

Кочнев Егор Евгеньевич,

студент,

4 курс, факультет «Комплексной безопасности ТЭК»

РГУ нефти и газа (НИУ) имени И.М. Губкина

Россия, г. Москва

Научный руководитель: Уймин Антон Григорьевич,

Старший преподаватель кафедры БИТ

РГУ нефти и газа (НИУ) имени И.М. Губкина

Россия, г. Москва

НАСТРОЙКА ПОЛИТИК CONTROL GROUPS ДЛЯ ОГРАНИЧЕНИЯ РЕСУРСОВ НА УРОВНЕ ЯДРА ОС В УСЛОВИЯХ DDOS-АТАК

Аннотация: В данной статье исследуется применение технологии контрольных групп (Control Groups) в операционных системах семейства ALT Linux для обеспечения отказоустойчивости систем в условиях DDoS-атак. Проведено исследование механизмов управления ресурсами ядра и разработана методика автоматизированного реагирования на аномальные нагрузки с использованием системы мониторинга Zabbix. В ходе экспериментов протестированы сценарии жесткого ограничения ресурсов и динамической приоритезации критически важных служб (SSH, Zabbix-agent) для сохранения управляемости сервера под атакой. Предложенный алгоритм настройки и скрипты автоматизации позволяют эффективно изолировать вредоносный трафик и предотвращать полный выход системы из строя.

Ключевые слова: Control Groups (cgroups), DDoS-атака, ALT Linux, Zabbix, ограничение ресурсов, системное администрирование.

Annotation: *This article examines the application of Control Groups (cgroups) technology in ALT Linux operating systems to ensure system resilience under DDoS attack conditions. A study of kernel-level resource management mechanisms was conducted, and an automated response method was developed using the Zabbix monitoring system. Experiments tested scenarios of strict resource limitation and dynamic prioritization of critical services (SSH, Zabbix-agent) to maintain server manageability during an attack. The proposed configuration algorithm and automation scripts effectively isolate malicious traffic and prevent total system failure.*

Key words: *Control Groups (cgroups), DDoS attack, ALT Linux, Zabbix, resource limitation, system administration.*

Введение

Контрольные группы в Linux – это комплексное решение ряда проблем по определению выделяемого количества ресурсов системы для тех или иных процессов или сервисов системы. Они позволяют точнее управлять системами, выделяя на них лишь необходимое количество ресурсов и решают проблему чрезмерного потребления ресурсов процессами, вызывающего нехватку тех же ресурсов для других сервисов. В условиях DDoS-атак это подходящее решение, позволяющее в определенный момент времени ограничить ресурсы для ряда процессов и предотвратить выход остальных процессов или сразу всей системы из строя. Но, учитывая, что направленность и функционал Control Groups, заключается именно в создании контрольных групп с ограниченными ресурсами, необходимо использовать дополнительные утилиты и сервисы.

Использование технологии контрольных групп в отечественных операционных системах несколько описано, в частности в документации Alt Linux, но не освещено использование данной технологии с целью защиты от Ddos-атак.

Таким образом, актуальность данной работы заключается в отсутствии описания и примеров использования технологии контрольных групп с целью защиты от Ddos-атак

Целью данной работы является разработка и тестирование метода автоматизированного реагирования на DDoS-атаки в ОС семейства ALT Linux с использованием Control Groups.

Литературный обзор

Тема cgroups описана в Linux Kernel, и там указано, что cgroups, как правило, состоит из двух частей – ядра и контроллера, где первый отвечает за иерархическую организацию процессов, а второй за распределение ресурсов между ними. [2] При этом возможно использование вспомогательных контроллеров, выполняющих функции помимо распределения ресурсов. Также указаны базовые команды для работы с технологией.

