Samurai

Library version:	RENAT 0.1.7
Library scope:	test suite
Named arguments:	supported

Introduction

A library provides functions to control Samurai application

The library utilize *Selenium2Library* and adds more functions to control Samurai application easily. Without other furthur mentions, all of the concepts of `user`, `user group` are Samurai concepts.

By default, RENAT will try to connecto all Samurai nodes defined in active `local.yaml` at the beginning of the test and disconnect from them at the end of the test automatically. Usually user does not need to use `Connect All` and `Close` explicitly.

Currently, this module supposed that Samurai is used in Japanese locale. When Samurai module has error, it tried to make the last snapshot in `result/selenium-screenshot-x.png`. Checking this capture will help to understand the reason of the error.

Some keywords of Samurai is using `xpath` to identify elements. See *Selenium2Library* for more details about xpath.

See WebApp for common keywords of web applications.

Selenium2Library keywords still could be used together within this library. See Selenium2Library for more details.

Shortcuts

Add Policy · Add Policy Group · Add User · Capture Screenshot · Change Policy View Group · Click All Elements · Close · Close All · Connect · Connect All · Delete Policy · Delete Policy Group · Delete User · Edit Policy · Left Menu · Login · Logout · Make Item Map · Reset Capture Counter · Select Items In Table · Set Capture Counter · Set Capture Format · Show Policy Basic · Show Policy Mitigation · Show Policy Mo · Show Policy Monitor · Start Mitigation · Stop Mitigation · Switch

