

# Содержание

[Введение](#)

[Перед началом работы](#)

[Условные обозначения](#)

[Предварительные условия](#)

[Используемые компоненты](#)

[Теоретические сведения](#)

[Настройка](#)

[Просмотр поддержки активности сценария в сервисе](#)

[Примитивы разъема](#)

[Администрирование сокета](#)

[Копирование файлов сценария](#)

[Выборки сценария](#)

[Проверка](#)

[Устранение неполадок](#)

[Дополнительные сведения](#)

## [Введение](#)

Этот документ описывает начальное внедрение заданных сценарием средств поддержки активности. Этот метод сценариев является самым тесно связанным с функциональностью, которая присутствует в Надежности, Доступности и Удобстве обслуживания (RAS) клиенты удаленного доступа, программы терминала и общие утилиты сценариев. Эта функция использует богатый язык сценария CSS.

Настройка, которую это с API простого разъема (подключает/разъединяет/передает/получает), даст пользователю способность адаптировать их собственный протокол или записать их собственную последовательность шагов для обеспечения надежного `ALIVE` или состояния `DOWN` сервиса. Вы в настоящее время ограничиваетесь FTP, HTTP, ICMP и TCP. С этой новой характеристикой можно остаться поверх текущих протоколов путем записи собственных сценариев. Например, пользователь может разработать сценарий, в частности настроенный для соединения с сервером POP3, не требуя, чтобы Cisco создала POP3 типа проверки активности для удовлетворения их потребностям. Эта функция позволит клиентам создавать свои собственные пакеты Keeralive для удовлетворения их определенным требованиям.

## [Перед началом работы](#)

### [Условные обозначения](#)

[Дополнительные сведения об условных обозначениях в документах см. Cisco Technical Tips Conventions.](#)

### [Предварительные условия](#)

Для данного документа отсутствуют предварительные условия.

## Используемые компоненты

Этот документ применяется к CSS 11000/CSS 11500 или CSS 11800 с Программным обеспечением webns.

Сведения, содержащиеся в данном документе, были получены с устройств в специальной лабораторной среде. Все устройства, описанные в данном документе, были запущены с конфигурацией по умолчанию. При работе с реальной сетью необходимо полностью осознавать возможные результаты использования всех команд.

## Теоретические сведения

Как только вы разработали сценарий Интерфейса командной строки (CLI) оффлайн с помощью текстового редактора, такого как Блокнот, можно загрузить тот сценарий к / каталогу сценария CSS и настроить параметр keeralive сценария на сервисе. Вам разрешают создать поддержку активности сценария, не имея подарка сценария в системе. В случае поддержки активности сценария, работающей без сценария в системе, постоянное состояние `DOWN` останется на сервисе. Это позволяет администратору писать конфигурацию и внедрять конфигурацию перед выписыванием (или загрузка) все сценарии.

Сценарий должен находиться в / <текущая рабочая версия>/script/каталог для поддержки активности сценария для обнаружения сценария. Пути не приняты, только имена сценариев при настройке поддержки активности сценария. Если сценарий будет присутствовать в другом месте в системе, то поддержка активности сценария предположит, что это не существует. Это означает, однако, когда программное обеспечение обновлено в системе, старые сценарии находятся в старой версии? s каталог сценария. Для копирования сценариев со старого каталога посмотрите [Раздел копирования файлов сценариев](#) этого документа.

Можно передать 128 символов в указанном аргументе. Принимая стандарт семи символов на аргумент, можно получить приблизительно 18 аргументов в одном сценарии. Командная строка только примет 90 символов как бы то ни было.

Рекомендуется, чтобы пакеты Кеераливе сценария были настроены с нижней частотой, чем стандартное поддержание активности, потому что много сценариев будут иметь многоступенчатый процесс, такой как соединение, отправление запроса и ожидание определенного типа ответа. Из-за этого, рекомендуется иметь частоту десяти секунды или выше так, чтобы поддержка активности имела время для завершения полностью. В противном случае изменения состояния могут произойти чаще.

