Lists

About Lists

Lists can be used to create and maintain sets of data for use when creating rules. Lists allow you to easily reuse the same sets of data across multiple rules. Lists can be created on individual sites (Site Lists) as well as the corp as a whole (Corp Lists) to be easily used in multiple sites.

For example, you could create a list of prohibited countries that you don't do business with. You could then use this list in any rules that involve those countries, such as rules to track registration or login attempts originating from those countries. If a prohibited country changes, simply update the list instead of updating every rule that uses it.

Lists can consist of the following types of data:

- Countries
- IP addresses
- Strings
- Wildcards

> **Note:** Lists support CIDR notation for IP address ranges.

Creating a List

Create both Corp and Site lists using these steps.

Corp Lists

1. From the **Corp Rules** menu, select **Corp Lists**. The corp lists menu page appears.
2. Click **Add corp list**. The add corp list menu page appears.
3. From the **Type** menu, select the type of data the list will contain.
4. In the **Name** field, enter the name of the list.
5. Optionally, in the **Description (optional)** field, enter a description for the list.
6. In the **Entries** field, enter the items that will comprise the list. Each entry must be on its own line.
7. Click **Create corp list**.

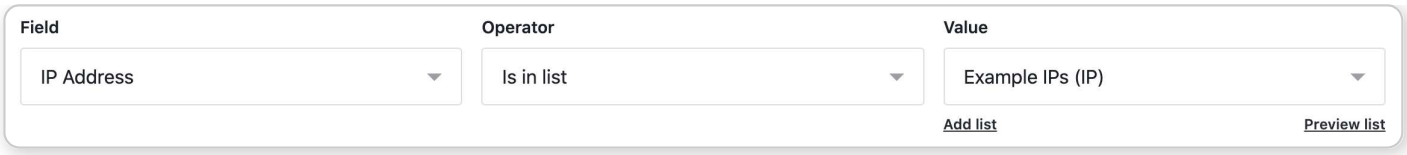
> **Note:** Only Owners can create, edit, and delete Corp Lists. This is because Corp Lists have the ability to manipulate traffic across every site and other user types can only manage Rules and Lists for sites they have access to.

Site Lists

1. From the **Site Rules** menu, select **Site Lists**. The site lists menu page appears.
2. Click **New list**. The new list menu page appears.
3. From the **Type** menu, select the type of data the list will contain.
4. In the **Name** field, enter the name of the list.
5. Optionally, in the **Description (optional)** field, enter a description for the list.
6. In the **Entries** field, enter the items that will comprise the list. Each entry must be on its own line.
7. Click **Create site list**.

Using a List

When creating a rule, select **Is in list** or **Is not in list** for the operator, then select the list from the value dropdown menu.



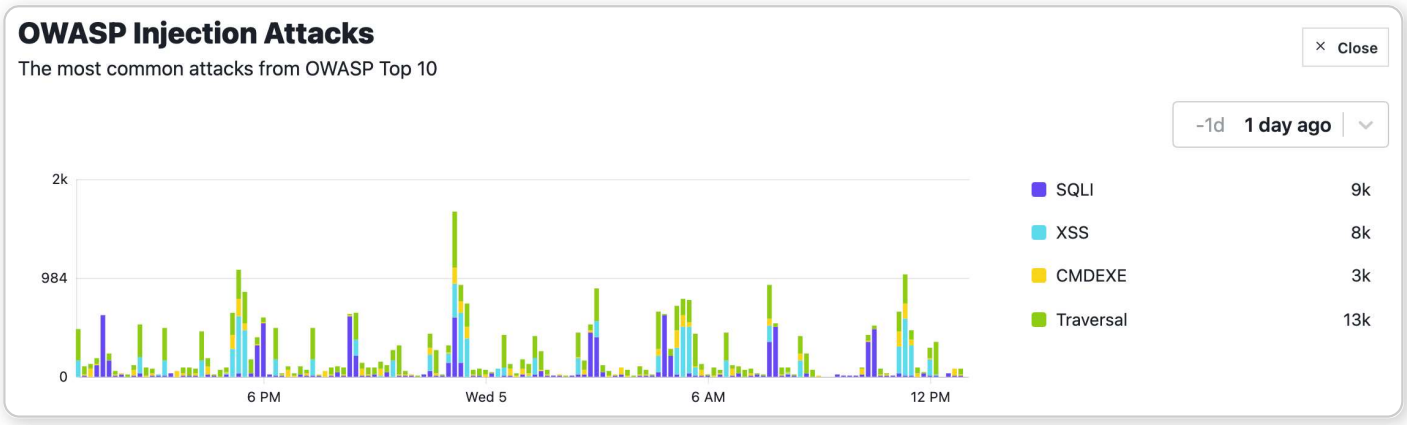
For more information about creating rules, see Rules.

