



Desarrollo de plugins de servidor



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

Permanent link:

https://pandorafms.com/manual!/current/es/documentation/pandorafms/technical_reference/05_anexo_server_plugins_development

2016/10 14:36



Desarrollo de plugins de servidor

Características básicas del plugin de servidor

El *plugin* de servidor es ejecutado por el Plugin Server de Pandora FMS, por lo que debe tener unas características especiales:

- Cada ejecución del *plugin* deberá devolver un único valor. Esto debe ser así ya que el Plugin Server realiza una ejecución por cada Módulo de tipo *plugin*.
- Debe poder acceder a los recursos a monitorizar de forma remota.
- Es posible usar cualquier lenguaje de programación que soporte el sistema operativo donde está instalado el servidor de Pandora FMS.
- Todas las dependencias o software necesario para ejecutar el *plugin* deberá estar disponible o ser instalado en la misma máquina que ejecuta el servidor de Pandora FMS.

Puede obtener más información acerca de la monitorización con *plugins* remotos de servidor en [este enlace](#). Allí obtendrá ejemplos más sencillos y cómo funcionan con los Módulos y los Agentes que los contienen. En este artículo se analizan en detalle la creación de los *plugin* de servidor.

Ejemplo de desarrollo plugin de servidor

A continuación se describe un ejemplo posible de *plugin* de servidor para Pandora FMS.

El siguiente *plugin* devuelve la suma del tráfico de entrada y salida de una interfaz de un dispositivo, los datos se obtienen por SNMP.

El código del *plugin* sería el siguiente:

```
#!/usr/bin/perl -w

use strict;
use warnings;

sub get_param($) {
    my $param = shift;
    my $value = undef;

    $param = "-" . $param;

    for(my $i=0; $i< $#ARGV; $i++) {

        if ($ARGV[$i] eq $param) {
            $value = $ARGV[$i+1];
            last;
        }
    }
}
```

```
    }

    }
    return $value;
}

sub usage () {
    print "iface_bandwidth.pl version v1r1\n";
    print "\nusage: $0 -ip <device_ip> -community <community> -ifname
<iface_name>\n";
    print "\nIMPORTANT: This plugin uses SNMP v1\n\n";
}

#Global variables
my $ip = get_param("ip");
my $community = get_param("community");
my $ifname = get_param("ifname");

if (!defined($ip) ||
    !defined($community) ||
    !defined($ifname) ) {
    usage();
    exit;
}

#Browse interface name
my $res = `snmpwalk -c $community -v1 $ip .1.3.6.1.2.1.2.2.1.2 -On`;

my $suffix = undef;

my @iface_list = split(/\n/, $res);

foreach my $line (@iface_list) {

    #Parse snmpwalk line
    if ($line =~ m/^(([\d|\.]*) = STRING: (.*)$/) {
        my $aux = $1;

        #Chec if this is the interface requested
        if ($2 eq $ifname) {

            my @suffix_array = split(/\./, $aux);

            #Get last number of OID
            $suffix = $suffix_array[$#suffix_array];
        }
    }
}

#Check if iface name was found
if (defined($suffix)) {
    #Get octets stats
```

```

    my $inoctets = `snmpget $ip -c $community -v1
.1.3.6.1.2.1.2.2.1.10.$suffix -OUevqt`;
    my $outoctets = `snmpget $ip -c $community -v1
.1.3.6.1.2.1.2.2.1.16.$suffix -OUevqt`;

    print $inoctets+$outoctets;
}

```

Una parte importante del código es la función `usage`:

```

sub usage () {
    print "iface_bandwidth.pl version v1r1\n";
    print "\nusage: $0 -ip <device_ip> -community <community> -ifname
<iface_name>\n";
    print "\nIMPORTANT: This plugin uses SNMP v1\n\n";
}

```

En esta función describe la versión y cómo usar el *plugin*, es muy importante y siempre se debe mostrar al ejecutar el *plugin* sin ningún tipo de parámetro o bien con una opción tipo `-h` o:

```
--help
```

Respecto al valor que devuelve el *plugin*, este se imprime en la salida estándar en la penúltima línea con la siguiente instrucción:

```
print $inoctets+$outoctets;
```

El valor devuelto por el *plugin* es un único dato, que luego el Plugin Server de Pandora FMS añadirá como dato al Módulo asociado.

