



# Мониторинг Web



From:

<https://pandorafms.com/manual/!775/>

Permanent link:

[https://pandorafms.com/manual/!775/ru/documentation/03\\_monitoring/06\\_web\\_monitoring](https://pandorafms.com/manual/!775/ru/documentation/03_monitoring/06_web_monitoring)

2024/03/18 21:03



# Мониторинг Web

[Вернуться в оглавление Документации Pandora FMS](#)

## Классический Мониторинг Web

### Введение

**E** В версии Enterprise можно осуществлять мониторинг веб-сайта с помощью компонента WEB Server (Goliath Server).



Эта функция перешла из старого проекта основателя Pandora FMS: Goliath F.I.S.T. Это был проект *opensource* для проведения динамического аудита нагрузки на веб-сервисы. Вы по-прежнему можете найти [исходный код \(с 2002 года\)](#); перестал поддерживаться в июне 2010 года.

В Pandora FMS он работает как независимый сервер, подобно [Сетевому серверу](#), [Серверу WMI](#) или [Серверу удаленных плагинов](#). Эта система работает по принципу *веб-транзакции*, где каждая завершенная транзакция против одной или нескольких веб-страниц определяется одним или несколькими последовательными шагами, которые должны быть успешно завершены, чтобы считать транзакцию успешной. Выполнение *веб-транзакции* точно воспроизводит полный процесс навигации, который может включать такие аспекты, как аутентификация в форме, нажатие на опцию меню или заполнение формы, проверка того, что каждый шаг возвращает определенную текстовую строку.

Любой сбой на одном из этапов процесса приведет к сбою проверки. Полная транзакция включает загрузку всех ресурсов (графики, анимации и т.д.), включенных в фактическую навигацию. Помимо проверки производительности и времени отклика, можно извлекать значения из веб-страниц и затем обрабатывать их.

Goliath способен контролировать HTTP и HTTPS прозрачным для пользователя образом; он поддерживает управление сессиями через cookies, передачу параметров и, конечно, загрузку ресурсов, связанных с каждой страницей. Он имеет важные ограничения, такие

как динамическое управление JavaScript во время выполнения.. Для более сложных веб-транзакций Pandora FMS имеет другой компонент, гораздо более мощный (и сложный), который называется **Мониторинг WUX (Web User Experience)**.

## Установка и настройка

Для того чтобы использовать Goliath, **сначала необходимо активировать его** на сервере Pandora FMS Enterprise:

```
webserver 1
```

В зависимости от количества запросов, которые вы хотите сделать, вам, возможно, придется увеличить количество потоков и *тайм-аут* по умолчанию:

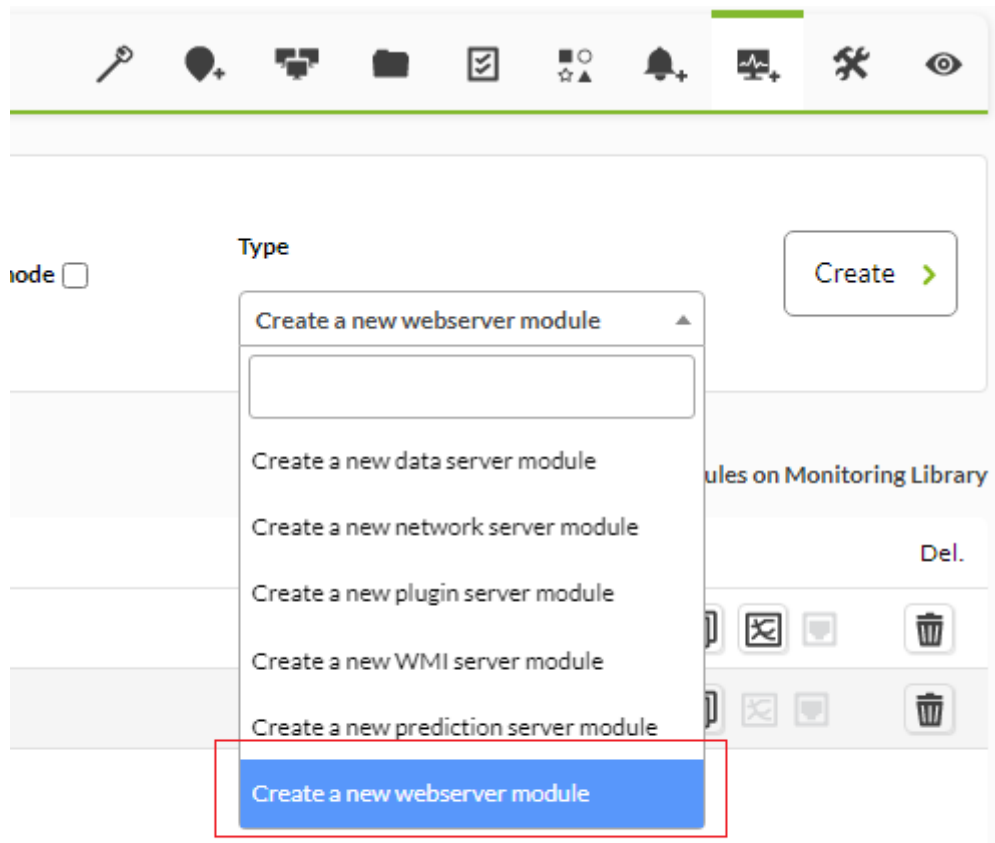
```
web_threads 1  
web_timeout 60
```

Существует *token*, расширенной конфигурации который позволит вам изменить тип библиотеки, которую использует Goliath ниже, LWP или CURL. CURL используется по умолчанию, но может быть изменен в любое время:

