Requêtes décentralisées en utilisant la propagation des transactions dans l'environnement blockchain

Bastien Etchegoyen

[1] ENSIMAG, Grenoble, France,
Bastien.Etchegoyen@grenoble-inp.org,
[2] Ecole Polytechnique de Montréal, Quebec, Canada
bastien.etchegoyen@polymtl.ca

Résumé Schéma d'un envoi décentralisé de requêtes en utilisant la blockchain comme moyen de propagation de commandes aux machines victimes d'un logiciel malveillant. Chaque requête est encapsulée dans une transaction blockchain. Cette approche permet à l'attaquant d'anonymiser entièrement ses actions tout en recevant les réponses de ses victimes.

Keywords: Blockchain, InfoSec, Bitcoin, Ethereum, Cyberattaque

1 Principe

1.1 Définition de la blockchain

L'utilisation de la blockchain pour ce schéma d'attaque repose sur la définition même de décentralisation. La blockchain offre un environnement pair à pair où les connexions se font à travers les noeuds du réseau. Chaque transaction intégrée à la blockchain suit un processus cryptographique précis veillant à garantir l'intégrité de cette dernière dans le temps, ce modèle permet de s'abstraire d'une autorité centrale de confiance et d'établir une confiance générale entre les utilisateurs sous forme de consensus cryptographique. Les principes cryptogaphiques suivent un chiffrement à clé publique/clé privée : pour recevoir une transaction un utilisateur diffusera sa clé publique, pour envoyer une transaction il signera cette dernière avec sa clé privée.

Une transaction contient entre autre l'adresse publique de l'envoyeur, l'adresse publique du destinataire, et le montant à envoyer. Puisqu'il n'y a pas d'autorité centrale dans la blockchain, ce sont les utilisateurs eux-mêmes qui vont exécuter ces transactions. Ainsi sous ce modèle, tous les utilisateurs voient les transactions qui transitent et participent à leur validation. Les transactions non-validées sont regroupées dans un bloc, et lorsque ce dernier valide certaines propriétés cryptographiques, on l'ajoute à la chaine des blocs déjà existants.

On parle de système pseudo-anonyme, l'anonymat de l'utilisateur qui se cache derrière son adresse publique est garanti, mais l'historique de ses transactions est visible par tous.

La blockchain est donc une entité publique, infalsifiable et permettant de retracer l'historique de toutes les transactions faites à ce jour. Cette propriété de transparence permet à quiconque de consulter l'historique des transactions entrantes et sortantes d'une adresse et d'en déduire son solde.

1.2 Intérêt de la blockchain

Dans la blockchain, une personne est identifiée par son adresse publique seule. L'adresse IP de la personne effectuant la transaction n'est pas sauvegardée dans la blockchain, elle est connue seulement en tant que noeud par le réseau. Il est impossible de faire la distinction entre un noeud à l'origine d'une transaction et un noeud qui la relaie [3], car tant que la transaction est valide elle est relayée automatiquement aux pairs qui ne la possèdent pas encore. Au moment de la propagation aucune adresse IP n'est associée à la transaction, et les transactions sont validées par des tiers.

Lors d'une attaque, réussir à préserver son anonymat est un des enjeux central. Mon idée est qu'en utilisant la blockchain pour propager de l'information, on rend intraçable l'attaquant puisque ce dernier est caché derrière une adresse publique cryptographique qui suit un algorithme connu et pourrait être générée par n'importe quel utilisateur. L'identité de l'attaquant est préservée, car seul le contenu cryptographique de la transaction transitera. De même, l'obtention de la cryptomonnaie pour effectuer les transactions est obtenue anonymement en minant sur le réseau.

On utilisera une blockchain publique pour faire transiter nos informations. Dans la plupart des blockchains publiques, il est possible d'envoyer de l'information en même temps que la transaction, qui se traduit la plupart du temps par un champ `DATA` dans la transaction. C'est ce champ que nous allons utiliser pour faire transiter de l'information et produire des comportements chez les machines déjà infectées qui le lisent.

Enfin, la blockchain est une structure impossible à censurer. Il n'y a pas de noeud central à faire tomber pour empêcher la propagation d'une transaction. On considère que la blockchain est stable tant qu'il n'existe pas d'entité possédant plus de 51% de la puissance de calcul [4]. Le hashrate courant du réseau bitcoin est actuellement évalué à 14.850.000TH/s [5] ce qui représente une puissance de

3. https ://en.bitcoin.it/wiki/Network#Standard_relaying
4. https ://en.bitcoin.it/wiki/Majority_attack
5. https ://blockchain.info/fr/charts/hash-rate

calcul bien supérieure aux meilleurs super-calculateurs du monde combinés [6]. Ainsi dans notre schéma d'attaque, cela se traduit par l'impossibilité technique de faire chanceler la blockchain pour empêcher une transaction d'infecter un utilisateur.

