



Понадобилось настроить автоматическое выключение двух гипервизоров Xenserver 6.5 при отключении электроэнергии с помощью arcupsd. Информация уже не очень актуальная, так как сейчас вышел новый гипервизор под версией 7, но тем не менее, хочу зафиксировать свой опыт, может еще кому-нибудь пригодится. Подсказу для XenServer 7 тоже приведу. Был приобретен один UPS APC Smart-UPS SC 1500VA/865W и подключен по USB порту к одному из xenserver.

Если у вас есть желание научиться искать и эксплуатировать уязвимости в информационных сетях, рекомендую познакомиться с **онлайн-курсом «Практикум по Kali Linux»** в OTUS. Курс рассчитан на тех, у кого нет опыта в информационной безопасности, для поступления нужно пройти .

Содержание:

- 1 Введение
- 2 Устанавливаем arcupsd на XenServer
- 3 Настройка arcupsd
- 4 Заключение

## Введение

Идея решения заключается в том, что мы ставим демон управления упсом arcupsd на один из xenserver, а второй получает состояние упса по сети с первого сервера. Когда пропадет электричество, первый гипервизор напрямую получает информация от упса о том, что надо выключиться, а второй на пару минут раньше получает эту информацию от первого гипервизора. В итоге они оба благополучно завершают свою работу.

Идея не нова, утилиту **arcupsd** я постоянно использую в тех или иных случаях. Она очень удобна и функциональна. Позволяет настроить завершение работы на целом парке всевозможных систем при наличии всего одного упса, подключенного по usb к какому-нибудь серверу. Это достигается благодаря кроссплатформенности утилиты. Конкретно в моем случае, слушать состояние упса будут еще и другие серверы на базе линукс и windows. Но в рамках



данной статьи я рассмотрю вариант с двумя серверами. По аналогии вы можете добавить сколько угодно серверов.

## Устанавливаем arpcupsd на XenServer

У меня 2 гипервизора, соответственно установку нужно будет произвести на оба. Отличия будут позже в конфигурационных файлах. Работать будем на следующих версиях:

```
# cat /etc/redhat-release
XenServer release 6.5.0-90233c (xenenterprise)
```

В зависимости от установленных обновлений, какие-то пакеты вам будут не нужны. Я приведу полный набор пакетов, которые нужно установить. Мне, к примеру, пакет **glibc-common** не понадобился, уже был установлен. Устанавливаем необходимые пакеты:

```
# yum --enablerepo=base --disablerepo=citrix list glibc-common gcc
```

Скачиваем остальные пакеты. Если ссылки со временем умрут, поищите в гугле по названию пакета:

```
# mkdir /root/apc && cd /root/apc
# wget https://serveradmin.ru/files/rpm/mailx-8.1.1-44.2.2.x86_64.rpm
# wget https://serveradmin.ru/files/rpm/apcupsd-3.14.10-1.el5.x86_64.rpm
# rpm -i mailx-8.1.1-44.2.2.x86_64.rpm
# rpm -i apcupsd-3.14.10-1.el5.x86_64.rpm
```

Установка arpcupsd закончена. Все достаточно просто и быстро. Я сделаю подсказку для тех, кто попадет сюда в поиске информации по установке arpcupsd на XenServer 7. У меня нет возможности проверить, но скорее всего все заработает, если вы просто установите пакеты для версии centos 7, а не 5, как в моем примере. XenServer базируется на дистрибутиве CentOS, поэтому пакеты подходят. Попробуйте установить arpcupsd по этой ссылке, а mailx из пакетов в стандартном репозитории и у вас должно все заработать:

```
# yum --enablerepo=base --disablerepo=citrix list glibc-common gcc mailx
# wget https://dl.fedoraproject.org/pub/epel/7/x86_64/a/apcupsd-3.14.12-1.el7.x86_64.rpm
```



Двигаемся дальше и настроим arpcupsd в соответствии с нашими задачами.

## Настройка arpcupsd

Открываем конфигурационный файл arpcupsd на сервере, к которому подключен ups и редактируем. Я не рекомендую копировать и вставлять приведенный мной файл, лучше в своем файле измените нужные параметры. Так надежнее. Утилита как минимум читает комментарии в заголовке конфига, если вы их удалите, получите ошибку при старте сервиса.

```
# mcedit /etc/apcupsd/apcupsd.conf
```

```
## apcupsd.conf v1.1 ##  
#  
# for apcupsd release 3.14.10 (13 September 2011) - redhat  
#  
# "apcupsd" POSIX config file  
  
UPSCABLE usb  
UPSTYPE usb  
DEVICE  
LOCKFILE /var/lock  
SCRIPTDIR /etc/apcupsd  
PWRFAILDIR /etc/apcupsd  
NOLOGINDIR /etc  
ONBATTERYDELAY 6  
BATTERYLEVEL 10  
MINUTES 2  
TIMEOUT 0  
ANNOY 300  
ANNOYDELAY 60  
NOLOGON disable  
KILLDELAY 0
```



```
NETSERVER on
NISIP 192.168.0.1
NISPORT 3551
EVENTSFILE /var/log/apcupsd.events
EVENTSFILEMAX 50
UPSCCLASS sharemaster
UPSMODE share
STATTIME 10
STATFILE /var/log/apcupsd.status
LOGSTATS off
DATETIME 0
```

Я не буду приводить описание параметров, они очень хорошо прокомментированы разработчиками, там все понятно. Обращаю внимание на адрес **192.168.0.1**. В данном случае это адрес сервера, на котором установлен arpcupsd. По этому адресу к нему будут обращаться остальные серверы за состоянием упса.