Keywords

<table>
<tr><th>Keyword</th><th>Arguments</th><th>Documentation</th></tr>
<tr><td><b>Add Policy</b></td><td><i>**policy</i></td><td>
Adds a new Samurai policy
`policy` is a map containing the below information to create the new policy.
<table>
<tr><th>key</th><th>meaning</th><th>mandatory</th><th>sample</th></tr>
<tr><td>name</td><td>name of the policy</td><td>yes</td><td><i>test001</i></td></tr>
<tr><td>basic_alias</td><td>alias name of the policy</td><td></td><td><i>test001</i></td></tr>
<tr><td>basic_port_id</td><td>another alias</td><td></td><td></td></tr>
<tr><td>basic_facing</td><td>`customer` or `backbone`</td><td></td><td><i>customer</i></td></tr>
<tr><td>basic_intf_list</td><td>list of router and interface pair, separated by comma</td><td>yes</td><td><i>10.128.18.31:xe-0/0/0.1</i></td></tr>
<tr><td>basic_cidr_list</td><td>list of CIDR separate by comma</td><td></td><td></td></tr>
<tr><td>basic_option_filter</td><td>optinal filter</td><td></td><td></td></tr>
<tr><td>basic_direction</td><td>direction of the traffic (`incoming` or `outgoing`)</td><td></td><td><i>Incoming</i></td></tr>
<tr><td>traffic_enabled</td><td>Enable traffic monitoring or not</td><td>yes</td><td><i>${True}</i> or <i>${False}</i></td></tr>
<tr><td>detection_enabled</td><td>Enable detection or not</td><td>yes</td><td><i>${True}</i> or <i>${False}</i></td></tr>
<tr><td>mitigation_zone_name</td><td>Name of the zone for mitigation</td><td></td><td><i>zone001</i></td></tr>
<tr><td>mitigation_zone_prefix</td><td>Prefixes that could mitigate</td><td></td><td><i>1.1.1.1/32</i></td></tr>
<tr><td>mitigation_thr_bps</td><td>Upper limit (bps)</td><td></td><td><i>800,000,000</i></td></tr>
<tr><td>mitigation_thr_pps</td><td>Upper limit (pps)</td><td></td><td><i>54,000,000</i></td></tr>
<tr><td>mitigation_mo_enabled</td><td>Using Arbor TMS MO or not</td><td>yes</td><td><i>${True}</i> or <i>${False}</i></td></tr>
<tr><td>mitigation_device_list</td><td>Devices used for TMS, separated by comma</td><td></td><td><i>ArborSP-A</i></td></tr>
<tr><td>mitigation_mo_name</td><td>MO name, separated by comma</td><td></td><td><i>OCN12(ALU)_LOOSE</i></td></tr>
<tr><td>mitigation_comm_list</td><td>comnna separated peer/community list</td><td>yes</td><td><i>1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777</i></td></tr>
<tr><td>nw_monitor_gre1</td><td>1st GRE address for NW monitor</td><td></td><td><i>210.0.1.1</i></td></tr>
<tr><td>nw_monitor_gre2</td><td>2nd GRE address for NW monitor</td><td></td><td><i>210.0.1.1</i></td></tr>
<tr><td>nw_monitor_ce1</td><td>1st CE address for NW monitor</td><td></td><td><i>210.0.1.2</i></td></tr>
<tr><td>nw_monitor_ce2</td><td>2nd CE address for NW monitor</td><td></td><td><i>210.0.1.2</i></td></tr>
<tr><td>nw_monitor_pe1</td><td>1st PE for NW monitor (list)</td><td></td><td><i>edge01hige-MX2020-15(118.23.176.244)</i></td></tr>
<tr><td>nw_monitor_pe2</td><td>2nd PE for NW monitor (list)</td><td></td><td><i>edge01hige-MX2020-15(118.23.176.244)</i></td></tr>
<tr><td>event_name</td><td>name of the message event to make</td><td></td><td><i>info1</i></td></tr>
<tr><td>event_addr</td><td>address to send the events</td><td></td><td><i>user@mail.com</i></td></tr>
<tr><td>view_group</td><td>user group that could view this policy, separated by comma</td><td>yes</td><td><i>SuperGroup,test_group_007</i></td></tr>
</table>
Example:
<table>
<tr><td>Samurai.<i>Switch</i></td><td>samurai-1</td><td></td></tr>
<tr><td>Samurai.<i>Add Policy</i></td><td>name=${POLICY_NAME}</td><td>basic_alias=${POLICY_NAME}</td></tr>
<tr><td>...</td><td>basic_facing=customer</td><td>basic_intf_list=10.128.18.31:xe-0/0/0.1</td></tr>
<tr><td>...</td><td>basic_cidr_list=1.1.1.0/24</td><td>basic_direction=incoming</td></tr>
<tr><td>...</td><td>traffic_enabled=${TRUE}</td><td></td></tr>
<tr><td>...</td><td>detection_enabled=${TRUE}</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td>...</td><td>mitigation_zone_name=test_zone001</td><td>mitigation_zone_prefix=1.1.1.1/32</td></tr>
<tr><td>...</td><td>mitigation_device_list=ArborSP-A,ArborSP-B</td><td></td></tr>
<tr><td>...</td><td>mitigation_mo_enabled=${TRUE}</td><td></td></tr>
<tr><td>...</td><td>mitigation_mo_name=N000000012_LOOSE</td><td></td></tr>
<tr><td>...</td><td>mitigation_comm_list=1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777</td><td></td></tr>
<tr><td>...</td><td>event_name=test</td><td>event_addr=user@mail.com</td></tr>
<tr><td>...</td><td>view_group=SuperGroup</td><td></td></tr>
</table>
</td></tr>
<tr><td>**Add Policy Group**</td><td>*group_name*, *policy_list=**, *limit_bps=4000000000*, *limit_pps=2700000*</td><td>Add a new policy group
