Analyse des journaux d'un Honeypot Web

Elian Loraux, Arthur Gaillard et Lilian Steimer
Étudiant à l'Ensimag
elian.loraux@grenoble-inp.org
arthur.gaillard@grenoble-inp.org
lilian.steimer@grenoble-inp.org

1er février 2024

Résumé

abstract : Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
keyword : Honeypot, sécurité, HTTP(s), SQL.

1 Introduction

Dans le cadre du module 'Introduction à la cybersécurité', nous avons mis en place un honeypot [1] web. Nous allons y capturer toutes les connexions au serveur.

Les honeypots HTTP et HTTPS sont des outils cruciaux pour la recherche scientifique en cybersécurité. Ces leurres informatiques sont délibérément configurés pour simuler des serveurs web vulnérables, attirant ainsi les cybercriminels et permettent d'étudier leurs tactiques, techniques et procédures. Les honeypots HTTP imitent des sites non sécurisés, tandis que les honeypots HTTPS simulent des sites sécurisés avec chiffrement. En analysant les attaques dirigées contre ces honeypots, nous pouvons mieux comprendre les tendances actuelles en matière de cyberattaques, identifier de nouvelles vulnérabilités et élaborer des contre-mesures plus efficaces pour renforcer la sécurité des systèmes informatiques. En résumé, l'utilisation de honeypots HTTP et HTTPS dans la recherche scientifique offre une plateforme d'observation précieuse pour étudier les menaces en ligne et améliorer la résilience des infrastructures numériques.

Dans ce document, nous allons vous expliquer le contexte de la mise en place du honeypot. Quel système a été mis en place, quelle machine a été utilisée, quelle durée, etc. Puis, dans un second temps, nous allons faire une analyse des journaux afin de pouvoir voir quelques patterns, les éléments principalement recherchés, les pays de provenance, etc. Pour finir, nous allons essayer de concorder quelques tendances avec des explications plausibles..

2 Contexte

Au départ, nous voulions déployer le projet T-pots pour ses multiples fonctionnalités comme la visualisation des journaux intégrés, le déploiement de plusieurs types de honeypot, ou sa finesse de configuration. Cependant, toutes ces fonctionnalités entraînent une complexité de mise en place trop

1. `Pot de miel`, service qui parait vulnérable afin de collecter des journaux symptomatique d'attaque

importante vis-à-vis du temps disponible pour le projet. Nous avons donc choisi le projet qeeqbox pour déployer le honeypot. C'est un projet open source qui est en fait une brique du projet T-pot. Nous avons choisi ce projet pour sa versatilité et sa facilité de mise en place. C'est un module Python qui s'installe simplement avec le gestionnaire de paquets `pip` et qui permet de déployer des honeypots en une seule commande. Vous pouvez retrouver toutes les commandes nécessaires sur le lien du projet.

Nous avons ainsi déployé un honeypot HTTP et un HTTPS sur un serveur ubuntu hébergé par Amazon Web Services (AWS). L'avantage d'utiliser AWS est d'éviter toute attaque réelle sur nos infrastructures personnelles et de bénéficier d'une instance gratuite. Cependant, cela peut potentiellement introduire quelques biais. En effet, il est facilement imaginable qu'un pirate préfère s'attaquer à une cible ne bénéficiant pas du support d'Amazon.

Le système de honeypot utilisé demande quelques configurations, notament pour les ports utilisé (ici 80 et 443) et la redirection des journaux. Pour gérer ces paramètre, on utilise des fichier de configuration que vous retrouverez en annexe.

La capture de journaux à débuté le 06/12/23 et s'est fini le 12/12/23. La capture aurait dû durée plus longtemps mais un problème technique a arreté la capture.

3 Analyse des journaux

Nous allons maintenant faire l'analyse des journaux. Lors de la capture des journaux, nous avons récupéré 1514 lignes de connexions. Ensuite, nous avons récupéré tous les journaux via un script Python (cf annexe) pour extraire les informations JSON des logs afin de les exporter en CSV, facilitant ainsi leur analyse à l'aide d'un tableur type Excel ou OpenCalc. Cela nous sera utile pour tracer des graphiques et appliquer des tris.

