

```

CPU[|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%] Tasks: 64, 146 thr; 1 running
Mem[|||||||||||||||||||||||||||||||||||||||||||||||||||||||||1.01G/1.90G] Load average: 1.05 1.03 1.00
Swp[|||||||||||||||||||||||||||||||||||||||||||||||||||||||||0K/0K] Uptime: 188 days(!), 16:42:34

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU%  MEM%   TIME+  Command
 19480 999        20   0   298M   261M    4  S  100.  13.4   4h13:29 1vLA4Q
 19737 999        20   0   298M   261M    4  R  100.  13.4   4h13:26 1vLA4Q
 30146 root        20   0 2625M  369M 11420  S  0.7  19.0  51:55.80 java -Dpg.host=db -jar app.jar
 25982 root        20   0 32316  4744  3760  R  0.7  0.2   0:00.05 htop
    1 root        20   0  220M  6364  3740  S  0.7  0.3  17:43.57 /sbin/init
 30163 root        20   0 2625M  369M 11420  S  0.7  19.0  24:23.63 java -Dpg.host=db -jar app.jar
 23630 root        20   0  372M  9608  4840  S  0.0  0.5   6:21.76 php-fpm: master process (/etc/php/7.4/fpm/php-fpm.conf)
 16419 root        20   0  940M 52108  3096  S  0.0  2.6   3h26:06 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
 17662 root        20   0  940M 52108  3096  S  0.0  2.6   7:41.74 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
 16278 root        20   0  782M 22448  2172  S  0.0  1.1  17:43.82 /usr/bin/containerd
 23645 www-data    20   0  605M 12100  4560  S  0.0  0.6   0:00.41 php-fpm: pool www
 23644 www-data    20   0  605M 11736  4192  S  0.0  0.6   0:00.42 php-fpm: pool www
 29947 root        20   0  468M  2808  1412  S  0.0  0.1   0:03.34 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29952 root        20   0  468M  2808  1412  S  0.0  0.1   0:00.00 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29951 root        20   0  468M  2808  1412  S  0.0  0.1   0:00.00 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29950 root        20   0  468M  2808  1412  S  0.0  0.1   0:00.00 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29949 root        20   0  468M  2808  1412  S  0.0  0.1   0:00.00 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29948 root        20   0  468M  2808  1412  S  0.0  0.1   0:03.34 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29888 root        20   0  540M  2860  1224  S  0.0  0.1   0:03.74 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5001 -container-ip 172

```

Небольшой пример из практики на тему того, как ломают web сервера на Linux. А то некоторые думают, что под линуксом вирусов нет, или что Linux настолько безопасен, что его даже защищать не надо. В общем, показательная история из жизни. Ничего особенного, просто пример, которых наверняка много у любого админа, который работает с разработчиками.

Теоретический курс по основам **сетевых технологий**. Позволит системным администраторам упорядочить и восполнить пробелы в знаниях. Цена очень доступная, есть бесплатный доступ. Все подробности по serveradmin.ru. Можно пройти тест на знание сетей, бесплатно и без

регистрации.

Пересылает мне утром владелец учетной записи в Hetzner информацию о том, что пришло уведомление, что у нас на одном из серверов подозрительная активность и его вырубят, если мы ничего не сделаем. Это тестовый сервер разработчика. Иду по ssh на сервер и вижу.


```

CPU[|||||||||||||||||||||||||||||||||||||||||100.0%] Tasks: 64, 146 thr; 1 running
Mem[|||||||||||||||||||||||||||||||||||||1.01G/1.90G] Load average: 1.05 1.03 1.00
Swp[|||||||||||||||||||||||||||||||||0K/0K] Uptime: 188 days(!), 16:42:34

  PID USER      PRI  NI  VIRT   RES    SHR  S  CPU%  MEM%   TIME+  Command
 19480 999        20   0   298M   261M    4 S 100.  13.4   4h13:29 lvLA4Q
 19737 999        20   0   298M   261M    4 R 100.  13.4   4h13:26 lvLA4Q
 30146 root        20   0 2625M   369M 11420 S  0.7  19.0 51:55.60 java -Dpg.host=db -jar app.jar
 25982 root        20   0 32316   4744  3760 R  0.7  0.2  0:00.05 htop
    1 root        20   0   220M   6364  3740 S  0.7  0.3 17:43.57 /sbin/init
 30163 root        20   0 2625M   369M 11420 S  0.7  19.0 24:23.63 java -Dpg.host=db -jar app.jar
 23630 root        20   0   372M   9608  4840 S  0.0  0.5  6:21.76 php-fpm: master process (/etc/php/7.4/fpm/php-fpm.conf)
 16419 root        20   0   940M  52108  3096 S  0.0  2.6  3h26:06 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
 17662 root        20   0   940M  52108  3096 S  0.0  2.6  7:41.74 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
 16278 root        20   0   782M  22448  2172 S  0.0  1.1 17:43.82 /usr/bin/containerd
 23645 www-data    20   0   605M  12100  4560 S  0.0  0.6  0:00.41 php-fpm: pool www
 23644 www-data    20   0   605M  11736  4192 S  0.0  0.6  0:00.42 php-fpm: pool www
 29947 root        20   0   468M   2808  1412 S  0.0  0.1  0:03.34 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29952 root        20   0   468M   2808  1412 S  0.0  0.1  0:00.00 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29951 root        20   0   468M   2808  1412 S  0.0  0.1  0:00.00 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29950 root        20   0   468M   2808  1412 S  0.0  0.1  0:00.00 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29949 root        20   0   468M   2808  1412 S  0.0  0.1  0:00.00 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29948 root        20   0   468M   2808  1412 S  0.0  0.1  0:03.34 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5432 -container-ip 172
 29888 root        20   0   540M   2860  1224 S  0.0  0.1  0:03.74 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 5001 -container-ip 172
  