`group_name` is the name of the new group. `policy_list` is a comma separated of existed policy that should be bound to this policy. An asterisk for this parameter (`*`) means *all of the existed policy*. `limit_bps` and `limit_pps` are the mitigation capacity threshold of this group.</td></tr>
<tr><td>**Add User**</td><td>*group*, ***user_info*</td><td>Adds user to the current group `user_info` is a dictionary contains user information that has following keys: `name`, `password`, `privilege` and `policy`
`privilege` is existed privilege that has been created (e.g: *system_admin*.
`policy` could be `*` for all current policies or a list of policy names that are binded to this user.
`group` is the user group. `Dot(.)` means current group
Examples:
<table>
<tr><td>Samurai.*Add User*</td><td>OCNDDoS</td><td>name=user000</td><td>password=Test12345678</td></tr>
<tr><td>...</td><td>privilege=system_admin</td><td>policy=*</td><td></td></tr>
<tr><td>Samurai.*Add User*</td><td>OCNDDoS</td><td>username=user001</td><td>password=Test12345678</td></tr>
<tr><td>...</td><td>privilege=system_admin</td><td>policy=OCN11,OCN12</td><td></td></tr>
</table>
</td></tr>
<tr><td>**Capture Screenshot**</td><td>*filename=None*, *extra=*</td><td>Captures the current screen to file
Using the internal counter for filename if `filename` is not specified. In this case, the filename is defined by a pre-set format. *Set Capture Format* could be used to change the current format.
An extra information will be add to the filename if `extra` is defined
Examples:
<table>
<tr><td>Samurai.*Capture Screenshot*</td><td></td><td># samurai_0000000001.png</td></tr>
<tr><td>Samurai.*Capture Screenshot*</td><td>extra=_list</td><td># samurai_0000000002_*list*.png</td></tr>
<tr><td>Arbor.*Capture Screenshot*</td><td></td><td># arbor_0000000001.png</td></tr>
<tr><td>Arbor.*Capture Screenshot*</td><td>extra=_xxx</td><td># arbor_0000000001_*xxx*.png</td></tr>
<tr><td>Samurai.*Capture Screenshot*</td><td>file_name=1111.png</td><td># 1111.png</td></tr>
</table>
</td></tr>
<tr><td>**Change Policy View Group**</td><td>*name*, **group_name*</td><td>Changes the groups that could see this policy
`name` is the policy name. `group_name` is a list of policies
Example:
<table>
<tr><td>Samurai.*Change Policy View Group*</td><td>super_admin</td><td>test_group001</td></tr>
</table>
</td></tr>
<tr><td>**Click All Elements**</td><td>*xpath*</td><td>Click all element in current page defined by `xpath`
Returns the number of elements that have been clicked</td></tr>
<tr><td>**Close**</td><td></td><td>Closes the current active browser</td></tr>
<tr><td>**Close All**</td><td></td><td>Closes all current opened applications</td></tr>
<tr><td>**Connect**</td><td>*app*, *name*</td><td>Opens a web browser and connects to application and assigns a `name`.
If not defined in `local.yaml` those following key will have defaut values:
<table>
<tr><td>browser</td><td>firefox</td><td>optional</td></tr>
<tr><td>login_url</td><td>/</td><td>optiona</td></tr>
<tr><td>proxy:</td><td></td><td>optional</td></tr>
<tr><td>http: 10.128.8.210:8080</td><td>optional</td><td></td></tr>
<tr><td>ssl: 10.128.8.210:8080</td><td>optional</td><td></td></tr>
<tr><td>socks: 10.128.8.210:8080</td><td>optional</td><td></td></tr>
<tr><td>profile_dir</td><td>./config/samurai.profile</td><td>optional</td></tr>
</table>
</td></tr>
<tr><td>**Connect All**</td><td></td><td>Connects to all applications defined in `local.yaml`
The name of the connection will be the same of the *webapp* name</td></tr>
<tr><td>**Delete Policy**</td><td>**policy_names*</td><td>Deletes poilcies by their names
Returned the number of deleted users
Notes: If the policy does not exists, the system will not report any error.
Examples:
<table>
<tr><td>Samurai.*Delete Policy*</td><td>test001</td><td>test002</td></tr>
</table>
</td></tr>
<tr><td>**Delete Policy Group**</td><td>**group_list*</td><td>Deletes policy groups
Returns the number of deleted policy groups Example:
<table>
<tr><td>Samurai.*Delete Policy Group*</td><td>test_group001</td><td>test_group002</td></tr>
</table>
</td></tr>
<tr><td>**Delete User**</td><td>*group*, **user_list*</td><td>Deletes user from the user group
`group` is the user group. And `.` means current group Returns the number of deleted users
Examples:
<table>
<tr><td>Samurai.*Delete User*</td><td>SuperGroup</td><td>user001</td><td>user002</td></tr>
<tr><td>Samurai.*Delete User*</td><td>.</td><td>user002</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td><b>Edit Policy</b></td><td><i>**policy</i></td><td>Edits a Samurai policy
