Modélisation et test des ambiguïtés de recouvrement de données pour l'obtention des politiques de ré-assemblage dans les protocoles réseaux

Lucas Aubard, Johan Mazel, Gilles Guette, Pierre Chifflier, Olivier Levillain, Gregory Blanc, Ludovic Mé

Modélisation et test des ambiguïtés de recouvrement de données pour l'obtention des politiques de ré-assemblage dans les protocoles réseaux

Lucas Aubard
Inria
lucas.aubard@irisa.fr

Johan Mazel
ANSSI
johan.mazel@ssi.gouv.fr

Gilles Guette
Université de Rennes
gilles.guette@univ-rennes.fr

Pierre Chifflier
ANSSI
pierre.chifflier@ssi.gouv.fr

Olivier Levillain
Télécom SudParis,
Institut Polytechnique de Paris
olivier.levillain@telecom-sudparis.eu

Grégory Blanc
Télécom SudParis,
Institut Polytechnique de Paris
gregory.blanc@telecom-sudparis.eu

Ludovic Mé
Inria
ludovic.me@irisa.fr

***Résumé*—Les protocoles d'Internet, tels IPv4, TCP ou encore IPv6, disposent de mécanismes permettant de découper les données si nécessaire. Le découpage de ces morceaux peut donner lieu à du recouvrement, c'est-à-dire que plusieurs morceaux ainsi créés peuvent se chevaucher. Or, différentes politiques de ré-assemblage sont utilisées par les systèmes d'exploitation (OS), ce qui implique que selon l'OS de la machine destinataire, une même suite de fragments n'est pas ré-assemblée de la même manière. Dès lors, un système de détection d'intrusion (NIDS) qui ne ré-assemble pas ce type de fragments de la même manière que l'hôte surveillé est aveugle au flux réellement traité par cet hôte laissant la place à son contournement. Les derniers travaux qui documentent les politiques de ré-assemblage de fragments de données des OS datent de plus de 10 ans [1]–[3].**

Dans ce papier, nous proposons une approche par raisonnement spatio-temporel afin de modéliser l'ensemble des cas de recouvrement pour un nombre quelconque de fragments, pour ce faire, nous utilisons l'algèbre des intervalles d'Allen. Cela nous permet de mettre à jour les politiques de ré-assemblage des OS les plus utilisés (Windows, Linux, FreeBSD, SunOS) pour des paires de fragments IPv4. En outre, nous montrons que les politiques de ré-assemblage obtenues en testant des paires de fragments IPv4 ne permettent pas de prédire le ré-assemblage de $n > 2$ fragments recouvrants.

***Index Terms*—politique de ré-assemblage, recouvrement de données, algèbre des intervalles d'Allen**

I. Introduction

Un des moyens pour détecter les intrusions repose sur l'analyse du trafic réseau. Ptaceck et Newsham [4] évoquent, en 1998, les ambiguïtés auxquelles les systèmes d'analyse de trafic réseau (tels que les systèmes de détection d'intrusion (NIDS), par exemple) doivent faire face. Une des nombreuses ambiguïtés recensées par les auteurs provient du ré-assemblage de fragments lorsqu'il y a du recouvrement de données entre les fragments. Les RFC associées aux protocoles IPv4 [5] et TCP [6] n'interdisent pas le recouvrement et ne spécifient pas de comportement à privilégier si le recouvrement est présent. En ce qui concerne IPv6, le standard [7], qui précise qu'il faut rejeter un tel datagramme, ne semble pas suivi par toutes les implémentations de pile réseau [3].

La Figure 1 donne un exemple simple du problème de recouvrement de données. Par exemple, selon que la politique de ré-assemblage favorise le premier ou le second fragment, le paquet final reconstitué n'est pas le même. Une conséquence pratique est que si le premier ré-assemblage donne un paquet bénin et que le second donne un paquet avec une *paylaod* malveillante, si le NIDS ré-assemble selon le premier cas et l'OS destinataire selon le second, le NIDS passe à côté de l'attaque sans lever d'alerte.

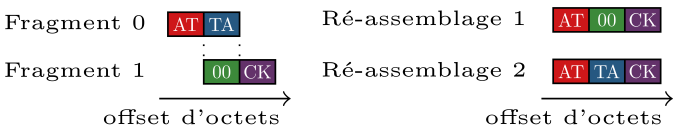


FIGURE 1 – Problème du recouvrement de données.

