# So you want to do some logging. . . (PT. 6 IIS Logs)

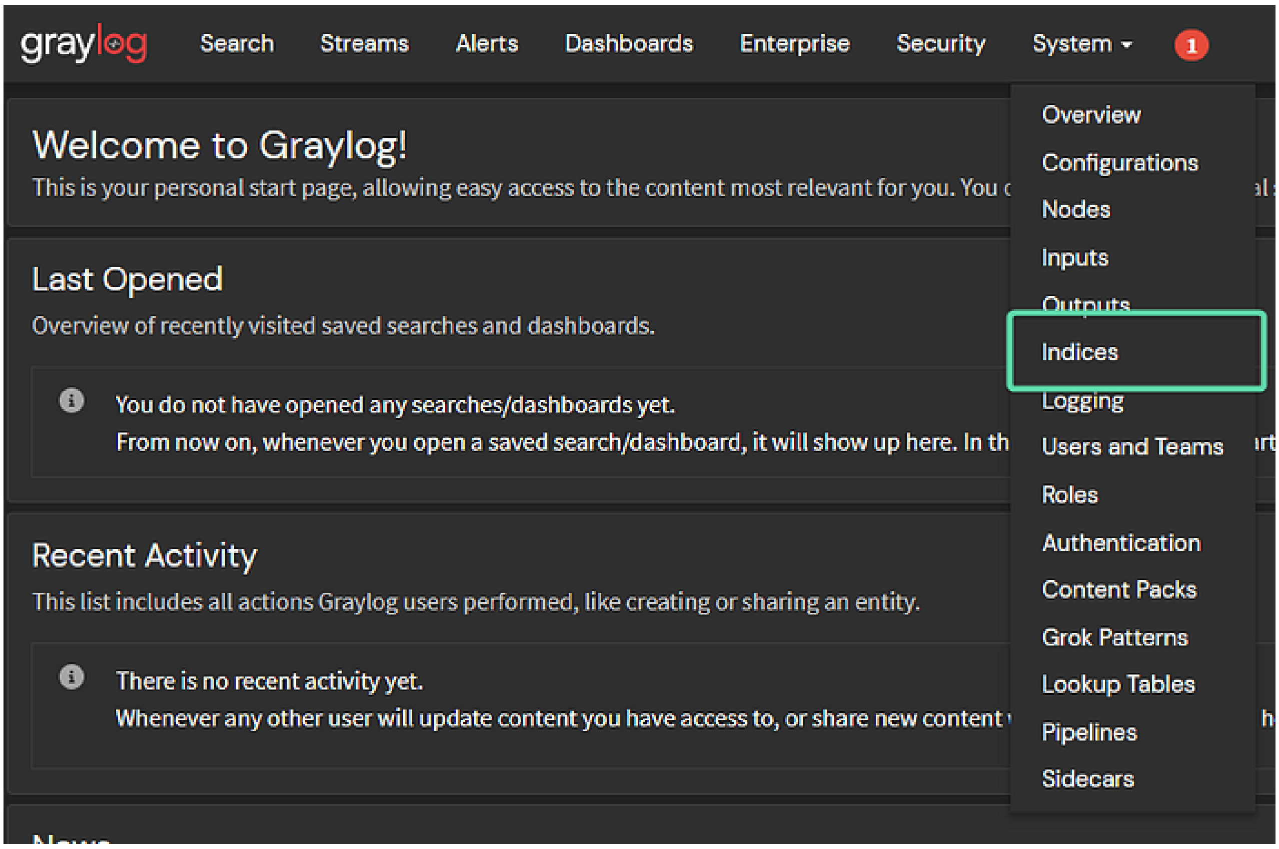
blog.iso365down.com/so-you-want-to-do-some-logging-pt-6-iis-logs-52f70819567a

HanSolo71 December 26, 2023

## Creating a New Index

Before moving forward we will want to create a new index in Graylog. Logs from web servers can be generate a lot of traffic and you may want to change retention based on storage needs.

To make a new index set, use the top menu and select **System > Indices.**



Welcome to Graylog!

This is your personal start page, allowing easy access to the content most relevant for you. You c

Last Opened

Overview of recently visited saved searches and dashboards.

You do not have opened any searches/dashboards yet.
From now on, whenever you open a saved search/dashboard, it will show up here. In th

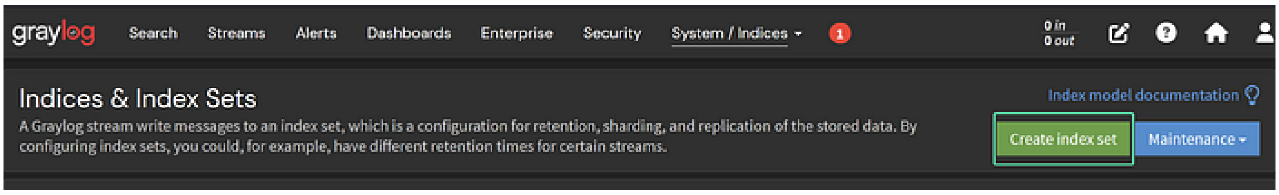
Recent Activity

This list includes all actions Graylog users performed, like creating or sharing an entity.

There is no recent activity yet.
Whenever any other user will update content you have access to, or share new content

Time for a new index!

Then select **Create index set.**

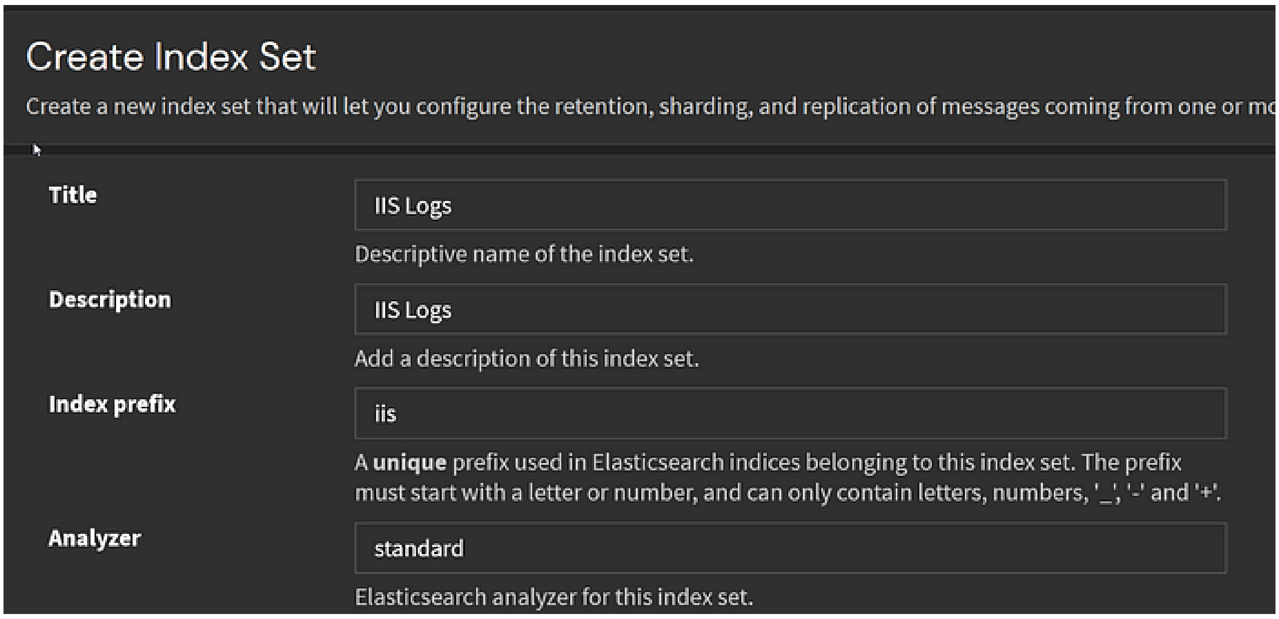


Indices & Index Sets

A Graylog stream write messages to an index set, which is a configuration for retention, sharding, and replication of the stored data. By configuring index sets, you could, for example, have different retention times for certain streams.

Make the new index

We will only need to worry about *Title* and *Index prefix* fields along with rotation strategy.



Create Index Set

Create a new index set that will let you configure the retention, sharding, and replication of messages coming from one or mo

**Title**: IIS Logs
Descriptive name of the index set.

**Description**: IIS Logs
Add a description of this index set.

**Index prefix**: iis
A **unique** prefix used in Elasticsearch indices belonging to this index set. The prefix must start with a letter or number, and can only contain letters, numbers, '_', '-' and '+'.