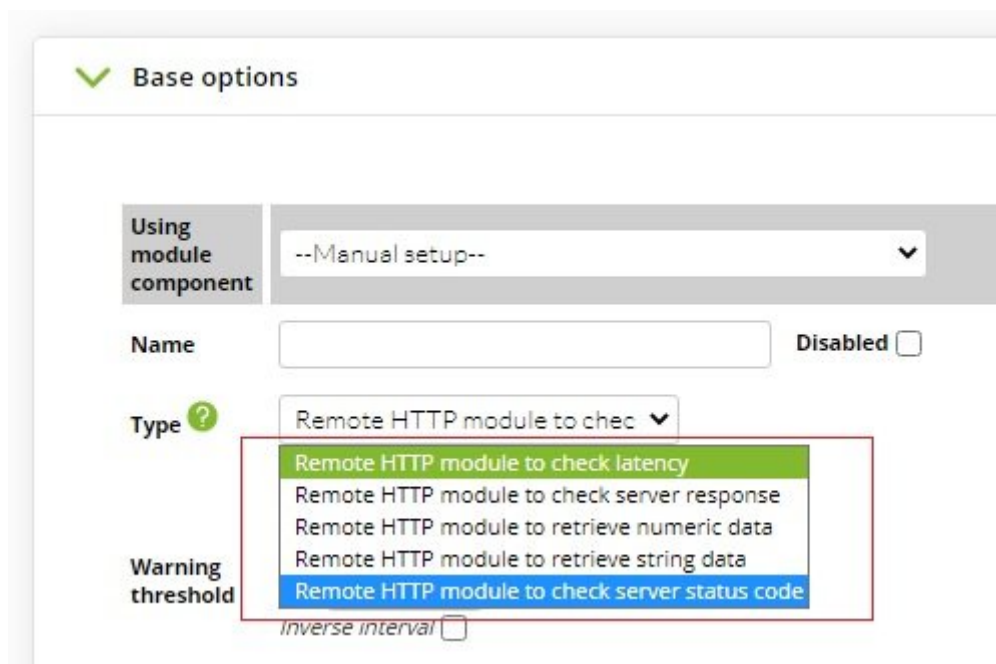
```
web_engine curl
```

## Создание веб-модулей

Чтобы удаленно контролировать веб-страницу, после создания агента перейдите на верхнюю вкладку модулей (Modules). В нем выберите создание нового модуля веб-сервера (Webserver module) и нажмите кнопку Create:



После нажатия на кнопку Create появится форма, в которой нужно заполнить необходимые поля для мониторинга web.



Вы должны выбрать тип WEB-проверки:

- Remote HTTP module to check latency: Получает общее время, которое проходит от первого

запроса до проверки последнего (в WEB-тесте есть один или несколько промежуточных запросов, которые завершают транзакцию). Если в определении проверки было указано, что транзакция выполняется более одного раза, то используется среднее время каждого запроса.

- Remote HTTP module to check server response: Получает 1 ( OK ) или 0 ( CRITICAL ) в результате проверки всей транзакции. Если есть несколько попыток, но хотя бы одна из них не удалась, тест в целом также считается неудачным. Точнее, количество попыток иногда используется, чтобы избежать ложных срабатываний, для этого используйте поле *повторные попытки* в расширенных полях.
- Remote HTTP module to retrieve numeric data: Получает числовое значение, *анализируя* ответ HTTP с помощью регулярного выражения для получения этого значения.
- Remote HTTP module to retrieve string data: Аналогично предыдущему пункту, но с текстовой строкой.
- Remote HTTP module to check server status code: С помощью инструмента curl, правильно включенного с помощью *token* конфигурации `web_engine curl`, вы можете компенсировать HTTP-заголовки.

**Web checks:** Это важное поле определяет выполняемую WEB-проверку. Это определяется одним или несколькими шагами, или простыми запросами. Эти простые запросы должны быть записаны в специальном формате в поле Web checks. Проверки начинаются со страницы вкладки `task_begin` и заканчиваются страницей вкладки `task_end`.

Полный пример простой транзакции выглядит следующим образом:

```
task_begin
head https://apache.org/
task_end
```

✓ Base options

Using module component: --Manual setup--

Name: Testing web\_engine curl Disabled

Type <sup>?</sup>: Remote HTTP module to chec

Warning threshold: Str:  Inverse interval

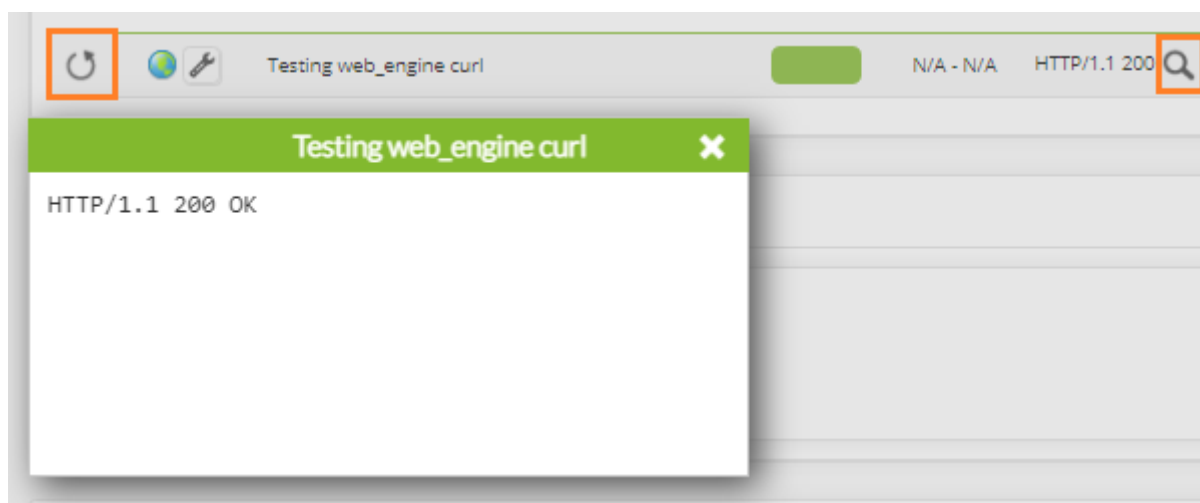
Critical threshold: Str:  Inverse interval