Plusieurs travaux [1]–[3], [8] ont montré que les OS résolvaient de manière différente le recouvrement de données. Cependant, aucun de ces travaux ne s'intéressent aux recouvrements de données entre un nombre de fragments supérieur à 2. Or, on ne peut pas être certains que les politiques de ré-assemblage pour des paires de fragments s'étendent aux cas où le nombre de fragments et de recouvrements sont supérieurs à 2.

Les contributions de ce papier sont 1) la modélisation des relations entre un nombre quelconque de fragments par un formalisme de raisonnement spatio-temporel, l'algèbre des intervalles d'Allen ; 2) la démonstration de la viabilité de notre approche en mettant à jour les politiques de ré-assemblage de paires de fragments IPv4 pour les familles d'OS Windows, Linux, FreeBSD et SunOS ; 3) l'étude préliminaire du cas des recouvrements impliquant plus de 2 fragments IPv4 et qui montre qu'on ne peut pas appliquer telles quelles les politiques de ré-assemblage de paires à un nombre $n > 2$ de fragments.

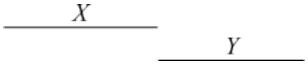
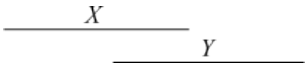
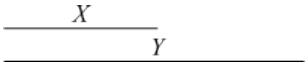
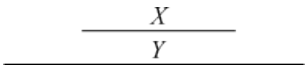
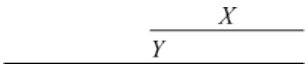
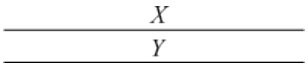
Relation	Illustration	Interprétation
X B Y Y Bi X		X précède Y
X M Y Y Mi X		X rencontre Y
X O Y Y Oi X		X recouvre Y
X S Y Y Si X		X commence Y
X D Y Y Di X		X pendant Y
X F Y Y Fi X		X finit Y
X Eq Y		X est égal à Y

TABLE I – Les 13 relations de l'algèbre des intervalles d'Allen.

OS	Version	Source	F	Fi	S	Si	O	Oi	D	Di	Eq
Windows	95 à XP	[8]	-	-	-	-	o	o	-	-	o
	95 à 2003	[1]	o	o	o	o	o	o	s	o	o
	10 (21H1)	Nous	∅	∅	∅	**o**	∅	∅	∅	**o**	∅
Linux	2.x	[8]	-	-	-	-	o	s	-	-	s
	2.x	[1]	s	o	s	s	o	s	s	o	s
	5.10.0	Nous	∅	∅	∅	**o**	∅	∅	∅	**o**	∅
FreeBSD	??	[8]	-	-	-	-	o	s	-	-	o
	??	[1]	s	o	o	o	o	s	s	o	o
	13.1	Nous	**s**	∅	**o**	**o**	**o**	**s**	**s**	**o**	∅
SunOS	4.1.4	[8]	-	-	-	-	o	s	-	-	o
	5.5.1 à 5.8	[8]	-	-	-	-	o	o	-	-	o
	5.10	[1]	s	o	o	o	o	o	s	o	o
	5.11	Nous	**s**	∅	**o**	**o**	**o**	**o**	**s**	**o**	∅

TABLE II – Politiques de ré-assemblage IPv4 pour des paires de fragments. **o** si fragment original préféré, **s** si fragment subséquent préféré, ∅ si aucune réponse obtenue et **-** si relation non testée. **Rouge** différent de [1] ; **Vert** identique à [1].

II. État de l'art

Afin que les NIDS puissent traiter le trafic réseau dont ils ont la surveillance de la même manière que l'OS de la machine destinataire, des travaux ont documenté les politiques de ré-assemblage des différents OS.

En 1998, Ptaceck et Newsham [4] cherchent à déterminer si les IDS résolvent divers cas de recouvrement de données de la même manière que l'hôte surveillé. Pour ce faire, ils développent et publient l'outil Fragroute [9] implémentant les cas de recouvrement Eq et F (voir Table I). Les auteurs montrent que les IDS à l'état de l'art en 1998 (*i.e.* RealSecure, NetRanger, SessionWall et NFR) échouent à ré-assembler les flux de données de la même manière que l'hôte FreeBSD 2.2 pour chacun des cas de recouvrement testé.