```

Вижу явный вирус. Думаю, что майнер какой-нибудь (на виртуалке за 3 евро, хе-хе). Смотрю дерево процессов, чтобы понять, откуда ноги растут.


```
16273 root      20    0  782M 22448  2172 S  0.0  1.1  4h42:24 - /usr/bin/containerd
29957 root      20    0  106M  1160   408 S  0.0  0.1  1:17.18 - containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd.ru
30177 root      20    0  106M  1160   408 S  0.0  0.1  0:01.00 - containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd
30176 root      20    0  106M  1160   408 S  0.0  0.1  0:28.12 - containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd
30010 999          20    0  208M 14824 13088 S  0.0  0.7  1:36.10 - postgres
30245 999          20    0  209M  2236   304 S  0.0  0.1  0:04.03 - postgres: logical replication launcher
30244 999          20    0  68364 2460   672 S  0.0  0.1  1:17.29 - postgres: stats collector
30243 999          20    0  209M  5144  3084 S  0.0  0.3  0:49.22 - postgres: autovacuum launcher
30242 999          20    0  208M  6040  4288 S  0.0  0.3  0:27.71 - postgres: walwriter
30241 999          20    0  208M  4360  2600 S  0.0  0.2  0:27.32 - postgres: background writer
30240 999          20    0  208M  9984  8196 S  0.0  0.5  0:01.97 - postgres: checkpointer
25774 999          20    0  209M 11696  9304 S  0.0  0.6  0:00.00 - postgres: postgres postgres 172.27.0.2(41148) idle
25647 999          20    0  209M 11692  9300 S  0.0  0.6  0:00.00 - postgres: postgres postgres 172.27.0.2(41146) idle
25641 999          20    0  209M 11692  9300 S  0.0  0.6  0:00.00 - postgres: postgres postgres 172.27.0.2(41144) idle
25613 999          20    0  209M 11696  9304 S  0.0  0.6  0:00.00 - postgres: postgres postgres 172.27.0.2(41142) idle
25605 999          20    0  209M 11692  9300 S  0.0  0.6  0:00.00 - postgres: postgres postgres 172.27.0.2(41140) idle
25599 999          20    0  209M 11696  9304 S  0.0  0.6  0:00.00 - postgres: postgres postgres 172.27.0.2(41138) idle
25578 999          20    0  209M 11692  9300 S  0.0  0.6  0:00.00 - postgres: postgres postgres 172.27.0.2(41136) idle
25557 999          20    0  209M 11692  9300 S  0.0  0.6  0:00.00 - postgres: postgres postgres 172.27.0.2(41134) idle
25482 999          20    0  209M 11692  9300 S  0.0  0.6  0:00.00 - postgres: postgres postgres 172.27.0.2(41132) idle
25320 999          20    0  209M 11692  9300 S  0.0  0.6  0:00.00 - postgres: postgres postgres 172.27.0.2(41130) idle
19480 999          20    0  298M  261M    4 S 98.0 13.4  4h13:59 - 1vLA4Q
19737 999          20    0  298M  261M    4 R 98.0 13.4  4h13:56 - 1vLA4Q
19729 999          20    0  298M  261M    4 S  0.0 13.4  0:00.00 - 1vLA4Q
19728 999          20    0  298M  261M    4 S  0.0 13.4  0:00.00 - 1vLA4Q
19727 999          20    0  298M  261M    4 S  0.0 13.4  0:00.00 - 1vLA4Q
19726 999          20    0  298M  261M    4 S  0.0 13.4  0:00.00 - 1vLA4Q
```

