


Sprawozdanie 1 Ochrona Systemów Operacyjnych

Jonatan Kasperczak
24.03.2022
Cyberbezpieczeństwo 2022

Dziennik 3

 CYBERBEZPIECZEŃSTWO 2022	ĆWICZENIE 1	15
--	-------------	----

Zadania do realizacji na dowolnym systemie linux:

- Dla dostarczonego dziennika 3 (serwer pocztowy) należy ustalić:

1. Liczbę nieudanych prób logowania
2. Listę adresów IP z których nastąpiły błędne logowania
3. Kraje, z których było najwięcej błędnych logowań - Top 10
4. Nazwy użytkowników lokalnych: user-1 .. user-101
5. Listę 20 użytkowników lokalnych na których następuje wyraźnie najwięcej prób ataku. Jeśli są wnioski ?
6. Listę prób wykorzystania serwera jako "open relay" (opcjonalnie)

Rozwiązanie wykonane w **bash**

Aby uruchomić skrypt na systemie linux należy wpisać w terminal: `./dziennik_3_script.sh`

Zadanie 1

`echo " Nieudane Logowania: "`

`grep "login authenticator failed" final.log | wc -l`

Za pomocą polecenia **grep** wyszukuje frazy *"login authenticator failed"* a następnie **wc -l** liczę ile tych nieudanych logowań wystąpiło

```
Nieudane Logowania:
234
```

Zadanie 2

`echo " Lista adresów na które wykonano nieudane logowania "`

`grep "login authenticator failed" final.log | awk '$8 ~ /^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+/' { print substr($8, 2, length($8)-3) } | sort | uniq > most_freq.txt`

```
grep "login authenticator failed" final.log | awk '$9 ~ /^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+:/ { print
substr($9, 2, length($9)-3) }' | sort | uniq >> most_freq.txt
cat most_freq.txt
wc -l most_freq.txt
```

awk szuka po regex w kolumnie 8 i 9 sformatowanych odpowiednio adresów IP, poleceniem **uniq** usuwane są powtórzenia, i te adresy na końcu zapisywane są w pliku, plik odczytywany, i liczone linie, bo w każdej linii jest jeden adres, by podać ile było adresów IP z których były podejmowane próby logowań

Zadanie 3

```
grep "login authenticator failed" final.log | awk '$8 ~ /^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+:/ { print
substr($8, 2, length($8)-3) }' | sort > ips.txt
grep "login authenticator failed" final.log | awk '$9 ~ /^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+:/ { print
substr($9, 2, length($9)-3) }' | sort >> ips.txt
```

```
echo "KRAJE"
while read p; do
  geoiplookup $p | awk '{ $1=$2=$3=$4=""; print $0 }' | sed -r 's/[ ]+/_/g' | cut -c2- >>
after_iplookup.txt
done <ips.txt
```

```
awk '{count[$1]++} END{for (ele in count) printf "%s\t%s\n", count[ele], ele}' after_iplookup.txt
| sort -rn | sed -r 's/[ ]+/_/g' | head -10
```

Zapisane adresy w pliku są przez pętlę poddawane **geoiplookup** który sprawdza lokalizację adresu, zapisuje do pliku, a następnie **awk** z pętlą **for** liczy ile jest tych samych linii, po policzeniu sortowane są malejąco, poleceniem **sed** zamieniam znak podkreślenia spowrotem na spacje i poleceniem **head -10** wypisuje tylko 10 pierwszych linii

```
KRAJE
105    Russian Federation
38     United States
24     United Kingdom
19     Netherlands
11     Vietnam
9      Singapore
9      Address not found
4      Egypt
2      Thailand
2      Germany
```

Zadanie 4

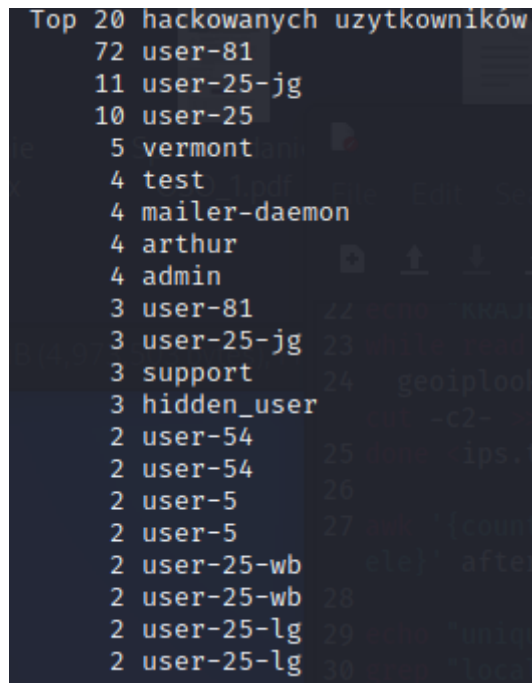
```
echo "unique users"
grep "localuser" final.log | awk '{ print $5 }' | cut -d '@' -f 1 | sort | uniq
```

