ウェブモニタリング

Pandora FMS ドキュメント一覧に戻る

クラッシックなウェブ監視

概要

Enterprise 版では、Goliat サーバと呼ばれる Webサーバコンポーネントを用いて、Web を監視することができます。

この機能は、Pandora FMS の創設者の古いプロジェクトが元になっています。Goliat F.I.S.T. は、Web サービスにおいて動的な認証の実行を行うオープンソースのプロジェクトでした。(2002年から)ソースコードを確認することができますが、更新は 2010年で終了しました。
https://sourceforge.net/projects/goliat/

Pandora FMS では、ネットワークサーバ、WMIサーバ、プラグインサーバなどと同様に、独立したサーバとして機能します。 このシステムは、“Webトランザクション ” という考えのもとで動作します。ここでは、各トランザクションは 1つまたは複数の連続ステップで定義され、トランザクションを正常に完了させるために、正しく順序だてられている必要があります。“Webトランザクション ” の実行では、フォーム内での自分自身の認証、メニューオプションのクリック、フォームの入力、各ステップで特定のテキスト文字列を返すことの確認など、完全なブラウジングプロセスを忠実に再現します。

処理のある時点で障害が発生するとチェックに失敗します。 完全なトランザクションには、実際のナビゲーションに含まれるすべてのリソース（グラフィックス、アニメーションなど）のダウンロードが含まれます。 応答時間とパフォーマンスチェックの実行に加えて、Webページから値を抽出して処理することも可能です。

Goliatは、HTTP と HTTPS の両方を透過的に監視し、クッキーによるセッション管理、パラメータの受け渡しをサポートします。もちろん、各ページに関連するリソースをダウンロードできます。ただし、実行時の動的な javascript の管理などに制限があります。 より複雑な Web トランザクションの場合、Pandora FMS には WUX 監視というより強力で複雑なコンポーネントがあります。

インストールと設定

Goliat を利用できるようにするには、最初に Pandora FMS Enterprise サーバを有効化する必要があります。

```
webserver 1
```

実行したいリクエスト数に応じて、スレッド数およびデフォルトのタイムアウトを増やします。

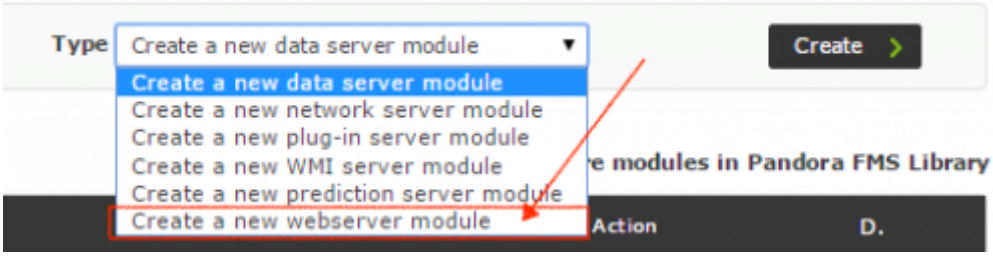
```
web_threads 1
web_timeout 60
```

Goliat が使うライブラリのタイプ (LWP または CURL) を変更できる拡張設定トークンがあります。デフォルトでは CURL が利用されていますが、切り替えることができます。

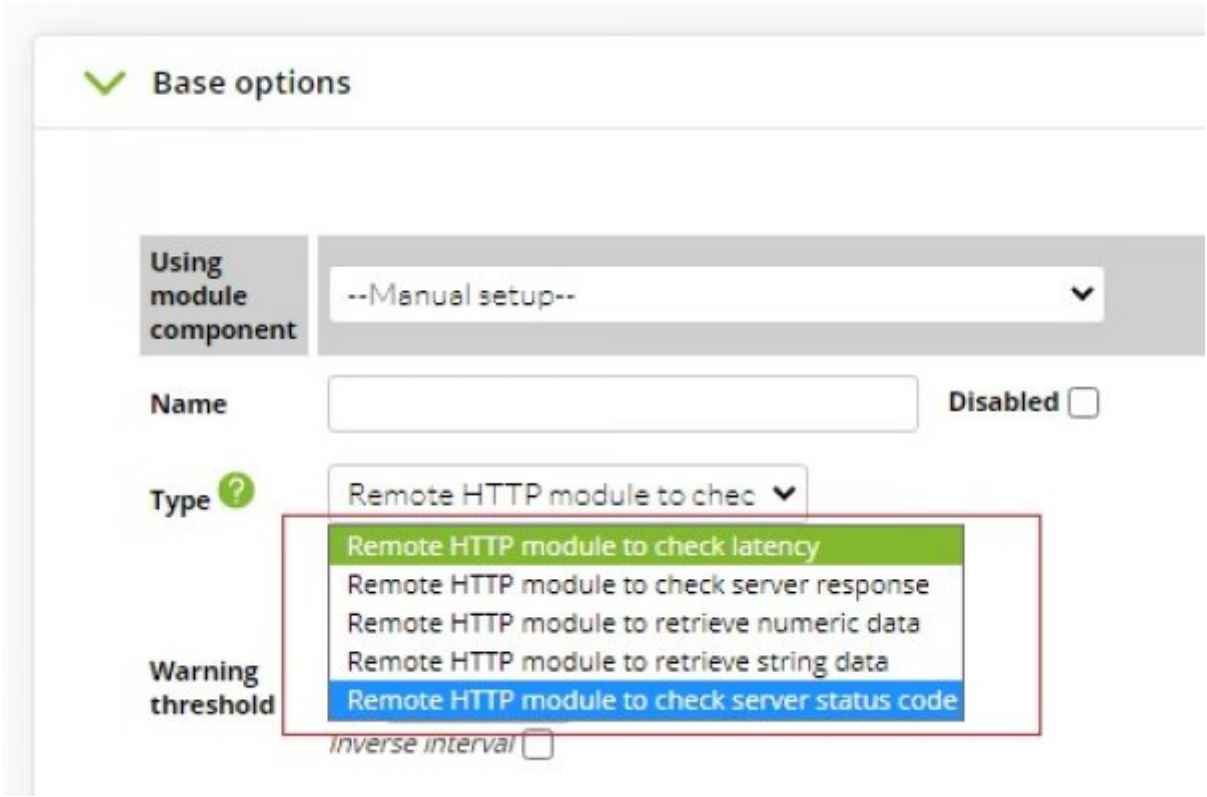
```
web_engine curl
```

ウェブモジュールの作成方法

ウェブページをモニタするには、まずはモジュールタブをクリックします。その後 'ウェブサーバモジュールの新規作成(Create a new webserver module)' を選択し、*作成(Create)* をクリックします。



