

DomainTools Threat Feeds API

Generated on April 27, 2026

Contents

1	Common capabilities	8
2	Getting started	8
2.1	Quick Start: Your First Request	8
2.1.1	1. Get your API key	8
2.1.2	2. Choose a feed	8
2.1.3	3. Make your first request	8
2.1.4	4. Understand the response	9
2.1.5	5. Set up continuous polling	9
2.1.6	6. Handle the response in your application	9
2.1.7	Next steps	9
2.2	Access methods	10
3	Predictive risk feeds	11
3.1	Domain Hotlist	11
3.2	Domain Risk	11
3.3	IP Hotlist	12
3.4	IP Risk	12
4	Discovery feeds	13
4.1	Newly Active Domains (NAD)	13
4.2	Newly Observed Domains (NOD)	13
4.3	Newly Observed Hostnames (NOH)	13
4.4	Domain Discovery	14
4.5	Parsed Domain RDAP	14
4.6	5-Minute Domain WHOIS	14
4.7	5-Minute IP WHOIS	15
4.8	Need help?	15
5	Threat Feeds data retrieval patterns	16
5.1	Before you begin	16
5.2	Which API to use	16
5.3	Retrieve a full day from the Download API	17
5.4	Retrieve a full day from the Feed API	17

5.5	Recover from ingestion lag	18
5.6	Related resources	18
6	Domain Hotlist	19
6.1	Overview	19
6.2	Requirements	19
6.3	Authentication	19
6.3.1	API key (header) authentication	19
6.3.2	HMAC authentication	20
6.3.3	Open key authentication	21
6.4	Real-time Feed API	21
6.4.1	Base URL	21
6.4.2	Rate limits	21
6.4.3	Response formats	22
6.4.4	Session management	22
6.4.5	Quick start	22
6.4.6	Feed API parameters	23
6.4.7	Feed API response structure	25
6.4.8	Feed API response codes	26
6.4.9	Feed API examples	27
6.5	Real-time Download API	28
6.5.1	Base URL	29
6.5.2	Download API parameters	29
6.5.3	Download API response structure	29
6.5.4	Download API response codes	30
6.5.5	Download API file contents	30
6.5.6	Download API examples	30
6.6	Daily Download API	31
6.6.1	Overview	31
6.6.2	Base URL	31
6.6.3	Daily Download parameters	31
6.6.4	Daily Download response structure	32
6.6.5	Daily Download response codes	33
6.6.6	Daily Download file naming	33
6.6.7	File contents	33
6.6.8	Daily Download examples	34
6.7	RPZ Access	34
6.7.1	Overview	34
6.7.2	Available zones	35
6.7.3	Configuration	35
6.7.4	Testing	36
6.7.5	RPZ examples	36
6.8	Related resources	36
7	Domain Risk	37
7.1	Overview	37
7.2	Requirements	37
7.3	Authentication	37
7.3.1	API key (header) authentication	37
7.3.2	HMAC authentication	38
7.3.3	Open key authentication	39
7.4	Real-time Feed API	39
7.4.1	Base URL	39
7.4.2	Rate limits	40

7.4.3	Response formats	40
7.4.4	Session management	40
7.4.5	Quick start	41
7.4.6	Feed API parameters	41
7.4.7	Feed API response structure	43
7.4.8	Feed API response codes	45
7.4.9	Feed API examples	45
7.5	Real-time Download API	47
7.5.1	Base URL	47
7.5.2	Download API parameters	47
7.5.3	Download API response structure	47
7.5.4	Download API response codes	48
7.5.5	Download API file contents	48
7.5.6	Download API examples	48
7.6	Daily Download API	49
7.6.1	Overview	49
7.6.2	Base URL	49
7.6.3	Daily Download parameters	49
7.6.4	Daily Download response structure	50
7.6.5	Daily Download response codes	51
7.6.6	Daily Download file naming	51
7.6.7	File contents	51
7.6.8	Daily Download examples	51
7.7	Related resources	52
8	IP Hotlist	53
8.1	Overview	53
8.2	Requirements	53
8.3	Authentication	53
8.3.1	API key (header) authentication	54
8.3.2	HMAC authentication	54
8.3.3	Open key authentication	55
8.4	Daily Download API	55
8.4.1	Base URL	55
8.4.2	Parameters	55
8.4.3	Response structure	56
8.4.4	Response codes	56
8.4.5	File naming	56
8.4.6	File contents	57
8.4.7	Examples	59
8.5	Related resources	59
9	IP Risk	60
9.1	Overview	60
9.2	Requirements	60
9.3	Authentication	60
9.3.1	API key (header) authentication	60
9.3.2	HMAC authentication	61
9.3.3	Open key authentication	62
9.4	Daily Download API	62
9.4.1	Base URL	62
9.4.2	Parameters	62
9.4.3	Response structure	63
9.4.4	Response codes	63

9.4.5	File naming	63
9.4.6	File contents	63
9.4.7	Examples	65
9.5	Related resources	66
10	Newly Active Domains (NAD)	67
10.1	Overview	67
10.2	Requirements	67
10.3	Authentication	67
10.3.1	API key (header) authentication	67
10.3.2	HMAC authentication	68
10.3.3	Open key authentication	69
10.4	Real-time Feed API	69
10.4.1	Base URL	69
10.4.2	Rate limits	69
10.4.3	Response formats	70
10.4.4	Session management	70
10.4.5	Quick start	70
10.4.6	Feed API parameters	71
10.4.7	Feed API response structure	73
10.4.8	Feed API response codes	73
10.4.9	Feed API examples	74
10.5	Download API	75
10.5.1	Base URL	75
10.5.2	Download API parameters	75
10.5.3	Download API response structure	75
10.5.4	Download API response codes	76
10.5.5	Download API file contents	76
10.5.6	Download API examples	77
10.6	RPZ Access	77
10.6.1	Overview	77
10.6.2	Available zones	78
10.6.3	Configuration	78
10.6.4	Testing	79
10.6.5	RPZ examples	79
10.7	Related resources	79
11	Newly Observed Domains (NOD)	80
11.1	Overview	80
11.2	Requirements	80
11.3	Authentication	80
11.3.1	API key (header) authentication	80
11.3.2	HMAC authentication	81
11.3.3	Open key authentication	82
11.4	Real-time Feed API	82
11.4.1	Base URL	82
11.4.2	Rate limits	82
11.4.3	Response formats	83
11.4.4	Session management	83
11.4.5	Quick start	83
11.4.6	Feed API parameters	84
11.4.7	Feed API response structure	86
11.4.8	Feed API response codes	86
11.4.9	Feed API examples	87

11.5 Download API	88
11.5.1 Base URL	88
11.5.2 Download API parameters	88
11.5.3 Download API response structure	88
11.5.4 Download API response codes	89
11.5.5 Download API file contents	89
11.5.6 Download API examples	90
11.6 RPZ Access	90
11.6.1 Overview	90
11.6.2 Available zones	91
11.6.3 Configuration	91
11.6.4 Testing	92
11.6.5 RPZ examples	92
11.7 Related resources	93
12 Newly Observed Hostnames (NOH)	94
12.1 Overview	94
12.2 Requirements	94
12.3 Authentication	94
12.3.1 API key (header) authentication	94
12.3.2 HMAC authentication	95
12.3.3 Open key authentication	96
12.4 Real-time Feed API	96
12.4.1 Base URL	96
12.4.2 Rate limits	97
12.4.3 Response formats	97
12.4.4 Session management	97
12.4.5 Quick start	98
12.4.6 Feed API parameters	98
12.4.7 Feed API response structure	100
12.4.8 Feed API response codes	100
12.4.9 Feed API examples	101
12.5 Download API	102
12.5.1 Base URL	102
12.5.2 Download API parameters	102
12.5.3 Download API response structure	102
12.5.4 Download API response codes	103
12.5.5 Download API file contents	103
12.5.6 Download API examples	104
12.6 Related resources	104
13 Domain Discovery	105
13.1 Overview	105
13.2 Requirements	105
13.3 Authentication	105
13.3.1 API key (header) authentication	105
13.3.2 HMAC authentication	106
13.3.3 Open key authentication	107
13.4 Real-time Feed API	107
13.4.1 Base URL	107
13.4.2 Rate limits	108
13.4.3 Response formats	108
13.4.4 Session management	108
13.4.5 Quick start	109

13.4.6	Feed API parameters	109
13.4.7	Feed API response structure	111
13.4.8	Feed API response codes	111
13.4.9	Feed API examples	112
13.5	Real-time Download API	113
13.5.1	Base URL	113
13.5.2	Download API parameters	113
13.5.3	Download API response structure	113
13.5.4	Download API response codes	114
13.5.5	Download API file contents	115
13.5.6	Download API examples	115
13.6	Daily Download API	115
13.6.1	Overview	115
13.6.2	Base URL	116
13.6.3	Daily Download parameters	116
13.6.4	Daily Download response structure	116
13.6.5	Daily Download response codes	117
13.6.6	Daily Download file naming	117
13.6.7	File contents	117
13.6.8	Daily Download examples	117
13.7	Related resources	118
14	Parsed Domain RDAP	119
14.1	Overview	119
14.2	Requirements	119
14.3	Authentication	119
14.3.1	API key (header) authentication	119
14.3.2	HMAC authentication	120
14.3.3	Open key authentication	121
14.4	Real-time Feed API	121
14.4.1	Base URL	122
14.4.2	Rate limits	122
14.4.3	Response formats	122
14.4.4	Session management	122
14.4.5	Quick start	122
14.4.6	Feed API parameters	123
14.4.7	Feed API response structure	124
14.4.8	Feed API response codes	126
14.4.9	Feed API examples	126
14.5	Real-time Download API	127
14.5.1	Base URL	127
14.5.2	Download API parameters	127
14.5.3	Download API response structure	128
14.5.4	Download API response codes	128
14.5.5	Download API file contents	129
14.5.6	Download API examples	129
14.6	Related resources	130
15	5-Minute Domain WHOIS	131
15.1	Overview	131
15.2	Requirements	131
15.3	Authentication	131
15.3.1	API key (header) authentication	132
15.3.2	HMAC authentication	132

15.3.3 Open key authentication	133
15.4 Daily Download API	133
15.4.1 Base URLs	133
15.4.2 Daily Download parameters	134
15.4.3 Daily Download response structure	135
15.4.4 Daily Download response codes	135
15.4.5 Daily Download file naming	135
15.4.6 File contents	135
15.4.7 Daily Download examples	137
15.5 Related resources	137
16 5-Minute IP WHOIS	138
16.1 Overview	138
16.2 Requirements	138
16.3 Authentication	138
16.3.1 API key (header) authentication	139
16.3.2 HMAC authentication	139
16.3.3 Open key authentication	140
16.4 Daily Download API	140
16.4.1 Base URLs	140
16.4.2 Daily Download parameters	141
16.4.3 Daily Download response structure	142
16.4.4 Daily Download response codes	142
16.4.5 Daily Download file naming	142
16.4.6 File contents	142
16.4.7 Daily Download examples	143
16.5 Related resources	144
17 Threat Feeds via Response Policy Zone (RPZ)	145
17.1 Introduction	145
17.2 Available feeds	145
17.2.1 Newly Observed Domains (NOD)	145
17.2.2 Newly Active Domains (NAD)	145
17.2.3 Risk Score Based Domain Hotlists	145
17.3 Zone naming format	146
17.4 How RPZ blocks domains	146
17.5 Configure your RPZ connection	147
17.5.1 Provide your IP addresses to DomainTools	147
17.5.2 Configure your firewall	147
17.5.3 Authenticate with TSIG	147
17.6 Test your RPZ configuration	147
17.7 Advanced configuration	148
17.7.1 Maintain a local allowlist	148
17.7.2 Log NAD matches instead of blocking	148
17.7.3 Redirect to a walled garden	149

Proactive cybersecurity through curated lists of malicious and suspicious infrastructure indicators.

Threat Feeds deliver real-time intelligence directly to your security tools, enabling you to block threats before they cause harm. Instead of waiting to react to an attack, you can integrate this data into systems like firewalls, proxies, and SIEMs to automatically defend against threats that leverage new or high-risk domains.

1 Common capabilities

- **Real-time data delivery:** Access the latest threat intelligence within minutes of observation, crucial for stopping attacks leveraging ephemeral (short-lived) domains.
- **Configurable polling frequency:** Tailor data retrieval to your operational needs, with updates available as frequently as every 1 minute.
- **Comprehensive historical data:** Retrieve up to 90 days of historical feed data via the Download API to ensure no data is missed.
- **Diverse threat intelligence feeds:** Access a variety of specialized feeds, including those for newly observed domains, high-risk domains, and domain registration changes, to match specific threat intelligence needs.
- **Reliable data continuity:** Ensure seamless data ingestion with mechanisms to prevent data loss or duplication, allowing for continuous and uninterrupted threat monitoring.
- **Granular filtering:** Efficiently narrow down threat data using domain pattern filtering, reducing the need for extensive downstream processing.

2 Getting started

Tip: Quick start with Python

Our [Python SDK](#) provides native support for this API. Install with `pip install domaintools_api --upgrade`

2.1 Quick Start: Your First Request

Get started with threat feeds in under 5 minutes by following these steps.

2.1.1 1. Get your API key

Your API key is available in your DomainTools account dashboard. For this tutorial, we'll use header authentication (recommended).

2.1.2 2. Choose a feed

Start with the **Newly Observed Domains (NOD)** feed - it's ideal for general threat detection and has moderate volume.

2.1.3 3. Make your first request

```
curl -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/nod/?sessionID=myFirstSession'
```

This prints the last hour of newly observed domains to your terminal. To save the output to a file instead, add `-o` :

```
curl -o nod_output.json -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/nod/?sessionID=myFirstSession'
```

2.1.4 4. Understand the response

You'll receive JSON data (NDJSON format) with entries like:

```
{"timestamp":"2025-01-06T15:30:42Z","domain":"example-new-domain.com"}
{"timestamp":"2025-01-06T15:30:45Z","domain":"another-domain.net"}
```

Each line contains:

- `timestamp`: When the domain or IP was first observed
- `domain`: The apex-level domain name

IP-based feeds (IP Risk, IP Hotlist) return `ip` instead of `domain`, along with additional enrichment fields. See the individual feed documentation for details.

2.1.5 5. Set up continuous polling

Call the same endpoint again with the same `sessionID`:

```
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nod/?sessionID=myFirstSession'
```

This returns only domains observed since your last request — no duplicates. As before, the output prints to your terminal unless you redirect it to a file with `-o`.

2.1.6 6. Handle the response in your application

Python example:

```
import requests
import json
import time

api_key = "YOUR_API_KEY"
url = "https://api.domaintools.com/v1/feed/nod/"
headers = {"X-API-Key": api_key}
params = {"sessionID": "myFirstSession"}

while True:
    response = requests.get(url, headers=headers, params=params)

    for line in response.text.strip().split('\n'):
        if line:
            domain_data = json.loads(line)
            print(f"New domain: {domain_data['domain']}")

    time.sleep(60) # Poll every minute
```

2.1.7 Next steps

- **Filter domains:** Add `?domain=*.io` to monitor specific TLDs
- **Try other feeds:** Explore NAD, NOH, Domain Hotlist, and Domain Risk
- **Download historical data:** Use the Download API for backfill
- **Integrate with DNS firewalls:** Configure RPZ for NOD and NAD

[View detailed NOD documentation](#)

2.2 Access methods

Each feed supports different access methods:

- **Real-time Feed API:** Stream current data with configurable polling (as often as every 60 seconds)
- **Real-time Download API:** Historical archives organized by hour, available for 90 days
- **Daily Download:** Daily batch files for archival and bulk processing
- **RPZ:** Response Policy Zone format for direct DNS firewall integration (select feeds)

All feeds support flexible authentication methods and provide comprehensive filtering options to match your specific security requirements.

Feed	Real-time Feed	Real-time Download	Daily Download	RPZ
Predictive Risk Feeds				
Domain Hotlist	Yes	Yes	Yes	Yes
Domain Risk	Yes	Yes	Yes	
IP Hotlist	(coming soon)	(coming soon)	Yes	
IP Risk	(coming soon)	(coming soon)	Yes	
Discovery Feeds				
Newly Active Domains (NAD)	Yes	Yes		Yes
Newly Observed Domains (NOD)	Yes	Yes		Yes
Newly Observed Hostnames (NOH)	Yes	Yes		

Feed	Real-time Feed	Real-time Download	Daily Download	RPZ
Domain Discovery	Yes	Yes	Yes	
Parsed Domain RDAP	Yes	Yes		
5-Minute Domain WHOIS			Yes	
5-Minute IP WHOIS			Yes	

3 Predictive risk feeds

Identify new, potentially risky infrastructure before it's used in attacks.

3.1 Domain Hotlist

The Domain Hotlist identifies malicious and currently operational apex-level domains with high Domain Risk Scores that have shown activity within the last 24 hours. Each entry includes a 24-hour expiration time, making it a focused alternative to the broader Domain Risk feed.

Use this feed when you need to:

- Build high-confidence block lists
- Identify currently active and highly risky domains for immediate action
- Enhance SOC and Threat Intel workflows with log and alert enrichment
- Create custom network or endpoint block rules
- Triage domain-based alerts
- Integrate as a blocklist into DNS resolvers via RPZ

Inclusion criteria: Apex-level domains with high Domain Risk Scores (≥ 70 Proximity OR ≥ 90 Phish OR ≥ 90 Malware OR ≥ 90 Spam) that have shown activity within the last 24 hours. Each entry expires after 24 hours.

[View Domain Hotlist documentation](#)

3.2 Domain Risk

The Domain Risk feed provides a continuous data stream of all newly-scored, high-risk domains, regardless of their recent activity. This offers a more comprehensive view of potentially dangerous infrastructure that may not be currently active but still poses a risk.

Use this feed when you need to:

- Gain comprehensive visibility into potentially dangerous infrastructure
- Enable proactive threat intelligence for early detection
- Set up automated detection rules in TIPs or SIEMs
- Trigger alerts when network devices communicate with high-risk domains
- Create automated playbooks for domain enrichment
- Prioritize threats more quickly

Inclusion criteria: All apex-level domains with a combined Domain Risk Score of 70 or higher, regardless of observed activity.

[View Domain Risk documentation](#)

3.3 IP Hotlist

The IP Hotlist identifies high-risk IP addresses hosting hostile domains that are observed to be active within a 24-hour time window. Each entry includes comprehensive risk scores and enrichment data to help prioritize threats based on hosting infrastructure.

Use this feed when you need to:

- Block high-risk hosting infrastructure
- Monitor active threat actor IP addresses
- Correlate domain threats with hosting patterns
- Build IP-based threat intelligence
- Detect malicious hosting providers
- Enrich security alerts with IP context

Inclusion criteria: More than 50% of domains on the IP have proximity score of 70+ OR Threat Profile score of 90+; pDNS activity on malicious domains within 24 hours.

[View IP Hotlist documentation](#)

3.4 IP Risk

The IP Risk feed provides comprehensive risk intelligence for all IP addresses known to be hosting domains. This feed includes extensive enrichment data such as threat scores, geographic information, ASN details, and domain hosting metrics.

Use this feed when you need to:

- Monitor IP addresses hosting domains for threat intelligence
- Analyze hosting infrastructure risk patterns
- Correlate IP-based threats with domain activity
- Build IP reputation databases
- Detect suspicious hosting patterns
- Enrich security alerts with IP risk context
- Track threat actor infrastructure

Inclusion criteria: IP is actively hosting one or more domains (regardless of risk level).

[View IP Risk documentation](#)

4 Discovery feeds

Track new domains and hostnames as they appear.

4.1 Newly Active Domains (NAD)

The NAD feed lists domains that have become active in our global passive DNS sensor network, either for the first time or after a period of inactivity (at least 10 days). This helps identify when a previously dormant domain is being repurposed, a common tactic for attackers.

Use this feed when you need to:

- Detect the reactivation of previously dormant domains
- Identify potentially suspicious infrastructure
- Monitor domains that might be used for malicious activities
- Integrate as a blocklist into DNS resolvers via RPZ

Inclusion criteria: Domains observed in passive DNS to be newly active in the latest lifecycle of the domain, either for the first time or after an inactive period of at least 10 days.

[View Newly Active Domains documentation](#)

4.2 Newly Observed Domains (NOD)

The NOD feed provides a real-time list of domains that have been observed for the first time in our global passive DNS sensor network. This feed is ideal for brand protection, corporate intelligence, and temporarily blocking outbound connections to brand-new domains until their reputation can be assessed.

Use this feed when you need to:

- Identify newly registered domains for typosquatting (registering domains similar to popular brands) or brand abuse
- Monitor the emergence of new domains relevant to your organization
- Temporarily block access to newly observed domains to mitigate risks
- Integrate as a blocklist into DNS resolvers via RPZ

Inclusion criteria: Domains observed in passive DNS for the first time.

[View Newly Observed Domains documentation](#)

4.3 Newly Observed Hostnames (NOH)

The NOH feed offers a more granular view by listing fully qualified domain names (FQDNs) the first time they are observed on our global passive DNS sensor network. This helps you detect threats like phishing and domain shadowing (creating malicious subdomains on compromised legitimate domains) that often use unique subdomains on legitimate-looking domains to evade detection.

Use this feed when you need to:

- Detect phishing attempts using unique subdomains
- Identify domain shadowing tactics
- Enhance real-time threat detection with high-volume hostname data

- Monitor subdomain creation on legitimate domains

Inclusion criteria: Fully qualified domain names (FQDNs) observed in passive DNS for the first time.

[View Newly Observed Hostnames documentation](#)

4.4 Domain Discovery

This feed contains the largest dataset of its kind, providing a daily list of all newly registered and newly observed domains from all TLDs, including those that don't publish zone files.

Use this feed when you need to:

- Perform comprehensive domain monitoring for security and intelligence
- Track the global landscape of new domain registrations
- Identify emerging threats and trends related to new domain creation
- Analyze patterns in domain registration activity

Inclusion criteria: All newly registered and newly observed domains from all TLDs, including TLDs that do not publish zone files.

[View Domain Discovery documentation](#)

4.5 Parsed Domain RDAP

This feed provides parsed and normalized domain information extracted from raw RDAP records, including contact information, registrar details, name servers, and important dates. Use this feed for efficient data searching, indexing, and automated processing in security workflows.

Use this feed when you need to:

- Search for, index, or cross-reference data from RDAP records
- Enable programmatic access to structured RDAP data
- Analyze domain registration data to identify patterns
- Track threat actors through registration information
- Monitor changes in registration data for brand protection

Inclusion criteria: Changes to global domain registration information, populated by the Registration Data Access Protocol (RDAP).

[View Parsed Domain RDAP documentation](#)

4.6 5-Minute Domain WHOIS

The 5-Minute Domain WHOIS feed provides the most recently updated domain WHOIS records, processed on a 5-minute basis.

Use this feed when you need to:

- Monitor domain registration changes in near real-time
- Track WHOIS record updates for threat intelligence
- Analyze domain ownership and contact information changes
- Build domain registration databases

- Detect suspicious registration patterns

Inclusion criteria: All domain names processed since the previous 5-minute update cycle.

[View 5-Minute Domain WHOIS documentation](#)

4.7 5-Minute IP WHOIS

The 5-Minute IP WHOIS feed provides the most recently updated IPv4 WHOIS records, processed on a 5-minute basis.

Use this feed when you need to:

- Monitor IP address allocation and ownership changes
- Track IP WHOIS record updates for threat intelligence
- Analyze network infrastructure changes
- Build IP intelligence databases
- Correlate IP ownership with threat activity

Inclusion criteria: All IPv4 addresses processed since the previous 5-minute update cycle.

[View 5-Minute IP WHOIS documentation](#)

4.8 Need help?

- **API Reference:** Complete endpoint documentation with interactive reference
- **Authentication issues:** Contact enterprisesupport@domaintools.com
- **Questions about which feed to use:** Review the feed descriptions above or contact your DomainTools counterpart

5 Threat Feeds data retrieval patterns

This page shows you how to:

- Choose between the Download API and Feed API for bulk data retrieval
- Retrieve a full day of feed data using either API
- Recover from ingestion gaps without losing your place in the stream

These patterns apply to any Threat Feed that supports the Real-time Feed API and Real-time Download API. Examples on this page use the Newly Observed Domains (NOD) feed.

5.1 Before you begin

- Obtain your API credentials from your DomainTools account dashboard.
- Confirm that your account has access to the feed you want to retrieve.
- Review [session management](#) to understand how `sessionID` (a persistent cursor that tracks your position in the feed), HTTP 206, and HTTP 200 responses work.

5.2 Which API to use

Scenario	Recommended API	Why
Backfill a full day or more	Download API	Files are pre-built hourly snapshots. No session state to manage, and you avoid loading the real-time stream.
Retrieve historical data with filters	Feed API	Supports query parameters like <code>domain</code> , <code>risk score thresholds</code> , and <code>time windows</code> that the Download API doesn't offer.
Catch up after a short outage	Feed API	Your existing <code>sessionID</code> already points to where you left off.

5.3 Retrieve a full day from the Download API

The Download API provides hourly snapshot files, already serialized and organized by date. This is the simplest way to retrieve a full day of data.

1. Request the file list for a specific day. Each hour produces a data file and a checksum file, so set `limit=48` to cover 24 hours:

```
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/nod/?limit=48' > files.json
```

2. Download the hourly files. Each `.json.gz` file contains one hour of feed data in the same NDJSON format as the Feed API:

```
for url in $(jq -r '.response.files[].url' files.json | grep '\.json\.gz$'); do
  curl -O "$url"
done
```

3. Verify file integrity using the `.sha256` checksum files:

```
for url in $(jq -r '.response.files[].url' files.json | grep '\.sha256$'); do
  curl -O "$url"
done
sha256sum -c *.sha256
```

The Download API retains 90 days of hourly files. If a request fails with HTTP 403, verify your API credentials and feed access. For details on response structure and file naming, see the Download API section of your feed's documentation (for example, [Domain Hotlist Download API](#)).