Voici une ligne type des log reçu :


```
{
  "action": "connection",
  "data": {
    "Accept": "text/html,application/xhtml+
    xml,application/xml;q=0.9,image/webp,*/*;
    q=0.8",
    "Accept-Encoding": "gzip, deflate",
    "Accept-Language": "en-GB,en;q=0.5", "
    Authorization": "Basic
    cmVwb3J0OjhKZzBTUjhLNTA=",
    "Connection": "close",
    "Host": "13.39.18.9",
    "Upgrade-Insecure-Requests": "1",
    "User-Agent": "Mozilla/5.0 (X11; Ubuntu;
     Linux x86_64; rv:76.0) Gecko/20100101
    Firefox/76.0",
    "method": "GET",
    "uri": "/"
  },
  "dest_ip": "0.0.0.0",
  "dest_port": "80",
  "server": "http_server",
  "src_ip": "38.75.137.31",
  "src_port": "35228",
  "timestamp": "2023-12-12T12:17:55.694875"}
```


Dans ce fichier, nous avons :

- L'adresse IP source
- Le port source
- Le User-agent
- L'URL de la requêtes
- L'heure de la requête
- Le pays de provenance de l'adresse IP
- Un nom de domaine lié à l'adresse IP

Pour l'analyse, nous allons commencer par faire une analyse générale, puis nous ferons une analyse individuelle des champs User-Agent, URL et domaine. Enfin, nous finirons par faire une analyse des patterns d'attaques. Chacune des analyses fera l'étude de ses résultats et donnera des explications ou des conclusions les concernant.

3.1 Analyse générale

Pour commencer, il est important de souligner que nous avons traité les journaux et effectué des calculs d'occurrence par rapport au nombre total de requêtes.

Nous avons effectué des recherches sur les adresses

IP sources de nos fichiers CSV et nous avons retracé le pays d'origine de ces adresses IP. Néanmoins, il y a une trentaine de requêtes pour lesquelles nous n'avons pas réussi à trouver exactement le pays. Nous savons seulement qu'elles viennent d'Europe. Sur les 1514 requêtes recu, voici les occurences de chaques pays :

- US = 719
- Allemagne = 240
- Chine = 233
- Royaume-Uni = 105
- Ukraine = 56
- EU = 33
- Pays-Bas = 30
- Canada = 24
- Russie = 16
- France = 14
- Maroc = 12
- Inde = 11
- Suisse = 10
- Suède = 8
- Pologne = 4
- Philippines = 4
- Brésil = 4
- Roumanie = 3
- Mongolie = 3

Nous avons également le Vietnam, Singapour, la Moldavie, le Japon, l'Italie, l'Iran, l'Indonésie, l'Espagne et l'Argentine avec deux requêtes. Il y aussi la Thaïlande, Taïwan, le Panama, la Lituanie, le Kazakhstan, l'Islande, l'Irlande, la Finlande, la Colombie, le Chili, la Bulgarie, la Bosnie, l'Albanie, la Corée du Sud et l'Afrique du Sud avec une seule requête.

Au niveau de la provenance des pays, les États-Unis et la Chine ne sont pas une surprise au vu de l'effervescence de la cybersécurité et de la cybercriminalité. Cependant, la présence de l'Allemagne est un peu plus surprenante. Deux explications sont possibles : certains hébergeurs allemands sont moins regardants sur des pratiques comme le spoofing, l'autre possibilité est que, en Allemagne, le scan est illégal, sauf s'il est à des fins éducatives. Il suffit donc pour quelqu'un faisant du scan de simplement indiquer que c'est à des fins éducatives, théoriques, cela réduit ses chances d'un bannissement d'IP.

On peut rapporter cela à l'échelle des continent et on obtient :

- Amérique = 751
- Europe = 502
- Asie = 265
- Afrique = 15

En premier, on retrouve l'Amérique. C'est logique et explicable par sa grande taille et par le nombre élevé d'entreprises qui scannent le web

En deuxième position, nous avons l'Europe. C'est assez étonnant car nous avions fait l'hypothèse que l'Asie serait en seconde position. Il y a plusieurs raisons à cette position. Tout d'abord, il faut prendre en compte le fait que nous n'avons qu'une semaine de journaux. Peut-être que l'Asie aurait été plus présente si nous avions pu laisser la capture tourner plus longtemps. De plus, il faut prendre en compte que ce sont 240 journaux sur 502 qui proviennent d'Allemagne. Ce fait peut s'expliquer par le manque de sécurité des hébergeurs allemands car, comme nous vous l'expliquerons plus bas, presque toutes les requêtes allemandes ont été faites depuis un serveur d'un hébergeur.