serveradmin.ru

Все понятно. На сервере стоит Docker. В нем контейнер Postgres. Вирус сидит в контейнере. Иду посмотреть, как запущен контейнер. Ничуть не удивлен тем, что он смотрит напрямую в инет.


```
root@r :~# netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:443             0.0.0.0:*                LISTEN      13616/nginx: master
tcp        0      0 0.0.0.0:80              0.0.0.0:*                LISTEN      13616/nginx: master
tcp        0      0 127.0.0.53:53           0.0.0.0:*                LISTEN      775/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      1024/sshd
tcp6       0      0 :::5001                 :::*                    LISTEN      29888/docker-proxy
tcp6       0      0 :::22                   :::*                    LISTEN      1024/sshd
tcp6       0      0 :::3000                 :::*                    LISTEN      6083/docker-proxy
tcp6       0      0 :::5432                 :::*                    LISTEN      29947/docker-proxy
udp        0      0 127.0.0.53:53           0.0.0.0:*                775/systemd-resolve
udp        0      0 0.0.0.0:68             0.0.0.0:*                763/dhclient
root@r :~#
```

Дальше решил проверить, с какими параметрами он запущен. Опять не удивляюсь.


```
/root/docker-compose.yml [----] 0 L:[ 1+ 0 1/ 28] *(0 / 638b) 0118 0x0
version: '3.1'
services:
  db:
    container_name:
    image: postgres
    restart: always
    volumes:
      - ./postgres-data:/var/lib/postgresql/data
    environment:
      - POSTGRES_USERNAME=postgres # fixme
      - POSTGRES_PASSWORD=postgres # fixme
    ports:
      - 0.0.0.0:5432:5432
```

В лучших традициях жанра, все запущено с дефолтной учеткой postgres/postgres. Фраза fixme игнорируется.

Решил на всякий случай проверить прод. **Захожу, а там все то же самое**, но вируса пока нет. Не стал разбираться, чтобы ненароком не словить бан от Hetzner. Первую виртуалку просто удалили. На второй сразу поменяли пароль.

Вот так легко и просто взламываются веб сервера под Linux. Ожидали чего-то большего? Не нужны никакие хакеры, суперсофт и скилл. Достаточно взять какой-то сканер и найти открытые в инет службы. Затем проверить дефолтные пароли к ним и наверняка что-то подойдет. Разработчиков с каждым годом все больше и больше, докер контейнеров тоже. Так что ломать сервера будет все проще и проще.

Тут, конечно, есть недоработка докера. Он зачем-то по дефолту мапит контейнеры к 0.0.0.0, а не к локалхосту. Понятно, что так проще разработчикам, но результат налицо. Разработчики чаще всего не парятся над этим и даже не обращают внимание. Я постоянно вижу на dev серверах контейнеры с backend, смотрящие напрямую в инет. Так что надо обязательно вручную заблокировать доступ к контейнерам Docker из интернета.

В данном случае, достаточно было заблокировать доступ к postgres с помощью iptables:

```
/sbin/iptables -I DOCKER-USER -i eth0 -p tcp --dport 5432 -j DROP
```

Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоём распоряжении. Расскажи, как сделать правильно!

Расскажите, встречались с похожими, простыми и очевидными факапами?

Онлайн курс Основы сетевых технологий

Теоретический курс с самыми **базовыми знаниями по сетям**. Курс подходит и начинающим, и людям с опытом. Практикующим системным администраторам курс поможет упорядочить знания и восполнить пробелы. А те, кто только входит в профессию, получают на курсе базовые знания и навыки, без воды и избыточной теории. После обучения вы сможете ответить на вопросы:

- На каком уровне модели OSI могут работать коммутаторы;
- Как лучше организовать работу сети организации с множеством отделов;
- Для чего и как использовать технологию VLAN;
- Для чего сервера стоит выносить в DMZ;
- Как организовать объединение филиалов и удаленный доступ сотрудников по vpn;
- и многое другое.

Уже знаете ответы на вопросы выше? Или сомневаетесь? Попробуйте пройти тест по основам сетевых технологий. Всего 53 вопроса, в один цикл теста входит 10 вопросов в случайном порядке. Поэтому тест можно проходить несколько раз без потери интереса. Бесплатно и без регистрации. Все подробности на странице .

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.