ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

A Review on: Automatic Test Packet Generation

Bhosale Uday M., Prof. Amrit Priyadarshi

P.G. Scholar, Dept. of Computer Engineering. DGOI, FOE, Bhigwan, Savitribai Phule University of Pune, Pune, India

Professor, Dept. of Computer Engineering, DGOI, FOE, Bhigwan, Savitribai Phule University of Pune, Pune, India

ABSTRACT: Recently networks are growing wide and more complex. However administrators use tools like ping and trace route to debugproblems. Hence we proposed an automatic and Methodical approach for testing and debugging networks called Automatic Test PacketGeneration (ATPG). This approach gets router configurations and generates a device-independent model. ATPG generate a few set of testpackets to find every link in the network. Test packets are forwarded frequently and it detect fail lures to localize the fault. ATPG can detect bothfunctional and performance (throughput, latency) problems. We found, less number of test packets is enough to test all rules in networks. Forexample, 4000 packets can cover all rules in Stanford backbone network, while 53 are much enough to cover all links.

KEYWORDS*:* Fault Localization, Test Packet Selection, Network Debugging, Automatic Test packet Generation (ATPG), Forwarding InformationBase (FIB).

I.INTRODUCTION

Thepopularly known us, very difficult to troubleshoot oridentify and remove errors in networks. Every day, networkengineers fight with mislabeled cables, software bugs, routermisconfigurations, fiber cuts, faulty interfaces and otherreasons that cause networks to drop down. Networkengineers hunt down bugs with various tools (e.g., Ping,trace route, SNMP) and track down the reason for networkfailure using a combination of accrued wisdom andimpression. Debugging networks is becoming moreharderas networks are growing larger (modern data centers maycontain 10 000 switches, a campus network may serve 50000 users, a 100-Gb/s long-haul link may carry 100 000flows) and are getting complicated (with over 6000 RFCs,router software was based on millions of lines of sourcecode, and network chips contain billions of gates.

In current system, the administrator manually decides which ping packet to be sent. Sending programs between everypair of edge ports is neither extensive nor scalable. Thissystem is enough to find minimum set of end-to-end packetsthat travel each link. However, doing this need a way ofabstracting across device specific configuration filesgenerating headers and links they reach and finallycalculating a minimum set of test packets. It is not designedto identify failures caused from failed links and routers, bugs caused from faulty router hardware or software, andperformance problems. The common causes of network failure are hardware failures and software bugs, in whichthat problems manifest both as reachability failures andthroughput/latency degradation. To overcome this we areproposing new system.

II.LITERATURE SURVEY

1**. Modelling botnet propagation using time zones.**

D. Dagon, C. Zou, andW. Leeis Presented by Time zones play an important and unexplored role in malware epidemics. To understand how time and location affect malware spread dynamics, we studied botnets, or large coordinated collections of victim machines (zombies) controlled by attackers. Over a six month period we observed dozens of botnets representing millions of victims. We noted diurnal properties in botnet activity, which we suspect occurs because victims turn their computers off at night. Through binary analysis, we also confirmed that some botnets demonstrated a bias in infecting regional populations. Clearly, computers that are offline are not infectious, and any regional bias in infections will affect the overall growth of the botnet. We therefore created a diurnal propagation model. The model uses diurnal shaping functions to capture regional variations in online vulnerable populations. The

DOI: 10.15680/IJIRCCE.2015. 0311250

diurnal model also lets one compare propagation rates for different botnets, and prioritize response. Because of variations in release times and diurnal shaping functions particular to an infection, botnets released later in time may actually surpass other botnets that have an advanced start. Since response times for malware outbreaks is now measured in hours, being able to predict short-term propagation dynamics lets us allocate resources more intelligently. We used empirical data from botnets to evaluate the analytical model.

2. **Dissecting android malware: Characterization and evolution**

Y. Zhou and X. Jiangis presented the popularity and adoption of smart phones has greatly stimulated the spread of mobile malware, especially on the popular platforms such as Android. In light of their rapid growth, there is a pressing need to develop effective solutions. However, our defence capability is largely constrained by the limited understanding of these emerging mobile malware and the lack of timely access to related samples. In this paper, we focus on the Android platform and aim to systematize or characterize existing Android malware. Particularly, with more than one year effort, we have managed to collect more than 1,200 malware samples that cover the majority of existing Android malware families, ranging from their debut in August 2010 to recent ones in October 2011. In addition, we systematically characterize them from various aspects, including their installation methods, activation mechanisms as well as the nature of carried malicious payloads. The characterization and a subsequent evolution-based study of representative families reveal that they are evolving rapidly to circumvent the detection from existing mobile anti-virus software. Based on the evaluation with four representative mobile security software, our experiments show that the best case detects 79.6% of them while the worst case detects only 20.2% in our dataset. These results clearly call for the need to better develop next-generation anti-mobile-malware solutions.

