abonnement à prix fixe. En contrepartie, le fournisseur s'engage à fournir dans un délai contractuel une contre-mesure pour toute nouvelle vulnérabilité publiée qui menacerait le client. S'il ne respecte pas le délai, alors il paye au client une pénalité fixée par le contrat de service.
Il est nécessaire que le fournisseur et le client s'accordent sur ce que constitue une menace pour le client. Une possibilité pour cela est que le client fournisse une liste de tous les logiciels qu'il utilise, sous la forme d'un référentiel normalisé. Une autre solution est pour le fournisseur de détecter automatiquement les solutions logicielles installées par le client, mais il s'agit d'un problème non-trivial [16] et soulevant des questions de confidentialité entre le fournisseur et le client.
Dans notre modèle, le montant total de la pénalité n'est pas borné. Le fournisseur de cloud paye une pénalité pour chaque manquement au contrat, indépendamment de ce qu'a payé le client jusqu'ici.

3.2. Modélisation formelle du modèle économique

Nous décrivons formellement le modèle économique présenté précédemment.
Nous appelons *période* la granularité de l'abonnement proposé par le fournisseur de cloud. Par exemple, si l'offre d'abonnement est mensuelle, alors la période est d'un mois.
Soit T l'ensemble des nouvelles vulnérabilités publiées pendant la période en cours. Soit A le montant de la somme payée par le client au fournisseur pendant la période en cours. Soit Pen le montant agréé de la pénalité à rembourser par le fournisseur pour chaque vulnérabilité en dépassement de délai.
Nous pouvons donc modéliser chaque vulnérabilité de l'ensemble T comme un évènement procurant un gain ou une perte au fournisseur. Elle lui rapporte systématiquement une fraction de l'abonnement ($\frac{A}{|T|}$), mais peut en plus entraîner le paiement de la pénalité s'il ne remplit pas ses obligations ($-Pen$).
Nous définissons l'événement M comme le fait que la vulnérabilité étudiée constitue une menace pour le client, tel que défini contractuellement avec le fournisseur. Nous définissons l'événement D comme le fait que le fournisseur respecte son contrat de service et fournisse une contre-mesure dans les délais.
La figure 1 illustre cette modélisation. L'espérance de gain par vulnérabilité par client est donnée par l'équation 1. Précisons que celle-ci est un chiffre d'affaire plutôt qu'un bénéfice : le coût de la prestation technique du fournisseur n'est pas inclus.

$$E_{vuln} = \left(\frac{A}{|T|}\right)(\mathbb{P}(!M) + \mathbb{P}(D|M)) + \left(\frac{A}{|T|} - Pen\right)(\mathbb{P}(!D|M)) \quad (1)$$

L'espérance de gain par période par client peut être calculée en multipliant le gain par vulnérabilité par le nombre total de vulnérabilités publiées sur la période. Cela est illustré par l'équation 2.

$$E_{periode} = |T| \times \left(\left(\frac{A}{|T|}\right)(\mathbb{P}(!M) + \mathbb{P}(D|M)) + \left(\frac{A}{|T|} - Pen\right)(\mathbb{P}(!D|M))\right) \quad (2)$$

4. Évaluation

A partir de la modélisation du paragraphe précédent, nous pouvons calculer l'espérance de gain du fournisseur. Celle-ci dépend de plusieurs paramètres sur lesquels il dispose d'un degré de contrôle variable.
La taille de la période, A et Pen sont directement sous le contrôle du fournisseur, qui est libre de proposer à ses clients le contrat de service de son choix. Toutefois, le fournisseur et le client coexistent au sein d'un marché économique, qui borne ces valeurs dans des proportions acceptables par le fournisseur et par le client. Nos travaux ne nécessitent pas d'hypothèse sur ces bornes. Dans les scénarios d'évaluation qui suivent, nous fixons arbitrairement les modalités de l'abonnement à 100 € par mois ($A = 100€$ et $periode = 1mois$) afin de mesurer l'impact des autres variables sur l'espérance de gain.