5.4 Retrieve a full day from the Feed API

Use this approach when you need the Feed API's filtering capabilities or when you don't have access to the Download API for your feed.

1. Start a new session with `after=-86400` (24 hours ago) and `fromBeginning=true`. This returns the first hour of data within that window:

```
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nod/?sessionID=backfill-2025-01-
  ↵ 06&after=-86400&fromBeginning=true'
```

The API responds with HTTP 206, indicating more data is available. The `sessionID` now points to the end of that first batch.

2. Drop the `after` and `fromBeginning` parameters. Continue calling the Feed API with only the `sessionID`:
!!! warning Don't pass `fromBeginning=true` on subsequent calls. The API returns an error if `fromBeginning` is used with an existing session ID.

```
curl -H 'X-API-Key: YOUR_API_KEY' \  
'https://api.domaintools.com/v1/feed/nod/?sessionId=backfill-2025-01-06'
```

3. Repeat step 2 while the API returns HTTP 206. When you receive HTTP 200, you've retrieved all available data.

Note

Each request can return up to 10 million results. For high-volume feeds, a full day may require many iterations. Respect the [rate limits](#) of 2 queries per minute and 120 queries per hour.

5.5 Recover from ingestion lag

If your ingestion job goes down or falls behind, you don't need to start over. The Feed API retains your session position for up to 5 days. Your `sessionId` continues to point to the last record delivered to you.

To resume:

1. Restart your consumer with the same `sessionId` it was using before the outage.
2. Call the Feed API as you normally would — no extra parameters needed:

```
curl -H 'X-API-Key: YOUR_API_KEY' \  
'https://api.domaintools.com/v1/feed/nod/?sessionId=mySOC'
```

3. The API returns all data accumulated since your last successful request. Process the 206 responses as usual until you receive a 200.

If your outage exceeds 5 days, the session position expires. Create a new `sessionId` and use the [full-day retrieval](#) workflow or the [Download API](#) to backfill the gap.

5.6 Related resources

- [Threat Feeds overview](#) — feed descriptions and access method matrix
- [Session management](#) — how `sessionId`, HTTP 206, and session deletion work
- Individual feed documentation for API-specific parameters and response fields

6 Domain Hotlist

The Domain Hotlist identifies malicious and currently operational apex-level domains with high Domain Risk Scores that have shown activity within the last 24 hours. Each entry includes a 24-hour expiration time, making it a focused alternative to the broader Domain Risk feed for building high-confidence block lists.

6.1 Overview

Domains are apex-level (for example, `example.com` but not `www.example.com`), and the feed includes only those with high risk scores that have been active in the last 24 hours.

Use this feed when you need to:

- Build high-confidence block lists
- Identify currently active and highly risky domains for immediate action
- Enhance SOC and Threat Intel workflows with log and alert enrichment
- Create custom network or endpoint block rules
- Triage domain-based alerts
- Integrate as a blocklist into DNS resolvers via RPZ

Inclusion criteria: Apex-level domains with high Domain Risk Scores that have shown activity within the last 24 hours. Each entry expires after 24 hours.

6.2 Requirements

You need the following to access Threat Feeds:

- An Enterprise Account with DomainTools, accessible at <https://account.domaintools.com/my-account/>
- Authentication credentials (API key for header authentication, or API username and key for HMAC or open key authentication)
- A way to interact with a REST API delivered through AWS

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

For assistance, contact enterprisesupport@domaintools.com.

6.3 Authentication

You can authenticate to the Domain Hotlist APIs using three different methods. Choose the method that best fits your security requirements and technical environment.

6.3.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

Required header:

X-Api-Key: YOUR_API_KEY

Examples:

```
# Feed API request
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainhotlist/?sessionID=myBlocklist'
```

```
# Download API request
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domainhotlist/'
```

6.3.2 HMAC authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret, providing integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials shouldn't be stored in plain text or included directly in request URLs.

DomainTools supports MD5, SHA1, and SHA256 for the hashing algorithm.

Required query parameters:

- `api_username`: Your DomainTools API username
- `signature`: HMAC-SHA1 signature of `api_username` + `timestamp` + `uri_path`
- `timestamp`: Current UTC timestamp in ISO 8601 format (for example, `2025-06-01T15:30:00Z`)

Constructing the HMAC signature:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Important: URI path must include API version

The `uri_path` parameter must include the API version prefix. For example, use `/v1/feed/nod/` not `/feed/nod/`.

Example Python signing function:

```
import hmac
import hashlib

def sign(api_username, api_key, timestamp, uri):
    params = f"{api_username}{timestamp}{uri}"
    return hmac.new(api_key.encode("utf-8"), params.encode("utf-8"),
        ↪ hashlib.sha1).hexdigest()
```

Examples:

```
# Feed API request with HMAC
```

```
curl 'https://api.domaintools.com/v1/feed/domainhotlist?api_username=YOUR_-\n↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-\n↳ 06T15:30:00Z&sessionID=myBlocklist'
```

```
# Download API request with HMAC\ncurl 'https://api.domaintools.com/v1/download/domainhotlist?api_username=YOUR_-\n↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

6.3.3 Open key authentication

This is the easiest authentication scheme to implement, but also the least secure. Each request contains the full API key and API username as query parameters. We recommend using [API key header authentication](#) or [HMAC authentication](#) instead.

If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

Required query parameters:

- `api_username`: Your API username
- `api_key`: Your API key

Examples:

```
# Feed API request\ncurl 'https://api.domaintools.com/v1/feed/domainhotlist?api_username=YOUR_-\n↳ USERNAME&api_key=YOUR_API_KEY&sessionID=myBlocklist'
```

```
# Download API request\ncurl 'https://api.domaintools.com/v1/download/domainhotlist?api_username=YOUR_-\n↳ USERNAME&api_key=YOUR_API_KEY'
```

6.4 Real-time Feed API

The Feed API provides real-time access to current Domain Hotlist data. Use this API to poll for the latest feed updates at regular intervals, maintain a session to track your position in the feed, and filter results based on your specific needs.

6.4.1 Base URL

```
https://api.domaintools.com/v1/feed/domainhotlist/
```

6.4.2 Rate limits

Real-time feeds have the following rate limits:

- 2 queries per minute

- 120 queries per hour

If you exceed these limits, the API returns an error.

6.4.3 Response formats

The API supports two response formats:

NDJSON (Newline-Delimited JSON)

- Default format when no Accept header is specified
- Also known as JSON Lines (JSONL)
- One JSON object per line
- Efficient for streaming and processing large datasets
- Set Accept: `application/x-ndjson` to explicitly request this format

CSV (Comma-Separated Values)

- Set Accept: `text/csv` to request CSV format
- Add `&headers=1` to the query parameters to include column headers as the first line
- Not available for all feeds; check the specific feed documentation for CSV support

6.4.4 Session management

Session management allows you to maintain your position in the feed data stream, ensuring you don't miss or duplicate events when polling the API.

How sessions work:

- **Start a new session:** Provide a unique `sessionId` parameter of your choosing. By default, the API returns the past hour of results.
- **Resume a session:** Use the same `sessionId` in subsequent requests. The API returns all data since your last request.
- **Handle large result sets:** If a single request exceeds 10M results, the API returns an HTTP 206 response code. Repeat the same request with the same `sessionId` to receive the next batch of data until you receive an HTTP 200 response code.
- **One request at a time:** Do not send simultaneous requests with the same `sessionId` for the same feed. Wait for each request to complete before sending the next one. Concurrent requests with the same `sessionId` can produce errors or incomplete results.
- **Delete a session:** Use an HTTP DELETE request with your `sessionId` to clear the saved offset and start fresh.

Session ID requirements:

- 1 to 64 characters in length
- Alphanumeric characters and hyphens only (`[a-zA-Z0-9-]+`)
- Case-sensitive

6.4.5 Quick start

The standard access pattern is to periodically request the most recent feed data, as often as every 60 seconds.

```
curl -H 'X-API-Key: YOUR_API_KEY' \
```

```
'https://api.domaintools.com/v1/feed/domainhotlist?sessionID=myBlocklist'
```

This starts a new session and returns the last hour of data. Subsequent calls with the same `sessionID` return data since the last request.

6.4.6 Feed API parameters

6.4.6.1 sessionID

Type: String

Valid values: 1-64 alphanumeric characters and hyphens ([a-zA-Z0-9-]+)

Description: A unique identifier for the session, used for resuming data retrieval from the last point. Use a new `sessionID` to begin a new session, fetching the most recent hour by default. Reuse the same `sessionID` to return all feed data since your last request. If omitted, time window parameters (such as `after/before`) are required.

Example: `sessionID=mySOC`

Required: Yes, to continue where you left off (or use `after/before` instead)

6.4.6.2 after

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The start of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: `after=-60` or `after=2024-10-16T10:20:00Z`

Required: Yes, if `before` or `sessionID` not provided

6.4.6.3 before

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The end of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: `before=-120` or `before=2024-10-16T10:20:00Z`

Required: Yes, if `after` or `sessionID` not provided

6.4.6.4 domain

Type: String

Valid values: Domain character set restricted by the DNS specification (letters, digits, hyphens). International characters should be specified in punycode. A trailing dot is acceptable.

Description: Filter for an exact domain or a domain substring by prefixing or suffixing your string with *. Multiple parameters are supported (for example, `?domain=*apple*&domain=*microsoft*`). The URL-encoded version of * (%2A) may be required in some clients.

Example: `domain=*bank*` or `domain=example.com`

Required: No

6.4.6.5 fromBeginning

Type: Boolean

Valid values: true

Description: Requires a `sessionId`. When used with a new session ID, returns the first hour of data in the time window (rather than the last). Returns an error if the session ID already exists — drop `fromBeginning` from subsequent requests after the first call. Only the value true is accepted; any other value (including false) is ignored.

Example: `fromBeginning=true`

Required: No

6.4.6.6 top

Type: Integer

Valid values: Positive integer, 1-1,000,000,000

Description: Limits the number of results in the response payload. Primarily intended for testing. When you apply this parameter to risk feeds, results are sorted by `overall_risk` (descending).

Example: `top=10`

Required: No

6.4.6.7 headers

Type: Integer

Valid values: 1

Description: Adds a header row as the first line of the response when `text/csv` is requested. Set `headers=1` to enable. Only applies when requesting CSV format. Only the value 1 is accepted; any other value is invalid.

Example: `headers=1`

Required: No

6.4.6.8 Domain Hotlist filter parameters

Type: integer (optional)

Range: 1-99

Filter results to include only domains with risk scores greater than or equal to the specified threshold.

You can apply multiple risk score filters simultaneously. When multiple filters are used, domains must meet ALL specified thresholds to be included in the results.

Available risk score filters:

- `overall_min` - Minimum overall risk score
- `malware_min` - Minimum malware risk score
- `phishing_min` - Minimum phishing risk score
- `spam_min` - Minimum spam risk score
- `proximity_min` - Minimum proximity risk score

6.4.7 Feed API response structure

The API returns NDJSON format with one domain per line.

About Domain Risk Scores:

Domain Risk Scores predict the likelihood that a domain was registered with malicious intent. Each score ranges from 0-100, with special meanings for 0 (zero-listed/known legitimate) and 100 (blocklisted/known-bad). Regular scores range from 1-99, with null values indicating the domain has aged out of that threat profile.

Score Range	Score Color	Description
100	Red	Blocklisted. These domains can be considered known-bad, and have the highest likelihood of malicious intent. This includes sinkholed domains. DomainTools combines third party blocklists with our own scoring to determine which domains to blocklist.
90-99	Red	Strong confidence in near-term weaponization.
70-89	Orange	A potential threshold for suggesting malicious intent, and our default recommendation for significance in an investigation. Individual mileage may vary, depending on your security context and priorities.
50-69	Yellow	May require attention, depending on your security posture.

Score Range	Score Color	Description
1-49	Grey	Very little evidence of malicious intent.
0	Grey	Zero-listed. DomainTools zero-lists a domain when we have no evidence that it was registered with malicious intent. Zero-listing guards well-known legitimate domains against accidental blocking and includes domains which are vital to the expected operation of the Internet.

See the [Domain Risk Score user guide](#) for complete details.

Response fields:

timestamp (string): ISO 8601 UTC timestamp when the domain was scored or observed

domain (string): The apex-level domain name

phishing_risk (integer, nullable): Phishing risk score (0-100), or null if not applicable

malware_risk (integer, nullable): Malware risk score (0-100), or null if not applicable

spam_risk (integer, nullable): Spam risk score (0-100), or null if not applicable

proximity_risk (integer): Proximity risk score (0-100)

overall_risk (integer): Overall combined risk score (0-100)

expires (string): ISO 8601 UTC timestamp when this entry expires (24 hours from initial observation)

Example response:

```
{
  "timestamp": "2025-01-06T15:30:42Z",
  "domain": "malicious-example.com",
  "phishing_risk": 75,
  "malware_risk": 85,
  "spam_risk": 88,
  "proximity_risk": 99,
  "overall_risk": 99,
  "expires": "2025-01-07T15:30:42Z"
}
{
  "timestamp": "2025-01-06T15:30:45Z",
  "domain": "suspicious-domain.net",
  "phishing_risk": null,
  "malware_risk": null,
  "spam_risk": null,
  "proximity_risk": 95,
  "overall_risk": 95,
  "expires": "2025-01-07T15:30:45Z"
}
```

6.4.8 Feed API response codes

Code	Status	Description
200	OK	The request was successful and all data has been delivered

Code	Status	Description
206	Partial content	The request was successful, but only a portion of the data was returned. The request exceeded 10M results or the 1-hour evaluation window. Repeat the same request with the same <code>sessionID</code> to receive the next batch of data until you receive an HTTP 200 response
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
404	Not found	The requested resource (such as a <code>sessionID</code>) doesn't exist
406	Not acceptable	The specified <code>Accept</code> header value isn't supported. Only <code>application/x-ndjson</code> and <code>text/csv</code> are accepted
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

6.4.9 Feed API examples

Basic session polling:

```
# Start a new session
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainhotlist/?sessionID=myBlocklist'
```

```
# Resume the session (returns data since last request)
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainhotlist/?sessionID=myBlocklist'
```

Time window filtering:

```
# Get data from a specific time range
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainhotlist?after=2025-01-
↳ 06T10:00:00Z&before=2025-01-06T11:00:00Z'
```

Domain filtering:

```
# Filter for specific domain patterns
curl -H 'X-API-Key: YOUR_API_KEY' \

↳ 'https://api.domaintools.com/v1/feed/domainhotlist/?domain=*.example.com&sessionID=myBlocklist'
```

CSV format:

```
# Request CSV format with headers
curl -H 'Accept: text/csv' -H 'X-API-Key: YOUR_API_KEY' \
↳ 'https://api.domaintools.com/v1/feed/domainhotlist/?headers=1&sessionID=myBlocklist'
```

```
# Request CSV format without headers
curl -H 'Accept: text/csv' -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainhotlist/?sessionID=myBlocklist'
```

Risk score filtering:

```
# Filter for domains with overall risk >= 90
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainhotlist/?overall_-'
↳ min=90&sessionID=myBlocklist'
```

```
# Filter for domains with high phishing and malware risk
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainhotlist/?phishing_min=80&malware_-'
↳ min=80&sessionID=myBlocklist'
```

Handling large result sets:

```
# If you receive HTTP 206, repeat the request to get the next batch
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainhotlist/?sessionID=myBlocklist'
```

```
# Repeat until you receive HTTP 200
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainhotlist/?sessionID=myBlocklist'
```

Delete a session:

```
# Clear the saved offset and start fresh
curl -X DELETE -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainhotlist/?sessionID=myBlocklist'
```

6.5 Real-time Download API

The Real-time Download API provides access to historical Domain Hotlist data through temporary AWS S3 file links. Use this API to retrieve archived data you may have missed or to backfill your systems with historical information. Files are organized by hour and available for 90 days.

6.5.1 Base URL

```
https://api.domaintools.com/v1/download/domainhotlist/
```

6.5.2 Download API parameters

6.5.2.1 limit

Type: Integer

Valid values: Positive integer

Description: Limits the number of files returned in the response, starting from the most recent. Use to control payload size or test specific cases.

Example: `limit=10`

Required: No

6.5.3 Download API response structure

The API returns a JSON response containing an array of downloadable files. Each file entry includes:

download_name (string): The feed identifier (`domainhotlist`)

files (array): List of downloadable file entries

Each file object contains:

Field	Type	Description
<code>name</code>	string	Path and filename of the downloadable file
<code>last_modified</code>	string	Timestamp of last modification in ISO 8601 UTC format
<code>etag</code>	string	ETag (hash) used to verify file identity and versioning
<code>size</code>	integer	File size in bytes
<code>url</code>	string	Temporary signed URL to download the file from AWS

File naming convention:

- Data file:
`domainhotlist/{YYYY-MM-DD}/domainhotlist-{{YYYYMMDD}}.{{HH00}}-{{HH00}}.json.gz`
- Checksum file: `domainhotlist/{YYYY-MM-DD}/domainhotlist-{{YYYYMMDD}}.{{HH00}}-{{HH00}}.json.gz.sha256`

Example response:

```
{  
  "response": {
```

```
    "download_name": "domainhotlist",
    "files": [
      {
        "name":
↪ "domainhotlist/2024-11-19/domainhotlist-20241119.1900-2000.json.gz.sha256",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 64,
        "url": "https://download.example.com/domainhotlist/2024-11-
↪ 19/domainhotlist-20241119.1900-2000.json.gz.sha256?Expires=..."
      },
      {
        "name": "domainhotlist/2024-11-19/domainhotlist-20241119.1900-2000.json.gz",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 245000,
        "url": "https://download.example.com/domainhotlist/2024-11-
↪ 19/domainhotlist-20241119.1900-2000.json.gz?Expires=..."
      }
    ]
  }
}
```

6.5.4 Download API response codes

Code	Status	Description
200	OK	The request was successful
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

6.5.5 Download API file contents

The *.json.gz.sha256 file is a checksum containing a SHA-256 hash value used to verify the integrity of the downloaded file.

The *.json.gz file, when uncompressed, contains JSON data in the same format as the Feed API response (NDJSON with timestamp, domain, risk score fields, and expires).

6.5.6 Download API examples

List available files:

```
# Get the most recent files
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domainhotlist/?limit=10'
```

Download and verify a file:

```
# Get the file list
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domainhotlist/?limit=2' > files.json

# Extract the URL and download the data file
curl -o domainhotlist-data.json.gz "$(jq -r '.response.files[1].url' files.json)"

# Download the checksum file
curl -o domainhotlist-data.json.gz.sha256 "$(jq -r '.response.files[0].url'
↵ files.json)"

# Verify the integrity
sha256sum -c domainhotlist-data.json.gz.sha256
```

Batch processing:

```
# Download multiple files in a loop
for url in $(curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domainhotlist/?limit=24' | \
  jq -r '.response.files[] .url' | grep '\.json\.gz$'); do
  curl -O "$url"
done
```

6.6 Daily Download API

The Daily Download API provides daily batch summaries as an alternative to hourly real-time data. Use this when you need daily aggregated data rather than real-time updates.

6.6.1 Overview

Daily feed of high-risk domains that are observed to be active within a 24 hour time window.

The Daily Domain Hotlist is available in multiple variants, each filtered by different risk score thresholds. All variants require passive DNS activity within 24 hours.

Inclusion threshold: Varies by variant. See [File naming](#) section below.

Format: Tab-separated text file (gzipped); one domain per line with component risk scores

Size: ~900,000 domains, ~3.5MB compressed

Note: The Daily Domain Hotlist is also available in RPZ format. Contact enterprisesupport@domaintools.com for information about RPZ delivery.

6.6.2 Base URL

```
https://api.domaintools.com/v1/download/daily_domain_hotlist/
```

6.6.3 Daily Download parameters

The Daily Download API supports standard download parameters:

6.6.3.1 api_username

Type: string (required for HMAC and open key auth)

Your DomainTools API username

6.6.3.2 api_key

Type: string (required for open key auth)

Your DomainTools API key

6.6.3.3 signature

Type: string (required for HMAC auth)

HMAC signature of your request

6.6.3.4 timestamp

Type: string (required for HMAC auth)

Current timestamp for HMAC authentication in ISO 8601 format

6.6.3.5 limit

Type: integer (optional)

Limit the list of signed files. Ordering of files is always descending, so the latest files are first.

6.6.3.6 page

Type: integer (optional)

Select which page of results are returned. Pages begin at 0 with latest results.

6.6.3.7 prefix

Type: string (optional)

Filter results by date using the file prefix.

6.6.4 Daily Download response structure

The API returns a JSON response with signed URLs for downloadable files:

download_name (string): The feed identifier (`daily_domain_hotlist`)

files (array): List of downloadable file entries

Each file object contains:

Field	Type	Description
<code>name</code>	string	File path (e.g., <code>daily_domain_hotlist/99s.domainhotlist.gz</code>)
<code>last_modified</code>	string	Last modified date in ISO 8601 format
<code>etag</code>	string	Entity tag (hash of the file)
<code>size</code>	integer	Size in bytes

Field	Type	Description
url	string	Signed AWS download URL (valid for 12 hours)

6.6.5 Daily Download response codes

Code	Status	Description
200	OK	The request was successful
400	Bad request	Invalid request parameters
401	Unauthorized	Missing or invalid authentication
403	Forbidden	Insufficient permissions
404	No data to download	No files available

6.6.6 Daily Download file naming

The Daily Domain Hotlist is available in five variants, filtered by [Domain Risk Score](#) thresholds:

- **90s.domainhotlist.gz:** Domains with Proximity ≥ 70 OR (Malware Risk ≥ 90 AND Phishing Risk ≥ 90)
- **95s.domainhotlist.gz:** Domains with Proximity ≥ 85 OR (Malware Risk ≥ 95 AND Phishing Risk ≥ 95)
- **99s.domainhotlist.gz:** Domains with Proximity ≥ 85 OR (Malware Risk ≥ 99 AND Phishing Risk ≥ 99)
- **1k.domainhotlist.gz:** Top 1,000 highest-risk domains with Proximity ≥ 75 OR (Malware Risk ≥ 90 AND Phishing Risk ≥ 90)
- **100k.domainhotlist.gz:** Top 100,000 highest-risk domains with Proximity ≥ 75 OR (Malware Risk ≥ 90 AND Phishing Risk ≥ 90)

All variants require passive DNS activity within 24 hours.

6.6.7 File contents

The tab-separated file contains one domain per line with the following fields:

- **Domain Name:** The apex-level domain
- **Phishing:** Machine learning classifier prediction for phishing (0-100)
- **Malware:** Machine learning classifier prediction for malware (0-100)
- **Spam:** Machine learning classifier prediction for spam (0-100)
- **Proximity:** Indicates shared registration or infrastructure with known-bad domains (0-100)

Example line:

```
example.com 75 85 88 99
```

The risk score fields are used to filter domains into the different hotlist variants as described in the [File naming](#) section. For detailed information about these scores, see the [Domain Risk Score user guide](#).

6.6.8 Daily Download examples

List available files:

```
curl -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/daily_domain_hotlist/'
```

Download a specific file:

```
# Get the file list  
curl -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/daily_domain_hotlist/?limit=1' >  
↵ files.json  
  
