# Client-side security API


- Endpoints
- API notes
- Common API calls
  - Get client-side security settings
  - Enable client-side security
  - Fetch list of detected scripts
  - Fetch list of infrequently reported scripts
  - Get details of a detected script
  - Fetch list of detected connections
  - Get details of a detected connection
  - Fetch list of detected cookies
  - Get details of a detected cookie
  - Create a content security rule


import { GlossaryTooltip, APIRequest } from "~/components";

You can enable and disable client-side security's resource monitoring, configure settings, and fetch information about detected scripts and connections using the client-side security API (formerly known as Page Shield API).

To authenticate API requests you need an API token. For more information on the required API token permissions, refer to Roles and permissions.

:::note Refer to API deprecations for details on client-side security API changes. :::

# Endpoints

You can obtain the complete endpoint by appending the client-side security API endpoints to the Cloudflare API base URL:

```
https://api.cloudflare.com/client/v4
```

The `{zone_id}` argument is the zone ID (a hexadecimal string). You can find this value in the Cloudflare dashboard or using the Cloudflare API's `/zones` endpoint.

The `{script_id}` argument is the script ID (a hexadecimal string). This value is included in the response of the List client-side security scripts operation for every detected script.

The `{connection_id}` argument is the connection ID (a hexadecimal string). This value is included in the response of the List client-side security connections API operation for every detected connection.

The following table summarizes the available operations:

| Operation | Method + URL stub | Notes |
|---|---|---|
| Get client-side security settings | `GET zones/{zone_id}/page_shield` | Fetch client-side security settings (including the status). |
| Update client-side security settings | `PUT zones/{zone_id}/page_shield` | Update client-side security settings. |
| List client-side security scripts | `GET zones/{zone_id}/page_shield/scripts` | Fetch a list of detected scripts. |
| Get a client-side security script | `GET zones/{zone_id}/page_shield/scripts/{script_id}` | Fetch the details of a script. |
| List client-side security connections | `GET zones/{zone_id}/page_shield/connections` | Fetch a list of detected connections. |
| Get a client-side security connection | `GET zones/{zone_id}/page_shield/connections/{connection_id}` | Fetch the details of a connection. |
| List client-side security cookies | `GET zones/{zone_id}/page_shield/cookies` | Fetch a list of detected cookies. |
| Get a client-side security cookie | `GET zones/{zone_id}/page_shield/cookies/{cookie_id}` | Fetch the details of a cookie. |
| List content security rules | `GET zones/{zone_id}/page_shield/policies` | Fetch a list of all configured content security rules. |
| Get a content security rule | `GET zones/{zone_id}/page_shield/policies/{policy_id}` | Fetch the details of a content security rule. |
| Create a content security rule | `POST zones/{zone_id}/page_shield/policies` | Creates a content security rule with the provided configuration. |
| Update a content security rule | `PUT zones/{zone_id}/page_shield/policies/{policy_id}` | Updates an existing content security rule. |

| Operation | Method + URL stub | Notes |
| --- | --- | --- |
| Delete a content security rule | `DELETE zones/{zone_id}/page_shield/policies/{policy_id}` | Deletes an existing content security rule. |

# API notes

The malicious script classification (`Malicious` or `Not malicious`) is not directly available in the API. To determine this classification, compare the script's `js_integrity_score` value with the classification threshold, which is currently set to 10. Scripts with a score value lower than the threshold are considered malicious.

# Common API calls

## Get client-side security settings

This example obtains the current settings of Cloudflare's client-side security, including the status (enabled/disabled).


```
{
  "result": {
    "enabled": true,
    "updated_at": "2023-05-14T11:47:55.677555Z",
    "use_cloudflare_reporting_endpoint": true,
    "use_connection_url_path": false
  },
  "success": true,
  "errors": [],
  "messages": []
}
```


## Enable client-side security

This example enables Cloudflare's client-side security in the specified zone.

<APIRequest path="/zones/{zone_id}/page_shield" method="PUT" json={{ enabled: true }} />

```
{
  "result": {
    "enabled": true,
    "updated_at": "2023-05-14T11:50:41.756996Z"
  },
  "success": true,
  "errors": [],
  "messages": []
}
```


# Fetch list of detected scripts

This `GET` request fetches a list of scripts detected by Cloudflare's client-side security on hostname `example.net`, requesting the first page with 15 items per page. The URL query string includes filtering and paging parameters.