Сценарий может вернуть код статуса нуля или ненулевой. На return ненулевых CSS отметит состояние сервиса как `DOWN`. На return нуля CSS отметит состояние сервиса как `ALIVE`. См. следующий сценарий для примера:

Если переменная `$_{SOCKET}` будет установлена в отрицательную, CSS будет всегда возвращать состояние `DOWN`. Очень важно проверить логику ваших сценариев, чтобы гарантировать, что возвращено правильное значение.

## Настройка

В этом разделе содержатся сведения о настройке функций, описанных в этом документе.

Для больших чисел сервисов, которые требуют использования пакетов Кеералив сценария, оно настоятельно рекомендовано это, они используют меньшее подмножество глобальных пакетов Кеералив для обработки работы для них.

Для настройки поддержки активности сценария придерживайтесь тех же рекомендаций что касается всех других типов проверки активности. Синтаксис команды показывается ниже.

```
CS100 (config-service[serv1])# keepalive type script ap-kal-smtp "einstein"  
или
```

```
CS100 (config-service[serv1])# keepalive type script ap-kal-pop3 "einstein vxworks mipspci"
```

Первая линия настроит текущую сервисную поддержку активности, чтобы быть сценария типа и иметь имя сценария **ap-kal-smtp** с аргументом **Einstein**. Вторая линия показывает **ap-kal-pop3**, который подобен первому сценарию, но передаст три аргумента: **эйнштейнов**, **vxworks**, и **mipspci**. Также возможно поразить **Вкладку** (или?) ключ для наблюдения полного списка всех сценариев, доступных в / <текущая рабочая версия> / каталог сценария. Это поощряет вас принимать соглашение о записи имен сценария так, чтобы при ударе **Вкладки** или?, можно ясно видеть сценарии поддержки активности, доступные использованию. Соглашение о записи имен **ap-kal-type** используется для сценариев. Например, сценарий SMTP назвали бы **ap-kal-smtp**. Дополнительно, можно ввести в одном не найденном в том списке, предположив, что вы хотите загрузить его позднее. Сценариями можно манипулировать через **архив**, **очиститься** и **скопировать** команды, и могут быть загрузкой/загрузкой из / каталога сценария на коммутаторе.

Имя сценария может быть 32 символа длиной. Аргументы нужно передать в строке в кавычках и могут быть 128 символов длиной. Для отключения поддержки активности сценария вы реконфигурировали бы сервис с помощью следующей команды:

```
CS100 (config-service[cs100])# keepalive type none
```

## [Просмотр поддержки активности сценария в сервисе](#)

Когда поддержка активности сценария настроена под сервисом, имя сценария может быть замечено в выходных данных команды **show service**. Сценарий обнаруживается под типом проверки активности, и любые потенциальные аргументы могут быть найдены непосредственно под той линией в поле `Script Arguments:`. Если нет никаких аргументов в сценарии, то поле не отображено. Пример сервиса **cs100** со сценарием, названным **ap-kal-pop3** с аргументами **vxworks** и **mipspci**, показывают ниже.

```
Name: cs100                               Index: 1  
Type: Local                               State: Suspended  
Rule ( 13.1.1.2  TCP  23 )  
Redirect Domain:  
Keepalive: (SCRIPT ap-kal-pop3  255  3  5 )  
Script Arguments: "einstein vxworks mipspci"  
Mtu: 1500                               State Transitions: 0  
Connections: 0                           Max Connections: 0  
Total Connections: 0                     Total Reused Conns: 0  
Weight: 1                                Load: 255
```

С этим питанием сценарий как **ap-kal-pop3** мог взять три параметра, которые будут совпадать с именем хоста, именем пользователя и паролем для сервера POP3. Просто регистрация к серверу POP достаточно для сценария, чтобы решить, что этот сервер находится в состоянии `ALIVE`.

