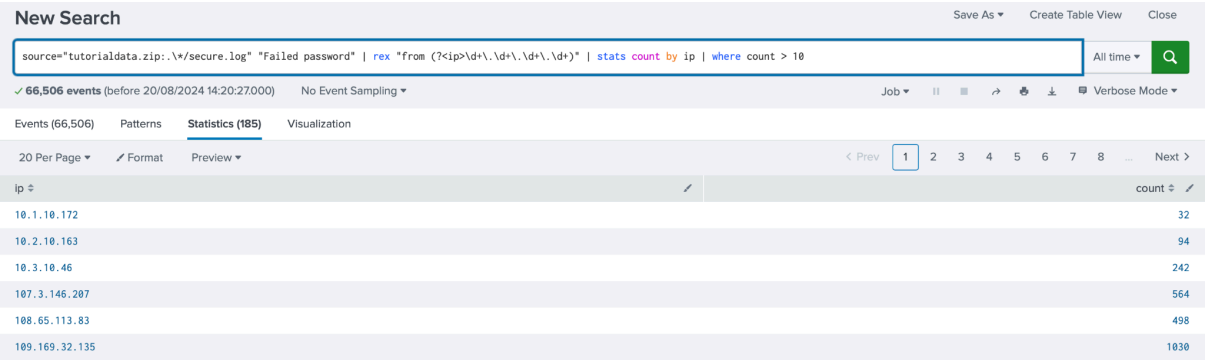
Part I. Can you find people trying to break into the servers?

Q1. How many hackers are trying to get access to our servers? And how many attempts are there? Explain/define how you count distinct hackers.

<u>Ans</u> 185 people and 66,506 attempts, counted from events where the password failed more than 10 times.

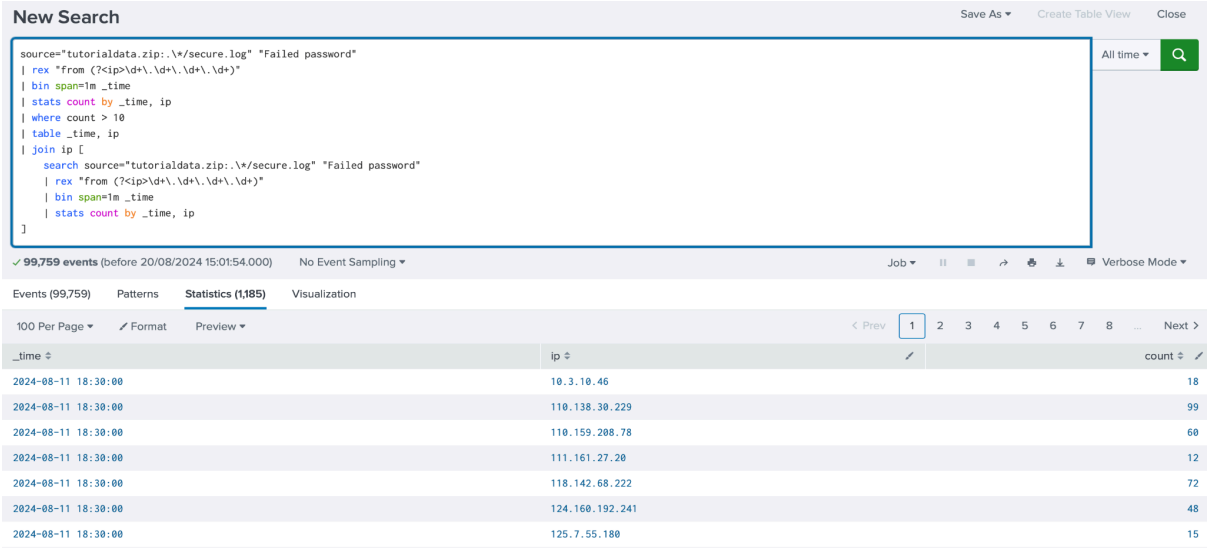
```
source="tutorialdata.zip:.\*/secure.log" "Failed password"
| rex "from (?<ip>\d+\.\d+\.\d+\.\d+)"
| stats count by ip
| where count > 10
```



Q2. What time do hackers appear to try to hack our servers?

