Use nslookup for Test Lookups (DNS Suffixes)

Contents

Introduction

This document describes how to use nslookup for test lookups.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on Cisco Umbrella.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Overview

Using nslookup to check DNS query responses is commonly used in troubleshooting DNS issues. In some scenarios, queries can appear to return an extra level of a domain. For example, looking up sub.domain.com yields a query and answer for sub.domain.com.domain.com.

nslookup: Resolution Algorithm Differences

When querying DNS, one utility is ubiquitous across all modern operating systems: nslookup. While older and less capable than dig, Windows users are limited by default to nslookup. It is important to note that nslookup handles DNS differently than dig or the local system.

For a Public Query with No Public Wildcard

nslookup:

1. Query made for domain.com (nslookup domain.com).

2. nslookup sends "domain.com.suffix" and checks for a response - NXDOMAIN.

3. nslookup sends "domain.com.secondsuffix" and checks for a response - NXDOMAIN.

4. nslookup sends "domain.com" and returns the response.

System DNS or Dig

1. Query made for domain.com (dig domain.com).

2. dig or the system sends a DNS packet lookup up "domain.com" and returns the response

3. If the earlier information is non-existing, a DNS packet can be generated for "domain.com.suffix"

4. If the earlier information is non-existing, a DNS packet can be generated for "domain.com.secondsuffix"

In a scenario where there is no local answer and only a public answer exists, this acts exactly the same. The only difference in the earlier scenario is if packets are captured, the nslookup scenario can be sending strange looking suffix appended queries.

For a Public Query where a DNS Suffix has a Public Wildcard

nslookup:

1. Query made for domain.com (nslookup domain.com)

2. nslookup sends "domain.com.suffix" and checks for a response. Response is returned (suffix is a public wildcard domain). An answer is found for domain.com.suffix, no further queries are made.

System DNS or Dig

1. Query made for domain.com (dig domain.com).

2. dig or the system sends a DNS packet lookup up "domain.com" and returns the response for domain.com.

As a result, nslookup can return a completely different DNS answer than users utilizing a computer's web browser and can lead to perceived incorrect DNS responses. This can also lead to "double" appearances of domains if the queried DNS record match the computer's suffix list.

Working Solution to Use nslookup for Public Wildcard DNS Search Suffix Domain

When querying DNS, apply a "." at the end of the query unless using nslookup to query a hostname. This can look up the exact query requested. "nslookup domain.com." can request only domain.com without suffixes first.

Appearance in Umbrella Reporting

In certain scenarios, this behavior can be observable in Umbrella reports. Entries can appear such as "facebook.com.domain.local" or "google.com.domain.local" when this is occurring. In most cases, this is nslookup performing those local queries first. If your suffixes are not authoritative on the DNS zone, they can be forwarded to Umbrella rather than being returned NXDOMAIN by the local DNS server on network.

Special Case: Umbrella Roaming Client

If your applied DNS search suffix domain is a public wildcard and also used internally, you can also observe the suffix-doubled behavior noted earlier. Queries for host.domain.com can appear as host.domain.com.domain.com in your reports (despite being on the internal domains list). If domain.com is a public wildcard, add "domain.com.domain.com" to your internal domains list to resolve any observed user impact.