Rules for Open-AudIT

Hi Everybody,

With the release of Open-AudIT 3.2.0 comes a major new feature - Rules.

Rules as a collection of entries that essentially say "If the device has an attribute with X, then make the device's other attribute Y.". That may seem abstract, so what about "If the device has an SNMP OID of 1.3.6.1.4.1.9.1.620, then it's a Cisco 1851 router.".

Out of the box we have rules for MAC address prefixes, SNMP manufacturer IDs and all the aforementioned SNMP OIDs previously within Open-AudIT. We also ship various other rules that were previously hard coded. All up, for our first release we're shipping almost 100,000 rules!

"So what?" you say "What does this mean for me?" - well it means that you no longer need to send me your OIDs and device models, for a start. You can create custom Rules that will detect (almost) anything you like and set the appropriate device attribute.

The Rules are processed when a device's details are processed - during discovery and/or upon processing an audit result (hence, they usually run multiple times). Rules conform to the usual priority system - they will override every thing that's not a user input via the GUI. Rules are considered to be YOUR rules. Not something derived from a device. Hence they mean more than (say) something retrieved via SSH or SNMP or WMI. This is because if they don't do what you want YOU CAN CHANGE THEM.

NOTE - At present we cannot delete a rule input or output that contains a /. This is because the framework is parsing the / as part of the URL and returning a 404, even before our code runs. The work-around for this is to delete the Rule itslef, then recreate the inputs and outputs as required. Fortunately inputs and outputs that contain a / are rare (indeed, none exist by default).

Rules have two main sections - inputs and outputs.

Inputs are what is used to detect and match an attribute (or multiple attributes).

Outputs are what is to be set if the inputs match.

Inputs can use several operators to detect a match, not just equals. We can use the following operators:

equals
does not equal
greater than
greater than or equals
less than
less than or equals
like (which is case-insensitive)
not like (again, case insensitive)
in (a list)
not in (again, a list)
starts with

When we test multiple attributes in a single "input", those attributes are ANDed together. You cannot OR them. The below is an example (from the database, stored as JSON).

```
[{"table":"system","attribute":"manufacturer","operator":"eq","value":"Ubiquiti Networks Inc."},{"table":"
system","attribute":"sysDescr","operator":"li","value":"UAP"}]
```

This translates to If system.manufacturer = Ubiquiti Networks Inc. AND system.sysDescr like UAP then we have a match.

The corresponding "output" section from this rules states (again, in JSON from the database):

```
[{"table":"system","attribute":"type","value":"wap","value_type":"string"},{"table":"system","attribute":"
model","value":"UniFi AP","value_type":"string"}]
```

Which means set the system.type to wap and the system.model to UniFi AP. Outputs can set an attribute using one of three 'types'. a number, text or a timestamp. For a timestamp you have the option of providing a date/time OR leaving it blank and having the system use the current date/time when the Rule is processed.

And don't worry - we wouldn't ask you to write the JSON. The web interface takes care of that for you. Of course, if you're using the API, then the JSON creation is on you :-)

One further item of note is the "weight" attribute we assign Rules. By default it's 100, but any Rules with a higher or lower weight will be run before or after those at weight 100. This provides a way to order the list in which Rules are applied. Mostly you won't need to worry about this, but if required, it's a life-saver.

Rule inputs also don't need to apply only to the "system" table. You can have a Rule on the service table to say if we detect a service named "My Service" then set the device description to "My Service Server" (for a bad example).

At this stage, Rule outputs can only set an attribute (or multiple attributes) on the system table. Custom fields are not supported right now (stay tuned).

The Rules engine is used by Community and available for editing in Professional and Enterprise.