Para poder ejecutar este *plugin* de servidor será necesario instalar los comandos `snmpwalk` y `snmpget` en la máquina que ejecuta el servidor de Pandora FMS.

Registro manual de un plugin en la consola

- Plugin type

La diferencia con PFMS estriba principalmente en que los *plugins* de Nagios devuelven un *error level* para indicar si la prueba ha tenido éxito o no. Si se necesita obtener un dato, no un estado (Bien/Mal), se puede utilizar un plugin de tipo Nagios en el modo "Standard" (para MS Windows® utilice [Nagios wrapper for agent plugin](#)).

- Max. timeout

Es el tiempo de expiración del complemento. Si no se recibe una respuesta en ese tiempo, se

marcará el Módulo como desconocido y no se actualizará su valor. Este es un factor muy importante a la hora de implementar monitorización con *plugins*, ya que si el tiempo que tarda en ejecutar el *plugin* es mayor que este número, nunca se podrá obtener valores con él. Este valor siempre debe ser mayor que el tiempo que tarde normalmente en devolver un valor el *script* o ejecutable usado como *plugin*. Si no se indica nada, se utilizará el valor indicado en la configuración como `plugin_timeout`.

- Plug-in command

Es la ruta donde está el comando del complemento. De forma predeterminada, si la instalación ha sido estándar, estarán en el directorio `/usr/share/pandora_server/util/plugin/`, aunque puede ser cualquier ruta del sistema. Para este caso, escribir `/usr/share/pandora_server/util/plugin/udp_nmap_plugin.sh` en el campo.

El servidor ejecutará ese *script*, por lo que éste debe tener permisos de acceso y de ejecución sobre él.

- Plug-in parameters

Una cadena con los parámetros del plugin, que irán tras el comando y un espacio en blanco. Este campo acepta macros tales como `_field1_ _field2_ ... _fieldN_`. En la siguiente sección se amplía el funcionamiento de las macros.















- Parameters macros

Es posible agregar macros ilimitadas para usarlas en el campo de los parámetros del plugin. Estas macros aparecerán como campos de texto en la configuración del Módulo. En la siguiente sección se amplía el funcionamiento de las macros.

Macros de plugin

Considere al DNS Plugin que viene instalado por defecto en un servidor Pandora FMS.

Plug-ins registered on Pandora FMS

Name	Type	Command	Op.
DNS Plugin	Standard	<code>/usr/share/pandora_server/util/plugin/dns_plugin.sh</code>	 
IPMI Plugin	Standard	<code>/usr/share/pandora_server/util/plugin/ipmi-plugin.pl</code>	 
MySQL Plugin	Standard	<code>/usr/share/pandora_server/util/plugin/mysql_plugin.sh</code>	  
Network bandwidth SNMP	Standard	<code>perl /usr/share/pandora_server/util/plugin/pandora_snmp_bandwidth.pl</code>	 
Packet Loss	Standard	<code>/usr/share/pandora_server/util/plugin/packet_loss.sh</code>	  
SMTP Check	Standard	<code>/usr/share/pandora_server/util/plugin/SMTP_check.pl</code>	 

Dicho complemento, **entre otros usos**, permite que para un dominio web y su correspondiente dirección IP ambos sean comprobados por un servidor de resolución de dominios (servidor DNS).

- `-i` : Dirección IP conocida del dominio.
- `-d` : Dominio web correspondiente.
- `-s` : Servidor DNS a consultar.

Conociendo estos tres parámetros proceda a **registrar manualmente** un nuevo *plugin*, coloque en el nombre "New DNS Plugin" y en la descripción copie el resumen anterior y además indique que es necesario que el Módulo recoja para la monitorización un valor falso (cero) o verdadero (uno). Este tipo de dato es también conocido como *booleano* y en Pandora FMS se denomina `generic_proc` .

En el apartado de parámetros de macro (Macro parameters) agregue tres campos `macro field1`, `field2` y `field3`.