Выходные данные команды **show running-config** показывают ниже.

```
service cs100
  protocol tcp
  port 23
  ip address 13.1.1.2
  keepalive frequency 255
  keepalive type script ap-kal-pop3 "einstein vxworks mipsci"
```

Вышеупомянутые выходные данные команды показывают точно, какой сценарий (и аргументы) были настроены на этом сервисе. Если никакие аргументы не будут присутствовать, то указанный текст после имени сценария не будет присутствовать.

При работе с ведением журнала сценариев конфигурация следующие:

```
service S1
  ip address 192.168.2.3
  keepalive type script ap-kal-pop3 "192.168.2.3 lab labtest1"
  keepalive frequency 10
  keepalive logging file testlog.log
  active
```

Данный пример показывает сервис, регистрирующий его выходные данные `keepalive` сценария к **testlog.log**.

## [Примитивы разъема](#)

Существует несколько **сокетных** команд, которые могут использоваться в поддержке активности сценария для помощи в построении структурированного протокола. Примитивы разъема обеспечат ASCII или шестнадцатеричную передачу и получают функциональность. Каждая команда, которая имеет дополнительное ключевое слово `RAW`, изменит данные от стандартного ASCII до шестнадцатеричного преобразования. Например, **abcd** в ASCII был бы представлен **61626364**, который обозначает `0x61 0x62 0x63 0x64`.

См. следующий список **сокетной** команды для получения дополнительной информации:

- **снабдите tcp порта порта имени хоста хоста подключения сокетом | udp [целое число] [сеанс]**- эта команда выполняет или TCP или UDP - подключение. Выполнение TCP - подключения включает квитирование (SYN-SYNACK...) к определенному IP/порту. Выполнение UDP - подключения является просто резервированием хоста/порта. Сокетное значение получено в переменной `_${SOCKET}` в сценарии.**Примечание:** Только 32 сокета могут быть открыты в любой момент через все сценарии на коммутаторе.Список ниже предоставляет сведения о каждом параметре.**host** - ключевое слово, которое должно придерживаться именем хоста или IP-адресом удаленной системы.**порт** - ключевое слово, которое должно придерживаться портом, относительно которого можно

выполнить согласование о соединении с TCP /\*- соединение с помощью TCP.udp - соединение с помощью UDP.целое число - значение таймаута для установки сети в секундах. Если временной предел истекает, прежде чем соединение было успешно сделано, сбой попытки. Это только применяется к TCP - подключению, поскольку UDP без установления соединения. Значение по умолчанию является таймаутом пяти секунд.сеанс - ключевое слово, которое говорит сокету оставаться открытым до сеанса, закончено. Это означает любые сценарии, которые открывают сокеты на сеансе и не закрывают их, самостоятельно останусь открытым до входов пользователя в систему.

