» incapsula_acl_security_rule

Provides a Incapsula ACL Security Rule resource. ACL Security Rules allow for blacklisting or whitelisting countries, IP addresses, and URLs.

» Example Usage

```
resource "incapsula_acl_security_rule" "example-global-blacklist-country-rule" {
  site_id = "${incapsula_site.example-site.id}"
  rule_id = "api.acl.blacklisted_countries"
  countries = "AI,AN"
}

resource "incapsula_acl_security_rule" "example-global-blacklist-ip-rule" {
  rule_id = "api.acl.blacklisted_ips"
  site_id = "${incapsula_site.example-site.id}"
  ips = "192.168.1.1,192.168.1.2"
}

resource "incapsula_acl_security_rule" "example-global-blacklist-url-rule" {
  rule_id = "api.acl.blacklisted_urls"
  site_id = "${incapsula_site.example-site.id}"
  url_patterns = "CONTAINS,EQUALS"
  urls = "/alpha,/bravo"
}

resource "incapsula_acl_security_rule" "example-global-whitelist-ip-rule" {
  rule_id = "api.acl.whitelisted_ips"
  site_id = "${incapsula_site.example-site.id}"
  ips = "192.168.1.3,192.168.1.4"
}
```

» Argument Reference

The following arguments are supported:

- `site_id` - (Required) Numeric identifier of the site to operate on.
- `rule_id` - (Required) The id of the acl, e.g api.acl.blacklisted_ips. Options are `api.acl.blacklisted_countries`, `api.acl.blacklisted_urls`, `api.acl.blacklisted_ips`, and `api.acl.whitelisted_ips`.
- `continents` - (Optional) A comma separated list of continent codes.
- `countries` - (Optional) A comma separated list of country codes.
- `ips` - (Optional) A comma separated list of IPs or IP ranges, e.g: `192.168.1.1`, `192.168.1.1-192.168.1.100` or `192.168.1.1/24`.

- `urls` - (Optional) A comma separated list of resource paths.
- `url_patterns` - (Optional) The patterns should be in accordance with the matching urls sent by the urls parameter. Options are `CONTAINS`, `EQUALS`, `PREFIX`, `SUFFIX`, `NOT_EQUALS`, `NOT_CONTAIN`, `NOT_PREFIX`,and `NOT_SUFFIX`.

» Attributes Reference

The following attributes are exported:

- `id` - Unique identifier in the API for the ACL security rule.

» incapsula_custom_certificate

Provides a Incapsula Custom Certificate resource. Custom certificates must be one of the following formats: PFX, PEM, or CER.

» Example Usage

```
resource "incapsula_custom_certificate" "custom-certificate" {
    site_id = "${incapsula_site.example-site.id}"
    certificate = "${file("path/to/your/cert.crt")}"
    private_key = "${file("path/to/your/private_key.key")}"
    passphrase = "yourpassphrase"
}
```

» Argument Reference

The following arguments are supported:

- `site_id` - (Required) Numeric identifier of the site to operate on.
- `certificate` - (Required) The certificate file in base64 format. You can use the Terraform HCL `file` directive to pull in the contents from a file. You can also inline the certificate in the configuration.
- `private_key` - (Optional) The private key of the certificate in base64 format. Optional in case of PFX certificate file format.
- `passphrase` - (Optional) The passphrase used to protect your SSL certificate.

» Attributes Reference

The following attributes are exported:

- `id` - At the moment, only one active certificate can be stored. This exported value is always set as `12345`. This will be augmented in future versions of the API.

» incapsula_data_center

Provides a Incapsula Data Center resource.

» Example Usage

```
resource "incapsula_data_center" "example-data-center" {
  site_id = "${incapsula_site.example-site.id}"
  name = "Example data center"
  server_address = "8.8.4.4"
  is_content = "true"
}
```

» Argument Reference

The following arguments are supported:

- `site_id` - (Required) Numeric identifier of the site to operate on.
- `name` - (Required) The new data center's name.
- `server_address` - (Required) The server's address. Possible values: IP, CNAME.
- `is_enabled` - (Optional) Enables the data center.
- `is_standby` - (Optional) Defines the data center as standby for failover.
- `is_content` - (Optional) The data center will be available for specific resources (Forward Delivery Rules).

» Attributes Reference

The following attributes are exported:

- `id` - Unique identifier in the API for the data center.

» incapsula_data_center_server

Provides a Incapsula Data Center Server resource.

» Example Usage

```
resource "incapsula_data_center_server" "example-data-center-server" {
  dc_id = "${incapsula_data_center.example-data-center.id}"
  site_id = "${incapsula_site.example-site.id}"
  server_address = "4.4.4.4"
  is_enabled = "true"
}
```

» Argument Reference

The following arguments are supported:

- `dc_id` - (Required) Numeric identifier of the data center server to operate on.
- `site_id` - (Required) Numeric identifier of the site to operate on.
- `server_address` - (Optional) The server's address.
- `is_standby` - (Optional) Set the server as Active (P0) or Standby (P1).
- `is_enabled` - (Optional) Enables the data center server.