En troisième position, nous avons l'Asie. Ce n'est pas étonnant sachant que 233 requêtes sur les 265 proviennent de Chine. La Chine reste réputée pour son contrôle des réseaux informatiques avec leur Great Firewall. Ce n'est donc pas étonnant qu'elle scanne énormément le web à la recherche d'informations ou de cibles.

Enfin, on retrouve l'Afrique avec 12 requêtes sur 15 provenant du Maroc. L'Afrique n'est pas un continent qui est réputé pour les scans du web. Par contre, elle est assez réputée pour le phishing et l'ingénierie sociale.

Au final, nous avons des journaux d'un peu par-

tout dans le monde, sauf d'Océanie. On observe une grande diversité, mais il y a quand même une grande hégémonie des États-Unis.

3.2 Analyse par champs

Nous allons maintenant faire l'analyse par champs. Pour rappel, il y a les champs : 'User-agent', 'URL' et 'Domaine'. L'utilité de chacun des champs vous sera expliquée dans leur section.

3.2.1 Champs "User-agent"

Le 'User-agent' correspond à l'outil utilisé pour accéder à notre site web, ou tout simplement, l'outil qui permet d'interroger le service web de notre honeypot. Cette analyse est utile car elle nous permet de voir si des outils/logiciels spécifiques ont été utilisés pour faire des requêtes à notre serveur web.

Voici la liste des différent "User-agent" observé :

- Go-http-client
- python-requests
- Mozilla/5.0
- zgrab
- curl
- Python-urllib
- masscan
- fasthttp
- xfa1

On peut catégoriser ces différent en plusieurs type.

1. Les navigateurs : un navigateurs classique comme chrome, Firefox ou Safarie peut vous permettre d'accéder à notre serveur web. Ici, on le retrouve avec le "User-agent" Mozilla/5.0.
2. Les scanner : les scanners sont des logiciels concu pour scanner tout les adresses IP possibles et détecter les services présent dessus. Ici, nous avons masscan, xfa1, python-request.
3. Les crawler : Les crawler sont une sous partie des scanners. Ils scannent la cible demander par l'utilisateur et vont scanner ses services web pour en savoir plus. Il faut aussi vérifier ce qui est présent sur les pages web. Ici, on retrouve curl, zgrab, go-http-client, Python-urllib et fasthttp.

Ces différents 'User-agent' nous apprennent des choses concernant les entités qui ont interrogé notre serveur et leur but. Les crawlers et les scanners sont là pour scanner notre serveur afin d'obtenir des informations dessus et de connaître d'éventuelles failles. Ils sont notamment utiles car ils permettent de lancer des scans ou des attaques massives à l'aide de scripts. C'est notamment le cas pour curl qui est du Bash et les bibliothèques Python. Les navigateurs sont un peu plus difficiles à cerner. Ce sont peut-être des personnes isolées qui ont préalablement fait un scan de notre serveur et qui tentent d'attaquer notre site. Cela peut aussi être des machines infectées où tourne un script qui scanne Internet pour des pirates. Dans les deux cas, leur but est d'obtenir des informations sur notre serveur pour d'éventuelles attaques futures.

3.2.2 Champs "URL"

Parlons maintenant du champ 'URL'. L'URL désigne l'adresse d'un site internet (par exemple : www.google.com). Dans cette adresse, il y a la racine du site web, puis les différents fichiers et dossiers qui composent le site. Les différents fichiers et dossiers sont ceux qui sont affichés lorsque l'on fait une recherche d'un site. Lorsqu'on navigue sur un site web, on change ainsi de page. Toutes les pages ne sont pas accessibles par des requêtes ; certaines sont protégées et seulement accessibles par une page de connexion du site web. C'est la partie d'internet qu'on appelle le deep web, des pages internet qui ne sont pas directement indexées et qui sont seulement accessibles à travers les sites web. Ces différentes pages peuvent renfermer des informations capitales pour les attaquants, notamment sur les versions de certains logiciels pour connaître les failles.

Dans les dossiers des sites web, on retrouve aussi des fichiers de configuration, des fichiers contenant des variables, des fichiers contenant les liens des

sites non indexés, etc. Tous ces fichiers sont capitaux pour les sites web et sont donc très intéressants pour les attaquants. Ainsi, on observe de nombreuses tentatives d'accès à des pages non indexées ou des fichiers critiques.

