» panos_system_info

Use this data source to retrieve "show system info" from the NGFW or Panorama.

All contents of "show system info" are saved to the `info` variable. In addition, the version number of PAN-OS encountered is saved to multiple fields for ease of access.

» Example Usage

```
data "panos_system_info" "example" {}
```

» Attribute Reference

The following attributes are present:

- `info` - a map containing the contents of `show system info`.
- `version_major` - Major version number.
- `version_minor` - Minor version number.
- `version_patch` - Patch version number.

» panos_panorama_address_group

This resource allows you to add/update/delete Panorama address groups.

Address groups are either statically defined or dynamically defined, so only `static_addresses` or `dynamic_match` should be defined within a given address group.

» Example Usage

```
# Static group
resource "panos_panorama_address_group" "example1" {
    name = "static ntp grp"
    description = "My NTP servers"
    static_addresses = ["ntp1", "ntp2", "ntp3"]
}

# Dynamic group
resource "panos_panorama_address_group" "example2" {
    name = "dynamic grp"
    description = "My internal NTP servers"
```

```
    dynamic_match = "'internal' and 'ntp'"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The address group's name.
- `device_group` - (Optional) The device group to put the address group into (default: `shared`).
- `static_addresses` - (Optional) The address objects to include in this statically defined address group.
- `dynamic_match` - (Optional) The IP tags to include in this DAG.
- `description` - (Optional) The address group's description.
- `tags` - (Optional) List of administrative tags.

» panos_panorama_address_object

This resource allows you to add/update/delete address objects on Panorama.

» Example Usage

```
resource "panos_panorama_address_object" "example" {
    name = "localnet"
    value = "192.168.80.0/24"
    description = "The 192.168.80 network"
    tags = ["internal", "dmz"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The address object's name.
- `device_group` - (Optional) The device group to put the address object into (default: `shared`).
- `type` - (Optional) The type of address object. This can be `ip-netmask` (default), `ip-range`, or `fqdn`.
- `value` - (Required) The address object's value. This can take various forms depending on what type of address object this is, but can be something like `192.168.80.150` or `192.168.80.0/24`.
- `description` - (Optional) The address object's description.

- `tags` - (Optional) List of administrative tags.

» panos_panorama_administrative_tag

This resource allows you to add/update/delete Panorama administrative tags.

» Example Usage

```
resource "panos_panorama_administrative_tag" "example" {
    name = "tag1"
    color = "color5"
    comment = "Internal resources"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The administrative tag's name.
- `device_group` - (Optional) The device group to put the administrative tag into (default: `shared`).
- `color` - (Optional) The tag's color. This should be either an empty string (no color) or a string such as `color1` or `color15`. Note that for maximum portability, you should limit color usage to `color16`, which was available in PAN-OS 6.1. PAN-OS 8.1's colors go up to `color42`. The value `color18` is reserved internally by PAN-OS and thus not available for use.
- `comment` - (Optional) The administrative tag's description.

» panos_panorama_device_group

This resource allows you to add/update/delete Panorama device groups.

This resource has some overlap with the `panos_panorama_device_group_entry` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_device_group` spec does not define any `device` blocks, and just stays as "computed".

This is the appropriate resource to use if `terraform destroy` should delete the device group.

» Example Usage

```
resource "panos_panorama_device_group" "example" {
    name = "my device group"
    description = "description here"
    device {
        serial = "00112233"
    }
    device {
        serial = "44556677"
        vsys_list = ["vsys1", "vsys2"]
    }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The device group's name.
- `description` - (Optional) The device group's description.
- `device` - The device definition (see below).

The following arguments are valid for each `device` section:

- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

» panos_panorama_device_group_entry

This resource allows you to add/update/delete a specific device in a Panorama device group.

This resource has some overlap with the `panos_panorama_device_group` resource. If you want to use this resource with the other one, then make sure that your `panos_panorama_device_group` spec does not define any `device` blocks, and just stays as "computed".

This is the appropriate resource to use if you have a pre-existing device group in Panorama and don't want Terraform to delete it on `terraform destroy`.

An interesting side effect of the underlying XML API - if the device group does not already exist, then this resource can actually create it. However, since only the single entry for the specific serial number is deleted, then a `terraform destroy` would not remove the device group itself in this situation.

» Example Usage

```
# Example for a virtual firewall.
resource "panos_panorama_device_group_entry" "example1" {
    device_group = "my device group"
    serial = "00112233"
}

# Example for a physical firewall with multi-vsys enabled.
resource "panos_panorama_device_group_entry" "example2" {
    device_group = "my device group"
    serial = "44556677"
    vsys_list = ["vsys1", "vsys2"]
}
```

» Argument Reference

The following arguments are supported:

- `device_group` - (Required) The device group's name.
- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

» panos_panorama_nat_policy

This resource allows you to add/update/delete Panorama NAT policies.

The prefix `sat` stands for "Source Address Translation" while the prefix "dat" stands for "Destination Address Translation". The order of the params in this resource and their naming matches how the params are presented in the GUI. Thus, having a GUI window open while creating your resource definition will simplify the process.

Note that while many of the params for this resource are optional in an absolute sense, depending on what type of NAT you wish to configure, certain params may become necessary to correctly configure the NAT policy.