Historical data

Web Checks

```
task_begin
head http://apache.org/
task_end
```

После сохранения вы можете принудительно выполнить модуль и визуализировать его результат:



Другой пример с большим количеством команд:

```
task_begin
get https://apache.org/
cookie 0
resource 0
```

```
check_string Apache Software Foundation
task_end
```

В этом базовом примере мы проверяем, есть ли строка в веб-странице; для этого используется переменная `check_string`. Эта переменная не позволяет проверять сам HTML, она ищет только текстовые подстроки. Мы производим поиск «Apache Software Foundation» в web <<https://apache.org>>. Если эта строка существует, проверка возвращает ОК (если она имеет тип *Remote HTTP module to check server response*).

Чтобы убедиться, что строка не существует на веб-странице, вы можете использовать переменную `check_not_string`

```
check_not_string Section 3
```

Сценарии, принимаемые синтаксисом `check_string`, не являются обычными текстовыми строками, это *регулярные выражения*. То есть, если вы ищете строку Pandora FMS (4.0), вам придется искать ее с помощью регулярного выражения, например: Pandora FMS `\(4.0\)`. Это позволяет выполнять гораздо более мощный поиск, но следует учитывать, что любой символ, не являющийся буквой или цифрой, должен быть *экранирован* с помощью `\`.

Для проверки формы существует несколько дополнительных переменных:

- `resource` (1 или 0): Загружает все веб-ресурсы (изображения, видео и т.д.).
- `cookie` (1 или 0): Сохраняет *cookie*, или открытую сессию для последующих проверок.
- `variable_name` : Имя переменной в форме.
- `variable_value`: Значение вышеуказанной переменной в форме.

С помощью этих переменных вы можете отправлять данные в формы и проверять правильность их работы.

В некоторых случаях перенаправления доменов, проверки могут не сработать. Одним из способов решения этой проблемы является модификация модуля, указывающего на конечный домен.

Для приведенного выше случая команда `curl` имеет параметр

```
--L
```

в своей короткой версии и

```
--location
```



в своей длинной версии, поэтому при получении перенаправлений HTTP 3XX она запускается снова для перенаправленного домена. Однако гибкость Pandora FMS позволяет использовать кнопку отладки.

✓ Base options

Using module component: --Manual setup--

Name: Test mode debug  Disabled  Module group: Networking

Type: Remote HTTP module to chei

Warning threshold: Min: 0, Max: 0, Inverse interval:

Critical threshold: Min: 0, Max: 0, Inverse interval:

Historical data:

Web Checks: 

```
task_begin
get https://pandorafms.com/
cookie 0
resource 0
check_string Pandora FMS
task_end
```