<u>Ans</u> 18:30

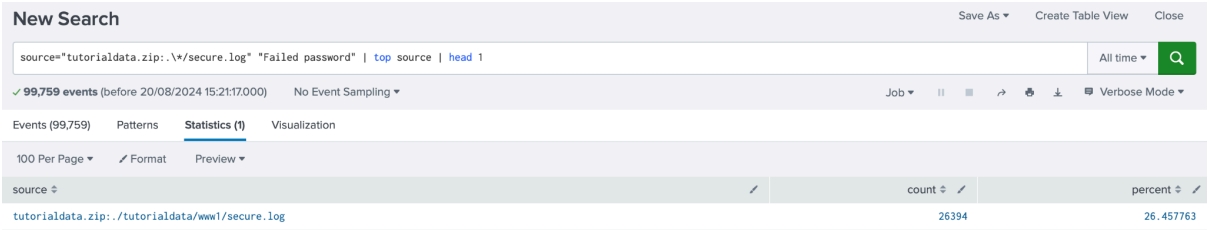
```
source="tutorialdata.zip:.\*/secure.log" "Failed password"
| rex "from (?<ip>\d+\.\d+\.\d+\.\d+)"
| bin span=1m _time
| stats count by _time, ip
| where count > 10
| table _time, ip | join ip [
    search source="tutorialdata.zip:.\*/secure.log" "Failed password"
    | rex "from (?<ip>\d+\.\d+\.\d+\.\d+)"
    | bin span=1m _time
    | stats count by _time, ip ]
```



Q3. Which server (mailsv, www1, www2, www3) had the most attempts?

Ans www1

```
source="tutorialdata.zip:.\*/secure.log" "Failed password" | top source | head 1
```

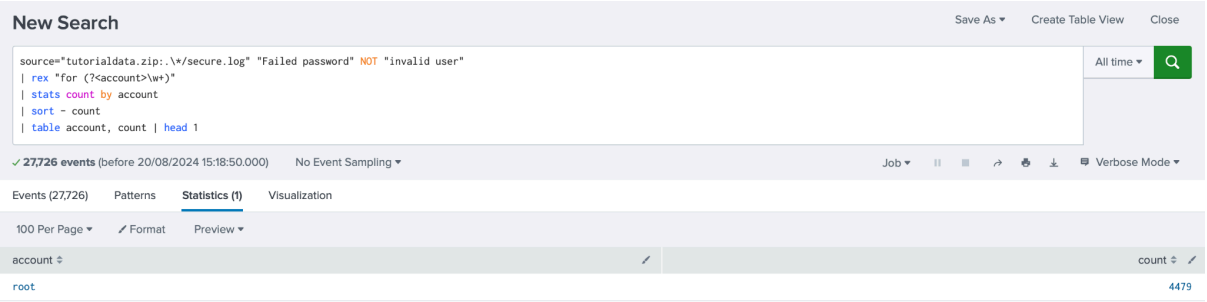


Q4. What is the most popular account that hackers use to try to break in?

Ans root

```
source="tutorialdata.zip:.\*/secure.log" "Failed password" NOT "invalid user"
| rex "for (?<account>\w+)"
| stats count by account
```

```
| sort - count
| table account, count
| head 1
```

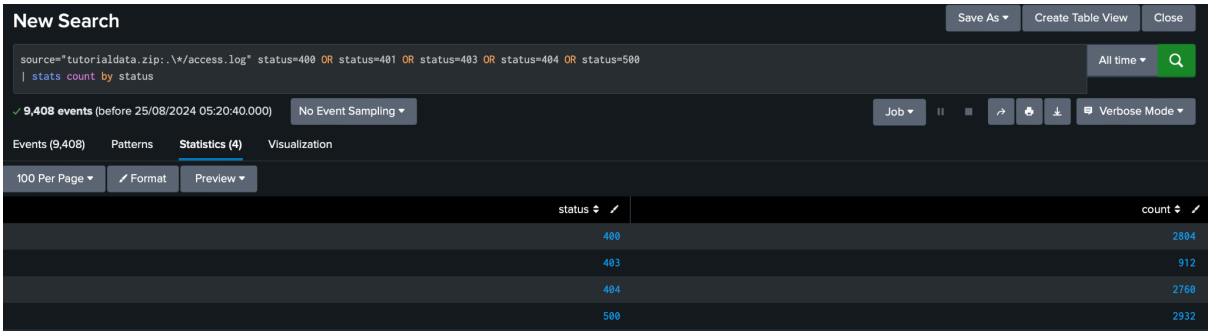


Part II. Sensitive Files on Web Servers

Q5. Can you find attempts to get access to sensitive information from our web servers? How many attempts were there?

<u>Ans</u> `Yes`, after reviewing the web server's access logs, I found `9,408` attempts to access sensitive information. These were flagged by filtering for HTTP status codes like 401 (Unauthorized), 403 (Forbidden), 404 (Not Found), and 500 (Internal Server Error), which often indicate unauthorized access attempts or suspicious activity.