» Example Usage

```
resource "panos_panorama_nat_policy" "example" {
    name = "my nat policy"
    source_zones = ["zone1"]
```

```
    destination_zone = "zone2"
    to_interface = "ethernet1/3"
    source_addresses = ["any"]
    destination_addresses = ["any"]
    sat_type = "none"
    dat_type = "static"
    dat_address = "my dat address object"
    target {
        serial = "123456"
        vsys_list = ["vsys1", "vsys2"]
    }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The NAT policy's name.
- `device_group` - (Optional) The device group to put the NAT policy into (default: `shared`).
- `rulebase` - (Optional) The rulebase. This can be `pre-rulebase` (default), `post-rulebase`, or `rulebase`.
- `description` - (Optional) The description.
- `type` - (Optional). NAT type. This can be `ipv4` (default), `nat64`, or `nptv6`.
- `source_zones` - (Required) The list of source zone(s).
- `destination_zone` - (Required) The destination zone.
- `to_interface` - (Optional) Egress interface from route lookup (default: `any`).
- `service` - (Optional) Service (default: `any`).
- `source_addresses` - (Required) List of source address(es).
- `destination_addresses` - (Required) List of destination address(es).
- `sat_type` - (Optional) Type of source address translation. This can be `none` (default), `dynamic-ip-and-port`, `dynamic-ip`, or `static-ip`.
- `sat_address_type` - (Optional) Source address translation address type.
- `sat_translated_addresses` - (Optional) Source address translation list of translated addresses.
- `sat_interface` - (Optional) Source address translation interface.
- `sat_ip_address` - (Optional) Source address translation IP address.
- `sat_fallback_type` - (Optional) Source address translation fallback type. This can be `none`, `interface-address`, or `translated-address`.
- `sat_fallback_translated_addresses` - (Optional) Source address translation list of fallback translated addresses.
- `sat_fallback_interface` - (Optional) Source address translation fallback interface.

- `sat_fallback_ip_type` - (Optional) Source address translation fallback IP type. This can be `ip` or `floating`.
- `sat_fallback_ip_address` - (Optional) The source address translation fallback IP address.
- `sat_static_translated_address` - (Optional) The statically translated source address.
- `sat_static_bi_directional` - (Optional) Set to `true` to enable bi-directional source address translation.
- `dat_type` - (Optional) Destination address translation type. This should be either `static` or `dynamic`. The `dynamic` option is only available on PAN-OS 8.1+.
- `dat_address` - (Optional) Destination address translation's address. Requires `dat_type` be set to "static" or "dynamic".
- `dat_port` - (Optional) Destination address translation's port number. Requires `dat_type` be set to "static" or "dynamic".
- `dat_dynamic_distribution` - (Optional, PAN-OS 8.1+) Distribution algorithm for destination address pool. The PAN-OS 8.1 GUI doesn't seem to set this anywhere, but this is added here for completeness' sake. Requires `dat_type` of "dynamic".
- `disabled` - (Optional) Set to `true` to disable this rule.
- `tags` - (Optional) List of administrative tags.
- `target` - (Optional) A target definition (see below). If there are no target sections, then the policy will apply to every vsys of every device in the device group.
- `negate_target` - (Optional, bool) Instead of applying the policy for the given serial numbers, apply it to everything except them.

The following arguments are valid for each `target` section:

- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

» panos_panorama_security_policies

This resource allows you to add/update/delete Panorama security policies.

This resource manages the full set of security policies, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block. As this manages the full set of security policies for a given rulebase, any extraneous rules are removed on `terraform apply`.

For each security policy, there are three styles of profile settings:

- `None` (the default)

- `Group`
- `Profiles`

The Profile Setting is implicitly chosen based on what params are configured for the security policy. If you want a Profile Setting of `Group`, then the `group` param should be set to the desired Group Profile. If you want a Profile Setting of `Profiles`, then you will need to specify one or more of the following params:

- `virus`
- `spyware`
- `vulnerability`
- `url_filtering`
- `file_blocking`
- `wildfire_analysis`
- `data_filtering`

If the `group` param and none of the `Profiles` params are specified, then the Profile Setting is set to `None`.

» Example Usage

```
resource "panos_panorama_security_policies" "example" {
    rule {
        name = "allow bizdev to dmz"
        source_zones = ["bizdev"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["dmz"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "allow"
    }
    rule {
        name = "deny sales to eng"
        source_zones = ["sales"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["eng"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
```