# Download the file  
curl -o daily-domain-hotlist.gz "$(jq -r '.response.files[0].url' files.json)"  
  
# Decompress and view  
gunzip daily-domain-hotlist.gz  
head daily-domain-hotlist
```

6.7 RPZ Access

The Domain Hotlist feed is available via Response Policy Zone (RPZ) for direct integration with DNS firewalls.

6.7.1 Overview

Response Policy Zone (RPZ) provides DNS-level blocking by integrating threat feeds directly into your DNS resolver. This allows you to automatically block or redirect DNS queries for domains in the feed, providing real-time protection before connections are established.

How RPZ works:

When a user attempts to access a domain in the RPZ feed, your DNS resolver responds with an NXDOMAIN (“no such domain”) status code, effectively making the domain unavailable. This blocks malicious domains at the DNS layer, preventing endpoint communication before any connection is established.

Benefits:

- **Real-time protection:** Blocks threats at the DNS resolver level
- **Automatic updates:** Receives updates via DNS zone transfers (AXFR/IXFR)
- **No client configuration:** Works transparently for all clients using your DNS resolver
- **Efficient:** Minimal performance impact on DNS resolution
- **Standard protocol:** Uses DNS Response Policy Zones specification

Zone naming format:

RPZ zones are named using the pattern: [interval].[feed].rpz.domaintools.com

Available time intervals: 5m, 10m, 30m, 1h, 3h, 12h (and 24h for some feeds)

Larger time intervals are supersets that include smaller intervals. Smaller intervals have smaller zone sizes and may be available faster.

6.7.2 Available zones

The Domain Hotlist feed is available in the following RPZ zones:

- `1k.domainhotlist.rpz.domaintools.com` - Top 1,000 highest-risk domains with Proximity ≥ 75 OR (Malware Risk ≥ 90 AND Phishing Risk ≥ 90)
- `100k.domainhotlist.rpz.domaintools.com` - Top 100,000 highest-risk domains with Proximity ≥ 75 OR (Malware Risk ≥ 90 AND Phishing Risk ≥ 90)
- `90s.domainhotlist.rpz.domaintools.com` - Domains with Proximity ≥ 70 OR (Malware Risk ≥ 90 AND Phishing Risk ≥ 90)
- `95s.domainhotlist.rpz.domaintools.com` - Domains with Proximity ≥ 85 OR (Malware Risk ≥ 95 AND Phishing Risk ≥ 95)
- `99s.domainhotlist.rpz.domaintools.com` - Domains with Proximity ≥ 85 OR (Malware Risk ≥ 99 AND Phishing Risk ≥ 99)

All zones require passive DNS activity within the last 24 hours.

Choosing a zone:

- **Smaller numbers = higher confidence, lower volume:** The 1k and 99s zones contain the highest-risk domains with the fewest false positives
- **Larger numbers = broader coverage, higher volume:** The 100k and 90s zones provide more comprehensive coverage but may include more borderline cases
- Choose based on your risk tolerance and infrastructure capacity

6.7.3 Configuration

Configuration requirements:

To access RPZ feeds, you need to:

1. **Provide IP addresses:** Contact enterprisesupport@domaintools.com with:
 - IP address(es) from which you connect to the RPZ provider DNS server
 - IP address(es) to receive DNS NOTIFY messages (typically the same)
2. **Configure firewall:** Add rules to allow DomainTools hosts to send UDP packets to port 53:
 - IPv4: 104.244.13.88 Port: 53
 - IPv4: 104.244.14.88 Port: 53
3. **Set up TSIG authentication:** DomainTools uses TSIG (Secret Key Transaction Authentication) for authorization:
 - TSIG key algorithm: `hmac-sha512`
 - TSIG key and key name: Provided by DomainTools Enterprise Support

Delivery method:

RPZ feeds are delivered via:

- Incremental Zone Transfers (IXFR)
- Full zone transfers (AXFR)
- DNS NOTIFY messages to trigger zone updates

For detailed configuration instructions, including DNS resolver setup, advanced features (allowlists, walled gardens, logging), and troubleshooting, see the [Response Policy Zone documentation](#).

6.7.4 Testing

Testing your RPZ configuration:

When your DNS resolver blocks a domain using RPZ, the response includes an SOA (Start of Authority) record that identifies which RPZ feed was used.

Test domain:

Each RPZ feed includes a test domain entry: `test.rpz.domaintools.test`

Use this to verify the RPZ feed is loaded and working. A successful test returns:

- NXDOMAIN status code
- SOA record in the ADDITIONAL section showing the specific feed name

Example SOA record:

```
;; ADDITIONAL SECTION:  
3h.nad.rpz.domaintools.com. 86400 IN SOA rpz-ns1.domaintools.com.  
↪ noc.domaintools.com. 946684799 600 300 86400 86400
```

The SOA SERIAL number (Unix epoch timestamp) indicates when the feed was last regenerated.

Troubleshooting:

If the SOA record appears in the AUTHORITY section instead of ADDITIONAL, or doesn't show the specific feed name, the response didn't come from RPZ. Check your DNS resolver's RPZ-related logs for additional debugging information.

6.7.5 RPZ examples

Test the RPZ feed:

```
# Query the test domain  
dig test.rpz.domaintools.test  
  
# Expected response includes NXDOMAIN and SOA record showing the feed name  
# Example SOA in ADDITIONAL section:  
# 1k.domainhotlist.rpz.domaintools.com. 86400 IN SOA rpz-ns1.domaintools.com. ...
```

Verify a specific zone is loaded:

```
# Test with the top 1,000 highest-risk zone  
dig @your-dns-server test.rpz.domaintools.test  
  
# Check the SOA record shows: 1k.domainhotlist.rpz.domaintools.com
```

6.8 Related resources

- [Domain Risk feed](#)
- [Newly Active Domains feed](#)
- [Domain Risk Score user guide](#)
- [Response Policy Zone documentation](#)

7 Domain Risk

The Domain Risk feed provides a continuous data stream of all newly-scored, high-risk domains (combined score of 70+), regardless of their recent activity. This offers comprehensive visibility into potentially dangerous infrastructure that may not be currently active but still poses a risk.

7.1 Overview

Domains are apex-level (for example, `example.com` but not `www.example.com`), and the feed includes all domains that reach a combined Domain Risk Score of 70 or higher, whether or not they show recent activity.

Use this feed when you need to:

- Gain comprehensive visibility into potentially dangerous infrastructure
- Enable proactive threat intelligence for early detection
- Set up automated detection rules in TIPs or SIEMs
- Trigger alerts when network devices communicate with high-risk domains
- Create automated playbooks for domain enrichment
- Prioritize threats more quickly

Inclusion criteria: All apex-level domains with a combined Domain Risk Score of 70 or higher, regardless of observed activity.

Important note: While Domain Risk entries don't expire, we recommend using entries no older than 24 hours for optimal threat detection.

7.2 Requirements

You need the following to access Threat Feeds:

- An Enterprise Account with DomainTools, accessible at <https://account.domaintools.com/my-account/>
- Authentication credentials (API key for header authentication, or API username and key for HMAC or open key authentication)
- A way to interact with a REST API delivered through AWS

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

For assistance, contact enterprisesupport@domaintools.com.

7.3 Authentication

You can authenticate to the Domain Risk APIs using three different methods. Choose the method that best fits your security requirements and technical environment.

7.3.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

Required header:

X-Api-Key: YOUR_API_KEY

Examples:

```
# Feed API request
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrisk/?sessionID=myThreatIntel'
```

```
# Download API request
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domainrisk/'
```

7.3.2 HMAC authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret, providing integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials shouldn't be stored in plain text or included directly in request URLs.

DomainTools supports MD5, SHA1, and SHA256 for the hashing algorithm.

Required query parameters:

- `api_username`: Your DomainTools API username
- `signature`: HMAC-SHA1 signature of `api_username` + `timestamp` + `uri_path`
- `timestamp`: Current UTC timestamp in ISO 8601 format (for example, 2025-06-01T15:30:00Z)

Constructing the HMAC signature:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Important: URI path must include API version

The `uri_path` parameter must include the API version prefix. For example, use `/v1/feed/nod/` not `/feed/nod/`.

Example Python signing function:

```
import hmac
import hashlib

def sign(api_username, api_key, timestamp, uri):
    params = f"{api_username}{timestamp}{uri}"
    return hmac.new(api_key.encode("utf-8"), params.encode("utf-8"),
        ↪ hashlib.sha1).hexdigest()
```

Note: HMAC timestamp requirements

The timestamp parameter in HMAC authentication must be current (within a few minutes of the server time). The timestamps shown in these examples are static for demonstration purposes. In production, generate a fresh timestamp for each request using your system's current time in ISO 8601 UTC format (e.g., 2025-01-06T15:30:00Z).

Examples:

```
# Feed API request with HMAC
curl 'https://api.domaintools.com/v1/feed/domainrisk/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-
↳ 06T15:30:00Z&sessionID=myThreatIntel'
```

```
# Download API request with HMAC
curl 'https://api.domaintools.com/v1/download/domainrisk/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

7.3.3 Open key authentication

This is the easiest authentication scheme to implement, but also the least secure. Each request contains the full API key and API username as query parameters. We recommend using [API key header authentication](#) or [HMAC authentication](#) instead.

If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

Required query parameters:

- api_username: Your API username
- api_key: Your API key

Examples:

```
# Feed API request
curl 'https://api.domaintools.com/v1/feed/domainrisk/?api_username=YOUR_
↳ USERNAME&api_key=YOUR_API_KEY&sessionID=myThreatIntel'
```

```
# Download API request
curl 'https://api.domaintools.com/v1/download/domainrisk/?api_username=YOUR_
↳ USERNAME&api_key=YOUR_API_KEY'
```

7.4 Real-time Feed API

The Feed API provides real-time access to current Domain Risk data. Use this API to poll for the latest feed updates at regular intervals, maintain a session to track your position in the feed, and filter results based on your specific needs.

7.4.1 Base URL

```
https://api.domaintools.com/v1/feed/domainrisk/
```

7.4.2 Rate limits

Real-time feeds have the following rate limits:

- 2 queries per minute
- 120 queries per hour

If you exceed these limits, the API returns an error.

7.4.3 Response formats

The API supports two response formats:

NDJSON (Newline-Delimited JSON)

- Default format when no Accept header is specified
- Also known as JSON Lines (JSONL)
- One JSON object per line
- Efficient for streaming and processing large datasets
- Set Accept: `application/x-ndjson` to explicitly request this format

CSV (Comma-Separated Values)

- Set Accept: `text/csv` to request CSV format
- Add `&headers=1` to the query parameters to include column headers as the first line
- Not available for all feeds; check the specific feed documentation for CSV support

7.4.4 Session management

Session management allows you to maintain your position in the feed data stream, ensuring you don't miss or duplicate events when polling the API.

How sessions work:

- **Start a new session:** Provide a unique `sessionId` parameter of your choosing. By default, the API returns the past hour of results.
- **Resume a session:** Use the same `sessionId` in subsequent requests. The API returns all data since your last request.
- **Handle large result sets:** If a single request exceeds 10M results, the API returns an HTTP 206 response code. Repeat the same request with the same `sessionId` to receive the next batch of data until you receive an HTTP 200 response code.
- **One request at a time:** Do not send simultaneous requests with the same `sessionId` for the same feed. Wait for each request to complete before sending the next one. Concurrent requests with the same `sessionId` can produce errors or incomplete results.
- **Delete a session:** Use an HTTP DELETE request with your `sessionId` to clear the saved offset and start fresh.

Session ID requirements:

- 1 to 64 characters in length

- Alphanumeric characters and hyphens only ([a-zA-Z0-9-]+)
- Case-sensitive

7.4.5 Quick start

The standard access pattern is to periodically request the most recent feed data, as often as every 60 seconds.

```
curl -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domainrisk/?sessionID=myThreatIntel'
```

This starts a new session and returns the last hour of data. Subsequent calls with the same `sessionID` return data since the last request.

7.4.6 Feed API parameters

7.4.6.1 sessionID

Type: String

Valid values: 1-64 alphanumeric characters and hyphens ([a-zA-Z0-9-]+)

Description: A unique identifier for the session, used for resuming data retrieval from the last point. Use a new `sessionID` to begin a new session, fetching the most recent hour by default. Reuse the same `sessionID` to return all feed data since your last request. If omitted, time window parameters (such as `after/before`) are required.

Example: `sessionID=mySOC`

Required: Yes, to continue where you left off (or use `after/before` instead)

7.4.6.2 after

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The start of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: `after=-60` or `after=2024-10-16T10:20:00Z`

Required: Yes, if `before` or `sessionID` not provided

7.4.6.3 before

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The end of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: before=-120 or before=2024-10-16T10:20:00Z

Required: Yes, if after or sessionID not provided

7.4.6.4 domain

Type: String

Valid values: Domain character set restricted by the DNS specification (letters, digits, hyphens). International characters should be specified in punycode. A trailing dot is acceptable.

Description: Filter for an exact domain or a domain substring by prefixing or suffixing your string with *. Multiple parameters are supported (for example, ?domain=*apple*&domain=*microsoft*). The URL-encoded version of * (%2A) may be required in some clients.

Example: domain=*bank* or domain=example.com

Required: No

7.4.6.5 fromBeginning

Type: Boolean

Valid values: true

Description: Requires a sessionID. When used with a new session ID, returns the first hour of data in the time window (rather than the last). Returns an error if the session ID already exists — drop fromBeginning from subsequent requests after the first call. Only the value true is accepted; any other value (including false) is ignored.

Example: fromBeginning=true

Required: No

7.4.6.6 top

Type: Integer

Valid values: Positive integer, 1-1,000,000,000

Description: Limits the number of results in the response payload. Primarily intended for testing. When you apply this parameter to risk feeds, results are sorted by overall_risk (descending).

Example: top=10

Required: No

Note: When using the top parameter with Domain Risk, results are automatically sorted by overall_risk in descending order (highest risk first).

7.4.6.7 headers

Type: Integer

Valid values: 1

Description: Adds a header row as the first line of the response when `text/csv` is requested. Set `headers=1` to enable. Only applies when requesting CSV format. Only the value 1 is accepted; any other value is invalid.

Example: `headers=1`

Required: No

7.4.6.8 Domain Risk filter parameters

Type: integer (optional)

Range: 1-99

Filter results to include only domains with risk scores greater than or equal to the specified threshold.

You can apply multiple risk score filters simultaneously. When multiple filters are used, domains must meet ALL specified thresholds to be included in the results.

Available risk score filters:

- `overall_min` - Minimum overall risk score
- `malware_min` - Minimum malware risk score
- `phishing_min` - Minimum phishing risk score
- `spam_min` - Minimum spam risk score
- `proximity_min` - Minimum proximity risk score

7.4.7 Feed API response structure

The API returns NDJSON format with one domain per line.

About Domain Risk Scores:

Domain Risk Scores predict the likelihood that a domain was registered with malicious intent. Each score ranges from 0-100, with special meanings for 0 (zero-listed/known legitimate) and 100 (blocklisted/known-bad). Regular scores range from 1-99, with null values indicating the domain has aged out of that threat profile.

Score Range	Score Color	Description
100	Red	Blocklisted. These domains can be considered known-bad, and have the highest likelihood of malicious intent. This includes sinkholed domains. DomainTools combines third party blocklists with our own scoring to determine which domains to blocklist.
90-99	Red	Strong confidence in near-term weaponization.

Score Range	Score Color	Description
70-89	Orange	A potential threshold for suggesting malicious intent, and our default recommendation for significance in an investigation. Individual mileage may vary, depending on your security context and priorities.
50-69	Yellow	May require attention, depending on your security posture.
1-49	Grey	Very little evidence of malicious intent.
0	Grey	Zero-listed. DomainTools zero-lists a domain when we have no evidence that it was registered with malicious intent. Zero-listing guards well-known legitimate domains against accidental blocking and includes domains which are vital to the expected operation of the Internet.

See the [Domain Risk Score user guide](#) for complete details.

Response fields:

timestamp (string): ISO 8601 UTC timestamp when the domain was scored or observed

domain (string): The apex-level domain name

phishing_risk (integer, nullable): Phishing risk score (0-100), or null if not applicable

malware_risk (integer, nullable): Malware risk score (0-100), or null if not applicable

spam_risk (integer, nullable): Spam risk score (0-100), or null if not applicable

proximity_risk (integer): Proximity risk score (0-100)

overall_risk (integer): Overall combined risk score (0-100)

Example response:

```
{
  "timestamp": "2025-04-22T16:08:33Z",
  "domain": "omaintools.com",
  "phishing_risk": 94,
  "malware_risk": 88,
  "spam_risk": 93,
  "proximity_risk": 80,
  "overall_risk": 94
}
{
  "timestamp": "2025-04-22T16:08:29Z",
  "domain": "v-domaintools.com",
  "phishing_risk": 96,
  "malware_risk": 91,
  "spam_risk": 99,
  "proximity_risk": 85,
  "overall_risk": 99
}
```

7.4.8 Feed API response codes

Code	Status	Description
200	OK	The request was successful and all data has been delivered
206	Partial content	The request was successful, but only a portion of the data was returned. The request exceeded 10M results or the 1-hour evaluation window. Repeat the same request with the same <code>sessionID</code> to receive the next batch of data until you receive an HTTP 200 response
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
404	Not found	The requested resource (such as a <code>sessionID</code>) doesn't exist
406	Not acceptable	The specified <code>Accept</code> header value isn't supported. Only <code>application/x-ndjson</code> and <code>text/csv</code> are accepted
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

7.4.9 Feed API examples

Basic session polling:

```
# Start a new session
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrisk/?sessionID=myThreatIntel'
```

```
# Resume the session (returns data since last request)
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrisk/?sessionID=myThreatIntel'
```

Time window filtering:

```
# Get data from a specific time range
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrisk/?after=2025-01-
↳ 06T10:00:00Z&before=2025-01-06T11:00:00Z'
```

Domain filtering:

```
# Filter for specific domain patterns
curl -H 'X-API-Key: YOUR_API_KEY' \
```

```
↪ 'https://api.domaintools.com/v1/feed/domainrisk/?domain=*.example.com&sessionID=myThreatIntel'
```

CSV format:

```
# Request CSV format with headers  
curl -H 'Accept: text/csv' -H 'X-Api-Key: YOUR_API_KEY' \  
↪ 'https://api.domaintools.com/v1/feed/domainrisk/?headers=1&sessionID=myThreatIntel'
```

```
# Request CSV format without headers  
curl -H 'Accept: text/csv' -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domainrisk/?sessionID=myThreatIntel'
```

Risk score filtering:

```
# Filter for domains with overall risk >= 90  
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domainrisk/?overall_  
↪ min=90&sessionID=myThreatIntel'
```

```
# Filter for domains with high phishing and malware risk  
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domainrisk/?phishing_min=80&malware_  
↪ min=80&sessionID=myThreatIntel'
```

```
# Get top 10 highest-risk domains (sorted by overall_risk)  
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domainrisk/?top=10&sessionID=myThreatIntel'
```

Handling large result sets:

```
# If you receive HTTP 206, repeat the request to get the next batch  
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domainrisk/?sessionID=myThreatIntel'
```

```
# Repeat until you receive HTTP 200  
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domainrisk/?sessionID=myThreatIntel'
```

Delete a session:

```
# Clear the saved offset and start fresh  
curl -X DELETE -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domainrisk/?sessionID=myThreatIntel'
```

7.5 Real-time Download API

The Real-time Download API provides access to historical Domain Risk data through temporary AWS S3 file links. Use this API to retrieve archived data you may have missed or to backfill your systems with historical information. Files are organized by hour and available for 90 days.

7.5.1 Base URL

```
https://api.domaintools.com/v1/download/domainrisk/
```

7.5.2 Download API parameters

7.5.2.1 limit

Type: Integer

Valid values: Positive integer

Description: Limits the number of files returned in the response, starting from the most recent. Use to control payload size or test specific cases.

Example: `limit=10`

Required: No

7.5.3 Download API response structure

The API returns a JSON response containing an array of downloadable files. Each file entry includes:

download_name (string): The feed identifier (domainrisk)

files (array): List of downloadable file entries

Each file object contains:

Field	Type	Description
name	string	Path and filename of the downloadable file
last_modified	string	Timestamp of last modification in ISO 8601 UTC format
etag	string	ETag (hash) used to verify file identity and versioning
size	integer	File size in bytes
url	string	Temporary signed URL to download the file from AWS

File naming convention:

- Data file: `domainrisk/{YYYY-MM-DD}/domainrisk-{YYYYMMDD}.{HH00}-{HH00}.json.gz`
- Checksum file: `domainrisk/{YYYY-MM-DD}/domainrisk-{YYYYMMDD}.{HH00}-{HH00}.json.gz.sha256`

Example response:

```
{
  "response": {
    "download_name": "domainrisk",
    "files": [
      {
        "name":
        ↪ "domainrisk/2024-11-19/domainrisk-20241119.1900-2000.json.gz.sha256",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 64,
        "url": "https://download.example.com/domainrisk/2024-11-19/domainrisk-
        ↪ 20241119.1900-2000.json.gz.sha256?Expires=..."
      },
      {
        "name": "domainrisk/2024-11-19/domainrisk-20241119.1900-2000.json.gz",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 1850000,
        "url": "https://download.example.com/domainrisk/2024-11-19/domainrisk-
        ↪ 20241119.1900-2000.json.gz?Expires=..."
      }
    ]
  }
}
```

7.5.4 Download API response codes

Code	Status	Description
200	OK	The request was successful
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

7.5.5 Download API file contents

The *.json.gz.sha256 file is a checksum containing a SHA-256 hash value used to verify the integrity of the downloaded file.

The *.json.gz file, when uncompressed, contains JSON data in the same format as the Feed API response (NDJSON with timestamp, domain, and risk score fields).

7.5.6 Download API examples

List available files:

```
# Get the most recent files
curl -H 'X-API-Key: YOUR_API_KEY' \
```

```
'https://api.domaintools.com/v1/download/domainrisk/?limit=10'
```

Download and verify a file:

```
# Get the file list
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domainrisk/?limit=2' > files.json

# Extract the URL and download the data file
curl -o domainrisk-data.json.gz "$(jq -r '.response.files[1].url' files.json)"

# Download the checksum file
curl -o domainrisk-data.json.gz.sha256 "$(jq -r '.response.files[0].url'
↳ files.json)"

# Verify the integrity
sha256sum -c domainrisk-data.json.gz.sha256
```

Batch processing:

```
# Download multiple files in a loop
for url in $(curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domainrisk/?limit=24' | \
  jq -r '.response.files[].url' | grep '\.json\.gz$'); do
  curl -O "$url"
done
```

7.6 Daily Download API

The Daily Download API provides daily batch summaries as an alternative to hourly real-time data. Use this when you need daily aggregated data rather than real-time updates. The Daily Domain Risk feed is significantly larger than the real-time feed, containing approximately 30 million domains.

7.6.1 Overview

Daily feed of high-risk domains, regardless of observed traffic.

Inclusion threshold: Combined Risk score of 70+

Format: Tab-separated text file (gzipped); one domain per line with component risk scores

Size: ~30 million domains, ~400MB compressed

7.6.2 Base URL

```
https://api.domaintools.com/v1/download/daily_domain_risk/
```

7.6.3 Daily Download parameters

The Daily Download API supports standard download parameters:

7.6.3.1 api_username

Type: string (required for HMAC and open key auth)

Your DomainTools API username

7.6.3.2 api_key

Type: string (required for open key auth)

Your DomainTools API key

7.6.3.3 signature

Type: string (required for HMAC auth)

HMAC signature of your request

7.6.3.4 timestamp

Type: string (required for HMAC auth)

Current timestamp for HMAC authentication in ISO 8601 format

7.6.3.5 limit

Type: integer (optional)

Limit the list of signed files. Ordering of files is always descending, so the latest files are first.

7.6.3.6 page

Type: integer (optional)

Select which page of results are returned. Pages begin at 0 with latest results.

7.6.3.7 prefix

Type: string (optional)

Filter results by date using the file prefix.

7.6.4 Daily Download response structure

The API returns a JSON response with signed URLs for downloadable files:

download_name (string): The feed identifier (`daily_domain_risk`)

files (array): List of downloadable file entries

Each file object contains:

Field	Type	Description
<code>name</code>	string	File path (e.g., <code>domain_risk_feed/threat_profile_proximity.gz</code>)
<code>last_modified</code>	string	Last modified date in ISO 8601 format
<code>etag</code>	string	Entity tag (hash of the file)
<code>size</code>	integer	Size in bytes

Field	Type	Description
url	string	Signed AWS download URL (valid for 12 hours)

7.6.5 Daily Download response codes

Code	Status	Description
200	OK	The request was successful
400	Bad request	Invalid request parameters
401	Unauthorized	Missing or invalid authentication
403	Forbidden	Insufficient permissions
404	No data to download	No files available

7.6.6 Daily Download file naming

Daily files follow this naming pattern:

```
domain_risk_feed/threat_profile_proximity.gz
```

7.6.7 File contents

The tab-separated file contains one domain per line with the following fields:

- Domain Name
- Phishing (risk score)
- Malware (risk score)
- Spam (risk score)
- Proximity (risk score)
- Overall (combined risk score - equals highest of the component scores)

7.6.8 Daily Download examples

List available files:

```
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/daily_domain_risk/'
```

Download a specific file:

```
# Get the file list
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/daily_domain_risk/?limit=1' > files.json

# Download the file
curl -o daily-domain-risk.gz "$(jq -r '.response.files[0].url' files.json)"
```

```
# Decompress and view  
gunzip daily-domain-risk.gz  
head daily-domain-risk
```

7.7 Related resources

- [Domain Hotlist feed](#)
- [Newly Active Domains feed](#)
- [Domain Risk Score user guide](#)

8 IP Hotlist

The IP Hotlist feed identifies high-risk IP addresses hosting hostile domains that are observed to be active within a 24-hour time window. This focused feed provides risk scores and enrichment data for IPs where more than 50% of hosted domains are high-risk and actively communicating.

8.1 Overview

This feed captures IP addresses that meet strict criteria for both risk level and recent activity, making it ideal for immediate blocking and threat response. The feed provides the same comprehensive enrichment data as the IP Risk feed, but filtered to show only the most dangerous and currently active infrastructure.

Use this feed when you need to:

- Build high-confidence IP block lists
- Identify currently active hostile infrastructure for immediate action
- Enhance SOC and Threat Intel workflows with IP-based enrichment
- Create custom network or endpoint block rules
- Triage IP-based alerts
- Monitor threat actor hosting infrastructure
- Detect and respond to active C2 servers

Inclusion criteria: More than 50% of domains on the IP have proximity score of 70+ OR Threat Profile score of 90+; pDNS activity on malicious domains within 24 hours.

Format: Gzip-compressed tab-separated (TSV) text file

Size: 40-50,000 IP addresses, ~1MB compressed

8.2 Requirements

You need the following to access Threat Feeds:

- An Enterprise Account with DomainTools, accessible at <https://account.domaintools.com/my-account/>
- Authentication credentials (API key for header authentication, or API username and key for HMAC or open key authentication)
- A way to interact with a REST API delivered through AWS

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

For assistance, contact enterprisesupport@domaintools.com.

8.3 Authentication

You can authenticate to the IP Hotlist API using three different methods. Choose the method that best fits your security requirements and technical environment.

8.3.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

Required header:

X-Api-Key: YOUR_API_KEY

Example:

```
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/daily_ip_hotlist/'
```

8.3.2 HMAC authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret, providing integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials shouldn't be stored in plain text or included directly in request URLs.

DomainTools supports MD5, SHA1, and SHA256 for the hashing algorithm.

