Stateful Network Protocol Analysis and Symbolic Execution with Rule Based Specification

M. Muthupandi
M.E Software Engineering
University College of engineering
(BIT campus)
Trichirappalli
India.

Mrs. S. ChitraDevi. Ass.Prof
Department of Information Technology
University College of engineering
(BIT campus) Trichirappalli
India.

P. Raja Vishnu Prabhakaran
M.E Computer Science And Engineering
K.L.N College Of EngineeringPottapalayam,
India.

***Abstract*-Usage of system conventions, DNS, DHCP and Zeroconf, are inclined to blemishes, security vulnerabilities furthermore interoperability issues brought on by engineer botches and questionable necessities in convention particulars. Distinguishing such issues is not simple in light of the fact that (i) numerous bugs show themselves just after drawn out operation; (ii) thinking about semantic lapses obliges a machine-clear determination; and (iii) the state space of complex convention usage is substantial. The center thought behind our methodology is to (1) consequently produce high-scope test data bundles for a system convention execution utilizing single- and multi-parcel trade typical execution (focusing on stateless and stateful conventions, individually) and afterward (2) utilize these bundles to discover potential infringement of manual principles gotten from the convention particular, and check the interoperability of distinctive executions of the same system convention. We introduce a framework focused around these strategies, SYMBEXNET, and assess it on numerous executions of two system conventions: Zeroconf, an administration revelation convention, and DHCP, a system design convention. SYMBEXNET has the capacity find non-unimportant bugs and in addition interoperability issues, the majority of which have been affirmed by the engineers.**

Keywords: Symbolic execution, Rule based specification, Interoperability testing, Automated behavioral testing

I. INTRODUCTION

The implement of testing network protocol and complex when the approach that join typical execution a system examination method that can produce inputs that investigate different ways in a system with rule based particulars to check consequently a system convention execution against its determination and find different sorts of blunders, which would be hard to distinguish physically. SYMBEXNET takes as enter the C source code of a system convention execution and a set of guidelines, which characterize right and mistaken conduct.

Designers get administers physically from the convention detail and express them in an abnormal state parcel stream dialect, which states invalid examples in the grouping of bundles traded between a customer and a server. The dialect licenses tenets to allude to parcel header fields and their relationship inside a parcel stream. Guidelines can be concentrated effortlessly from RFC system convention particulars, in this manner encoding the externally visible conduct of a system convention regarding enter and yield parcels. Utilizing typical execution, SYMBEXNET creates a thorough set of info parcels that attain a wide and profound investigation of the project state space. To scale up to vast convention usage, SYMBEXNET blends cement and typical execution: for an expansive investigation, it considers thusly all blends of fields as typical, running every blend for an altered measure of time. For a investigation, keeping in mind the end goal to recognize slips that require complex parcel trades to achieve a given system convention state, SYMBEXNET over and over performs typical execution with extra parcels sent in numerous rounds.

SYMBEXNET then locally executes the usage on all produced test bundles and checks whether the usage effectively handles them as indicated by the detail standards. Moreover, SYMBEXNET can utilize the created test info bundles to check the interoperability of distinctive usage of the same system convention. After a set of test info bundles was created from a system usage, SYMBEXNET executes the bundles against all accessible usage and reports any distinctions as potential slips. The experimentally assess a model usage of SYMBEXNET with numerous system convention

executions. The capacity produce high coverage test data parcels and recognize low-level lapses prompting accidents. It discover hard-to-locate slips that lead to wrong convention conduct, for example, creating unintended reaction parcels for test inputs, by utilizing the standards got from the convention details. Our trials likewise uncover that numerous executions of the same convention can carry on contrastingly bringing about interoperability issues.