```
        action = "deny"
        target {
            serial = "01234"
        }
        target {
            serial = "56789"
            vsys_list = ["vsys1", "vsys3"]
        }
    }
}
```

» Argument Reference

The following arguments are supported:

- `device_group` - (Optional) The device group to put the security policy into (default: `shared`).
- `rulebase` - (Optional) The rulebase. This can be `pre-rulebase` (default), `post-rulebase`, or `rulebase`.
- `rule` - The security policy definition (see below). The security policy ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security policy name.
- `type` - (Optional) Rule type. This can be `universal` (default), `interzone`, or `intrazone`.
- `description` - (Optional) The description.
- `tags` - (Optional) List of tags for this security rule.
- `source_zones` - (Required) List of source zones.
- `source_addresses` - (Required) List of source addresses.
- `negate_source` - (Optional, bool) If the source should be negated.
- `source_users` - (Required) List of source users.
- `hip_profiles` - (Required) List of HIP profiles.
- `destination_zones` - (Required) List of destination zones.
- `destination_addresses` - (Required) List of destination addresses.
- `negate_destination` - (Optional, bool) If the destination should be negated.
- `applications` - (Required) List of applications.
- `services` - (Required) List of services.
- `categories` - (Required) List of categories.
- `action` - (Optional) Action for the matched traffic. This can be `allow` (default), `deny`, `drop`, `reset-client`, `reset-server`, or `reset-both`.
- `log_setting` - (Optional) Log forwarding profile.
- `log_start` - (Optional, bool) Log the start of the traffic flow.
- `log_end` - (Optional, bool) Log the end of the traffic flow (default: `true`).

- `disabled` - (Optional, bool) Set to `true` to disable this rule.
- `schedule` - (Optional) The security policy schedule.
- `icmp_unreachable` - (Optional) Set to `true` to enable ICMP unreachable.
- `disable_server_response_inspection` - (Optional) Set to `true` to disable server response inspection.
- `group` - (Optional) Profile Setting: `Group` - The group profile name.
- `virus` - (Optional) Profile Setting: `Profiles` - The antivirus setting.
- `spyware` - (Optional) Profile Setting: `Profiles` - The anti-spyware setting.
- `vulnerability` - (Optional) Profile Setting: `Profiles` - The Vulnerability Protection setting.
- `url_filtering` - (Optional) Profile Setting: `Profiles` - The URL filtering setting.
- `file_blocking` - (Optional) Profile Setting: `Profiles` - The file blocking setting.
- `wildfire_analysis` - (Optional) Profile Setting: `Profiles` - The WildFire Analysis setting.
- `data_filtering` - (Optional) Profile Setting: `Profiles` - The Data Filtering setting.
- `target` - (Optional) A target definition (see below). If there are no target sections, then the policy will apply to every vsys of every device in the device group.
- `negate_target` - (Optional, bool) Instead of applying the policy for the given serial numbers, apply it to everything except them.

The following arguments are valid for each `target` section:

- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

» panos_panorama_security_policy_group

This resource allows you to add/update/delete Panorama security policy groups.

This resource manages clusters of security policies in a single device group, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Because this resource only manages what it's told to, it will not manage any policies that may already exist on Panorama. This has implications on the effective security posture of Panorama, but it will allow you to spread your security policies across multiple Terraform state files. If you want to verify that the security policies are only what appears in the plan file, then you should probably be using the panos_panorama_security_policies resource.

Although you cannot modify non-group security policies with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other security policy that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

For each security policy, there are three styles of profile settings:

- `None` (the default)
- `Group`
- `Profiles`

The Profile Setting is implicitly chosen based on what params are configured for the security policy. If you want a Profile Setting of `Group`, then the `group` param should be set to the desired Group Profile. If you want a Profile Setting of `Profiles`, then you will need to specify one or more of the following params:

- `virus`
- `spyware`
- `vulnerability`
- `url_filtering`
- `file_blocking`
- `wildfire_analysis`
- `data_filtering`

If the `group` param and none of the `Profiles` params are specified, then the Profile Setting is set to `None`.

» Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as ”top” or ”directly below”, otherwise they’ll keep shoving each other out of the way indefinitely.

Best practices are to specify one group as `top` (if you need it), one group as `bottom` (this is where you have your logging deny policies), then all other groups should be `above` the first policy of the bottom group. You do it this way because rules will natually be added at the tail end of the rulebase, so they will always be `after` the first group, but what you want is for them to be `before` the last group’s policies.

» Example Usage

```
resource "panos_panorama_security_policy_group" "example" {
    position_keyword = "above"
    position_reference = "deny everything else"
    rule {
```

```
        name = "allow bizdev to dmz"
        source_zones = ["bizdev"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["dmz"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "allow"
    }
    rule {
        name = "deny sales to eng"
        source_zones = ["sales"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["eng"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "deny"
        target {
            serial = "01234"
        }
        target {
            serial = "56789"
            vsys_list = ["vsys1", "vsys3"]
        }
    }
}
```

» Argument Reference

The following arguments are supported:

- `device_group` - (Optional) The device group to put the security policy into (default: `shared`).
- `rulebase` - (Optional) The rulebase. This can be `pre-rulebase` (default), `post-rulebase`, or `rulebase`.
- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This

param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group policy to use as a reference to place this group.
- `rule` - The security policy definition (see below). The security policy ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security policy name.
- `type` - (Optional) Rule type. This can be `universal` (default), `interzone`, or `intrazone`.
- `description` - (Optional) The description.
- `tags` - (Optional) List of tags for this security rule.
- `source_zones` - (Required) List of source zones.
- `source_addresses` - (Required) List of source addresses.
- `negate_source` - (Optional, bool) If the source should be negated.
- `source_users` - (Required) List of source users.
- `hip_profiles` - (Required) List of HIP profiles.
- `destination_zones` - (Required) List of destination zones.
- `destination_addresses` - (Required) List of destination addresses.
- `negate_destination` - (Optional, bool) If the destination should be negated.
- `applications` - (Required) List of applications.
- `services` - (Required) List of services.
- `categories` - (Required) List of categories.
- `action` - (Optional) Action for the matched traffic. This can be `allow` (default), `deny`, `drop`, `reset-client`, `reset-server`, or `reset-both`.
- `log_setting` - (Optional) Log forwarding profile.
- `log_start` - (Optional, bool) Log the start of the traffic flow.
- `log_end` - (Optional, bool) Log the end of the traffic flow (default: `true`).
- `disabled` - (Optional, bool) Set to `true` to disable this rule.
- `schedule` - (Optional) The security policy schedule.
- `icmp_unreachable` - (Optional) Set to `true` to enable ICMP unreachable.
- `disable_server_response_inspection` - (Optional) Set to `true` to disable server response inspection.
- `group` - (Optional) Profile Setting: `Group` - The group profile name.
- `virus` - (Optional) Profile Setting: `Profiles` - The antivirus setting.
- `spyware` - (Optional) Profile Setting: `Profiles` - The anti-spyware setting.
- `vulnerability` - (Optional) Profile Setting: `Profiles` - The Vulnerability Protection setting.
- `url_filtering` - (Optional) Profile Setting: `Profiles` - The URL filtering setting.
- `file_blocking` - (Optional) Profile Setting: `Profiles` - The file blocking setting.
- `wildfire_analysis` - (Optional) Profile Setting: `Profiles` - The Wild-

Fire Analysis setting.
- `data_filtering` - (Optional) Profile Setting: `Profiles` - The Data Filtering setting.
- `target` - (Optional) A target definition (see below). If there are no target sections, then the policy will apply to every vsys of every device in the device group.
- `negate_target` - (Optional, bool) Instead of applying the policy for the given serial numbers, apply it to everything except them.

The following arguments are valid for each `target` section:

- `serial` - (Required) The serial number of the firewall.
- `vsys_list` - (Optional) A subset of all available vsys on the firewall that should be in this device group. If the firewall is a virtual firewall, then this parameter should just be omitted.

» panos_panorama_service_group

This resource allows you to add/update/delete Panorama service groups.

» Example Usage

