Check for malicious URLs

Check your cloud app for malicious URLs

The code samples below were created using the Pangea sample apps. If you want to follow along, check out the Pangea sample apps on GitHub:

- Golang
- Node.js
- Python
- Java

Create a token

Expand for details

Create a token so that you can access the URL Intel endpoints:

1. Go to the Pangea Console and click **URL Intel** in the left-hand navigation menu. The URL Intel Overview page will appear.
2. On the URL Intel Overview page, you'll see a notification asking you to set a service token. Click **Create new token** toward the bottom right side of your screen.
3. You'll be prompted to create a token. Enter a **Token name** and select an **Expiration Date**. You may also create a token for all Intel services, if you wish.
4. Once configured, the token is available in the Tokens section of the URL Intel Overview page.

Select a provider

Expand for details

Providers can be selected as default in the Pangea Console. Setting a provider as default in the Pangea Console means your API request calls will use this provider, unless another provider is specified as part of your API request.

To select a provider as default for an API:

1. Go to the Pangea Console
2. On the left-hand navigation menu, select **URL Intel**
3. Go to **Settings**
4. Click **Set as default** for your preferred provider

> **Tip**
>
> You can override the default provider by specifying their name in the `provider` field when making an API request to the `reputation` endpoint. This is helpful if your default provider returns a verdict of `Unknown` and you want a second opinion from another provider.

Configure your app for communication with the Pangea service

For your app to communicate with the Pangea service, you must set the following environment variables:

- `token`
- `domain`

All of these variables are created when you enable URL Intel and can be found in the **Overview** section under **URL Intel**.

Set environment variables

To set each variable in bash:

```
export PANGEA_DOMAIN="yourServiceDomain"
```

Note

Pangea services are cloud agnostic and deployed regionally, so service endpoints may vary.

```
export URL_INTEL_AUTH_TOKEN="yourAccessToken"
```

Send URL to URL Intel service

A `reputation` call from your app to the URL Intel service might look like this:

LANGUAGE Python JavaScript Go

```python
import os

import pangea.exceptions as pe
from pangea.config import PangeaConfig
from pangea.services import UrlIntel

token = os.getenv("PANGEA_URL_INTEL_TOKEN")
domain = os.getenv("PANGEA_DOMAIN")
config = PangeaConfig(domain=domain)
intel = UrlIntel(token, config=config)


def main():

    try:
        response = intel.reputation(
            url="http://113.235.101.11:54384",
            provider="crowdstrike",
            verbose=True,
            raw=True,
        )
```

```
        print(f"Response: {response.result}")
    except pe.PangeaAPIException as e:
        print(f"Request Error: {e.response.summary}")
        for err in e.errors:
            print(f"\t{err.detail} \n")


if __name__ == "__main__":
    main()
```

URL Intel API sends a response

After your app submits a URL to the URL Intel service, you will receive the following JSON response:


```
{
  "request_id": "prq_wled5snddz2cpu47nl2ra3gi2phbndrv",
  "request_time": "2022-12-20T23:11:00.241062Z",
  "response_time": "2022-12-20T23:11:00.287125Z",
  "status": "Success",
  "summary": "Url was found",
  "result": {
    "data": {
      "category": ["Not Provided"],
      "score": 100,
      "verdict": "malicious"
    },
    "parameters": {
      "url": "http://113.235.101.11:54384",
      "verbose": true,
      "raw": true,
      "provider": "crowdstrike"
    },
    "raw_data": {
      "indicator": "http://113.235.101.11:54384",
      "type": "url",
      "deleted": false,
      "published": 1604416704,
      "updated": 1609924025,
      "malware_families": [],
      "kill_chains": ["C2"],
      "ip_address_types": [],
```

```
      "domain_types": [],
      "confidence": "high",
      "labels": [
        {
          "name": "MaliciousConfidence/High",
          "created_on": 1604416704,
          "last_valid_on": 1605383137
        },
        {
          "name": "Actor/MUMMYSPIDER",
          "created_on": 1605383137,
          "last_valid_on": 1605383137
        },
        {
          "name": "KillChain/C2",
          "created_on": 1604416704,
          "last_valid_on": 1605383137
        }
      ],
      "threat_types": [],
      "vulnerabilities": []
    }
  }
}
```


In this instance, the verdict returned as `malicious`. Additional raw data (from the provider specified in the API request) was returned, like:

- `raw_data`
- `parameters`
- `threat_types`
- `vulnerabilities`

Understand and review results

The API response sent by URL Intel includes various fields and values; however, the ones listed below give you the most information about the disposition of a URL. To learn about more response fields, visit the URL Intel API Reference.

Based on the URL Intel API response, it's evident that the URL you submitted is **Malicious.**

`verdict`	The verdict normalized categorization as interpreted by the data returned by the third party provider. There are four possible verdicts: • `Benign` - Confirmed as non-malicious • `Suspicious` - Associated with actions that are malicious • `Malicious` - Confirmed as malicious • `Unknown` - No data
`score`	The normalized score as interpreted by the data returned by the third party provider. Scores are associated with the verdict values listed above: • `0` = `Benign` • `1` - `99` = `Suspicious` • `100` = `Malicious` • `-1` = `Unknown`
`summary`	A summary of the various categories associated with a URL, which help illustrate why a URL received a particular verdict.
`category`	Indicates the category associated with the URL (e.g. Adware, Malware). This field may return more than one category and may, at times, not be populated.
`raw`	Raw data returned by the provider you specified in the API request. You can investigate the raw data if its meaningful to your use case or if you want to supply it to your users. You must set the `raw` field to `true` to receive this data.

Decide what to do with URL

You decide how to respond and/or communicate with your users once a URL's reputation becomes evident. Here are some suggestions:

- Redact or remove the malicious URL from user-provided content
- Block the URL

In this use case, the URL will be blocked and no message will appear for the user to avoid giving them any hints that may help their potentially fraudulent intentions.

Was this article helpful? Yes No Contact us