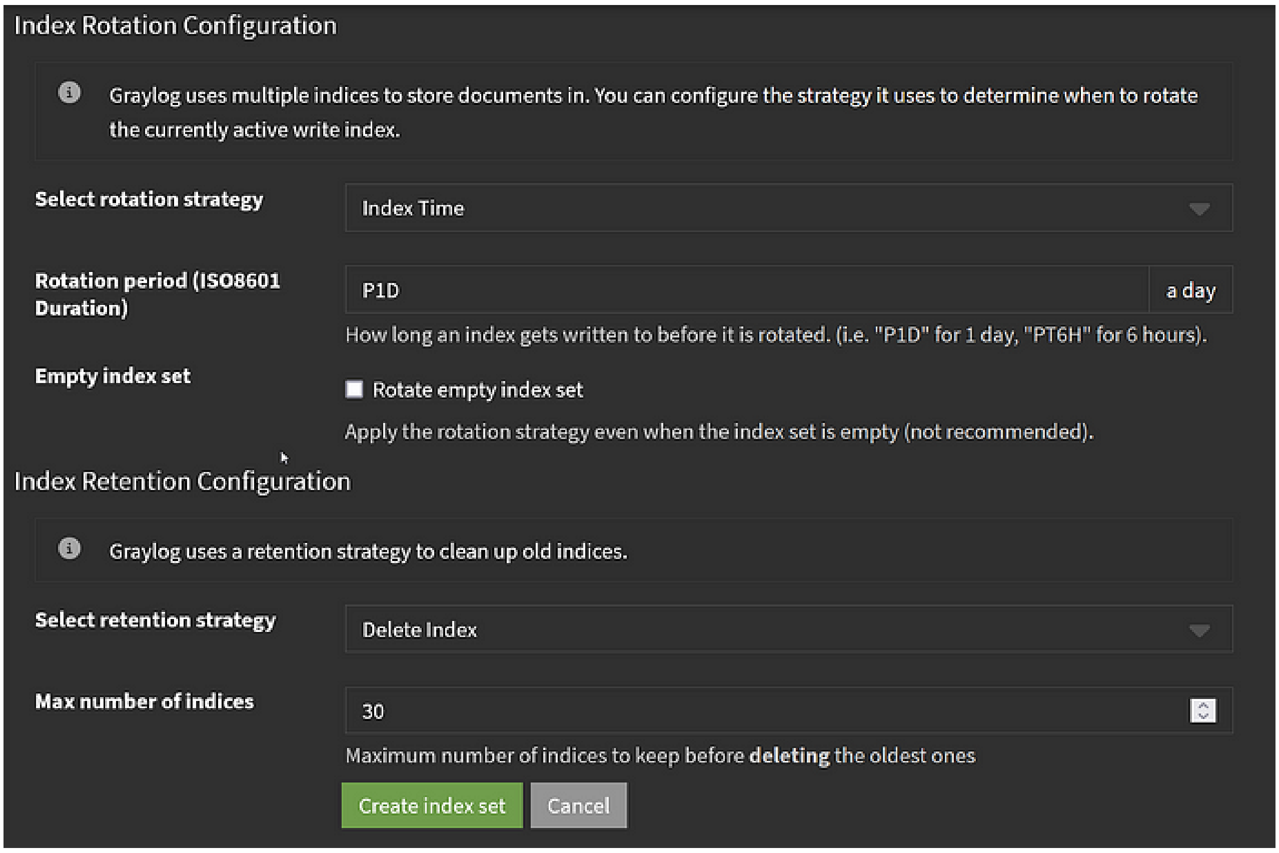
**Analyzer**: standard
Elasticsearch analyzer for this index set.

Name the new index set

We want to keep our IIS/Web access logs for 30 days.

To do this, we will use a rotation strategy of **Index Time** with a duration of **P1D**. We want to set the retention strategy to **Delete Index** and set the max number of indices to 30.

These setting will create a new index every day and delete the oldest index when index 31 is created



Index Rotation Configuration

Graylog uses multiple indices to store documents in. You can configure the strategy it uses to determine when to rotate the currently active write index.

**Select rotation strategy**: Index Time

**Rotation period (ISO8601 Duration)**: P1D a day
How long an index gets written to before it is rotated. (i.e. "P1D" for 1 day, "PT6H" for 6 hours).

**Empty index set**: ☐ Rotate empty index set
Apply the rotation strategy even when the index set is empty (not recommended).

Index Retention Configuration

Graylog uses a retention strategy to clean up old indices.

**Select retention strategy**: Delete Index

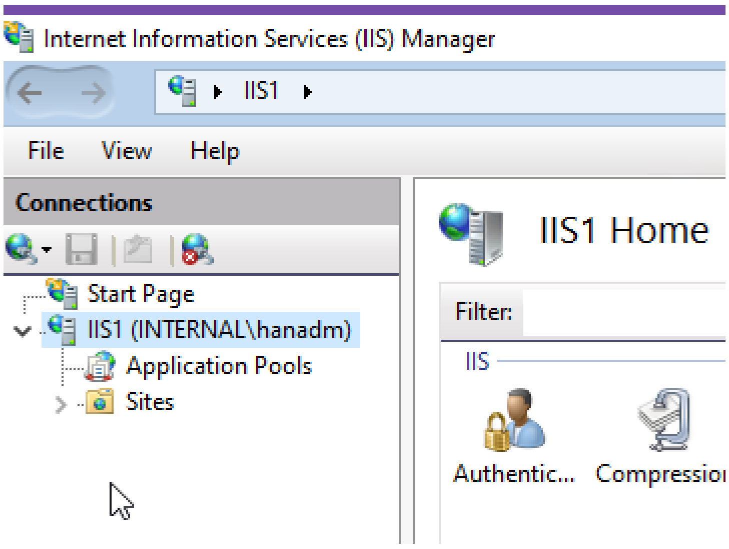
**Max number of indices**: 30
Maximum number of indices to keep before **deleting** the oldest ones

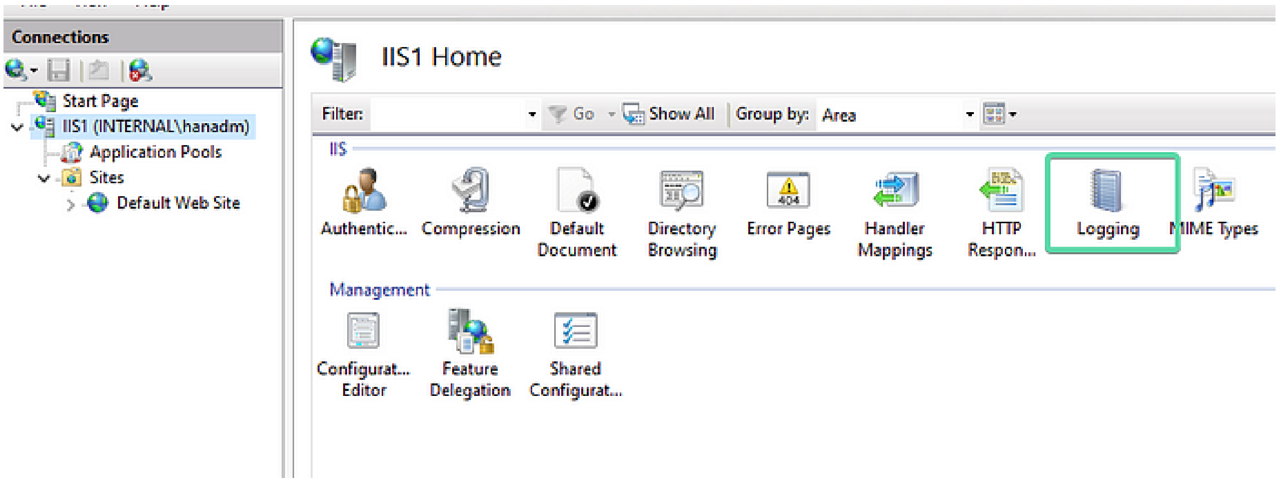
These setting will create a new index every day and delete the oldest index when index 31 is created

## Setting up IIS

Before moving forward we will want to enable all logging fields and ensure we are using W3C logs. We will start by opening the **IIS Manager.**