Wyszukuje użytkowników lokalnych, w kolumnie 5 poleceniem **cut** dzieli tekst na dwie części które oddziela znak "@" a następnie zostawiam tylko pierwszy element, sortuje i usuwa duplikaty

Zadanie 5

```
echo " Top 20 hackowanych uzytkowników "  
grep "login authenticator failed" final.log | awk ' $13 ~ /(set_id=)/ { print substr ($13, 9,  
length($13) ) }' | sort > top20user.txt  
grep "login authenticator failed" final.log | awk ' $14 ~ /(set_id=)/ { print substr ($14, 9,  
length($14) ) }' | sort >> top20user.txt  
sort top20user.txt | cut -d '@' -f 1 | sed -r 's/[()]+//g' > transformed.txt  
uniq -c transformed.txt | sort -rn | head -20
```

Wyszukuje nieudane logowania, znajduje w kolumnie 13 lub 14 nazwę użytkownika na którego padła próba logowania, zapisuje do pliku. W pliku przetwarza nazwę użytkownika do odpowiedniego formatu, dzieli na dwie części oddzielone znakiem "@" i poleceniem **sed** usuwam ostatni znak ")" jeżeli występuje. Następnie liczy i wypisuje 20 najczęściej występujących



```
Top 20 hackowanych uzytkowników  
72 user-81  
11 user-25-jg  
10 user-25  
5 vermont  
4 test  
4 mailer-daemon  
4 arthur  
4 admin  
3 user-81  
3 user-25-jg  
3 support  
3 hidden_user  
2 user-54  
2 user-54  
2 user-5  
2 user-5  
2 user-25-wb  
2 user-25-wb  
2 user-25-lg  
2 user-25-lg
```

Wnioski

Użytkownik 81 musi posiadać dobre zabezpieczenia, że potrzeba było tyle prób włamania się, albo ma dobre zabezpieczenia przed włamaniem. Może też być administratorem, i jego konto jest ściśle pożądane przez atakującego. Wnioskuje to po tym, że był średnio 7 razy częściej atakowany niż drugie konto po nim

W skrypcie występuje taki fragment

```

if [[ -e ips.txt ]]
then
    rm ips.txt
fi
if [[ -e after_iplookup.txt ]]
then
    rm after_iplookup.txt
fi

```

Jest on odpowiedzialny za to by usunąć pliki, aby dane które będą do nich zapisywać nie dopisały się i nie powtarzały już zapisane w nim wcześniej jakiegokolwiek dane.