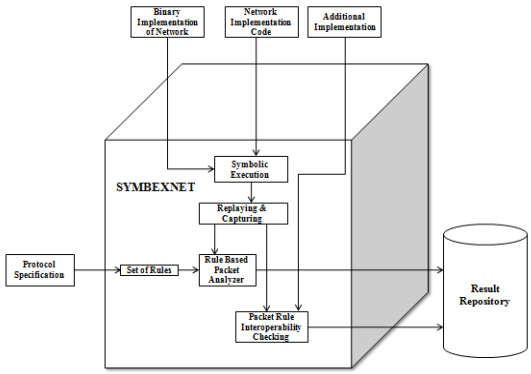
II. RELATED WORK

Existing works are analyzed for detecting the symbolic execution is to use symbolic values as input, as a replacement for of actual data, and to signify values of program variables like symbolic language. Since the program is executed, every statement that based on the symbolic inputs with symbolic constraints. When process reaches a branch to depend on the symbolic input, both probable paths are followed individually, adding together the constraint that the branch state is correct or incorrect, in that order. Two combinations of information are associated through both explored path: a path condition (PC) and a symbolic map (SM) The symbolic map associates symbolic input values among program variables, and the path condition is a first-order quantifier. As the program terminate or encounter an error, the current path condition can be resolved for actual values, which determine a test case to follow that correct path. Several classes of errors, for instance functional correctnessbugs, are hard to detect without executing a part ofsystem. The significance of such testing mutual withthe complexity and poor performance of manual and random approaches has lead to a latest work in using symbolic executionto mechanically generate test input packets.

Symbolic execution can mechanically achieve low-level generic errors. To detect semantic errors so, this can be used to build network protocol specifications. An important process in using SYMBEXNET is to transform a standard network protocol specification. Which describe behavioral violations among a rule-based language that match wrong sequences of input packets. This permits developers to recognize and exact errors of migration and conversion among different implementations of a rule specification.

III. PROPOSED WORK

An interoperability testing technique for testing input network protocol implementations of the similar network protocol. The process of the interoperability is to develop interoperability rules testing. Requirement, which evaluate response input packets from special implementations.



The creation of test packets to check the interoperability among two implementations, interoperability relies on a set of large coverage test input packets that can discover interoperability problems. These packets are produce by applying Single and Multi Packet Exchange Symbolic Execution (SPE-SE) and (MPE-SE) to both implementation. The test input packets generated for each implementation are replayed on both implementations. And all of the exchanged input and output packets are recorded. The recorded packets are compared using the interoperability testing rules from IOT reports a probable interoperability error.

IV. CONCLUSION

This paper described a new rational approach for testing network protocol implementations. use packet rules consequential from protocol specifications and input test packets generate using symbolic execution to determine violations in real-world network protocol implementations with interoperability problems. Discover complex sequence of packet exchanges, the system use an exploration technique is multi packet exchange protocol that frequently performs symbolic execution on selected test packets. These results show that the test interoperability, it observe behavioral difference between implementations of the same network protocol.

V. REFERENCES

[1] S. Alexander and R. Droms, "IETF RFC 2132: DHCP Options and BOOTP Vendor Extensions," Mar. 1997. [Online].Available: http://www.ietf.org/rfc/rfc2132.txt

[2] R. Alur and B.-Y. Wang, "Verifying network protocol implementations by symbolic refinement checking," in Proc. Of the 13th International Conference on Computer Aided Verification(CAV 2001), 2001, pp. 169–181.

[3] S. Avallone, S. Guadagno, D. Emma, A. Pescape, and G. Ventre, "D-ITG distributed internet traffic generator," in Proc. Ofthe 1st International Conference on Quantitative Evaluation (QEST2004), 2004, pp. 316–317.
[4] S. Bradner, "IETF RFC 2026: The Internet Standards Process – Revision 3," Oct. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc2026.txt
[5] "IETF RFC 2119: Key Words for Use in RFCs toIndicate Requirement Levels," Mar. 1997. [Online].Available:http://www.ietf.org/rfc/rfc2119.txt
[6] L. Brenna, A. Demers, J. Gehrke, M. Hong, J. Ossher, B. Panda,M. Riedewald, M. Thatte, and W. White, "Cayuga: a high performance event processing engine," in Proc. of the ACMSIGMOD International Conference on Management of Data (SIGMOD 2007), 2007, pp. 1100–1102.
[7] C. Cadar, D. Dunbar, and D. Engler, "KLEE: unassisted and automatic generation of high-coverage tests for complex systems programs," in Proc. of the 8th USENIX Conference on Operating Systems Design and Implementation (OSDI 2008), 2008, pp. 209–224.
[8] C. Cadar, V. Ganesh, P. M. Pawlowski, D. L. Dill, and D. R.Engler, "EXE: Automatically Generating Inputs of Death," inProc. of the 13th ACM Conference on Computer and Communications Security, 2006, pp. 322–335.
[9] C. Cadar, P. Godefroid, S. Khurshid, C. S. P as areanu, K. Sen,N. Tillmann, and W. Visser, "Symbolic execution for software testing in practice: preliminary assessment," in Proc. of the 33rdInternational Conference on Software Engineering (ICSE 2011),2011, pp. 1066–1071.
[10] S. Cheshire and M. Krochmal, "IETF Internet Draft:Multi cast DNS," Mar. 2010. [Online]. Available: http://files.multicastdns.org/