LOGZILLA DOCUMENTATION

# Squid

Rules, dashboards, and triggers for Squid Proxy

LogZilla App Store · Generated June 11, 2026 · logzilla.ai/docs/logzilla-appstore/squid

# Overview

Squid is a caching and forwarding HTTP web proxy used for speeding up web servers by caching repeated requests, caching web and DNS lookups for network users, and filtering traffic for security. Squid generates access logs that provide visibility into web traffic patterns and proxy performance.

# App Function

- Parse syslog-style Squid logs (forwarded via syslog)
- Parse native Squid access logs (direct file ingestion)
- Extract client IPs, destination servers, HTTP methods, and status codes
- Categorize events with `Event Class: network`
- Map security events to MITRE ATT&CK techniques
- Provide dashboards for traffic analysis and security monitoring

# Vendor Documentation

- Squid Official Website (http://www.squid-cache.org/)
- Squid Documentation (http://www.squid-cache.org/Doc/)
- Squid Log Format (https://wiki.squid-cache.org/Features/LogFormat)

# Device Configuration

Squid access logs can be forwarded to LogZilla via syslog. Configure Squid to send logs to the local syslog daemon, then configure syslog to forward to LogZilla.

## Step 1: Configure Squid Logging

Edit `/etc/squid/squid.conf` and add:

```
# LogZilla-compatible log format
logformat logzilla %ts.%03tu %6tr %>a %Ss/%03>Hs %<st %rm %ru %un %Sh/%<a %mt

# Send access logs to syslog (local0 facility, info level)
access_log syslog:local0.info logzilla
```

Restart Squid to apply changes:

```
systemctl restart squid
```

### Step 2: Configure Syslog Forwarding

Option A: syslog-ng

Add to `/etc/syslog-ng/syslog-ng.conf`:

```
destination d_logzilla { tcp("LOGZILLA_IP" port(514)); };
filter f_squid { facility(local0); };
log { source(s_src); filter(f_squid); destination(d_logzilla); };
```

Replace `LOGZILLA_IP` with the LogZilla server IP address or DNS name.

Option B: rsyslog

Add to `/etc/rsyslog.d/50-logzilla.conf`:

```
local0.* @@LOGZILLA_IP:514
```

Replace `LOGZILLA_IP` with the LogZilla server IP address or DNS name.

### Verification

Generate web traffic through the proxy, then verify events appear in LogZilla with `Vendor` tag set to `Squid`.

## Incoming Log Format

### Syslog-style Squid Logs

```
<date> <sensor> squid[<pid>]: <num> <action> <client> <sid>/<code> <size> <method> <url>
```

### Native Squid Access Logs

```
<timestamp> <duration> <client_ip> <status>/<code> <size> <method> <url> - <hierarchy>/<server>
<type>
```

# Parsed Metadata Fields

| Tag Name | Example | Description |
|---|---|---|
| Event Class | network | Cross-vendor classification |
| SrcIP | 192.168.1.100 | Source/client IP address |
| DstIP | 93.184.216.34 | Destination server IP (access logs only) |
| Squid HTTP Code | 200 OK | HTTP response status code with description |
| Squid HTTP Method | GET | HTTP request method |

# Log Examples

## Syslog-style Squid Log

```
Jan 01 00:00:00 sensor1 squid[1234]: 1234 CONNECT host1
SID1/500 1234 DELETE http://example.com/resource
```

## Native Squid Access Log

```
1684782466.123 2 192.168.1.100 TCP_MISS/200 1234 GET
http://example.com/path - DIRECT/93.184.216.34 text/html
```

# MITRE ATT&CK Mappings

| Event | MITRE ID | Tactic |
|---|---|---|
| Access denied (401/403) | T1110 | Credential Access |
| CONNECT tunnel | T1071 | Command and Control |

# Triggers

| Trigger | Description |
| --- | --- |
| `Squid: MITRE ATT&CK Threat Detected` | Catch-all for MITRE-mapped events |
| `Squid: Access Denied (T1110)` | HTTP 403 blocked request |
| `Squid: Authentication Failure (T1110)` | HTTP 401 auth failure |
| `Squid: CONNECT Tunnel (T1071)` | Tunnel requests (potential C2) |
| `Squid: Server Error` | HTTP 5xx backend errors |