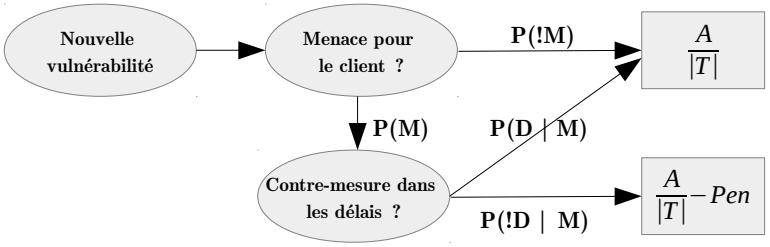

FIGURE 1 – Modélisation graphique de l'espérance de gain.

L'ensemble des vulnérabilités T publiées pendant une période n'est que partiellement contrôlable par le fournisseur. Bien qu'il n'ait pas le contrôle sur la nature et le nombre exact de vulnérabilités publiées pendant une période, il peut décider de restreindre son offre contractuelle à certains types de vulnérabilités (par exemple, les vulnérabilités critiques ou les vulnérabilités affectant des logiciels précis). Ces restrictions modulent la taille de T. Là encore, le marché dans lequel coexistent le fournisseur et le client impose des restrictions raisonnables sur T en fonction de A et Pen. Nous notons que dans notre échantillon, environ 150 vulnérabilités par mois recevaient une contre-mesure, parmi elles 40 critiques en moyenne. Nous utilisons ces chiffres dans les scénarios d'évaluation qui suivent.

D est le facteur sur lequel le fournisseur a le moins de contrôle mais il peut toutefois être évalué. Nous considérons que la capacité du fournisseur à fournir une contre-mesure dépend du distributeur de la contre-mesure, et reprenons donc les chiffres tirés de notre étude en paragraphe 2.

M, la probabilité qu'une vulnérabilité de T constitue une menace pour le client, est une variable composite dépendant de multiples facteurs. Pour l'étudier, il peut être judicieux de généraliser le calcul de l'espérance de gain à tous les clients d'un fournisseur plutôt qu'un seul. Dans ce contexte, un point d'attention est que nous ne formulons pas d'hypothèse sur l'écart-type de M : de nombreuses vulnérabilités peuvent menacer quelques clients, ou quelques vulnérabilités peuvent menacer de nombreux clients. M dépend donc de la nature de T et des restrictions placées par le fournisseur, ainsi que de la nature et l'homogénéité des logiciels utilisés par les clients du fournisseur. Des travaux additionnels seront nécessaires afin de mieux étudier cette métrique dans le futur. Dans les scénarios d'évaluation de cette étude, nous faisons l'hypothèse qu'un client est menacé par une vulnérabilité 1 à 5% du temps.

Nous proposons deux familles de scénarios, décrits dans la figure 2. Tous sont fondés sur l'hypothèse « exhaustive » décrite en section 2, c'est-à-dire que les fournisseurs de contre-mesures sont considérés comme n'oubliant aucune vulnérabilité dans leur processus de réponse. Les trois premiers scénarios sont fondés sur un contrat de service garantissant un délai de 30 jours pour le déploiement de contre-mesure pour toute vulnérabilité affectant un client, indépendamment de son score. Les trois suivants sont similaires, mais avec un délai de 7 jours et portent uniquement sur les vulnérabilités critiques.

De ces résultats, nous notons trois points d'attention. Le premier est le fort impact de M sur les espérances de gain projetées. Cette métrique étant très spécifique aux usages des clients, il apparait indispensable pour un fournisseur de cloud de l'étudier en pratique pour ses propres clients avant d'élaborer une offre commerciale fondée sur ces travaux. Le second est que moduler T impacte plus l'espérance de gain que moduler le délai contractuel. Il est notamment plus rentable pour un fournisseur de garantir un délai de 7 jours pour les vulnérabilités critiques, qu'un délai de 30 jours pour toutes les vulnérabilités en général. Enfin, sous certaines conditions (en particulier quand M est faible), il est possible pour le fournisseur de cloud de fournir des garanties substantielles (sous la forme de pénalités contractuelles très élevées)