Required query parameters:

- `api_username`: Your DomainTools API username
- `signature`: HMAC-SHA1 signature of `api_username + timestamp + uri_path`
- `timestamp`: Current UTC timestamp in ISO 8601 format (for example, 2025-06-01T15:30:00Z)

Constructing the HMAC signature:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Important: URI path must include API version

The `uri_path` parameter must include the API version prefix. For example, use `/v1/feed/nod/` not `/feed/nod/`.

Example Python signing function:

```
import hmac  
import hashlib  
  
def sign(api_username, api_key, timestamp, uri):  
    params = f"{api_username}{timestamp}{uri}"  
    return hmac.new(api_key.encode("utf-8"), params.encode("utf-8"),  
        ↪ hashlib.sha1).hexdigest()
```

Note: HMAC timestamp requirements

The `timestamp` parameter in HMAC authentication must be current (within a few minutes of the server time). The timestamps shown in these examples are static for demonstration purposes. In production, generate a fresh timestamp for each

request using your system's current time in ISO 8601 UTC format (e.g., 2025-01-06T15:30:00Z).

Example:

```
curl 'https://api.domaintools.com/v1/download/daily_ip_hotlist/?api_username=YOUR_-\n↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

8.3.3 Open key authentication

This is the easiest authentication scheme to implement, but also the least secure. Each request contains the full API key and API username as query parameters. We recommend using [API key header authentication](#) or [HMAC authentication](#) instead.

If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

Required query parameters:

- `api_username`: Your API username
- `api_key`: Your API key

Example:

```
curl 'https://api.domaintools.com/v1/download/daily_ip_hotlist/?api_username=YOUR_-\n↳ USERNAME&api_key=YOUR_API_KEY'
```

8.4 Daily Download API

The Daily Download API provides access to IP Hotlist data through temporary AWS S3 file links. The feed is updated daily with high-risk, actively communicating IPs.

8.4.1 Base URL

```
https://api.domaintools.com/v1/download/daily_ip_hotlist/
```

8.4.2 Parameters

The Daily Download API supports standard download parameters:

8.4.2.1 `api_username`

Type: string (required for HMAC and open key auth)

Your DomainTools API username

8.4.2.2 `api_key`

Type: string (required for open key auth)

Your DomainTools API key

8.4.2.3 signature

Type: string (required for HMAC auth)

HMAC signature of your request

8.4.2.4 timestamp

Type: string (required for HMAC auth)

Current timestamp for HMAC authentication in ISO 8601 format

8.4.2.5 limit

Type: integer (optional)

Limit the list of signed files. Ordering of files is always descending, so the latest files are first.

8.4.2.6 page

Type: integer (optional)

Select which page of results are returned. Pages begin at 0 with latest results.

8.4.3 Response structure

The API returns a JSON response with signed URLs for downloadable files:

download_name (string): The feed identifier (`daily_ip_hotlist`)

files (array): List of downloadable file entries

Each file object contains:

- **name** (string): File path
- **last_modified** (string): Last modified date in ISO 8601 format
- **etag** (string): Entity tag (hash of the file)
- **size** (integer): Size in bytes
- **url** (string): Signed AWS CloudFront download URL (valid for 12 hours)

8.4.4 Response codes

200: OK - The request was successful

400: Bad request

401: Unauthorized

403: Forbidden

404: No data to download

8.4.5 File naming

The feed provides a single file:

```
ip_hotlist.gz
```

This file contains high-risk IP addresses with recent malicious activity, updated daily.

8.4.6 File contents

The TSV file contains the following fields (tab-separated, one IP per line):

8.4.6.1 IP and infrastructure fields

Field	Description
ip	IP address that has www/apex domains pointing to it
asn	The IP's ASN (autonomous system number, routing provider)
organization	Organization associated with IP range based on geo data
city	City based on IP geo data
region	Region based on IP geo data
country	Country based on IP geo data
latitude	Geographic coordinates
longitude	Geographic coordinates

8.4.6.2 Domain activity metrics

Field	Description
pdns_resolutions	Number of domains seen on the IP in the last 24 hours
bad_pdns_resolutions	Number of confirmed bad domains seen on the IP in the last 24 hours
total_domains	Total number of domains seen on this IP in the last 7 days
zerolist_domains	Number of zero-listed domains seen on this IP
zerolist_ip	Indicates if this IP is zero-listed (e.g., CDN)

8.4.6.3 Threat intelligence metrics

Field	Description
third_party_threats	Number of domains on IP confirmed with any threat on a third-party intel feed
allthreats_combined_count	Number of confirmed or predicted domains on third-party intel feed or threat profile
allthreats_combined_percent	Percentage of domains that are confirmed or predicted malicious
all_threats_percent	Percentage of domains including all threat types

8.4.6.4 Combined threat predictions

Field	Description
malicious_combined_phishing_percent	Percentage of domains confirmed or predicted as phishing
combined_malware_percent	Percentage of domains confirmed or predicted as malware
combined_spam_percent	Percentage of domains confirmed or predicted as spam

8.4.6.5 Confirmed malicious threats

Field	Description
malicious_phishing	Number of malicious phishing domains on third-party intel feeds
malicious_malware	Number of malicious malware domains on third-party intel feeds
malicious_spam	Number of malicious spam domains on third-party intel feeds
percent_phishing	Percentage of domains that are confirmed phishing
percent_malware	Percentage of domains that are confirmed malware
percent_spam	Percentage of domains that are confirmed spam

8.4.6.6 Compromised threats

Field	Description
compromised_phishing	Number of compromised phishing domains on third-party intel feeds
compromised_malware	Number of compromised malware domains on third-party intel feeds
compromised_spam	Number of compromised spam domains on third-party intel feeds

8.4.6.7 Predicted threats

Field	Description
predicted_phishing	Number of domains (with no confirmed threat) predicted as phishing
predicted_malware	Number of domains (with no confirmed threat) predicted as malware
predicted_spam	Number of domains (with no confirmed threat) predicted as spam

8.4.7 Examples

List available files:

```
curl -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/daily_ip_hotlist/'
```

Download the file:

```
# Get the file list  
curl -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/daily_ip_hotlist/' > files.json  
  
# Download the file  
curl -o ip_hotlist.gz "$(jq -r '.response.files[0].url' files.json)"  
  
# Decompress and view  
gunzip ip_hotlist.gz  
head ip_hotlist
```

Parse TSV data:

```
# View first 10 IPs with their threat percentages  
gunzip -c ip_hotlist.gz | head -10 | cut -f1,6,7,8,9
```

Filter for specific threat types:

```
# Find IPs with high phishing percentage (field 7)  
gunzip -c ip_hotlist.gz | awk -F'\t' '$7 > 50 {print $1, $7}' | head -20
```

8.5 Related resources

- [IP Risk feed](#)
- [Domain Hotlist feed](#)
- [Domain Risk feed](#)

9 IP Risk

The IP Risk feed provides comprehensive risk intelligence for all IP addresses known to be hosting domains, regardless of risk level. This feed includes extensive enrichment data such as threat scores, geographic information, ASN details, and domain hosting metrics.

9.1 Overview

This feed captures all IP addresses actively hosting one or more domains, providing detailed risk assessment and enrichment data for each IP. The feed includes both confirmed threats from third-party intelligence feeds and predictive risk scores based on DomainTools machine learning models.

Use this feed when you need to:

- Monitor IP addresses hosting domains for threat intelligence
- Analyze hosting infrastructure risk patterns
- Correlate IP-based threats with domain activity
- Build IP reputation databases
- Detect suspicious hosting patterns
- Enrich security alerts with IP risk context
- Track threat actor infrastructure

Inclusion criteria: IP is actively hosting one or more domains (regardless of risk level).

Format: Gzip-compressed tab-separated (TSV) text file

Size: 15-20 million IP addresses, ~200MB compressed

9.2 Requirements

You need the following to access Threat Feeds:

- An Enterprise Account with DomainTools, accessible at <https://account.domaintools.com/my-account/>
- Authentication credentials (API key for header authentication, or API username and key for HMAC or open key authentication)
- A way to interact with a REST API delivered through AWS

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

For assistance, contact enterprisesupport@domaintools.com.

9.3 Authentication

You can authenticate to the IP Risk API using three different methods. Choose the method that best fits your security requirements and technical environment.

9.3.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

Required header:

X-Api-Key: YOUR_API_KEY

Example:

```
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/daily_ip_risk/'
```

9.3.2 HMAC authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret, providing integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials shouldn't be stored in plain text or included directly in request URLs.

DomainTools supports MD5, SHA1, and SHA256 for the hashing algorithm.

Required query parameters:

- `api_username`: Your DomainTools API username
- `signature`: HMAC-SHA1 signature of `api_username + timestamp + uri_path`
- `timestamp`: Current UTC timestamp in ISO 8601 format (for example, `2025-06-01T15:30:00Z`)

Constructing the HMAC signature:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Important: URI path must include API version

The `uri_path` parameter must include the API version prefix. For example, use `/v1/feed/nod/` not `/feed/nod/`.

Example Python signing function:

```
import hmac  
import hashlib  
  
def sign(api_username, api_key, timestamp, uri):  
    params = f"{api_username}{timestamp}{uri}"  
    return hmac.new(api_key.encode("utf-8"), params.encode("utf-8"),  
        ↪ hashlib.sha1).hexdigest()
```

Note: HMAC timestamp requirements

The `timestamp` parameter in HMAC authentication must be current (within a few minutes of the server time). The timestamps shown in these examples are static for demonstration purposes. In production, generate a fresh timestamp for each request using your system's current time in ISO 8601 UTC format (e.g., `2025-01-06T15:30:00Z`).

Example:

```
curl 'https://api.domaintools.com/v1/download/daily_ip_risk/?api_username=YOUR_-\n↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

9.3.3 Open key authentication

This is the easiest authentication scheme to implement, but also the least secure. Each request contains the full API key and API username as query parameters. We recommend using [API key header authentication](#) or [HMAC authentication](#) instead.

If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

Required query parameters:

- `api_username`: Your API username
- `api_key`: Your API key

Example:

```
curl 'https://api.domaintools.com/v1/download/daily_ip_risk/?api_username=YOUR_-\n↳ USERNAME&api_key=YOUR_API_KEY'
```

9.4 Daily Download API

The Daily Download API provides access to IP Risk data through temporary AWS S3 file links. The feed is updated daily with comprehensive risk intelligence for all IPs hosting domains.

9.4.1 Base URL

```
https://api.domaintools.com/v1/download/daily_ip_risk/
```

9.4.2 Parameters

The Daily Download API supports standard download parameters:

9.4.2.1 `api_username`

Type: string (required for HMAC and open key auth)

Your DomainTools API username

9.4.2.2 `api_key`

Type: string (required for open key auth)

Your DomainTools API key

9.4.2.3 `signature`

Type: string (required for HMAC auth)

HMAC signature of your request

9.4.2.4 timestamp

Type: string (required for HMAC auth)

Current timestamp for HMAC authentication in ISO 8601 format

9.4.2.5 limit

Type: integer (optional)

Limit the list of signed files. Ordering of files is always descending, so the latest files are first.

9.4.2.6 page

Type: integer (optional)

Select which page of results are returned. Pages begin at 0 with latest results.

9.4.3 Response structure

The API returns a JSON response with signed URLs for downloadable files:

download_name (string): The feed identifier (`daily_ip_risk`)

files (array): List of downloadable file entries

Each file object contains:

- **name** (string): File path
- **last_modified** (string): Last modified date in ISO 8601 format
- **etag** (string): Entity tag (hash of the file)
- **size** (integer): Size in bytes
- **url** (string): Signed AWS CloudFront download URL (valid for 12 hours)

9.4.4 Response codes

200: OK - The request was successful

400: Bad request

401: Unauthorized

403: Forbidden

404: No data to download

9.4.5 File naming

The feed provides a single file:

```
ip_fulllist.gz
```

This file contains all IP addresses actively hosting domains, updated daily.

9.4.6 File contents

The TSV file contains the following fields (tab-separated, one IP per line):

9.4.6.1 IP and infrastructure fields

Field	Description
ip	IP address that has www/apex domains pointing to it
asn	The IP's ASN (autonomous system number, routing provider)
organization	Organization associated with IP range based on geo data
city	City based on IP geo data
region	Region based on IP geo data
country	Country based on IP geo data
latitude	Geographic coordinates
longitude	Geographic coordinates

9.4.6.2 Domain activity metrics

Field	Description
pdns_resolutions	Number of domains seen on the IP in the last 24 hours
bad_pdns_resolutions	Number of confirmed bad domains seen on the IP in the last 24 hours
total_domains	Total number of domains seen on this IP in the last 7 days
zerolist_domains	Number of zero-listed domains seen on this IP
zerolist_ip	Indicates if this IP is zero-listed (e.g., CDN)

9.4.6.3 Threat intelligence metrics

Field	Description
third_party_threats	Number of domains on IP confirmed with any threat on a third-party intel feed
allthreats_combined_count	Number of confirmed or predicted domains on third-party intel feed or threat profile
allthreats_combined_percent	Percentage of domains that are confirmed or predicted malicious
all_threats_percent	Percentage of domains including all threat types

9.4.6.4 Combined threat predictions

Field	Description
malicious_combined_phishing_percent	Percentage of domains confirmed or predicted as phishing

Field	Description
combined_malware_percent	Percentage of domains confirmed or predicted as malware
combined_spam_percent	Percentage of domains confirmed or predicted as spam

9.4.6.5 Confirmed malicious threats

Field	Description
malicious_phishing	Number of malicious phishing domains on third-party intel feeds
malicious_malware	Number of malicious malware domains on third-party intel feeds
malicious_spam	Number of malicious spam domains on third-party intel feeds
percent_phishing	Percentage of domains that are confirmed phishing
percent_malware	Percentage of domains that are confirmed malware
percent_spam	Percentage of domains that are confirmed spam

9.4.6.6 Compromised threats

Field	Description
compromised_phishing	Number of compromised phishing domains on third-party intel feeds
compromised_malware	Number of compromised malware domains on third-party intel feeds
compromised_spam	Number of compromised spam domains on third-party intel feeds

9.4.6.7 Predicted threats

Field	Description
predicted_phishing	Number of domains (with no confirmed threat) predicted as phishing
predicted_malware	Number of domains (with no confirmed threat) predicted as malware
predicted_spam	Number of domains (with no confirmed threat) predicted as spam

9.4.7 Examples

List available files:

```
curl -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/daily_ip_risk/'
```

Download the file:

```
# Get the file list  
curl -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/daily_ip_risk/' > files.json  
  
# Download the file  
curl -o ip_fulllist.gz "$(jq -r '.response.files[0].url' files.json)"  
  
# Decompress and view  
gunzip ip_fulllist.gz  
head ip_fulllist
```

Parse TSV data:

```
# View first 10 IPs with their threat percentages  
gunzip -c ip_fulllist.gz | head -10 | cut -f1,6,7,8,9
```

9.5 Related resources

- [IP Hotlist feed](#)
- [Domain Risk feed](#)
- [Domain Hotlist feed](#)

10 Newly Active Domains (NAD)

The NAD feed lists domains that have become active in our global passive DNS sensor network, either for the first time or after a period of inactivity (at least 10 days). This helps identify when a previously dormant domain is being repurposed, a common tactic for attackers.

10.1 Overview

Domains are apex-level (for example, `example.com` but not `www.example.com`), and the feed emits them as they're observed.

Use this feed when you need to:

- Detect the reactivation of previously dormant domains
- Identify potentially suspicious infrastructure
- Monitor domains that might be used for malicious activities
- Integrate as a blocklist into DNS resolvers via RPZ

Inclusion criteria: Domains observed in passive DNS to be newly active in the latest lifecycle of the domain, either for the first time or after an inactive period of at least 10 days.

10.2 Requirements

You need the following to access Threat Feeds:

- An Enterprise Account with DomainTools, accessible at <https://account.domaintools.com/my-account/>
- Authentication credentials (API key for header authentication, or API username and key for HMAC or open key authentication)
- A way to interact with a REST API delivered through AWS

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

For assistance, contact enterprisesupport@domaintools.com.

10.3 Authentication

You can authenticate to the NAD APIs using three different methods. Choose the method that best fits your security requirements and technical environment.

10.3.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

Required header:

```
X-Api-Key: YOUR_API_KEY
```

Examples:

```
# Feed API request
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nad/?sessionID=mySOC'
```

```
# Download API request
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/nad/'
```

10.3.2 HMAC authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret, providing integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials shouldn't be stored in plain text or included directly in request URLs.

DomainTools supports MD5, SHA1, and SHA256 for the hashing algorithm.

Required query parameters:

- `api_username`: Your DomainTools API username
- `signature`: HMAC-SHA1 signature of `api_username` + `timestamp` + `uri_path`
- `timestamp`: Current UTC timestamp in ISO 8601 format (for example, `2025-06-01T15:30:00Z`)

Constructing the HMAC signature:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Important: URI path must include API version

The `uri_path` parameter must include the API version prefix. For example, use `/v1/feed/nod/` not `/feed/nod/`.

Example Python signing function:

```
import hmac
import hashlib

def sign(api_username, api_key, timestamp, uri):
    params = f"{api_username}{timestamp}{uri}"
    return hmac.new(api_key.encode("utf-8"), params.encode("utf-8"),
        ↪ hashlib.sha1).hexdigest()
```

Note: HMAC timestamp requirements

The `timestamp` parameter in HMAC authentication must be current (within a few minutes of the server time). The timestamps shown in these examples are static for demonstration purposes. In production, generate a fresh timestamp for each request using your system's current time in ISO 8601 UTC format (e.g., `2025-01-06T15:30:00Z`).

Examples:

```
# Feed API request with HMAC
curl 'https://api.domaintools.com/v1/feed/nad/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-
↳ 06T15:30:00Z&sessionID=mySOC'
```

```
# Download API request with HMAC
curl 'https://api.domaintools.com/v1/download/nad/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

10.3.3 Open key authentication

This is the easiest authentication scheme to implement, but also the least secure. Each request contains the full API key and API username as query parameters. We recommend using [API key header authentication](#) or [HMAC authentication](#) instead.

If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

Required query parameters:

- `api_username`: Your API username
- `api_key`: Your API key

Examples:

```
# Feed API request
curl 'https://api.domaintools.com/v1/feed/nad/?api_username=YOUR_USERNAME&api_
↳ key=YOUR_API_KEY&sessionID=mySOC'
```

```
# Download API request
curl 'https://api.domaintools.com/v1/download/nad/?api_username=YOUR_USERNAME&api_
↳ key=YOUR_API_KEY'
```

10.4 Real-time Feed API

The Feed API provides real-time access to current NAD data. Use this API to poll for the latest feed updates at regular intervals, maintain a session to track your position in the feed, and filter results based on your specific needs.

10.4.1 Base URL

```
https://api.domaintools.com/v1/feed/nad/
```

10.4.2 Rate limits

Real-time feeds have the following rate limits:

- 2 queries per minute
- 120 queries per hour

If you exceed these limits, the API returns an error.

10.4.3 Response formats

The API supports two response formats:

NDJSON (Newline-Delimited JSON)

- Default format when no Accept header is specified
- Also known as JSON Lines (JSONL)
- One JSON object per line
- Efficient for streaming and processing large datasets
- Set Accept: `application/x-ndjson` to explicitly request this format

CSV (Comma-Separated Values)

- Set Accept: `text/csv` to request CSV format
- Add `&headers=1` to the query parameters to include column headers as the first line
- Not available for all feeds; check the specific feed documentation for CSV support

10.4.4 Session management

Session management allows you to maintain your position in the feed data stream, ensuring you don't miss or duplicate events when polling the API.

How sessions work:

- **Start a new session:** Provide a unique `sessionId` parameter of your choosing. By default, the API returns the past hour of results.
- **Resume a session:** Use the same `sessionId` in subsequent requests. The API returns all data since your last request.
- **Handle large result sets:** If a single request exceeds 10M results, the API returns an HTTP 206 response code. Repeat the same request with the same `sessionId` to receive the next batch of data until you receive an HTTP 200 response code.
- **One request at a time:** Do not send simultaneous requests with the same `sessionId` for the same feed. Wait for each request to complete before sending the next one. Concurrent requests with the same `sessionId` can produce errors or incomplete results.
- **Delete a session:** Use an HTTP DELETE request with your `sessionId` to clear the saved offset and start fresh.

Session ID requirements:

- 1 to 64 characters in length
- Alphanumeric characters and hyphens only (`[a-zA-Z0-9-]+`)
- Case-sensitive

10.4.5 Quick start

The standard access pattern is to periodically request the most recent feed data, as often as every 60 seconds.

```
curl -H 'X-API-Key: YOUR_API_KEY' \
'https://api.domaintools.com/v1/feed/nad/?sessionID=mySOC'
```

This starts a new session and returns the last hour of data. Subsequent calls with the same sessionID return data since the last request.

10.4.6 Feed API parameters

10.4.6.1 sessionID

Type: String

Valid values: 1-64 alphanumeric characters and hyphens ([a-zA-Z0-9-]+)

Description: A unique identifier for the session, used for resuming data retrieval from the last point. Use a new sessionID to begin a new session, fetching the most recent hour by default. Reuse the same sessionID to return all feed data since your last request. If omitted, time window parameters (such as after/before) are required.

Example: sessionID=mySOC

Required: Yes, to continue where you left off (or use after/before instead)

10.4.6.2 after

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The start of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: after=-60 or after=2024-10-16T10:20:00Z

Required: Yes, if before or sessionID not provided

10.4.6.3 before

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The end of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: before=-120 or before=2024-10-16T10:20:00Z

Required: Yes, if after or sessionID not provided

10.4.6.4 domain

Type: String

Valid values: Domain character set restricted by the DNS specification (letters, digits, hyphens). International characters should be specified in punycode. A trailing dot is acceptable.

Description: Filter for an exact domain or a domain substring by prefixing or suffixing your string with *. Multiple parameters are supported (for example, `?domain=*apple*&domain=*microsoft*`). The URL-encoded version of * (%2A) may be required in some clients.

Example: `domain=*bank*` or `domain=example.com`

Required: No

10.4.6.5 fromBeginning

Type: Boolean

Valid values: true

Description: Requires a `sessionId`. When used with a new session ID, returns the first hour of data in the time window (rather than the last). Returns an error if the session ID already exists — drop `fromBeginning` from subsequent requests after the first call. Only the value true is accepted; any other value (including false) is ignored.

Example: `fromBeginning=true`

Required: No

10.4.6.6 top

Type: Integer

Valid values: Positive integer, 1-1,000,000,000

Description: Limits the number of results in the response payload. Primarily intended for testing. When you apply this parameter to risk feeds, results are sorted by `overall_risk` (descending).

Example: `top=10`

Required: No

10.4.6.7 headers

Type: Integer

Valid values: 1

Description: Adds a header row as the first line of the response when `text/csv` is requested. Set `headers=1` to enable. Only applies when requesting CSV format. Only the value 1 is accepted; any other value is invalid.

Example: `headers=1`

Required: No

10.4.7 Feed API response structure

The API returns responses in NDJSON (Newline-Delimited JSON), with each response containing one domain entry per line. Each entry contains a timestamp in ISO 8601 UTC format, and the domain.

Response fields:

timestamp (string): The observation timestamp in ISO 8601 UTC format.

- Example: "timestamp":"2024-11-15T16:14:39Z"

domain (string): The domain name without the trailing dot. Domain character set restricted by the DNS specification (letters, digits, hyphens).

- Example: "domain":"example.com"

Example NDJSON response:

```
{"timestamp":"2024-11-15T16:14:39Z","domain":"domiantools.com"}
{"timestamp":"2024-11-15T16:14:38Z","domain":"domsintools.com"}
{"timestamp":"2024-11-15T16:14:36Z","domain":"edomaintools.com"}
{"timestamp":"2024-11-15T16:14:35Z","domain":"omaintools.com"}
{"timestamp":"2024-11-15T16:14:35Z","domain":"v-domaintools.com"}
```

Example CSV response:

```
timestamp, domain
2024-11-15T16:14:39Z, domiantools.com
2024-11-15T16:14:38Z, domsintools.com
2024-11-15T16:14:36Z, edomaintools.com
```

10.4.8 Feed API response codes

Code	Status	Description
200	OK	The request was successful and all data has been delivered
206	Partial content	The request was successful, but only a portion of the data was returned. The request exceeded 10M results or the 1-hour evaluation window. Repeat the same request with the same <code>sessionID</code> to receive the next batch of data until you receive an HTTP 200 response
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
404	Not found	The requested resource (such as a <code>sessionID</code>) doesn't exist

Code	Status	Description
406	Not acceptable	The specified Accept header value isn't supported. Only application/x-ndjson and text/csv are accepted
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

10.4.9 Feed API examples

Basic session polling:

```
# Start a new session
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nad/?sessionID=mySOC'
```

```
# Resume the session (returns data since last request)
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nad/?sessionID=mySOC'
```

Time window filtering:

```
# Get data from a specific time range
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nad/?after=2025-01-06T10:00:00Z&before=2025-
↵ 01-06T11:00:00Z'
```

Domain filtering:

```
# Filter for specific domain patterns
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nad/?domain=*.example.com&sessionID=mySOC'
```

CSV format:

```
# Request CSV format with headers
curl -H 'Accept: text/csv' -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nad/?headers=1&sessionID=mySOC'
```

```
# Request CSV format without headers
curl -H 'Accept: text/csv' -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nad/?sessionID=mySOC'
```

Handling large result sets:

```
# If you receive HTTP 206, repeat the request to get the next batch
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nad/?sessionID=mySOC'
```