Data Storage and Sampling

When our agent sends requests to our collectors, we store two types of data: **timeseries data** and **individual request data**.

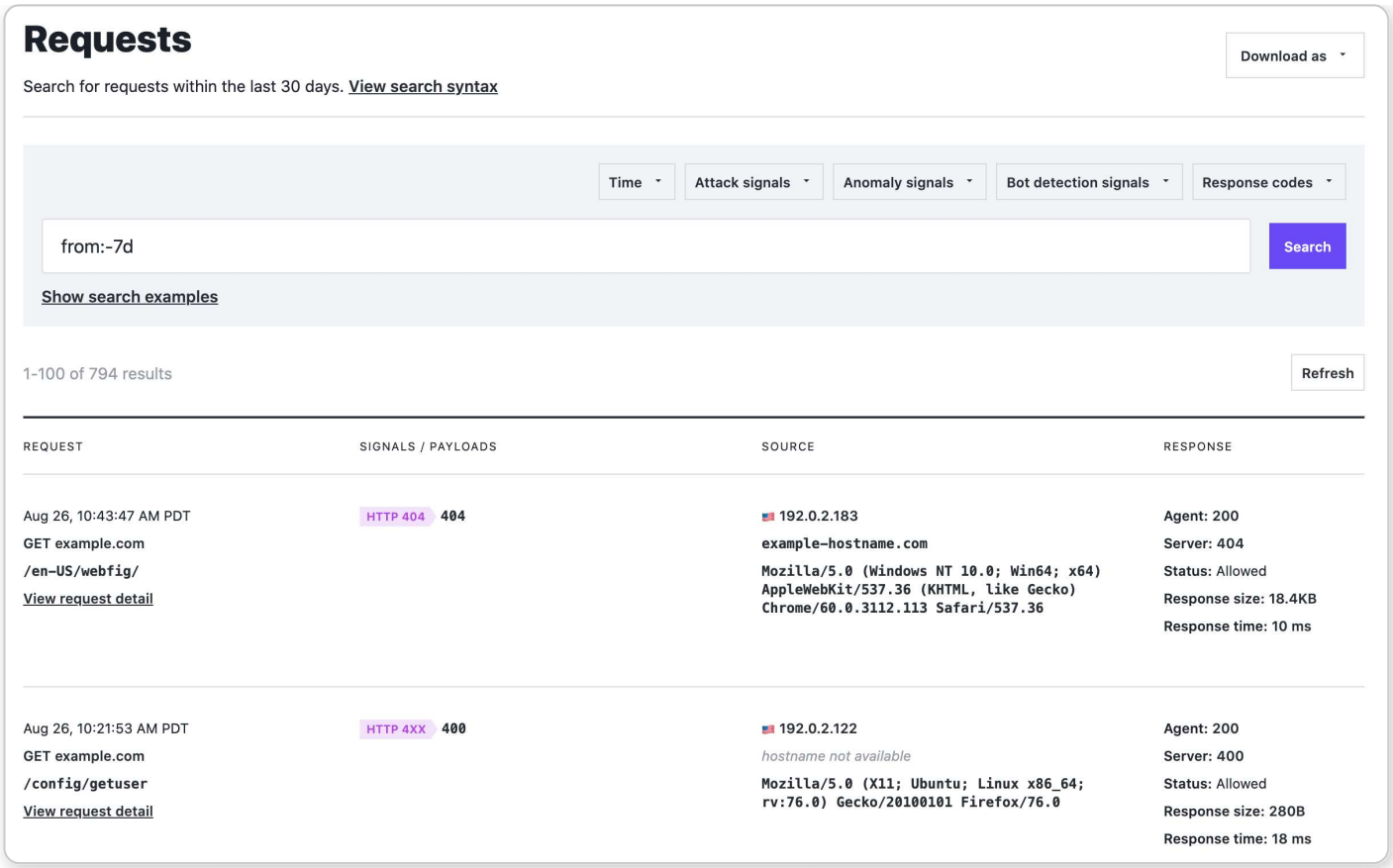
Timeseries data

Timeseries data counts the number of signals (e.g., XSS, SQLi, 404s) observed per minute, while individual request data includes individual records of anonymized requests. Timeseries data powers graphs visible throughout the product, as well as metrics such as tallies of request types.