<code>policy</code> contains information about the policy. See <i>Add Policy</i> for more details about <code>policy</code> format</td></tr>
<tr><td><b>Left Menu</b></td><td><i>menu</i></td><td>Chooses the left panel menu by its displayed name
Return a list of 1st meaningful column Example:<table><tr><td>Samurai.<i>Left Menu</i></td><td>Traffic</td></tr><tr><td>Samurai.<i>Left Menu</i></td><td>Detection</td></tr><tr><td>Samurai.<i>Left Menu</i></td><td>ポリシー管理</td></tr></table></td></tr>
<tr><td><b>Login</b></td><td></td><td>Logs-in into the application
User and password is set by the template and authentication methods in the master files</td></tr>
<tr><td><b>Logout</b></td><td></td><td>Logs-out the current application, the browser remains</td></tr>
<tr><td><b>Make Item Map</b></td><td><i>xpath</i></td><td>Makes a item/webelement defined <i>xpath</i>
The map is a dictionary from <i>item</i> to the <i>WebElement</i> Items name found by <code>xpath</code> are used as keys</td></tr>
<tr><td><b>Reset Capture Counter</b></td><td></td><td>Resets the counter of the screen capture</td></tr>
<tr><td><b>Select Items In Table</b></td><td><i>xpath</i>, <i>xpath2</i>, <i>*item_list</i></td><td>Checks items in Samurai table by xpath
<code>xpath</code> points to the column that used as key and <code>xpath2</code> is the relative xpath contains the checkbox column.
<code>item_list</code> is a list of item that need to check. Item in the list could be a regular expresion with the format <code>reg=&lt;regular expression</code>.
The keyword is called with assuming that the table is already visible.
Returns the tupple of all items and selected items
<b>Note:</b> Non-width-space (\u200b) will be take care by the keyword.
<b>Note:</b> if the first item_list is * then the keywork will try to click a link named <i>すべてを選択</i>.</td></tr>
<tr><td><b>Set Capture Counter</b></td><td><i>value=0</i></td><td>Sets the counter of the screen capture to <code>value</code></td></tr>
<tr><td><b>Set Capture Format</b></td><td><i>format</i></td><td>Sets the format for the screen capture file
The format does not include the default prefix <code>.png</code> The default format is <code>&lt;mod&gt;_%010d</code>. <code>mod</code> could be <code>samurai</code> or <code>arbor</code>
See https://docs.python.org/2/library/string.html#format-specification-mini-language for more details about the format string.
Examples:<table><tr><td>Samurai.<i>Set Capture Format</i></td><td>${case}_%010d</td><td># ${case} is a predefined variable</td></tr></table></td></tr>
<tr><td><b>Show Policy Basic</b></td><td><i>policy_name</i></td><td>Makes the virtual browser show basic setting of the policy <i>name</i>.
A following Samurai.<i>Capture Screenshot</i> is necessary to capture the result.</td></tr>
<tr><td><b>Show Policy Mitigation</b></td><td><i>policy_name</i></td><td>Make the virtual browser show the mitigation setting of a policy
A following Samurai.<i>Capture Screenshot</i> is necessary to capture the result.</td></tr>
<tr><td><b>Show Policy Mo</b></td><td><i>policy_name</i></td><td>Make the virtual browser show the MO setting of a policy
Automatically expand the MO section of other devices if necessary.
A following Samurai.<i>Capture Screenshot</i> is necessary to capture the result.</td></tr>
<tr><td><b>Show Policy Monitor</b></td><td><i>policy_name</i></td><td>Make a virtual browser show the mitigation setting of a policy
A following Samurai.<i>Capture Screenshot</i> is necessary to capture the result.</td></tr>
<tr><td><b>Start Mitigation</b></td><td><i>policy</i>, <i>prefix</i>, <i>comment=mitigation started by RENAT</i>, <i>device=None</i>, <i>force=False</i></td><td>Starts a mitigation with specific <code>prefix</code>
<code>device</code> is used for matching real device name configured by Samurai If <code>force</code> is <code>TRUE</code> then the keyword will fail if selected device does not contain <code>device</code>
Returns mitigation <code>id</code> and selected <code>arbor device</code>
Example:<table><tr><td>${id}</td><td>${device}=</td><td>Samurai.<i>Start Mitigation</i></td><td>211.1.12.1/32</td><td>mitigation by RENAT</td><td>SP-A</td><td>${TRUE}</td></tr></table></td></tr>
<tr><td><b>Stop Mitigation</b></td><td><i>id</i></td><td>Stops a mitigation by its ID
Example:<table><tr><td>Samurai.<i>Stop Mitigation</i></td><td>700</td></tr></table></td></tr>
<tr><td><b>Switch</b></td><td><i>name</i></td><td>Switches the current browser to <code>name</code></td></tr>
</table>

Altogether 29 keywords.
Generated by Libdoc on 2018-03-20 02:58:14.