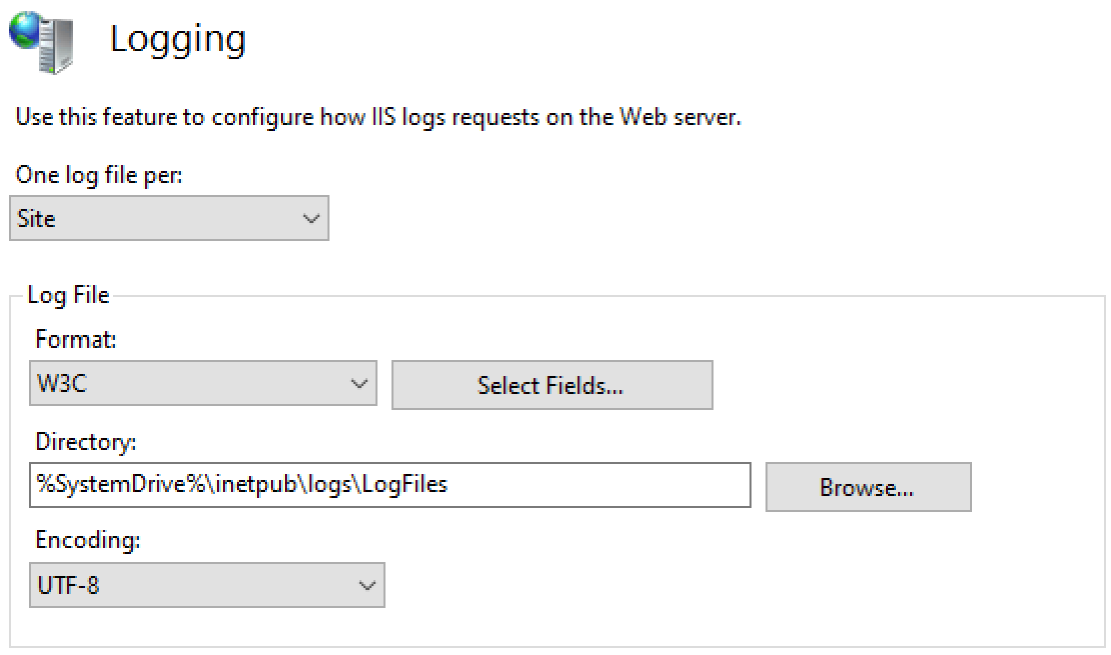
Select the IIS Servers Logging Rules





Edit logging

Ensure you are using W3C format also.



Logging

Use this feature to configure how IIS logs requests on the Web server.

One log file per: Site

Log File

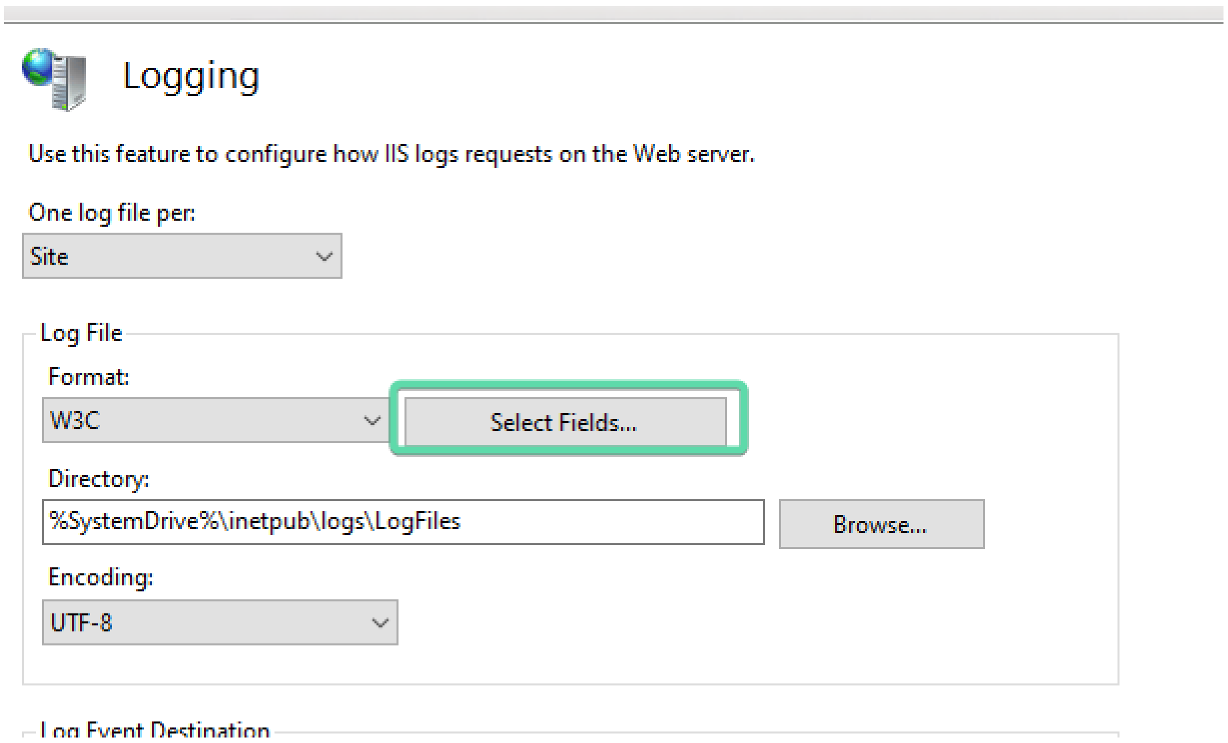
Format: W3C

Directory: %SystemDrive%\inetpub\logs\LogFiles

Encoding: UTF-8

Ensure logs are W3C format

Lastly enable all fields



Logging

Use this feature to configure how IIS logs requests on the Web server.

One log file per: Site

Log File

Format: W3C

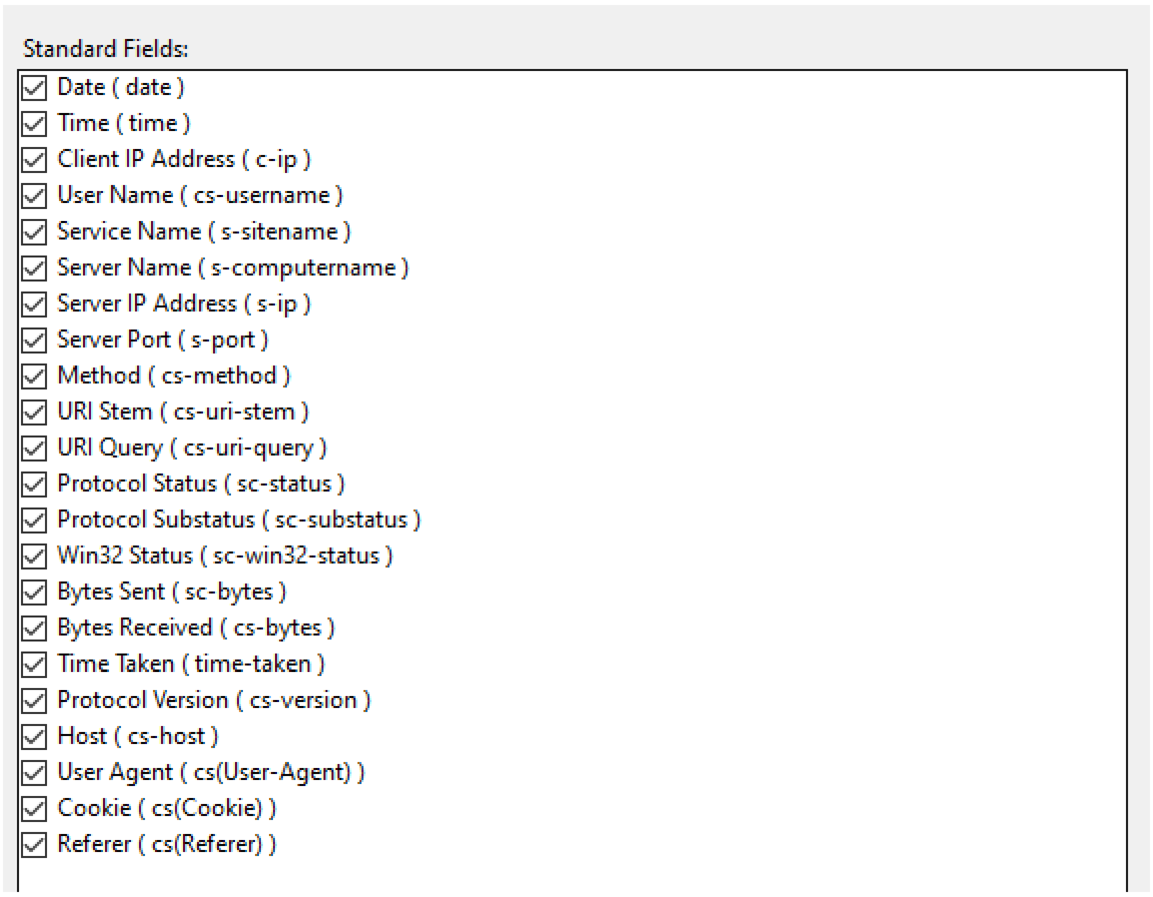
Directory: %SystemDrive%\inetpub\logs\LogFiles

Encoding: UTF-8

Log Event Destination

Edit enabled Fields

Go ahead and enable all standard fields


Standard Fields:

- ☑ Date ( date )
- ☑ Time ( time )
- ☑ Client IP Address ( c-ip )
- ☑ User Name ( cs-username )
- ☑ Service Name ( s-sitename )
- ☑ Server Name ( s-computername )
- ☑ Server IP Address ( s-ip )
- ☑ Server Port ( s-port )
- ☑ Method ( cs-method )
- ☑ URI Stem ( cs-uri-stem )
- ☑ URI Query ( cs-uri-query )
- ☑ Protocol Status ( sc-status )
- ☑ Protocol Substatus ( sc-substatus )
- ☑ Win32 Status ( sc-win32-status )
- ☑ Bytes Sent ( sc-bytes )
- ☑ Bytes Received ( cs-bytes )
- ☑ Time Taken ( time-taken )
- ☑ Protocol Version ( cs-version )
- ☑ Host ( cs-host )
- ☑ User Agent ( cs(User-Agent) )
- ☑ Cookie ( cs(Cookie) )
- ☑ Referer ( cs(Referer) )

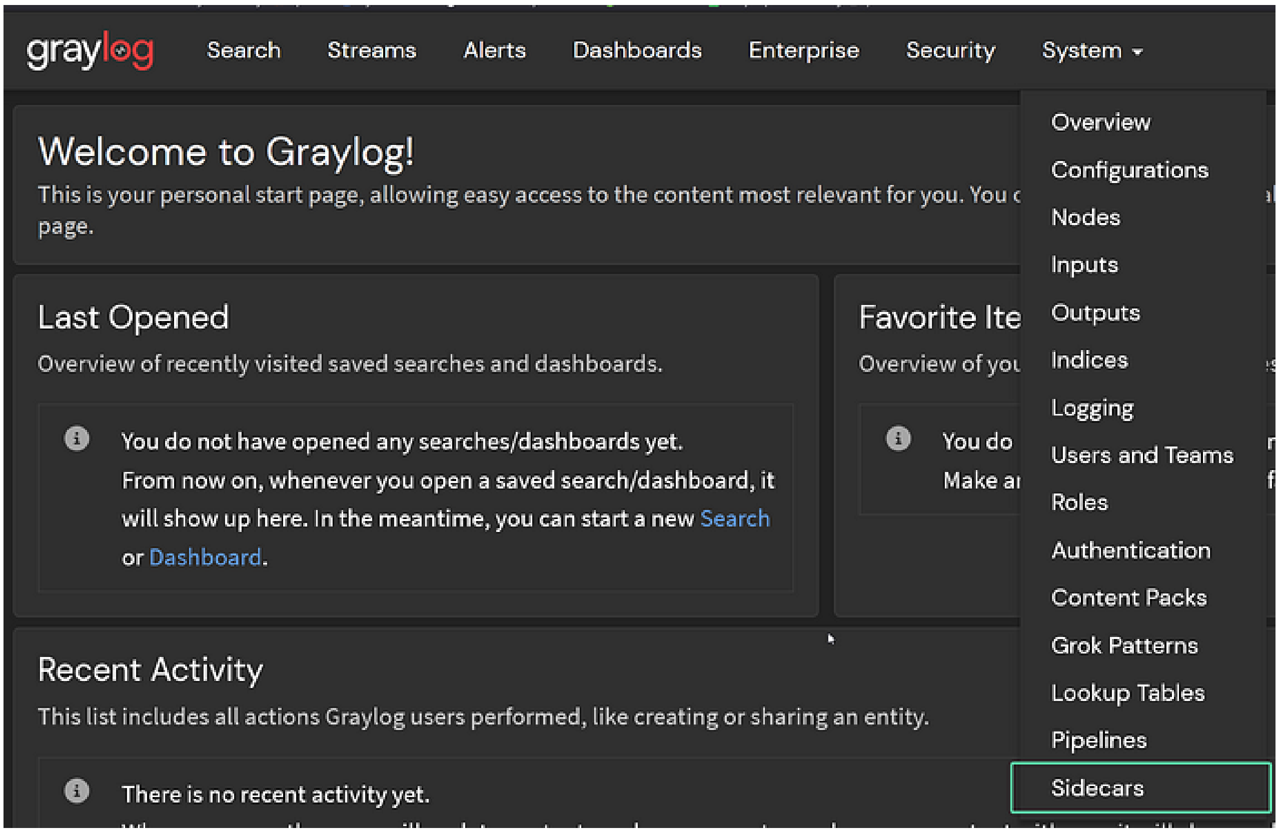
Enable all standard fields

If you have custom fields you will want to take note of them and add them to our GROK pattern latter.

## Setting up the Sidecar Configuration

With our index ready to receive those logs now we need to configure the Graylog Sidecar that will be installed to read our IIS logs.

Using the top menu select **System > Sidecars.**



Welcome to Graylog!

This is your personal start page, allowing easy access to the content most relevant for you. You c
page.

Last Opened

Overview of recently visited saved searches and dashboards.

You do not have opened any searches/dashboards yet.
From now on, whenever you open a saved search/dashboard, it will show up here. In the meantime, you can start a new Search or Dashboard.

Favorite Ite
Overview of you

You do
Make a

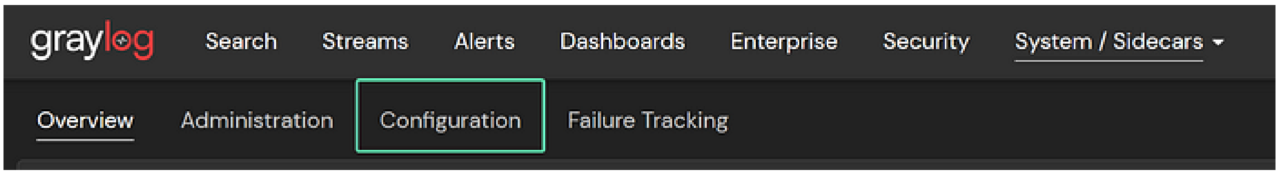
Recent Activity

This list includes all actions Graylog users performed, like creating or sharing an entity.

There is no recent activity yet.

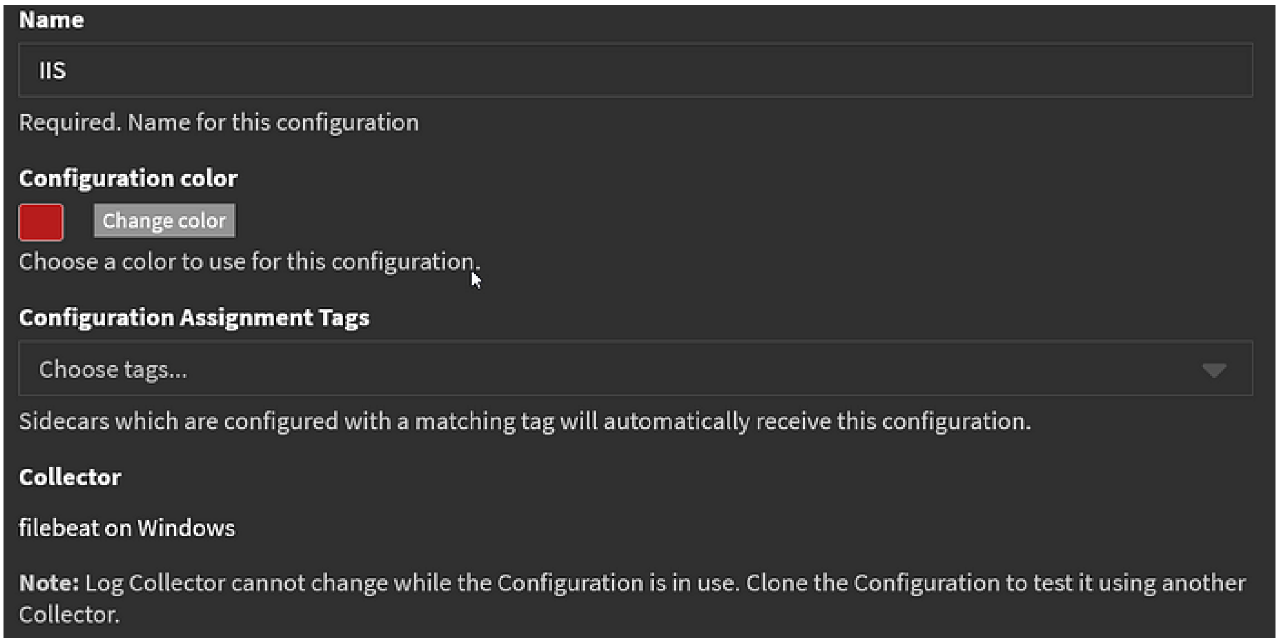
System > Sidecars

Select **configuration** at the from the sidecar menu.



Select Configuration

Create a configuration. We will be creating a **filebeat collector for windows.**



**Name**

IIS

Required. Name for this configuration

**Configuration color**

🟥 Change color

Choose a color to use for this configuration.

**Configuration Assignment Tags**

Choose tags...

Sidecars which are configured with a matching tag will automatically receive this configuration.