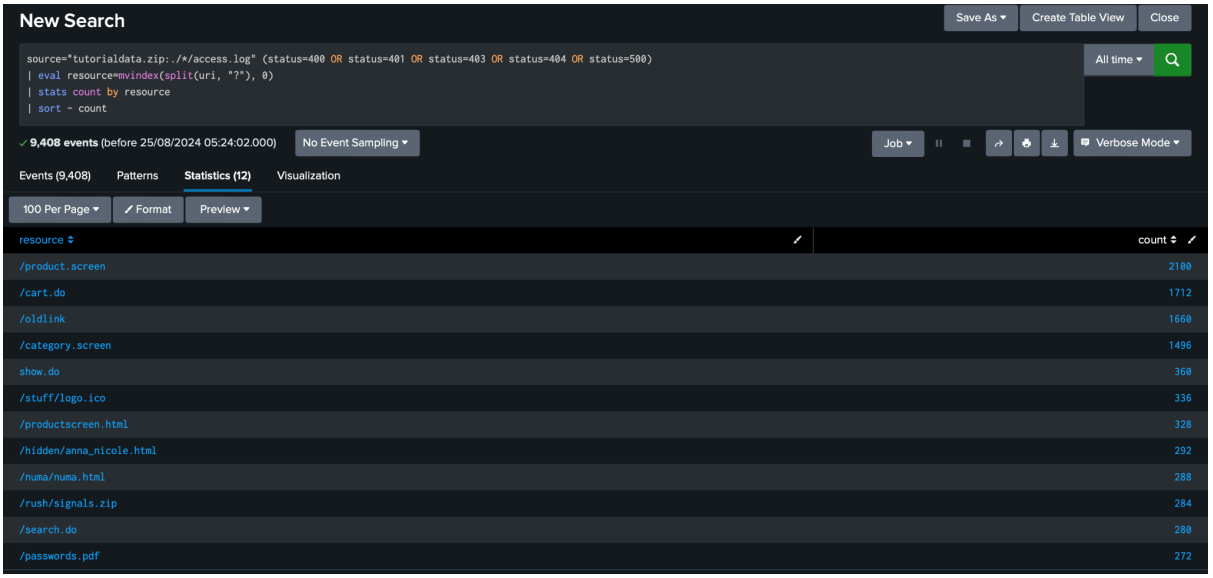
```
source="tutorialdata.zip:.\*/access.log" status=400 OR status=401 OR status=403
OR status=404 OR status=500
| stats count by status
```



Q6. What resource/file are hackers looking for?

Ans The files in the image below are the ones the hacker is likely looking for, based on the status codes.

```
source="tutorialdata.zip:./*/access.log" (status=400 OR status=401 OR status=403 OR status=404 OR status=500)
| eval resource=mvindex(split(uri, "?"), 0)
| stats count by resource
| sort - count
```

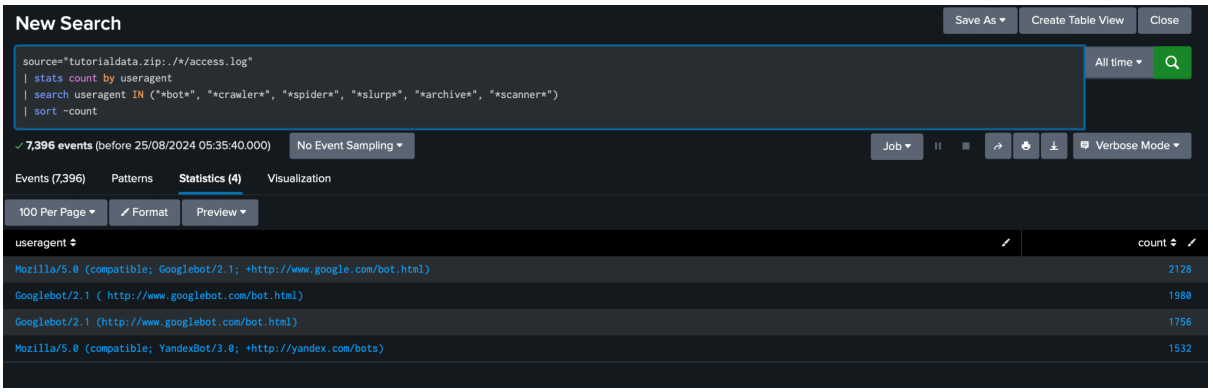


Part III. Are there bots crawling our websites?

Q7. Can you find any bots crawling our websites?

Ans Yes, I found `Mozilla/5.0` and `Googlebot/2.1` after filtering the user agents for terms like `bot`, `crawler`, `spider`, `slurp`, `archive`, and `scanner`.

```
source="tutorialdata.zip:./*/access.log"
| stats count by useragent
| search useragent IN ("*bot*", "*crawler*", "*spider*", "*slurp*", "*archive*", "*scanner*")
| sort -count
```



Q8. What are they doing on the site? (Hint: Look for User-Agent in the web access.logs.)

<u>Ans</u> The bots, specifically Googlebot/2.1 and YandexBot/3.0, are crawling various parts of the site to index its content for search engines like Google and Yandex. They're accessing pages like /cart.do, /product.screen, and /category.screen, as well as some older links and specific files. This activity helps ensure that these pages and files are included in search results, making them easier for users to find online.

```
source="tutorialdata.zip:./*/access.log"
| search useragent IN ("*bot*", "*crawler*", "*spider*", "*slurp*", "*archive*", "*scanner*")
| eval resource=mvindex(split(uri, "?"), 0)
| stats count by resource, useragent
| sort - count
```