Para `_field1_` coloque la descripción correspondiente al parámetro `-i` y así sucesivamente para los parámetros `-d` y `-s` . Deje los valores por defecto (default value) vacíos, escriba en la ayuda (Help) de cada uno el texto que usted considere conveniente.

En el cuadro de texto Plugin command escriba:

```
/usr/share/pandora_server/util/plugin/dns_plugin.sh
```

En el cuadro de texto Plugin parameters escriba:

```
-i _field1_ -d _field2_ -s _field3_
```

Cuando este *plugin* sea utilizado en un Módulo, el DNS Plugin será ejecutado sustituyendo cada uno de los campos por los parámetros que sean escritos en el Módulo.

```
/usr/share/pandora_server/util/plugin/dns_plugin.sh -i _field1_ -d _field2_ -s  
_field3_
```

Funcionamiento

De la misma manera que funcionan los `_field1_`, `_field2_`, (...), `_fieldN_` así también trabajan las macros, solo que albergarán valores especiales tanto de los Módulos como de los Agentes que contienen esos Módulos.

Retome el ejemplo de la sección anterior donde los valores por defecto de los `_fieldN_` fueron dejados vacíos. Edítelo y para el valor por defecto de `_field2_` coloque la macro `_module_`.

Si algún Módulo o componente se encuentra utilizando este *plugin* de servidor, aparecerá un icono de candado que no podrá ser actualizado.

Esta macro `_module_` devuelve el nombre del Módulo que utiliza el *plugin* y le será colocado al usuario o política al momento de crear dicho Módulo. Para comprobar esto, proceda a crear un nuevo Agente llamado "DNS verify" y agregue un nuevo Módulo por medio de la opción Create a new plugin server module.

Una vez entre en el formulario de edición del nuevo Módulo, en Plugin seleccione de la lista "New DNS Plugin" y la macro `_module_` aparecerá de la siguiente manera:

Type ?

Warning threshold
 Min.
 Max.
 Inverse interval

Critical threshold
 Min.
 Max.
 Inverse interval

Historical data

Plugin ? This plugin is used to check if a specific domain return a specific IP address.

IP

Domain

Server DNS

Recuerde colocar que el tipo de dato a recoger es “Generic boolean” y para el nombre del nuevo Módulo simplemente coloque el dominio web al cual va a verificar su correspondiente dirección IP ante un servidor DNS especificado. El `token critical_on_error` del servidor PFMS está configurado por defecto para que **los módulos en estado desconocido pasen a estado crítico**. Guarde y compruebe el funcionamiento.

La ventaja de este método es que contará con tantos Módulos como dominios web necesite verificar y serán identificados fácilmente en diferentes componentes (Dashboards, informes, etc.) por medio de su nombre.

Lista de macros

- `_agent_`: Alias del Agente a utilizar en el macro. Si no tiene asignado alias, se usará el nombre del Agente.
- `_agentalias_`: Alias del Agente a utilizar en el macro.
- `_agentdescription_`: Descripción del Agente a utilizar en el macro.
- `_agentstatus_`: Estado actual del Agente cuando es utilizado en el macro.
- `_agentgroup_`: Nombre del grupo del Agente a utilizar el macro.
- `_agentname_`: Nombre del Agente a utilizar en el macro (ver también `_agent_`).
- `_address_`: Dirección del Agente a utilizar el macro.
- `_module_`: Nombre del Módulo a utilizar el macro.
- `_modulegroup_`: Nombre del grupo del Módulo a utilizar el macro.
- `_moduledescription_`: Descripción del Módulo a utilizar el macro.
- `_modulestatus_`: Estado del Módulo a utilizar el macro.

- `_moduletags_`: URL asociadas a los *tags* de módulos.
- `_id_module_`: Identificador del Módulo a utilizar el macro.
- `_id_agente_`: Identificador del Agente a utilizar el macro, útil para construir URL de acceso a la Consola de Pandora FMS.
- `_id_group`: Identificador del grupo de Agente a utilizar el macro.
- `_interval_`: Intervalo de la ejecución del Módulo a utilizar el macro.
- `_target_ip_`: Dirección IP del objetivo del Módulo a utilizar el macro.
- `_target_port_`: Puerto del objetivo del Módulo a utilizar el macro.
- `_policy_`: Nombre de la política a la que pertenece el Módulo (si aplica).
- `_plugin_parameters_`: Parámetros del *plugin* del Módulo a utilizar el macro.
- `_email_tag_`: Buzones de correo electrónico asociados a las etiquetas (*tags*) de Módulos.
- `_phone_tag_`: Teléfonos asociados a los *tags* de módulos.
- `_name_tag_`: Nombre de los *tags* asociados al Módulo a utilizar el macro.