Shankar et Paxson [8] introduisent une séquence de fragments permettant de tester les cas O, Oi et Eq. Ils montrent la diversité des politiques de ré-assemblage IPv4 existantes : les différents OS testés ont permis de révéler 5 politiques IPv4 différentes de résolution des recouvrements (*BSD*, *BSD-right*, *Linux*, *First* et *Last*).

En 2005, Novak et Sturges [1] étendent les scénarios de test de Ptaceck et Newsham [4] et de Shankar et Paxson [8]. Ils permettent désormais de couvrir l'ensemble des scénarios de recouvrement possible, soit les 9 cas O, Oi, S, Si, F, Fi, D, Di et Eq. Ces tests supplémentaires ont permis de préciser la politique *BSD-right* ainsi que les politiques des OS Windows et SunOS. Quelques unes de ces politiques de ré-assemblage sont retranscrites dans la Table II.

Les travaux existants ciblent l'ambiguïté de recouvrement de données par la création de cas de tests construits manuellement. Or, cette méthode, si l'on souhaite l'exhaustivité, est impraticable pour un nombre de fragments $n > 2$. Nous répondons à cette limite par l'application du raisonnement spatio-temporel.

III. Modélisation du recouvrement de données par raisonnement spatio-temporel

Nous utilisons le raisonnement spatio-temporel introduit par Allen [10] afin de représenter l'ensemble des relations possibles entre un nombre quelconque de morceaux de données : l'algèbre des intervalles d'Allen est particulièrement bien adapté pour la représentation et la manipulation de morceaux de données pouvant se recouvrir. Il introduit 13 relations différentes pouvant lier 2 morceaux ; elles sont présentées dans la Table I. Nous utilisons l'outil SparQ [11], qui propose une implémentation de l'algèbre d'Allen, pour nos expériences.

Les relations introduites dans la Table I permettent de représenter l'ensemble des relations possibles entre 2 fragments de manière exhaustive et, la combinaison de plusieurs de ces relations permet la description de $n > 2$ fragments.

IV. Résultats

Dans cette section, nous rendons compte des politiques de ré-assemblage de fragments IPv4 des OS Windows 10, Debian 11, FreeBSD 13.1 et SunOS 5.11 représentant chacun une grande famille d'OS. Ces dernières disposent de leur propre politique de ré-assemblage dans les travaux les plus récents [1]. De la même manière que [1], nous déduisons ces politiques via un échange de paquets *ICMP Echo Request* et *ICMP Echo Reply*. Les paquets *ICMP Echo Request* sont fragmentés de sorte à tester les relations souhaitées.

A. Résultats pour 2 fragments

Les 13 relations spatio-temporelles, décrites dans la Table I, sont testées. La Table II rend compte des résultats pour les 9 relations impliquant un recouvrement.

Nous pouvons constater qu'aucune des politiques actualisées n'est complètement cohérente avec les politiques de ré-assemblage trouvées par Novak [1]. La grande majorité des incohérences est due à une non-réponse au *ICMP Echo Request* lorsqu'il y a du recouvrement de données. Seul

OS	Version	Ré-assemblage	X M Y	X S Z	Y Oi Z
Windows	10 (21H1)	attendu	Na	∅	∅
		obtenu	Na	o	o
Linux	5.10.0	attendu	Na	∅	∅
		obtenu	Na	o	o
FreeBSD	13.1	attendu	Na	o	s
		obtenu	Na	o	o
SunOS	5.11	attendu	Na	o	o
		obtenu	Na	o	o

TABLE III – Cohérence du ré-assemblage IPv4 des paires de fragments du triplet (X M Y, X S Z, Y Oi Z) avec le ré-assemblage de paires seules. **Na** : relation non recouvrante. **Rouge** différent du ré-assemblage attendu ; **Vert** identique au ré-assemblage attendu.

Linux montre un comportement différent pour la relation Si : il préférait le fragment subséquent alors qu'il préfère désormais le fragment original. Windows et Linux disposent désormais de la même politique de ré-assemblage pour les paires de fragments : leur politique est bien plus restrictive qu'auparavant concernant l'acceptation de paquets fragmentés, dans le sens où ils répondent à la requête seulement dans 2 des 9 cas de recouvrement. Les nouvelles politiques FreeBSD et SunOS ont peu évolué : seules les relations Fi et Eq n'obtiennent pas de réponse à la requête.

Ainsi, on constate que les piles réseaux des différents OS ont évolué. Elles résolvent de manière différente le recouvrement de fragments IPv4 par rapport aux OS précédemment testés.