1.3 Étude des coûts et temps de propagation

Une blockchain utilise le système de cryptomonnaies pour l'envoi de transactions. Dans cette Proof Of Concept nous allons comparer les deux blockchains publiques dominantes [7] pour évaluer celle qui correspond le mieux à notre besoin. Nous avons besoin d'envoyer des données en même temps que notre transaction, le coût d'une transaction sera un critère non négligeable. On étudie aussi le temps de propagation sur le réseau, les coûts fixes par transaction, le coût par bytes de données envoyées. [8] [9] [10] [11]

La taille maximale des champs de données est différente dans chaque blockchain mais son seuil n'est pas un soucis : on fractionnera les requêtes lourdes en plusieurs transactions.

	Bitcoin	Ethereum
Prix à l'unité	11.600$	468$
Nombres de bytes d'une transaction classique	190	150
Coût moyen par bytes	1.5e-7 BTC	1.56e-6 ETH
Coût moyen par bytes en dollars	0.00174$	0.0007332$
Temps de propagation	10min	30sec

Au vu de cette analyse nous allons donc utiliser le réseau Ethereum pour optimiser les performances et minimiser le coût de notre logiciel.

1.4 Comportement du logiciel malveillant

On considère un duo attaquant/victime où l'attaquant enverra des requêtes à la victime pour lui faire exécuter un comportement prédéfini. La victime est en possession d'un logiciel infecté. Pour cet article on ne s'occupe pas de savoir comment elle a obtenu ce dernier (clef usb, mail...) ni comment le logiciel s'exécute sur la machine victime (au démarrage, modification registre, migration

6. https ://fr.wikipedia.org/wiki/TOP500
7. Celles qui possèdent les capitalisations boursières les plus élevées https ://coinmarketcap.com
8. Nombre de bytes pour une transaction bitcoin https ://blockexplorer.com/
9. Nombre de bytes pour une transaction ethereum https ://etherscan.io/blocks
10. Coût moyen par bytes bitcoin https ://bitcoinfees.earn.com/api/v1/fees/recommended
11. Coût moyen par bytes ethereum calculé avec un Gas Price de 10 Gwei

vers un processus...).

Le logiciel comporte en interne une adresse publique bitcoin/ethereum que l'attaquant aura préalablement créé. Cette adresse sera vérifiée à intervalle régulier par le logiciel. Dès qu'une nouvelle transaction est reçue à cette adresse, le logiciel va récupérer le champ `DATA` contenu au sein de la transaction et exécuter une commande en fonction de cette dernière. En fonction des informations reçues par l'attaquant, la victime exécute une série d'instructions impliquant ou non une réponse.


```
"inputs": [
  {
    "sequence": 5,
    "addresses": [
      "45ed4d2dc57870d3cb2009db5857670a5f2b1c6b"
    ]
  }
],
"outputs": [
  {
    "value": 100000000000000,
    "script": "00acd91683",
    "addresses": [
      "3e58ee6294a8ee14abd5207d7bd73466a0ae93f4"
    ]
  }
]
```


Figure 1. Extrait du contenu d'une transaction sur le réseau Ethereum

1.5 Limites

L'envoi de transactions nous soumet à certaines restrictions :

1. Le passage d'informations via une blockchain publique étant coûteux, l'attente d'une réponse lourde se révèle problématique au vu des frais conséquents qu'elle engendre.
2. Le fractionnement des requêtes permet de contourner le problème lié à la taille maximale de données envoyées par transaction, mais ne corrige pas le problème du coût.
3. Il est impossible d'évaluer à l'avance le temps de transfert, d'exécution et de réponse du client.

4. L'envoi et les réponses sont exposés publiquement dans la blockchain, un client qui se rend compte de l'infection peut ainsi cibler précisément les comportements qui lui ont été administrés et les données qu'il a retourné.
5. Pour réduire le coût d'une transaction dans la blockchain Ethereum, il est judicieux de réduire optimalement le `gas price`[12]. Cependant une valeur trop basse donnera une très faible priorité à la transaction et pourrait faire exploser son temps de propagation voire la noyer (elle ne serait jamais minée car pas assez profitable).

2 Transfert d'information via la blockchain

2.1 Envoi de commandes

Le but est de construire un champ de données permettant l'exécution de commandes différentes. Pour cela on met en place un système d'Opcode traduisant le type de requêtes à effectuer en fonction du premier octet du champ de données.

Dans cette POC, il a été fait les choix suivants :

OP code	Comportement
0	ping <ip >
1	delete <path >
2	Envoyer requête HTTP <url >
3	Modifier valeur registre <Reg > : <Val >
4	Exécuter commande <cmd >
5	Fractionner l'envoi d'une commande <Numéro du paquet ><cmd >

Figure 2.

On souhaite par exemple faire exécuter une commande ping à l'adresse 172.217.22.131. En suivant le tableau prévu, on encode notre commande en débutant par l'opcode 00 puis l'adresse IP. Dans un soucis de minimisation des coûts de transfert, on maximise l'entropie pour limiter au maximum le nombre de bytes transférés sur le réseau. On encode donc l'adresse IP sur 4 octets consécutifs.