Редактируем конфигурационный файл на клиенте:

```
# mcedit /etc/apcupsd/apcupsd.conf
```

```
## apcupsd.conf v1.1 ##
#
# for apcupsd release 3.14.10 (13 September 2011) - redhat
#
# "apcupsd" POSIX config file

UPSCABLE ether
UPSTYPE net
DEVICE 192.168.0.1:3551
LOCKFILE /var/lock
SCRIPTDIR /etc/apcupsd
PWRFAILDIR /etc/apcupsd
```



```
NOLOGINDIR /etc
ONBATTERYDELAY 6
BATTERYLEVEL 20
MINUTES 3
TIMEOUT 0
ANNOY 300
ANNOYDELAY 60
NOLOGON disable
KILLDELAY 0
NETSERVER off
NISIP 127.0.0.1
NISPORT 3551
EVENTSFILE /var/log/apcupsd.events
EVENTSFILEMAX 50
UPSCCLASS standalone
UPSMODE disable
STATTIME 10
STATFILE /var/log/apcupsd.status
LOGSTATS off
DATETIME 0
```

Не забудьте так сконфигурировать apcupsd на всех серверах, чтобы самым последним у вас завершал работу сервер, к которому подключен УПС. Если ошибиться и этого не сделать, то если завершит работу сервер с упс, остальные не смогут правильно определить свое время отключения. После пропадания связи с сервером, клиенты apcupsd не будут ничего предпринимать.

Теперь запускаем на обоих серверах службу:

```
# service apcupsd start
```



В XenServer 7:

```
# systemctl start apcupsd
```

Проверяете лог событий и файл состояния упса. В логе должны увидеть информацию о том, что сервис запущен, в файле состояния информацию от упса. На сервере и клиенте она будет разная.

```
# cat /var/log/apcupsd.events  
# cat /var/log/apcupsd.status
```

Статус сервера:



Статус клиента:



Одно важное замечание. Не забудьте открыть необходимые порты на фаерволе. Лично у меня на гипервизоре был включен фаервол и клиентская утилита arpcupsd не смогла подключиться к серверу и прочитать состояние упса. Исправляем это:

```
# iptables -I INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 3551 -j ACCEPT  
# iptables -I INPUT -p udp -m state --state NEW,ESTABLISHED -m udp --dport 3551 -j ACCEPT  
# iptables -I OUTPUT -p tcp --sport 3551 -m state --state ESTABLISHED -j ACCEPT  
# iptables -I OUTPUT -p udp --sport 3551 -m state --state ESTABLISHED -j ACCEPT  
# service iptables save
```

Я не знаю, по tcp или udp работает arpcupsd, поэтому открыл оба порта на вход и выход. После этого клиент успешно подключился к серверу и получил состояние упса. В завершение настройки, убедившись, что все работает, добавляем arpcupsd в автозагрузку:

```
# chkconfig --add apcupsd
```



```
# chkconfig arpcupsd on
```

В случае с Xenserver 7 команда будет такая:

```
# systemctl enable arpcupsd
```

## Заключение

Данная инструкция подойдет практически для любой ситуации использования arpcupsd. Она кроссплатформенная, конфиги легко переносятся с одной системы в другую. Будут отличия только в командах установки и запуска, в расположении лог файлов. В случае с линуксом, даже этих отличий почти не будет. Изменить пути в конфигурационном файле придется при переносе конфига с linux на windows.

Я рекомендую эту утилиту к использованию. Сам пользуюсь ей повсеместно. Лучшего решения по управлению упсами и безопасному завершению работы в разнородных сетях я не знаю. Обычно мое применение выглядит следующим образом. Я покупаю один УПС компании APC и подключаю его к одному серверу. Дальше приобретается сколько угодно упсов любой фирмы для любого количества серверов. Сервера подключаются по питанию к различным упсам, а информацию о состоянии электросети берут с сервера, к которому подключен APC. Когда пропадает свет, серверы по команде с этого упса завершают свою работу.

Получается бюджетная и эффективная схема. Проблема только одна — приходится завершать работу с большим запасом заранее, чтобы точно быть уверенным, что все остальные сервера успеют получить команду на завершение работы до того, как отключится основной сервер. Но я в этом не вижу большого неудобства, так как предпочитаю завершать работу серверов заранее, минут через 10 после пропадания электричества, не рассчитывая особо на длительное время работы от батарей.

## Практикум по Kali Linux

Курс для тех, кто интересуется проведением тестов на проникновение и хочет практически попробовать себя в ситуациях, близких к реальным. Курс рассчитан на тех, у кого еще нет опыта в информационной безопасности. Обучение длится 3 месяца по 4 часа в неделю. Что даст вам этот курс:

- Искать и эксплуатировать уязвимости или изъяны конфигурации в корпоративных сетях, web сайтах, серверах. Упор на пентест ОС



Windows и на безопасность корпоративного сегмента.

- Изучение таких инструментов, как metasploit, sqlmap, wireshark, burp suite и многие другие.
- Освоение инструментария Kali Linux на практике - с ним должен быть знаком любой специалист по ИБ.

Проверьте себя на вступительном тесте и смотрите подробнее программу по .

Помогла статья? Есть возможность отблагодарить автора