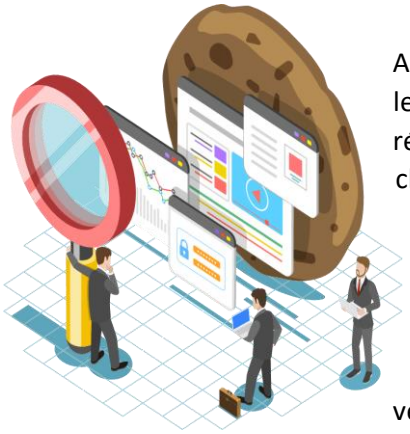
Crowds

Anonymous Web Transactions

Table des matières

I. Introduction

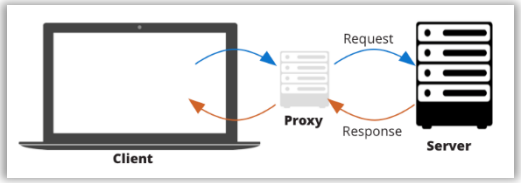


Avec l'avancement des sites web, et les services multiples qu'ils présentent, les données des visiteurs de ces sites sont devenues très importantes à récolter. En effet, avec le nombre de services très important, les entreprises cherchent à cibler leurs contenue et produits vers les clients qui pourront être les plus intéressés par ces produits. C'est pour cela que chaque site récolte toutes les informations possibles sur un utilisateur, et peut les vendre pour d'autres entreprises.

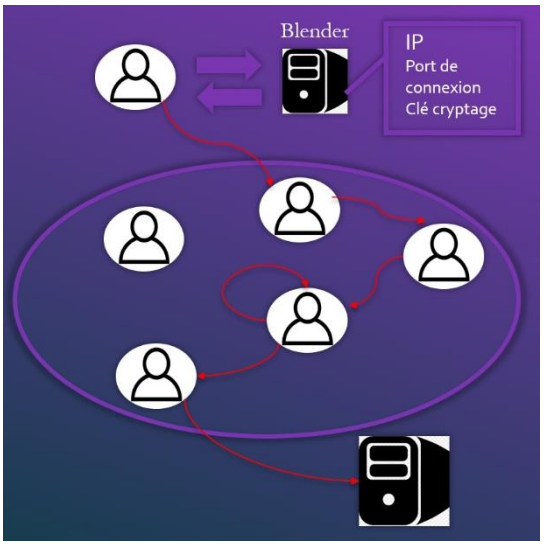
Il existe aujourd'hui plusieurs méthodes et protocoles afin de garantir un niveau plus ou moins important d'anonymat lors de l'envoie des requêtes vers un serveur web. Néanmoins, la définition d'anonymat est très large quand il s'agit de la navigation sur le web. Il peut s'agir de cacher l'identité du client lors d'une transaction avec un serveur web, ou même garantir un niveau de sécurité de toutes les données du client et empêchant les attaques sur ses données.

Le protocole Crowds inventé par Michael Reiter et Aviel Rubin, a pour but d'empêcher un serveur web ou des eavesdroppers (espions) d'identifier le client responsable de la transaction. D'où le nom l'indique, le protocole Crowds se base sur la configuration d'un groupes d'utilisateurs pour former un « Crowd», et utiliser cette foule pour envoyer ou recevoir des requêtes d'un serveur web.

II. Fonctionnement



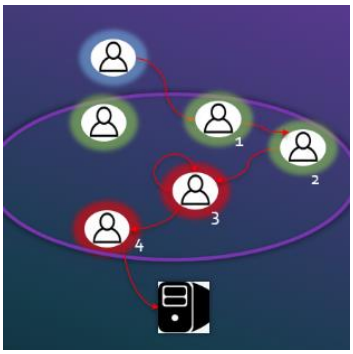
Une transaction normale entre un client, et un serveur web se fait comme l'indique le schéma suivant. Cette transaction se fait via un proxy, qui permet donc aussi au fournisseur d'accès internet, de surveiller les requêtes envoyées par le client. Afin de modifier ce chemin de transfert de données, le protocole Crowds propose d'utiliser un proxy sur le navigateur pour toutes les transactions d'internet, et qui va modifier le chemin d'envoi de ces données. L'utilisateur est représenté donc par un Jondo. Ce nom donné au Proxy vient du nom John Doe, qui en anglais est donnée çà toutes personnes avec un prénom inconnu. En effet, le navigateur du chaque client dans le Crowd envoie les requêtes à travers le Jondo et non le proxy local.



