Написал вот такой скрипт для того, чтобы логи отправлялись на сервер.

```
import requests
import pandas

logs = pandas.read_csv('/var/log/super_mega_critical.csv')
data = {}

for i in logs.columns:
        data[i] = [str(i) for i in logs[i]]

status = requests.post(url='https://siem.yandex.ru/input', json=data,
headers={'Content-Type': 'application/json'})

if status.status_code == 200:
        print('Done')
else:
        print('Something wrong!')
```

Dockerfile будет выглядеть вот так:

```
FROM python:latest

RUN pip3 install requests pandas

RUN useradd log

WORKDIR /app

COPY send_logs.py .

USER log
CMD ["python3", "send_logs.py"]
```

Нужно собрать этот образ для того, чтобы дальше его использовать на машине.

```
docker build -t cron_job .
```

Важно, чтобы этот образ не был удален из локального реестра образов докера на машине. Или можно его использовать из докер хаба, предварительно скачав оттуда. Сам запуск контейнера будет выполнен с помощью такой команды:

```
docker run --rm -it -v
/var/log/super_mega_critical.csvtest.csv:/app/super_mega_critical.csv:ro
cron_job
```

Чтобы отправлять каждый час логи, я воспользовался cron задачами. Сделал в формате, что у нас есть уже образ контейнера на хосте, и этот образ мы будем запускать каждый час. Соотвественно, я "не тянул" cron в контейнер, а оставил его на хостовой машине.
Сама строчка, которая будет запускать задачу в cron выглядит так:

```
0 * * * * docker run -it --rm -d -v
/var/log/super_mega_critical.csv:/app/super_mega_critical.csv:ro
cron_job
```

Неожиданная находка

Тут можно сразу заметить, что PID=1920 это реверс шелл по адресу 130.193.55.241 на порт 443. Кстати, ip-адрес принадлежит Yandex Cloud по данным из whois.
Еще бы я присмотрелся к 721 процессу. В процессе обновления пакетов можно занести себе какую-нибудь малварь, в формате атак на цепочку поставок.