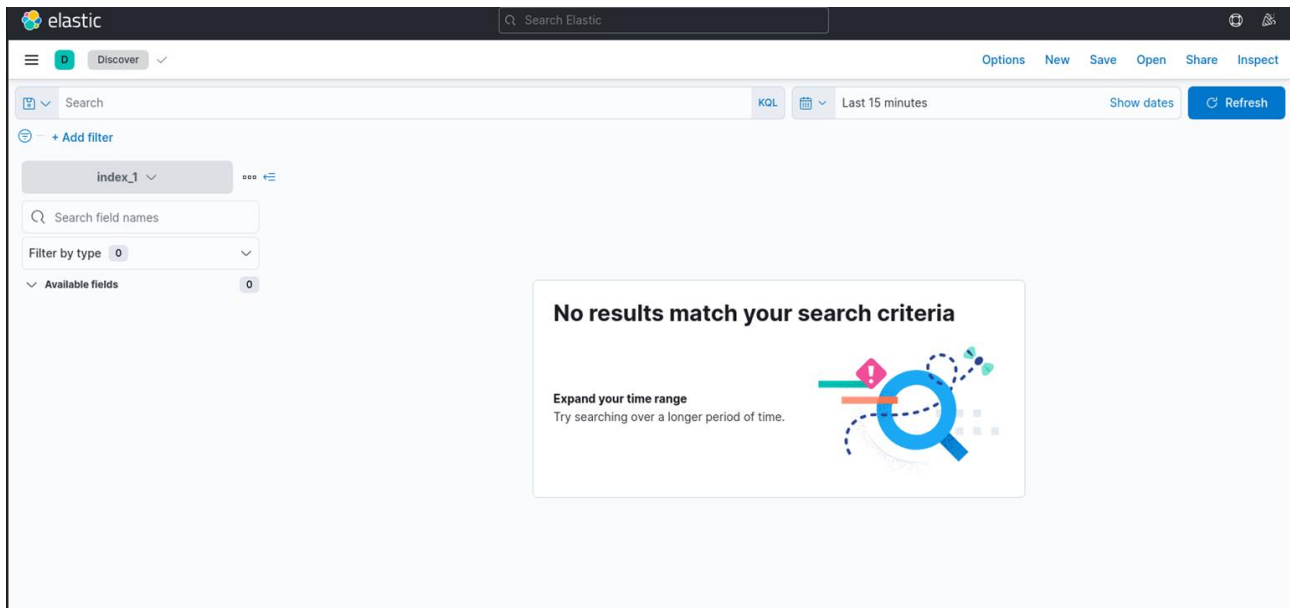


corp_station

Writeup for SearchloC








Nous recevons un lien (<http://qualif.hackerlab.bj:5601/app/home#/>) ELK.

Une fois sur Elasticsearch j'accède à l'onglet **Home** puis **Discover** pour afficher les logs disponibles.



Le défi mentionnait que les traces de compromission dans le système étaient présentes depuis un bout de temps. Donc je modifie la date pour afficher les logs couvrant un intervalle d'un an. Je me retrouve alors avec 500 logs, 100 dans chaque index allant de 0 à 4. Pour commencer je constate que 4 des filtres disponibles sont encodés en base64.

Popular

-  _index
-  _score
-  _type
-  bm9tbnFtZQ
-  cGF5bG9hZA
-  REFUQQ==
-  ZGF0YQ

Une fois décodé j'obtiens **"nomname data payload DATA"**.

Le nom du défi SearchloC nous demandais selon moi de chercher loC dans les indexes. Après plusieurs essais je ne trouvais rien. Et là je me suis dit que loC était probablement encodée en base64 comme les filtres décodés tout à l'heure. loC en base 64 nous donnait SW9DCgoKCg. J'ai ensuite commencé à chercher cette valeur dans les logs sans succès. En faisant quelques recherches sur comment rechercher des expressions dans les indexes dans Elasticsearch je tombe sur ce site <https://book.hacktricks.xyz/network-services-pentesting/9200-pentesting-elasticsearch> qui détaillais très bien le processus.