- **сокетная передача socket# строка [сырые данные | base64]**- эта команда пишет данные посредством ранее связанного TCP - подключения.Список ниже предоставляет сведения о каждом параметре.**socket#** - сокетный дескриптор файла (целочисленная форма). Этот дескриптор возвращен из подключения.**строка** - указала текстовую строку до 128 символов в длине.**сырые данные** - если задано, заставляет строковые значения быть переданными как действительные шестнадцатеричные байты, а не простая строка. Например, **0D0A** не передается как '0?? D?? 0??', а скорее как **0x0D 0x0A** (или возврат каретки, перевод строки).**base64** - этот будет base64 кодировать строку прежде, чем передать его через соединение. Полезный для Базовой аутентификации HTTP при соединении с защищенным паролем веб-сайтом.
- **сокет получает socket# [целое число] [сырые данные]**- эта команда заполняет 10К внутренний буфер данными, входящими от удаленного хоста. Эта команда тогда блокирует буфер так, чтобы никакие новые данные не были помещены в этот внутренний буфер. Можно продолжить использовать, осматривают для формирования дампа всех данных, находящихся в этом внутреннем буфере 10К к стандартному выводу. **Примечание:** Все предыдущие данные в 10К внутреннем буфере спугивают, прежде чем новые данные вставлены.Список ниже предоставляет сведения о каждом параметре.**socket#** - сокетный дескриптор файла (целочисленная форма). Этот дескриптор возвращен из подключения.**целое число** - целое значение, представляющее количество миллисекунд для ожидания прежде, чем заблокировать внутренний буфер 10К и возвратиться к пользователю. Если никакой целочисленный таймаут не будет задан, получите, то будет ждать 100 мс прежде, чем возвратиться к пользователю.**сырые данные** - если задано, заставляет строковые значения быть переданными как действительные шестнадцатеричные байты, а не простая строка. Например, **0D0A** не передается как '0?? D?? 0??', а скорее как **0x0D 0x0A** (или возврат каретки, перевод строки).
- **waitfor сокета socket# строка [целое число] [регистрозависимое] [смещение] [сырые данные]**- эта команда подобна сокету, получают, за исключением того, что это возвращается непосредственно после обнаружения аргумента заданной строки. Как только заданная строка найдена, она возвратит `$(STATUS)` нуля. В противном случае это возвращается 1. Полученные данные могут быть далее просмотрены путем запуска команды **socket inspect**.Список ниже предоставляет сведения о каждом параметре.**socket#** - сокетный дескриптор файла (целочисленная форма). Этот дескриптор возвращен из подключения.**строка** - определенная строка, которая, как должны находить, приводит к `$(STATUS)` нуля, который указывает найденный. Если строка найдена, она сразу возвращается и не ждет весь целочисленный период времени ожидания.**целое число** - целое значение, представляющее количество миллисекунд для ожидания прежде, чем заблокировать внутренний буфер 10К и возвратиться к пользователю. Если никакой целочисленный таймаут не будет задан, получите, то будет ждать 100 мс прежде, чем возвратиться к пользователю.**case-sensitive** - если

задано, указывает, что сравнение строк должно учесть регистр. Например, **Пользователь:** не было бы эквивалентно **user:смещение** - сколько байтов в полученные данные строка должно быть найдено. Например, если вы будете искать **a0** и давать смещение десять, то вы будете искать десять байт **a0** в полученные данные. **сырые данные** - если задано, заставляет строковые значения быть переданными как действительные шестнадцатеричные байты, а не простая строка. Например, **0D0A** не передается как '0?? D?? 0??', а скорее как **0x0D 0x0A** (или возврат каретки, перевод строки).

- **сокеты осматривает [довольно] [необработанный] socket#**- эта команда осматривает сокеты с буфер данных (внутренний) для реальных данных. Если данные найдены, те данные отображены к стандартному выводу. Если отображенные символы будут непечатаемыми, то они будут представлены точкой (.) для удобочитаемости. Список ниже предоставляет сведения о каждом параметре. **socket#** - сокеты дескриптор файла (целочисленная форма). Этот дескриптор возвращен из подключения. **сырые данные** - если задано, заставляет строковые значения быть отображенными как действительные шестнадцатеричные байты скорее тогда простая строка. Скорее затем распечатывая **ABCD** к стандарту, это распечатало бы **41424344** (1 шестнадцатеричный эквивалент байта). **симпатичный** - структурированная печать выходных данных. Каждая линия будет содержать обоих ASCII или шестнадцатеричный эквивалент для каждого байта данных. Будет 16 байтов, распечатанных на каждой линии. Например, **0x41 0x42 0x43 0x44 0x10 0x05 ABCD**.
- **соедините сокеты разъединяют [постепенный] socket#**- эта команда закрывает соединение с удаленным хостом. Это сделано путем передачи RST к удаленному хосту так, чтобы это знало, что вы сделали, передав данные. Список ниже предоставляет сведения о каждом параметре. **socket#** - сокеты дескриптор файла (целочисленная форма). Этот дескриптор возвращен из подключения. **постепенный** - постепенное разъединение, которое передает FIN, а не RST к удаленному хосту для закрытия подключения.