Individual request data

While all timeseries data is stored and available in the product, a representative sample of individual request data is stored. Individual request data provides detailed information about specific requests, such as the originating IP address and request parameters:



What data does Signal Sciences store?

We store all timeseries data sent to our collectors (powering graphs and metrics throughout the product).

We store individual request data based on the type of signals that requests are tagged with or the way that custom rules are configured. Storage categories include:

- **All:** all requests matching this storage category are stored and available for reference throughout the console.
- **Sampled:** a random sample of requests matching this storage category will be stored and available for reference throughout the console.
- **Timeseries only:** requests matching this storage category aren't stored. Timeseries data for all signals tagged to the request will be stored and visible.
- **Not stored:** requests matching this category aren't stored.

> **Note:** Timeseries-only data storage category is only available on agents 3.12 and above. Matching requests processed on earlier agents will be processed according to the Sampled data storage category.

Request signal type	Description	Storage category
Individual requests containing attack signals	Any requests containing 1 or more attack signals (e.g., SQLi, XSS)	All
Individual requests containing CVE signals	Any requests containing 1 or more CVE signals applied by virtual patching rules	All
Individual requests containing only anomaly signals	Requests that contain only anomaly signals (e.g., 404, Tor traffic) but no attack or CVE signals	Sampled
Individual requests containing custom signals	Requests containing custom signals but no attack or CVE signals. See Custom Signals for more information about creating and using signals.	Sampled
Individual requests containing only API or ATO templated rules signals, known as informational signals	Requests which are tagged with only a specific set of API or ATO templated rules signals, and no custom, anomaly, attack, or CVE signals	Timeseries only
Individual requests that aren't tagged with a signal	Requests containing no signals	Not stored

> **Note:** Any requests containing at least one attack or CVE signal will be stored,

> including requests that also have anomaly, informational, or custom signals.

Heroku Install

The Signal Sciences agent can be deployed with Heroku. The installation process is compatible with any of the language buildpacks.

Installation

1. Log in to Heroku.

```
heroku login
```

2. Add the Signal Sciences buildpack to your application settings.

```
heroku buildpacks:add --index 1 https://dl.signalsciences.net/sigsci-heroku-buildpack/
```

> **Note:** The Signal Sciences buildpack must run first or before your application's primary buildpack.

3. In your `Procfile` file, add `sigsci/bin/sigsci-start` so it precedes your existing start command:

```
web: sigsci/bin/sigsci-start YOUR-APPLICATION-START-COMMAND
```

Example:

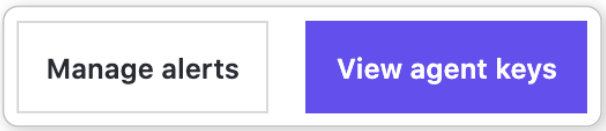
```
web: sigsci/bin/sigsci-start node index.js
```

4. Locate the **Agent Keys** for your Signal Sciences site:

   a. Log in to the Signal Sciences console.

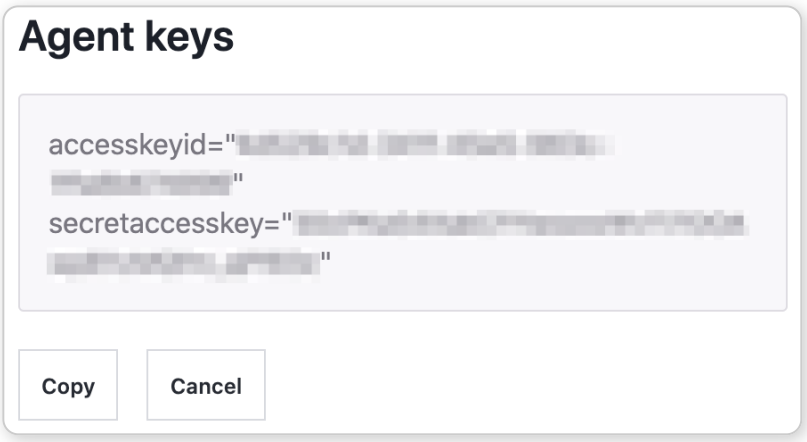
   b. Select a site if you have more than one site.