```
# Repeat until you receive HTTP 200
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nad/?sessionID=mySOC'
```

Delete a session:

```
# Clear the saved offset and start fresh
curl -X DELETE -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nad/?sessionID=mySOC'
```

10.5 Download API

The Download API provides access to historical NAD data through temporary AWS S3 file links. Use this API to retrieve archived data you may have missed or to backfill your systems with historical information. Files are organized by hour and available for 90 days.

10.5.1 Base URL

```
https://api.domaintools.com/v1/download/nad/
```

10.5.2 Download API parameters

10.5.2.1 limit

Type: Integer

Valid values: Positive integer

Description: Limits the number of files returned in the response, starting from the most recent. Use to control payload size or test specific cases.

Example: limit=10

Required: No

10.5.3 Download API response structure

The API returns a JSON response containing an array of downloadable files. Each file entry includes:

download_name (string): The feed identifier (nad)

files (array): List of downloadable file entries

Each file object contains:

- **name** (string): Path and filename of the downloadable file

- **last_modified** (string): Timestamp of last modification in ISO 8601 UTC format
- **etag** (string): ETag (hash) used to verify file identity and versioning
- **size** (integer): File size in bytes
- **url** (string): Temporary signed URL to download the file from AWS

File naming convention:

- Data file: `nad/{YYYY-MM-DD}/nad-{YYYYMMDD}.{HH00}-{HH00}.json.gz`
- Checksum file: `nad/{YYYY-MM-DD}/nad-{YYYYMMDD}.{HH00}-{HH00}.json.gz.sha256`

Example response:

```
{
  "response": {
    "download_name": "nad",
    "files": [
      {
        "name": "nad/2024-11-19/nad-20241119.1900-2000.json.gz.sha256",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 64,
        "url": "https://download.example.com/nad/2024-11-19/nad-20241119.1900-
↵ 2000.json.gz.sha256?Expires=..."
      },
      {
        "name": "nad/2024-11-19/nad-20241119.1900-2000.json.gz",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 140725,
        "url": "https://download.example.com/nad/2024-11-19/nad-20241119.1900-
↵ 2000.json.gz?Expires=..."
      }
    ]
  }
}
```

10.5.4 Download API response codes

Code	Status	Description
200	OK	The request was successful
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

10.5.5 Download API file contents

The `*.json.gz.sha256` file is a checksum containing a SHA-256 hash value used to verify the integrity of the downloaded file.

The *.json.gz file, when uncompressed, contains JSON data in the same format as the Feed API response (NDJSON with timestamp and domain fields).

10.5.6 Download API examples

List available files:

```
# Get the most recent files
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/nad/?limit=10'
```

Download and verify a file:

```
# Get the file list
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/nad/?limit=2' > files.json

# Extract the URL and download the data file
curl -o nad-data.json.gz "$(jq -r '.response.files[1].url' files.json)"

# Download the checksum file
curl -o nad-data.json.gz.sha256 "$(jq -r '.response.files[0].url' files.json)"

# Verify the integrity
sha256sum -c nad-data.json.gz.sha256
```

Batch processing:

```
# Download multiple files in a loop
for url in $(curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/nad/?limit=24' | \
  jq -r '.response.files[] .url' | grep '\.json\.gz$'); do
  curl -O "$url"
done
```

10.6 RPZ Access

The NAD feed is available via Response Policy Zone (RPZ) for direct integration with DNS firewalls.

10.6.1 Overview

Response Policy Zone (RPZ) provides DNS-level blocking by integrating threat feeds directly into your DNS resolver. This allows you to automatically block or redirect DNS queries for domains in the feed, providing real-time protection before connections are established.

How RPZ works:

When a user attempts to access a domain in the RPZ feed, your DNS resolver responds with an NXDOMAIN (“no such domain”) status code, effectively making the domain unavailable. This blocks malicious domains at the DNS layer, preventing endpoint communication before any connection is established.

Benefits:

- **Real-time protection:** Blocks threats at the DNS resolver level
- **Automatic updates:** Receives updates via DNS zone transfers (AXFR/IXFR)
- **No client configuration:** Works transparently for all clients using your DNS resolver
- **Efficient:** Minimal performance impact on DNS resolution
- **Standard protocol:** Uses DNS Response Policy Zones specification

Zone naming format:

RPZ zones are named using the pattern: [interval].[feed].rpz.domaintools.com

Available time intervals: 5m, 10m, 30m, 1h, 3h, 12h (and 24h for some feeds)

Larger time intervals are supersets that include smaller intervals. Smaller intervals have smaller zone sizes and may be available faster.

10.6.2 Available zones

The NAD feed is available in the following RPZ zones:

- 5m.nad.rpz.domaintools.com - Last 5 minutes
- 10m.nad.rpz.domaintools.com - Last 10 minutes
- 30m.nad.rpz.domaintools.com - Last 30 minutes
- 1h.nad.rpz.domaintools.com - Last 1 hour
- 3h.nad.rpz.domaintools.com - Last 3 hours
- 12h.nad.rpz.domaintools.com - Last 12 hours

10.6.3 Configuration

Configuration requirements:

To access RPZ feeds, you need to:

1. **Provide IP addresses:** Contact enterprisesupport@domaintools.com with:
 - IP address(es) from which you connect to the RPZ provider DNS server
 - IP address(es) to receive DNS NOTIFY messages (typically the same)
2. **Configure firewall:** Add rules to allow DomainTools hosts to send UDP packets to port 53:
 - IPv4: 104.244.13.88 Port: 53
 - IPv4: 104.244.14.88 Port: 53
3. **Set up TSIG authentication:** DomainTools uses TSIG (Secret Key Transaction Authentication) for authorization:
 - TSIG key algorithm: hmac-sha512
 - TSIG key and key name: Provided by DomainTools Enterprise Support

Delivery method:

RPZ feeds are delivered via:

- Incremental Zone Transfers (IXFR)
- Full zone transfers (AXFR)
- DNS NOTIFY messages to trigger zone updates

For detailed configuration instructions, including DNS resolver setup, advanced features (allowlists, walled gardens, logging), and troubleshooting, see the [Response Policy Zone documentation](#).

10.6.4 Testing

Testing your RPZ configuration:

When your DNS resolver blocks a domain using RPZ, the response includes an SOA (Start of Authority) record that identifies which RPZ feed was used.

Test domain:

Each RPZ feed includes a test domain entry: `test.rpz.domaintools.test`

Use this to verify the RPZ feed is loaded and working. A successful test returns:

- NXDOMAIN status code
- SOA record in the ADDITIONAL section showing the specific feed name

Example SOA record:

```
;; ADDITIONAL SECTION:
3h.nad.rpz.domaintools.com. 86400 IN SOA rpz-ns1.domaintools.com.
↪ noc.domaintools.com. 946684799 600 300 86400 86400
```

The SOA SERIAL number (Unix epoch timestamp) indicates when the feed was last regenerated.

Troubleshooting:

If the SOA record appears in the AUTHORITY section instead of ADDITIONAL, or doesn't show the specific feed name, the response didn't come from RPZ. Check your DNS resolver's RPZ-related logs for additional debugging information.

10.6.5 RPZ examples

Test the RPZ feed:

```
# Query the test domain
dig test.rpz.domaintools.test

# Expected response includes NXDOMAIN and SOA record showing the feed name
# Example SOA in ADDITIONAL section:
# 3h.nad.rpz.domaintools.com. 86400 IN SOA rpz-ns1.domaintools.com. ...
```

Verify a specific zone is loaded:

```
# Test with the 1-hour NAD zone
dig @your-dns-server test.rpz.domaintools.test

# Check the SOA record shows: 1h.nad.rpz.domaintools.com
```

10.7 Related resources

- [Domain Risk Score user guide](#)
- [Newly Observed Domains feed](#)
- [Domain Hotlist feed](#)
- [Response Policy Zone documentation](#)

11 Newly Observed Domains (NOD)

The NOD feed provides a real-time list of domains that have been observed for the first time in our global passive DNS sensor network. This feed is ideal for brand protection, corporate intelligence, and temporarily blocking outbound connections to brand-new domains until their reputation can be assessed.

11.1 Overview

Domains are apex-level (for example, `example.com` but not `www.example.com`), and the feed emits them as they're observed.

Use this feed when you need to:

- Identify newly registered domains that might be used for typosquatting or other brand abuse
- Monitor the emergence of new domains relevant to your organization
- Temporarily block access to newly observed domains to mitigate risks until their legitimacy is verified
- Integrate as a blocklist into DNS resolvers via RPZ

Inclusion criteria: Domains observed in passive DNS for the first time.

11.2 Requirements

You need the following to access Threat Feeds:

- An Enterprise Account with DomainTools, accessible at <https://account.domaintools.com/my-account/>
- Authentication credentials (API key for header authentication, or API username and key for HMAC or open key authentication)
- A way to interact with a REST API delivered through AWS

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

For assistance, contact enterprisesupport@domaintools.com.

11.3 Authentication

You can authenticate to the NOD APIs using three different methods. Choose the method that best fits your security requirements and technical environment.

11.3.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

Required header:

X-Api-Key: YOUR_API_KEY

Examples:

```
# Feed API request
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nod/?sessionID=myThreatIntel'
```

```
# Download API request
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/nod/'
```

11.3.2 HMAC authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret, providing integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials shouldn't be stored in plain text or included directly in request URLs.

DomainTools supports MD5, SHA1, and SHA256 for the hashing algorithm.

Required query parameters:

- `api_username`: Your DomainTools API username
- `signature`: HMAC-SHA1 signature of `api_username` + `timestamp` + `uri_path`
- `timestamp`: Current UTC timestamp in ISO 8601 format (for example, `2025-06-01T15:30:00Z`)

Constructing the HMAC signature:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Important: URI path must include API version

The `uri_path` parameter must include the API version prefix. For example, use `/v1/feed/nod/` not `/feed/nod/`.

Example Python signing function:

```
import hmac
import hashlib

def sign(api_username, api_key, timestamp, uri):
    params = f"{api_username}{timestamp}{uri}"
    return hmac.new(api_key.encode("utf-8"), params.encode("utf-8"),
        ↪ hashlib.sha1).hexdigest()
```

Note: HMAC timestamp requirements

The `timestamp` parameter in HMAC authentication must be current (within a few minutes of the server time). The timestamps shown in these examples are static for demonstration purposes. In production, generate a fresh timestamp for each request using your system's current time in ISO 8601 UTC format (e.g., `2025-01-06T15:30:00Z`).

Examples:

```
# Feed API request with HMAC
curl 'https://api.domaintools.com/v1/feed/nod/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-
↳ 06T15:30:00Z&sessionID=myThreatIntel'
```

```
# Download API request with HMAC
curl 'https://api.domaintools.com/v1/download/nod/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

11.3.3 Open key authentication

This is the easiest authentication scheme to implement, but also the least secure. Each request contains the full API key and API username as query parameters. We recommend using [API key header authentication](#) or [HMAC authentication](#) instead.

If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

Required query parameters:

- `api_username`: Your API username
- `api_key`: Your API key

Examples:

```
# Feed API request
curl 'https://api.domaintools.com/v1/feed/nod/?api_username=YOUR_USERNAME&api_
↳ key=YOUR_API_KEY&sessionID=myThreatIntel'
```

```
# Download API request
curl 'https://api.domaintools.com/v1/download/nod/?api_username=YOUR_USERNAME&api_
↳ key=YOUR_API_KEY'
```

11.4 Real-time Feed API

The Feed API provides real-time access to current NOD data. Use this API to poll for the latest feed updates at regular intervals, maintain a session to track your position in the feed, and filter results based on your specific needs.

11.4.1 Base URL

```
https://api.domaintools.com/v1/feed/nod/
```

11.4.2 Rate limits

Real-time feeds have the following rate limits:

- 2 queries per minute
- 120 queries per hour

If you exceed these limits, the API returns an error.

11.4.3 Response formats

The API supports two response formats:

NDJSON (Newline-Delimited JSON)

- Default format when no Accept header is specified
- Also known as JSON Lines (JSONL)
- One JSON object per line
- Efficient for streaming and processing large datasets
- Set Accept: `application/x-ndjson` to explicitly request this format

CSV (Comma-Separated Values)

- Set Accept: `text/csv` to request CSV format
- Add `&headers=1` to the query parameters to include column headers as the first line
- Not available for all feeds; check the specific feed documentation for CSV support

11.4.4 Session management

Session management allows you to maintain your position in the feed data stream, ensuring you don't miss or duplicate events when polling the API.

How sessions work:

- **Start a new session:** Provide a unique `sessionId` parameter of your choosing. By default, the API returns the past hour of results.
- **Resume a session:** Use the same `sessionId` in subsequent requests. The API returns all data since your last request.
- **Handle large result sets:** If a single request exceeds 10M results, the API returns an HTTP 206 response code. Repeat the same request with the same `sessionId` to receive the next batch of data until you receive an HTTP 200 response code.
- **One request at a time:** Do not send simultaneous requests with the same `sessionId` for the same feed. Wait for each request to complete before sending the next one. Concurrent requests with the same `sessionId` can produce errors or incomplete results.
- **Delete a session:** Use an HTTP DELETE request with your `sessionId` to clear the saved offset and start fresh.

Session ID requirements:

- 1 to 64 characters in length
- Alphanumeric characters and hyphens only (`[a-zA-Z0-9-]+`)
- Case-sensitive

11.4.5 Quick start

The standard access pattern is to periodically request the most recent feed data, as often as every 60 seconds.

```
curl -H 'X-API-Key: YOUR_API_KEY' \
'https://api.domaintools.com/v1/feed/nod/?sessionID=myThreatIntel'
```

This starts a new session and returns the last hour of data. Subsequent calls with the same sessionID return data since the last request.

11.4.6 Feed API parameters

11.4.6.1 sessionID

Type: String

Valid values: 1-64 alphanumeric characters and hyphens ([a-zA-Z0-9-]+)

Description: A unique identifier for the session, used for resuming data retrieval from the last point. Use a new sessionID to begin a new session, fetching the most recent hour by default. Reuse the same sessionID to return all feed data since your last request. If omitted, time window parameters (such as after/before) are required.

Example: sessionID=mySOC

Required: Yes, to continue where you left off (or use after/before instead)

11.4.6.2 after

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The start of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: after=-60 or after=2024-10-16T10:20:00Z

Required: Yes, if before or sessionID not provided

11.4.6.3 before

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The end of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: before=-120 or before=2024-10-16T10:20:00Z

Required: Yes, if after or sessionID not provided

11.4.6.4 domain

Type: String

Valid values: Domain character set restricted by the DNS specification (letters, digits, hyphens). International characters should be specified in punycode. A trailing dot is acceptable.

Description: Filter for an exact domain or a domain substring by prefixing or suffixing your string with *. Multiple parameters are supported (for example, `?domain=*apple*&domain=*microsoft*`). The URL-encoded version of * (%2A) may be required in some clients.

Example: `domain=*bank*` or `domain=example.com`

Required: No

11.4.6.5 fromBeginning

Type: Boolean

Valid values: true

Description: Requires a `sessionId`. When used with a new session ID, returns the first hour of data in the time window (rather than the last). Returns an error if the session ID already exists — drop `fromBeginning` from subsequent requests after the first call. Only the value true is accepted; any other value (including false) is ignored.

Example: `fromBeginning=true`

Required: No

11.4.6.6 top

Type: Integer

Valid values: Positive integer, 1-1,000,000,000

Description: Limits the number of results in the response payload. Primarily intended for testing. When you apply this parameter to risk feeds, results are sorted by `overall_risk` (descending).

Example: `top=10`

Required: No

11.4.6.7 headers

Type: Integer

Valid values: 1

Description: Adds a header row as the first line of the response when `text/csv` is requested. Set `headers=1` to enable. Only applies when requesting CSV format. Only the value 1 is accepted; any other value is invalid.

Example: `headers=1`

Required: No

11.4.7 Feed API response structure

The API returns responses in NDJSON (Newline-Delimited JSON), with each response containing one domain entry per line. Each entry contains a `timestamp` in ISO 8601 UTC format, and the `domain`.

Response fields:

timestamp (string): The observation timestamp in ISO 8601 UTC format.

- Example: `"timestamp":"2024-11-15T16:14:39Z"`

domain (string): The domain name without the trailing dot. Domain character set restricted by the DNS specification (letters, digits, hyphens).

- Example: `"domain":"example.com"`

Example NDJSON response:

```
{"timestamp":"2024-11-15T16:14:39Z","domain":"domiantools.com"}
{"timestamp":"2024-11-15T16:14:38Z","domain":"domsintools.com"}
{"timestamp":"2024-11-15T16:14:36Z","domain":"edomaintools.com"}
{"timestamp":"2024-11-15T16:14:35Z","domain":"omaintools.com"}
{"timestamp":"2024-11-15T16:14:35Z","domain":"v-domaintools.com"}
```

Example CSV response:

```
timestamp, domain
2024-11-15T16:14:39Z, domiantools.com
2024-11-15T16:14:38Z, domsintools.com
2024-11-15T16:14:36Z, edomaintools.com
```

11.4.8 Feed API response codes

Code	Status	Description
200	OK	The request was successful and all data has been delivered
206	Partial content	The request was successful, but only a portion of the data was returned. The request exceeded 10M results or the 1-hour evaluation window. Repeat the same request with the same <code>sessionID</code> to receive the next batch of data until you receive an HTTP 200 response
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
404	Not found	The requested resource (such as a <code>sessionID</code>) doesn't exist

Code	Status	Description
406	Not acceptable	The specified Accept header value isn't supported. Only application/x-ndjson and text/csv are accepted
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

11.4.9 Feed API examples

Basic session polling:

```
# Start a new session
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nod/?sessionID=myThreatIntel'
```

```
# Resume the session (returns data since last request)
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nod/?sessionID=myThreatIntel'
```

Time window filtering:

```
# Get data from a specific time range
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nod/?after=2025-01-06T10:00:00Z&before=2025-
↳ 01-06T11:00:00Z'
```

Domain filtering:

```
# Filter for specific domain patterns
curl -H 'X-API-Key: YOUR_API_KEY' \

↳ 'https://api.domaintools.com/v1/feed/nod/?domain=*.example.com&sessionID=myThreatIntel'
```

CSV format:

```
# Request CSV format with headers
curl -H 'Accept: text/csv' -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nod/?headers=1&sessionID=myThreatIntel'
```

```
# Request CSV format without headers
curl -H 'Accept: text/csv' -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nod/?sessionID=myThreatIntel'
```

Handling large result sets:

```
# If you receive HTTP 206, repeat the request to get the next batch
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nod/?sessionID=myThreatIntel'
```

```
# Repeat until you receive HTTP 200
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nod/?sessionID=myThreatIntel'
```

Delete a session:

```
# Clear the saved offset and start fresh
curl -X DELETE -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/nod/?sessionID=myThreatIntel'
```

11.5 Download API

The Download API provides access to historical NOD data through temporary AWS S3 file links. Use this API to retrieve archived data you may have missed or to backfill your systems with historical information. Files are organized by hour and available for 90 days.

11.5.1 Base URL

```
https://api.domaintools.com/v1/download/nod/
```

11.5.2 Download API parameters

11.5.2.1 limit

Type: Integer

Valid values: Positive integer

Description: Limits the number of files returned in the response, starting from the most recent. Use to control payload size or test specific cases.

Example: limit=10

Required: No

11.5.3 Download API response structure

The API returns a JSON response containing an array of downloadable files. Each file entry includes:

download_name (string): The feed identifier (nod)

files (array): List of downloadable file entries

Each file object contains:

- **name** (string): Path and filename of the downloadable file

- **last_modified** (string): Timestamp of last modification in ISO 8601 UTC format
- **etag** (string): ETag (hash) used to verify file identity and versioning
- **size** (integer): File size in bytes
- **url** (string): Temporary signed URL to download the file from AWS

File naming convention:

- Data file: `nod/{YYYY-MM-DD}/nod-YYYYMMDD}.{HH00}-{HH00}.json.gz`
- Checksum file: `nod/{YYYY-MM-DD}/nod-YYYYMMDD}.{HH00}-{HH00}.json.gz.sha256`

Example response:

```
{
  "response": {
    "download_name": "nod",
    "files": [
      {
        "name": "nod/2024-11-19/nod-20241119.1900-2000.json.gz.sha256",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 64,
        "url": "https://download.example.com/nod/2024-11-19/nod-20241119.1900-
↳ 2000.json.gz.sha256?Expires=..."
      },
      {
        "name": "nod/2024-11-19/nod-20241119.1900-2000.json.gz",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 140725,
        "url": "https://download.example.com/nod/2024-11-19/nod-20241119.1900-
↳ 2000.json.gz?Expires=..."
      }
    ]
  }
}
```

11.5.4 Download API response codes

Code	Status	Description
200	OK	The request was successful
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

11.5.5 Download API file contents

The `*.json.gz.sha256` file is a checksum containing a SHA-256 hash value used to verify the integrity of the downloaded file.

The *.json.gz file, when uncompressed, contains JSON data in the same format as the Feed API response (NDJSON with timestamp and domain fields).

11.5.6 Download API examples

List available files:

```
# Get the most recent files
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/nod/?limit=10'
```

Download and verify a file:

```
# Get the file list
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/nod/?limit=2' > files.json

# Extract the URL and download the data file
curl -o nod-data.json.gz "$(jq -r '.response.files[1].url' files.json)"

# Download the checksum file
curl -o nod-data.json.gz.sha256 "$(jq -r '.response.files[0].url' files.json)"

# Verify the integrity
sha256sum -c nod-data.json.gz.sha256
```

Batch processing:

```
# Download multiple files in a loop
for url in $(curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/nod/?limit=24' | \
  jq -r '.response.files[] .url' | grep '\.json\.gz$'); do
  curl -O "$url"
done
```

11.6 RPZ Access

The NOD feed is available via Response Policy Zone (RPZ) for direct integration with DNS firewalls.

11.6.1 Overview

Response Policy Zone (RPZ) provides DNS-level blocking by integrating threat feeds directly into your DNS resolver. This allows you to automatically block or redirect DNS queries for domains in the feed, providing real-time protection before connections are established.

How RPZ works:

When a user attempts to access a domain in the RPZ feed, your DNS resolver responds with an NXDOMAIN (“no such domain”) status code, effectively making the domain unavailable. This blocks malicious domains at the DNS layer, preventing endpoint communication before any connection is established.

Benefits:

- **Real-time protection:** Blocks threats at the DNS resolver level
- **Automatic updates:** Receives updates via DNS zone transfers (AXFR/IXFR)
- **No client configuration:** Works transparently for all clients using your DNS resolver
- **Efficient:** Minimal performance impact on DNS resolution
- **Standard protocol:** Uses DNS Response Policy Zones specification

Zone naming format:

RPZ zones are named using the pattern: `[interval].[feed].rpz.domaintools.com`

Available time intervals: 5m, 10m, 30m, 1h, 3h, 12h (and 24h for some feeds)

Larger time intervals are supersets that include smaller intervals. Smaller intervals have smaller zone sizes and may be available faster.

11.6.2 Available zones

The NOD feed is available in the following RPZ zones:

- `5m.nod.rpz.domaintools.com` - Last 5 minutes
- `10m.nod.rpz.domaintools.com` - Last 10 minutes
- `30m.nod.rpz.domaintools.com` - Last 30 minutes
- `1h.nod.rpz.domaintools.com` - Last 1 hour
- `3h.nod.rpz.domaintools.com` - Last 3 hours
- `12h.nod.rpz.domaintools.com` - Last 12 hours
- `24h.nod.rpz.domaintools.com` - Last 24 hours

11.6.3 Configuration

Configuration requirements:

To access RPZ feeds, you need to:

1. **Provide IP addresses:** Contact enterprisesupport@domaintools.com with:
 - IP address(es) from which you connect to the RPZ provider DNS server
 - IP address(es) to receive DNS NOTIFY messages (typically the same)
2. **Configure firewall:** Add rules to allow DomainTools hosts to send UDP packets to port 53:
 - IPv4: 104.244.13.88 Port: 53
 - IPv4: 104.244.14.88 Port: 53
3. **Set up TSIG authentication:** DomainTools uses TSIG (Secret Key Transaction Authentication) for authorization:
 - TSIG key algorithm: `hmac-sha512`
 - TSIG key and key name: Provided by DomainTools Enterprise Support

Delivery method:

RPZ feeds are delivered via:

- Incremental Zone Transfers (IXFR)
- Full zone transfers (AXFR)
- DNS NOTIFY messages to trigger zone updates

For detailed configuration instructions, including DNS resolver setup, advanced features (allowlists, walled gardens, logging), and troubleshooting, see the [Response Policy Zone documentation](#).

11.6.4 Testing

Testing your RPZ configuration:

When your DNS resolver blocks a domain using RPZ, the response includes an SOA (Start of Authority) record that identifies which RPZ feed was used.

Test domain:

Each RPZ feed includes a test domain entry: `test.rpz.domaintools.test`

Use this to verify the RPZ feed is loaded and working. A successful test returns:

- NXDOMAIN status code
- SOA record in the ADDITIONAL section showing the specific feed name

Example SOA record:

```
;; ADDITIONAL SECTION:  
3h.nad.rpz.domaintools.com. 86400 IN SOA rpz-ns1.domaintools.com.  
↪ noc.domaintools.com. 946684799 600 300 86400 86400
```

The SOA SERIAL number (Unix epoch timestamp) indicates when the feed was last regenerated.

Troubleshooting:

If the SOA record appears in the AUTHORITY section instead of ADDITIONAL, or doesn't show the specific feed name, the response didn't come from RPZ. Check your DNS resolver's RPZ-related logs for additional debugging information.