Load basic  ⓘ

Check  ⓘ ✓

Debug  ⓘ

100  
80  
60  
40  
20  
0  
-20  
-40  
-60  
-80  
-100

■ Normal Status  
■ Warning Status  
■ Critical Status

> Advanced options

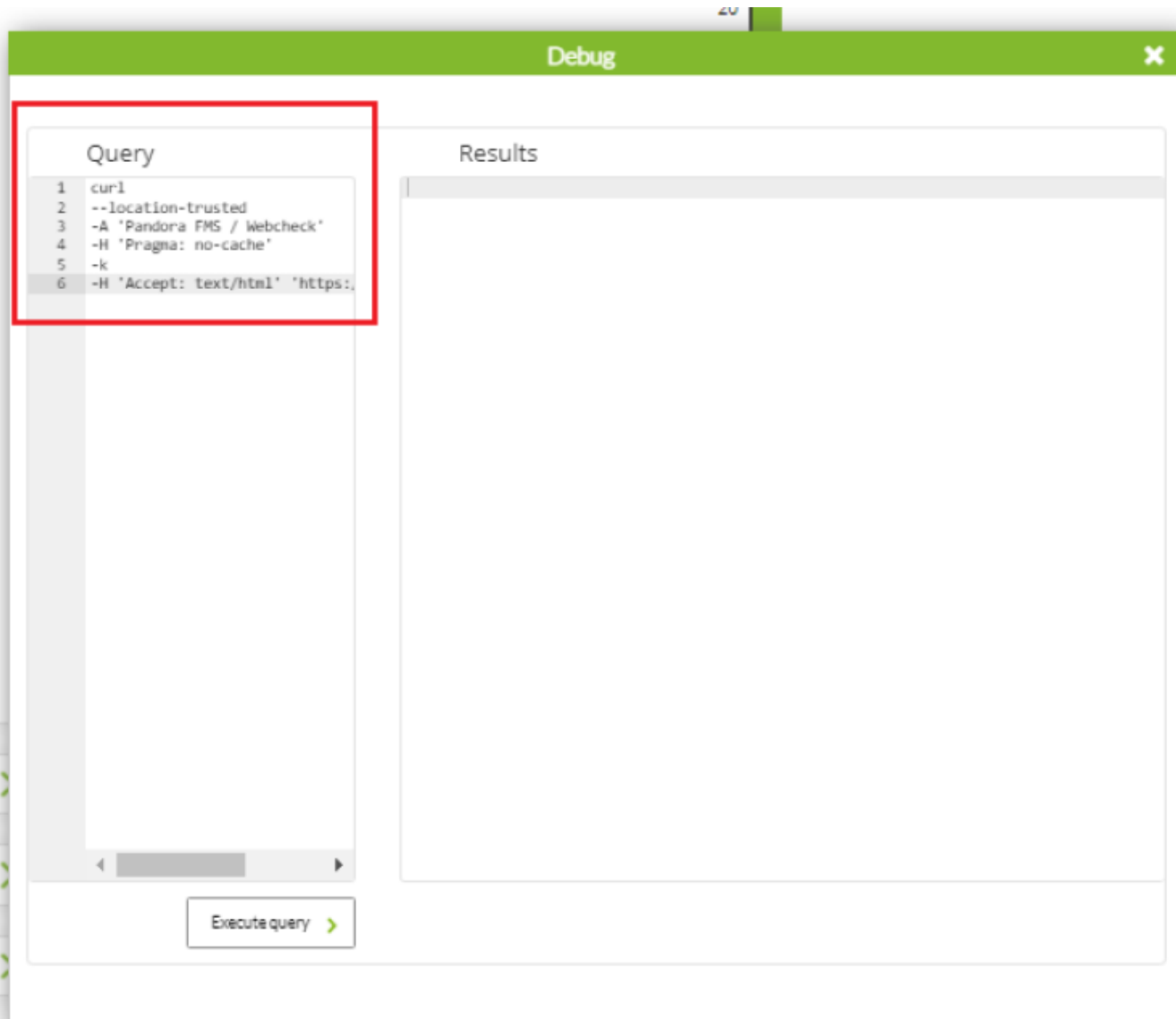
> Custom macros

> Module relations

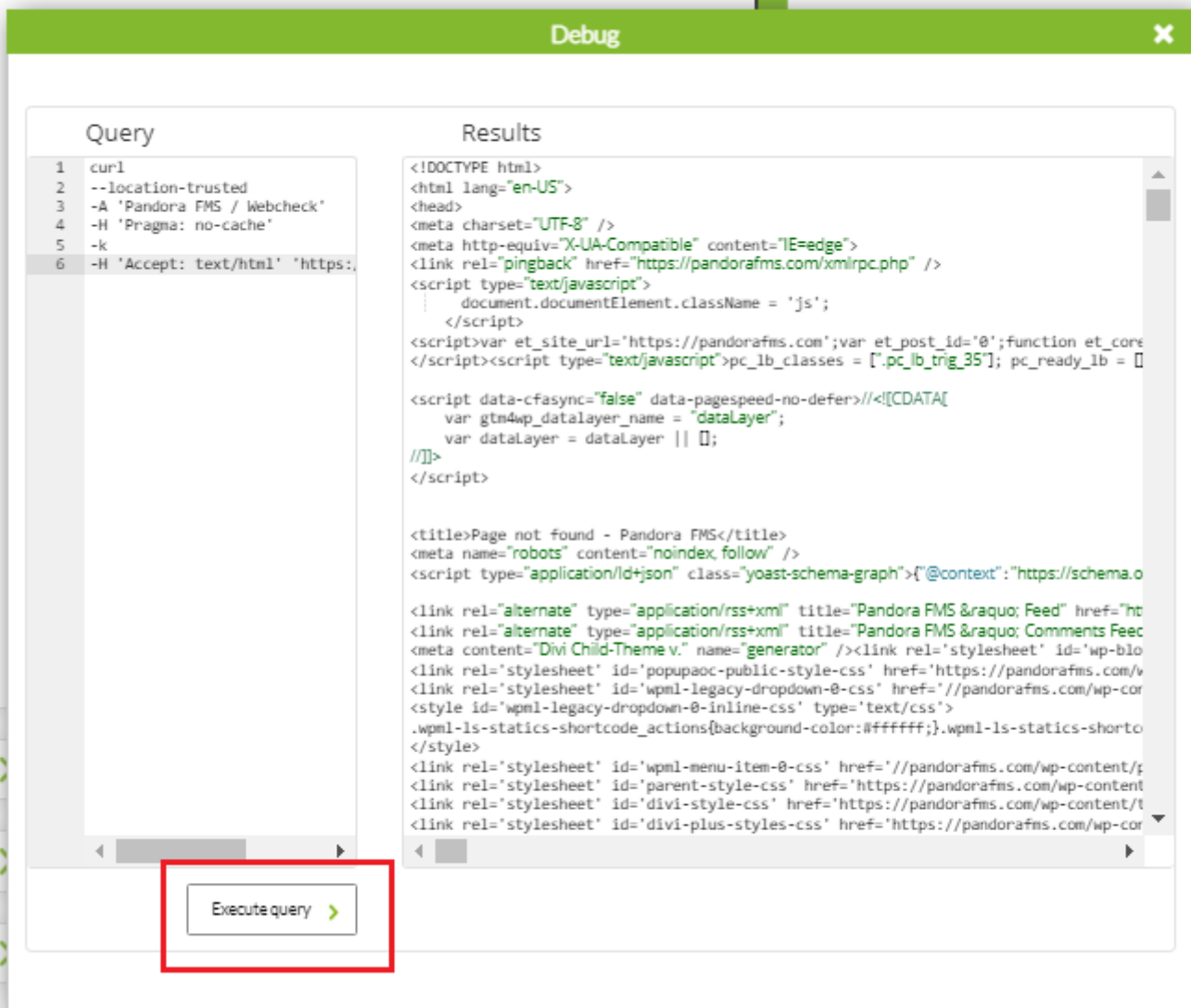
Create

В момент создания модуля он неактивен, и вы сможете использовать его после выполнения первой проверки, которую вы можете принудительно запустить для экономии времени.

