Samurai

Library version:	RENAT 0.1.10
Library scope:	test suite
Named arguments:	supported

Introduction

A library provides functions to control Samurai application

The library utilize *SeleniumLibrary* 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 connec to 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.

Currently the module support Samurai 09/14/16

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 and how to configure the `local.yaml` file.

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 · Close Window · Connect · Connect All · Delete Policy · Delete Policy Group · Delete User · Edit Mitigation Controller · Edit Policy · Get Mitigation List · Left Menu · Login · Logout · Make Item Map · Reconnect · Reset Capture Counter · Select Items In Table · Select Window · Set Ajax Wait · Set Capture Counter · Set Capture Format · Show Detail Mitigation · Show Policy Basic · Show Policy Detection · Show Policy Mitigation · Show Policy Mo · Show Policy Monitor · Start Mitigation · Stop Mitigation · Switch

Keywords

Keyword	Arguments	Documentation
Add Policy	***policy*	Adds a new Samurai policy `policy` is a map containing the below information to create the new policy.

key	meaning	mandatory	sample
name	name of the policy	yes	*test001*
basic_alias	alias name of the policy		*test001*
basic_port_id	another alias		
basic_facing	`customer` or `backbone`		*customer*
basic_intf_list	list of router and interface pair, separated by comma	yes	*10.128.18.31:xe-0/0/0.1*
basic_cidr_list	list of CIDR separate by comma		
basic_option_filter	optinal filter		
basic_direction	direction of the traffic (`incoming` or `outgoing`)		*Incoming*
traffic_enabled	Enable traffic monitoring or not	yes	*${TRUE} or ${FALSE}*
detection_enabled	Enable detection or not	yes	*${TRUE} or ${FALSE}*
mitigation_enabled	Enable Mitigation or not	yes	*${TRUE} or ${FALSE}*
mitigation_zone_name	Name of the zone for mitigation		*zone001*
mitigation_zone_prefix	Prefixes that could mitigate		*1.1.1.1/32*
mitigation_thr_bps	Upper limit (bps)		*800,000,000*
mitigation_thr_pps	Upper limit (pps)		*54,000,000*
mitigation_auto_enabled	Enable automitgation or not		*${TRUE} or ${FALSE}*
mitigation_auto_level	Automitgation level		0:overLow 1:overMedium 2:High
mitigation_auto_time	Automitigation detect attack time (min)		default is 15
mitigation_mo_enabled	Using Arbor TMS MO or not	yes	*${TRUE} or ${FALSE}*
mitigation_auto_stop_enabled	Enable automitgation stop or not		*${TRUE} or ${FALSE}*
mitigation_auto_stop_level	Automitgation level		0:overLow 2:High
mitigation_auto_stop_time	Automitigation stop detect attack time (min)		default is 15
mitigation_device_list	Devices used for TMS, separated by comma		*ArborSP-A*
mitigation_mo_name	MO name, separated by comma		*OCN12(ALU)_LOOSE*
mitigation_comm_list	commna separated peer/community list		*1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777*
nw_monitor_gre1	1st GRE address for NW monitor		*210.0.1.1*
nw_monitor_gre2	2nd GRE address for NW monitor		*210.0.1.1*
nw_monitor_ce1	1st CE address for NW monitor		*210.0.1.2*
nw_monitor_ce2	2nd CE address for NW monitor		*210.0.1.2*

nw_monitor_pe1	1st PE for NW monitor (list)		*edge01hige-MX2020-15(118.23.176.244)*
nw_monitor_pe2	2nd PE for NW monitor (list)		*edge01hige-MX2020-15(118.23.176.244)*
event_name	name of the message event to make		*info1*
event_addr	address to send the events		*user@mail.com*
view_group	user group that could view this policy, separated by comma	yes	*SuperGroup,test_group_007*

Example:

Samurai.*Switch*	samurai-1	
Samurai.*Add Policy*	name=${POLICY_NAME}	basic_alias=${POLICY_NAME}
...	basic_facing=customer	basic_intf_list=10.128.18.31:xe-0/0/0.1
...	basic_cidr_list=1.1.1.0/24	basic_direction=incoming
...	traffic_enabled=${TRUE}	
...	detection_enabled=${TRUE}	
...	mitigation_zone_name=test_zone001	mitigation_zone_prefix=1.1.1.1/32
...	mitigation_device_list=ArborSP-A,ArborSP-B	
...	mitigation_mo_enabled=${TRUE}	
...	mitigation_mo_name=N000000012_LOOSE	
...	mitigation_comm_list=1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777	
...	event_name=test	event_addr=user@mail.com
...	view_group=SuperGroup	

Command	Arguments	Description
Add Policy Group	*group_name, policy_list=*, limit_bps=4000000000, limit_pps=2700000*	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.
Add User	*group, **user_info*	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: | Samurai.*Add User*	OCNDDoS	name=user000	password=Test12345678
---	---	---	---
...	privilege=system_admin	policy=*	
Samurai.*Add User*	OCNDDoS	username=user001	password=Test12345678
...	privilege=system_admin	policy=OCN11,OCN12		
Capture Screenshot	*filename=None, extra=*	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: 	Samurai.*Capture Screenshot*		# samurai_0000000001.png
Samurai.*Capture Screenshot*	extra=_list	# samurai_0000000002_*list*.png
Arbor.*Capture Screenshot*		# arbor_0000000001.png
Arbor.*Capture Screenshot*	extra=_xxx	# arbor_0000000001_*xxx*.png
Samurai.*Capture Screenshot*	filename=1111.png	# 1111.png	
Change Policy View Group	*name, *group_name*	Changes the groups that could see this policy `name` is the policy name. `group_name` is a list of policies Example: | Samurai.*Change Policy View Group*	super_admin	test_group001	
Click All Elements	*xpath*	Click all element in current page defined by `xpath` Returns the number of elements that have been clicked
Close		Closes the current active browser
Close All		Closes all current opened applications
Close Window		Closes the current window
Connect	*app, name*	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: 	browser	firefox	optional
login_url	/	optiona
proxy:		optional
http: 10.128.8.210:8080	optional	
ssl: 10.128.8.210:8080	optional	
socks: 10.128.8.210:8080	optional	
profile_dir	./config/samurai.profile	optional	
Connect All		Connects to all applications defined in `local.yaml` The name of the connection will be the same of the *webapp* name