Après plusieurs essais je suis tombé sur cette valeur

"SW9D" :

"SexCMjAyNHtNYlNzdnlkTm1qY2h5RWWhMSHlOeHNZZlFucTlBV01QZGZ1eVAyOGZHUU40OFI9"

```
kali@kali: ~/Téléchargements


1890 curl -X GET "http://qualif.hackerlab.bj:9200/index_6/_search?pretty=true&size=1000" | grep -i HLB
1891 curl -X GET "http://qualif.hackerlab.bj:9200/index_7/_search?pretty=true&size=1000" | grep -i HLB
1892 curl -X GET "http://qualif.hackerlab.bj:9200/index_8/_search?pretty=true&size=1000" | grep -i HLB
1893 curl -X GET "http://qualif.hackerlab.bj:9200/index_*/_search?pretty=true&size=1000" | grep -i HLB
1894 curl -X GET "http://qualif.hackerlab.bj:9200/index_*/_search?pretty=true&size=1000" | grep -i flag
1895 curl -X GET "http://qualif.hackerlab.bj:9200/*/_search?pretty=true&size=1000" | grep -i
1896 curl -X GET "http://qualif.hackerlab.bj:9200/*/_search?pretty=true&size=1000" | grep -i hlb
1897 curl -X GET "http://qualif.hackerlab.bj:9200/*/_search?pretty=true&size=1000" | grep -i HLB2024
1898 curl -X GET "http://qualif.hackerlab.bj:9200/index_*/_search?pretty=true&size=1000" | grep -i IoC
1899 curl -X GET "http://qualif.hackerlab.bj:9200/*/_search?pretty=true&size=1000" | grep -i IoC
1900 curl -X GET "http://qualif.hackerlab.bj:9200/*/_search?pretty=true&size=1000" | grep ioc
1901 curl -X GET "http://qualif.hackerlab.bj:9200/index*/_search?pretty=true&size=1000" | grep ioc
1902 curl -X GET "http://qualif.hackerlab.bj:9200/index*/_search?pretty=true&size=1000" | grep -i IoC
1903 curl -X GET "http://qualif.hackerlab.bj:9200/index_2/_search?pretty=true&size=1000" | grep -i IoC
1904 curl -X GET "http://qualif.hackerlab.bj:9200/index_5/_search?pretty=true&size=1000" | grep -i IoC
1905 curl -X GET "http://qualif.hackerlab.bj:9200/index_6/_search?pretty=true&size=1000" | grep -i IoC
1906 curl -X GET "http://qualif.hackerlab.bj:9200/index_0/_search?pretty=true&size=1000" | grep -i IoC
1907 curl -X GET "http://qualif.hackerlab.bj:9200/index_1/_search?pretty=true&size=1000" | grep -i IoC
1908 curl -X GET "http://qualif.hackerlab.bj:9200/index_3/_search?pretty=true&size=1000" | grep -i IoC
1909 curl -X GET "http://qualif.hackerlab.bj:9200/index_4/_search?pretty=true&size=1000" | grep -i IoC
1910 curl -X GET "http://qualif.hackerlab.bj:9200/index_4/_search?pretty=true&size=10000000" | grep -i IoC
1911 curl -X GET "http://qualif.hackerlab.bj:9200/index_4/_search?pretty=true&size=100" | grep -i IoC
1912 curl -X GET "http://qualif.hackerlab.bj:9200/index_4/" | grep -i IoC
1913 curl -X GET "http://qualif.hackerlab.bj:9200/index_4/_search?pretty=true&size=100" | grep -i SW9D
1914 curl -X GET "http://qualif.hackerlab.bj:9200/index_*/_search?pretty=true&size=100" | grep -i SW9D
1915 curl -X GET "http://qualif.hackerlab.bj:9200/index_*/_search?pretty=true" | grep -i SW9D
1916 curl -X GET "http://qualif.hackerlab.bj:9200/index_*/_search?pretty=true&size=1000" | grep -i SW9D
1932 strings download.dat | grep -i HLB
1933 strings -a download.dat | grep -i HLB
1938 strings * | grep -i hlb
1939 strings * | grep -i hlb2024
1940 strings * | grep -i flag
2002 curl http://qualif.hackerlab.bj:9200/index_*/_search?pretty=true&size=1000 | grep -i SW9DCgoKCg
2003 curl -X GET http://qualif.hackerlab.bj:9200/index_*/_search?pretty=true&size=1000 | grep -i SW9DCgoKCg
2004 curl -X GET http://qualif.hackerlab.bj:9200/index_*/_search?pretty=true&size=1000 | grep -i SW9

--(kali@kali)-[~/Téléchargements]
$ curl -X GET "http://qualif.hackerlab.bj:9200/index_*/_search?pretty=true&size=1000" | grep -i SW9D
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total   Spent    Left   Speed

  5 232k    5 13380    0    27114    0  0:00:08 --:--:--  0:00:08 27085      "SW9D" : "SExCMjAyNHhNYlNzdnlkTm1qY2h5RWhMSH10eHNZZlFucTlBV01QZGZ1eVAY0GZHUU400Fl9"
100 232k  100 232k    0    202k    0  0:00:01 0:00:01 --:--:-- 202k
```