Про суть и виды DDoS атак так же существует множество статей и работ, но стоит обратить внимание на лабораторную работу преподавателя ИТМО Пантюхина Игоря Сергеевича, где подробно описана суть, виды и процесс работы DDoS-атак, в том числе и описана используемая в статье атака вида HTTP-Flood. [3]

Готовые решения для проведения атак аналогично существуют и находятся в открытом доступе. В статье на GitHub автор предоставляет готовое решение с множеством видов атак в зависимости от протоколов и служб. [4]

Но вот статей или работ, в целом, с описанием использования cgroups с целью защиты от DDoS-атак найти не удалось – это подчеркивает актуальность и оригинальность данной работы.

Отслеживать нагрузку на устройство с запущенным веб-сервисом и его интерфейсы позволит система мониторинга Zabbix — свободная система мониторинга статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования, написанная Алексеем Владышевым[1]. Для

хранения данных используется MySQL, PostgreSQL, SQLite или Oracle Database, веб-интерфейс написан на PHP.

Методы эксперимента

Описание эксперимента:

Для эксперимента:

Для эксперимента нам понадобится 4 виртуальные машины с ОС Альт. Одна из них – атакуемая машина с установленными cgroups и Zabbix-agent, вторая – Zabbix-сервер, оставшиеся – атакующие с установленными DDos-скриптами с открытых источников.

Порядок работы:

1. Проверка работоспособности cgroups
2. Автоматизация работы
3. Дополнение эксперимента до аналогичных реальному миру действий

Ход эксперимента:

1. Проверка работоспособности cgroups

На атакуемой машине запускаем веб-сервер на python с формой ввода и отправки сообщений на сервер и с атакующих машин запускаем скрипт атаки командой:

```
#python3 start.py POST http://192.168.10.1:8080/send 0 200 0 200 600
```

В результате чего видим высокую нагрузку на процессор атакуемой машины.

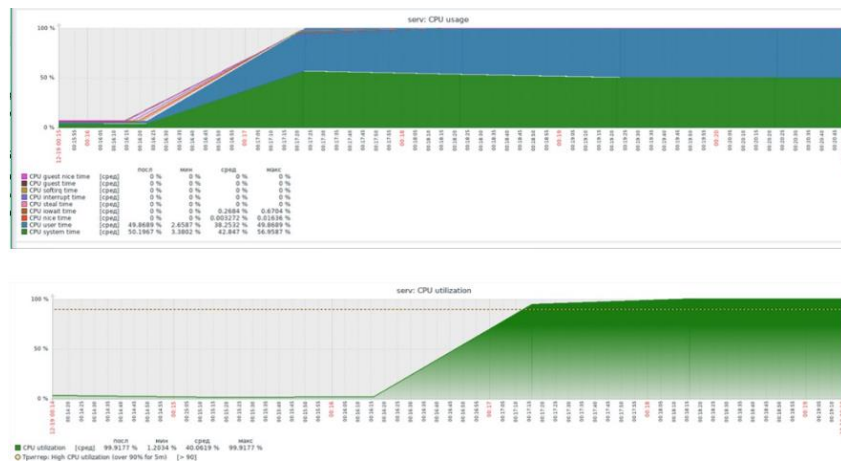


Рисунок 1. Нагрузка на CPU в момент атаки

После чего создаем контрольную группу с установленными ограничениями ресурсов, запускаем веб-сервер внутри этой группы и той же командой запускаем DDoS-атаку

```
[root@pcl ~]# cgcreate -g cpu,memory:/ddos
[root@pcl ~]# cgset -r cpu.max="10000 100000" ddos
[root@pcl ~]# cgset -r memory.max=500M ddos
[root@pcl ~]# cgexec -g cpu,memory:ddos python3 serv.py
```

Рисунок 2. Создание контрольной группы

В результате чего видим, что ограничение на потребление ресурсов работает и количество потребляемых ресурсов