   c. Click **Agents** in the navigation bar. The agents page appears.



   d. Click **View agent keys**. The agent keys window appears.

   e. Copy the **Agent Access Key** and **Agent Secret Key**.



5. Add the Signal Sciences agent keys to your application's environment variables.

```
heroku config:set SIGSCI_ACCESSKEYID=access-key-goes-here
heroku config:set SIGSCI_SECRETACCESSKEY=secret-key-goes-here
```

6. Deploy your application. Heroku applications are typically deployed with the following commands:

```
git add .
git commit -m "my comment here"
git push heroku main
```

Configuration

- Each time you deploy your application, Heroku will automatically assign a new random name for the agent. An agent name for each deployment can be specified by setting the `SIGSCI_SERVER_HOSTNAME` environment variable:

```
heroku config:set SIGSCI_SERVER_HOSTNAME=agent-name
```

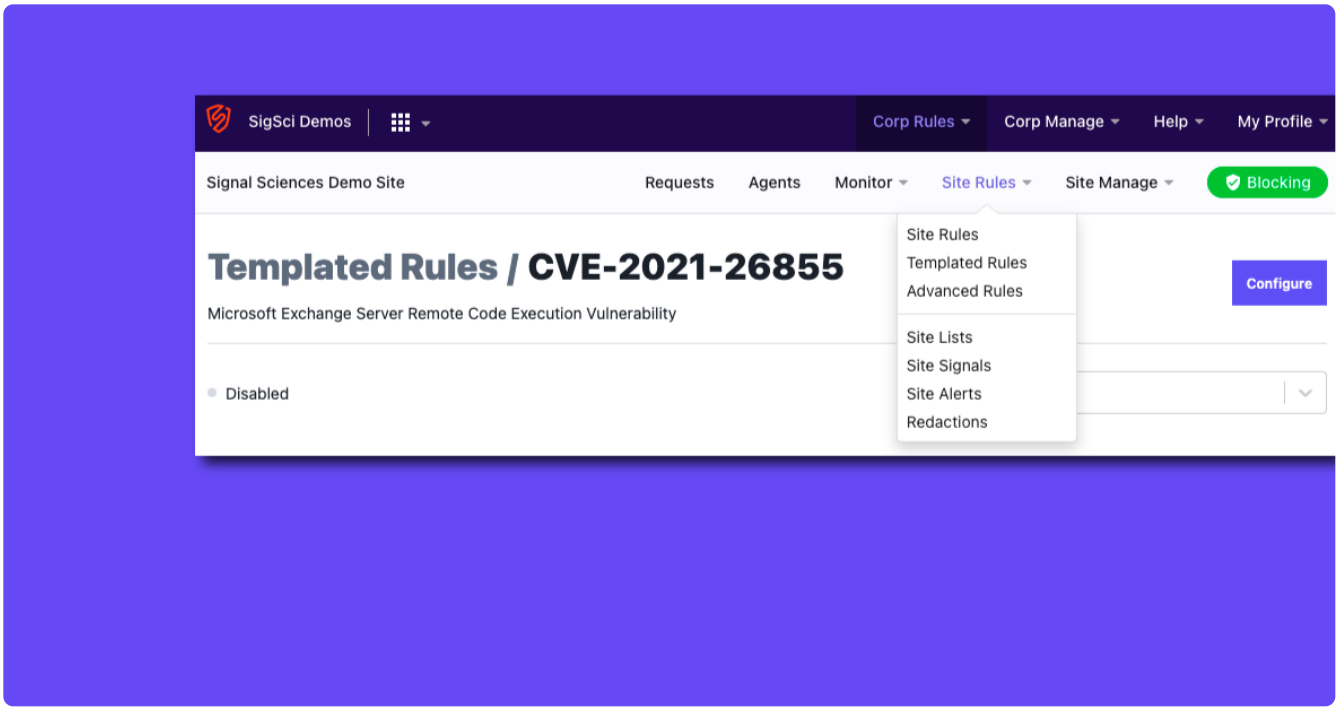
- Agent access logging can be enabled by setting the `SIGSCI_REVERSE_PROXY_ACCESSLOG` environment variable:

```
heroku config:set SIGSCI_REVERSE_PROXY_ACCESSLOG /tmp/sigsci_access.log
```