11.6.5 RPZ examples

Test the RPZ feed:

```
# Query the test domain  
dig test.rpz.domaintools.test  
  
# Expected response includes NXDOMAIN and SOA record showing the feed name  
# Example SOA in ADDITIONAL section:  
# 3h.nod.rpz.domaintools.com. 86400 IN SOA rpz-ns1.domaintools.com. ...
```

Verify a specific zone is loaded:

```
# Test with the 24-hour NOD zone  
dig @your-dns-server test.rpz.domaintools.test  
  
# Check the SOA record shows: 24h.nod.rpz.domaintools.com
```

11.7 Related resources

- [Domain Risk Score user guide](#)
- [Newly Observed Hostnames feed](#)
- [Newly Active Domains feed](#)
- [Response Policy Zone documentation](#)

12 Newly Observed Hostnames (NOH)

The NOH feed offers a more granular view by listing fully qualified domain names (FQDNs) the first time they are observed on our global passive DNS sensor network. This is crucial for detecting threats like phishing and domain shadowing that often use unique subdomains on legitimate-looking domains to evade detection. The NOH feed typically has a much higher volume than the NOD feed, identifying about ninety new hostnames per second.

12.1 Overview

This feed tracks fully qualified domain names (FQDNs) (for example, `www.example.com` or `mail.subdomain.example.com`) as they're observed.

Use this feed when you need to:

- Detect phishing attempts that utilize unique subdomains on legitimate-looking domains
- Identify domain shadowing tactics where attackers create malicious subdomains on compromised legitimate domains
- Enhance real-time threat detection with high-volume hostname data
- Monitor subdomain creation patterns for security analysis

Inclusion criteria: Fully qualified domain names (FQDNs) observed in passive DNS for the first time.

12.2 Requirements

You need the following to access Threat Feeds:

- An Enterprise Account with DomainTools, accessible at <https://account.domaintools.com/my-account/>
- Authentication credentials (API key for header authentication, or API username and key for HMAC or open key authentication)
- A way to interact with a REST API delivered through AWS

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

For assistance, contact enterprisesupport@domaintools.com.

12.3 Authentication

You can authenticate to the NOH APIs using three different methods. Choose the method that best fits your security requirements and technical environment.

12.3.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

Required header:

X-Api-Key: YOUR_API_KEY

Examples:

```
# Feed API request
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/noh/?sessionID=myPhishDetector'
```

```
# Download API request
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/noh/'
```

12.3.2 HMAC authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret, providing integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials shouldn't be stored in plain text or included directly in request URLs.

DomainTools supports MD5, SHA1, and SHA256 for the hashing algorithm.

Required query parameters:

- `api_username`: Your DomainTools API username
- `signature`: HMAC-SHA1 signature of `api_username + timestamp + uri_path`
- `timestamp`: Current UTC timestamp in ISO 8601 format (for example, `2025-06-01T15:30:00Z`)

Constructing the HMAC signature:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Important: URI path must include API version

The `uri_path` parameter must include the API version prefix. For example, use `/v1/feed/nod/` not `/feed/nod/`.

Example Python signing function:

```
import hmac
import hashlib

def sign(api_username, api_key, timestamp, uri):
    params = f"{api_username}{timestamp}{uri}"
    return hmac.new(api_key.encode("utf-8"), params.encode("utf-8"),
        ↪ hashlib.sha1).hexdigest()
```

Note: HMAC timestamp requirements

The `timestamp` parameter in HMAC authentication must be current (within a few minutes of the server time). The timestamps shown in these examples are static for demonstration purposes. In production, generate a fresh timestamp for each request using your system's current time in ISO 8601 UTC format (e.g., `2025-01-06T15:30:00Z`).

Examples:

```
# Feed API request with HMAC
curl 'https://api.domaintools.com/v1/feed/noh/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-
↳ 06T15:30:00Z&sessionID=myPhishDetector'
```

```
# Download API request with HMAC
curl 'https://api.domaintools.com/v1/download/noh/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

12.3.3 Open key authentication

This is the easiest authentication scheme to implement, but also the least secure. Each request contains the full API key and API username as query parameters. We recommend using [API key header authentication](#) or [HMAC authentication](#) instead.

If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

Required query parameters:

- `api_username`: Your API username
- `api_key`: Your API key

Examples:

```
# Feed API request
curl 'https://api.domaintools.com/v1/feed/noh/?api_username=YOUR_USERNAME&api_
↳ key=YOUR_API_KEY&sessionID=myPhishDetector'
```

```
# Download API request
curl 'https://api.domaintools.com/v1/download/noh/?api_username=YOUR_USERNAME&api_
↳ key=YOUR_API_KEY'
```

12.4 Real-time Feed API

The Feed API provides real-time access to current NOH data. Use this API to poll for the latest feed updates at regular intervals, maintain a session to track your position in the feed, and filter results based on your specific needs. Due to the high volume of this feed (~90 hostnames per second), consider using appropriate filtering and polling strategies.

12.4.1 Base URL

```
https://api.domaintools.com/v1/feed/noh/
```

12.4.2 Rate limits

Real-time feeds have the following rate limits:

- 2 queries per minute
- 120 queries per hour

If you exceed these limits, the API returns an error.

12.4.3 Response formats

The API supports two response formats:

NDJSON (Newline-Delimited JSON)

- Default format when no Accept header is specified
- Also known as JSON Lines (JSONL)
- One JSON object per line
- Efficient for streaming and processing large datasets
- Set Accept: `application/x-ndjson` to explicitly request this format

CSV (Comma-Separated Values)

- Set Accept: `text/csv` to request CSV format
- Add `&headers=1` to the query parameters to include column headers as the first line
- Not available for all feeds; check the specific feed documentation for CSV support

12.4.4 Session management

Session management allows you to maintain your position in the feed data stream, ensuring you don't miss or duplicate events when polling the API.

How sessions work:

- **Start a new session:** Provide a unique `sessionID` parameter of your choosing. By default, the API returns the past hour of results.
- **Resume a session:** Use the same `sessionID` in subsequent requests. The API returns all data since your last request.
- **Handle large result sets:** If a single request exceeds 10M results, the API returns an HTTP 206 response code. Repeat the same request with the same `sessionID` to receive the next batch of data until you receive an HTTP 200 response code.
- **One request at a time:** Do not send simultaneous requests with the same `sessionID` for the same feed. Wait for each request to complete before sending the next one. Concurrent requests with the same `sessionID` can produce errors or incomplete results.
- **Delete a session:** Use an HTTP DELETE request with your `sessionID` to clear the saved offset and start fresh.

Session ID requirements:

- 1 to 64 characters in length
- Alphanumeric characters and hyphens only (`[a-zA-Z0-9-]+`)
- Case-sensitive

12.4.5 Quick start

The standard access pattern is to periodically request the most recent feed data, as often as every 60 seconds.

```
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/noh/?sessionID=myPhishDetector'
```

This starts a new session and returns the last hour of data. Subsequent calls with the same `sessionID` return data since the last request.

12.4.6 Feed API parameters

12.4.6.1 sessionID

Type: String

Valid values: 1-64 alphanumeric characters and hyphens ([a-zA-Z0-9-]+)

Description: A unique identifier for the session, used for resuming data retrieval from the last point. Use a new `sessionID` to begin a new session, fetching the most recent hour by default. Reuse the same `sessionID` to return all feed data since your last request. If omitted, time window parameters (such as `after/before`) are required.

Example: `sessionID=mySOC`

Required: Yes, to continue where you left off (or use `after/before` instead)

12.4.6.2 after

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The start of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: `after=-60` or `after=2024-10-16T10:20:00Z`

Required: Yes, if `before` or `sessionID` not provided

12.4.6.3 before

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The end of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: before=-120 or before=2024-10-16T10:20:00Z

Required: Yes, if after or sessionID not provided

12.4.6.4 domain

Type: String

Valid values: Domain character set restricted by the DNS specification (letters, digits, hyphens). International characters should be specified in punycode. A trailing dot is acceptable.

Description: Filter for an exact domain or a domain substring by prefixing or suffixing your string with *. Multiple parameters are supported (for example, ?domain=*apple*&domain=*microsoft*). The URL-encoded version of * (%2A) may be required in some clients.

Example: domain=*bank* or domain=example.com

Required: No

12.4.6.5 fromBeginning

Type: Boolean

Valid values: true

Description: Requires a sessionID. When used with a new session ID, returns the first hour of data in the time window (rather than the last). Returns an error if the session ID already exists — drop fromBeginning from subsequent requests after the first call. Only the value true is accepted; any other value (including false) is ignored.

Example: fromBeginning=true

Required: No

12.4.6.6 top

Type: Integer

Valid values: Positive integer, 1-1,000,000,000

Description: Limits the number of results in the response payload. Primarily intended for testing. When you apply this parameter to risk feeds, results are sorted by overall_risk (descending).

Example: top=10

Required: No

12.4.6.7 headers

Type: Integer

Valid values: 1

Description: Adds a header row as the first line of the response when text/csv is requested. Set headers=1 to enable. Only applies when requesting CSV format. Only the value 1 is accepted; any other value is invalid.

Example: headers=1

Required: No

12.4.7 Feed API response structure

The API returns responses in NDJSON (Newline-Delimited JSON), with each response containing one domain entry per line. Each entry contains a `timestamp` in ISO 8601 UTC format, and the `domain`.

Response fields:

timestamp (string): The observation timestamp in ISO 8601 UTC format.

- Example: `"timestamp": "2024-11-15T16:14:39Z"`

domain (string): The domain name without the trailing dot. Domain character set restricted by the DNS specification (letters, digits, hyphens).

- Example: `"domain": "example.com"`

Example NDJSON response:

```
{"timestamp": "2024-11-15T16:14:39Z", "domain": "domiantools.com"}
{"timestamp": "2024-11-15T16:14:38Z", "domain": "domsintools.com"}
{"timestamp": "2024-11-15T16:14:36Z", "domain": "edomaintools.com"}
{"timestamp": "2024-11-15T16:14:35Z", "domain": "omaintools.com"}
{"timestamp": "2024-11-15T16:14:35Z", "domain": "v-domaintools.com"}
```

Example CSV response:

```
timestamp, domain
2024-11-15T16:14:39Z, domiantools.com
2024-11-15T16:14:38Z, domsintools.com
2024-11-15T16:14:36Z, edomaintools.com
```

12.4.8 Feed API response codes

Code	Status	Description
200	OK	The request was successful and all data has been delivered
206	Partial content	The request was successful, but only a portion of the data was returned. The request exceeded 10M results or the 1-hour evaluation window. Repeat the same request with the same <code>sessionID</code> to receive the next batch of data until you receive an HTTP 200 response
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
404	Not found	The requested resource (such as a <code>sessionID</code>) doesn't exist

Code	Status	Description
406	Not acceptable	The specified Accept header value isn't supported. Only application/x-ndjson and text/csv are accepted
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

12.4.9 Feed API examples

Basic session polling:

```
# Start a new session
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/noh/?sessionID=myPhishDetector'
```

```
# Resume the session (returns data since last request)
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/noh/?sessionID=myPhishDetector'
```

Time window filtering:

```
# Get data from a specific time range
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/noh/?after=2025-01-06T10:00:00Z&before=2025-
↳ 01-06T11:00:00Z'
```

Domain filtering:

```
# Filter for specific hostname patterns (useful for monitoring specific domains)
curl -H 'X-API-Key: YOUR_API_KEY' \

↳ 'https://api.domaintools.com/v1/feed/noh/?domain=*.example.com&sessionID=myPhishDetector'
```

CSV format:

```
# Request CSV format with headers
curl -H 'Accept: text/csv' -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/noh/?headers=1&sessionID=myPhishDetector'
```

```
# Request CSV format without headers
curl -H 'Accept: text/csv' -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/noh/?sessionID=myPhishDetector'
```

Handling large result sets:

```
# If you receive HTTP 206, repeat the request to get the next batch
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/noh/?sessionID=myPhishDetector'
```

```
# Repeat until you receive HTTP 200
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/noh/?sessionID=myPhishDetector'
```

Delete a session:

```
# Clear the saved offset and start fresh
curl -X DELETE -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/noh/?sessionID=myPhishDetector'
```

12.5 Download API

The Download API provides access to historical NOH data through temporary AWS S3 file links. Use this API to retrieve archived data you may have missed or to backfill your systems with historical information. Files are organized by hour and available for 90 days.

12.5.1 Base URL

```
https://api.domaintools.com/v1/download/noh/
```

12.5.2 Download API parameters

12.5.2.1 limit

Type: Integer

Valid values: Positive integer

Description: Limits the number of files returned in the response, starting from the most recent. Use to control payload size or test specific cases.

Example: limit=10

Required: No

12.5.3 Download API response structure

The API returns a JSON response containing an array of downloadable files. Each file entry includes:

download_name (string): The feed identifier (noh)

files (array): List of downloadable file entries

Each file object contains:

- **name** (string): Path and filename of the downloadable file

- **last_modified** (string): Timestamp of last modification in ISO 8601 UTC format
- **etag** (string): ETag (hash) used to verify file identity and versioning
- **size** (integer): File size in bytes
- **url** (string): Temporary signed URL to download the file from AWS

File naming convention:

- Data file: noh/{YYYY-MM-DD}/noh-{YYYYMMDD}.{HH00}-{HH00}.json.gz
- Checksum file: noh/{YYYY-MM-DD}/noh-{YYYYMMDD}.{HH00}-{HH00}.json.gz.sha256

Example response:

```
{
  "response": {
    "download_name": "noh",
    "files": [
      {
        "name": "noh/2024-11-19/noh-20241119.1900-2000.json.gz.sha256",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 64,
        "url": "https://download.example.com/noh/2024-11-19/noh-20241119.1900-
↵ 2000.json.gz.sha256?Expires=..."
      },
      {
        "name": "noh/2024-11-19/noh-20241119.1900-2000.json.gz",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 324850,
        "url": "https://download.example.com/noh/2024-11-19/noh-20241119.1900-
↵ 2000.json.gz?Expires=..."
      }
    ]
  }
}
```

12.5.4 Download API response codes

Code	Status	Description
200	OK	The request was successful
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

12.5.5 Download API file contents

The *.json.gz.sha256 file is a checksum containing a SHA-256 hash value used to verify the integrity of the downloaded file.

The *.json.gz file, when uncompressed, contains JSON data in the same format as the Feed API response (NDJSON with timestamp and domain fields, where domain contains the full hostname).

12.5.6 Download API examples

List available files:

```
# Get the most recent files
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/noh/?limit=10'
```

Download and verify a file:

```
# Get the file list
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/noh/?limit=2' > files.json

# Extract the URL and download the data file
curl -o noh-data.json.gz "$(jq -r '.response.files[1].url' files.json)"

# Download the checksum file
curl -o noh-data.json.gz.sha256 "$(jq -r '.response.files[0].url' files.json)"

# Verify the integrity
sha256sum -c noh-data.json.gz.sha256
```

Batch processing:

```
# Download multiple files in a loop
for url in $(curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/noh/?limit=24' | \
  jq -r '.response.files[] .url' | grep '\.json\.gz$'); do
  curl -O "$url"
done
```

12.6 Related resources

- [Newly Observed Domains feed](#)
- [Newly Active Domains feed](#)
- [Domain Risk Score user guide](#)
- [Response Policy Zone documentation](#)

13 Domain Discovery

The Domain Discovery feed is the largest feed of its kind, containing a daily list of all newly registered and newly observed domains from all TLDs, including those that do not publish zone files. On average, this feed contains nearly 350,000 new domains each day, making it a comprehensive source for any workflow that needs to track the creation of new domains.

13.1 Overview

Domains are apex-level (for example, `example.com` but not `www.example.com`), and the feed provides comprehensive coverage of the global domain landscape.

Use this feed when you need to:

- Comprehensive domain monitoring for a wide range of security and intelligence applications
- Track the global landscape of new domain registrations and observations
- Identify emerging threats and trends related to new domain creation
- Build comprehensive domain databases for research and analysis

Inclusion criteria: All newly registered and newly observed domains from all TLDs, including TLDs that do not publish zone files.

13.2 Requirements

You need the following to access Threat Feeds:

- An Enterprise Account with DomainTools, accessible at <https://account.domaintools.com/my-account/>
- Authentication credentials (API key for header authentication, or API username and key for HMAC or open key authentication)
- A way to interact with a REST API delivered through AWS

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

For assistance, contact enterprisesupport@domaintools.com.

13.3 Authentication

You can authenticate to the Domain Discovery APIs using three different methods. Choose the method that best fits your security requirements and technical environment.

13.3.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

Required header:

```
X-Api-Key: YOUR_API_KEY
```

Examples:

```
# Feed API request
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domaindiscovery/?sessionID=myDomainMonitor'
```

```
# Download API request
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domaindiscovery/'
```

13.3.2 HMAC authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret, providing integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials shouldn't be stored in plain text or included directly in request URLs.

DomainTools supports MD5, SHA1, and SHA256 for the hashing algorithm.

Required query parameters:

- `api_username`: Your DomainTools API username
- `signature`: HMAC-SHA1 signature of `api_username` + `timestamp` + `uri_path`
- `timestamp`: Current UTC timestamp in ISO 8601 format (for example, `2025-06-01T15:30:00Z`)

Constructing the HMAC signature:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Important: URI path must include API version

The `uri_path` parameter must include the API version prefix. For example, use `/v1/feed/nod/` not `/feed/nod/`.

Example Python signing function:

```
import hmac
import hashlib

def sign(api_username, api_key, timestamp, uri):
    params = f"{api_username}{timestamp}{uri}"
    return hmac.new(api_key.encode("utf-8"), params.encode("utf-8"),
        ↪ hashlib.sha1).hexdigest()
```

Note: HMAC timestamp requirements

The `timestamp` parameter in HMAC authentication must be current (within a few minutes of the server time). The timestamps shown in these examples are static for demonstration purposes. In production, generate a fresh timestamp for each request using your system's current time in ISO 8601 UTC format (e.g., `2025-01-06T15:30:00Z`).

Examples:

```
# Feed API request with HMAC
curl 'https://api.domaintools.com/v1/feed/domaindiscovery/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-
↳ 06T15:30:00Z&sessionID=myDomainMonitor'
```

```
# Download API request with HMAC
curl 'https://api.domaintools.com/v1/download/domaindiscovery/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

13.3.3 Open key authentication

This is the easiest authentication scheme to implement, but also the least secure. Each request contains the full API key and API username as query parameters. We recommend using [API key header authentication](#) or [HMAC authentication](#) instead.

If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

Required query parameters:

- `api_username`: Your API username
- `api_key`: Your API key

Examples:

```
# Feed API request
curl 'https://api.domaintools.com/v1/feed/domaindiscovery/?api_username=YOUR_
↳ USERNAME&api_key=YOUR_API_KEY&sessionID=myDomainMonitor'
```

```
# Download API request
curl 'https://api.domaintools.com/v1/download/domaindiscovery/?api_username=YOUR_
↳ USERNAME&api_key=YOUR_API_KEY'
```

13.4 Real-time Feed API

The Feed API provides real-time access to current Domain Discovery data. Use this API to poll for the latest feed updates at regular intervals, maintain a session to track your position in the feed, and filter results based on your specific needs. Due to the high volume of this feed (~350,000 domains per day), consider using appropriate filtering and polling strategies.

13.4.1 Base URL

```
https://api.domaintools.com/v1/feed/domaindiscovery/
```

13.4.2 Rate limits

Real-time feeds have the following rate limits:

- 2 queries per minute
- 120 queries per hour

If you exceed these limits, the API returns an error.

13.4.3 Response formats

The API supports two response formats:

NDJSON (Newline-Delimited JSON)

- Default format when no Accept header is specified
- Also known as JSON Lines (JSONL)
- One JSON object per line
- Efficient for streaming and processing large datasets
- Set Accept: `application/x-ndjson` to explicitly request this format

CSV (Comma-Separated Values)

- Set Accept: `text/csv` to request CSV format
- Add `&headers=1` to the query parameters to include column headers as the first line
- Not available for all feeds; check the specific feed documentation for CSV support

13.4.4 Session management

Session management allows you to maintain your position in the feed data stream, ensuring you don't miss or duplicate events when polling the API.

How sessions work:

- **Start a new session:** Provide a unique `sessionID` parameter of your choosing. By default, the API returns the past hour of results.
- **Resume a session:** Use the same `sessionID` in subsequent requests. The API returns all data since your last request.
- **Handle large result sets:** If a single request exceeds 10M results, the API returns an HTTP 206 response code. Repeat the same request with the same `sessionID` to receive the next batch of data until you receive an HTTP 200 response code.
- **One request at a time:** Do not send simultaneous requests with the same `sessionID` for the same feed. Wait for each request to complete before sending the next one. Concurrent requests with the same `sessionID` can produce errors or incomplete results.
- **Delete a session:** Use an HTTP DELETE request with your `sessionID` to clear the saved offset and start fresh.

Session ID requirements:

- 1 to 64 characters in length
- Alphanumeric characters and hyphens only (`[a-zA-Z0-9-]+`)
- Case-sensitive

13.4.5 Quick start

The standard access pattern is to periodically request the most recent feed data, as often as every 60 seconds.

```
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domaindiscovery/?sessionID=myDomainMonitor'
```

This starts a new session and returns the last hour of data. Subsequent calls with the same `sessionID` return data since the last request.

13.4.6 Feed API parameters

13.4.6.1 sessionID

Type: String

Valid values: 1-64 alphanumeric characters and hyphens ([a-zA-Z0-9-]+)

Description: A unique identifier for the session, used for resuming data retrieval from the last point. Use a new `sessionID` to begin a new session, fetching the most recent hour by default. Reuse the same `sessionID` to return all feed data since your last request. If omitted, time window parameters (such as `after`/`before`) are required.

Example: `sessionID=mySOC`

Required: Yes, to continue where you left off (or use `after`/`before` instead)

13.4.6.2 after

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The start of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: `after=-60` or `after=2024-10-16T10:20:00Z`

Required: Yes, if `before` or `sessionID` not provided

13.4.6.3 before

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The end of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: before=-120 or before=2024-10-16T10:20:00Z

Required: Yes, if after or sessionID not provided

13.4.6.4 domain

Type: String

Valid values: Domain character set restricted by the DNS specification (letters, digits, hyphens). International characters should be specified in punycode. A trailing dot is acceptable.

Description: Filter for an exact domain or a domain substring by prefixing or suffixing your string with *. Multiple parameters are supported (for example, ?domain=*apple*&domain=*microsoft*). The URL-encoded version of * (%2A) may be required in some clients.

Example: domain=*bank* or domain=example.com

Required: No

13.4.6.5 fromBeginning

Type: Boolean

Valid values: true

Description: Requires a sessionID. When used with a new session ID, returns the first hour of data in the time window (rather than the last). Returns an error if the session ID already exists — drop fromBeginning from subsequent requests after the first call. Only the value true is accepted; any other value (including false) is ignored.

Example: fromBeginning=true

Required: No

13.4.6.6 top

Type: Integer

Valid values: Positive integer, 1-1,000,000,000

Description: Limits the number of results in the response payload. Primarily intended for testing. When you apply this parameter to risk feeds, results are sorted by overall_risk (descending).

Example: top=10

Required: No

13.4.6.7 headers

Type: Integer

Valid values: 1

Description: Adds a header row as the first line of the response when text/csv is requested. Set headers=1 to enable. Only applies when requesting CSV format. Only the value 1 is accepted; any other value is invalid.

Example: headers=1

Required: No

13.4.7 Feed API response structure

The API returns responses in NDJSON (Newline-Delimited JSON), with each response containing one domain entry per line. Each entry contains a `timestamp` in ISO 8601 UTC format, and the `domain`.

Response fields:

timestamp (string): The observation timestamp in ISO 8601 UTC format.

- Example: `"timestamp":"2024-11-15T16:14:39Z"`

domain (string): The domain name without the trailing dot. Domain character set restricted by the DNS specification (letters, digits, hyphens).

- Example: `"domain":"example.com"`

Example NDJSON response:

```
{ "timestamp": "2024-11-15T16:14:39Z", "domain": "domiantools.com" }
{ "timestamp": "2024-11-15T16:14:38Z", "domain": "domsintools.com" }
{ "timestamp": "2024-11-15T16:14:36Z", "domain": "edomaintools.com" }
{ "timestamp": "2024-11-15T16:14:35Z", "domain": "omaintools.com" }
{ "timestamp": "2024-11-15T16:14:35Z", "domain": "v-domaintools.com" }
```

Example CSV response:

```
timestamp, domain
2024-11-15T16:14:39Z, domiantools.com
2024-11-15T16:14:38Z, domsintools.com
2024-11-15T16:14:36Z, edomaintools.com
```

13.4.8 Feed API response codes

Code	Status	Description
200	OK	The request was successful and all data has been delivered
206	Partial content	The request was successful, but only a portion of the data was returned. The request exceeded 10M results or the 1-hour evaluation window. Repeat the same request with the same <code>sessionID</code> to receive the next batch of data until you receive an HTTP 200 response
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
404	Not found	The requested resource (such as a <code>sessionID</code>) doesn't exist

Code	Status	Description
406	Not acceptable	The specified Accept header value isn't supported. Only application/x-ndjson and text/csv are accepted
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

13.4.9 Feed API examples

Basic session polling:

```
# Start a new session  
curl -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domaindiscovery/?sessionID=myDomainMonitor'
```

```
# Resume the session (returns data since last request)  
curl -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domaindiscovery/?sessionID=myDomainMonitor'
```

Time window filtering:

```
# Get data from a specific time range  
curl -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domaindiscovery/?after=2025-01-  
↳ 06T10:00:00Z&before=2025-01-06T11:00:00Z'
```

Domain filtering:

```
# Filter for specific domain patterns (useful for monitoring specific TLDs or  
↳ patterns)  
curl -H 'X-API-Key: YOUR_API_KEY' \  
  
↳ 'https://api.domaintools.com/v1/feed/domaindiscovery/?domain=*.io&sessionID=myDomainMonitor'
```

CSV format:

```
# Request CSV format with headers  
curl -H 'Accept: text/csv' -H 'X-API-Key: YOUR_API_KEY' \  
  
↳ 'https://api.domaintools.com/v1/feed/domaindiscovery/?headers=1&sessionID=myDomainMonitor'
```

```
# Request CSV format without headers  
curl -H 'Accept: text/csv' -H 'X-API-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/feed/domaindiscovery/?sessionID=myDomainMonitor'
```

Handling large result sets:

```
# If you receive HTTP 206, repeat the request to get the next batch
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domaindiscovery/?sessionID=myDomainMonitor'
```

```
# Repeat until you receive HTTP 200
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domaindiscovery/?sessionID=myDomainMonitor'
```

Delete a session:

```
# Clear the saved offset and start fresh
curl -X DELETE -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domaindiscovery/?sessionID=myDomainMonitor'
```

13.5 Real-time Download API

The Real-time Download API provides access to historical Domain Discovery data through temporary AWS S3 file links. Use this API to retrieve archived data you may have missed or to backfill your systems with historical information. Files are organized by hour and available for 90 days.