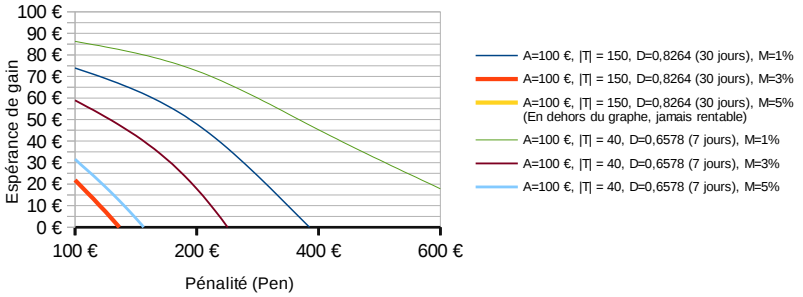

FIGURE 2 – Espérance de gain pour différents choix de pénalité *Pen*. Plusieurs scénarios sont proposés, définissant des valeurs pour A, $|T|$, D et M. Les valeurs de D sont sélectionnées à partir de la table 3.

tout en restant gagnant sur le long terme. Par exemple si M est évalué à 1% ou moins, le fournisseur de cloud peut proposer une pénalité par vulnérabilité plusieurs fois supérieure au montant de l'abonnement mensuel et rester rentable. D'une façon générale, il est possible pour le fournisseur de rembourser un mois d'abonnement par manquement contractuel ($A = Pen$) dans cinq scénarios sur six. Cette évaluation permet de montrer la viabilité dans de nombreuses situations du modèle économique que nous proposons.

5. Travaux apparentés

De multiples travaux [17, 14, 19, 12] ont été réalisés afin d'étudier le cycle de vie des vulnérabilités publiées par le réseau CVE. Toutefois, nos travaux proposent un nouveau regard sur le sujet à travers trois facteurs différenciants. Le premier est la fraîcheur des données que nous avons analysées : notre échantillon recouvre une période allant de juin 2014 à octobre 2017. A titre de comparaison, [17] portait sur la période 1996 - 2006, [14] sur 1999 à 2007, [19] sur 1988 à 2011 et [12] sur 2009 à 2011. Le second est notre choix de généraliser l'étude à la notion de contre-mesures plutôt qu'aux seuls correctifs applicatifs. A notre connaissance, notre étude est la première à comparer le délai de publication de règles de signature pour IDS avec celui de la publication d'un correctif applicatif. Le troisième est notre choix de considérer l'accès pratique à ces contre-mesures en plus de leur publication. En effet nous considérons que la date de publication d'une contre-mesure par le vendeur logiciel est moins pertinente que la date de mise à disposition de la contre-mesure dans un dépôt facilement accessible (eg Debian).
Les contrats de service sur la sécurité dans le cloud ont fait l'objet de plusieurs études [13, 20, 18, 15, 21, 11]. Toutefois, ces travaux ne se concentrent pas sur la notion d'évolution de menace ou la publication de nouvelles vulnérabilités et de leurs contre-mesures. La modélisation probabiliste de la sécurité dans le cloud a été étudiée par plusieurs auteurs [10, 22]. Toutefois, ces travaux modélisent les interactions entre un attaquant et un défenseur plutôt qu'un modèle économique entre un fournisseur de cloud et un client.

6. Conclusion et futurs travaux

La difficulté inhérente à mesurer la sécurité dans le cloud crée de mauvaises incitations économiques pour les fournisseurs et leurs clients. Nous avons présenté ici deux contributions. En premier lieu une étude inédite sur le cycle de vie des vulnérabilités, ainsi qu'une analyse statistique des données de cette étude. Nous avons ensuite proposé un modèle économique où un fournisseur de cloud assure à ses clients des garanties contractuelles relatives à l'évolution de la menace et la publication de nouvelles vulnérabilités. Nous avons montré que sous certaines conditions, ce modèle économique est viable. Dans le futur nous envisageons de travailler à mieux comprendre les relations entre les usages des clients de cloud et les vulnérabilités qui les menacent afin d'affiner notre modèle.