By default, the response will only include scripts with `active` status when you do not specify a `status` filter parameter in the URL query string.

<APIRequest path="/zones/{zone_id}/page_shield/scripts" parameters={{ hosts: "example.net", page: "1", per_page: "15", }} method="GET" />


```
{
  "result": [
    {
      "id": "8337233faec2357ff84465a919534e4d",
      "url": "https://malicious.example.com/badscript.js",
      "added_at": "2023-05-18T10:51:10.09615Z",
      "first_seen_at": "2023-05-18T10:51:08Z",
      "last_seen_at": "2023-05-22T09:57:54Z",
      "host": "example.net",
      "domain_reported_malicious": false,
      "url_reported_malicious": true,
      "malicious_url_categories": ["Malware"],
      "first_page_url": "http://malicious.example.com/page_one.html",
      "status": "active",
      "url_contains_cdn_cgi_path": false,
      "hash": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",
      "js_integrity_score": 10,
      "obfuscation_score": 10,
```

```
      "dataflow_score": 8,
      "malware_score": 8,
      "cryptomining_score": 9,
      "magecart_score": 8,
      "fetched_at": "2023-05-21T16:58:07Z"
    }
    // (...)
  ],
  "success": true,
  "errors": [],
  "messages": [],
  "result_info": {
    "page": 1,
    "per_page": 15,
    "count": 15,
    "total_count": 24,
    "total_pages": 2
  }
}
```


Some fields displayed in the example response may not be available, depending on your Cloudflare plan.

For details on the available filtering, paging, and sorting parameters, refer to the API reference.

# Fetch list of infrequently reported scripts

This `GET` request fetches a list of infrequently reported scripts on hostname `example.net`, requesting the first page with 15 items per page. The URL query string includes filtering and paging parameters.

<APIRequest path="/zones/{zone_id}/page_shield/scripts" parameters={{ status: "infrequent", hosts: "example.net", page: "1", per_page: "15", }} method="GET" />


```
{
  "result": [
    {
      "id": "83c8da2267394ce8465b74c299658fea",
      "url": "https://scripts.example.com/anotherbadscript.js",
      "added_at": "2023-05-17T13:16:03.419619Z",
```

```
      "first_seen_at": "2023-05-17T13:15:23Z",
      "last_seen_at": "2023-05-18T09:05:20Z",
      "host": "example.net",
      "domain_reported_malicious": false,
      "url_reported_malicious": false,
      "first_page_url": "http://malicious.example.com/page_one.html",
      "status": "infrequent",
      "url_contains_cdn_cgi_path": false,
      "hash": "9245aad577e846dd9b990b1b32425a3fae4aad8b8a28441a8b80084b6bb75a45",
      "js_integrity_score": 48,
      "obfuscation_score": 49,
      "dataflow_score": 45,
      "malware_score": 45,
      "cryptomining_score": 37,
      "magecart_score": 49,
      "fetched_at": "2023-05-18T03:58:07Z"
    }
    // (...)
  ],
  "success": true,
  "errors": [],
  "messages": [],
  "result_info": {
    "page": 1,
    "per_page": 15,
    "count": 15,
    "total_count": 17,
    "total_pages": 2
  }
}
```


Some fields displayed in the example response may not be available, depending on your Cloudflare plan.

For details on the available filtering, paging, and sorting parameters, refer to the API reference.

# Get details of a detected script

This `GET` request obtains the details of a script detected by Cloudflare's client-side security with script ID `8337233faec2357ff84465a919534e4d`.