## [Администрирование сокета](#)

Существует ограничение 32 открытых (в использовании) сокеты на коммутаторе в любой момент. Если пользователь сокет соединяется и не заканчивается сокетным разъединением для того сокета? s дескриптор файла (сохраненный в `$(SOCKET)`), сокет остается открытым, пока сокетное разъединение не вызвали с тем сокетом как параметр. Сокеты, открытые в рамках сценариев, закрыты, когда сценарий заканчивается (пока аргумент сеанса не передают в сокетном подключении). Когда сеанс заканчивается, закрыты сокеты, открытые в сеансах.

Если сокет остается открытым, это обычно - случай, где пользователь сделал соединение без надлежащего закрытия. Тот сокет останется открытым и используемым, пока он не будет закрыт или пока не был закрыт сценарий (или подключение терминала). **Команда show sockets** была внедрена для распечатки всех используемых сокетных дескрипторов файла, которые используются в настоящее время так, чтобы пользователь знал то, что открыто и что закрыто.

**Примечание:** Если удаленный хост вызывает таймаут сокета, или сокет закрыт удаленным хостом, архитектура сокета достаточно умна, чтобы очистить его и вынуть его из списка используемых сокеты (найденный путем запуска команды **show sockets**). Это только



произойдет после того, как пользователь пытается сделать другую передачу на сокет, который был закрыт удаленным хостом (иначе, это простаивает, ожидая пользовательских потребностей).

Выходные данные команды `show sockets` показывают ниже.

| Socket ID | Remote Host:Port | User    | Time            |
|-----------|------------------|---------|-----------------|
| 38        | 192.168.2.3:80   | console | 0 days 00:00:23 |
| 37        | 192.168.2.4:80   | console | 0 days 00:00:27 |
| 36        | 192.168.2.4:80   | console | 0 days 00:29:23 |
| 40        | 192.168.2.3:80   | console | 0 days 00:00:19 |
| 45        | 192.168.2.4:80   | vty1    | 0 days 00:00:02 |
| 44        | 192.168.2.4:80   | vty1    | 0 days 00:00:03 |

Экран перечисляет идентификатор сокета (дескриптор файла), пара/Port подключенного узла, Пользователь и таймер того, сколько времени дескриптор был открыт. Поле `User` представляло бы идентификатор линии, как замечено в выходных данных команды **выставочных подвидов** на CLI.

Получение данных с помощью сокета получает, в состоянии буферизовать 10К значимость данных когда-то. Этот буфер останется неизменным, пока пользователь не сделает другой сокет, получают, в этот момент буфер вычищен и пополнен с большим количеством данных, отрывающихся провод. Каждый дескриптор разъема (созданный от сокетного подключения) будет иметь свой собственный буфер 10К.

## Копирование файлов сценария

При обновлении к новой версии кода, все файлы сценария, которые вы модифицировали в предыдущей версии? с каталог сценария должен будет быть скопирован к новой версии.

Выполните эти действия прежде, чем обновить ваш коммутатор:

1. FTP к коммутатору CSS. Используйте адрес канала VLAN или порт управления.
2. CD к каталогу сценария.
3. Загрузите все сценарии, которые вы отредактировали к своему локальному компьютеру.
4. Обновите свой коммутатор.
5. Загрузить сценарии от своего локального компьютера до коммутатора? с новый каталог сценария.

## Выборки сценария

Следующие сценарии доступны для обеспечения вас некоторые реализации по умолчанию. Эти выборки содержат сценарии, записанные для DNS, Эха, NetBios, Finger, Время, HttpList, PingList, HttpAuth, Imap4, CookieSet, POP3, HttpTag, MailHost и SMTP.