Empaquetado en PSPZ

El Plugin Zipfile (.pspz) del Servidor de Pandora

Existe una manera de registrar *plugins* y Módulos que usan el nuevo *plugin* (como por ejemplo la biblioteca de módulos que depende del *plugin*). Es, básicamente, una extensión de administrador para subir un fichero en formato `.pspz`, descrito en detalle en las secciones siguientes a esta. El sistema lee el fichero, desempaqueta e instala los binarios y/o *scripts* en el sistema. Además, registra el *plugin* y crea todos los módulos definidos en el `.pspz` en la biblioteca de módulos de Pandora FMS (componentes de red).

Esta sección describe cómo crear un fichero `.pspz`.

Package File

Un `.pspz` es un fichero comprimido en formato zip con dos filas:

`plugin_definition.ini`: contiene la especificación del *plugin* y de los módulos. Deberá tener este nombre exactamente (es *case sensitive*, diferencia mayúsculas de minúsculas).

`<script_file>`: es el *plugin script/binary* (guión de comandos o código binario ejecutable) en sí mismo. Podrá tener cualquier nombre válido. Un ejemplo de un fichero `.pspz` (a su vez comprimido como `.zip` para poder incluir su documentación) puede ser descargado [en este enlace](#).

Estructura de plugin_definition.ini

Cabecera/Definición

Este es un fichero INI clásico con secciones opcionales. La primera sección, que es la más importante, es una sección de nombre fijo llamada `plugin_definition`. Este es un ejemplo:

```
[plugin_definition]
name = Remote SSH exec
filename = ssh_pandoraplugin.sh
description = This plugin execute remotely any command provided
timeout = 20
execution_command =
execution_postcommand =
ip_opt = -h
user_opt = -u
port_opt =
pass_opt =
plugin_type = 0
total_modules_provided = 1
```

`filename`: Debería tener el mismo nombre que el script incluido en el fichero `.pspz`, nombrado antes como `<script_file>`. En este ejemplo es un *shell script* (formato `.sh`) llamado `ssh_pandoraplugin.sh`.

`*_opt`: Aquí están las opciones de registro del *plugin*, mostradas en el formulario para registrar “manualmente” el plugin en la consola de Pandora FMS.

`plugin_type`: 0 para un *plugin* estándar de Pandora FMS, y 1 para un *plugin* tipo Nagios.

`total_modules_provided`: Especifica cuantos módulos se definen en las secciones siguientes del fichero `.ini`. Debe establecer uno como mínimo (para usar en un ejemplo como mínimo).

`execution_command`: Si se emplea, hay que ponerlo delante del *script*. Podrá ser un intérprete, como por ejemplo `java -jar`. Así pues, el *plugin* se llamará a ejecución, desde el Plugin Server de Pandora FMS, con el siguiente código: `java -jar <plugin_path>/<plugin_filename>`.

`execution_postcommand`: Si se emplea, define los parámetros adicionales transmitidos al *plugin* después del `<plugin_filename>`, que es invisible para el usuario.

Definición del Módulo / Componentes de Red

Debe definir aquí el mismo número de módulos que los definidos en `total_modules_provided` de la sección anterior.

Si tiene por ejemplo cuatro módulos, los nombres de las secciones deberán ser: `module1`, `module2`, `module3` y `module4`.