<APIRequest path="/zones/{zone_id}/page_shield/scripts/{script_id}" method="GET" parameters={{ script_id: "8337233faec2357ff84465a919534e4d", }} />


```
{
  "result": {
    "id": "8337233faec2357ff84465a919534e4d",
    "url": "https://malicious.example.com/badscript.js",
    "added_at": "2023-05-18T10:51:10.09615Z",
    "first_seen_at": "2023-05-18T10:51:08Z",
    "last_seen_at": "2023-05-22T09:57:54Z",
    "host": "example.net",
    "domain_reported_malicious": false,
    "url_reported_malicious": true,
    "malicious_url_categories": ["Malware"],
    "first_page_url": "http://malicious.example.com/page_one.html",
    "status": "active",
    "url_contains_cdn_cgi_path": false,
    "hash": "9245aad577e846dd9b990b1b32425a3fae4aad8b8a28441a8b80084b6bb75a45",
    "js_integrity_score": 48,
    "obfuscation_score": 49,
    "dataflow_score": 45,
    "malware_score": 42,
    "cryptomining_score": 32,
    "magecart_score": 44,
    "fetched_at": "2023-05-21T16:58:07Z",
    "page_urls": [
      "http://malicious.example.com/page_two.html",
      "http://malicious.example.com/page_three.html",
      "http://malicious.example.com/page_four.html"
    ],
    "versions": [
      {
        "hash": "9245aad577e846dd9b990b1b32425a3fae4aad8b8a28441a8b80084b6bb75a45",
        "js_integrity_score": 48,
        "obfuscation_score": 49,
        "dataflow_score": 45,
        "malware_score": 42,
        "cryptomining_score": 32,
        "magecart_score": 44,
        "fetched_at": "2023-05-21T16:58:07Z"
```

```
    }
  ]
},
"success": true,
"errors": [],
"messages": []
}
```

Some fields displayed in the example response may not be available, depending on your Cloudflare plan.

# Fetch list of detected connections

This `GET` request fetches a list of connections detected by Cloudflare's client-side security, requesting the first page with 15 items per page.

By default, the response will only include connections with `active` status when you do not specify a `status` filter parameter in the URL query string.

<APIRequest path="/zones/{zone_id}/page_shield/connections" parameters={{ page: "1", per_page: "15", }} method="GET" />


```
{
"result": [
  {
    "id": "0a7bb628776f4e50a50d8594c4a01740",
    "url": "https://malicious.example.com",
    "added_at": "2022-09-18T10:51:10.09615Z",
    "first_seen_at": "2022-09-18T10:51:08Z",
    "last_seen_at": "2022-09-02T09:57:54Z",
    "host": "example.net",
    "domain_reported_malicious": true,
    "malicious_domain_categories": ["Malware", "Spyware"],
    "url_reported_malicious": false,
    "malicious_url_categories": [],
    "first_page_url": "https://example.net/one.html",
    "status": "active",
    "url_contains_cdn_cgi_path": false
  }
  // (...)
],
```

```
  "success": true,
  "errors": [],
  "messages": [],
  "result_info": {
    "page": 1,
    "per_page": 15,
    "count": 15,
    "total_count": 16,
    "total_pages": 2
  }
}
```


For details on the available filtering, paging, and sorting parameters, refer to the API reference.

# Get details of a detected connection

This `GET` request obtains the details of a connection detected by Cloudflare's client-side security with connection ID `0a7bb628776f4e50a50d8594c4a01740`.

<APIRequest path="/zones/{zone_id}/page_shield/connections/{connection_id}" method="GET" parameters={{ connection_id: "0a7bb628776f4e50a50d8594c4a01740", }} />


```
{
  "result": {
    "id": "0a7bb628776f4e50a50d8594c4a01740",
    "url": "https://malicious.example.com",
    "added_at": "2022-09-18T10:51:10.09615Z",
    "first_seen_at": "2022-09-18T10:51:08Z",
    "last_seen_at": "2022-09-02T09:57:54Z",
    "host": "example.net",
    "domain_reported_malicious": true,
    "malicious_domain_categories": ["Malware", "Spyware"],
    "url_reported_malicious": false,
    "malicious_url_categories": [],
    "first_page_url": "https://example.net/one.html",
    "status": "active",
    "url_contains_cdn_cgi_path": false
  },
  "success": true,
```

```
  "errors": [],
  "messages": []
}
```


# Fetch list of detected cookies

This `GET` request fetches a list of cookies detected by Cloudflare's client-side security, requesting the first page with 15 items per page.