Nous avons donc fait une liste des fichier et dossier les plus recherchés :

- .env
- /Core/Skin/Login.aspx
- /.git/config
- /systembc/password.php
- /phpinfo.php
- /.aws/credentials
- /robots.txt

.env : Le fichier `.env` est un fichier qui contient des variables d'environnement. On peut stocker de nombreuses informations dans ces variables, comme des chemins ou parfois des mots de passe. Ce fichier n'est normalement pas utilisé sur les serveurs web, mais il existe des frameworks PHP tels que Laravel qui en ont besoin. Ainsi, une mauvaise configuration du serveur web peut entraîner l'exposition de ce fichier sur le web.

/Core/Skin/Login.aspx : Un fichier ASPX est un type de fichier associé à ASP.NET, un framework développé par Microsoft pour la création d'applications web dynamiques. ASPX signifie "Active Server Pages eXtended" et représente des pages web qui contiennent du code source ASP.NET. Ces fichiers sont généralement écrits en langage de programmation C# ou Visual Basic.NET. Lorsqu'une page ASPX est demandée par un utilisateur, le serveur web exécute le code côté serveur et génère une page HTML dynamique qui est renvoyée au navigateur de l'utilisateur.

On peut supposer que ficher `/Core/Skin/Login.aspx` est un chemin vers un éléments critique de la gestion de connexion du framework ASP.NET.

/.git/config : Le fichier `/.git/config` est recherché pour savoir si un projet Git est présent ou non sur le serveur. Si oui et que le projet est mal protégé, alors un attaquant peut accéder à l'historique du projet et trouver une version où il n'y a pas ou moins de protection sur le serveur, facilitant ainsi le piratage. Il peut également trouver des mots de passe en clair sur des anciennes versions du projet.

/systembc/password.php : Visiblement, SystemBC est un outil de cybermalveillance servant de porte dérobée pour, notamment, lancer un rançongiciel. Le fichier `/systembc/password.php` est peut-être le fichier pour accéder à cette porte dérobée.

phpinfo.php : Le fichier `phpinfo.php` est une page internet contenant des informations sur la version de PHP utilisée par le serveur web. De nombreuses requêtes l'ont cherché dans plusieurs dossiers différents. Connaître la version de PHP d'un serveur est un atout pour un attaquant car, une fois de plus, il peut connaître les failles de sécurité qui en découlent.

/.aws/credentials : Le fichier `/.aws/credentials` est un fichier qui permet de spécifier les mots de passe et les comptes d'un serveur AWS en cas de perte du mot de passe ou de la clé SSH. Si un attaquant atteint ce fichier, il peut donc redéfinir le mot de passe des utilisateurs du serveur et avoir un accès à celui-ci.

robots.txt : Le fichier `robots.txt` contient les chemins des pages qui ne doivent pas être indexées. Globalement, les crawlers de Google ou d'autres sociétés vont scanner les pages et mettre un index sur celles-ci pour qu'elles soient accessibles via une recherche. Cependant, le crawler ne va pas prendre en compte les pages indiquées dans le fichier robots.txt. Ainsi, si un attaquant parvient à accéder à ce fichier, il connaîtra tous les chemins des pages non indexées du serveur en question.

Il y a bien sûr d'autres URLs, mais ce sont celles qui sont le plus revenues et bien sûr les plus critiques en cas d'exploitation.

3.2.3 Champs "Domaine"

Lorsque nous avons effectué le traçage des adresses IP sources, nous avons récupéré le pays de provenance, mais aussi un nom de domaine associé lorsqu'il y en avait un. Le nom de domaine peut nous indiquer la nature de l'entité qui a interrogé notre serveur. Ainsi, nous avons listé ces 4 types d'entités et leur occurrence :

- Scanner
- Cloud
- Domicile

Les scanners sont tout simplement des entreprises qui scannent l'intégralité d'Internet. Certaines de ces entreprises le font pour dresser une carte d'Internet en temps réel, et d'autres le font pour détecter les failles sur des serveurs puis pour prévenir les propriétaires. Ils font également des statistiques des failles présentes sur Internet. Dans ce lot, nous avons les domaines : censys-scanner.com, security.ipip.net, internet-census.org, cyberresilience.io, security.criminalip.com, Xpanse, shadowserver.org et stretchoid.com.