~~So, what's the downside? Well running 100,000 rules several times does take its toll. I was however pleasantly surprised to see it takes less than 1 second each time all 100,000 Rules are processed. It does however mean more memory is consumed. In my testing it uses about 500MB. You shouldn't need to worry about increasing the PHP memory limit as we do this in code, but you will need to keep an eye on your server. Those users that process many devices AT ONCE may run into memory constraints. In general though, most users shouldn't notice any discernible difference. If you do, first thing to try is giving the server a memory bump. It is a database application after all, so more memory and fast disk is always the answer :)~~

UPDATE - With the release of 3.2.2, we no longer store ~100,000 rules in the database. This was fine on my test device (a core i7, 16GB memory and Samsung 860 NVMe), but in practice was causing customers servers to choke.

As per the Release Notes for 3.2.2 -

> *So, that was a ride... In* **testing** *our new Rules feature worked a treat. In* **practice**, *not so much. Most servers (ie, not mine) can't cope with loading the rule set, even if we break it down to smaller chunks, when processing multiple devices. What to do? What to do? Well we've taken a small step back. Rules still exist as a feature, and they still work a treat. But instead of inserting 100,000 Rules into the database, we've split them up into four distinct files and implemented them as code only. Hence, no loading all 100,000 Rules, decoding JSON and running them against a device. Now we just load the files and run the statements. Much, much faster and more memory efficient. No load on MySQL, and hence the CPU also drops. No populating a massive recordset and hence the memory drops. The not so good thing - these are no longer editable in the GUI. But it's not the end of the world. You can still make Rules as you see fit and they will be run after the "default" rules (those in code), hence you can override the "default" rules. So we don't lose much, but we gain a LOT of performance. We also added a few new Rules for Mac Models.*

For those curious, the "new" files that replace the Rules are:

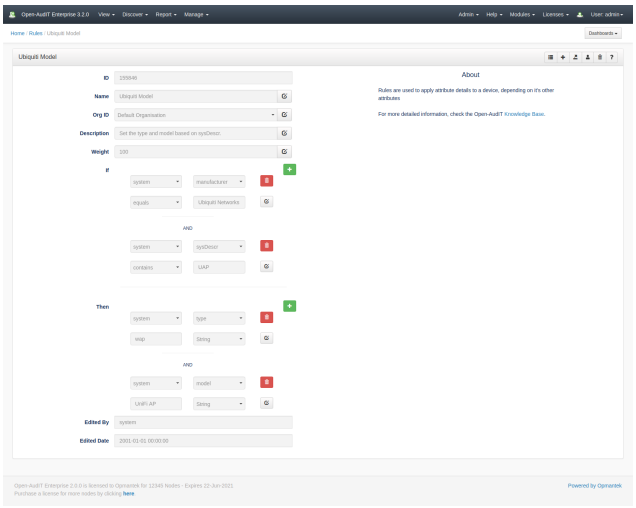
File	Description
/open-audit/code_igniter/application/helpers/mac_helper.php	Matches MAC addresses to manufacturers.
/open-audit/code_igniter/application/helpers/mac_model_helper.php	Matches Apple manufacturer codes to models (stored in system.manufacturer_code).
/open-audit/code_igniter/application/helpers/snmp_model_helper.php	Matches the device's SNMP OID to a model and type.
/open-audit/code_igniter/application/helpers/snmp_model_helper.php	Matches the devices's SNMP OID to the manufacturer.