By default, the response will only include cookies with `active` status when you do not specify a `status` filter parameter in the URL query string.

<APIRequest path="/zones/{zone_id}/page_shield/cookies" parameters={{ page: "1", per_page: "15", }} method="GET" />


```
{
  "result": [
    {
      "id": "beee03ada7e047e79f076785d8cd8b8e",
      "type": "first_party",
      "name": "PHPSESSID",
      "host": "example.net",
      "domain_attribute": "example.net",
      "expires_attribute": "2024-10-21T12:28:20Z",
      "http_only_attribute": true,
      "max_age_attribute": null,
      "path_attribute": "/store",
      "same_site_attribute": "strict",
      "secure_attribute": true,
      "first_seen_at": "2024-05-06T10:51:08Z",
      "last_seen_at": "2024-05-07T11:56:01Z",
      "first_page_url": "example.net/store/products",
      "page_urls": ["example.net/store/products/1"]
    }
    // (...)
  ],
  "success": true,
  "errors": [],
  "messages": [],
  "result_info": {
```

```
  "page": 1,
  "per_page": 15,
  "count": 15,
  "total_count": 16,
  "total_pages": 2
 }
}
```


For details on the available filtering, paging, and sorting parameters, refer to Make API calls.

# Get details of a detected cookie

This `GET` request obtains the details of a cookie detected by Cloudflare's client-side security with ID `beee03ada7e047e79f076785d8cd8b8e`.

<APIRequest path="/zones/{zone_id}/page_shield/cookies/{cookie_id}" method="GET" parameters={{ cookie_id: "beee03ada7e047e79f076785d8cd8b8e", }} />


```
{
 "result": {
  "id": "beee03ada7e047e79f076785d8cd8b8e",
  "type": "first_party",
  "name": "PHPSESSID",
  "host": "example.net",
  "domain_attribute": "example.net",
  "expires_attribute": "2024-10-21T12:28:20Z",
  "http_only_attribute": true,
  "max_age_attribute": null,
  "path_attribute": "/store",
  "same_site_attribute": "strict",
  "secure_attribute": true,
  "first_seen_at": "2024-05-06T10:51:08Z",
  "last_seen_at": "2024-05-07T11:56:01Z",
  "first_page_url": "example.net/store/products",
  "page_urls": ["example.net/store/products/1"]
 },
 "success": true,
 "errors": [],
 "messages": []
```

```
}
```

# Create a content security rule

This `POST` request creates a content security rule (previously called a policy) with *Log* action, defining the following scripts as allowed based on where they are hosted:

- Scripts hosted in `myapp.example.com` (which does not include scripts in `example.com`).
- Scripts hosted in `cdnjs.cloudflare.com`.
- The Google Analytics script using its full URL.
- All scripts in the same origin (same HTTP or HTTPS scheme and hostname).

All other scripts would trigger a rule violation, but those scripts would not be blocked.

For more information on Content Security Policy (CSP) directives and values, refer to the MDN documentation.

:::note For a list of CSP directives and keywords supported by content security rules, refer to Supported CSP directives. :::

<APIRequest path="/zones/{zone_id}/page_shield/policies" method="POST" json={{ description: "My first content security rule in log mode", action: "log", expression: 'http.host eq "myapp.example.com"', enabled: "true", value: "script-src myapp.example.com cdnjs.cloudflare.com https://www.google-analytics.com/analytics.js 'self'", }} />

```
{
  "success": true,
  "errors": [],
  "messages": [],
  "result": {
    "id": "<RULE_ID>",
    "description": "My first content security rule in log mode",
    "action": "log",
    "expression": "http.host eq \"myapp.example.com\"",
    "enabled": "true",
    "value": "script-src myapp.example.com cdnjs.cloudflare.com https://www.google-analytics.com/analytics.js 'self'"
  }
}
```

To create a content security rule with an *Allow* action instead of *Log*, use `"action": "allow"` in the request body. In the case of such rule, all scripts not allowed by the rule would be blocked.

---

□□□ #2
□ □□□ □□□ 2026-11-01 00:19:41 UTC
□ □□□ □□□ 2026-11-01 00:19:43 UTC