Securedns Hostcheck

Endpoints Summary

Method	Path	Swagger
`GET`	`/securedns_hostcheck/`	Swagger ↗

> The SecureDNS Host Check API provides real-time hostname filtering and parental control capabilities. This endpoint allows you to verify whether access to specific hostnames should be allowed, refused, or ignored based on your configured security policies and content filtering rules.

Base URL: `https://control.zequenze.com/api/v1`

Authentication: All endpoints require a Bearer token:

```
Authorization: Bearer <your-api-token>
```

Overview

The SecureDNS Host Check API is designed for implementing DNS-based content filtering and parental control systems. This service evaluates hostnames against your configured security policies and returns appropriate actions based on category classifications, access control lists (ACLs), and subnet-specific rules.

Key Features:

- Real-time hostname evaluation against security policies
- Category-based content filtering (malware, adult content, social media, etc.)
- Subnet-specific access control rules
- IP redirection capabilities for blocked content
- Response caching for improved performance
- Detailed logging and audit trails

Common Integration Scenarios:

- DNS resolver integration for enterprise networks
- Parental control software and routers
- Network security appliances
- Content filtering proxies
- Educational institution network protection

The API returns structured responses indicating whether requests should be allowed through, redirected to warning pages, or blocked entirely, along with performance metrics and caching information.

Endpoints

GET /securedns_hostcheck/

Description: Performs real-time evaluation of a hostname against your configured SecureDNS policies. The endpoint checks the requested hostname against category filters, access control lists, and subnet-specific rules to determine the appropriate action. All requests are logged for auditing and reporting purposes.

Use Cases:

- DNS resolver integration to filter malicious or inappropriate content
- Real-time content filtering for corporate networks
- Parental control systems checking web access requests
- Security appliances validating outbound connections
- Educational network protection against inappropriate content

Full URL Example:

```
https://control.zequenze.com/api/v1/securedns_hostcheck/?hostname=example.com&client_ip=192.168.1.100&subnet=192.168.1.0/24
```

Parameters:

Parameter	Type	In	Required	Description
hostname	string	query	Yes	The hostname or domain to check against SecureDNS policies

Parameter	Type	In	Required	Description
client_ip	string	query	Yes	The IP address of the client making the request
subnet	string	query	No	The subnet identifier to apply specific ACL rules
category_override	string	query	No	Override automatic category detection with specific category

cURL Example:

```
curl -X GET "https://control.zequenze.com/api/v1/securedns_hostcheck/?hostname=social-
media.com&client_ip=10.0.1.50&subnet=10.0.1.0/24" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json"
```

Example Response (Allowed):


```
[
  {
    "action": "A",
    "redirect_ip": "0.0.0.0",
    "match_subnet": "10.0.1.0/24",
    "uuid": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
    "category": "business",
    "acl": "corporate_allow",
    "cached": true,
    "response_time": 12.5
  }
]
```


Example Response (Blocked):


```
[
  {
    "action": "R",
    "redirect_ip": "203.0.113.10",
    "match_subnet": "192.168.1.0/24",
    "uuid": "550e8400-e29b-41d4-a716-446655440000",
    "category": "adult_content",
```

```
    "acl": "family_safe",
    "cached": false,
    "response_time": 8.3
  }
]
```


Response Fields:

Field	Type	Description
action	string	Action to take: "A" (Allow), "R" (Refuse/Block), "I" (Ignore)
redirect_ip	string	IP address to redirect blocked requests to (0.0.0.0 for allow)
match_subnet	string	The subnet rule that matched this request
uuid	string	Unique identifier for this policy rule match
category	string	Content category classification (malware, adult_content, social_media, etc.)
acl	string	Access Control List name that was applied
cached	boolean	Whether this response was served from cache
response_time	number	Response time in milliseconds

Response Codes:

Status	Description
200	Rule matched and request allowed - hostname passes all filters
401	Request not authorized - invalid or missing API token
403	Request forbidden - hostname blocked by security policy
404	No rule matched - hostname not found in any policy rules
404	Rules matched and request not allowed - hostname explicitly blocked

---

Common Use Cases

Use Case 1: DNS Resolver Integration

Integrate the SecureDNS API into your DNS resolver to automatically filter malicious domains and inappropriate content. Query each hostname before resolving DNS requests and either allow resolution or redirect to a block page based on the API response.

Use Case 2: Parental Control Router

Implement family-safe internet filtering by checking all outbound web requests through the API. Configure different policies for children's devices versus adult devices using subnet-based rules and redirect blocked content to age-appropriate explanations.

Use Case 3: Corporate Network Security

Protect enterprise networks by filtering access to social media, streaming services, or malware domains during business hours. Use the API to enforce acceptable use policies and maintain productivity while ensuring security compliance.

Use Case 4: Educational Institution Filtering

Deploy content filtering for school networks to block inappropriate content while allowing educational resources. Implement time-based restrictions and category-specific filtering based on different areas of the campus network.

Use Case 5: Security Appliance Integration

Embed the SecureDNS check into network security devices to provide an additional layer of threat protection. Combine with other security feeds to create comprehensive protection against emerging threats and malicious domains.

Best Practices

- **Implement Response Caching**: Use the `cached` field to implement your own local caching layer for frequently requested hostnames to reduce API calls and improve response times
- **Handle All Response Codes**: Implement proper error handling for different HTTP status codes, especially distinguishing between 403 (blocked) and 404 (no rule matched) responses
- **Subnet-Specific Policies**: Leverage subnet parameters to implement different filtering policies for various network segments (guest networks, employee networks, etc.)

- **Monitor Response Times**: Use the `response_time` field to monitor API performance and implement timeout handling for slow responses
- **Log Policy Matches**: Store the returned `uuid` values for audit trails and policy effectiveness analysis
- **Graceful Degradation**: Implement fallback behavior when the API is unavailable - either allow all traffic or apply local blocking rules
- **Rate Limiting**: Implement client-side rate limiting to avoid overwhelming the API during high-traffic periods
- **Batch Processing**: For high-volume scenarios, consider implementing request queuing and batch processing to optimize API usage

---

Revision #4
Created 2026-02-04 05:11:43 UTC by ipena@zequenze.com
Updated 2026-02-11 03:09:39 UTC by ipena@zequenze.com