12. https ://myetherwallet.github.io/knowledge-base/gas/what-is-gas-ethereum.html #gas-price

IP	HEX
172	AC
217	D9
22	16
131	83

Ainsi en envoyant une transaction contenant `0x00acd91683` dans le champ de données, le logiciel qui surveille cette adresse va décoder l'information reçue, et déduire qu'il doit exécuter une requête ping à l'adresse 172.217.22.131. Un extrait de cette transaction est présenté dans la figure 1.4, où le champ `script` est le champ de données de la blockchain Ethereum.

2.2 Vérification dans la blockchain

La blockchain ethereum est publique, n'importe quel utilisateur a accès à l'API pour consulter les transactions du réseau. Le logiciel observe le trafic lié à son adresse puis, dès qu'une nouvelle transaction est reçue il récupère le champ data de cette dernière. Pour utiliser l'API il faut possèder l'intégralité de la blockchain sur sa machine, qui représente des dizaines de GB à ce jour. Dans un soucis pratique et puisque tous les utilisateurs ne possèdent pas la blockchain sur leur PC, on utilise côté victime une API Web qui va interfacer cette dernière.

2.3 Comportement logiciel

Le logiciel côté victime observe son adresse dans la blockchain toutes les 15 secondes. Dès qu'une nouvelle transaction est reçue, il décode son champ `DATA` et exécute le comportement attendu.

3 Propagation

3.1 Attaque unidirectionnelle 1-N

L'intérêt réside dans le fait d'hardcoder une adresse publique dans le logiciel, potentiellement distribué à N utilisateurs. Cette adresse est vérifiée via une api publique. Dès que cette adresse recoit transaction, le client effectue la commande associée au champ `DATA` fournis. Aucune réponse n'est requise de sa part et tous les clients possédant cette adresse exécutent la commande associée. L'attaquant n'a à priori pas de retour utilisateur sur l'exécution.

Intérêt Cela permet de faire exécuter sans distinction un comportement à autant de cibles que disponibles, utile si l'on souhaite réaliser une action qui demande le plus de puissance de calcul possible.

3.2 Attaque bidirectionnelle 1-1

L'intérêt réside dans le fait d'hardcoder une adresse publique ainsi qu'une clef privée dans le logiciel (pour ethereum, la clef privée suffit). Ce couple de clefs est choisi préalablement par l'attaquant. Lorsque l'adresse associée à cette clef reçoit une transaction, le client effectue la commande associée au champ `DATA` fournis.

Si la requête requiert une réponse, il utilise le surplus de monnaie reçu par l'attaquant pour renvoyer les informations à l'attaquant toujours en passant par la blockchain depuis son adresse en signant avec la clef privée hardcodée.

Pour répondre, le client utilise soit la blockchain qu'il possède sur son ordinateur soit une API web qui interface les appels à la blockchain.

Intérêt Cela permet à l'attaquant de cibler une personne en particulier et d'obtenir une réponse de sa part

3.3 Attaque bidirectionnelle 1-N

Cette attaque repose sur une blockchain théorique ne necessitant pas de coût de transaction, où par exemple le consensus est atteint non pas via une équipe de mineurs mais par la contribution des utilisateurs qui valident entre eux leurs transactions. [13]

Le principe réside dans le fait d'hardcoder une adresse à vérifier dans le logiciel et de générer automatiquement au besoin des couples clé publique/clé privée dans cette blockchain.

Dès que le client reçoit une transaction, le logiciel client effectue la commande associée au champ `DATA` fournis. Si la requête requiert une réponse, le client génère alors une adresse, et envoie de l'information depuis cette adresse dans la blockchain à l'adresse de l'attaquant (celui qui a envoyé la transaction initialement).

S'affranchir des frais de transaction offre beaucoup plus de malléabilité dans le sens fonctionnel du logiciel, car l'attaquant n'aurait plus à restreindre le nombre de transactions qu'il effectue et pourrait recevoir des réponses fréquemment de ses victimes (uptime, nombre de machines en écoute, information sur chacune des machines etc...)

13. Consensus inspiré de la blockchain IOTA https ://learn.iota.org/faqs

4 Perspectives d'amélioration

Utilisation de blockchain publique plus souple en terme de coût de données. Exemple : SIA, filecoin, Stellar Lumen...

Utilisation de blockchain publique sans frais de transaction. Exemple : IOTA.

Limites Moins connues et moins matures, ces blockchains n'offrent pour le moment pas autant de facilité d'utilisation que leurs soeurs.

Idée Utilisation d'une blockchain privée pour exécuter exactement le comportement attendu. L'attaquant envoie ses transactions dans cette blockchain. Pas de frais, la puissance de calcul des victimes est utilisée pour miner les transactions. Les victimes deviennent les mineurs de leur propres transactions infectées.

Références

1. Code source du projet https ://github.com/Betcheg/BlockchainQueries
2. Ethereum Yellow-paper http ://gavwood.com/paper.pdf
3. Info sur les transactions ethereum https ://ethgasstation.info/
4. Coût moyen par bytes Bitcoin
   https ://bitcoinfees.earn.com/api/v1/fees/recommended
5. Exploreur de la blockchain Bitcoin https ://blockexplorer.com/
6. Exploreur de la blockchain Ethereum https ://etherscan.io/
7. Statistiques cryptomonnaies https ://bitinfocharts.com/