Bibliographie

1. Common Vulnerability Scoring System. – https://www.first.org/cvss/.
2. CVENew sur Twitter. – https://twitter.com/CVEnew.
3. Debian – Security Information. – https://www.debian.org/security/.
4. LAMP (software bundle). – https://en.wikipedia.org/wiki/LAMP_(software_bundle).
5. MEAN (software bundle). – https://en.wikipedia.org/wiki/MEAN_(software_bundle).
6. National Vulnerability Database. – https://nvd.nist.gov/.
7. Spring framework. – https://spring.io/.
8. Talos - Author of the Official Snort Rule Sets. – https://snort.org/talos.
9. The MITRE Corporation. – https://www.mitre.org/.
10. Barth (A.), Rubinstein (B. I. P.), Sundararajan (M.), Mitchell (J. C.), Song (D. X.) et Bartlett (P. L.). – A Learning-Based Approach to Reactive Security. *CoRR*, vol. abs/0912.1155, 2009.
11. Bernsmed (K.), Jaatun (M. G.), Meland (P. H.) et Undheim (A.). – Security SLAs for Federated Cloud Services. – In *2011 Sixth International Conference on Availability, Reliability and Security*, pp. 202–209, Aug 2011.
12. Bilge (L.) et Dumitraş (T.). – Before We Knew It : An Empirical Study of Zero-day Attacks in the Real World. – In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pp. 833–844, New York, NY, USA, 2012. ACM.
13. Cheng (P.), Wang (L.), Jajodia (S.) et Singhal (A.). – Aggregating CVSS Base Scores for Semantics-Rich Network Security Metrics. – In *2012 IEEE 31st Symposium on Reliable Distributed Systems*, pp. 31–40, Oct 2012.
14. Clark (S.), Frei (S.), Blaze (M.) et Smith (J.). – Familiarity Breeds Contempt : The Honeymoon Effect and the Role of Legacy Code in Zero-day Vulnerabilities. – In *Proceedings of the 26th Annual Computer Security Applications Conference*, pp. 251–260, New York, NY, USA, 2010. ACM.
15. de Chaves (S. A.), Westphall (C. B.) et Lamin (F. R.). – SLA Perspective in Security Management for Cloud Computing. – In *2010 Sixth International Conference on Networking and Services*, pp. 212–217, March 2010.
16. Dulaunoy (A.). – The Myth of Software and Hardware Vulnerability Management. – https://www.foo.be/2016/05/The_Myth_of_Vulnerability_Management/, 2016.
17. Frei (S.), May (M.), Fiedler (U.) et Plattner (B.). – Large-scale Vulnerability Analysis. – In *Proceedings of the 2006 SIGCOMM Workshop on Large-scale Attack Defense*, pp. 131–138, New York, NY, USA, 2006. ACM.
18. Petcu (D.) et Crăciun (C.). – Towards a security SLA-based cloud monitoring service. 01 2014, pp. 598–603.
19. Shahzad (M.), Shafiq (M. Z.) et Liu (A. X.). – A Large Scale Exploratory Analysis of Software Vulnerability Life Cycles. – In *Proceedings of the 34th International Conference on Software Engineering*, pp. 771–781, Piscataway, NJ, USA, 2012. IEEE Press.
20. Teshome (A.), Rilling (L.) et Morin (C.). – Including Security Monitoring in Cloud SLA. – In *SEC2 2016 - Second workshop on Security in Clouds*, Lorient, France, juillet 2016.
21. Trapero (R.), Modic (J.), Stopar (M.), Taha (A.) et Suri (N.). – A novel approach to manage cloud security SLA incidents. *Future Generation Computer Systems*, vol. 72, 2017, pp. 193 – 205.
22. Zhang (S.), Zhang (X.) et Ou (X.). – After We Knew It : Empirical Study and Modeling of Cost-effectiveness of Exploiting Prevalent Known Vulnerabilities Across IaaS Cloud. – In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, *ASIA CCS '14*, ASIA CCS '14, pp. 317–328, New York, NY, USA, 2014. ACM.