```
resource "panos_panorama_service_group" "example" {
    name = "static ntp grp"
    services = ["svc1", "svc2"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The service group's name.
- `device_group` - (Optional) The device group to put the service group into (default: `shared`).
- `services` - (Required) List of services to put in this service group.
- `tags` - (Optional) List of administrative tags.

» panos_panorama_service_object

This resource allows you to add/update/delete Panorama service objects.

» Example Usage

```
resource "panos_panorama_service_object" "example" {
    name = "my_service"
    protocol = "tcp"
    description = "My service object"
    source_port = "2000-2049,2051-2099"
    destination_port = "32123"
    tags = ["internal", "dmz"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The service object's name.
- `device_group` - (Optional) The device group to put the service object into (default: `shared`).
- `description` - (Optional) The service object's description.
- `protocol` - (Required) The service's protocol. This should be `tcp` or `udp`.
- `source_port` - (Optional) The source port. This can be a single port number, range (1-65535), or comma separated (80,8080,443).
- `destination_port` - (Required) The destination port. This can be a single port number, range (1-65535), or comma separated (80,8080,443).
- `tags` - (Optional) List of administrative tags.

» panos_address_group

This resource allows you to add/update/delete address groups.

Address groups are either statically defined or dynamically defined, so only `static_addresses` or `dynamic_match` should be defined within a given address group.

» Example Usage

```
# Static group
resource "panos_address_group" "example1" {
    name = "static ntp grp"
    description = "My NTP servers"
    static_addresses = ["ntp1", "ntp2", "ntp3"]
}
```

```
# Dynamic group
resource "panos_address_group" "example2" {
    name = "dynamic grp"
    description = "My internal NTP servers"
    dynamic_match = "'internal' and 'ntp'"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The address group's name.
- `vsys` - (Optional) The vsys to put the address group into (default: `vsys1`).
- `static_addresses` - (Optional) The address objects to include in this statically defined address group.
- `dynamic_match` - (Optional) The IP tags to include in this DAG.
- `description` - (Optional) The address group's description.
- `tags` - (Optional) List of administrative tags.

» panos_address_object

This resource allows you to add/update/delete address objects.

» Example Usage

```
resource "panos_address_object" "example" {
    name = "localnet"
    value = "192.168.80.0/24"
    description = "The 192.168.80 network"
    tags = ["internal", "dmz"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The address object's name.
- `vsys` - (Optional) The vsys to put the address object into (default: `vsys1`).
- `type` - (Optional) The type of address object. This can be `ip-netmask` (default), `ip-range`, or `fqdn`.
- `value` - (Required) The address object's value. This can take various forms depending on what type of address object this is, but can be something like `192.168.80.150` or `192.168.80.0/24`.

- `description` - (Optional) The address object's description.
- `tags` - (Optional) List of administrative tags.

» panos_administrative_tag

This resource allows you to add/update/delete administrative tags.

» Example Usage

```
resource "panos_administrative_tag" "example" {
    name = "tag1"
    vsys = "vsys2"
    color = "color5"
    comment = "Internal resources"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The administrative tag's name.
- `vsys` - (Optional) The vsys to put the administrative tag into (default: `vsys1`).
- `color` - (Optional) The tag's color. This should be either an empty string (no color) or a string such as `color1` or `color15`. Note that for maximum portability, you should limit color usage to `color16`, which was available in PAN-OS 6.1. PAN-OS 8.1's colors go up to `color42`. The value `color18` is reserved internally by PAN-OS and thus not available for use.
- `comment` - (Optional) The administrative tag's description.

» panos_dag_tags

This resource allows you to add and remove dynamic address group tags.

The `ip` field should be unique in the `panos_dag_tags` block, and there should only be one `panos_dag_tags` block defined in a given plan.

Note - Tags are only removed during `terraform destroy`. Updating an applied terraform plan to have alternative tags will leave behind the old tags from the previously published plan(s).

» Example Usage

```
resource "panos_dag_tags" "example" {
    vsys = "vsys1"
    register {
        ip = "10.1.1.1"
        tags = ["tag1", "tag2"]
    }
    register {
        ip = "10.1.1.2"
        tags = ["tag3"]
    }
}
```

» Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys to put the DAG tags in (default: `vsys1`).
- `register` - (Required) A set that includes `ip`, the IP address to be tagged and `tags`, a list of tags to associate with the given IP.

» panos_ethernet_interface

This resource allows you to add/update/delete ethernet interfaces.