Delete Policy	*policy_names*	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:
		`Samurai.Delete Policy` `test001` `test002`
Delete Policy Group	*group_list*	Deletes policy groups
		See `Select Items In Table` for more detail about how to choose *group_list*
		Returns the number of deleted policy groups Example:
		`Samurai.Delete Policy Group` `test_group001` `test_group002`
Delete User	*group*, **user_list*	Deletes user from the user group
		`group` is the user group. And `.` means current group Returns the number of deleted users
		Examples:
		`Samurai.Delete User` `SuperGroup` `user001` `user002`
		`Samurai.Delete User` `.` `user002`
Edit Mitigation Controller	*controller*, ***config*	Change the setting of the mitigation control
		▪ `control` : name of the mitigation controller
		▪ `config` : configuration need to be changed. Currently only `tms_group` is configurable with the following format: `groupname1:action1,groupname2:action2` . `groupname` is currently set TMS group name and action could be *click*,*check* or *uncheck*.
		Example:
		`Samurai.Edit Mitigation Controller` `controller=vSP-A` `tms_group=Logical0_SOCN_IPv4:uncheck`
Edit Policy	***policy*	Edits a Samurai policy
		`policy` contains information about the policy. See `Add Policy` for more details about `policy` format
Get Mitigation List	*status=実行中*	Gets current mitigation list
		Return current active mitgation name, ID and the number of them
		Example:
		`${MITI}` `${IDS}` `${NUM}=` `Samurai.Get Mitigation List`
Left Menu	*menu*, *locator=None*, *ignore_first_element=True*	Chooses the left panel menu by its displayed name
		When `locater` is not null, the keyword will return a list of text attribute of all elements specified by the `locator` . `locator` could be a xpath or a predefined string.
		`locator` predefined strings are: `MITIGATE_REALTIME` , `MITIGATE_LIST` , `DETECT_LIST`
		For example, a xpath `//div[@id='infoareain2']/*/td[1]/a` means the list of *link* of all elements in a 1st column of a table insides a `div` with id `infoareain2` .
		Examples:
		`Samurai.Left Menu` `Traffic`
		`Samurai.Left Menu` `Detection`
		`Samurai.Left Menu` `ポリシー管理`
		`@{LIST}=` `Samurai.Left Menu` `Active Mitigation` `//div[@id='infoareain2']/*/td[1]/a`
Login		Logs-in into the application
		User and password is set by the template and authentication methods in the master files
Logout		Logs-out the current application, the browser remains
Make Item Map	*xpath*	Makes a item/webelement defined *xpath*
		The map is a dictionary from *item* to the *WebElement* Items name found by `xpath` are used as keys
Reconnect		Reconnects to the server
Reset Capture Counter		Resets the counter of the screen capture
Select Items In Table	*xpath*, *xpath2*, **item_list*	Checks items in Samurai table by xpath
		`xpath` points to the column that used as key and `xpath2` is the relative xpath contains the target column.
		`item_list` is a list of item and its action that need to check. Item in the list could be a regular expresion with the format `re:<regular expression>|action` .
		The default action for the item could be `click` `(default),`check` or `uncheck`
		The keyword is called with assuming that the table is already visible.
		Returns the tupple of all items and selected items
		Note: Non-width-space (\u200b) will be take care by the keyword.
		Note: if the first item_list is *** then the keywork will try to click a link named *すべてを選択*.
Select Window	*title*	Selects a window by its title
Set Ajax Wait	*wait_time=2s*	Set the ajax wait time
Set Capture Counter	*value=0*	Sets the counter of the screen capture to `value`
Set Capture Format	*format*	Sets the format for the screen capture file
		The format does not include the default prefix `.png` The default format is `<mod>_%010d` . `mod` could be `samurai` or `arbor`

See https://docs.python.org/2/library/string.html#format-specification-mini-language for more details about the format string.

Examples:

Samurai.*Set Capture Format* ${case}_%010d # ${case} is a predefined variable

Show Detail Mitigation	*id*	Shows detail information of a mitigation
Show Policy Basic	*policy_name*	Makes the virtual browser show basic setting of the policy *name*.
		A following Samurai.*Capture Screenshot* is necessary to capture the result.
Show Policy Detection	*policy_name*	Shows the detction pannel of *policy_name* policy
Show Policy Mitigation	*policy_name*	Make the virtual browser show the mitigation setting of a policy
		A following Samurai.*Capture Screenshot* is necessary to capture the result.
Show Policy Mo	*policy_name*	Make the virtual browser show the MO setting of a policy
		Automatically expand the MO section of other devices if necessary.
		A following Samurai.*Capture Screenshot* is necessary to capture the result.
Show Policy Monitor	*policy_name*	Make a virtual browser show the mitigation setting of a policy
		A following Samurai.*Capture Screenshot* is necessary to capture the result.
Start Mitigation	*policy*, *prefix*, *comment=mitigation started by RENAT*, *device=None*, *force=False*	Starts a mitigation with specific `prefix`
		`device` is used for matching real device name configured by Samurai If `force` is `TRUE` then the keyword will fail if selected device does not contain `device`
		Returns mitigation `id` and selected `arbor device`
		Example:
		${id} ${device}= Samurai.*Start Mitigation* 211.1.12.1/32 mitigation by RENAT SP-A ${TRUE}
Stop Mitigation	*id*, *raise_error=True*	Stops a mitigation by its ID
		The keyword will raise an error if *raise_error* is `True`. Otherwise it will ignore any errors.
		Example:
		Samurai.*Stop Mitigation* 700
Switch	*name*	Switches the current browser to `name`

Altogether 37 keywords.
Generated by Libdoc on 2018-10-23 21:34:39.