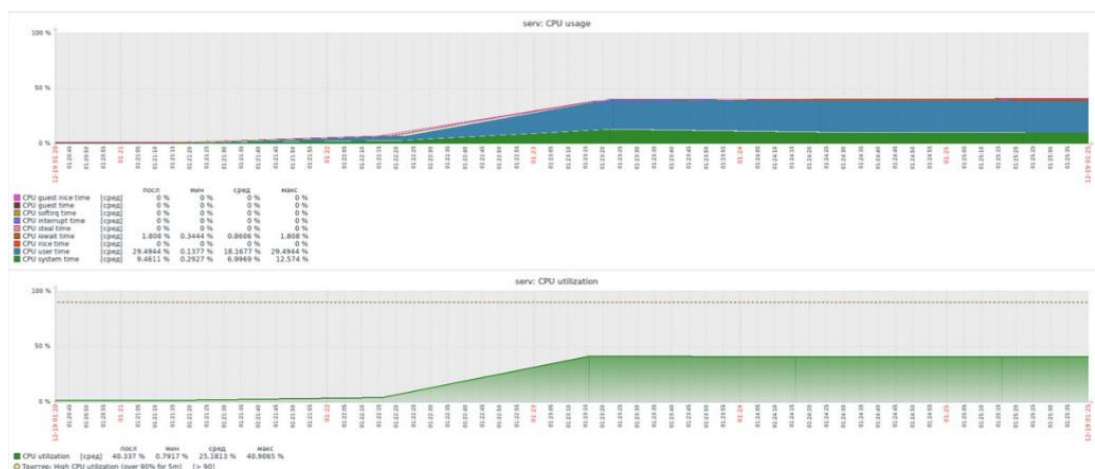


Рисунок 3. Работа контрольной группы

2. Для автоматизации работы:

Сначала дополним код веб-сервиса логированием ошибок связанных с предположительной DDoS-атакой, а именно с большим количеством запросов за короткий промежуток времени с одного и того же ip-адреса.

После чего создадим в Zabbix два триггера: на количество логов с ошибкой за определенное количество времени и на среднюю нагрузку на процессор за некоторый промежуток времени.

<input type="checkbox"/>	Важность	Значение	Имя	Оперативные данные	Выражение	Состояние	Инфо	Тег
<input type="checkbox"/>	Высокая	OK	ddos attack - info from srv		<code>count(/serv/log/var/log/serv_errors.log,429',2m) > 10000</code>	Активировано		
<input type="checkbox"/>	Чрезвычайная	OK	DDos_attack for cpu usage		<code>avg(/serv/system.cpu.util,2m) > 80</code>	Активировано		

Рисунок 4. Триггеры в Zabbix

Следующий шаг это указание действия zabbix-агента после срабатывания триггера, а именно выполнение скрипта, выдача прав на выполнение скрипта и его написание

```
#!/bin/bash
CGCREATE="/usr/bin/cgcreate"
CGSET="/usr/bin/cgset"
LSOF="/usr/bin/lsof"
TEE="/usr/bin/tee"
sudo $CGCREATE -g cpu,memory:ddos_quarantine
sudo $CGSET -r cpu.max="15000 100000" ddos_quarantine
PIDS=$(sudo $LSOF -t -i:8080)

if [ -z "$PIDS" ]; then
    echo "$(date): No processes found on port 8080"
    exit 1
fi

for PID in $PIDS; do
    echo "Moving PID $PID to quarantine"
    echo $PID | sudo $TEE /sys/fs/cgroup/ddos_quarantine/cgroup.procs
done
~
~
```

Рисунок 5. Скрипт для переноса процесса в контрольную группу

После чего снова проводим DDoS-атаку и анализируем графики, логи zabbix и результаты выполнения скрипта.