Este es un ejemplo de una definición de Módulo:

```
[module1]
name = Load Average 1Min
description = Get load average from command uptime
id_group = 12
type = 1
max = 0
min = 0
module_interval = 300
id_module_group = 4
id_modulo = 4
plugin_user = root
plugin_pass =
plugin_parameter = "uptime | awk '{ print $10 }' | tr -d ','"
max_timeout = 20
history_data = 1
min_warning = 2
min_critical = 5
str_warning = "danger"
min_critical = "alert"
min_ff_event = 0
tcp_port = 0
critical_inverse = 0
warning_inverse = 0
critical_instructions = "Call the department head"
warning_instructions = "Call the server manager to reduce the load."
unknown_instructions = "Verify that the Pandora FMS agent is running"
```

Algunas cosas debe tener en cuenta:

- Todos los campos *deben* estar definidos. Si no dispone de datos, déjelo en blanco; fíjese usted en el campo `plugin_pass` del ejemplo anterior.
- Utilice dobles comillas por pares “...” para definir valores que contengan caracteres especiales o espacios, como el campo `plugin_parameter` del ejemplo de arriba. Los ficheros INI que contengan caracteres como ' / - _ () [] y otros, *deben* tener dobles comillas. Intente evitar el uso del carácter " para los datos. Si debe emplearlo, *escape* con la combinación \".
- Si tiene dudas sobre el propósito o significado de estos campos, puede revisar `tnetwork_component` en la base de datos de Pandora FMS, ya que esta tiene casi los mismos campos. Cuando se crea un nuevo componente de red, este es almacenado en esta base de datos. Intente crear un componente de red que utilice su *plugin* y analice el registro de entrada en esa tabla para entender todos los valores.
- `id_module`: Deberá ser siempre 4 (esto significa que es un Módulo *plugin*).
- `type`: Define que tipo de Módulo es: `generic_data` (1), `generic_proc` (2), `generic_data_string` (3) o `generic_data_inc` (4) tal y como fueron definidos en `ttipo_modulo`.
- `id_group`: Es la PK (*primary key*) de la tabla grupo que contiene definiciones de grupo. El grupo 1 es “todos los grupos” (“All”), y actúa como un grupo especial.
- `id_module_group`: Procede de la tabla `tmodule_group`. Es una asociación de módulo por funcionalidad, puramente descriptiva. Puede usar 1 para el General module group.

Versión 2

A partir de Pandora FMS v5.1.SP1, los plugins de servidor utilizan macros.

Se diferenciarán estos plugins por la extensión del fichero pspz2. Si no tiene esta extensión se asumirá una PSPZ de tipo 1 (sin macros dinámicas en la extensión de *plugin* remoto)

Además `plugin_definition.ini` ha cambiado. Se han añadido los siguientes campos:

En la sección `plugin_definition`:

- `total_macros_provided` que define el numero de macros dinámicas que tiene el *plugin*.

En la sección `module<N>`:

- `macro_<N>_value` que define el valor para ese Módulo usando esa macro dinámica, si no existe se toma el valor por defecto.

Y debe crear una sección por cada macro dinámica, por ejemplo:

```
[macro_<N>]
hide = 0
description = <your_description>
help = <text_help>
value = <your_value>
```

- Se debe llamar en la sección `execution_postcommand` a los macros para que se realice la substitución ([ver ejemplo](#)).
- La versión anterior sigue siendo compatible. Si el parámetro versión no está definido, se asume que la versión es 1.

Ejemplo

Definición de un plugin v2 (.pspz2) con propósitos didácticos:

```
[plugin_definition]
name = PacketLoss
filename = packet_loss.sh
description = "Measure packet loss in the network in %"
timeout = 20
ip_opt =
execution_command =
```

```
execution_postcommand =
parameters = _field1_ _field2_
user_opt =
port_opt =
pass_opt =
plugin_type = 0
total_modules_provided = 1
total_macros_provided = 2

[macro_1]
hide = 0
description = Timeout
help = Timeout in seconds
value = 5

[macro_2]
hide = 0
description = Target IP
help = IP address
value = 127.0.0.1

[module1]
name = Packet loss
description = "Measure target packet loss in % "
id_group = 10
type = 1
max = 0
min = 0
module_interval = 300
id_module_group = 2
id_modulo = 4
max_timeout = 20
history_data = 1
min_warning = 30
min_critical = 40
min_ff_event = 0
tcp_port = 0
macro_1_value = 5
macro_2_value = localhost
unit = %
```

[Volver al índice de documentación de Pandora FMS](#)