» Attributes Reference

The following attributes are exported:

- `id` - Unique identifier in the API for the data center server.

» incapsula_site

Provides a Incapsula Incap Rule resource. Incap Rules include security, delivery, and rate rules.

» Example Usage

```
resource "incapsula_incap_rule" "example-incap-rule-alert" {
  name = "Example incap rule alert"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_ALERT"
  filter = "Full-URL == \"/someurl\""
}

# Incap Rule: Require javascript support
resource "incapsula_incap_rule" "example-incap-rule-require-js-support" {
```

```
  name = "Example incap rule require javascript support 3"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_INTRUSIVE_HTML"
  filter = "Full-URL == \"/someurl\""
}

# Incap Rule: Block IP
resource "incapsula_incap_rule" "example-incap-rule-block-ip" {
  name = "Example incap rule block ip"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_BLOCK_IP"
  filter = "Full-URL == \"/someurl\""
}

# Incap Rule: Block Request
resource "incapsula_incap_rule" "example-incap-rule-block-request" {
  name = "Example incap rule block request"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_BLOCK"
  filter = "Full-URL == \"/someurl\""
}

# Incap Rule: Block Session
resource "incapsula_incap_rule" "example-incap-rule-block-session" {
  name = "Example incap rule block session"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_BLOCK_USER"
  filter = "Full-URL == \"/someurl\""
}

# Incap Rule: Delete Cookie (ADR)
resource "incapsula_incap_rule" "example-incap-rule-delete-cookie" {
  name = "Example incap rule delete cookie"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_DELETE_COOKIE"
  filter = "Full-URL == \"/someurl\""
  rewrite_name = "my_test_header"
}

# Incap Rule: Delete Header (ADR)
resource "incapsula_incap_rule" "example-incap-rule-delete-header" {
  name = "Example incap rule delete header"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_DELETE_HEADER"
  filter = "Full-URL == \"/someurl\""
  rewrite_name = "my_test_header"
```

```
}

# Incap Rule: Forward to Data Center (ADR)
resource "incapsula_incap_rule" "example-incap-rule-fwd-to-data-center" {
  name = "Example incap rule forward to data center"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_FORWARD_TO_DC"
  filter = "Full-URL == \"/someurl\""
  dc_id = "${incapsula_data_center.example-data-center.id}"
}

# Incap Rule: Redirect (ADR)
resource "incapsula_incap_rule" "example-incap-rule-redirect" {
  name = "Example incap rule redirect"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_REDIRECT"
  filter = "Full-URL == \"/someurl\""
  response_code = "302"
  from = "https://site1.com/url1"
  to = "https://site2.com/url2"
}

# Incap Rule: Require Cookie Support (IncapRule)
resource "incapsula_incap_rule" "example-incap-rule-require-cookie-support" {
  name = "Example incap rule require cookie support"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_RETRY"
  filter = "Full-URL == \"/someurl\""
}

# Incap Rule: Rewrite Cookie (ADR)
resource "incapsula_incap_rule" "example-incap-rule-rewrite-cookie" {
  name = "Example incap rule rewrite cookie"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_REWRITE_COOKIE"
  filter = "Full-URL == \"/someurl\""
  add_missing = "true"
  from = "some_optional_value"
  to = "some_new_value"
  rewrite_name = "my_cookie_name"
}

# Incap Rule: Rewrite Header (ADR)
resource "incapsula_incap_rule" "example-incap-rule-rewrite-header" {
  name = "Example incap rule rewrite header"
  site_id = "${incapsula_site.example-site.id}"
```

```
  action = "RULE_ACTION_REWRITE_HEADER"
  filter = "Full-URL == \"/someurl\""
  add_missing = "true"
  from = "some_optional_value"
  to = "some_new_value"
  rewrite_name = "my_test_header"
}

# Incap Rule: Rewrite URL (ADR)
resource "incapsula_incap_rule" "example-incap-rule-rewrite-url" {
  name = "ExampleRewriteURL"
  site_id = "${incapsula_site.example-site.id}"
  action = "RULE_ACTION_REWRITE_URL"
  filter = "Full-URL == \"/someurl\""
  add_missing = "true"
  from = "*"
  to = "/redirect"
  rewrite_name = "my_test_header"
}
```

» Argument Reference

The following arguments are supported:

- `site_id` - (Required) Numeric identifier of the site to operate on.
- `name` - (Required) Rule name.
- `action` - (Required) Rule action. See the detailed descriptions in the API documentation. Possible values: `RULE_ACTION_REDIRECT`, `RULE_ACTION_SIMPLIFIED_REDIRECT`, `RULE_ACTION_REWRITE_URL`, `RULE_ACTION_REWRITE_HEADER`, `RULE_ACTION_REWRITE_COOKIE`, `RULE_ACTION_DELETE_HEADER`, `RULE_ACTION_DELETE_COOKIE`, `RULE_ACTION_RESPONSE_REWRITE_HEADER` `RULE_ACTION_RESPONSE_DELETE_HEADER`, `RULE_ACTION_RESPONSE_REWRITE_RESPONSE_CODE`, `RULE_ACTION_FORWARD_TO_DC`, `RULE_ACTION_ALERT`, `RULE_ACTION_BLOCK`, `RULE_ACTION_BLOCK_USER`, `RULE_ACTION_BLOCK_IP`, `RULE_ACTION_RETRY`, `RULE_ACTION_INTRUSIVE_HTML`, `RULE_ACTION_CAPTCHA`, `RULE_ACTION_RATE`, `RULE_ACTION_CUSTOM_ERROR_RESPONSE`.
- `filter` - (Required) The filter defines the conditions that trigger the rule action. For action `RULE_ACTION_SIMPLIFIED_REDIRECT` filter is not relevant. For other actions, if left empty, the rule is always run.
- `response_code` - (Optional) For `RULE_ACTION_REDIRECT` or `RULE_ACTION_SIMPLIFIED_REDIRECT` rule's response code, valid values are `302`, `301`, `303`, `307`, `308`. For `RULE_ACTION_RESPONSE_REWRITE_RESPONSE_CODE` rule's response code, valid values are all 3-digits numbers. For `RULE_ACTION_CUSTOM_ERROR_RESPONSE`, valid values are `400`, `401`, `402`, `403`, `404`, `405`, `406`, `407`, `408`, `409`, `410`, `411`, `412`, `413`, `414`, `415`, `416`, `417`, `419`, `420`, `422`, `423`, `424`, `500`, `501`, `502`,

503, 504, 505, 507.

- add_missing - (Optional) Add cookie or header if it doesn't exist (Rewrite cookie rule only).
- from - (Optional) Pattern to rewrite. For RULE_ACTION_REWRITE_URL - Url to rewrite. For RULE_ACTION_REWRITE_HEADER and RULE_ACTION_RESPONSE_REWRITE_HEADER - Header value to rewrite. For RULE_ACTION_REWRITE_COOKIE - Cookie value to rewrite.
- to - (Optional) Pattern to change to. RULE_ACTION_REWRITE_URL - Url to change to. RULE_ACTION_REWRITE_HEADER and RULE_ACTION_RESPONSE_REWRITE_HEADER - Header value to change to. RULE_ACTION_REWRITE_COOKIE - Cookie value to change to.
- rewrite_name - (Optional) Name of cookie or header to rewrite. Applies only for RULE_ACTION_REWRITE_COOKIE, RULE_ACTION_REWRITE_HEADER and RULE_ACTION_RESPONSE_REWRITE_HEADER.
- dc_id - (Optional) Data center to forward request to. Applies only for RULE_ACTION_FORWARD_TO_DC.
- rate_context - (Optional) The context of the rate counter. Possible values IP or Session. Applies only to rules using RULE_ACTION_RATE.
- rate_interval - (Optional) The interval in seconds of the rate counter. Possible values is a multiple of 10; minimum 10 and maximum 300. Applies only to rules using RULE_ACTION_RATE.
- error_type - (Optional) The error that triggers the rule. error.type.all triggers the rule regardless of the error type. Applies only for RULE_ACTION_CUSTOM_ERROR_RESPONSE. Possible values: error.type.all, error.type.connection_timeout, error.type.access_denied, error.type.parse_req_error, error.type.parse_resp_error, error.type.connection_failed, error.type.deny_and_retry, error.type.ssl_failed, error.type.deny_and_captcha, error.type.2fa_required, error.type.no_ssl_config, error.type.no_ipv6_config.
- error_response_format - (Optional) The format of the given error response in the error_response_data field. Applies only for RULE_ACTION_CUSTOM_ERROR_RESPONSE. Possible values: json, xml.
- error_response_data - (Optional) The response returned when the request matches the filter and is blocked. Applies only for RULE_ACTION_CUSTOM_ERROR_RESPONSE.

» Attributes Reference

The following attributes are exported:

- id - Unique identifier in the API for the Incap Rule.

» incapsula_site

Provides a Incapsula Site resource. Sites are the core resource that is required by all other resources.