Ensuite, nous avons les serveurs cloud. Nous observons énormément de scans et d'attaques provenant de serveurs hébergés chez des sociétés de cloud computing. Les serveurs sont souvent gratuits pendant un certain temps ou peu chers. Il est facile pour un attaquant d'avoir un serveur et de l'utiliser à des fins illégales. C'est aussi bien plus facile que de réussir à pirater des machines pour en faire des bots. De plus, on peut reprendre les cas précédents depuis les serveurs cloud allemands qui ne sont pas bien contrôlés. Dans la liste des hébergeurs cloud, on retrouve : OVH, Amazon AWS, Akamai.com, Glesys.com, DigitalOcean.com, Googleusercontent.com, Privatelayer.com et Hostglobal.plus. Les domaines recensés en tant que cloud composent environ 67% de tous les domaines recensés.

Enfin, la catégorie 'domicile' regroupe toutes les adresses IP provenant de FAI. Ce sont donc, en général, des serveurs ou ordinateurs domestiques. Ainsi, il y a deux options possibles concernant ces postes. Soit les utilisateurs effectuent délibérément des scans et attaques à domicile. Soit ce sont des machines infectées, dites zombies, sur lesquelles tourne un script. Cette dernière catégorie est très faible car elle compose environ que 0.04% des noms de domaine recensé.

3.3 Analyse d'attaques

Nous allons maintenant faire l'analyse d'attaque. Ce qu'on appelle attaque, ce sont tout simplement des entités qui vont envoyer énormément de requêtes à la recherche de fichiers ou dossiers décrits plus tôt dans la partie URL. Ces attaques peuvent être comparées à du brute force car ces paquets sont envoyés dans un laps de temps très court et sont donc générés par des scripts. Nous avons sélectionné trois attaques qui nous paraissent pertinentes.

3.3.1 Attaque n°1

La première attaque provient des États-Unis. Elle émane du domaine linodeusercontent.com, qui est un domaine de la société Akamai spécialisée dans le cloud hosting. Après vérification, nous avons constaté que le domaine avait une très mauvaise réputation et qu'il générait de nombreuses attaques dans le monde.

L'attaque a généré 149 lignes dans nos journaux, ce qui signifie que 149 requêtes ont été envoyées à notre serveur. Le "User-agent" des requêtes est curl, qui, comme mentionné précédemment, est une commande bash permettant de faire des requêtes sur des serveurs web.

L'attaque a duré environ deux minutes, indiquant qu'un script a été utilisé pour tester de nombreux URLs.

Pendant l'attaque, on observe une première requête sans "User-agent" qui n'interroge aucune URL. Il s'agit probablement d'un premier paquet d'observation ayant probablement détecté notre serveur. Suite à cette requête, la totalité de l'attaque est

lancée. Dans les requêtes, on constate que l'attaquant recherche une page de connexion administrateur, comme par exemple "admin.php". L'attaque explore de nombreux fichiers et dossiers différents, mais ne teste pas les fichiers décrits précédemment.

3.3.2 Attaque n°2

La deuxième attaque provient également des États-Unis et émane du domaine amazonaws.com, indiquant qu'il s'agit d'un serveur cloud hébergé par Amazon.

Cette attaque est orchestrée par deux serveurs amazonaws et se compose de trois vagues différentes.

La première vague est effectuée par le premier serveur et ne comprend que 8 requêtes. Elle débute le 06/12/2023 à 20 :07 et dure une minute, suggérant à nouveau l'utilisation d'un script. L'attaque cible le fichier "phpinfo.php" en essayant différentes URL.

La deuxième vague, toujours effectuée par le premier serveur, compte cette fois 114 requêtes. Elle débute le 07/12/2023 à 05 :55 et dure également une minute. Cette fois-ci, l'attaque ne se contente pas de chercher le fichier "phpinfo.php". Elle explore activement le fichier ".env", et on observe également des requêtes cherchant le fichier "/.aws/crédential".

La troisième vague est effectuée par le deuxième serveur et comprend 85 requêtes. Elle débute le 07/12/2023 à 22 :28 et dure deux minutes. Comme les deux vagues précédentes, cette vague cherche le fichier ".env" et le fichier "phpinfo.php". On observe également d'autres URL, telles qu'une recherche du fichier "admin.php".

Le champ "User-agent" de ces trois vagues est "Mozilla/5.0", ne fournissant aucune information sur la nature de l'outil utilisé pour l'attaque. Il n'y a pas de paquet de reconnaissance.