Le schéma suivant indue le fonctionnement général du protocole Crowds. Afin qu'un nouvel utilisateur se connecte à une foule, il doit d'abord se connecter à un serveur spécifique qui s'appelle Blender (Mixeur en français). Ce serveur contient toutes les informations de tous les Jondo dans le Crowd, comme leurs adresses IP, les ports de connexion, ainsi que la clé de cryptage. Toutes les communications sont cryptées entre les Jondo, ce qui empêche un eavsdropper local de pouvoir observer les échanges entre Jondo.
Tous les Jondo présents sont ensuite notifié qu'un nouvel utilisateur a rejoint le Crowd. La formation de chemin de transfert des requêtes est spéciale dans le protocole Crowds. En effet, c'est la façon avec laquelle est calculé le chemin qui permet de garantir l'anonymat de l'utilisateur.

Comme j'ai expliqué auparavant, le protocole Crowds consiste à cacher l'identité de l'envoyeur des requêtes. Donc, le message envoyé par le client est d'abord crypté avec la clé de cryptage donnée par le Blender. Ensuite, le Jondo de l'utilisateur passe la requête au Jondo suivant. La prochaine étape est différente de la première, la formation de chemin est complètement aléatoire. Or, sur chaque décision d'envoie de message au serveur ou de le passer au prochain a une probabilité qui n'est pas égale. La formation du chemin favorise l'envoie de message vers le serveur, avec une probabilité strictement supérieur à ½ . Comme on peut le voir sur le schéma, le Jondo peut aussi envoyer le message à lui-même ou de le renvoyer à l'initiateur original de la requête. L'initiateur ou d'autres utilisateurs peuvent en fait paraitre plusieurs fois sur le même chemin, ce qui garanti encore un niveau d'anonymat.
Le chemin n'est pas cofiguté non plus avec tous les Jondo, en fait la configuration des chemins minimise la probabilité qu'un Jondo apparait plusieurs fois sur un chemin d'un envoyeur.
Quand le Jondo décide en fin de passer la requête au serveur web, comme on le voit sur le schéma, le message est décrypté et transmis au serveur.
La réponse du serveur suit le même chemin que le chemin initial, afin de minimiser les reformulations de chemin. Ceci peut imposer des risques de sécurité qu'on va aborder ultérieurement. Néanmoins, en cas où un nouvel utilisateur rejoint la foule, il serait nécessaire de reconfigurer tous les chemins, afin de garantir que le nouvel utilisateur soit détecté directement.
En effet, les risques de sécurité, ne viennent pas seulement du serveur web, ou des eavsdropper locales, mais aussi des Jondo dits corrompus. Ces Jondo, peuvent même se collaborer entre eux afin de déterminer l'initiateur de la requête. C'est pour cela que le protocole Crowds se dispose de plusieurs méthodes afin d'empêcher cela.

Le protocole Crowds permet de garantir avec sa façon de configurer les chemins, une innocence probable des envoyeurs des requêtes. Reprenons le chemin au-dessus, en illustrant le cas où il y a des Jondo corrompus dans le système. Sur le schéma en-dessous, on represente l'initateur en bleu, les Jondo corrompus en rouges et les Jondo non corrompus en vert. Une méthode de détérminer l'initiateur dans un Crowd, et de supposer qu'il precede le premier Jondo corrompus sur le chemin. En effet, plus il y aura des Jondo corrompus qui se collaborent, plus il sera possible de détérminier l'initateur. Or, ces collaborateurs, ne peuvent



pas détérminer avec précision l'initateur, sauf si ce dernier était avant le premier collaboratuer.
Afin de prouver qu'un tel evenement est peu probable, c'est-à-dire il a une probabilité inféreur à 50%, nous allons défifinr quelques variables :

• Soit H_{k+} symbolise un Jondo, avec le premier Jondo corrompus à la posiotion K (K=>1), et supppose que l'initateur est à la position 0.
• Soit I symbolise que l'initateur à la position avant le premier collaboratuer corrompu.

