USENIX'23 Artifact Appendix: How Effective is Multiple-Vantage-Point Domain Control Validation?

Grace H. Cimaszewski[†] Henry Birge-Lee[†] Liang Wang[†] Jennifer Rexford[†]
Prateek Mittal[†]
[†] Princeton University

A Artifact Appendix

A.1 Abstract

We model the security of multiple-vantage-point domain control validation (more briefly, multiVA) by performing quantitative Internet-level simulations of the full-graph DNS resolution of domain names included in Let's Encrypt certificates. At a high level, the submitted artifact consists of 3 parts:

1. the *Internet topology simulator*, which calculates the effects of equally-specific prefix length BGP hijacks by selected attacker ASes;
2. the *DNS resolver*, which performs full-graph DNS lookups of domain names to record all IP addresses vulnerable to BGP hijacks (i.e., not DNSSEC-signed);
3. the *resilience processor*, which combines the output of (1) and (2) to compute a resilience value in the range of 0-1.0 to describe how likely a domain name may be attacked by a random attacker AS using BGP hijacks during the domain control validation process to gain a fraudulent certificate.

This artifact aims to reproduce the results in sections 7 and 8 of our paper.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

We extract only the domain names listed in the subjects of Let's Encrypt-issued certificates, which are publicly available in certificate transparency logs. The DNS lookup tool performs a default number of 10 lookups per domain name, which is assumed to be a manageable request volume for the domains' nameservers. BGP simulations do not entail any real hijacks or announcement of prefixes, and do not leak information about private routing policies. Running our code does not require any admin/sudo privileges or elevated access.

A.2.2 How to access

The artifact can be accessed by downloading our tagged public Github project Github project. All the requisite data, containers, and code are contained within the Git repository. Git clone the artifact access URL provided in HotCRP submission.

A.2.3 Hardware dependencies

For Experiment E1, simulation load is CPU-intensive: to be able to run the end-to-end experiments in a practicable amount of time, access to a many-cored (e.g., 64+) computing system with ample memory (approximately 2GB per core). Performing DNS lookups will require Internet access and corresponding firewall rules to allow inbound/outbound traffic.

A.2.4 Software dependencies

The main software dependency needed to run the artifact is Docker (available for install at `https://www.docker.com/`). Other required dependencies (Python 3.8, libraries, etc.) are packaged within the containerized environment.

A.2.5 Benchmarks

None.

A.3 Set-up

A.3.1 Installation

Git clone the artifact access URL: `https://github.com/inspire-group/routing-aware-dns/commit/23194fc824633122cbfb79206a62ac662389f63c`.
`cd` into the cloned repository directory. From here, build the Docker container:

```
docker build --tag full-graph-dns-resolver .
```

After image build successfully completes, begin running container in the background:

```
docker run --name dns-resolver -d
full-graph-dns-resolver
```

From here, enter the container with interactive shell to execute the subsequent commands for the artifact:

```
docker exec -it dns-resolver bash
```

More detailed instructions for setup are included in the README of the artifact repository.

A.3.2 Basic Test

Validate that the DNS full-graph resolver tool properly executes (can send/receive DNS queries, local Unbound stub resolver is live):

```
python3 log_processor_artifact.py -d
data/domains_random_samp_small.txt
```

This runs lookups for a sample of 1397 domains (0.1% of our dataset) for validation purposes.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): *Resilience of domain names takes a noticeable hit when including the DNS nameserver in the BGP hijack attack surface. Considering the current level of RPKI deployment in the Internet counterbalances some of this resilience hit. This is illustrated by experiment (E3), which reproduces results described in Section 7 of the paper.*

(C1): *multiVA deployments with only one or two additional vantage points in diverse public cloud providers can strengthen resilience values to above 90%. This is reproduced by experiments (E3,4) (corresponding to Section 8 of the paper).*

A.4.2 Experiments

(E1): *[Full Internet-scale topology simulations]*
How to: Please see the README of the `pki-topology-simulator` submodule for instructions.
Approximate runtime: 192 CPU hours.

(E2): *[DNS full-graph resolution of Let's Encrypt domain names]*
How to: Please see the README of the `routing-aware-dns` repo for instructions and commands.
Approximate runtime: 1.5 hours

(E3): *[Calculation of domain name-level resilience]*
How to: See instructions for `resilience.py` in `princeton-letsencrypt/resilience-computation`. lease see documentation for the `resilience.py` script in the `pki-resilience-processing` submodule.
Approximate runtime: 2 hours

(E4): *[Results analysis]*
How to: Please see documentation for the `interpret_results.py` script in the `pki-resilience-processing` submodule.

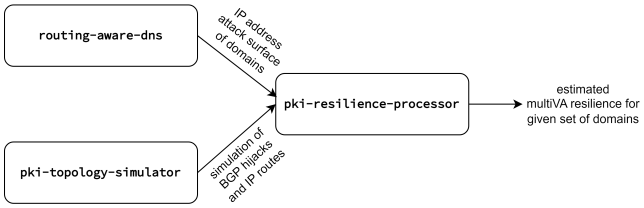

Figure 1: Integration of the artifact submodules.

Approximate runtime: <5 minutes.

A diagram showing the interconnection between the above listed experiments/submodules is given in Figure A.4.2.

A.5 Version

Based on the LaTeX template for Artifact Evaluation V20220926. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2023/.