*作成*をクリックすると、ウェブをモニタするために必要な設定を入力するためのフォームが表示されます。名前や、Webチェックのタイプなど、基本的なものです。



複数のチェックタイプが選択できます。

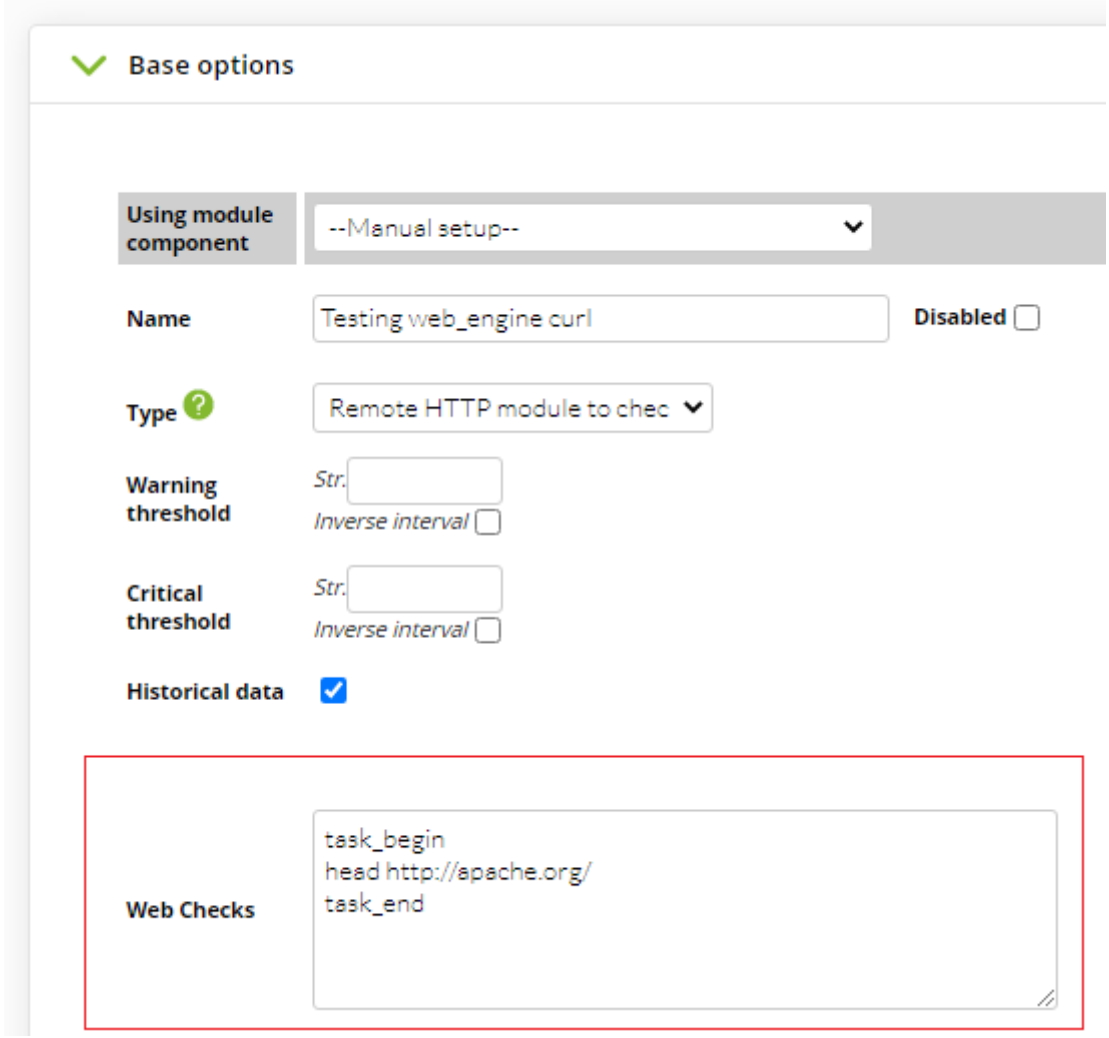
- **Remote HTTP module to check latency**: 最初のリクエストから最後のチェックが完了するまでのトータルの時間を取得します (ウェブチェックを完了するには、1つ以上のトランザクションがあります)。複数のリクエストが定義されている場合、それぞれの平均時間が利用されます。
- **Remote HTTP module to check server response**: すべてのトランザクションの結果をチェックし、1 (正常) もしくは、0 (異常) を返します。一部のステップが失敗すると、全体を障害として認識します。誤検出を避けるために、リトライ回数を設定することができます。 障害が発生した場合にテストを何回か実行する場合は、*リトライ*フィールドを使用してください(下記の詳細フィールドを参照)。
- **Remote HTTP module to retrieve numeric data**: 正規表現を利用して HTTP 応答から数値を取得します。
- **Remote HTTP module to retrieve string data**: 正規表現を利用して HTTP 応答から文字列を取得します。
- **Remote HTTP module to check server status code:**: `web_engine curl` の設定トークンで **curl** ツールの利用を有効化すると HTTPヘッダーを返すことができます。