**Collector**

filebeat on Windows

**Note:** Log Collector cannot change while the Configuration is in use. Clone the Configuration to test it using another Collector.

Create the config

The following block of configuration should give the basic outline needed to read any IIS file using the Graylog Sidecar. **Make sure to update the field to fit your logging scheme.**

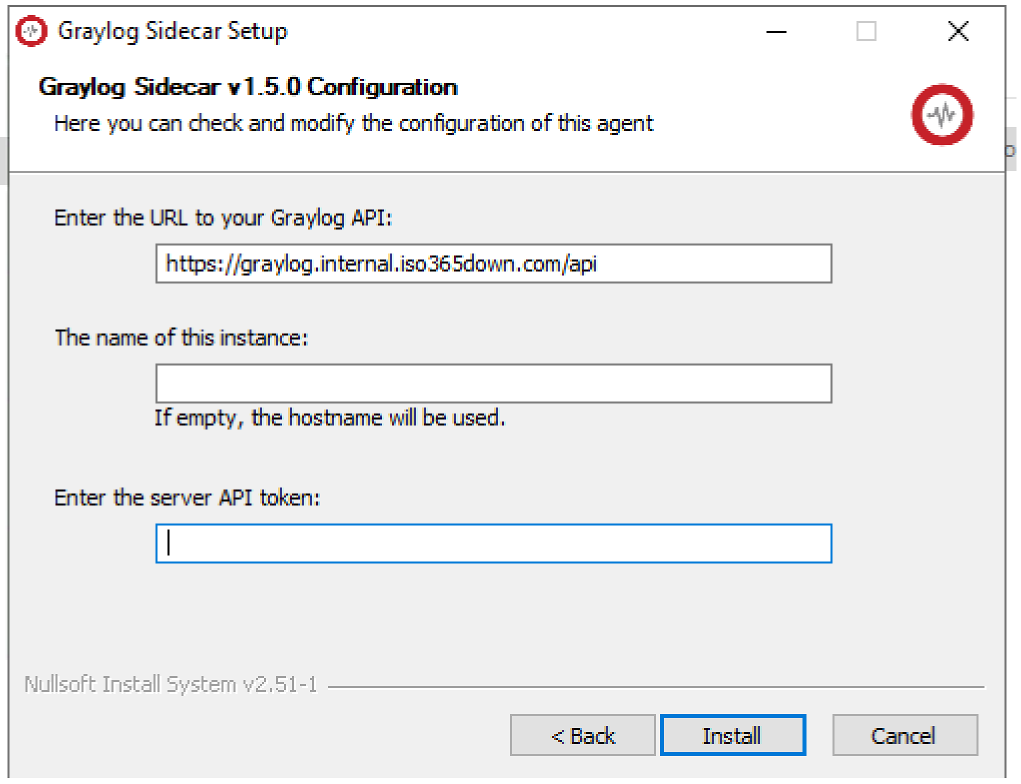
```
# Needed for Graylog
fields_under_root:true
fields.collector_node_id:${sidecar.nodeName}
fields.gl2_source_collector:${sidecar.nodeId}


output.logstash:
hosts: ["graylog.internal.iso365down.com:5044"]
path:
data:${sidecar.spoolDir!"C:\\ProgramFiles\\Graylog\\sidecar\\cache\\winlogbeat"}\da
logs:${sidecar.spoolDir!"C:\\ProgramFiles\\Graylog\\sidecar"}\logs
```

## Windows Graylog Sidecar Install

Because we are using Graylog 5.2.2 we can use the Graylog Sidecar 1.5.0 code. It can be downloaded **.**

Using the API key we generated in part 4, install the Graylog collector and point it at our Graylog instance.

Make sure you point the system at your Graylog instance using a https://FQDN/api.



**Graylog Sidecar v1.5.0 Configuration**
Here you can check and modify the configuration of this agent

Enter the URL to your Graylog API:
https://graylog.internal.iso365down.com/api

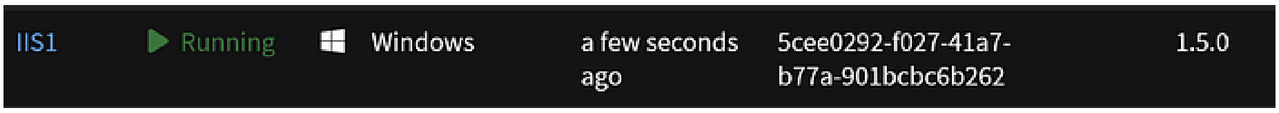
The name of this instance:

If empty, the hostname will be used.

Enter the server API token:

Make sure you use the FQDN of the Graylog Server

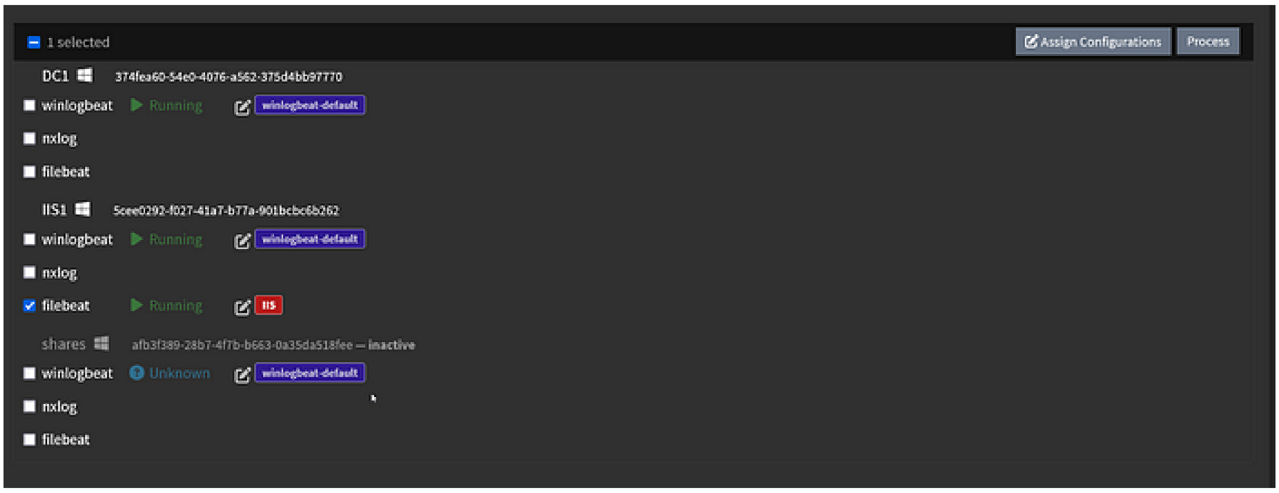
And let it install and validate it is showing up in Graylog.



| | | | | | |
|---|---|---|---|---|---|
| IIS1 | ▶ Running | Windows | a few seconds ago | 5cee0292-f027-41a7-b77a-901bcbc6b262 | 1.5.0 |

itsworking.jpg

Now go to **Administration** and edit your IIS server and assign your filebeat configuration.

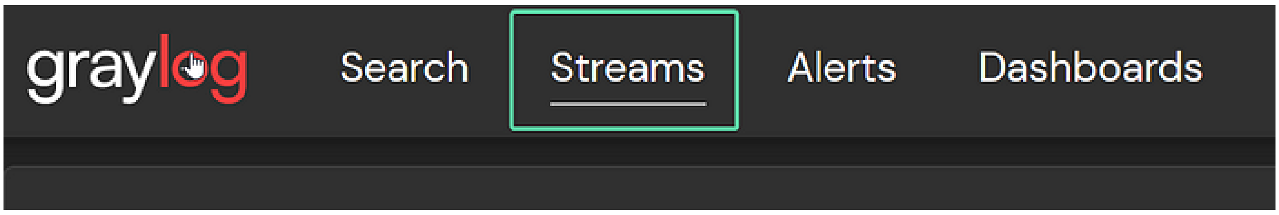


```
1 selected                                          Assign Configurations   Process
DC1     374fea60-54e0-4076-a562-375d4bb97770
☐ winlogbeat   ▶ Running     winlogbeat-default
☐ nxlog
☐ filebeat
IIS1    5cee0292-f027-41a7-b77a-901bcbc6b262
☐ winlogbeat   ▶ Running     winlogbeat-default
☐ nxlog
☑ filebeat     ▶ Running     IIS
shares    afb3f389-28b7-4f7b-b663-0a35da518fee — inactive
☐ winlogbeat   ? Unknown     winlogbeat-default
☐ nxlog
☐ filebeat
```