» Example Usage

```
# Configure a bare-bones ethernet interface.
resource "panos_ethernet_interface" "example1" {
    name = "ethernet1/3"
    vsys = "vsys1"
    mode = "layer3"
    static_ips = ["10.1.1.1/24"]
    comment = "Configured for internal traffic"
}

# Configure a DHCP ethernet interface for vsys1 to use.
resource "panos_ethernet_interface" "example2" {
    name = "ethernet1/4"
    vsys = "vsys1"
    mode = "layer3"
    enable_dhcp = true
```

```
    create_dhcp_default_route = true
    dhcp_default_route_metric = 10
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The ethernet interface's name. This should be something like `ethernet1/X`.
- `vsys` - (Required) The vsys that will use this interface. This should be something like `vsys1` or `vsys3`.
- `mode` - (Required) The interface mode. This can be any of the following values: `layer3`, `layer2`, `virtual-wire`, `tap`, `ha`, `decrypt-mirror`, or `aggregate-group`.
- `static_ips` - (Optional) List of static IPv4 addresses to set for this data interface.
- `enable_dhcp` - (Optional) Set to `true` to enable DHCP on this interface.
- `create_dhcp_default_route` - (Optional) Set to `true` to create a DHCP default route.
- `dhcp_default_route_metric` - (Optional) The metric for the DHCP default route.
- `ipv6_enabled` - (Optional) Set to `true` to enable IPv6.
- `management_profile` - (Optional) The management profile.
- `mtu` - (Optional) The MTU.
- `adjust_tcp_mss` - (Optional) Adjust TCP MSS (default: false).
- `netflow_profile` - (Optional) The netflow profile.
- `lldp_enabled` - (Optional) Enable LLDP (default: false).
- `lldp_profile` - (Optional) LLDP profile.
- `link_speed` - (Optional) Link speed. This can be any of the following: `10`, `100`, `1000`, or `auto`.
- `link_duplex` - (Optional) Link duplex setting. This can be `full`, `half`, or `auto`.
- `link_state` - (Optional) The link state. This can be `up`, `down`, or `auto`.
- `aggregate_group` - (Optional) The aggregate group (applicable for physical firewalls only).
- `comment` - (Optional) The interface comment.
- `ipv4_mss_adjust` - (Optional, PAN-OS 8.0+) The IPv4 MSS adjust value.
- `ipv6_mss_adjust` - (Optional, PAN-OS 8.0+) The IPv6 MSS adjust value.

» panos_general_settings

This resource allows you to update the general device settings, such as DNS or the hostname.

All params are optional for this resource. If any options are not specified, then whatever is already configured on the firewall is left as-is. The general device settings will always exist on the firewall, so `terraform destroy` does not remove config from the firewall.

» Example Usage

```
resource "panos_general_settings" "example" {
    hostname = "ngfw220"
    dns_primary = "10.5.1.10"
    ntp_primary = "10.5.1.10"
    ntp_primary_auth_type = "none"
}
```

» Argument Reference

The following arguments are supported:

- `hostname` - Firewall hostname.
- `timezone` - The timezone (e.g. - `US/Pacific`).
- `domain` - The domain.
- `update_server` - The update server (Default: `updates.paloaltonetworks.com`).
- `verify_update_server` - Verify update server identity (Default: `true`).
- `dns_primary` - Primary DNS server.
- `dns_secondary` - Secondary DNS server.
- `ntp_primary_address` - Primary NTP server.
- `ntp_primary_auth_type` - Primary NTP auth type. This can be `none`, `autokey`, or `symmetric-key`.
- `ntp_primary_key_id` - Primary NTP `symmetric-key` key ID.
- `ntp_primary_algorithm` - Primary NTP `symmetric-key` algorithm. This can be `sha1` or `md5`.
- `ntp_primary_auth_key` - Primary NTP `symmetric-key` auth key. This is the SHA1 hash if the algorithm is `sha1`, or the md5sum if the algorithm is `md5`.
- `ntp_secondary_address` - Secondary NTP server.
- `ntp_secondary_auth_type` - Secondary NTP auth type. This can be `none`, `autokey`, or `symmetric-key`.
- `ntp_secondary_key_id` - Secondary NTP `symmetric-key` key ID.
- `ntp_secondary_algorithm` - Secondary NTP `symmetric-key` algorithm. This can be `sha1` or `md5`.
- `ntp_secondary_auth_key` - Secondary NTP `symmetric-key` auth key. This is the SHA1 hash if the algorithm is `sha1`, or the md5sum if the algorithm is `md5`.

» panos_management_profile

This resource allows you to add/update/delete interface management profiles.

» Example Usage

```
resource "panos_management_profile" "example" {
    name = "allow ping"
    ping = true
    permitted_ips = ["10.1.1.0/24", "192.168.80.0/24"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The management profile's name.
- `ping` - (Optional) Allow ping.
- `telnet` - (Optional) Allow telnet.
- `ssh` - (Optional) Allow SSH.
- `http` - (Optional) Allow HTTP.
- `http_ocsp` - (Optional) Allow HTTP OCSP.
- `https` - (Optional) Allow HTTPS.
- `snmp` - (Optional) Allow SNMP.
- `response_pages` - (Optional) Allow response pages.
- `userid_service` - (Optional) Allow User ID service.
- `userid_syslog_listener_ssl` - (Optional) Allow User ID syslog listener for SSL.
- `userid_syslog_listener_udp` - (Optional) Allow User ID syslog listener for UDP.
- `permitted_ips` - (Optional) The list of permitted IP addresses or address ranges for this management profile.

» panos_nat_policy

This resource allows you to add/update/delete NAT policies.

The prefix `sat` stands for "Source Address Translation" while the prefix "dat" stands for "Destination Address Translation". The order of the params in this resource and their naming matches how the params are presented in the GUI. Thus, having a GUI window open while creating your resource definition will simplify the process.

Note that while many of the params for this resource are optional in an absolute sense, depending on what type of NAT you wish to configure, certain params may become necessary to correctly configure the NAT policy.