3. **Protecting against network infections: A game theoretic perspective**

J. Omic, A. Orda, and P. V. Mieghemthe presented Security breaches and attacks are critical problems in today's networking. A key-point is that the security of each host depends not only on the protection strategies it chooses to adopt but also on those chosen by other hosts in the network. The spread of Internet worms and viruses is only one example. This class of problems has two aspects. First, it deals with epidemic processes, and as such calls for the employment of epidemic theory. Second, the distributed and autonomous nature of decision-making in major classes of networks (e.g., P2P, ad- hoc, and most notably the Internet) call for the employment of game theoretical approaches. Accordingly, we propose a unified framework that combines the N-intertwined, SIS epidemic model with a no cooperative game model. We determine the existence of Nash equilibrium of the respective game and characterize its properties. We show that its quality, in terms of overall network security, largely depends on the underlying topology. We then provide a bound on the level of system inefficiency due to the no cooperative behaviour, namely, the "price of anarchy" of the game. We observe that the price of anarchy may be prohibitively high; hence we propose a scheme for steering users towards socially efficient behaviour.

4. **Power laws, pareto distributions and zipf's law**

M. E. J. Newman is presented the when the probability of measuring a particular value of some quantity varies inversely as a power of that value, the quantity is said to follow a power law, also known variously as Zipf's law or the Pareto distribution. Power laws appear widely in physics, biology, earth and planetary sciences, economics and finance, computer science, demography and the social sciences. For instance, the distributions of the sizes of cities, earthquakes, solar flares, moon craters, wars and people's personal fortunes all appear to follow power laws. The origin of power-law behaviour has been a topic of debate in the scientific community for more than a century. Here we review some of the empirical evidence for the existence of power-law forms and the theories proposed to explain them.

5. **The effect of network topology on the spread of epidemics**

A. J. Ganesh, L. Massouli´e, and D. F. Towsleyis presented the Many network phenomena are well modelled as spreads of epidemics through a network. Prominent examples include the spread of worms and email viruses, and, more generally, faults. Many types of information dissemination can also be modelled as spreads of epidemics. In this paper we address the question of what makes an epidemic either weak or potent. More precisely, we identify topological properties of the graph that determine the persistence of epidemics. In particular, we show that if the ratio of cure to infection rates is larger than the spectral radius of the graph, then the mean epidemic lifetime is of order log n, where n is the number of nodes. Conversely, if this ratio is smaller than a generalization of the isoperimetric constant of the graph, then the mean epidemic lifetime is of order ena, for a positive constant a. We apply these results to several network topologies including the hypercube, which is a representative connectivity graph for a distributed hash table, the complete graph, which is an important connectivity graph for BGP, and the power law graph, of which the AS-level Internet graph is a prime example. We also study the star topology and the Erdos-Renyi graph as their epidemic spreading behaviours determine the spreading behaviour of power law graph.

III.EXISTING SYSTEM

Testing liveness of a network is a fundamental problem for ISPs and large data centre operators. Sending probes between every pair of edge ports is neither exhaustive nor scalable. It suffices to find a minimal set of end-to-end packets that traverse each link. However, doing this requires a way of abstracting across device specific configuration files, generating headers and the links they reach, and finally determining a minimum set of test packets (Min-Set-Cover). To check enforcing consistency between policy and the configuration.

Disadvantages:

1) Not designed to identify liveness failures, bugs router hardware or software, or performance problems.
2) The two most common causes of network failure are hardware failures and software bugs, and that problems manifest themselves both as reachability failures and throughput/latency degradation.