3.3.3 Attaque n°3

La dernière attaque provient du Royaume-Uni et est effectuée par deux serveurs, chacun réalisant deux vagues d'attaques identiques. Chaque vague d'attaque est exactement la même, avec les mêmes requêtes et les mêmes URL, mais à des heures différentes. Les serveurs sont des serveurs cloud hébergés par hostglobal.plus.

Sur le premier serveur, on observe une vague d'attaque le 08/12/2023 à 18 :36, suivie d'une seconde le 11/12/2023 à 11 :45.

Sur le deuxième serveur, une vague d'attaque a lieu le 08/12/2023 à 18 :05, suivie de la seconde le 11/12/2023 à 11 :35.

On remarque donc que les vagues d'attaques sont très proches dans le temps. Chacune des vagues dure une minute.

Cette attaque vise les fichiers "phpinfo.php" et ".env", ainsi que le dossier "/.aws/" avec notamment les fichiers "credentials" ou "config". Il n'y a pas de paquet de reconnaissance.

4 Conclusion

Pour conclure ce rapport, il est important de prendre en compte que les Honeypots web ne sont pas les plus utilisés. En général, on observe des Honeypots web avec des applications spécifiques telles que WordPress, destinées à tester les vulnérabilités.

De plus, la période de collecte des journaux est limitée à une semaine, ce qui peut ne pas être très représentatif de la réalité, notamment en ce qui concerne le nombre de requêtes par pays.

Enfin, il est intéressant de noter qu'en dehors des hackers, de nombreuses entités explorent l'ensemble d'Internet pour créer des cartes ou scanner les vulnérabilités présentes sur les serveurs.

Références

[1] Guy Bruneau. Systembc malware activity. https://isc.sans.edu/diary/rss/30138. Publication : 20-08-2023.

[2] Milena Dimitrova. Porte dérobée systembc tor – le nouvel outil préféré des opérateurs de ransomware. https://sensorstechforum.com/fr/systembc-tor-backdoor-ransomware-tool/. Publication : 17-12-2020.

Annexe A : Configuration

Fichier "config_http.json" :

```
{
"logs": "file,terminal,json",
"logs_location": "/home/ubuntu/log/http",
"syslog_address": "",
"syslog_facility": 0,
"postgres": "",
"sqlite_file":"",
"db_options": [],
"sniffer_filter": "",
"sniffer_interface": "",
"honeypots": {
    "http": {
    "port": 80,
    "ip": "0.0.0.0",
    "username": "http",
    "password": "anonymous",
    "log_file_name": "http.log",
    "max_bytes": 10000,
    "backup_count": 100,
    "options":["capture_commands"]
    }
}
}
```

Fichier "config_https.json" :

```
{
"logs": "file,terminal,json",
"logs_location": "/home/ubuntu/log/https/",
"syslog_address": "",
"syslog_facility": 0,
"postgres": "",
"sqlite_file":"",
"db_options": [],
"sniffer_filter": "",
"sniffer_interface": "",
"honeypots": {
    "https": {
    "port": 443,
    "ip": "0.0.0.0",
    "username": "rie",
    "password": "Pa$$word",
    "log_file_name": "https.log",
    "max_bytes": 100000,
    "backup_count": 100,
    "options":["capture_commands"]
    }
}
}
```

Annexe B : Configuration

Script python employé pour exporter les logs en csv :

```
import json
import csv
import os

logs = []
path = "./log/http"
files = os.listdir(path)

for file in files:
    with open(path+"/"+file, 'r') as content:
        lines = content.readlines()
        for line in lines:
            log = json.loads(line)
            if log["action"] == "connection":
                logs.append(log)

path = "./log/https"
files = os.listdir(path)
for file in files:
    with open(path+"/"+file, 'r') as content:
        lines = content.readlines()
        for line in lines:
            log = json.loads(line)
            if "action" in log:
                if log["action"] == "connection":
                    logs.append(log)

csvfile = csv.writer(open("http.csv", "w"))
csvfile.writerow(["src_ip", "src_port", "User-Agent", "uri", "timestamp"])
for log in logs:
    if 'User-Agent' in log['data']:
        csvfile.writerow([log['src_ip'], log['src_port'], log['data']['User-Agent'], log['data']['uri'], log['timestamp']])
```

```
33      else:
34          csvfile.writerow([log['src_ip'], log
        ['src_port'], "", log['data']['uri'], log
        ['timestamp']])
```