Когда вы изменяете этот модуль, нажмите кнопку Debug, и вы сможете войти в режим отладки для редактирования запроса Query:



Вы можете выполнить запрос, который содержит модуль, с помощью кнопки Execute query, а также изменить и повторно выполнить с другими значениями, пока не получите желаемый результат.



## Проверка времени загрузки веб-сайта

Если мы хотим проверить время отклика или латентность веб-страницы, нам достаточно выбрать тип модуля Remote HTTP module to check latency. Например, если мы хотим узнать задержку загрузки веб-страницы <https://pandorafms.com>.

```
task_begin
get https://pandorafms.com
task_end
```

Мы можем добавить *token* конфигурации resource 1, чтобы вычисленное время загрузки загружало все ресурсы (JavaScript, CSS, изображения и т.д.), таким образом вычисляя данные, приближенные к реальности.

Время загрузки веб-сайта не является временем, которое требуется для отображения веб-сайта в браузере, и обычно зависит от времени загрузки

JavaScript. Хотя Goliat загружает его, он не выполняет его.

## Проверки через Proxu

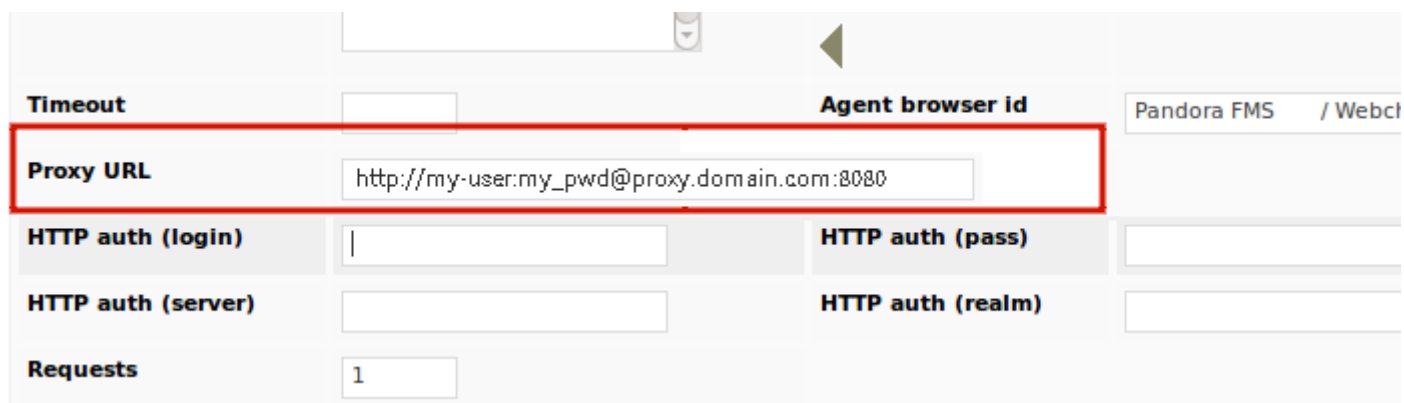
Веб-проверки также поддерживают использование *proxu*. Чтобы настроить его, необходимо добавить URL *прокси* в графе, которая находится при нажатии на Advanced options:

Например:

```
http://proxy.domain.com:8080
```

Если *прокси* требует аутентификации, вы должны использовать следующую схему, где *my-user* - это имя пользователя, а *my\_pwd* - это пароль:

```
http://my-user:my_pwd@proxy.domain.com:8080
```



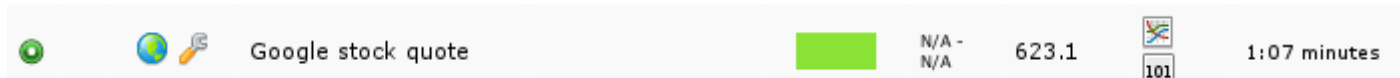
The screenshot shows a configuration form for Pandora FMS. The 'Proxy URL' field is highlighted with a red box and contains the text 'http://my-user:my\_pwd@proxy.domain.com:8080'. Other fields include 'Timeout', 'Agent browser id' (with value 'Pandora FMS / Webct'), 'HTTP auth (login)', 'HTTP auth (pass)', 'HTTP auth (server)', 'HTTP auth (realm)', and 'Requests' (with value '1').

## Получение данных с веб-сайта

Иногда мониторинг заключается не в том, чтобы проверить, работает ли конкретный сайт или сколько времени он работает, а в том, чтобы получить значение в реальном времени, например, рыночную стоимость акций Google. Для этой цели мы будем использовать модуль Remote HTTP module to retrieve numeric data с подходящим *регулярным выражением* или *regex*:

```
task_begin
get http://finance.google.com/finance/info?client=ig&q=NASDAQ%3aGOOG
get_content \d+\.\d+
task_end
```

Вы получите что-то похожее:



The screenshot shows a Pandora FMS console interface. At the top, there is a navigation bar with a globe icon, a pencil icon, and the text "Google stock quote". To the right, there is a green status indicator, the text "N/A - N/A", the value "623.1", a small line graph icon, and the text "1:07 minutes". Below the navigation bar, there is a task configuration window with the following content:

```
task_begin
get http://finance.yahoo.com/q?s=G00G
get_content_advanced <span id="yfs_l84_goog">([\d\.]+)</span>
task_end
```

Часть регулярного выражения, определенная в `get_content_advanced`, должна быть заключена в круглые скобки.