Une fois décodé on obtient

IoCHLB2024{MbSsvydNmjchyEhLHyNxsYfQnq9AWMPdfuyP28fGQN48Y}. Et là nous sommes proches d'avoir quelque chose. L'intérieur était encodé. Je l'envoie sur dcode, j'ai utilisé l'option identifier hash pour voir que c'était en base58 on le décode et on obtient



Rechercher un outil

★ RECHERCHE SUR dCODE PAR MOTS-CLÉS :

★ PARCOURIR LA LISTE COMPLÈTE DES OUTILS


Résultats

ESCL8ST3RS_FOR_SEARCH_IOC_HLB2339

Base 58 - dCode

Catégorie(s) : Arithmétique, Codage de Caractères

Partager



dCode et plus

dCode est gratuit et ses outils sont une aide précieuse dans les jeux, les maths, les énigmes, les géocaches, et les problèmes à résoudre au quotidien !
Une suggestion ? un problème ? une idée ? [Écrire à dCode](#) !

BASE 58

Mathématiques > Arithmétique > Base 58

DÉCHIFFREMENT DE LA BASE 58

★ ALPHABET 123456789ABC...XYZabc...xyz (Bitcoin BTC) ▼

★ MESSAGE CHIFFRÉ PAR BASE 58 (?)

MbSsvydNmjchyEhLHyNxsYfQnq9ANMPdfuyP28fGQN48Y

★ FORMAT DES RÉSULTATS

☒ CHAÎNE DE CARACTÈRES IMPRIMABLES (ASCII/UNICODE)
☐ HEXADÉCIMAL 00-FF
☐ DÉCIMAL 0-127-255
☐ OCTAL 000-177-377
☐ BINAIRE 00000000-11111111
☐ NOMBRE ENTIER
☐ FICHIER À TÉLÉCHARGER

► DÉCHIFFRER

Voir aussi : [Code Base64](#) — [Conversion en Base N](#)

CHIFFREMENT AVEC BASE 58

★ ALPHABET 123456789ABC...XYZabc...xyz (Bitcoin BTC) ▼

A PARTIR D'UN MESSAGE TEXTE (ASCII)

★ MESSAGE CLAIR À CHIFFRER AVEC BASE 58 (?)

dCode Base 58

► CHIFFRER

A PARTIR D'UN NOMBRE

Menu

- ★ Déchiffrement de la Base 58
- ★ Chiffrement avec Base 58
- ★ Qu'est ce que la Base58 ? (Définition)
- ★ Comment encoder avec la Base58 ? (Principe de chiffrement)
- ★ Comment décoder la Base58 ? (Principe de déchiffrement)
- ★ Comment reconnaître le chiffre Base 58 ?
- ★ Quelles sont les variantes de la Base 58 ?

Pages similaires

- ★ Conversion en Base N
- ★ Code Base64
- ★ Codage Base62
- ★ Code Binaire
- ★ Codage Base45
- ★ Code Base91
- ★ Code Barres EAN13
- ★ LISTE DES OUTILS DCODE

Faire un don

- ★ Paypal
- ★ Amazon
- ★ Autre

flag : HLB2024{ESCL8ST3RS_FOR_SEARCH_IOC_HLB2339}