» Example Usage

```
resource "incapsula_site" "example-site" {
  domain                = "examplesite.com"
  account_id            = "123"
  ref_id                = "123"
  send_site_setup_emails = "false"
  site_ip               = "2.2.2.2"
  force_ssl             = "false"
  log_level             = "full"
  logs_account_id       = "456"
}
```

» Argument Reference

The following arguments are supported:

- `domain` - (Required) The fully qualified domain name of the site. For example: www.example.com, hello.example.com.
- `account_id` - (Optional) The account to operate on. If not specified, operation will be performed on the account identified by the authentication parameters.
- `send_site_setup_emails` - (Optional) If this value is false, end users will not get emails about the add site process such as DNS instructions and SSL setup.
- `site_ip` - (Optional) The web server IP/CNAME.
- `force_ssl` - (Optional) Force SSL. This option is only available for sites with manually configured IP/CNAME and for specific accounts.
- `log_level` - (Optional) Log level. Available only for Enterprise Plan customers that purchased the Logs Integration SKU. Sets the log reporting level for the site. Options are `full`, `security`, `none`, and `default`.
- `logs_account_id` - (Optional) Account where logs should be stored. Available only for Enterprise Plan customers that purchased the Logs Integration SKU. Numeric identifier of the account that purchased the logs integration SKU and which collects the logs. If not specified, operation will be performed on the account identified by the authentication parameters.

» Attributes Reference

The following attributes are exported:

- `id` - Unique identifier in the API for the site.
- `site_creation_date` - Numeric representation of the site creation date.
- `dns_cname_record_name` - The CNAME record name.
- `dns_cname_record_value` - The CNAME record value.
- `dns_a_record_name` - The A record name.
- `dns_a_record_value` - The A record value.

» incapsula_waf_security_rule

Provides a Incapsula WAF Security Rule resource.

» Example Usage

```
resource "incapsula_waf_security_rule" "example-waf-backdoor-rule" {
  site_id = "${incapsula_site.example-site.id}"
  rule_id = "api.threats.backdoor"
  security_rule_action = "api.threats.action.quarantine_url" # (api.threats.action.quarantin
}

resource "incapsula_waf_security_rule" "example-waf-cross-site-scripting-rule" {
  site_id = "${incapsula_site.example-site.id}"
  rule_id = "api.threats.cross_site_scripting"
  security_rule_action = "api.threats.action.block_ip" # (api.threats.action.disabled | api.
}

resource "incapsula_waf_security_rule" "example-waf-illegal-resource-rule" {
  site_id = "${incapsula_site.example-site.id}"
  rule_id = "api.threats.illegal_resource_access"
  security_rule_action = "api.threats.action.block_ip" # (api.threats.action.disabled | api.
}

resource "incapsula_waf_security_rule" "example-waf-remote-file-inclusion-rule" {
  site_id = "${incapsula_site.example-site.id}"
  rule_id = "api.threats.remote_file_inclusion"
  security_rule_action = "api.threats.action.block_ip" # (api.threats.action.disabled | api.
}

resource "incapsula_waf_security_rule" "example-waf-sql-injection-rule" {
  site_id = "${incapsula_site.example-site.id}"
  rule_id = "api.threats.sql_injection"
```

```
  security_rule_action = "api.threats.action.block_ip" # (api.threats.action.disabled | api.
}

resource "incapsula_waf_security_rule" "example-waf-bot-access-control-rule" {
  site_id = "${incapsula_site.example-site.id}"
  rule_id = "api.threats.bot_access_control"
  block_bad_bots = "true" # true | false (optional, default: true)
  challenge_suspected_bots = "true" # true | false (optional, default: true)
}

resource "incapsula_waf_security_rule" "example-waf-ddos-rule" {
  site_id = "${incapsula_site.example-site.id}"
  rule_id = "api.threats.ddos"
  activation_mode = "api.threats.ddos.activation_mode.on" # (api.threats.ddos.activation_mod
  ddos_traffic_threshold = "5000" # valid values are 10, 20, 50, 100, 200, 500, 750, 1000, 2
}
```

» Argument Reference

The following arguments are supported:

- `site_id` - (Required) Numeric identifier of the site to operate on.
- `rule_id` - (Required) The identifier of the WAF rule, e.g api.threats.cross_site_scripting.
- `security_rule_action` - (Optional) The action that should be taken when a threat is detected, for example: api.threats.action.block_ip. See above examples for `rule_id` and `action` combinations.
- `activation_mode` - (Optional) The mode of activation for ddos on a site. Possible values: off, auto, on.
- `ddos_traffic_threshold` - (Optional) Consider site to be under DDoS if the request rate is above this threshold. The valid values are 10, 20, 50, 100, 200, 500, 750, 1000, 2000, 3000, 4000, 5000.
- `block_bad_bots` - (Optional) Whether or not to block bad bots. Possible values: true, false.
- `challenge_suspected_bots` - (Optional) Whether or not to send a challenge to clients that are suspected to be bad bots (CAPTCHA for example). Possible values: true, false.

» Attributes Reference

The following attributes are exported:

- `id` - Unique identifier in the API for the Incap Rule.