ウェブチェック

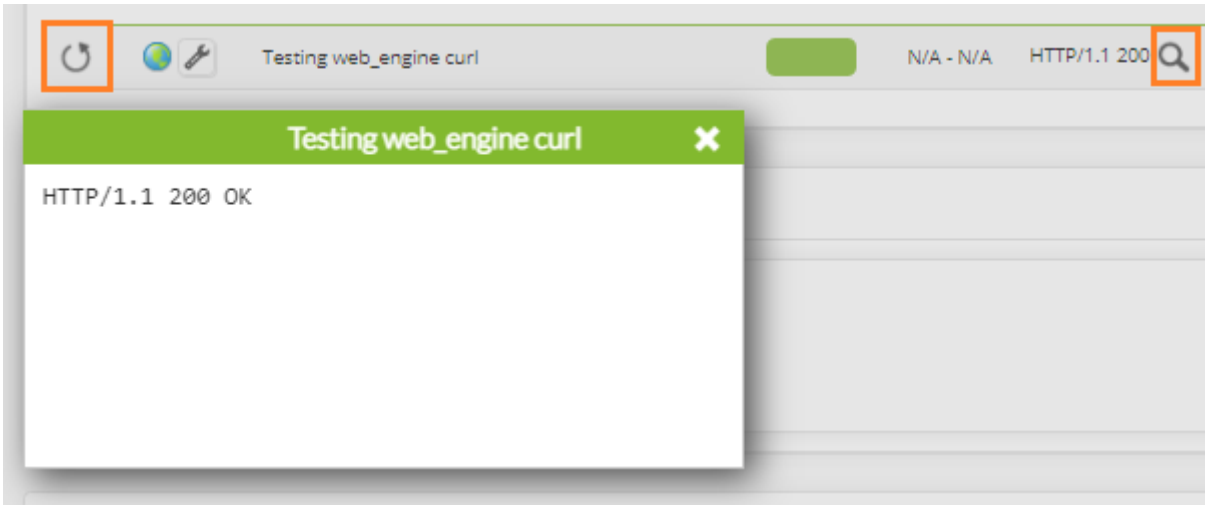
この必須フィールドは、実行される WEB チェックを定義します。 これは、1つ以上のステップまたは単一のリクエストで定義されます。 これらのリクエストは、Web 検査フィールドに特別な形式で設定しなければなりません。 チェックは *task_begin* で開始し、*task_end* で終了します。

簡単なトランザクションの例を以下に示します。

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



保存したのち、モジュールを強制実行して結果を見ることができます。



コマンドを追加した別の例:

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

```
check_string Apache Software Foundation
task_end
```

この基本的な例では、ウェブページに文字列があるかどうかをチェックしています。これは変数 **check_string** があるためです。 この変数では HTML 自体をチェックすることはできません。テキストのサブストリングのみを検索します。 私たちは Web サイト http://apache.org で “Apache Software Foundation” を探しています。 その文字列が存在する場合、チェックは OK を返します（サーバーの応答を確認するリモート HTTP モジュールの場合）

文字列が Web ページに存在しないことを確認するには 'check_not_string' を利用できます。

```
check_not_string Section 3
```

check_string 構文がとる引数は、通常のテキスト文字列ではなく、“ 正規表現 ” です。 つまり、文字列 “Pandora FMS (4.0)” の検索は正規表現で行う必要があります。 例えば、Pandora FMS \(4.0\) です。 これにより、強力な検索を行うことができますが、文字や数字以外の文字は \ でエスケープする必要があることに注意してください。