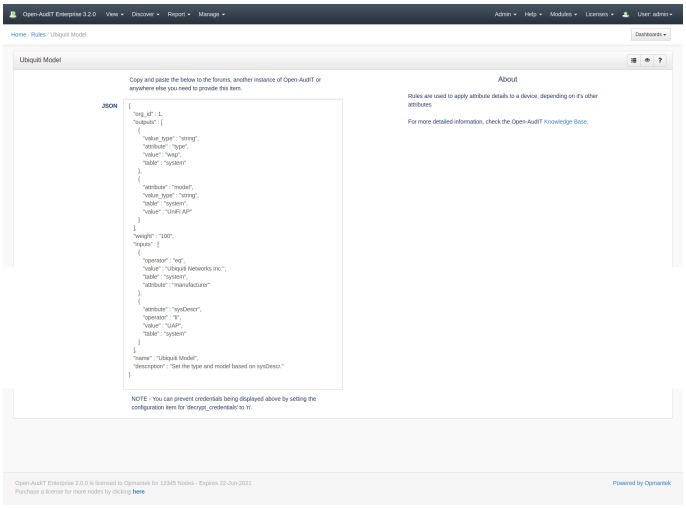
One final thing of note is the new GUI widget. Because we have almost 100,000 Rules, it's just not feasible to display them all in a list in the GUI. ***UPDATE - this is still in place, but you will not see all 100,000 Rules in the GUI as now (as per above) most are back in code files.*** So we don't. We have built a new widget that sits on the panel header and is used to search the Rules. Input anything and the rules name, description, inputs and outputs will be searched and anything matching will be returned. That result-set will still be limited to the default page size (1,000 items), so don't simply search for Cisco and expect to retrieve every Rule (there are 7,828 Cisco Rules by the way).

With this feature we essentially remove the "Tell us your unknown devices" issue as well as provide a powerful tool for you to automatically set attributes of your liking to your devices. Easy.

And one more item - we now have the ability to export a Rule - or anything else for that matter. Exporting an item will provide a JSON object that you can then use for Import. The export button is on each items details page and the import button is on the list page for each collection. With this in place, feel free to send us or post to **Questions** any Rules or Queries you think others may benefit from.

Happy auditing.
Mark Unwin.


Open-AudIT Enterprise 3.2.0
View
Discover
Report
Manage
Admin
Help
Modules
Licenses
User: admin
Home / Rules / Ubiquiti Model
Dashboards
Ubiquiti Model
ID
155949
Name
Ubiquiti Model
Org ID
Default Organisation
Description
Set the type and model based on sysDescr.
Weight
100
If
system
manufacturer
equals
Ubiquiti Networks
AND
system
sysDescr
contains
UAP
Then
system
type
wap
String
AND
system
model
UniFi AP
String
Edited By
system
Edited Date
2001-01-01 00:00:00
About
Rules are used to apply attribute details to a device, depending on it's other attributes
For more detailed information, check the Open-AudIT Knowledge Base.
Open-AudIT Enterprise 2.0.0 is licensed to Opmantek for 12345 Nodes - Expires 22-Jun-2021
Purchase a license for more nodes by clicking here.
Powered by Opmantek


Open-AudIT Enterprise 3.2.0
View
Discover
Report
Manage
Admin
Help
Modules
Licenses
User: admin
Home / Rules / Ubiquiti Model
Dashboards
Ubiquiti Model
Copy and paste the below to the forums, another instance of Open-AudIT or anywhere else you need to provide this item.
JSON
{
"org_id" : 1,
"outputs" : [
{
"value_type" : "string",
"attribute" : "type",
"value" : "wap",
"table" : "system"
},
{
"attribute" : "model",
"value_type" : "string",
"table" : "system",
"value" : "UniFi AP"
}
],
"weight" : "100",
"inputs" : [
{
"operator" : "eq",
"value" : "Ubiquiti Networks Inc.",
"table" : "system",
"attribute" : "manufacturer"
},
{
"attribute" : "sysDescr",
"operator" : "li",
"value" : "UAP",
"table" : "system"
}
],
"name" : "Ubiquiti Model",
"description" : "Set the type and model based on sysDescr."
}
NOTE - You can prevent credentials being displayed above by setting the configuration item for 'decrypt_credentials' to 'n'.
About
Rules are used to apply attribute details to a device, depending on it's other attributes
For more detailed information, check the Open-AudIT Knowledge Base.
Open-AudIT Enterprise 2.0.0 is licensed to Opmantek for 12345 Nodes - Expires 22-Jun-2021
Purchase a license for more nodes by clicking here.
Powered by Opmantek