- The buildpack will install the latest version of the Signal Sciences agent by default. You can specify which agent version to install by setting the `SIGSCI_AGENT_VERSION` environment variable:

```
heroku config:set SIGSCI_AGENT_VERSION=1.15.3
```

Additional configuration options are listed on the agent configuration page.

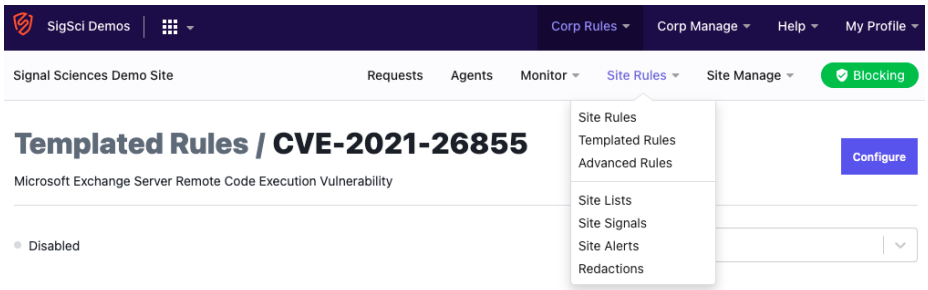


INFORMATION SECURITY, WEB APPLICATION SECURITY

Next-Gen WAF protection for recent Microsoft Exchange vulnerabilities

Protecting our customers

Our security research team has built and deployed a rule to protect Fastly's Signal Sciences Next-Gen WAF customers against the recently announced Microsoft Exchange Server vulnerabilities. The custom rule is available in the console under "Templated Rules".



We strongly suggest that customers using Signal Sciences Next-Gen WAF in front of their Microsoft Exchange servers enable this rule as soon as possible and configure it to block requests if the signal is observed. Additionally, follow all guidance from Microsoft to patch affected systems. The vulnerabilities in question are actively being exploited globally and have severe impact.

Patching Microsoft Exchange systems

We are seeing a large uptick in exploitation attempts in the wild. This is an evolving story and our teams are working continuously to ensure the rules are catching the latest attacks, but this should not be your only line of defense. We strongly recommend that you patch affected systems, perform incident response, and follow recommendations from Microsoft.

Exploit chain

The observed attacks on Microsoft Exchange systems chain together multiple CVEs (Common Vulnerabilities and Exposures) to carry out the attack. The impact of these attacks range from full system takeover through Remote Code Execution (RCE), as well as email inbox exfiltration and compromise. At a high level, the exploit chain is carried out as follows:

1. A Server-Side Request Forgery (SSRF) vulnerability in Microsoft Exchange Server identified as CVE-2021-26855 allows attackers to send HTTP requests to the exposed Exchange server and access other endpoints as the Exchange server itself. This is an unauthenticated step of the attack which makes the vulnerability exceptionally easy to exploit.
2. An insecure deserialization vulnerability identified by CVE-2021-26857 leverages the SYSTEM-level authentication obtained by the above SSRF attack to send specially-crafted SOAP payloads which are insecurely deserialized by the Unified Messaging Service. This gives the attacker the ability to run code as SYSTEM on the Exchange server.

3. After CVE-2021-26855 is successfully exploited, attackers can then utilize CVE-2021-27065 and CVE-2021-26858 to write arbitrary files to the Exchange server itself on any path. This code that is uploaded by the attacker is run as SYSTEM on the server. Lateral movement, malware implanting, data loss, escalation, and more can be carried out through these vulnerabilities.

By enabling the Signal Sciences Next-Gen WAF templated rule, the first step in the exploit chain cannot be carried out. If you would like to dig deeper into the technical details of this chain of attacks please see this post by the folks at Praetorian. To enable the templated rule, please refer to our documentation for details on how to enable templated rules.