Выполнить **сценарий показа?** команда для наблюдения сценариев. Ниже представлен пример выходных данных команды.

|                 |                        |                   |                            |
|-----------------|------------------------|-------------------|----------------------------|
| 17 09:58:36     | 1920 ap-kal-finger     | NOV 17 09:58:36   | 1172 ap-kal-httpauth       |
| NOV 17 09:58:36 | 1927 ap-kal-httpplist  | NOV 17 09:58:36   | 1674 ap-kal-               |
| httpstag        | NOV 17 09:58:36        | 1180 ap-kal-imap4 | NOV 17 09:58:36            |
| 1556            | ap-kal-ldap            | NOV 17 09:58:36   | 1640 ap-kal-mailhost       |
| NOV 17 09:58:36 | 2437 ap-kal-netbios    | NOV 17 09:58:36   | 1632 ap-kal-pinglist       |
| NOV 17 09:58:36 | 739 ap-kal-pop3        | NOV 17 09:58:36   | 1568 ap-kal-setcookie      |
| NOV 17 09:58:38 | 1436 ap-kal-smtp       | NOV 17 09:58:38   | 1310 ap-kal-ssl            |
| NOV 17 09:58:38 | 2053 ap-kal-time       | NOV 17 09:58:38   | 1064 cache.map             |
| NOV 17 09:58:38 | 1615 commit_redundancy | NOV 17 09:58:38   | 109224 commit_vip_redund.. |
| NOV 17 09:58:38 | 132147 default-profile | NOV 17 09:58:38   | 1240 dnslookup             |
| NOV 17 09:58:40 | 8009 eql-cacheable     | NOV 17 09:58:40   | 1186 eql-graphics          |
| NOV 17 09:58:40 | 234 eql-multimedia     | NOV 17 09:58:40   | 279 flowinfo               |
| NOV 17 09:58:40 | 5665 monitor           | NOV 17 09:58:40   | 3734 pcm-collect-cfgs      |
| NOV 17 09:58:40 | 2373 pcm-repeat-cmd    | NOV 17 09:58:40   | 4995 service-load          |
| NOV 17 09:58:40 | 920 setup              | NOV 17 09:58:40   | 24328 showtech             |
| NOV 17 09:58:40 | 2528 testpeering       | NOV 17 09:58:40   | 34142 upgrade              |
| NOV 24 09:18:00 | 17117 ap-kal-ssl.txt   | NOV 24 09:18:00   | 2053                       |

## Проверка

В настоящее время для этой конфигурации нет процедуры проверки.

## Устранение неполадок

Используйте следующие рекомендации для устранения проблем конфигурации:

- Выполните сценарий из командной строки путем выдачи команды **воспроизведения сценария**. Выполните эту команду, когда вошли как суперпользователя, чтобы удостовериться, что она завершает без ошибки. Если это убирает ошибку, линия, которая вызвала ошибку, должна быть выполнена.
- Возьмите отслеживание средств прослушивания между CSS и Web-сервером для наблюдения то, что фактически возвращено, когда сценарий выполнен по сравнению с тем, что сценарий ожидает видеть.
- В 5.x версии кода, была добавлена команда **use-output**. Эта команда должна использоваться со всеми сценариями, которые используют команду **grep**. Например, **выведенный использованием аргументов Ap-kal-dns сценария типа проверки активности**.

## Дополнительные сведения

- [Поддержка программного обеспечения Web Network Services](#)
- [Аппаратная поддержка коммутаторов контент-сервисов CSS 11000](#)
- [Аппаратная поддержка коммутаторов контент-сервисов CSS 11500](#)
- [Загрузки программного обеспечения WEBNS CSS11500 Cisco](#)
- [Загрузки программного обеспечения WEBNS CSS11000 Cisco](#)
- [Техническая поддержка - Cisco Systems](#)