- Soit C, le nombre de collaborateurs corrompus
- N, le nombre de Jondo total

Afin de garantir la probabilté d'innocence mentionnée auparavant, il faut que N soit définit de la façon suivante ;

$$n \geq \frac{p_f}{p_f - \frac{1}{2}}(c+1)$$

Avec Pf Probabilty of Fowrwording, c'est-à-dire la probalité que le Jondo passe le message au serveur au lieu de le passer au Jondo d'après. Cette probabilté est > ½ comme expliqué auparavant.

Soit la probabilté qu'un collaboratuer soit l'envoyeur de message vers le serveur ;

$P(H_i) = (p_f \frac{n-c}{n})^{i-1}(\frac{c}{n})$. Cette formule représente plusieurs probabilté, $\frac{n-c}{n}$ est donc la probailté que le collaborateur soit à la lième position de chamin , avec $\frac{c}{n}$ la probabilité que le message soit transmis à lui. Donc la formule suivante permet de calculer la probabilté que le premier collaborateur envoie le message au serveur ;

$$P(H_{1+}) = (\frac{c}{n})(\frac{1}{1-\frac{p_f(n-c)}{n}})$$ et donc la probabilté que le collabortauer d'après

Envoie le message ;

$$P(H_{2+}) = (\frac{c}{n})(\frac{\frac{p_f(n-c)}{n}}{1-\frac{p_f(n-c)}{n}})$$. Ces deux formules permettent ensuite de calculer la

probabilité que l'initateur de la requete précede le premier collaborateur ;

$$P(I|H_{1+}) = \frac{P(I \wedge H1+)}{P(H1+)} = \frac{P(I)}{P(H1+)} = \frac{n - p_f(n-c-1)}{n}$$

En remplaçant N par sa valeur montrée auparavant, et Pf par une valeur strictement supérieur à ½ . On obtient bient une probabilité inférieur ou égale à ½ .

Une autre méthode de déterminer l'initateur de la requete par des collaboratuers corrompus, et le chronométrage. En effet, certain liens Url contiennent des images, donc quand la page se charge le serveur web envoie de demandes de requetes automatique afin de charger les images. Les Jondo corrupmus mesurent ensuite le délai entre la première requete, et les requetes automatiques faites par l'envoyeur. Si ce délai est court, les Jondo peuvent en déduire que l'initiateur précede le premier

collaborateur ou non loin. Ce problème est resolu en analysant à l'avance la page HTML, et donc toutes les requetes nécessaire pour charger cette page sont envoyées d'un seul coup.

III. Performance

En se basant sur un test de performence fait dans l'annexe (6), nous allons analyser les resultats afin d'etuder la preformence du protocole Crowds. Ce modèle met en question la preformence du protocole quand il y a des reformulations de chamin de transfert. En effet en ayant plusieurs collaboratuers corrompus, il est possible de détécter l'initiateur de la requete, si ce dernier passe plusieurs foirs dans un chemin où il y les Jondo corrompus.

Crowd:		Path reformulations:			
		3	4	5	6
5 honest, 1 corrupt	Positive	0.138	0.235	0.333	0.427
	False positive	0.051	0.091	0.129	0.158
	"Confidence"	1.000	0.974	0.931	0.869
10 honest, 2 corrupt	Positive	0.104	0.181	0.263	0.346
	False positive	0.029	0.055	0.082	0.108
	"Confidence"	1.000	0.989	0.962	0.925
15 honest, 3 corrupt	Positive	0.094	0.165	0.241	0.318
	False positive	0.020	0.039	0.059	0.079
	"Confidence"	1.000	0.989	0.975	0.950
20 honest, 4 corrupt	Positive	0.089	0.156	0.230	0.305
	False positive	0.016	0.030	0.046	0.063
	"Confidence"	1.000	0.994	0.978	0.961
10 honest, 1 corrupt	Positive	0.037	0.068	0.105	0.145
	False positive	0.016	0.030	0.048	0.168
	"Confidence"	1.000	0.996	0.981	0.966
20 honest, 2 corrupt	Positive	0.030	0.055	0.086	0.120
	False positive	0.008	0.016	0.026	0.038
	"Confidence"	1.000	0.996	0.988	0.983

Cette simulation de performence a été faite avec une probabilité Pf égale à 0.8 et N prend la formule vue dans la partie précedente.