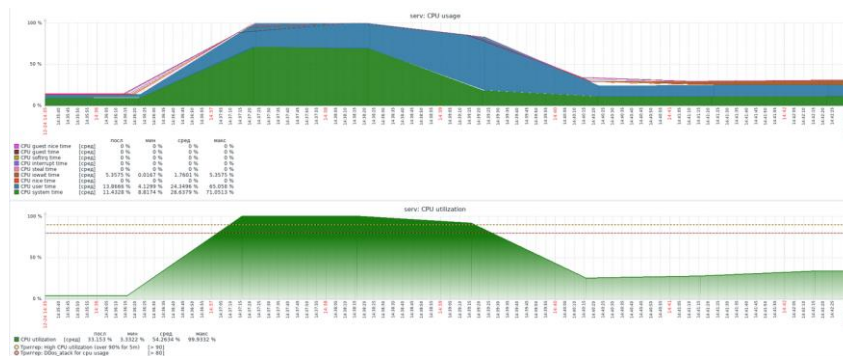


Рисунок 6. Результат отработки триггера и выполнения скрипта на графике

```
[root@pcl ~]# date
Пт 19 дек 2025 20:47:13 MSK
[root@pcl ~]# cat /sys/fs/cgroup/ddos_quarantine/cgroup.procs
[root@pcl ~]# tail -n 2 /var/log/zabbix/zabbix_agentd.log
973:20251219:204404.657 Executing command '/usr/local/bin/protect_system.sh'
981:20251219:204502.791 Executing command '/usr/local/bin/protect_system.sh'
[root@pcl ~]# cat /sys/fs/cgroup/ddos_quarantine/cgroup.procs
729031
[root@pcl ~]# tail -n 2 /var/log/zabbix/zabbix_agentd.log
981:20251219:204502.791 Executing command '/usr/local/bin/protect_system.sh'
970:20251219:205016.651 Executing command '/usr/local/bin/protect_system.sh'
[root@pcl ~]# date
Пт 19 дек 2025 20:50:29 MSK
[root@pcl ~]#
```

Рисунок 7. Логи Zabbix-agent и успешное выполнение скрипта

По логам и по наличию процесса в контрольной группе мы можем понять, что скрипт отработал успешно, что и отображено на графиках нагрузки на процессор атакуемой машины.

3. Дополнение эксперимента

Необходимость дополнения эксперимента заключается в том, что на данный момент происходит ограничение ресурсов веб-сервиса, что благоприятно сказывается на системе, но негативно на самом веб-сервисе. Это решение подходит для тех случаев, когда помимо веб-сервиса на машине одновременно работают еще ряд требовательных служб и сервисов, но в реальном мире веб-сервер это исключительно веб-сервер, а значит ограничением ресурсов не обойтись – необходимо в момент атаки обеспечить возможность удаленного управления и мониторинга системы.


```

[root@pcl ~]# date
Вс 11 янв 2026 15:59:45 MSK
[root@pcl ~]# systemctl show zabbix_agentd.service -p CPUWeight
CPUWeight=[not set]
[root@pcl ~]# systemctl show sshd.service -p CPUWeight
CPUWeight=[not set]
[root@pcl ~]# systemctl show web.service -p CPUWeight
CPUWeight=100
[root@pcl ~]# tail -f -n 2 /var/log/zabbix/zabbix_agentd.log
957:20260111:155716.353 Executing command '/usr/local/bin/protect_system.sh'
963:20260111:155916.446 Executing command '/usr/local/bin/protect_system.sh'
957:20260111:160116.363 Executing command '/usr/local/bin/protect_system.sh'
^C
[root@pcl ~]# systemctl show zabbix_agentd.service -p CPUWeight
CPUWeight=1000
[root@pcl ~]# systemctl show sshd.service -p CPUWeight
CPUWeight=1000
[root@pcl ~]# systemctl show web.service -p CPUWeight
CPUWeight=1
[root@pcl ~]# cat /sys/fs/cgroup/system.slice/web.service/cpu.max
90000 100000
[root@pcl ~]# date
Вс 11 янв 2026 16:02:47 MSK
[root@pcl ~]#

```

Рисунок 10. Результат отработки скрипта

В результате выполнения триггеров сменилась приоритезация распределения ресурсов процессора, из-за чего подключение по ssh произошло несколько быстрее, чем происходило в момент атаки до выполнения скрипта. На графиках нагрузки при этом, как и ожидалось, не произошло значительное изменение в показателях.