Assign the filebeat configuration

## Creating a new stream

Start by going to **Streams** in the top menu



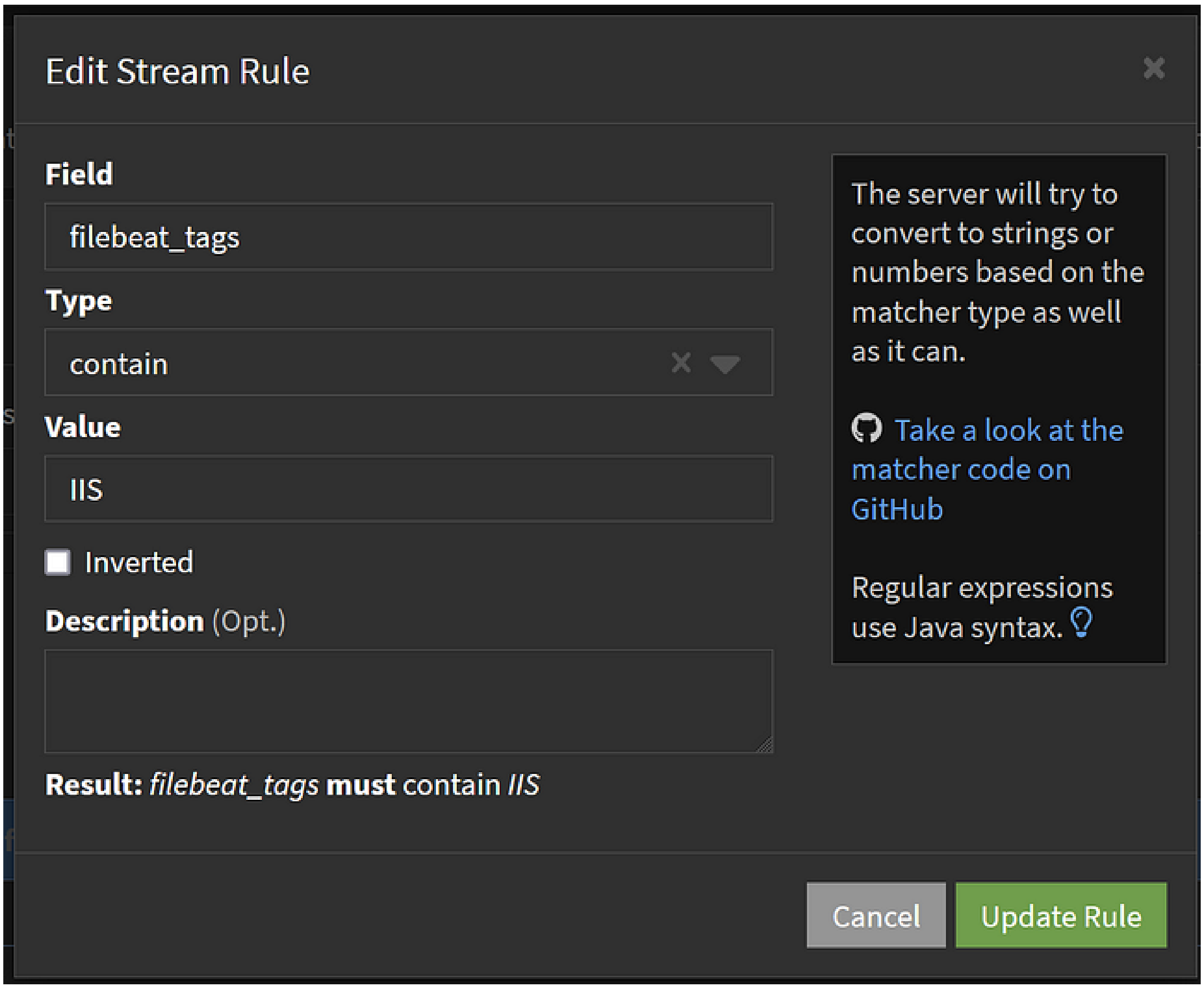
Enter the streams configuration

Create a new stream named IIS Logs and make sure to route the messages to the IIS index set and remove the messages from the default stream.

Name the stream, set the index set it will send data to, and remove the data from the default stream

The newly created stream needs rules configured to send data to it. To do this select **More > Manage Rules.**

Taking the Message ID and Index we saved earlier, load a the message to test against. Search for the field *filebeat_tags* and make a rule matching the field name *IIS.*


Edit Stream Rule

Field: filebeat_tags

Type: contain

Value: IIS

Inverted

Description (Opt.)

Result: *filebeat_tags* **must** contain *IIS*

The server will try to convert to strings or numbers based on the matcher type as well as it can.

Take a look at the matcher code on GitHub

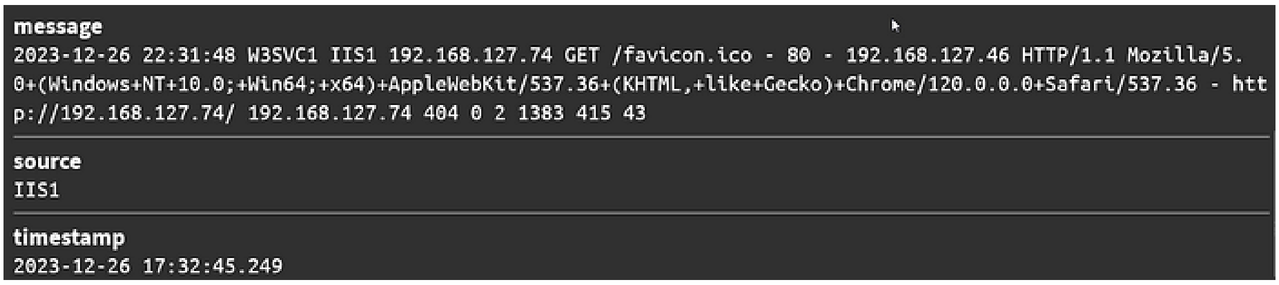
Regular expressions use Java syntax.

Creating our stream rule

Test the stream rule against the message we selected and make sure the rule matches.

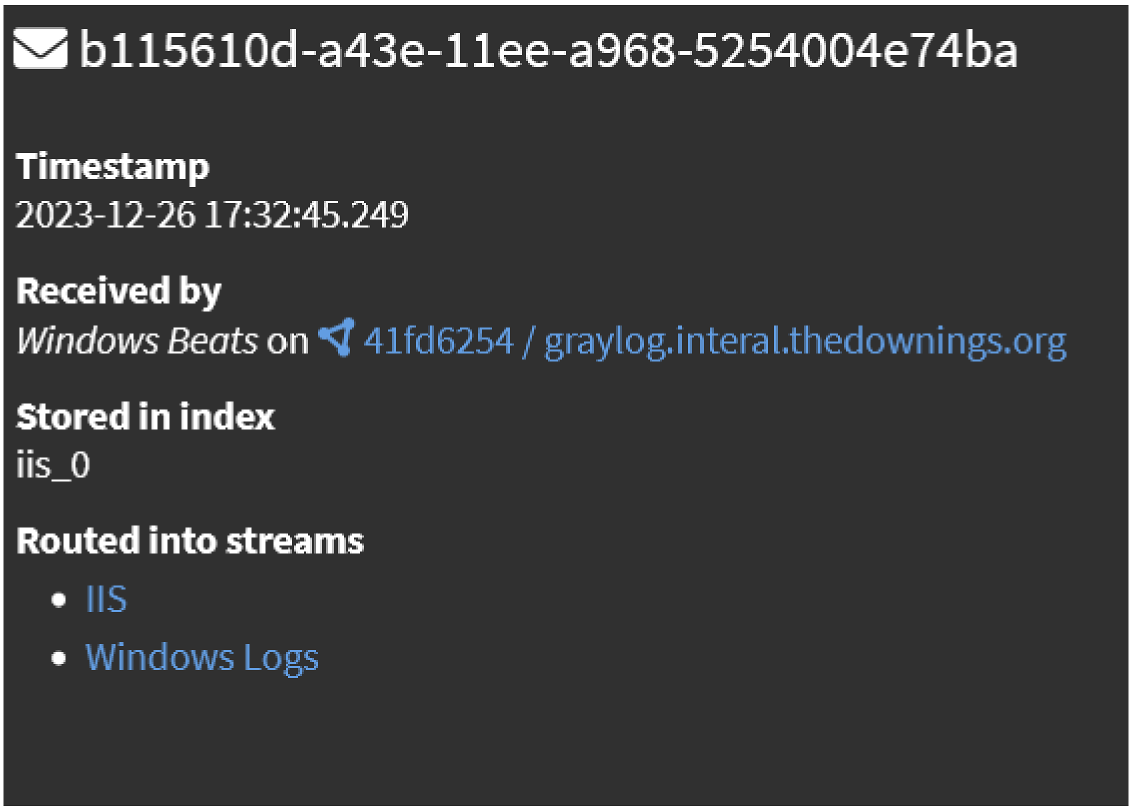
## Decoding our IIS Messages

We have raw IIS logs now, better than what we had before but not super useful. Lets enhance the data we have.


```
message
2023-12-26 22:31:48 W3SVC1 IIS1 192.168.127.74 GET /favicon.ico - 80 - 192.168.127.46 HTTP/1.1 Mozilla/5.
0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/120.0.0.0+Safari/537.36 - htt
p://192.168.127.74/ 192.168.127.74 404 0 2 1383 415 43

source
IIS1

timestamp
2023-12-26 17:32:45.249
```