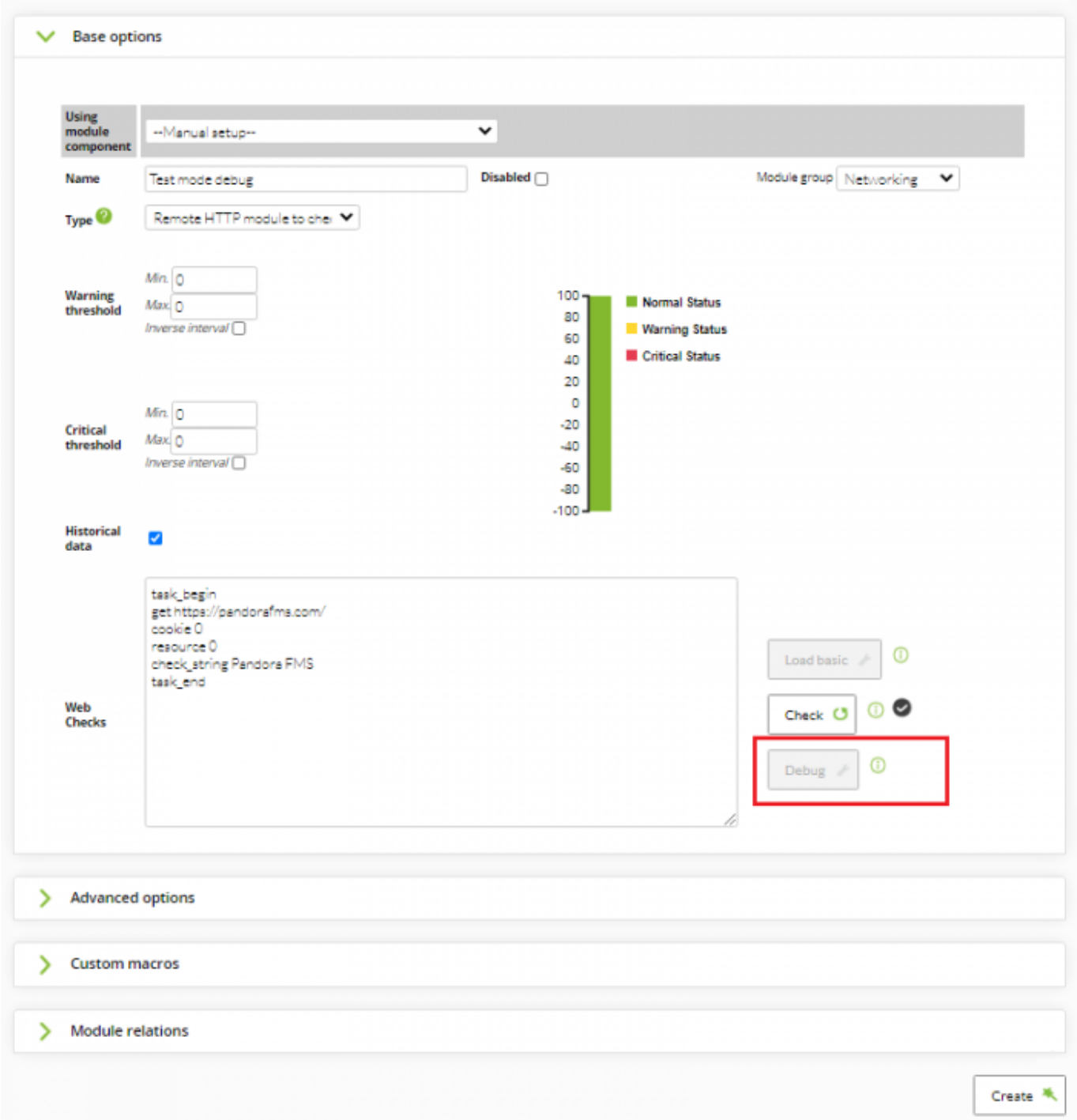
フォームをチェックするには、いくつかの拡張変数があります。

- **resource (1 または 0)**:ウェブリソース (画像、ビデオなど) のすべてをダウンロードします。
- **cookie (1 または 0)**:クッキーを保持し、以降のチェックのためにセッションを保持します。
- **variable_name**:フォームの変数名です。
- **variable_value**:上記変数名に対する値です。

これらの値を利用することにより、フォームにデータを送信し、正しく動作するかどうかをチェックすることができます。

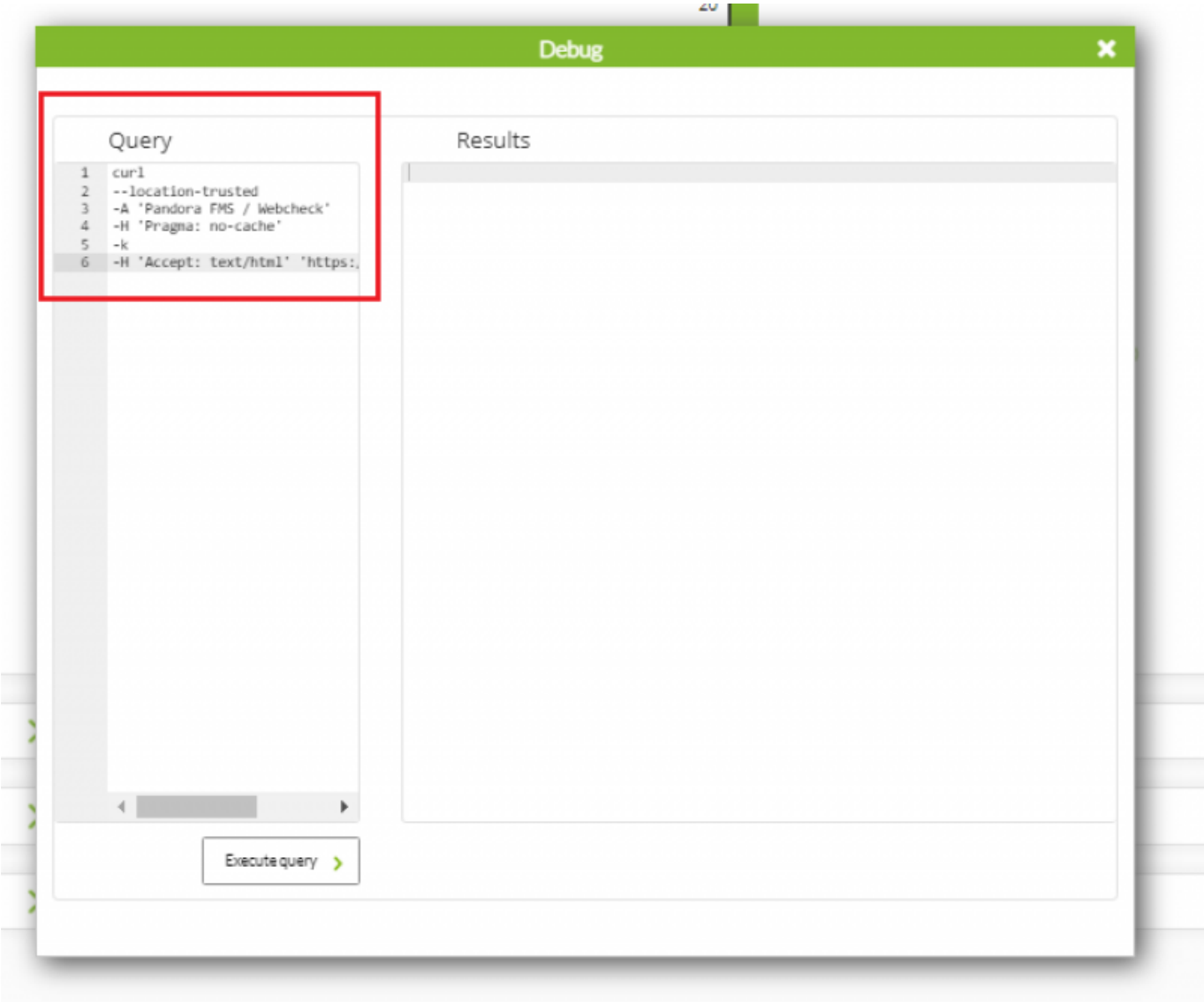
ドメインのリダイレクションには対応していません。この問題に対応するには、リダイレクトされた後にアクセスされるアドレスでモジュールを作成する必要があります。

