DOMAINTOOLS API:
Reverse IP Evaluation Plan

The DomainTools Reverse IP API finds domain names that share a common IP address (typically, the same web hosting server). It can be a useful way to uncover identities of individuals acting from a given IP or expanding a list of domain names when other methods such as Reverse Whois are ineffective due to domain privacy.

DOMAIN NAME QUERIES VERSUS IP QUERIES

The Reverse IP dataset can be queried either by a domain name or an IP address. This flexibility can lead to some confusion. Fundamentally, Reverse IP works on IP addresses – it finds domain names with DNS A records that were known to point to an IP address when DomainTools last refreshed its dataset. Often, research for connected domains begins with a domain name itself, and in those cases, you can submit that domain name to Reverse IP directly. The system will look for the last IP address DomainTools observed that domain to point to, and then it will find other domains pointed to the same IP.

It is important to note the automatic conversion in the Reverse IP API from a domain name to an IP address is not a live DNS query, and should not be used as a substitute for that. The purpose is to find connected domains as efficiently as possible, not to act as a DNS resolver.

BASIC IMPLEMENTATION

1. Start with either a domain name or a single IP address and select the correct Reverse IP API RESTful query path (see documentation for details).

2. Query the API and store the count of matched domains with the list of domains themselves.

3. For each domain in the result set, query the DomainTools Parsed Whois API and store the attributes of the domains for further analysis. Consider also obtaining the DomainTools Reputation Score from a subsequent query to that API as well.

INTEGRATED EVALUATION

This test plan begins with a single domain name, applies a basic test to determine if a Reverse IP result would be useful, and then expands that query to a list of related domains with their attributes. It also establishes persistent monitoring to keep the result set current. Review the Evaluation Plan for each referenced API for more details.

1. Query the DomainTools Domain Profile API for the target domain name. Locate the number of connected domains by IP address in the result set.

2. Apply a business rule to that number and decide whether to proceed with a Reverse IP query. As a starting point, consider that IP addresses with less than 500 domains are likely to return domains that are closely associated with each other.

3. If the number of matched domains is within accepted parameters, query the Reverse IP API for the domain and obtain the list of related domains.

4. For each domain, query the DomainTools Parsed Whois API, Domain Profile API and/or Domain Reputation API and store the results with the domains.

5. Add the IP address to the list of IPs you are monitoring with the DomainTools IP Monitor API. Align the structure of your Reverse Whois and IP Monitor data stores so the results of a Reverse IP query can be presented to the end user together with new changes over time.

USAGE TIPS

1. Unlike the web interface, the Reverse IP API does not support wildcards. If you are trying to find adjacent domains within the same IP netblock, you will need to submit multiple queries.

2. If you are planning to submit multiple, identical Reverse IP queries to find changes between result sets over time, consider using the IP Monitor API instead. It is ideally suited for that specific use case.

3. Your service level may not permit you to obtain all the domains on a given IP address, or that IP address may simply have too many domains to be useful. In those cases, the list of domains will be truncated at a certain upper limit. You should compare the count of matched domains returned at the start of the result set with the actual count of domains returned to determine whether your results have been cut off, and if they have, communicate that to the end user. This is why it is critical to store the result count as a distinct field in your data store.