Чтобы настроить пороговые значения, которые будут запускать состояния предупреждения или критичности, используйте конфигурацию модуля для проверки соответствия полученной строки ожидаемому значению.

## Проверка формы на веб-сайте

Проверка формы намного сложнее, чем просто проверка текста на веб-странице. В этом примере используется консоль Pandora FMS, запускается сессия и проверяется, что, действительно, это удалось сделать. Работа осуществляется путем проверки текста в разделе *Workspace*, где отображаются собственные данные пользователя, который вошел в систему. Если это консоль по умолчанию, пользователь `admin` содержит описание «Admin Pandora».

У вас должны быть необходимые полномочия для проведения такого рода проверок. Кроме того, вам нужно *зайти* на страницу и получить HTML-код, чтобы получить имена переменных, а затем вам нужно иметь минимальные навыки работы с HTML, чтобы ввести запрос для *Goliat*.

Наиболее практичным способом разработки транзакционного WEB-теста с несколькими этапами является тестирование их по одному в режиме отладки.

Пример Консоли находится:

```
http://192.168.70.116/pandora_console/
```

Анализируя его HTML-код, можно заметить, что переменными веб-формы входа в систему являются:

- `nick`> имя пользователя
- `pass`> пароль пользователя

Следует использовать переменные `variable_name` и `variable_value` вместе для того чтобы проверить форму. Консоль PFMS по умолчанию имеет значения `admin` и `pandora`.

Первым шагом является доступ к форме, ввод имени пользователя и пароля и аутентификация (определение успешности аутентификации выполняется на втором шаге).

```
task_begin
post http://192.168.70.116/pandora_console/index.php?
login=1
variable_name nick
variable_value admin
variable_name pass
variable_value pandora
cookie 1
resource 1
task_end
```

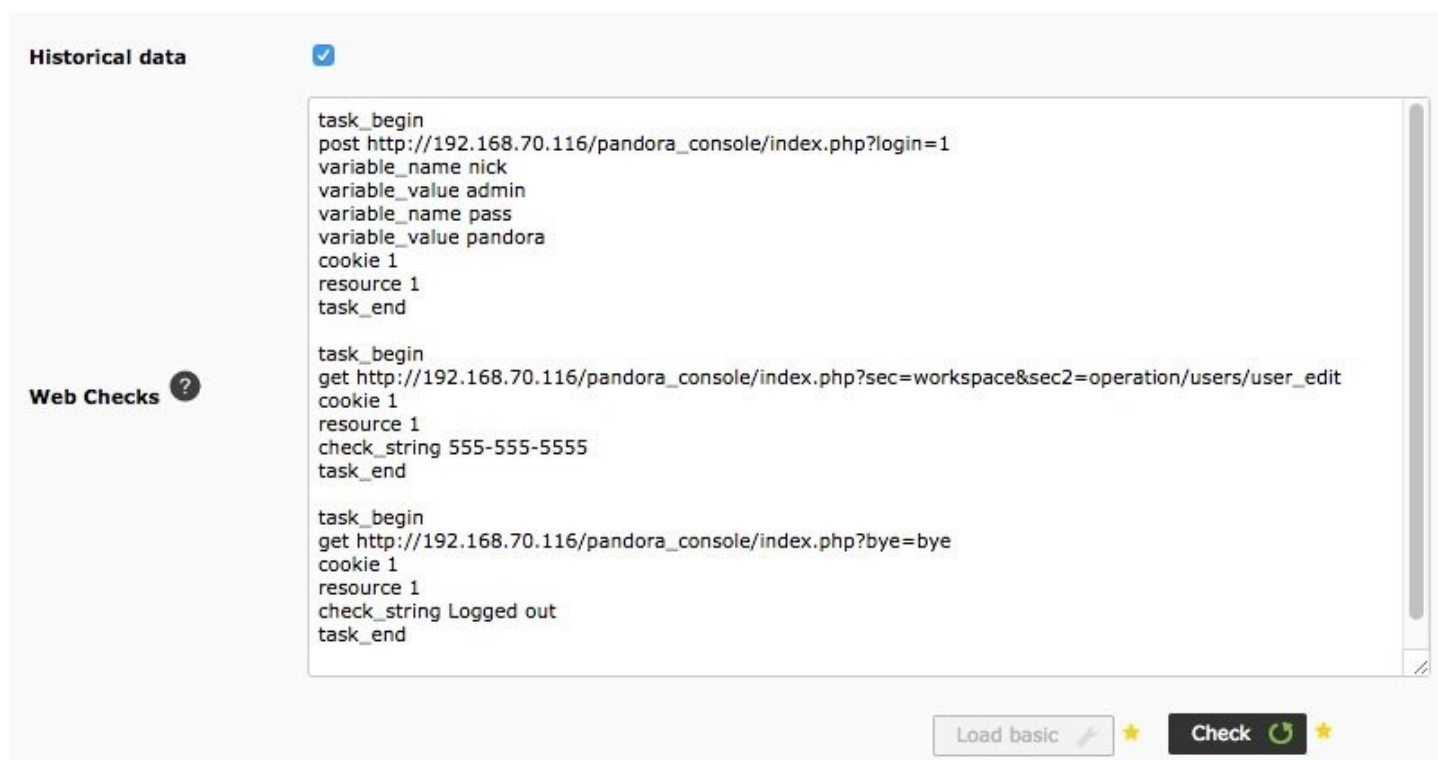
Токен `cookie 1` используется для поддержания постоянства значений, полученных на предыдущем этапе, при успешной аутентификации. Затем вы можете перейти на страницу дополнительных сведений о пользователе, чтобы найти номер телефона, который по умолчанию для пользователя «`admin`» равен `555-555-555`. Если вы его «видите», то это означает, что вы успешно вошли в консоль (обратите внимание на использование команды `check_string`):

```
task_begin
get http://192.168.70.116/pandora_console/index.php?
sec=workspace&sec2=operation/users/user_edit
cookie 1
resource 1
check_string 555-555-5555
task_end
```