Skrypt:

```
#!/bin/bash
```

```
echo " Nieudane Logowania: "
```

```
grep "login authenticator failed" final.log | wc -l
```

```
echo " Lista adresów na które wykonano nieudane logowania "
```

```
grep "login authenticator failed" final.log | awk '$8 ~ /[0-9]+.[0-9]+.[0-9]+.[0-9]+:/ { print substr($8, 2, length($8)-3) }' | sort | uniq > most_freq.txt
```

```
grep "login authenticator failed" final.log | awk '$9 ~ /[0-9]+.[0-9]+.[0-9]+.[0-9]+:/ { print substr($9, 2, length($9)-3) }' | sort | uniq >> most_freq.txt
```

```
cat most_freq.txt
```

```
wc -l most_freq.txt
```

```
if [[ -e ips.txt ]]
```

```
then
```

```
    rm ips.txt
```

```
fi
```

```
if [[ -e after_iplookup.txt ]]
```

```
then
```

```
    rm after_iplookup.txt
```

```
fi
```

```
grep "login authenticator failed" final.log | awk '$8 ~ /[0-9]+.[0-9]+.[0-9]+.[0-9]+:/ { print substr($8, 2, length($8)-3) }' | sort > ips.txt
```

```
grep "login authenticator failed" final.log | awk '$9 ~ /[0-9]+.[0-9]+.[0-9]+.[0-9]+:/ { print substr($9, 2, length($9)-3) }' | sort >> ips.txt
```

```
echo "KRAJE"
```

```
while read p; do
```

```
    geoiplookup $p | awk '{$1=$2=$3=$4=""; print $0}' | sed -r 's/[ ]+/_/g' | cut -c2- >>
```

```
after_iplookup.txt
```

```
done <ips.txt
```

```
awk '{count[$1]++} END{for (ele in count) printf "%s\t%s\n", count[ele], ele}' after_iplookup.txt | sort -rn | sed -r 's/[ ]+/_/g' | head -10
```

```
echo "unique users"
```

```
grep "localuser" final.log | awk '{ print $5}' | cut -d '@' -f 1 | sort | uniq
```

```
echo " Top 20 hackowanych uzytkowników "  
grep "login authenticator failed" final.log | awk ' $13 ~ /(set_id=)/ { print substr ($13, 9,  
length($13) ) }' | sort > top20user.txt  
grep "login authenticator failed" final.log | awk ' $14 ~ /(set_id=)/ { print substr ($14, 9,  
length($14) ) }' | sort >> top20user.txt  
sort top20user.txt | cut -d '@' -f 1 | sed -r 's/[+]/ /g' > transformed.txt  
uniq -c transformed.txt | sort -rn | head -20
```

Wynik skryptu:

Nieudane Logowania:

234

Lista adresów na które wykonano nieudane logowania

10.0.0.142
103.57.195.147
109.120.250.112
110.78.158.52
113.161.59.18
113.172.241.254
123.21.16.79
123.23.242.241
123.24.73.237
128.106.1.6
14.161.19.175
14.161.26.155
14.164.252.186
14.169.102.200
14.169.196.18
143.255.153.196
156.213.104.212
156.220.13.202
170.246.152.24
181.129.167.82
185.144.28.111
185.144.28.130
185.144.28.241
185.144.29.111
185.144.29.178
185.144.29.189
185.144.29.219
185.144.29.30
185.144.30.39
185.211.245.195
185.222.209.201
185.222.209.202
185.222.209.78
185.231.245.40
185.231.245.41
185.231.245.42

185.231.245.43
185.231.245.44
185.231.245.45
185.231.245.46
185.231.245.48
185.231.245.49
185.231.245.50
193.233.74.11
193.233.74.12
193.233.74.17
197.44.171.25
197.53.26.46
202.137.155.157
37.120.146.84
45.119.80.41
50.238.90.22
62.50.131.54
64.235.38.22
80.82.65.187
84.246.148.214
88.205.135.211
91.212.150.81
92.246.76.92
93.157.63.30
93.157.63.6
93.157.63.7
93.157.63.8
93.157.63.9
94.102.49.198
142.11.199.241
178.127.40.101
181.13.157.250
183.88.225.91
187.189.222.97
80.85.153.204
80.85.153.205
80.85.153.206
80.85.153.207
80.85.153.209
80.85.153.211
92.61.148.10

77 most_freq.txt

KRAJE

105	Russian Federation
38	United States
24	United Kingdom
19	Netherlands
11	Vietnam
9	Singapore
9	Address not found
4	Egypt
2	Thailand

2 Germany

unique users

user-10

user-11

user-12

user-13

user-14

user-16

user-17

user-18

user-19

user-20

user-23

user-24

user-25

user-25-jg

user-25-lg

user-25-wb

user-3

user-30

user-32

user-34

user-35

user-4

user-40

user-41

user-43

user-44

user-45

user-47

user-48

user-5

user-50

user-51

user-53

user-54

user-55

user-56

user-57

user-58

user-59

user-6

user-60

user-61

user-68

user-69

user-7

user-70

user-71

user-72

user-73

user-74

user-75
user-76
user-77
user-78
user-79
user-8
user-81
user-82
user-83
user-84
user-86
user-87
user-88
user-89
user-9
user-90
user-91
user-93
user-94
user-95
user-96
user-97
user-99

Top 20 zaatakowanych użytkowników

72 user-81
11 user-25-jg
10 user-25
5 vermont
4 test
4 mailer-daemon
4 arthur
4 admin
3 user-81
3 user-25-jg
3 support
3 hidden_user
2 user-54
2 user-54
2 user-5
2 user-5
2 user-25-wb
2 user-25-wb
2 user-25-lg
2 user-25-lg