IV.PROPOSED ALGORITHM

System Architecture:

ATPG framework generates minimum set of packets automatically, to debug the failures occurring in thenetwork. This tool could automatically generate packets forchecking performance assertions such as like packet latency.ATPG finds and determines errors by independently testingall forwarding entries, any packet processing rules andfirewall rules in network. Here, test packets are generatedalgorithmically from device configuration files and fromFIBs, which requires minimum number of packets forcomplete coverage.Test packets are fed into the network in which that everyrule is covered directly from the data plane. Since ATPG treats links like normal forwarding rules, its full coverageprovides testing of every link in the network. It can also bespecialized to form a minimal set of packets that obviouslytest every link for network liveness. At least in this basicform, we would feel that ATPG or some similar technique isfundamental to networks: Instead of reacting to failures,many network operators such as Internet2 proactively check the health of their network using pings between all pairs of sources. However, all-pairs does not provide testing of all links and has been found to be unsalable for large networks such as Planet Lab.

DOI: 10.15680/IJIRCCE.2015. 0311250

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

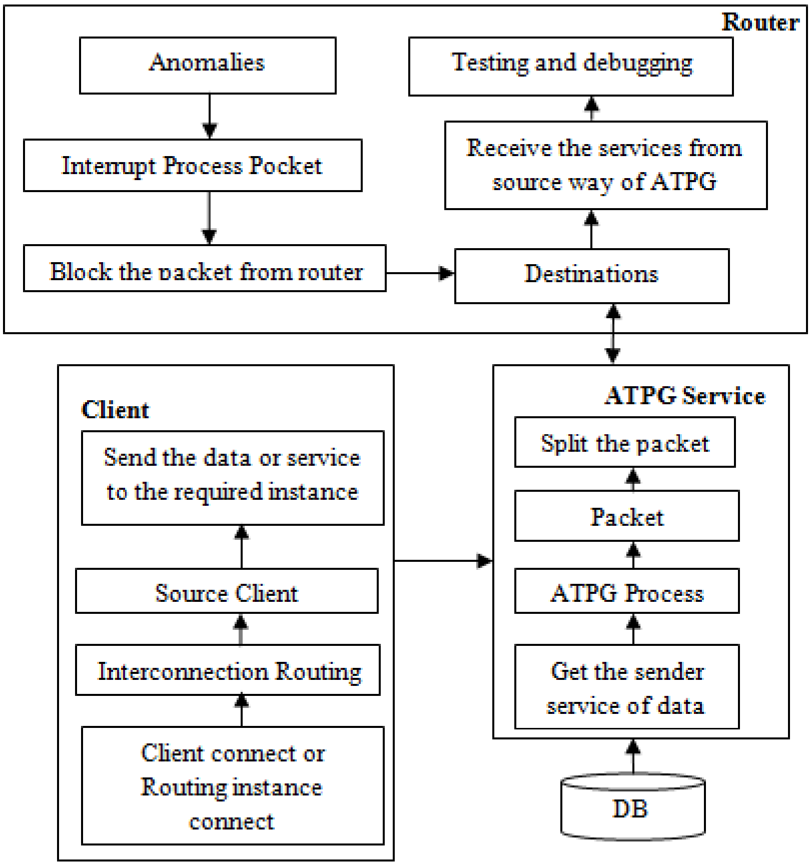
Router
Anomalies
Testing and debugging
Interrupt Process Pocket
Receive the services from source way of ATPG
Block the packet from router
Destinations
Client
Send the data or service to the required instance
Source Client
Interconnection Routing
Client connect or Routing instance connect
ATPG Service
Split the packet
Packet
ATPG Process
Get the sender service of data
DB

Figure 1: ATPG system block diagram.

Modules:

1. **Test Packet Generation:**
We assume a set of test terminals in the network can send and receive test packets. Our goal is to generate a set of test packets to exercise every rule in every switch function, so that any fault will be observed by at least one test packet. This is analogous to software test suites that try to test every possible branch in a program. The broader goal can be limited to testing every link or every queue. When generating test packets, ATPG must respect two key constraints First Port (ATPG must only use test terminals that are available) and Header (ATPG must only use headers that each test terminal is permitted to send).

2. **Generate All-Pairs Reachability Table:**
ATPG starts by computing the complete set of packet headers that can be sent from each test terminal to every other test terminal. For each such header, ATPG finds the complete set of rules it exercises along the path. To do so, ATPG applies the all-pairs reachability algorithm described. On every terminal port, an all- header (a header that has all wild carded bits) is applied to the transfer function of the first switch connected to each test terminal. Header constraints are applied here.

3. **ATPG Tool:**
ATPG generates the minimal number of test packets so that every forwarding rule in the network is exercised and covered by at least one test packet. When an error is detected, ATPG uses a fault localization algorithm to determine the failing rules or links.