Для завершения мы должны отключиться от консоли и найти сообщение об отсоединении `Logged out`>

```
task_begin
get http://192.168.70.116/pandora_console/index.php? bye=bye
cookie 1
resource 1
check_string Logged out
task_end
```

Общая проверка *script*:






**Historical data**

```
task_begin
post http://192.168.70.116/pandora_console/index.php?login=1
variable_name nick
variable_value admin
variable_name pass
variable_value pandora
cookie 1
resource 1
task_end

task_begin
get http://192.168.70.116/pandora_console/index.php?sec=workspace&sec2=operation/users/user_edit
cookie 1
resource 1
check_string 555-555-5555
task_end

task_begin
get http://192.168.70.116/pandora_console/index.php?bye=bye
cookie 1
resource 1
check_string Logged out
task_end
```

**Web Checks** ?

Load basic  **Check**  

## Поведение WEB-запросов

Поля расширенных свойств похожи на поля других типов модулей, хотя есть несколько различных полей, специфичных для WEB-проверки:

### Timeout

Время истечения срока действия запроса; если это время превышено, запрос на проверку будет отменен.

### Agent browser id

Идентификатор используемого веб-браузера, поскольку некоторые страницы принимают только определенные веб-браузеры (подробнее: [https://www.zytrax.com/tech/web/browser\\_ids.htm](https://www.zytrax.com/tech/web/browser_ids.htm)).

### Requests

Pandora FMS повторит проверку столько раз, сколько указано в этом параметре. Если одна из проверок завершится неудачно, о проверке будет сообщено как о неудачной. В зависимости от количества проверок в модуле получится определенное количество страниц, т.е. если модуль состоит из трех проверок, будет загружено три страницы; если в поле Requests установлено некоторое значение, то количество загрузок будет умножено на это значение. Это важно учитывать при расчете общего времени, которое потребуется модулю для выполнения операций.

## Retries

Количество повторений выполнения Request до получения успешного результата. Примеры:

- retries = 2 и Requests = 1: Если первая проверка не удалась, она повторит попытку еще раз, и если вторая сработает, то проверка действительна.
- Retries = 1 и Requests = 2: Выполняет две проверки, но если одна из них провалится, то проверка будет считаться неудавшейся.

## Использование аутентификации HTTP Simple

Для некоторых страниц может потребоваться **аутентификация simple HTTP**. Обычно она используется в качестве быстрого, минимального «приветствия» безопасности, которое позволяет получить доступ к более продвинутым проверкам безопасности (шифрование, сохранение данных и т.д.).

Timeout	<input type="text" value="0"/> ★	Retries	<input type="text" value="0"/> ★
Category	<input type="text" value="None"/> ▼		
Check type	<input type="text" value="Anyauth"/> ▼		
Requests	<input type="text" value="1"/>	Agent browser id	<input type="text" value="Pandora FMS / Webcheck"/>
HTTP auth (login)	<input type="text"/>	HTTP auth (password)	<input type="text"/>
Proxy URL	<input type="text"/>		
Proxy auth (login)	<input type="text"/>	Proxy auth (pass)	<input type="text"/>
Proxy auth (server)	<input type="text"/>	Proxy auth (realm)	<input type="text" value="public"/>

Его можно настроить в дополнительных параметрах проверки (или непосредственно в определении WEB-задачи) с помощью следующих токенов конфигурации:

### Check type

Тип проверки сервера HTTP.

http auth (login)

*Пользователь http.*

http auth (password)

*Пароль пользователя http.*



Proxy auth realm

Имя зоны (*realm*) аутентификации.

Proxy auth (server)

Web-домен и его порт.

Proxy URL

Адрес url прокси.

Proxy auth (login)

Пользователь связи *проху*.

Proxy auth (pass)


Пароль связи *проху*.

Полный пример:

```
task_begin
get http://artica.es/pandoraupdate4/ui/
cookie 1
resource 1
check_string Pandora FMS Update Manager \ (4.0\ )
http_auth_serverport artica.es:80
http_auth_realm Private area
http_auth_user admin
http_auth_pass xxxx
task_end
```

Использование кавычек в пароле для `http_auth_pass` не поддерживается. Избегайте использования одинарных кавычек ' '.

## Мониторинг веб-сервисов и API

С помощью Pandora FMS и Goliath вы можете контролировать API типа  **REST**, исключая более сложные API типы, основанные на протоколах, таких как SOAP или XMLRPC.