Just a blog of text

Now that we have IIS messages flowing into our IIS Index, gather the message ID and index ID.



b115610d-a43e-11ee-a968-5254004e74ba

**Timestamp**
2023-12-26 17:32:45.249

**Received by**
*Windows Beats* on 41fd6254 / graylog.interal.thedownings.org

**Stored in index**
iis_0

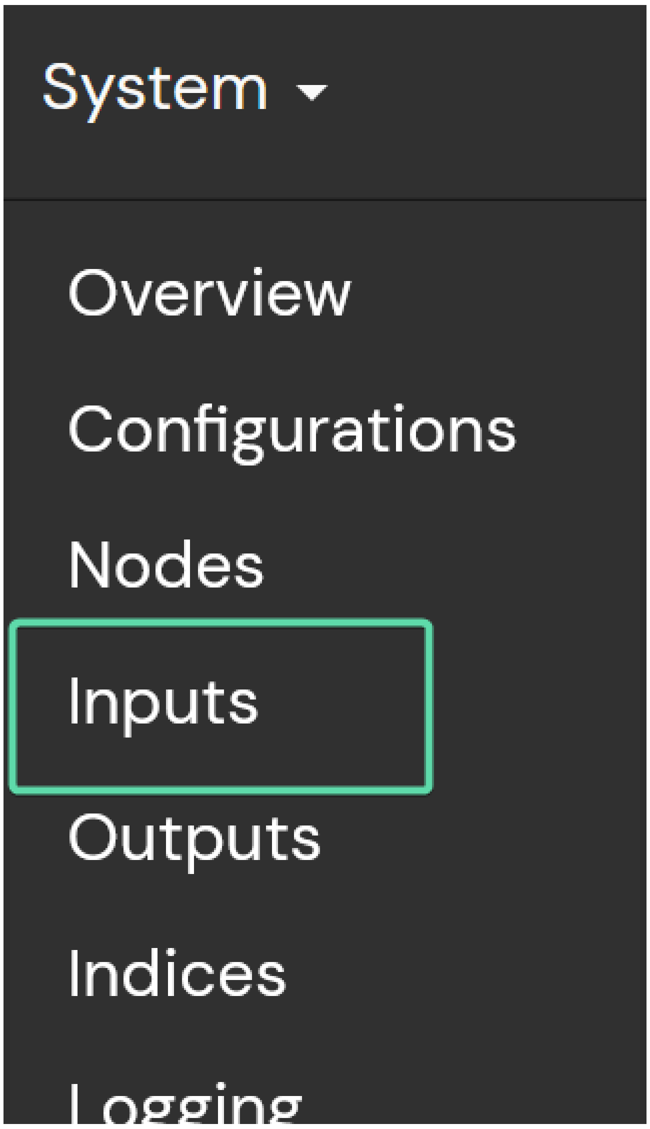
**Routed into streams**

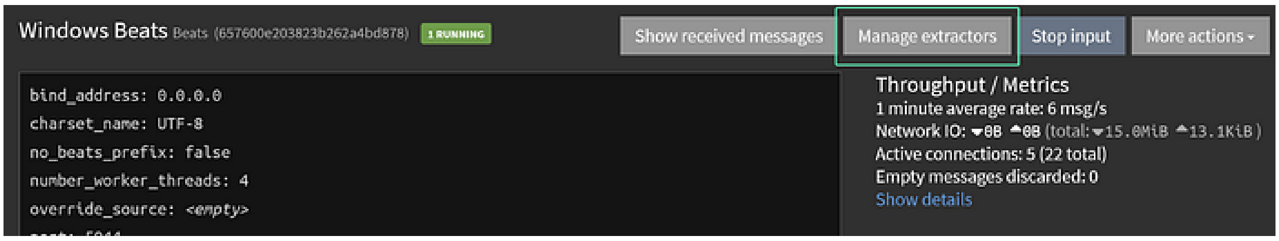
- IIS
- Windows Logs

Just to give credit, I am modifying the data found on flatricks github. I found it didn't work anymore and had to make a small modification. (Change %{SERVERNAME:serverName} to %{HOSTNAME:serverName})

To edit our extractor visit **System > Inputs**

Select **Manage Extractors** on our Windows Beat input we created earlier.





```
bind_address: 0.0.0.0
charset_name: UTF-8
no_beats_prefix: false
number_worker_threads: 4
override_source: <empty>
```

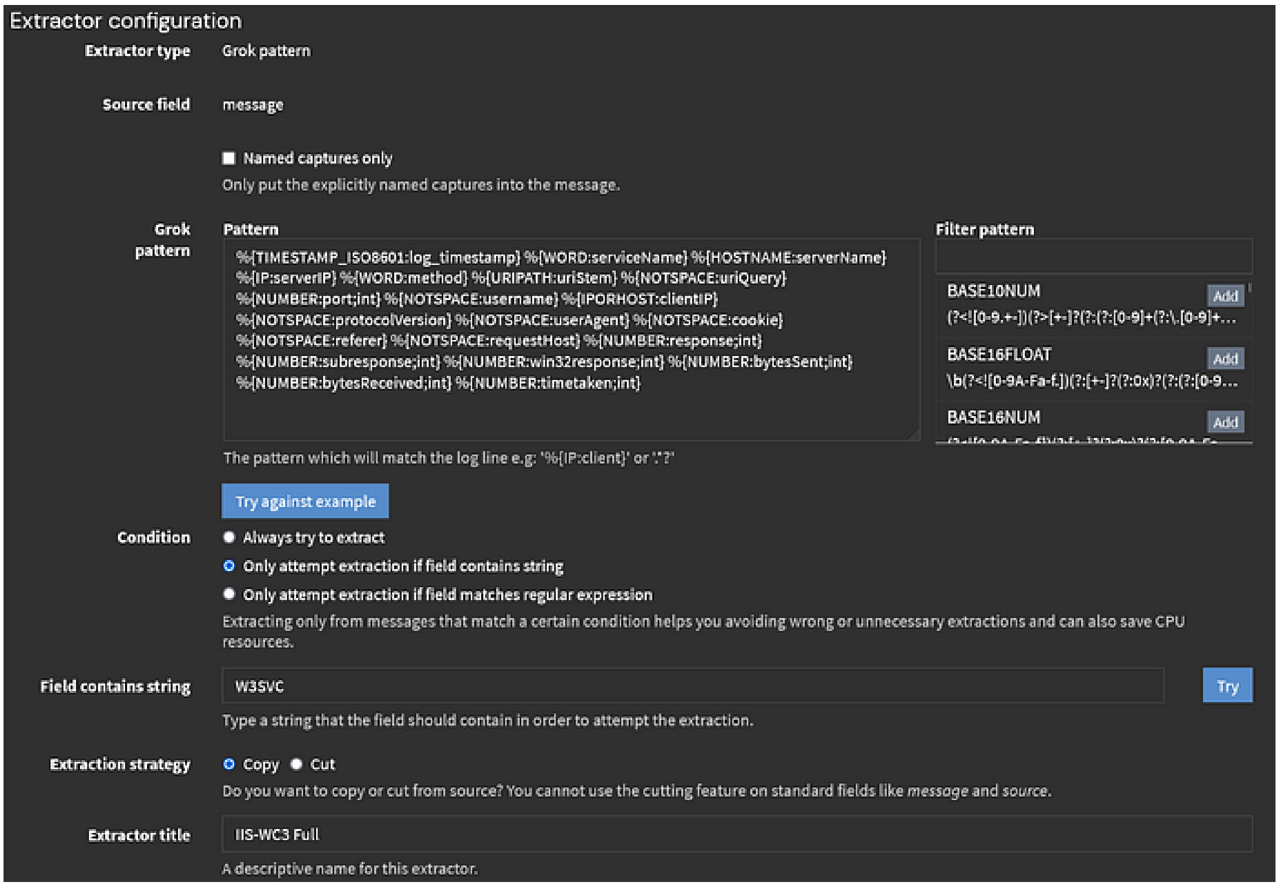
Throughput / Metrics
1 minute average rate: 6 msg/s
Network IO: ▾0B ▴0B (total: ▾15.0MiB ▴13.1KiB )
Active connections: 5 (22 total)
Empty messages discarded: 0
Show details

Create a new extractor

I have exported a working extractor.

```
{   [    {        ,        ,        [],        ,        ,        ,        ,        {
,                 },        ,                }   ],    }
```

Importing this extractor will create a GUI like this. You may want to adjust the field contains string if your applications uses a different naming scheme.