13.5.1 Base URL

```
https://api.domaintools.com/v1/download/domaindiscovery/
```

13.5.2 Download API parameters

13.5.2.1 limit

Type: Integer

Valid values: Positive integer

Description: Limits the number of files returned in the response, starting from the most recent. Use to control payload size or test specific cases.

Example: limit=10

Required: No

13.5.3 Download API response structure

The API returns a JSON response containing an array of downloadable files. Each file entry includes:

download_name (string): The feed identifier (domaindiscovery)

files (array): List of downloadable file entries

Each file object contains:

- **name** (string): Path and filename of the downloadable file
- **last_modified** (string): Timestamp of last modification in ISO 8601 UTC format
- **etag** (string): ETag (hash) used to verify file identity and versioning
- **size** (integer): File size in bytes
- **url** (string): Temporary signed URL to download the file from AWS

File naming convention:

- Data file:
domaindiscovery/{YYYY-MM-DD}/domaindiscovery-{YYYYMMDD}.{HH00}-{HH00}.json.gz
- Checksum file: domaindiscovery/{YYYY-MM-DD}/domaindiscovery-{YYYYMMDD}.{HH00}-{HH00}.json.gz.sha256

Example response:

```
{
  "response": {
    "download_name": "domaindiscovery",
    "files": [
      {
        "name":
↪ "domaindiscovery/2024-11-19/domaindiscovery-20241119.1900-2000.json.gz.sha256",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 64,
        "url": "https://download.example.com/domaindiscovery/2024-11-
↪ 19/domaindiscovery-20241119.1900-2000.json.gz.sha256?Expires=..."
      },
      {
        "name":
↪ "domaindiscovery/2024-11-19/domaindiscovery-20241119.1900-2000.json.gz",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 2450000,
        "url": "https://download.example.com/domaindiscovery/2024-11-
↪ 19/domaindiscovery-20241119.1900-2000.json.gz?Expires=..."
      }
    ]
  }
}
```

13.5.4 Download API response codes

Code	Status	Description
200	OK	The request was successful
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

13.5.5 Download API file contents

The *.json.gz.sha256 file is a checksum containing a SHA-256 hash value used to verify the integrity of the downloaded file.

The *.json.gz file, when uncompressed, contains JSON data in the same format as the Feed API response (NDJSON with timestamp and domain fields).

13.5.6 Download API examples

List available files:

```
# Get the most recent files
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domaindiscovery/?limit=10'
```

Download and verify a file:

```
# Get the file list
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domaindiscovery/?limit=2' > files.json

# Extract the URL and download the data file
curl -o domaindiscovery-data.json.gz "$(jq -r '.response.files[1].url' files.json)"

# Download the checksum file
curl -o domaindiscovery-data.json.gz.sha256 "$(jq -r '.response.files[0].url'
  ↵ files.json)"

# Verify the integrity
sha256sum -c domaindiscovery-data.json.gz.sha256
```

Batch processing:

```
# Download multiple files in a loop
for url in $(curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domaindiscovery/?limit=24' | \
  jq -r '.response.files[].url' | grep '\.json\.gz$'); do
  curl -O "$url"
done
```

13.6 Daily Download API

The Daily Download API provides daily batch summaries as an alternative to hourly real-time data. Use this when you need daily aggregated data rather than real-time updates.

13.6.1 Overview

Daily feed of newly registered and newly observed domains.

Inclusion threshold: Newly observed or registered

Format: Gzip-compressed CSV file; one domain name per line

Size: ~375,000 domains, ~2.5MB compressed

13.6.2 Base URL

```
https://api.domaintools.com/v1/download/daily_domain_discovery/
```

13.6.3 Daily Download parameters

The Daily Download API supports standard download parameters:

13.6.3.1 `api_username`

Type: string (required for HMAC and open key auth)

Your DomainTools API username

13.6.3.2 `api_key`

Type: string (required for open key auth)

Your DomainTools API key

13.6.3.3 `signature`

Type: string (required for HMAC auth)

HMAC signature of your request

13.6.3.4 `timestamp`

Type: string (required for HMAC auth)

Current timestamp for HMAC authentication in ISO 8601 format

13.6.3.5 `limit`

Type: integer (optional)

Limit the list of signed files. Ordering of files is always descending, so the latest files are first.

13.6.3.6 `page`

Type: integer (optional)

Select which page of results are returned. Pages begin at 0 with latest results.

13.6.3.7 `prefix`

Type: string (optional)

Filter results by date using the file prefix (format: NEW_DOMAINS_YYYYMMDD).

13.6.4 Daily Download response structure

The API returns a JSON response with signed URLs for downloadable files:

download_name (string): The feed identifier (`daily_domain_discovery`)

files (array): List of downloadable file entries

Each file object contains:

- **name** (string): File path (e.g., `domain_discovery/NEW_DOMAINS_20240704.csv.gz`)
- **last_modified** (string): Last modified date in ISO 8601 format
- **etag** (string): Entity tag (hash of the file)
- **size** (integer): Size in bytes
- **url** (string): Signed AWS download URL (valid for 12 hours)

13.6.5 Daily Download response codes

200: OK - The request was successful

400: Bad request

401: Unauthorized

403: Forbidden

404: No data to download

13.6.6 Daily Download file naming

Daily files follow this naming pattern:

```
domain_discovery/NEW_DOMAINS_YYYYMMDD.csv.gz
```

Example: `domain_discovery/NEW_DOMAINS_20240704.csv.gz`

13.6.7 File contents

The CSV file contains one domain name per line (no header, no timestamp field):

```
example.com
newdomain.net
another-domain.org
```

13.6.8 Daily Download examples

List available files:

```
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/daily_domain_discovery/'
```

Filter by date prefix:

```
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/daily_domain_discovery/?prefix=NEW_-
  ↵ DOMAINS_20240704'
```

Download a specific file:

```
# Get the file list
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/daily_domain_discovery/?limit=1' >
  ↵ files.json
```

```
# Download the file
curl -o daily-domains.csv.gz "$(jq -r '.response.files[0].url' files.json)"

# Decompress and view
gunzip daily-domains.csv.gz
head daily-domains.csv
```

13.7 Related resources

- [Newly Observed Domains feed](#)
- [Newly Active Domains feed](#)
- [Domain Risk Score user guide](#)

14 Parsed Domain RDAP

This feed provides parsed and normalized domain information extracted from raw RDAP records, including contact information, registrar details, name servers, and important dates. It's designed for efficient data searching, indexing, and automated processing in security workflows.

14.1 Overview

Domains are apex-level (for example, `example.com` but not `www.example.com`), and the feed provides structured, machine-readable RDAP data from both domain registries and registrars.

Use this feed when you need to:

- Search for, index, or cross-reference data from RDAP records
- Enable programmatic access to structured RDAP data
- Analyze domain registration data to identify patterns
- Track threat actors through registration information
- Monitor changes in registration data for brand protection
- Automate domain intelligence workflows

Inclusion criteria: Changes to global domain registration information, populated by the Registration Data Access Protocol (RDAP).

Note: This feed complements the 5-Minute WHOIS Feed as registries and registrars transition from WHOIS to RDAP.

14.2 Requirements

You need the following to access Threat Feeds:

- An Enterprise Account with DomainTools, accessible at <https://account.domaintools.com/my-account/>
- Authentication credentials (API key for header authentication, or API username and key for HMAC or open key authentication)
- A way to interact with a REST API delivered through AWS

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

For assistance, contact enterprisesupport@domaintools.com.

14.3 Authentication

You can authenticate to the Parsed Domain RDAP APIs using three different methods. Choose the method that best fits your security requirements and technical environment.

14.3.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

Required header:

X-Api-Key: YOUR_API_KEY

Examples:

```
# Feed API request
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrdap/?sessionID=myRDAPMonitor'
```

```
# Download API request
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domainrdap/'
```

14.3.2 HMAC authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret, providing integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials shouldn't be stored in plain text or included directly in request URLs.

DomainTools supports MD5, SHA1, and SHA256 for the hashing algorithm.

Required query parameters:

- `api_username`: Your DomainTools API username
- `signature`: HMAC-SHA1 signature of `api_username` + `timestamp` + `uri_path`
- `timestamp`: Current UTC timestamp in ISO 8601 format (for example, 2025-06-01T15:30:00Z)

Constructing the HMAC signature:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Important: URI path must include API version

The `uri_path` parameter must include the API version prefix. For example, use `/v1/feed/nod/` not `/feed/nod/`.

Example Python signing function:

```
import hmac
import hashlib

def sign(api_username, api_key, timestamp, uri):
    params = f"{api_username}{timestamp}{uri}"
    return hmac.new(api_key.encode("utf-8"), params.encode("utf-8"),
        ↪ hashlib.sha1).hexdigest()
```

Note: HMAC timestamp requirements

The timestamp parameter in HMAC authentication must be current (within a few minutes of the server time). The timestamps shown in these examples are static for demonstration purposes. In production, generate a fresh timestamp for each request using your system's current time in ISO 8601 UTC format (e.g., 2025-01-06T15:30:00Z).

Examples:

```
# Feed API request with HMAC
curl 'https://api.domaintools.com/v1/feed/domainrdap/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-
↳ 06T15:30:00Z&sessionID=myRDAPMonitor'
```

```
# Download API request with HMAC
curl 'https://api.domaintools.com/v1/download/domainrdap/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

14.3.3 Open key authentication

This is the easiest authentication scheme to implement, but also the least secure. Each request contains the full API key and API username as query parameters. We recommend using [API key header authentication](#) or [HMAC authentication](#) instead.

If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

Required query parameters:

- api_username: Your API username
- api_key: Your API key

Examples:

```
# Feed API request
curl 'https://api.domaintools.com/v1/feed/domainrdap/?api_username=YOUR_
↳ USERNAME&api_key=YOUR_API_KEY&sessionID=myRDAPMonitor'
```

```
# Download API request
curl 'https://api.domaintools.com/v1/download/domainrdap/?api_username=YOUR_
↳ USERNAME&api_key=YOUR_API_KEY'
```

14.4 Real-time Feed API

The Feed API provides real-time access to current Parsed Domain RDAP data. Use this API to poll for the latest feed updates at regular intervals, maintain a session to track your position in the feed, and filter results based on your specific needs.

Important: The Parsed Domain RDAP feed returns JSON format only and does not support CSV format.

14.4.1 Base URL

```
https://api.domaintools.com/v1/feed/domainrdap/
```

14.4.2 Rate limits

Real-time feeds have the following rate limits:

- 2 queries per minute
- 120 queries per hour

If you exceed these limits, the API returns an error.

14.4.3 Response formats

The Parsed Domain RDAP feed returns responses in JSON format only.

Accept: `application/json` (default): Returns JSON format. This is the only supported format for this feed.

Note: Unlike other feeds, the Parsed Domain RDAP feed does not support CSV format (`text/csv`).

14.4.4 Session management

Session management allows you to maintain your position in the feed data stream, ensuring you don't miss or duplicate events when polling the API.

How sessions work:

- **Start a new session:** Provide a unique `sessionId` parameter of your choosing. By default, the API returns the past hour of results.
- **Resume a session:** Use the same `sessionId` in subsequent requests. The API returns all data since your last request.
- **Handle large result sets:** If a single request exceeds 10M results, the API returns an HTTP 206 response code. Repeat the same request with the same `sessionId` to receive the next batch of data until you receive an HTTP 200 response code.
- **One request at a time:** Do not send simultaneous requests with the same `sessionId` for the same feed. Wait for each request to complete before sending the next one. Concurrent requests with the same `sessionId` can produce errors or incomplete results.
- **Delete a session:** Use an HTTP DELETE request with your `sessionId` to clear the saved offset and start fresh.

Session ID requirements:

- 1 to 64 characters in length
- Alphanumeric characters and hyphens only (`[a-zA-Z0-9-]+`)
- Case-sensitive

14.4.5 Quick start

The standard access pattern is to periodically request the most recent feed data, as often as every 60 seconds.

```
curl -H 'X-API-Key: YOUR_API_KEY' \
'https://api.domaintools.com/v1/feed/domainrdap/?sessionID=myRDAPMonitor'
```

This starts a new session and returns the last hour of data. Subsequent calls with the same sessionID return data since the last request.

14.4.6 Feed API parameters

14.4.6.1 sessionID

Type: String

Valid values: 1-64 alphanumeric characters and hyphens ([a-zA-Z0-9-]+)

Description: A unique identifier for the session, used for resuming data retrieval from the last point. Use a new sessionID to begin a new session, fetching the most recent hour by default. Reuse the same sessionID to return all feed data since your last request. If omitted, time window parameters (such as after/before) are required.

Example: sessionID=mySOC

Required: Yes, to continue where you left off (or use after/before instead)

14.4.6.2 after

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The start of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: after=-60 or after=2024-10-16T10:20:00Z

Required: Yes, if before or sessionID not provided

14.4.6.3 before

Type: Integer or string

Valid values:

- Integer: -1 to -432,000 (relative seconds before current time)
- String: ISO 8601 datetime in UTC format (YYYY-MM-DDTHH:MM:SSZ)

Description: The end of the query window (inclusive). When using an integer, the value is in seconds relative to the current time. When using a string, provide an absolute timestamp. The timestamp must represent a point between 1 second ago and 5 days ago, relative to the current UTC time.

Example: before=-120 or before=2024-10-16T10:20:00Z

Required: Yes, if after or sessionID not provided

14.4.6.4 domain

Type: String

Valid values: Domain character set restricted by the DNS specification (letters, digits, hyphens). International characters should be specified in punycode. A trailing dot is acceptable.

Description: Filter for an exact domain or a domain substring by prefixing or suffixing your string with *. Multiple parameters are supported (for example, `?domain=*apple*&domain=*microsoft*`). The URL-encoded version of * (%2A) may be required in some clients.

Example: `domain=*bank*` or `domain=example.com`

Required: No

14.4.6.5 fromBeginning

Type: Boolean

Valid values: `true`

Description: Requires a `sessionId`. When used with a new session ID, returns the first hour of data in the time window (rather than the last). Returns an error if the session ID already exists — drop `fromBeginning` from subsequent requests after the first call. Only the value `true` is accepted; any other value (including `false`) is ignored.

Example: `fromBeginning=true`

Required: No

14.4.6.6 top

Type: Integer

Valid values: Positive integer, 1-1,000,000,000

Description: Limits the number of results in the response payload. Primarily intended for testing. When you apply this parameter to risk feeds, results are sorted by `overall_risk` (descending).

Example: `top=10`

Required: No

14.4.7 Feed API response structure

The API returns JSON format (not NDJSON) with one record per response.

Note: The Parsed Domain RDAP feed does not support CSV format. Only JSON responses are available.

Response fields:

timestamp (string): ISO 8601 UTC timestamp when the domain was observed

domain (string): The apex-level domain name

raw_record (object): Contains the original raw RDAP responses from both registry and registrar

- **first_request_timestamp** (string): When the first RDAP request was made

- **requests** (array): Array of RDAP request/response pairs
 - **data** (string): JSON-encoded string of the raw RDAP record
 - **source_type** (string): Either “registry” or “registrar”
 - **timestamp** (string): When this request was made
 - **url** (string): The RDAP endpoint URL queried

parsed_record (object): Parsed and normalized domain information

- **parsed_fields** (object): Structured data extracted from RDAP records, may include:
 - **domain**: Domain name
 - **handle**: Registry/registrar handle
 - **creation_date**: When the domain was created
 - **expiration_date**: When the domain expires
 - **last_changed_date**: Last modification date
 - **dnssec**: DNSSEC status
 - **domain_statuses**: Array of domain status codes
 - **nameservers**: Array of nameserver hostnames
 - **contacts**: Contact information (registrant, admin, tech)
 - **registrar**: Registrar information
 - **links**: Related RDAP links
 - **conformance**: RDAP conformance levels
 - **email_domains**: Extracted email domains
 - **emails**: Contact email addresses
 - And more, depending on the registry
- **registrar_request_url** (string, nullable): URL of the registrar RDAP endpoint
- **registry_request_url** (string, nullable): URL of the registry RDAP endpoint

Example response:

```
{
  "timestamp": "2025-01-06T15:30:42Z",
  "domain": "example.com",
  "raw_record": {
    "first_request_timestamp": "2025-01-06T15:30:40Z",
    "requests": [
      {
        "data": "{\"rdapConformance\": [\"rdap_level_0\"], ...}\",
        "source_type": "registry",
        "timestamp": "2025-01-06T15:30:40Z",
        "url": "https://rdap.verisign.com/com/v1/domain/example.com"
      }
    ]
  },
  "parsed_record": {
    "parsed_fields": {
      "domain": "example.com",
      "creation_date": "1995-08-14T04:00:00Z",
      "expiration_date": "2026-08-13T04:00:00Z",
      "nameservers": ["a.iana-servers.net", "b.iana-servers.net"]
    },
    "registry_request_url": "https://rdap.verisign.com/com/v1/domain/example.com",
    "registrar_request_url": null
  }
}
```

```
}
```

14.4.8 Feed API response codes

Code	Status	Description
200	OK	The request was successful and all data has been delivered
206	Partial content	The request was successful, but only a portion of the data was returned. The request exceeded 10M results or the 1-hour evaluation window. Repeat the same request with the same <code>sessionID</code> to receive the next batch of data until you receive an HTTP 200 response
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
404	Not found	The requested resource (such as a <code>sessionID</code>) doesn't exist
406	Not acceptable	The specified <code>Accept</code> header value isn't supported. Only <code>application/x-ndjson</code> and <code>text/csv</code> are accepted
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

14.4.9 Feed API examples

Basic session polling:

```
# Start a new session
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrdap/?sessionID=myRDAPMonitor'
```

```
# Resume the session (returns data since last request)
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrdap/?sessionID=myRDAPMonitor'
```

Time window filtering:

```
# Get data from a specific time range
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrdap/?after=2025-01-
↳ 06T10:00:00Z&before=2025-01-06T11:00:00Z'
```

Domain filtering:

```
# Filter for specific domain patterns
curl -H 'X-API-Key: YOUR_API_KEY' \

↳ 'https://api.domaintools.com/v1/feed/domainrdap/?domain=*.example.com&sessionID=myRDAPMonitor'
```

Limit results for testing:

```
# Get only the first result for testing
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrdap/?top=1&sessionID=myRDAPMonitor'
```

Handling large result sets:

```
# If you receive HTTP 206, repeat the request to get the next batch
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrdap/?sessionID=myRDAPMonitor'
```

```
# Repeat until you receive HTTP 200
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrdap/?sessionID=myRDAPMonitor'
```

Delete a session:

```
# Clear the saved offset and start fresh
curl -X DELETE -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/feed/domainrdap/?sessionID=myRDAPMonitor'
```

14.5 Real-time Download API

The Real-time Download API provides access to historical Parsed Domain RDAP data through temporary AWS S3 file links. Use this API to retrieve archived data you may have missed or to backfill your systems with historical information. Files are organized by hour and available for 90 days.

14.5.1 Base URL

```
https://api.domaintools.com/v1/download/domainrdap/
```

14.5.2 Download API parameters

14.5.2.1 limit

Type: Integer

Valid values: Positive integer

Description: Limits the number of files returned in the response, starting from the most recent. Use to control payload size or test specific cases.

Example: limit=10

Required: No

14.5.3 Download API response structure

The API returns a JSON response containing an array of downloadable files. Each file entry includes:

download_name (string): The feed identifier (domainrdap)

files (array): List of downloadable file entries

Each file object contains:

- **name** (string): Path and filename of the downloadable file
- **last_modified** (string): Timestamp of last modification in ISO 8601 UTC format
- **etag** (string): ETag (hash) used to verify file identity and versioning
- **size** (integer): File size in bytes
- **url** (string): Temporary signed URL to download the file from AWS

File naming convention:

- Data file: domainrdap/{YYYY-MM-DD}/domainrdap-{YYYYMMDD}.{HH00}-{HH00}.json.gz
- Checksum file:
domainrdap/{YYYY-MM-DD}/domainrdap-{YYYYMMDD}.{HH00}-{HH00}.json.gz.sha256

Example response:

```
{
  "response": {
    "download_name": "domainrdap",
    "files": [
      {
        "name":
↪ "domainrdap/2024-11-19/domainrdap-20241119.1900-2000.json.gz.sha256",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 64,
        "url": "https://download.example.com/domainrdap/2024-11-19/domainrdap-
↪ 20241119.1900-2000.json.gz.sha256?Expires=..."
      },
      {
        "name": "domainrdap/2024-11-19/domainrdap-20241119.1900-2000.json.gz",
        "last_modified": "2024-11-19T20:00:11+00:00",
        "etag": "\"67a6d9b0973b2d31ffb779dc8f7f8cfa\"",
        "size": 1850000,
        "url": "https://download.example.com/domainrdap/2024-11-19/domainrdap-
↪ 20241119.1900-2000.json.gz?Expires=..."
      }
    ]
  }
}
```

14.5.4 Download API response codes

Code	Status	Description
200	OK	The request was successful
400	Bad request	The request is malformed
403	Forbidden	Missing or invalid API credentials
422	Unprocessable entity	The request is syntactically valid but violates semantic or domain-specific rules (for example, invalid query parameter values)

14.5.5 Download API file contents

The *.json.gz.sha256 file is a checksum containing a SHA-256 hash value used to verify the integrity of the downloaded file.

The *.json.gz file, when uncompressed, contains JSON data in the same format as the Feed API response (JSON with timestamp, domain, raw_record, and parsed_record fields).

14.5.6 Download API examples

List available files:

```
# Get the most recent files
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domainrdap/?limit=10'
```

Download and verify a file:

```
# Get the file list
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domainrdap/?limit=2' > files.json

# Extract the URL and download the data file
curl -o domainrdap-data.json.gz "$(jq -r '.response.files[1].url' files.json)"

# Download the checksum file
curl -o domainrdap-data.json.gz.sha256 "$(jq -r '.response.files[0].url'
↳ files.json)"

# Verify the integrity
sha256sum -c domainrdap-data.json.gz.sha256
```

Batch processing:

```
# Download multiple files in a loop
for url in $(curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/domainrdap/?limit=24' | \
  jq -r '.response.files[].url' | grep '\.json\.gz$'); do
  curl -O "$url"
done
```

14.6 Related resources

- [Domain Discovery feed](#)
- [Newly Observed Domains feed](#)

15 5-Minute Domain WHOIS

The 5-Minute Domain WHOIS feed provides the most recently updated domain WHOIS records, processed on a 5-minute basis. This feed is available in both raw (unparsed) and parsed formats, making it suitable for various integration and analysis workflows.

15.1 Overview

This feed captures all domain WHOIS records that have been updated since the previous 5-minute processing cycle. The feed is available in two versions:

- **Raw version** (`5_min_domain_whois`): Unparsed WHOIS data as received from registries
- **Parsed version** (`5_min_domain_whois_parsed`): Structured JSON format with normalized fields

Use this feed when you need to:

- Monitor domain registration changes in near real-time
- Track WHOIS record updates for threat intelligence
- Analyze domain ownership and contact information changes
- Build domain registration databases
- Detect suspicious registration patterns
- Automate domain intelligence workflows

Inclusion criteria: All domain names processed since the previous 5-minute update cycle.

Format: Gzip-compressed tab-separated (TSV) or JSON text files

Size: Up to 10MB per day

15.2 Requirements

You need the following to access Threat Feeds:

- An Enterprise Account with DomainTools, accessible at <https://account.domaintools.com/my-account/>
- Authentication credentials (API key for header authentication, or API username and key for HMAC or open key authentication)
- A way to interact with a REST API delivered through AWS

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

For assistance, contact enterprisesupport@domaintools.com.

15.3 Authentication

You can authenticate to the 5-Minute Domain WHOIS API using three different methods. Choose the method that best fits your security requirements and technical environment.

15.3.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

Required header:

X-Api-Key: YOUR_API_KEY

Examples:

```
# Raw WHOIS
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/5_min_domain_whois/'
```