» Example Usage

```
resource "panos_nat_policy" "example" {
    name = "my nat policy"
    source_zones = ["zone1"]
    destination_zone = "zone2"
    to_interface = "ethernet1/3"
    source_addresses = ["any"]
    destination_addresses = ["any"]
    sat_type = "none"
    dat_type = "static"
    dat_address = "my dat address object"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The NAT policy's name.
- `vsys` - (Optional) The vsys to put the NAT policy into (default: `vsys1`).
- `rulebase` - (Optional, Deprecated) The rulebase. For firewalls, there is only the `rulebase` value (default), but on Panorama, there is also `pre-rulebase` and `post-rulebase`.
- `description` - (Optional) The description.
- `type` - (Optional). NAT type. This can be `ipv4` (default), `nat64`, or `nptv6`.
- `source_zones` - (Required) The list of source zone(s).
- `destination_zone` - (Required) The destination zone.
- `to_interface` - (Optional) Egress interface from route lookup (default: `any`).
- `service` - (Optional) Service (default: `any`).
- `source_addresses` - (Required) List of source address(es).
- `destination_addresses` - (Required) List of destination address(es).
- `sat_type` - (Optional) Type of source address translation. This can be `none` (default), `dynamic-ip-and-port`, `dynamic-ip`, or `static-ip`.
- `sat_address_type` - (Optional) Source address translation address type.
- `sat_translated_addresses` - (Optional) Source address translation list of translated addresses.
- `sat_interface` - (Optional) Source address translation interface.
- `sat_ip_address` - (Optional) Source address translation IP address.

- `sat_fallback_type` - (Optional) Source address translation fallback type. This can be `none`, `interface-address`, or `translated-address`.
- `sat_fallback_translated_addresses` - (Optional) Source address translation list of fallback translated addresses.
- `sat_fallback_interface` - (Optional) Source address translation fallback interface.
- `sat_fallback_ip_type` - (Optional) Source address translation fallback IP type. This can be `ip` or `floating`.
- `sat_fallback_ip_address` - (Optional) The source address translation fallback IP address.
- `sat_static_translated_address` - (Optional) The statically translated source address.
- `sat_static_bi_directional` - (Optional) Set to `true` to enable bi-directional source address translation.
- `dat_type` - (Optional) Destination address translation type. This should be either `static` or `dynamic`. The `dynamic` option is only available on PAN-OS 8.1+.
- `dat_address` - (Optional) Destination address translation's address. Requires `dat_type` be set to "static" or "dynamic".
- `dat_port` - (Optional) Destination address translation's port number. Requires `dat_type` be set to "static" or "dynamic".
- `dat_dynamic_distribution` - (Optional, PAN-OS 8.1+) Distribution algorithm for destination address pool. The PAN-OS 8.1 GUI doesn't seem to set this anywhere, but this is added here for completeness' sake. Requires `dat_type` of "dynamic".
- `disabled` - (Optional) Set to `true` to disable this rule.
- `tags` - (Optional) List of administrative tags.

» panos_security_policies

This resource allows you to add/update/delete security policies.

This resource manages the full set of security policies, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

For each security policy, there are three styles of profile settings:

- `None` (the default)
- `Group`
- `Profiles`

The Profile Setting is implicitly chosen based on what params are configured for the security policy. If you want a Profile Setting of `Group`, then the `group` param should be set to the desired Group Profile. If you want a Profile Setting of `Profiles`, then you will need to specify one or more of the following params:

- virus
- spyware
- vulnerability
- url_filtering
- file_blocking
- wildfire_analysis
- data_filtering

If the group param and none of the Profiles params are specified, then the Profile Setting is set to None.

» Example Usage