前のケースでは、 **curl** コマンドのパラメータが短いバージョンでは -L であり、長いバージョンでは –location であるため HTTP 3XX リダイレクトを受けた場合、リダイレクトされたドメインに対して再度実行されます。 **ただし Pandora FMS の柔軟性により**、デバッグボタンを使用できます。

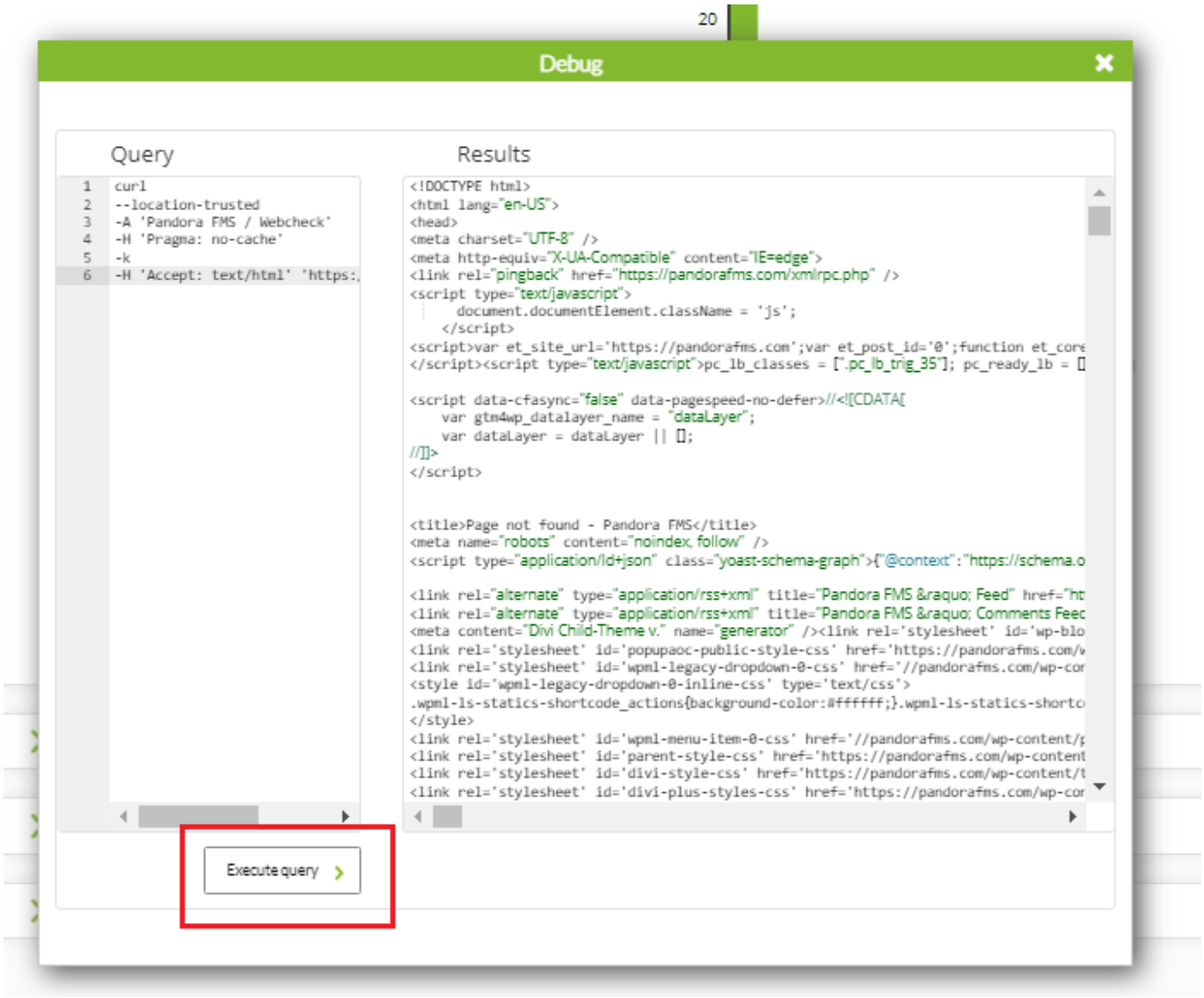


モジュールの作成時点では、モジュールは有効化されておらず、最初のチェックを実行した後モジュールを使用できるようになります。これを強制的に実行して時間を節約できます。

このモジュールを変更するときは、デバッグ(Debug) ボタンをクリックすると、デバッグモードに入って クエリ(Query) を編集できます。



モジュールのクエリを クエリの実行(Execute query) ボタンで実行したり、目的の結果が得られるまで他の値に変更して再実行したりできます。



ウェブの応答時間チェック

ウェブの応答時間をチェックしたい場合は、モジュールタイプ *Remote HTTP module to check latency* を選択する必要があるのみです。<https://pandorafms.com> の応答時間を知りたい場合、コードは次のようになります。

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

設定トークン `resource 1` を追加すると、すべてのリソース（JavaScript、CSS、イメージなど）をダウンロードし、それにかかった実際の時間を計算できるようになります。

ウェブサイトのダウンロード時間は、ブラウザで Web サイトを表示するのにかかる時間ではありません。通常、これはJavaScriptの読み込み時間に依存し、Goliat は JavaScriptをダウンロードしますが、実行しません。