Extractor configuration

**Extractor type** Grok pattern

**Source field** message

☐ Named captures only
Only put the explicitly named captures into the message.

**Grok pattern**

**Pattern**

```
%{TIMESTAMP_ISO8601:log_timestamp} %{WORD:serviceName} %{HOSTNAME:serverName}
%{IP:serverIP} %{WORD:method} %{URIPATH:uriStem} %{NOTSPACE:uriQuery}
%{NUMBER:port;int} %{NOTSPACE:username} %{IPORHOST:clientIP}
%{NOTSPACE:protocolVersion} %{NOTSPACE:userAgent} %{NOTSPACE:cookie}
%{NOTSPACE:referer} %{NOTSPACE:requestHost} %{NUMBER:response;int}
%{NUMBER:subresponse;int} %{NUMBER:win32response;int} %{NUMBER:bytesSent;int}
%{NUMBER:bytesReceived;int} %{NUMBER:timetaken;int}
```

The pattern which will match the log line e.g: '%{IP:client}' or '.*?'

**Filter pattern**

BASE10NUM
(?<![0-9.+-])(?>[+-]?(?:(?:[0-9]+(?:\.[0-9]+…

BASE16FLOAT
\b(?<![0-9A-Fa-f.])(?:[+-]?(?:0x)?(?:(?:[0-9…

BASE16NUM

**Condition**
○ Always try to extract
[✓ Only attempt extraction if field contains string]
○ Only attempt extraction if field matches regular expression
Extracting only from messages that match a certain condition helps you avoiding wrong or unnecessary extractions and can also save CPU resources.

**Field contains string** W3SVC
Type a string that the field should contain in order to attempt the extraction.

**Extraction strategy** [✓ Copy] ○ Cut
Do you want to copy or cut from source? You cannot use the cutting feature on standard fields like *message* and *source*.

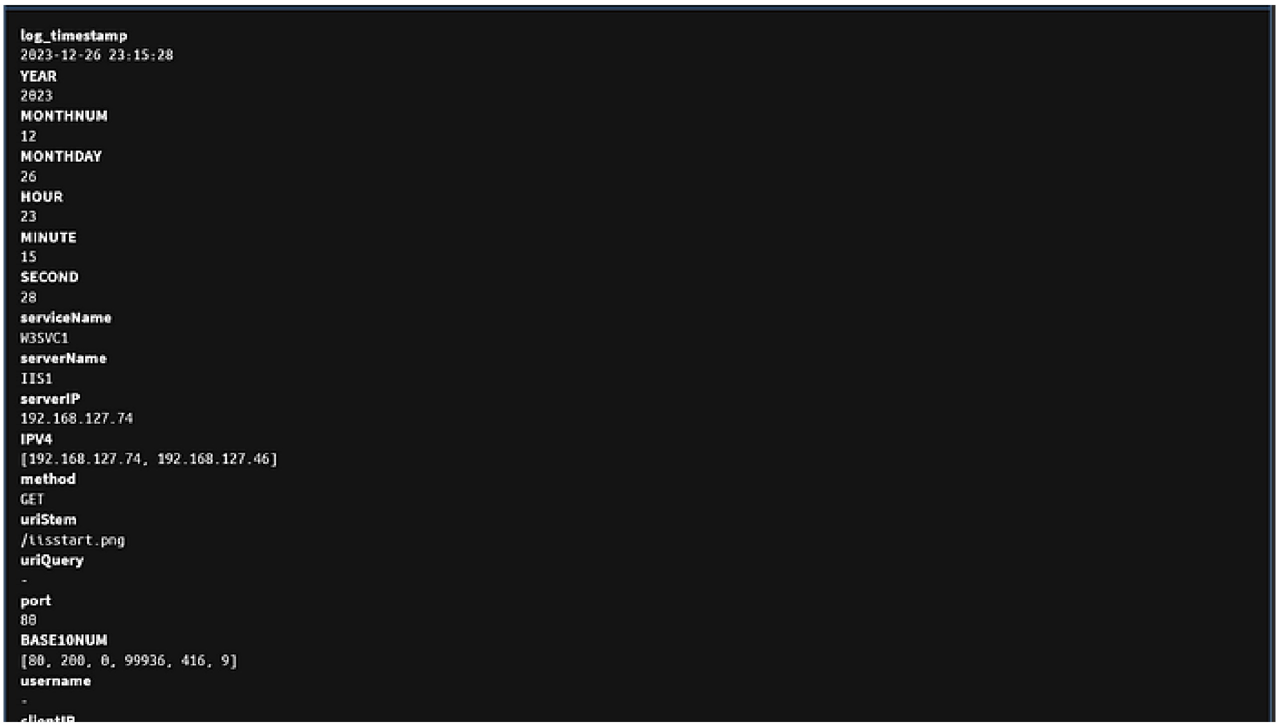
**Extractor title** IIS-WC3 Full
A descriptive name for this extractor.

Our import extractor

Here is the raw GROK pattern.

```
%{TIMESTAMP_ISO8601:log_timestamp} %{WORD:serviceName} %{HOSTNAME:serverName} %
{IP:serverIP} %{WORD:method} %{URIPATH:uriStem} %{NOTSPACE:uriQuery} %
{NUMBER:port;int} %{NOTSPACE:username} %{IPORHOST:clientIP} %
{NOTSPACE:protocolVersion} %{NOTSPACE:userAgent} %{NOTSPACE:cookie} %
{NOTSPACE:referer} %{NOTSPACE:requestHost} %{NUMBER:response;int} %
{NUMBER:subresponse;int} %{NUMBER:win32response;int} %{NUMBER:bytesSent;int} %
{NUMBER:bytesReceived;int} %{NUMBER:timetaken;int}
```

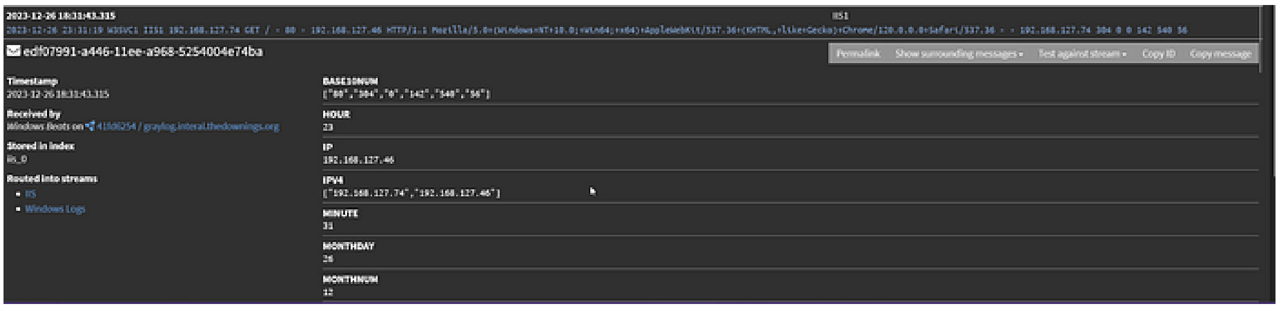
If we now generate a new IIS log and test that against our extractor you will see that Graylog has decoded the string of data we sent it.



```
log_timestamp
2023-12-26 23:15:28
YEAR
2023
MONTHNUM
12
MONTHDAY
26
HOUR
23
MINUTE
15
SECOND
28
serviceName
W3SVC1
serverName
IIS1
serverIP
192.168.127.74
IPV4
[192.168.127.74, 192.168.127.46]
method
GET
uriStem
/iisstart.png
uriQuery
-
port
80
BASE10NUM
[80, 200, 0, 99936, 416, 9]
username
-
```

Successful extraction

If we look at new incoming logs we will see new fields have been added to our message. We can now search on these fields.



```
2023-12-26 18:31:43.315                                                         IIS1
2023-12-26 23:31:19 W3SVC1 IIS1 192.168.127.74 GET / - 80 - 192.168.127.46 HTTP/1.1 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/120.0.0.0+Safari/537.36 - - 192.168.127.74 304 0 0 142 540 36
edf07991-a446-11ee-a968-5254004e74ba

Timestamp                                   BASE10NUM
2023-12-26 18:31:43.315                     ["80","304","0","142","540","36"]
Received by                                 HOUR
Windows Beats on 41fd6254 / graylog.interal.thedownings.org   23
Stored in index                             IP
iis_0                                       192.168.127.46
Routed into streams                         IPV4
  • IIS                                     ["192.168.127.74","192.168.127.46"]
  • Windows Logs                            MINUTE
                                            31
                                            MONTHDAY
                                            26
                                            MONTHNUM
                                            12
```

Our enhanced message

Thanks for sticking around! Our next part will be on SQL logs.