```
# Parsed WHOIS
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/5_min_domain_whois_parsed/'
```

15.3.2 HMAC authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret, providing integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials shouldn't be stored in plain text or included directly in request URLs.

DomainTools supports MD5, SHA1, and SHA256 for the hashing algorithm.

Required query parameters:

- `api_username`: Your DomainTools API username
- `signature`: HMAC-SHA1 signature of `api_username + timestamp + uri_path`
- `timestamp`: Current UTC timestamp in ISO 8601 format (for example, `2025-06-01T15:30:00Z`)

Constructing the HMAC signature:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Important: URI path must include API version

The `uri_path` parameter must include the API version prefix. For example, use `/v1/feed/nod/` not `/feed/nod/`.

Example Python signing function:

```
import hmac
import hashlib

def sign(api_username, api_key, timestamp, uri):
    params = f"{api_username}{timestamp}{uri}"
    return hmac.new(api_key.encode("utf-8"), params.encode("utf-8"),
        ↪ hashlib.sha1).hexdigest()
```

Note: HMAC timestamp requirements

The timestamp parameter in HMAC authentication must be current (within a few minutes of the server time). The timestamps shown in these examples are static for demonstration purposes. In production, generate a fresh timestamp for each request using your system's current time in ISO 8601 UTC format (e.g., 2025-01-06T15:30:00Z).

Examples:

```
# Raw WHOIS with HMAC
curl 'https://api.domaintools.com/v1/download/5_min_domain_whois/?api_
↳ username=YOUR_USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

```
# Parsed WHOIS with HMAC
curl 'https://api.domaintools.com/v1/download/5_min_domain_whois_parsed/?api_
↳ username=YOUR_USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

15.3.3 Open key authentication

This is the easiest authentication scheme to implement, but also the least secure. Each request contains the full API key and API username as query parameters. We recommend using [API key header authentication](#) or [HMAC authentication](#) instead.

If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

Required query parameters:

- api_username: Your API username
- api_key: Your API key

Examples:

```
# Raw WHOIS
curl 'https://api.domaintools.com/v1/download/5_min_domain_whois/?api_
↳ username=YOUR_USERNAME&api_key=YOUR_API_KEY'
```

```
# Parsed WHOIS
curl 'https://api.domaintools.com/v1/download/5_min_domain_whois_parsed/?api_
↳ username=YOUR_USERNAME&api_key=YOUR_API_KEY'
```

15.4 Daily Download API

The Daily Download API provides access to 5-minute WHOIS data through temporary AWS S3 file links. Files are organized by date and time, with new files generated every 5 minutes.

15.4.1 Base URLs

Raw WHOIS:

```
https://api.domaintools.com/v1/download/5_min_domain_whois/
```

Parsed WHOIS:

```
https://api.domaintools.com/v1/download/5_min_domain_whois_parsed/
```

15.4.2 Daily Download parameters

The Daily Download API supports standard download parameters:

15.4.2.1 api_username

Type: string (required for HMAC and open key auth)

Your DomainTools API username

15.4.2.2 api_key

Type: string (required for open key auth)

Your DomainTools API key

15.4.2.3 signature

Type: string (required for HMAC auth)

HMAC signature of your request

15.4.2.4 timestamp

Type: string (required for HMAC auth)

Current timestamp for HMAC authentication in ISO 8601 format

15.4.2.5 limit

Type: integer (optional)

Limit the list of signed files. Ordering of files is always descending, so the latest files are first.

15.4.2.6 page

Type: integer (optional)

Select which page of results are returned. Pages begin at 0 with latest results.

15.4.2.7 prefix

Type: string (optional)

Filter results by date and time using the file prefix (format: YYYYMMDDHHMM).

Example: `?prefix=2025062420` filters for files from June 24, 2025 at 8:00 PM

15.4.3 Daily Download response structure

The API returns a JSON response with signed URLs for downloadable files:

download_name (string): The feed identifier (5_min_domain_whois or 5_min_domain_whois_parsed)

files (array): List of downloadable file entries

Each file object contains:

- **name** (string): File path
- **last_modified** (string): Last modified date in ISO 8601 format
- **etag** (string): Entity tag (hash of the file)
- **size** (integer): Size in bytes
- **url** (string): Signed AWS download URL (valid for 12 hours)

15.4.4 Daily Download response codes

200: OK - The request was successful

400: Bad request

401: Unauthorized

403: Forbidden

404: No data to download

15.4.5 Daily Download file naming

Files follow this naming pattern:

Parsed version:

```
YYYYMMDDTTTT.json.gz
```

Raw version:

```
YYYYMMDDTTTT.gz
```

Where: - YYYYMMDD = Date (e.g., 20250624) - TTTT = Time in 24-hour format (e.g., 2000 for 8:00 PM)

Examples: - 202506242000.json.gz (parsed, June 24, 2025 at 8:00 PM) - 202506242000.gz (raw, June 24, 2025 at 8:00 PM)

15.4.6 File contents

15.4.6.1 Parsed version fields

The parsed version contains JSON with the following fields:

- Domain name
- Parse success (y/n)
- Server (WHOIS)
- Lookup Date

-
- Lookup Time
 - Create Date
 - Updated Date
 - Expires Date
 - Registrar Name
 - Registrar Abuse Contact: Phone
 - Registrar Abuse Contact: Email
 - Registrar IANA ID
 - Registrar URL
 - Registrar WHOIS Server
 - Admin Name
 - Admin Org
 - Admin Street
 - Admin City
 - Admin State/Province
 - Admin Postal Code
 - Admin Country
 - Admin Phone
 - Admin Fax
 - Admin Email
 - Billing Name
 - Billing Org
 - Billing Street
 - Billing City
 - Billing State/Province
 - Billing Postal Code
 - Billing Country
 - Billing Phone
 - Billing Fax
 - Billing Email
 - Registrant Name
 - Registrant Org
 - Registrant Street
 - Registrant City
 - Registrant State/Province
 - Registrant Postal Code
 - Registrant Country
 - Registrant Phone
 - Registrant Fax
 - Registrant Email
 - Technical Name
 - Technical Org
 - Technical Street
 - Technical City
 - Technical State/Province
 - Technical Postal Code
 - Technical Country
 - Technical Phone
 - Technical Fax
 - Technical Email
 - Name Server
 - Registrar Status
 - Raw WHOIS Data Blob

15.4.6.2 Raw version format

The raw version contains unparsed WHOIS data as tab-separated values, with the raw WHOIS text for each domain.

15.4.7 Daily Download examples

List available files (parsed version):

```
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/5_min_domain_whois_parsed/?limit=10'
```

Filter by date and hour:

```
# Get files from June 24, 2025 at 8:00 PM  
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/5_min_domain_whois_  
↳ parsed/?prefix=202506242000'
```

Download a specific file:

```
# Get the file list  
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/5_min_domain_whois_parsed/?limit=1' >  
↳ files.json  
  
# Download the file  
curl -o whois-data.json.gz "$(jq -r '.response.files[0].url' files.json)"  
  
# Decompress and view  
gunzip whois-data.json.gz  
head whois-data.json
```

Download raw version:

```
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/5_min_domain_whois/?limit=1' >  
↳ files.json  
  
curl -o whois-raw.gz "$(jq -r '.response.files[0].url' files.json)"
```

15.5 Related resources

- [Parsed Domain RDAP feed](#)
- [Domain Discovery feed](#)

16 5-Minute IP WHOIS

The 5-Minute IP WHOIS feed provides the most recently updated IPv4 WHOIS records, processed on a 5-minute basis. This feed is available in both raw (unparsed) and parsed formats, making it suitable for various integration and analysis workflows.

16.1 Overview

This feed captures all IPv4 WHOIS records that have been updated since the previous 5-minute processing cycle. The feed is available in two versions:

- **Raw version** (`5_min_ip_whois`): Unparsed WHOIS data as received from Regional Internet Registries (RIRs)
- **Parsed version** (`5_min_ip_whois_parsed`): Structured JSON format with normalized fields

Use this feed when you need to:

- Monitor IP address allocation and ownership changes
- Track IP WHOIS record updates for threat intelligence
- Analyze network infrastructure changes
- Build IP intelligence databases
- Detect suspicious IP allocation patterns
- Automate network intelligence workflows
- Correlate IP ownership with threat activity

Inclusion criteria: All IPv4 addresses processed since the previous 5-minute update cycle.

Format: Gzip-compressed tab-separated (TSV) or JSON text files

Size: Up to 10MB per day

16.2 Requirements

You need the following to access Threat Feeds:

- An Enterprise Account with DomainTools, accessible at <https://account.domaintools.com/my-account/>
- Authentication credentials (API key for header authentication, or API username and key for HMAC or open key authentication)
- A way to interact with a REST API delivered through AWS

Obtain your API credentials from your group's API administrator. API administrators can manage their API keys at <https://research.domaintools.com>, selecting the drop-down account menu and choosing API admin.

For assistance, contact enterprisesupport@domaintools.com.

16.3 Authentication

You can authenticate to the 5-Minute IP WHOIS API using three different methods. Choose the method that best fits your security requirements and technical environment.

16.3.1 API key (header) authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and authenticates your requests.

Required header:

X-Api-Key: YOUR_API_KEY

Examples:

```
# Raw WHOIS
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/5_min_ip_whois/'
```

```
# Parsed WHOIS
curl -H 'X-Api-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/5_min_ip_whois_parsed/'
```

16.3.2 HMAC authentication

HMAC authentication is a secure alternative to API key-based methods. It requires signing each request with a SHA1 HMAC digest derived from your API secret, providing integrity and authenticity without exposing credentials directly in the request.

This method is recommended for systems where authentication credentials shouldn't be stored in plain text or included directly in request URLs.

DomainTools supports MD5, SHA1, and SHA256 for the hashing algorithm.

Required query parameters:

- `api_username`: Your DomainTools API username
- `signature`: HMAC-SHA1 signature of `api_username + timestamp + uri_path`
- `timestamp`: Current UTC timestamp in ISO 8601 format (for example, 2025-06-01T15:30:00Z)

Constructing the HMAC signature:

```
signature = HMAC-SHA1(api_key, api_username + timestamp + uri_path)
```

Important: URI path must include API version

The `uri_path` parameter must include the API version prefix. For example, use `/v1/feed/nod/` not `/feed/nod/`.

Example Python signing function:

```
import hmac
import hashlib

def sign(api_username, api_key, timestamp, uri):
    params = f"{api_username}{timestamp}{uri}"
    return hmac.new(api_key.encode("utf-8"), params.encode("utf-8"),
        ↪ hashlib.sha1).hexdigest()
```

Note: HMAC timestamp requirements

The timestamp parameter in HMAC authentication must be current (within a few minutes of the server time). The timestamps shown in these examples are static for demonstration purposes. In production, generate a fresh timestamp for each request using your system's current time in ISO 8601 UTC format (e.g., 2025-01-06T15:30:00Z).

Examples:

```
# Raw WHOIS with HMAC
curl 'https://api.domaintools.com/v1/download/5_min_ip_whois/?api_username=YOUR_
↳ USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

```
# Parsed WHOIS with HMAC
curl 'https://api.domaintools.com/v1/download/5_min_ip_whois_parsed/?api_
↳ username=YOUR_USERNAME&signature=HMAC_SIGNATURE&timestamp=2025-01-06T15:30:00Z'
```

16.3.3 Open key authentication

This is the easiest authentication scheme to implement, but also the least secure. Each request contains the full API key and API username as query parameters. We recommend using [API key header authentication](#) or [HMAC authentication](#) instead.

If you're unsure about your authentication options, contact enterprisesupport@domaintools.com.

Required query parameters:

- api_username: Your API username
- api_key: Your API key

Examples:

```
# Raw WHOIS
curl 'https://api.domaintools.com/v1/download/5_min_ip_whois/?api_username=YOUR_
↳ USERNAME&api_key=YOUR_API_KEY'
```

```
# Parsed WHOIS
curl 'https://api.domaintools.com/v1/download/5_min_ip_whois_parsed/?api_
↳ username=YOUR_USERNAME&api_key=YOUR_API_KEY'
```

16.4 Daily Download API

The Daily Download API provides access to 5-minute IP WHOIS data through temporary AWS S3 file links. Files are organized by date and time, with new files generated every 5 minutes.

16.4.1 Base URLs

Raw WHOIS:

```
https://api.domaintools.com/v1/download/5_min_ip_whois/
```

Parsed WHOIS:

```
https://api.domaintools.com/v1/download/5_min_ip_whois_parsed/
```

16.4.2 Daily Download parameters

The Daily Download API supports standard download parameters:

16.4.2.1 api_username

Type: string (required for HMAC and open key auth)

Your DomainTools API username

16.4.2.2 api_key

Type: string (required for open key auth)

Your DomainTools API key

16.4.2.3 signature

Type: string (required for HMAC auth)

HMAC signature of your request

16.4.2.4 timestamp

Type: string (required for HMAC auth)

Current timestamp for HMAC authentication in ISO 8601 format

16.4.2.5 limit

Type: integer (optional)

Limit the list of signed files. Ordering of files is always descending, so the latest files are first.

16.4.2.6 page

Type: integer (optional)

Select which page of results are returned. Pages begin at 0 with latest results.

16.4.2.7 prefix

Type: string (optional)

Filter results by date and time using the file prefix (format: YYYYMMDDHHMM).

Example: `?prefix=2025062420` filters for files from June 24, 2025 at 8:00 PM

16.4.3 Daily Download response structure

The API returns a JSON response with signed URLs for downloadable files:

download_name (string): The feed identifier (5_min_ip_whois or 5_min_ip_whois_parsed)

files (array): List of downloadable file entries

Each file object contains:

- **name** (string): File path
- **last_modified** (string): Last modified date in ISO 8601 format
- **etag** (string): Entity tag (hash of the file)
- **size** (integer): Size in bytes
- **url** (string): Signed AWS download URL (valid for 12 hours)

16.4.4 Daily Download response codes

200: OK - The request was successful

400: Bad request

401: Unauthorized

403: Forbidden

404: No data to download

16.4.5 Daily Download file naming

Files follow this naming pattern:

Parsed version:

```
YYYYMMDDTTTT.json.gz
```

Raw version:

```
YYYYMMDDTTTT.gz
```

Where: - YYYYMMDD = Date (e.g., 20250624) - TTTT = Time in 24-hour format (e.g., 2000 for 8:00 PM)

Examples: - 202506242000.json.gz (parsed, June 24, 2025 at 8:00 PM) - 202506242000.gz (raw, June 24, 2025 at 8:00 PM)

16.4.6 File contents

16.4.6.1 Parsed version fields

The parsed version contains JSON with the following fields:

- RIR queried
- Net Range
- CIDR
- Net Name
- Net Handle

- Parent
- Net Type
- Origin AS
- Organization
- RegDate
- Updated
- Org Name
- Org ID
- City
- State
- Postal Code
- Country
- RegDate
- Updated
- Ref
- Referral Server
- OrgAbuseHandle
- OrgAbuseName
- OrgAbusePhone
- OrgAbuseEmail
- OrgAbuseRef
- OrgTechHandle
- OrgTechName
- OrgTechPhone
- OrgTechEmail
- OrgTechRef
- OrgNOCHandle
- OrgNOCName
- OrgNOCPhone
- OrgNOCEmail
- OrgNOCTRef
- Comments
- Raw IP WHOIS data blob

16.4.6.2 Raw version format

The raw version contains unparsed IP WHOIS data as tab-separated values, with the raw WHOIS text for each IP address.

16.4.7 Daily Download examples

List available files (parsed version):

```
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/5_min_ip_whois_parsed/?limit=10'
```

Filter by date and hour:

```
# Get files from June 24, 2025 at 8:00 PM  
curl -H 'X-Api-Key: YOUR_API_KEY' \  
  'https://api.domaintools.com/v1/download/5_min_ip_whois_  
↪ parsed/?prefix=202506242000'
```

Download a specific file:

```
# Get the file list
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/5_min_ip_whois_parsed/?limit=1' >
↳ files.json

# Download the file
curl -o ip-whois-data.json.gz "$(jq -r '.response.files[0].url' files.json)"

# Decompress and view
gunzip ip-whois-data.json.gz
head ip-whois-data.json
```

Download raw version:

```
curl -H 'X-API-Key: YOUR_API_KEY' \
  'https://api.domaintools.com/v1/download/5_min_ip_whois/?limit=1' > files.json

curl -o ip-whois-raw.gz "$(jq -r '.response.files[0].url' files.json)"
```

16.5 Related resources

- [5-Minute Domain WHOIS feed](#)
- [Domain Discovery feed](#)

17 Threat Feeds via Response Policy Zone (RPZ)

17.1 Introduction

DomainTools provides threat feeds as DNS Response Policy Zones (RPZ) via DNS zone transfers. RPZ is a standard method for DNS servers to block access to malicious domains.

Your RPZ-enabled DNS resolver responds with an NXDOMAIN (“no such domain”) status code for queries matching domains or subdomains in the threat feed. This effectively makes the listed domains unavailable for users of this DNS firewall.

17.2 Available feeds

17.2.1 Newly Observed Domains (NOD)

Newly Observed Domains are domains that DomainTools has observed for the first time in DNS traffic.

- 5m.nod.rpz.domaintools.com.
- 10m.nod.rpz.domaintools.com.
- 30m.nod.rpz.domaintools.com.
- 1h.nod.rpz.domaintools.com.
- 3h.nod.rpz.domaintools.com.
- 12h.nod.rpz.domaintools.com.
- 24h.nod.rpz.domaintools.com.

17.2.2 Newly Active Domains (NAD)

Newly Active Domains are existing domains that were recently seen after at least ten days of not being observed.

- 5m.nad.rpz.domaintools.com.
- 10m.nad.rpz.domaintools.com.
- 30m.nad.rpz.domaintools.com.
- 1h.nad.rpz.domaintools.com.
- 3h.nad.rpz.domaintools.com.
- 12h.nad.rpz.domaintools.com.

17.2.3 Risk Score Based Domain Hotlists

- 1k.domainhotlist.rpz.domaintools.com.
- 100k.domainhotlist.rpz.domaintools.com.
- 90s.domainhotlist.rpz.domaintools.com.
- 95s.domainhotlist.rpz.domaintools.com.
- 99s.domainhotlist.rpz.domaintools.com.

17.2.3.1 Hotlist Variant Descriptions

Each Domain Hotlist variant filters domains based on different [Domain Risk Score](#) thresholds:

- **90s.domainhotlist.rpz.domaintools.com:** Domains with Proximity ≥ 70 OR (Malware Risk ≥ 90 AND Phishing Risk ≥ 90)

- **95s.domainhotlist.rpz.domaintools.com**: Domains with Proximity ≥ 85 OR (Malware Risk ≥ 95 AND Phishing Risk ≥ 95)
- **99s.domainhotlist.rpz.domaintools.com**: Domains with Proximity ≥ 85 OR (Malware Risk ≥ 99 AND Phishing Risk ≥ 99)
- **1k.domainhotlist.rpz.domaintools.com**: Top 1,000 highest-risk domains with Proximity ≥ 75 OR (Malware Risk ≥ 90 AND Phishing Risk ≥ 90)
- **100k.domainhotlist.rpz.domaintools.com**: Top 100,000 highest-risk domains with Proximity ≥ 75 OR (Malware Risk ≥ 90 AND Phishing Risk ≥ 90)

All variants require passive DNS activity within the last 24 hours.

For detailed information about accessing these feeds via API, see the [Threat Feeds documentation](#).

17.3 Zone naming format

The nod and nad zone name labels are constructed (left to right) from a time period, followed by a list name, followed by `.rpz.domaintools.com`:

```
[interval].[list].rpz.domaintools.com
```

Available time intervals: 5m, 10m, 30m, 1h, 3h, 12h, and for the nod feed, 24h.

For example, the 1 hour NOD list `1h.nod.rpz.domaintools.com` contains the most recent hour of entries.

The larger time intervals are a superset that include the smaller time intervals. Smaller time intervals have smaller zone sizes and may be available a little faster.

For details about Domain Hotlist feeds, including risk score thresholds and expiration times, see the [Domain Hotlist documentation](#).

17.4 How RPZ blocks domains

RPZ zones use a standard DNS format to define blocking rules. Each DNS zone is formatted in accordance with the draft specification for [DNS Response Policy Zones](#). Each DNS firewall rule uses NXDOMAIN policy actions by default (CNAME `.`). Each domain entry contains both apex and wildcard `*.` entries (two records per domain).

For example:

```
$ORIGIN RPZ.EXAMPLE.ORG.  
example.com          CNAME  .  
*.example.com       CNAME  .
```

Your DNS resolver responds with an NXDOMAIN (“no such domain”) status code for queries matching domains in the RPZ feed. The NXDOMAIN response includes an SOA (Start of Authority) record in the ADDITIONAL section of the DNS response (and no ANSWER section). You can use this SOA record to verify the RPZ feed is working.

17.5 Configure your RPZ connection

DomainTools delivers the RPZ feeds over Incremental Zone Transfers (IXFR) and full zone transfers (AXFR). DNS NOTIFY is also used to indicate updates to the zones to trigger zone checks and transfers.

17.5.1 Provide your IP addresses to DomainTools

DomainTools restricts access to RPZ feeds by IP address to ensure security. Before you can connect, provide two sets of IP addresses to enterprisesupport@domaintools.com:

- The IP address(es) from which you connect to the RPZ provider DNS server
- The IP address(es) to which you would like DNS NOTIFY messages sent

You typically use the same addresses for both.

17.5.2 Configure your firewall

You must also add rules to your firewall's access control list(s) for DomainTools hosts to send UDP packets to port 53 of your DNS server, so that it can receive the DNS NOTIFY packets for timely updates.

- IPv4: 104.244.13.88 Port: 53
- IPv4: 104.244.14.88 Port: 53

17.5.3 Authenticate with TSIG

DomainTools uses TSIG (Secret Key Transaction Authentication for DNS) for authorization to use the RPZ feeds. This uses shared secrets and one-way hashing to authenticate DNS messages coming from approved DNS servers. Contact enterprisesupport@domaintools.com for the details.

- TSIG key: DomainTools Enterprise Support provides this
- TSIG key algorithm: hmac-sha512
- TSIG key name: DomainTools Enterprise Support provides this

17.6 Test your RPZ configuration

When your DNS resolver blocks a domain using RPZ, the response includes an SOA (Start of Authority) record that identifies which RPZ feed was used. For example, the following label tells you what RPZ feed was used and the SOA RNAME field indicates the primary DNS server (under `rpz.domaintools.com`) that hosts this RPZ zone:

```
;; ADDITIONAL SECTION:
3h.nod.rpz.domaintools.com. 86400 IN SOA rpz-ns1.domaintools.com.
↪ noc.domaintools.com. 946684799 600 300 86400 86400
```

DomainTools uses the SOA SERIAL number as the timestamp (in Unix epoch format) for when this feed was last regenerated.

The RPZ threat feeds also have a testing domain entry and its corresponding wildcard which you can use to verify the RPZ feed is loaded and working. You can do a DNS lookup, like with `dig`, for `test.rpz.domaintools.test` and it should respond with the NXDOMAIN and the ADDITIONAL section SOA record indicating its originating RPZ feed. If the SOA record response

is in the AUTHORITY section instead and it isn't under the specific feed name, such as `3h.nod.rpz.domaintools.com.`, then it didn't come from the RPZ.

See your RPZ related debugging logs for your DNS resolver for additional troubleshooting information.

17.7 Advanced configuration

17.7.1 Maintain a local allowlist

DomainTools recommends that you maintain an allowlist Response Policy Zone as needed for overrides when you have more specific information about a domain than DomainTools.

17.7.2 Log NAD matches instead of blocking

Because NAD feeds may include legitimate domains that happen to be inactive for extended periods, DomainTools recommends that you begin by logging instead of automatically blocking domains on the Newly Active Domains (NAD) list.

The following BIND (Berkeley Internet Name Domain) named configuration shows how to log the response policy rewrites (at the default info severity level):

```
logging {
  channel named-rpz {
    file "/var/log/named/named-rpz.log" versions 3 size 250k;
    print-time yes;
    print-category yes;
    print-severity yes;
  };
  category rpz {
    named-rpz;
  };
};
```

Or to log to the default logger:

```
logging {
  category rpz { default_syslog; };
};
```

The NAD (newly active domains) feeds contain existing domains that were recently seen after at least ten days of not being observed. It's recommended to not block on these NAD rules, but log only. You can do this by overriding the default NXDOMAIN behavior, for example, with BIND named:

```
response-policy {
  zone "5m.nad.rpz.domaintools.com" policy DISABLED;
};
```

An example single-line log output indicating a "disabled" match:

```
19-Nov-2025 21:00:11.305 rpz: info: client @0x7f10a0973168
127.0.0.1#55312 (test.rpz.domaintools.test): disabled rpz
```

```
QNAME NXDOMAIN rewrite test.rpz.domaintools.test/A/IN via
test.rpz.domaintools.test.5m.nad.rpz.domaintools.com
```

17.7.3 Redirect to a walled garden

A *walled garden* redirects blocked domains to an internal page instead of returning NXDOMAIN. This allows you to inform users why a domain is blocked. While the RPZ feeds define an NXDOMAIN response as the policy action, you can configure the DNS firewall to redirect users of the matching domain to a walled garden instead, restricting their access to specific external services or content. An example of configuring the RPZ for a walled garden for BIND named follows:

```
response-policy {
    # walled garden
    # override the NXDOMAIN and redirect to other domain instead
    zone "1k.domainhotlist.rpz.domaintools.com" policy CNAME captive.internal;
};
```

Based on this example, an internal website named `captive.internal` can be used to share information about the “Firewall” policies. That target URL’s domain name will also need to be configured.

The walled garden scenario can be useful for restricting website access but can’t work for anti-spam blocking. You may need to use separate DNS firewalls for walled garden website access versus DNS resolution used for email servers.