Measuring Security of Closed DNS Resolvers

Sarah Scheffler, Sean Smith, Yossi Gilad, Sharon Goldberg

BU Department of Computer Science

{sscheff, swsmith, yossigi}@bu.edu, goldbe@cs.bu.edu

bu.edu/cs

Goals

In response to the spread of cache poisoning attacks, many DNS resolvers have gone from being *open* to *closed* resolvers, meaning that they will only perform queries on behalf of hosts within a single organization or Internet Service Provider. As a result, measuring the security of the DNS infrastructure has been made more difficult. Closed resolvers will not respond to researcher queries to determine if they utilize security measures like port randomization or transaction id randomization. However, we can effectively turn a closed resolver into an open one by sending an email to a mail server (MTA) in the organization. This causes the MTA to make a query on the external researchers' behalf, and we can log the security features of the DNS resolver using information gained by a nameserver and email server under our control. The goals of this experiment are

1. to measure the security of closed DNS resolvers using Email
2. to measure the relationship between MTAs and DNS resolvers that make queries on their behalf
3. to measure what DNS queries are made as a result of sending an email under several different spam-prevention measures

Email Spam Prevention and DNS

Mail servers cause several DNS queries to be made as anti-spam measures. This experiment measures the DNS queries caused by sending an email

1. by itself
2. under several different Sender Policy Framework (SPF) configurations
3. with DomainKeys Identified Mail (DKIM) and Domain Message Authentication Reporting & Conformance (DMARC)

We have found instances where SPF records are checked such a in a way that allows us to craft an infinite chain of DNS lookups. Such an attack could be the injection vector for a Kaminsky DNS cache poisoning attack. We will determine how many systems are vulnerable to such an attack.

Measurement Design

Measurement Flow

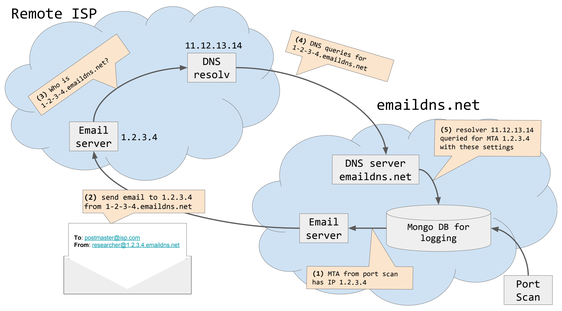

Figure 1: The experiment queries MTAs found in port scan, then measures what kinds of DNS queries were made from what servers in response.

DoS Amplification via SPF

Sender Policy framework (SPF) is an anti-spam email feature where the receiving MTA checks the DNS `TXT` records of the domain sending the email. The `TXT` record indicates which IP addresses are allowed to send email.

One feature of SPF records allows you to specify `include:anotherdomain.com` so that it recursively looks up another domain's SPF records. If this passes, the whole SPF record passes, but if it fails, it checks the next SPF record. The RFC states that the system must not lookup more than 10 includes, but in practice, we've seen some mail servers check more than 10 includes. Our attack is as follows:

- Send an email from `us@ourdomain.com`
- MTA looks up our SPF record in which we have N include records `include:anotherdomain.com`

We can change the length of N, have the SPF records include themselves, or have many nested includes to ourselves. Using these techniques we can create many DNS queries that come from the MTA and not directly from the attacker, bypassing some DoS defenses.

Overview

We send emails to MTAs, encoding information about the MTA in our sending address. We then measure what queries were made to our nameserver, and from what IP addresses. This enables us to determine which DNS resolver makes queries for which MTA, and by looking at repeated queries, we determine whether the resolver has security features like transaction id randomization and port randomization.

There are five phases to this experiment:

1. Do an Internet-wide port scan on ports commonly used for SMTP (25, 465, 587, and 2525)
2. For each IP address in the scan, send an email to a recipient served by that MTA with a sending address that encodes the IP address of the MTA
3. The MTA will ask its DNS resolver to do some anti-spam checks on the given email address
4. The resolver asks our nameserver about the email address we sent from
5. Log all queries to our nameserver, they tell us:
   - The IP address of the DNS resolver makes queries on behalf of this MTA
   - What anti-spam protection this MTA had implemented
   - Whether the DNS resolver had implemented security measures against cache poisoning, like port randomization or transaction id randomization

Each of these steps must be done quickly to ensure that the public IP addresses of the MTAs and DNS resolvers do not change.

Glossary

DNS - Domain Name System
MTA - Mail Transfer Agent (Email server)
SPF - Sender Policy Framework (anti-spam measure implemented in some MTAs)

DNS Cache Poisoning via Email

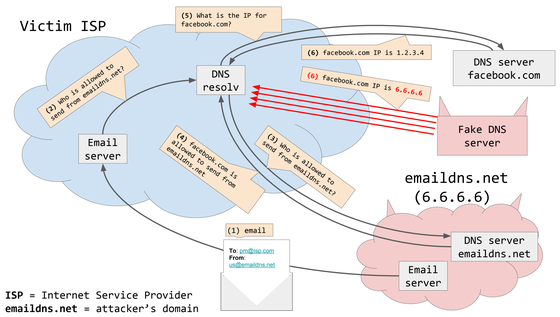

Figure 2: Executing a DNS cache poisoning attack on a closed resolver via Email

A Kaminsky attack "poisons" the cache of a DNS resolver by causing a query for an uncached subdomain and then sending a flood of packets to the resolver with fake responses claiming that that the IP address of the subdomain is a domain under the attacker's control. Eventually, a benign response from the real domain will come as well, but it is discarded because the resolver already has the false answer in its cache.

In response to this attack, several ISPs closed their DNS resolvers to reduce the ability of the attacker to cause queries from their resolver. Two other responses are *port randomization* and *transaction id randomization*, in which outgoing queries have their ports and transaction ids randomized, respectively. The resolver only accepts a response if the transaction id and port match the transaction id and port from the query. This forces the attacker to send a much larger amount of traffic in order to force one of their fraudulent packets to become the accepted answer. One of our goals in this experiment is to determine the extent to which closed resolvers have implemented these randomization defenses.

Acknowledgements

We thank Jared Mauch for his invaluable help in designing this experiment to be effective and repeatable, and for providing us infrastructure needs. We also thank Timothy Edgar for discussions on the legal implications of the attacks proposed.

@BUCompSci