B. Résultats préliminaires pour 3 fragments

1) Application de l'algèbre des intervalles d'Allen: Bien que manipulant initialement des paires, l'algèbre de Allen peut être utilisé pour modéliser des relations dans un triplet de fragments. Ce triplet sera modélisé par trois relations de Allen.

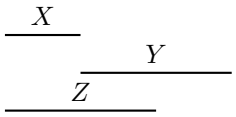


FIGURE 2 – Un exemple de triplet de fragments.

Par exemple, le cas exposé dans la Figure 2 est représentable par les relations de l'algèbre des intervalles d'Allen par le triplet (X M Y, X S Z, Y Oi Z). Tout triplet de fragment peut-être représenté par un triplet de relations de Allen. Cependant, tout triplet de relations ne représente pas forcément un cas possible (e.g. (X B Y, X Bi Z, Y B Z)). Après filtrage des cas impossibles à l'aide de l'outil SparQ [11], nous obtenons 418 triplets de relations cohérents qui représentent bien une situation réelle d'agencement valide de trois fragments.

2) Exemple d'incohérence: Le terme "incohérence" ici réfère au cas où, pour une relation d'Allen donnée, le ré-assemblage est différent entre 2 fragments testés individuellement (cf Table II) et 2 fragments au sein d'un triplet de fragments. La Table III résume les ré-assemblages des OS ainsi que leur cohérence avec les politiques qui s'appliquent sur les paires (cf. Table II). Ainsi, à part SunOS qui est cohérent dans son ré-assemblage (i.e. il répond en favorisant les données originales pour les relations S et Oi), les ré-assemblages obtenus ne sont pas cohérents. Windows et Linux répondent à la requête en favorisant les données originales alors qu'aucune réponse ne devrait être envoyée. FreeBSD, quant à lui, répond à la requête en favorisant, comme prévu, les données originales dans sa relation S. En revanche, il choisit les données les plus récentes lorsque confronté à la relation Oi, ce qui n'est pas le comportement attendu.

V. CONCLUSION ET FUTURS TRAVAUX

Nous modélisons le recouvrement de données au sein de protocoles réseaux via l'algèbre des intervalles d'Allen. De plus, nous démontrons la faisabilité de notre approche en mettant à jour les politiques de ré-assemblage des paires de fragments IPv4. Elles sont désormais plus restrictives qu'auparavant mais le recouvrement de données est toujours accepté dans 2 cas pour Linux et Windows et dans 7 cas pour SunOS et FreeBSD. Ces politiques de ré-assemblage changent lorsque plus de 2 fragments se recouvrent.

Dans de futurs travaux, nous souhaitons :

- trouver les politiques complètes (i.e. pour un nombre quelconque de fragments) des différentes familles d'OS mais aussi d'implémentations de piles réseaux embarquées (e.g. lwip, μip), *IoT*, d'implémentations dans du matériel (e.g. PLC de systèmes industriels), offloading dans les NIC ;
- prendre en compte plus de scénarios de tests (e.g. en faisant varier la position du bit MF valant 0) ;
- étendre la couverture de protocoles à IPv6, TCP et QUIC ;
- améliorer l'application des politiques de ré-assemblage dans les NIDS (prise d'empreinte, application de la politique à un flux).

RÉFÉRENCES

[1] J. Novak, "Target-based fragmentation reassembly," *Sourcefire, Columbia, MD*, 2005.

[2] J. Novak, S. Sturges, and I. Sourcefire, "Target-based tcp stream reassembly," *Aug*, 2007.

[3] A. Atlasis, "Attacking ipv6 implementation using fragmentation," *Black-hat europe*, 2012.

[4] T. H. Ptacek and T. N. Newsham, "Insertion, evasion, and denial of service : Eluding network intrusion detection," Secure Networks inc Calgary Alberta, Tech. Rep., 1998.

[5] "Internet Protocol," RFC 791, Sep. 1981.

[6] "Transmission Control Protocol," RFC 793, Sep. 1981.

[7] S. Krishnan, "Rfc 5722 : Handling of overlapping ipv6 fragments," 2009.

[8] U. Shankar and V. Paxson, "Active mapping : Resisting nids evasion without altering traffic," in *SP 2003*.

[9] "Fragroute," https://www.monkey.org/~dugsong/fragroute/.

[10] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, no. 11, 1983.

[11] "Sparq," https://github.com/dwolter/SparQ.