プロキシ経由でのウェブチェック

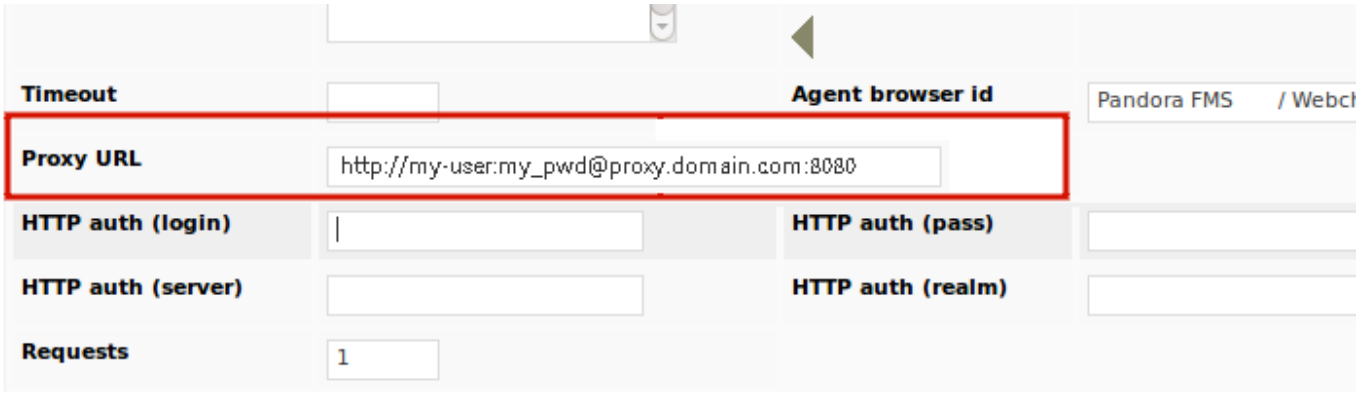
ウェブチェックは、プロキシ経由でも行うことができます。プロキシを設定するには、*拡張オプション(Advanced options)* をクリックすると表示される、*プロキシURL(Proxy URL)* フィールドにプロキシの URL を設定する必要があります。

例えばURL は次のようになります。

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

認証が必要なプロキシの場合は、次のように my-user にユーザ名、my_pwd にパスワードを指定します。

```
http://my_user:my_pwd@proxy.domain.com:8080
```

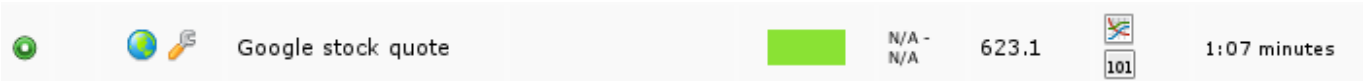


Webコンテンツの取得

特定の Web サイトが稼動しているか、どれくらい時間がかかっているかを知りたいわけではなくGoogle の株価などコンテンツの内容を確認したい場合もあります。そのためには、適切な正規表現で *Remote HTTP module to retrieve numeric data* モジュールを利用します。

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

出力は次のようになります。



また、より複雑な HTTP 応答からのデータを収集するための正規表現を設定トークン get_content_advanced で指定することもできます。

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

```
task_end
```

get_content_advanced に定義する正規表現は、カッコでくくらなければいけません。

警告または障害状態のしきい値を設定するには、モジュールの設定を使用して、受信した文字列が期待どおりのものであることを確認します。

ウェブページのフォームのチェック

より実用的な、Web フォームのチェックです。 しかし、これは単に Web ページ上のテキストをチェックするよりもはるかに複雑です。 このサンプルチェックでは Pandora 自身のコンソールを使用してログインし、ログインできたことを確認し、ログインしているユーザのデータが表示されているワークスペースのテキストを確認します。 デフォルトのコンソールであれば、管理者のユーザには “Admin Pandora” という記述が含まれています。

このタイプのチェックを実行するには、ログインに必要な資格情報が必要です。これらの値を使用して HTML フォームに「送信」するためです。また、ページに移動して HTML のソースから変数名を見る必要があります。どのように Goliat が動作するかを理解するためには HTML に関する最小限の知識が必要です。

複数ステップの WEB トランザクションテストを設定するときに、設定を確認する良い方法としては、ステップの 1つで何かが見逃された場合に備えて、ステップごとにテストすることです。

Pandora コンソールのログイン URL が以下であると仮定します。

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

HTML コードを確認すると、ログインフォームの変数は次の通りです。

- nick> ユーザ名
- pass> パスワード

フォームの認証を通すためには、変数 variable_name および variable_value の両方が必要です Pandora FMS コンソールのデフォルトは、admin および pandora です。

最初のステップはフォームへのアクセスです。次に、ユーザとパスワードを送り認証します。(認証の成功を 2つ目のステップで確認します)

```
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
```

上記の設定で、ウェブページにアクセスし認証することができます。これにより、認証した状態でのウェブページ上の何らかのチェックを実行できます cookie 1 トークンを使用して、前の手順で取得した cookie の永続性を維持します。 それらがなければ、セッションを再現することはできません。