```
resource "panos_security_policies" "example" {
    rule {
        name = "allow bizdev to dmz"
        source_zones = ["bizdev"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["dmz"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "allow"
    }
    rule {
        name = "deny sales to eng"
        source_zones = ["sales"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["eng"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "deny"
    }
}
```

» Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys to put the security policy into (default: `vsys1`).
- `rulebase` - (Optional, Deprecated) The rulebase. For firewalls, there is only the `rulebase` value (default), but on Panorama, there is also `pre-rulebase` and `post-rulebase`.
- `rule` - The security policy definition (see below). The security policy ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security policy name.
- `type` - (Optional) Rule type. This can be `universal` (default), `interzone`, or `intrazone`.
- `description` - (Optional) The description.
- `tags` - (Optional) List of tags for this security rule.
- `source_zones` - (Required) List of source zones.
- `source_addresses` - (Required) List of source addresses.
- `negate_source` - (Optional, bool) If the source should be negated.
- `source_users` - (Required) List of source users.
- `hip_profiles` - (Required) List of HIP profiles.
- `destination_zones` - (Required) List of destination zones.
- `destination_addresses` - (Required) List of destination addresses.
- `negate_destination` - (Optional, bool) If the destination should be negated.
- `applications` - (Required) List of applications.
- `services` - (Required) List of services.
- `categories` - (Required) List of categories.
- `action` - (Optional) Action for the matched traffic. This can be `allow` (default), `deny`, `drop`, `reset-client`, `reset-server`, or `reset-both`.
- `log_setting` - (Optional) Log forwarding profile.
- `log_start` - (Optional, bool) Log the start of the traffic flow.
- `log_end` - (Optional, bool) Log the end of the traffic flow (default: `true`).
- `disabled` - (Optional, bool) Set to `true` to disable this rule.
- `schedule` - (Optional) The security policy schedule.
- `icmp_unreachable` - (Optional) Set to `true` to enable ICMP unreachable.
- `disable_server_response_inspection` - (Optional) Set to `true` to disable server response inspection.
- `group` - (Optional) Profile Setting: `Group` - The group profile name.
- `virus` - (Optional) Profile Setting: `Profiles` - The antivirus setting.
- `spyware` - (Optional) Profile Setting: `Profiles` - The anti-spyware setting.
- `vulnerability` - (Optional) Profile Setting: `Profiles` - The Vulnerability Protection setting.
- `url_filtering` - (Optional) Profile Setting: `Profiles` - The URL filtering

setting.
- `file_blocking` - (Optional) Profile Setting: `Profiles` - The file blocking setting.
- `wildfire_analysis` - (Optional) Profile Setting: `Profiles` - The WildFire Analysis setting.
- `data_filtering` - (Optional) Profile Setting: `Profiles` - The Data Filtering setting.

» panos_security_policy_group

This resource allows you to add/update/delete security policy groups.

This resource manages clusters of security policies in a single vsys, enforcing both the contents of individual rules as well as their ordering. Rules are defined in a `rule` config block.

Because this resource only manages what it's told to, it will not manage any policies that may already exist on the firewall. This has implications on the effective security posture of your firewall, but it will allow you to spread your security policies across multiple Terraform state files. If you want to verify that the security policies are only what appears in the plan file, then you should probably be using the panos_security_policies resource.

Although you cannot modify non-group security policies with this resource, the `position_keyword` and `position_reference` parameters allow you to reference some other security policy that already exists, using it as a means to ensure some rough placement within the ruleset as a whole.

For each security policy, there are three styles of profile settings:

- `None` (the default)
- `Group`
- `Profiles`

The Profile Setting is implicitly chosen based on what params are configured for the security policy. If you want a Profile Setting of `Group`, then the `group` param should be set to the desired Group Profile. If you want a Profile Setting of `Profiles`, then you will need to specify one or more of the following params:

- `virus`
- `spyware`
- `vulnerability`
- `url_filtering`
- `file_blocking`
- `wildfire_analysis`
- `data_filtering`

If the `group` param and none of the `Profiles` params are specified, then the Profile Setting is set to `None`.

» Best Practices

As is to be expected, if you are separating your deployment across multiple plan files, make sure that at most only one plan specifies any given absolute positioning keyword such as "top" or "directly below", otherwise they'll keep shoving each other out of the way indefinitely.

Best practices are to specify one group as `top` (if you need it), one group as `bottom` (this is where you have your logging deny policies), then all other groups should be `above` the first policy of the bottom group. You do it this way because rules will natually be added at the tail end of the rulebase, so they will always be `after` the first group, but what you want is for them to be `before` the last group's policies.

» Example Usage

```
resource "panos_security_policy_group" "example" {
    position_keyword = "above"
    position_reference = "deny everything else"
    rule {
        name = "allow bizdev to dmz"
        source_zones = ["bizdev"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["dmz"]
        destination_addresses = ["any"]
        applications = ["any"]
        services = ["application-default"]
        categories = ["any"]
        action = "allow"
    }
    rule {
        name = "deny sales to eng"
        source_zones = ["sales"]
        source_addresses = ["any"]
        source_users = ["any"]
        hip_profiles = ["any"]
        destination_zones = ["eng"]
        destination_addresses = ["any"]
        applications = ["any"]
```

```
        services = ["application-default"]
        categories = ["any"]
        action = "deny"
    }
}
```

» Argument Reference

The following arguments are supported:

- `vsys` - (Optional) The vsys to put the security policy into (default: `vsys1`).
- `position_keyword` - (Optional) A positioning keyword for this group. This can be `before`, `directly before`, `after`, `directly after`, `top`, `bottom`, or left empty (the default) to have no particular placement. This param works in combination with the `position_reference` param.
- `position_reference` - (Optional) Required if `position_keyword` is one of the "above" or "below" variants, this is the name of a non-group policy to use as a reference to place this group.
- `rule` - The security policy definition (see below). The security policy ordering will match how they appear in the terraform plan file.

The following arguments are valid for each `rule` section:

- `name` - (Required) The security policy name.
- `type` - (Optional) Rule type. This can be `universal` (default), `interzone`, or `intrazone`.
- `description` - (Optional) The description.
- `tags` - (Optional) List of tags for this security rule.
- `source_zones` - (Required) List of source zones.
- `source_addresses` - (Required) List of source addresses.
- `negate_source` - (Optional, bool) If the source should be negated.
- `source_users` - (Required) List of source users.
- `hip_profiles` - (Required) List of HIP profiles.
- `destination_zones` - (Required) List of destination zones.
- `destination_addresses` - (Required) List of destination addresses.
- `negate_destination` - (Optional, bool) If the destination should be negated.
- `applications` - (Required) List of applications.
- `services` - (Required) List of services.
- `categories` - (Required) List of categories.
- `action` - (Optional) Action for the matched traffic. This can be `allow` (default), `deny`, `drop`, `reset-client`, `reset-server`, or `reset-both`.
- `log_setting` - (Optional) Log forwarding profile.
- `log_start` - (Optional, bool) Log the start of the traffic flow.
- `log_end` - (Optional, bool) Log the end of the traffic flow (default: `true`).
- `disabled` - (Optional, bool) Set to `true` to disable this rule.

- `schedule` - (Optional) The security policy schedule.
- `icmp_unreachable` - (Optional) Set to `true` to enable ICMP unreachable.
- `disable_server_response_inspection` - (Optional) Set to `true` to disable server response inspection.
- `group` - (Optional) Profile Setting: `Group` - The group profile name.
- `virus` - (Optional) Profile Setting: `Profiles` - The antivirus setting.
- `spyware` - (Optional) Profile Setting: `Profiles` - The anti-spyware setting.
- `vulnerability` - (Optional) Profile Setting: `Profiles` - The Vulnerability Protection setting.
- `url_filtering` - (Optional) Profile Setting: `Profiles` - The URL filtering setting.
- `file_blocking` - (Optional) Profile Setting: `Profiles` - The file blocking setting.
- `wildfire_analysis` - (Optional) Profile Setting: `Profiles` - The WildFire Analysis setting.
- `data_filtering` - (Optional) Profile Setting: `Profiles` - The Data Filtering setting.

» panos_service_group

This resource allows you to add/update/delete service groups.

» Example Usage