Например, для тестирования API с помощью этого типа вызова, который возвращает целое число, большее или равное нулю, если он работает, и ни одного символа в случае неудачи (с использованием учетных данных `my_user` и `my_pass`):

```
task_begin
get
http://artica.es/integria/include/api.php?user=my_user&pass=my_pass&op=get_stats
¶ms=opened,,1
check_string \n[0-9]+
task_end
```

Возвращает ответ, аналогичный:

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Connection: close
Date: Mon, 13 May 2013 15:39:27 GMT
Pragma: no-cache
Server: Apache
Vary: Accept-Encoding
Content-Type: text/html
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Client-Date: Mon, 13 May 2013 15:39:27 GMT
Client-Peer: 64.90.57.215:80
Client-Response-Num: 1
Client-Transfer-Encoding: chunked
Set-Cookie: a81d4c5e530ad73e256b7729246d3d2c=pcasWqI6pZzT2x2AuWo602; path=/
0
```

Проверяя вывод с помощью регулярного выражения, вы можете убедиться, что все правильно. Для более сложных ответов следует использовать другие регулярные выражения в зависимости от каждого конкретного случая.

Еще несколько примеров:

```
task_begin
get https://swapi.dev/api/planets/1/
get_content Tatooine
task_end
```

В этом случае модуль, созданный для демонстрации данных, должен быть установлен на тип Remote HTTP module to retrieve string data (web\_content\_string).

```
task_begin
get https://pokeapi.co/api/v2/pokemon/ditto/
get_content imposter
task_end
```

Как и в предыдущем модуле, тип данных должен быть 'Remote HTTP module to retrieve string data (web\_content\_string)' для правильной работы.

Вы также можете совершать вызовы с помощью get\_content\_advanced:

```

task_begin
get https://api.hillbillysoftware.com/Awards/ByYear/1990
get_content_advanced "Nominee":"([A-Za-z ]+)", "Year":"1990"
task_end

```

Результат:

```

-<ArrayOf_Awards>
  -<_Awards>
    <Category>Outstanding Lead Actor In A Miniseries Or Special</Category>
    <Nominee>Hume Cronyn</Nominee>
    <Type>Emmy</Type>
    <Winner>1</Winner>
    <Year>1990</Year>
  </_Awards>
  -<_Awards>
    <Category>Outstanding Lead Actor In A Miniseries Or Special</Category>
    <Nominee>Michael Caine</Nominee>
    <Type>Emmy</Type>
    <Winner>0</Winner>
    <Year>1990</Year>
  </_Awards>
  -<_Awards>
    <Category>Outstanding Lead Actor In A Miniseries Or Special</Category>
    <Nominee>Tom Hulce</Nominee>
    <Type>Emmy</Type>
    <Winner>0</Winner>
    <Year>1990</Year>
  </_Awards>
  -<_Awards>
    <Category>Outstanding Lead Actor In A Miniseries Or Special</Category>
    <Nominee>Albert Finney</Nominee>
    <Type>Emmy</Type>
    <Winner>0</Winner>
    <Year>1990</Year>
  </_Awards>
  -<_Awards>
    <Category>Outstanding Lead Actor In A Miniseries Or Special</Category>
    <Nominee>Art Carney</Nominee>
    <Type>Emmy</Type>
    <Winner>0</Winner>
    <Year>1990</Year>
  </_Awards>
  -<_Awards>
    -<Category>
      Outstanding Lead Actress In A Miniseries Or Special
    </Category>
    <Nominee>BARBARA HERSHEY</Nominee>
    <Type>Emmy</Type>
    <Winner>1</Winner>
    <Year>1990</Year>
  </_Awards>

```

Будет отражено в Pandora FMS как:

Pandora FMS Agent /Normal



Hume Cronyn

Важно правильно определить группы записи между скобками, чтобы вызов был выполнен правильно.

При выполнении вызовов API важно помнить, что целевой API должен иметь разрешения на получение запроса.

## Мониторинг https

Goliat может проверять как http, так и https. Для того чтобы иметь возможность осуществлять проверки в безопасном Интернете, достаточно указать этот протокол в URL, например:

```
task_begin
get
https://www.google.com/accounts/ServiceLogin?service=mail&passive=true&rm=false&
continue=https%3A%2F%2Fmail.google.com%2Fmail%2F%3Fui%3Dhtml%26zy%3Dl&bsv=zpwhty
gjnrz&ss=1&sc=1<mpl=default<mplcache=2
cookie 1
resource 0
check_string Google
task_end
```

## Дополнительные параметры

### Изменение заголовков HTTP

С помощью опции *header* вы можете изменять поля заголовков HTTP или создавать пользовательские поля. Например, чтобы изменить поле *Host* заголовка HTTP:

```
task_begin
get http://192.168.1.5/index.php
header Host 192.168.1.1
task_end
```

## Отладка веб-проверок

Вы можете отладить веб-проверки, добавив параметр `debug <log_file>`. Он создаст два файла `log_file.req` и `log_file.res` с содержимым HTTP-запроса и ответа, соответственно. Например:

```
task_begin
get http://192.168.1.5/index.php
debug /tmp/request.log
task_end
```

## Использование Curl вместо LWP

Библиотека LWP может иметь проблемы, когда много потоков выполняют HTTPS-запросы (из-за ограничения OpenSSL). Альтернативой является использование [инструмента curl](#). Чтобы исправить эту проблему, отредактируйте файл `/etc/pandora/pandora_server.conf` и добавьте следующую строку:

```
web_engine curl
```

Перезапустите сервер Pandora FMS, и для веб-проверки вместо LWP будет использоваться двоичный файл Curl.

## Расширенный мониторинг транзакций

В дополнение к функциям, предлагаемым Goliat, существуют и другие способы осуществления транзакционного веб-мониторинга.

- [Распределенный \(UX\)](#), разворачивается в режиме «агента» на системах, отличных от сервера, даже в недоступных сетях.
- Централизованно (WUX).

[Вернуться в оглавление Документации Pandora FMS](#)