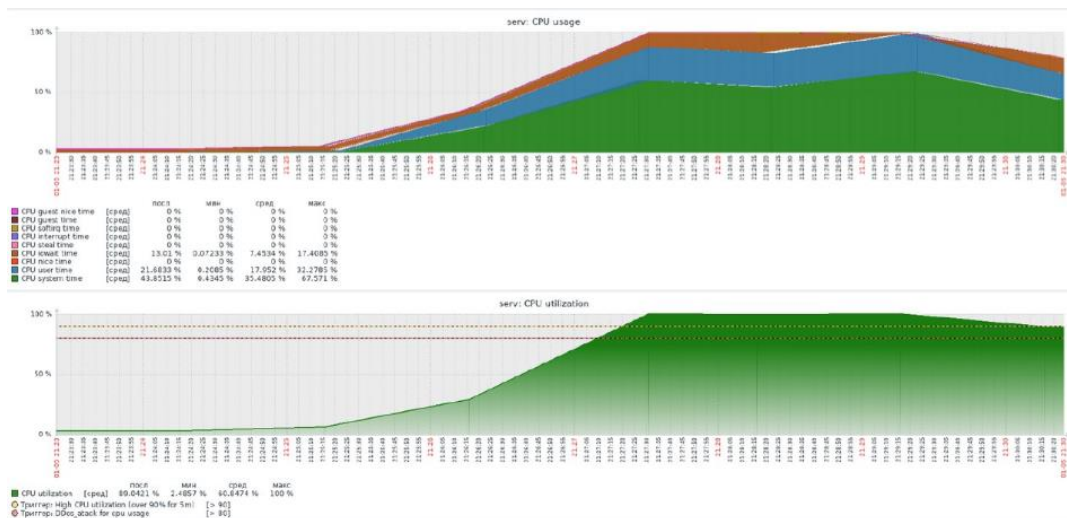


Рисунок 11. График нагрузки процессора

Результаты эксперимента

Проведенный эксперимент можно считать успешным, поскольку все действия соответствовали предполагаемому сценарию – высокая нагруженность сервера в результате атаки, реагирование Zabbix на

нестандартные данные о потреблении ресурсов и количестве логов сервера и изменение приоритизации, установка ограничения на потребление процессорного времени и обеспечение необходимыми ресурсами служб удаленного управления и мониторинга.

Список использованных источников:

1. ALT Linux Wiki Установка и первоначальная настройка ZABBIX [Электронный ресурс] – URL: https://www.altlinux.org/%D0%A3%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0_%D0%B8_%D0%BF%D0%B5%D1%80%D0%B2%D0%BE%D0%BD%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0_ZABBIX
2. Control Group v2 – The Linux Kernel documentation – [Электронный ресурс] – URL: <https://docs.kernel.org/admin-guide/cgroup-v2.html> (дата обращения: 23.12.2025)
3. Лабораторная №10 ОИБ ИТМО - [Электронный ресурс] – URL: <https://xn--80aqobguv5e.xn--p1ai/%D0%BE%D0%B8%D0%B1/lr10.html> (дата обращения: 15.12.2025)
4. Github MatrixTM/MHDDoS - [Электронный ресурс] – URL: <https://github.com/MatrixTM/MHDDoS> (дата обращения: 17.12.2025)
5. Уймин, А. Г. Демонстрационный экзамен базового уровня. Сетевое и системное администрирование : Практикум. Учебное пособие для вузов / А. Г. Уймин. – Санкт-Петербург : Издательство "Лань", 2024. – 116 с. – (Высшее образование). – ISBN 978-5-507-48647-2. – EDN BZJRIQ.