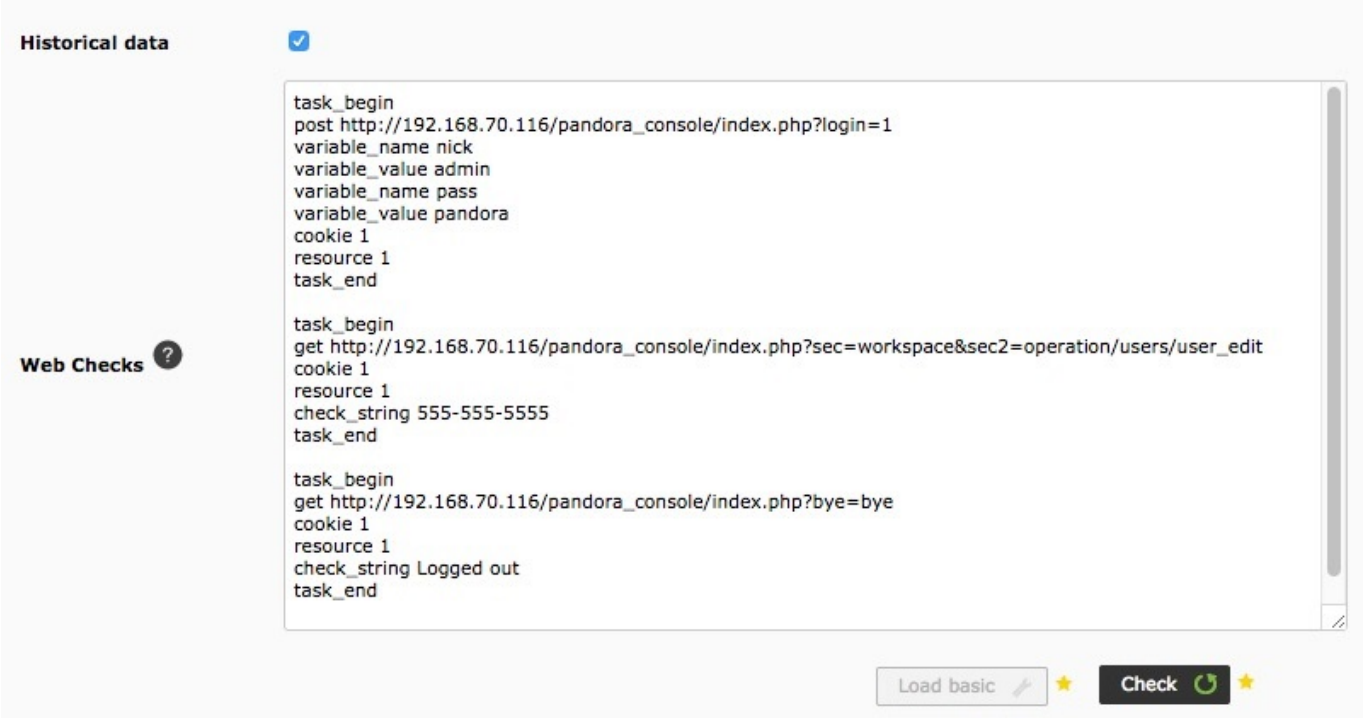
2つ目のステップでは、ユーザーの詳細ページにアクセスし電話番号を探します。ユーザ “admin” のデフォルトは、555-555-555 です。 コンソールに正しくログインできているかどうかがわかります。

```
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
```

最後にコンソールからログアウトし、ログアウトメッセージを探します。

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

Pandora FMS 上での全体の設定は次のようになります。



WEB リクエストの動作

拡張プロパティのフィールドは他のタイプのモジュールのフィールドと似ていますがWEB チェックではいくつかの異なるフィールドがあります。

タイムアウト(Timeout)

これはリクエストのタイムアウトです。この時間を超えるとリクエストは破棄されます。

エージェントブラウザID(Agent browser id)

これは、特定のページが一部の Web ブラウザのみを受け入れる場合に使用する Web ブラウザの識別子です。(詳細は、https://www.zytrax.com/tech/web/browser_ids.htm を参照してください)

リクエスト(Requests)

Pandora FMS は、このパラメータで示された回数だけチェックを繰り返します。 チェックの 1つが失敗した場合、障害とみなされます。モジュール内のチェックの数に応じて、一定数のページが取得されます。 つまり、モジュールが 3つのチェックで構成されている場合は、3ページがダウンロードされ、リクエストフィールドに値が設定されている場合は、ダウンロード数はその数を掛け合わせた数になります。 モジュールが処理を完了するのにかかる合計時間を把握するには、これを覚えておくことが重要です。

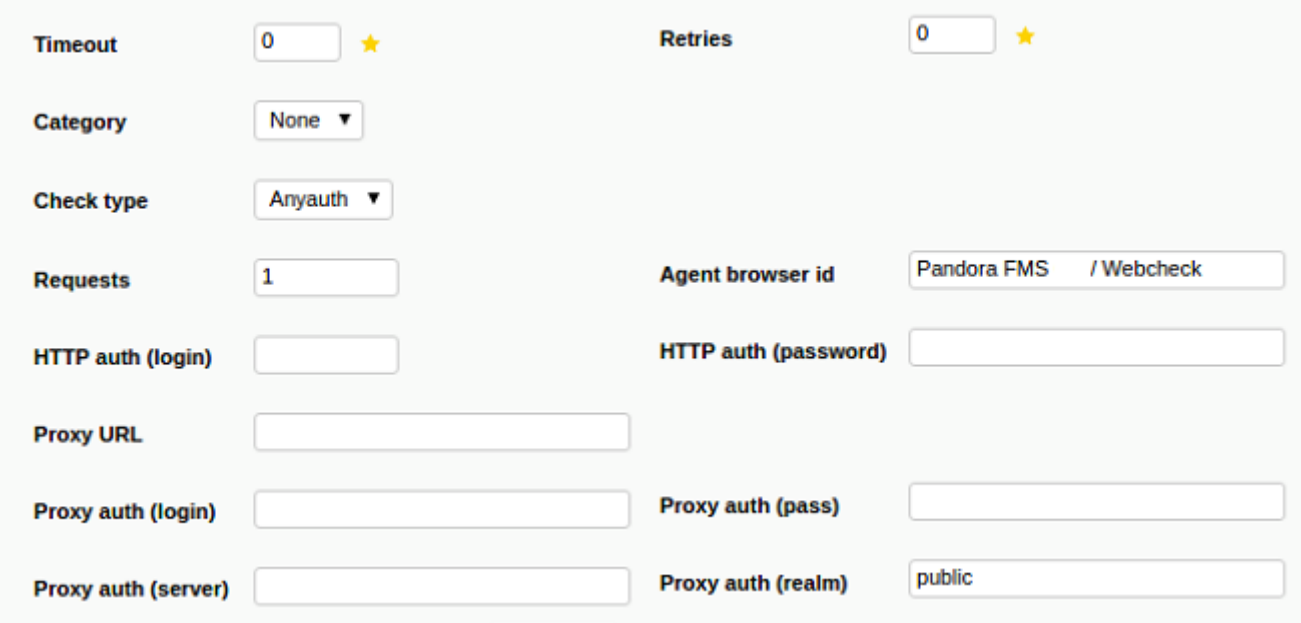
リトライ(Retries)