```
resource "panos_service_group" "example" {
    name = "static ntp grp"
    services = ["svc1", "svc2"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The service group's name.
- `vsys` - (Optional) The vsys to put the service group into (default: `vsys1`).
- `services` - (Required) List of services to put in this service group.
- `tags` - (Optional) List of administrative tags.

» panos_service_object

This resource allows you to add/update/delete service objects.

» Example Usage

```
resource "panos_service_object" "example" {
    name = "my_service"
    vsys = "vsys1"
    protocol = "tcp"
    description = "My service object"
    source_port = "2000-2049,2051-2099"
    destination_port = "32123"
    tags = ["internal", "dmz"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The service object's name.
- `vsys` - (Optional) The vsys to put the service object into (default: `vsys1`).
- `description` - (Optional) The service object's description.
- `protocol` - (Required) The service's protocol. This should be `tcp` or `udp`.
- `source_port` - (Optional) The source port. This can be a single port number, range (1-65535), or comma separated (80,8080,443).
- `destination_port` - (Required) The destination port. This can be a single port number, range (1-65535), or comma separated (80,8080,443).
- `tags` - (Optional) List of administrative tags.

» panos_telemetry

This resource allows you to add/update/delete telemetry sharing.

Join other Palo Alto Networks customers in a global sharing community, helping to raise the bar against the latest attack techniques. Your participation allows us to deliver new threat prevention controls across the attack lifecycle. Choose the type of data you share across applications, threat intelligence, and device health information to improve the fidelity of the protections we deliver. This is an opt-in feature controlled with granular policy, and we encourage you to join the community.

» Example Usage

```
resource "panos_telemetry" "example" {
    threat_prevention_reports = true
    threat_prevention_data = true
    threat_prevention_packet_captures = true
}
```

» Argument Reference

The following arguments are supported:

- `application_reports` - (Bool, optional) Application reports.
- `threat_prevention_reports` - (Bool, optional) Threat reports.
- `url_reports` - (Bool, optional) URL reports.
- `file_type_identification_reports` - (Bool, optional) File type identification reports.
- `threat_prevention_data` - (Bool, optional) Threat prevention data.
- `threat_prevention_packet_captures` - (Bool, optional) Enable sending packet- captures with threat prevention information. This requires that `threat_prevention_data` also be enabled.
- `product_usage_stats` - (Bool, optional) Health and performance reports.
- `passive_dns_monitoring` - (Bool, optional) Passive DNS monitoring.

» panos_virtual_router

This resource allows you to add/update/delete virtual routers.

Note - The `default` virtual router may be configured with this resource, however it will not be deleted from the firewall. It will only be unexported from the vsys that it is currently imported in, and any interfaces imported into the virtual router will be removed.

» Example Usage

```
# Configure a bare-bones ethernet interface.
resource "panos_virtual_router" "example" {
    name = "my virtual router"
    static_dist = 15
    interfaces = ["ethernet1/1", "ethernet1/2"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The virtual router's name.
- `vsys` - (Required) The vsys that will use this virtual router. This should be something like `vsys1` or `vsys3`.
- `interfaces` - (Optional) List of interfaces that should use this virtual router.
- `static_dist` - (Optional) Admin distance - Static (default: `10`).
- `static_ipv6_dist` - (Optional) Admin distance - Static IPv6 (default: `10`).
- `ospf_int_dist` - (Optional) Admin distance - OSPF Int (default: `30`).
- `ospf_ext_dist` - (Optional) Admin distance - OSPF Ext (default: `110`).
- `ospfv3_int_dist` - (Optional) Admin distance - OSPFv3 Int (default: `30`).
- `ospfv3_ext_dist` - (Optional) Admin distance - OSPFv3 Ext (default: `110`).
- `ibgp_dist` - (Optional) Admin distance - IBGP (default: `200`).
- `ebgp_dist` - (Optional) Admin distance - EBGP (default: `20`).
- `rip_dist` - (Optional) Admin distance - RIP (default: `120`).

» panos_zone

This resource allows you to add/update/delete zones.

» Example Usage

```
resource "panos_zone" "example" {
    name = "my_service"
    mode = "layer3"
    interfaces = ["ethernet1/1", "ethernet1/2"]
    enable_user_id = true
    exclude_acls = ["192.168.0.0/16"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The zone's name.
- `vsys` - (Optional) The vsys to put the zone into (default: `vsys1`).

- `mode` - (Required) The zone's mode. This can be `layer3`, `layer2`, `virtual-wire`, `tap`, or `tunnel`.
- `zone_profile` - (Optional) The zone protection profile.
- `log_setting` - (Optional) Log setting.
- `enable_user_id` - (Optional) Boolean to enable user identification.
- `interfaces` - (Optional) List of interfaces to associated with this zone.
- `include_acls` - (Optional) Users from these addresses/subnets will be identified. This can be an address object, an address group, a single IP address, or an IP address subnet.
- `exclude_acls` - (Optional) Users from these addresses/subnets will not be identified. This can be an address object, an address group, a single IP address, or an IP address subnet.