Afin de pouvoir analyser le tableau on définit les terms suivants ;

- **Positife** : Si cette valeur > 1, les Jondo corrompus ont pu observé l'initateur au moins une fois.
- **False positive** : Si cette valeur > 1 , les Jondo corrompus ont observé un Jondo autre que l'initiateur au moins une fois.
- **Confidence** : Si cette valeur =< 1, les Jondo corrompus ont observé seulement l'initiateur au moins une fois.

En effet, quand un nouveau utilisateur rejoint la foule les chemins sont reconfiguré. On peut bien observer que la variable Confidence est le plus bas (plus de chance de detcter l'initiateur et seulemnet l'initateur au moins une fois) quand on a le plus petit nombre de Jondo honnetes et avec 6 formulations. On remarque aussi qu'avec 5 Jondo honnetes et 6 reformulations, la variable Positive est la plus grande. Cette variable baisse en valeur comme on le voit pour qu'il atteint la plus petite valeur avec 20 Jondo honnetes et 2 Jondo corrompus avec 6 formulations. En compparant les differentes colonnes de tableau on remarque que plus il y a de reformulations de chemin, plus il sera possible de détécter l'initiateur. La situation optimale pour ce protocole est donc d'avoir le maximum de Jondo dans la foule , et ne pas reconfigurer les chemins que quand il est très nécessaire. On déduit aussi que le protocole est perfermant et garanti l'anonymat du client , avec la valeur de Confidence ne va pas plus bas que 0.869 dans le cas le moins optimal de fonctionnement.

IV. Désavantages et risques

Comme on a vu précedamenet le niveau d'anonymat de ce protocle est assez limité. En effet il n'assure pas une confidentialité de l'information envoyé par le client entre les Jondo. Il est même plus dengereux d'utiliser le protocole Crowds pour certains types de transactions web. Toutes les transcations nécessitant que le client envoie des informations confidentielles

comme des identifiants, mots de passe ou cordonnées bancaires ne sont pas assurées avec ce protocole. Pour des raisons de securité, certains site ont intérdit les transaction d'achat avec un protocole Crowds, car le numéro de credit bancaire peut être volé lors de la transaction. Il est donc conseillé d'utiliser le protocole Crowds pour des transcations où le client souhaite simplement proteger son identité sans vouloir proteger les données qu'il transmet.

Les applet Java peuvent présenter une risque pour le protocole Crowds. En effet, en executant le programme de certains navigateurs, des appelet Java peuvenet ouvrir une connexion avec le serveur web sans passer par le réseau Crowds. Ceci peut rendre l'identité du client visible par le serveur web, et donc ces applet java doivent être désavctivés afin de garantir le fonctionnement de Crowds.

Un désavantge de protocole Crowds est le délai supplémentaire de passer par des Jondo. Comme le chemin est calculé aléatoirement, le chemin peut ajouter un délai supplémentaire sur les transactions entre le client et le serveur web.

V. Conclusion

Le protocole Crowds permet d'avoir un certain niveau d'anonymat tout en ayant un système simple. Ce protocole permet de garantir l'anonymat d'un client contre un serveur web, un eavsdropper local ou même d'autres utilisateurs dans le réseau Crowd. Or, ce protocole présente des désavantage et parfois des risques, donc ce protocole ne doit pas être utilisé pour toutes type de transactions.

VI. Annexes

(1) https://en.wikipedia.org/wiki/Anonymous_web_browsing
(2) https://dl.acm.org/doi/10.1145/290163.290168https://dl.acm.org/doi/10.1145/290163.290168
(3) https://pdfs.semanticscholar.org/7880/382d2564609bb7415d83da1962850554f5e5.pdf?_ga=2.175852642.1985033930.1616322533-1949363918.1616322533
(4) http://www.lix.polytechnique.fr/~ehab/papers/crowds-trust.pdf
(5) https://cacm.acm.org/magazines/1999/2/7969-anonymous-web-transactions-with-crowds/fulltext?mobile=falsehttps://cacm.acm.org/magazines/1999/2/7969-anonymous-web-transactions-with-crowds/fulltext?mobile=false
(6) https://www.prismmodelchecker.org/casestudies/crowds.php#:~:text=The%20Crowds%20protocol%20was%20developed,a%20group%20of%20similar%20users.