成功するまで **リクエスト(Request)** を実行する数です。例:

- リトライ = 2、リクエスト = 1: 最初のテストに失敗すると、もう一度実行し、2回目で成功すると、正常と判断します。
- リトライ = 1、リクエスト = 2: 2回のチェックを実行します。しかし一方の失敗で、障害と判断します。

HTTP の簡単な認証

いくつかのウェブページでは、HTTP 基本認証 を必要とします。通常、これは高速認証、高度なセキュリティチェック(暗号化、データ永続性など)へのアクセスを可能にする最小限のセキュリティとして使用されます。



(上記スクリーンショットのように)拡張オプションで設定することも、次の設定トークンを使用して WEB タスク定義で直接設定することもできます。

チェックタイプ(Check type)

HTTP サーバチェックタイプ

http認証(ログイン)(http auth (login))

ユーザ名

http認証(パスワード)(http auth (password))

パスワード

プロキシ認証レルム(Proxy auth realm)

認証レルム名

プロキシ認証(サーバ)(Proxy auth (server))

待ち受けているドメインと HTTP ポート

プロキシURL(Proxy URL)

プロキシサーバの URL

プロキシ認証(ログイン)(Proxy auth (login))

プロキシ接続ユーザ

プロキシ認証(パスワード)(Proxy auth (pass))

プロキシ接続パスワード

タスク全体の例:

```
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
```

http_auth_pass に指定するパスワードではクォーテーションには対応していません。シングルクォート ' の利用は避けてください。

WEB サービスおよび API モニタリング

Pandora FMS と Goliat webチェックにて、REST APIを監視することができます。ただしSOAP や XML-RPC を用いた API は監視できません。

例えば、動作しているときに数値(0からn)で返すような特定の Web API を監視したい場合、次のようなコードで、Pandora は何も応答が無い場合に障害と認識します。

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

実際の応答は次の通りです。

```
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
```

正規表現で出力を確認することにより、全体が正しく動作しているかを確認できます。より複雑な出力の場合は、それに合わせた正規表現を用います。データ部分だけでなく、応答内容全体をチェックすることに注意してください。そのためHTTP ヘッダーにもマッチさせることができます。

別の例:

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

この場合、モジュールが監視を実行できるようにするデータのタイプは、'Remote HTTP module to retrieve string data (web_content_string)' である必要があります。

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

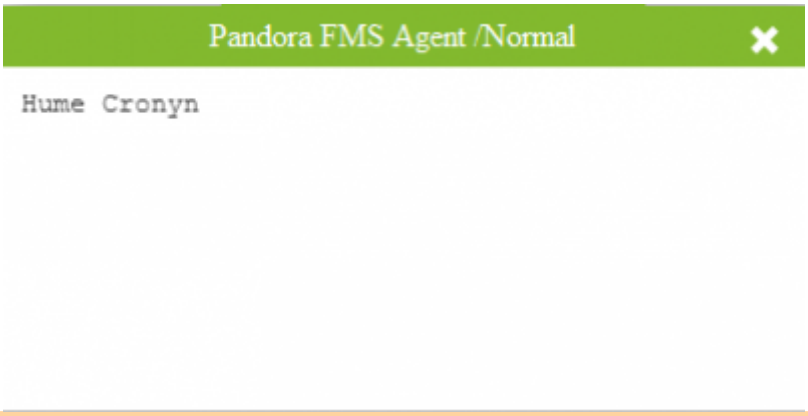
上記のモジュールと同様に、モジュールが正しく機能するには、定義されたデータのタイプが Remote HTTP module to retrieve string data (web_content_string)' である必要があります。

get_content_advanced でモジュールを作成することもできます。

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

この呼び出しは以下を返します。

Pandora FMS は、次のように結果を表示します。



呼び出しが正しく実行されるように、括弧内にキャプチャグループを適切に定義することが重要です。

API 呼び出しを作成するときは、宛先 API に呼び出しを許可する適切な権限があるかどうかを確認する必要があります。

HTTPS モニタリング

Goliat は HTTP と HTTPS の両方をチェックできます HTTPS を利用しているセキュリティで保護されたウェブサイトのチェックを行うには、その URL にプロトコルを組み込むだけです。

```
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=zpwhtygjntrz&ss=1&scc=1&ltmpl=default&ltmplcache=2
cookie 1
resource 0
check_string Google
task_end
```

拡張オプション

HTTP ヘッダーのカスタマイズ

header オプションで、HTTP ヘッダのカスタマイズしたり追加したりできます。たとえば、*Host* HTTP ヘッダーを変更するには次のようにします。

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

```
task_end
```

ウェブチェックのデバッグ

ウェブチェックをデバッグしたい場合は、debug <ログファイル> オプションを追加します。ログファイル.req および ログファイル.res というファイルが作成され、HTTP リクエストと応答が記録されます。たとえば次のようにします。

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

LWP の代わりの Curl の利用

LWP は、複数スレッドで HTTPS リクエストを実行するとクラッシュすることがあります(OpenSSL の制約による)。代替としては curl ツール を利用することです。この問題を解決するためには、*/etc/pandora/pandora_server.conf* を編集し、次の行を加えます。

```
web_engine curl
```

Pandora FMS サーバを再起動すると、ウェブチェックに LWP の代わりに Curl バイナリが利用されます。

高度なトランザクション監視

Goliath が提供する機能に加えて、Web トランザクション監視を実行する他の方法があります。一つは分散型(UX)で、アクセスできないネットワークであっても、サーバとは異なるシステムに“ エージェント ”として展開します。もう一つは集中化された方法(WUX)です。より詳細については ユーザエクスペリエンス監視 (UX および WUX) を参照してください。

Pandora FMS ドキュメント一覧に戻る

From:
https://pandorafms.com/manual/ - **Pandora FMS Documentation**

Permanent link:
https://pandorafms.com/manual/ja/documentation/03_monitoring/06_web_monitoring

Last update: **2021/11/05 12:05**