Exploiting Formal Methods To make The Domain Name System More Robust

Siva Kesava Reddy Kakarla
Department of Computer Science
UCLA
Los Angeles, USA.
sivakesava@cs.ucla.edu

ABSTRACT

The Domain Name System (DNS), one of the foundations of the modern-day Internet, primarily translates domain names into IP addresses. DNS name resolution seems simple at a high level, but has evolved into a complex and intricate protocol over time. Errors in either DNS configurations or DNS implementations have far-reaching disruptive consequences. This is evident from past DNS issues that have rendered popular services such as GitHub, Twitter, HBO, LinkedIn, Yelp, and Azure inaccessible for extended periods.

Formal methods, techniques based on mathematical logic, are used extensively in other computer science areas and have helped improve the robustness of systems. For example, they have been used to improve the robustness of routing by finding bugs in router configurations. Can they be used for other parts of the Internet infrastructure? In this talk we will describe our work towards making the DNS robust via formal methods.

First, I will present GROOT, a new verification tool that we have built that performs exhaustive and proactive static analysis of DNS *configuration files (zone files)* to guarantee key correctness properties. GROOT avoids verifying the huge space of DNS queries by first partitioning all possible queries into equivalence classes (ECs), each of which captures a distinct behavior. GROOT then symbolically executes the set of queries in each equivalence class to efficiently find (or prove the absence of) any bugs such as rewrite loops. We present a mathematical formalization that allows for automatically verifying DNS configurations and detecting any misconfiguration. We applied GROOT to the configuration files we obtained from a large campus network which has over a hundred thousand records, and it revealed 109 new bugs and completed in under 10 seconds. When applied to internal zone files consisting of over 3.5 million records from a large infrastructure service provider, GROOT revealed around 160k issues of blackholing, which initiated a cleanup of the zone files. GROOT was joint work with Ryan Beckett and Behnaz Arzani from Microsoft Research and Todd Millstein and George Varghese at UCLA.

Next, I will describe our experience building a tool called FERRET, which uses a formal model of the DNS to automatically generate high-coverage test suites for DNS software *nameserver implementations*. Today, developers use an ad hoc collection of regression tests they authored to test the implementations for crashes, RFC deviations and also to compare with other implementations. Writing regression tests manually is an onerous task and is highly incomplete. We will present a systematic and principled approach that automatically generates high-coverage test suites. Using FERRET we have identified 30 new bugs in 8 popular open-source DNS implementations, including 3 previously unknown critical security vulnerabilities. One of these was a new vulnerability in Bind that attackers could remotely exploit to crash DNS resolvers and nameservers. Bind released a patch and a high-severity CVE-2021-25215 as a result. FERRET is joint work with Ryan Beckett from Microsoft, and Todd Millstein and George Varghese at UCLA.

In summary, GROOT uses formal methods to *verify DNS zone files*, while FERRET uses formal methods to *generate tests for name server implementations*.

Siva Kesava Reddy Kakarla Siva Kakarla is a final year Ph.D. student in the Computer Science department at UCLA, advised by Todd Millstein and George Varghese. He received his undergraduate degree in Computer Science & Engineering from IIT Kharagpur in 2017. Siva won the best student paper award at SIGCOMM 2020 and was a Facebook 2021 Ph.D. fellowship finalist. Siva is also a recipient of the UCLA Graduate Dean's Scholar Award and UCLA Dissertation-Year Fellowship. During his Ph.D., he interned with the three major cloud providers, Microsoft, Google, and Amazon. His research interests lie at the intersection of networks and programming languages. He worked on finding network (router) misconfigurations by automatic template inference that resulted in a tool called SelfStarter that is used in Microsoft. His current focus is on using formal methods to improve the robustness of DNS.