4. **Fault Localization:**
   ATPG periodically sends a set of test packets. If test packets fail, ATPG pinpoints the fault(s) that caused the problem. A rule fails if its observed behaviour differs from its expected behaviour. ATPG keeps track of where rules fail using a result function "Success" and "failure" depend on the nature of the rule: A forwarding rule fails if a test packet is not delivered to the intended output port, whereas a drop rule behaves correctly when packets are dropped. Similarly, a link failure is a failure of a forwarding rule in the topology function. On the other hand, if an output link is congested, failure is captured by the latency of a test packet going above a threshold.

Advantage:

a. A survey of network operators revealing common failures and root causes.
b. A test packet generation algorithm.
c. A fault localization algorithm to isolate faulty devices and rules.
d. ATPG use cases for functional and performance testing.
e. Evaluation of a prototype ATPG system using rule sets collected from the Stanford and Internet2 backbones.

V.Conclusion and Future Work

In this paper, In current System it uses a method that is neither exhaustivenor scalable. Though it reaches all pairs of edge nodes itcould not detect faults in liveness properties. ATPG goesmuch further than liveness testing with same framework.ATPG could test for reachability policy (by checking allrules including drop rules) and performance measure (byassociating performance measures such as latency and lossof test packets). Our implementation also enlarges testingwith simple fault localization scheme also build usingheader space framework

References

[1] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: Analysis of a botnet takeover," in CCS '09: Proceedings of the 2009 ACM conference on computer communication security, 2009.
[2] D. Dagon, C. Zou, andW. Lee, "Modeling botnet propagation using time zones," in Proceedings of the 13 th Network and Distributed System Security Symposium NDSS, 2006.
[3] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging," in Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, 2007.
[4] D. Dagon, C. C. Zou, and W. Lee, "Modeling botnet propagation using time zones," in NDSS, 2006.
[5] P. V. Mieghem, J. Omic, and R. Kooij, "Virus spread in networks," IEEE/ACM Transactions on Networking, vol. 17, no. 1, pp. 1–14, 2009.
[6] Cabir, http://www.f-secure.com/en/web/labs global/2004- threat-summary.
[7] Ikee, http://www.f-secure.com/vdescs/worm iphoneosikee b.shtml.
[8] Brador, http://www.f-secure.com/v-descs/brador.shtml.
[9] S. Peng, S. Yu, and A. Yang, "Smartphone malware and its propagation modeling: A survey," IEEE Communications Surveys and Tutorials, in press, 2014.
[10] Z. Chen and C. Ji, "An information-theoretic view of network-aware malware attacks," IEEE Transactions on Information Forensics and Security, vol. 4, no. 3, pp. 530–541, 2009.
[11] A. M. Jeffrey, xiaohua Xia, and I. K. Craig, "When to initiate hiv therapy: A control theoretic approach," IEEE Transactions on Biomedical Engineering, vol. 50, no. 11, pp. 1213–1220, 2003.
[12] R. Dantu, J.W. Cangussu, and S. Patwardhan, "Fast worm containment using feedback control," IEEE Transactions on Dependable and Secure Computing, vol. 4, no. 2, pp. 119–136, 2007.
[13] S. H. Sellke, N. B. Shroff, and S. Bagchi, "Modeling and automated containment of worms," IEEE Trans. Dependable Sec. Comput., vol. 5, no. 2, pp. 71–86, 2008.
[14] P. De, Y. Liu, and S. K. Das, "An epidemic theoretic framework for vulnerability analysis of broadcast protocols in wireless sensor networks," IEEE Trans. Mob. Comput., vol. 8, no. 3, pp. 413–425, 2009.
[15] G. Yan and S. Eidenbenz, "Modeling propagation dynamics of bluetooth worms (extended version)," IEEE Trans. Mob.Comput., vol. 8, no. 3, pp. 353–368, 2009.
[16] C. C. Zou, W. Gong, D. Towsley, and L. Gao, "The monitoring and early detection of internet worms," IEEE/ACM Trans. Netw., vol. 13, no. 5, pp. 961–974, 2005.
[17] C. Gao and J. Liu, "Modeling and restraining mobile virus propagation," IEEE Trans. Mob. Comput., vol. 12, no. 3, pp. 529–541, 2013.
[18] D. J. Daley and J. Gani, Epidemic Modelling: An Introduction. Cambridge University, 1999.
[19] W. Willinger, D. Alderson, and J. C. Doyle, "Mathematics and the internet: A source of enormous confusion and great potential," Notices of the Ameriacan Mathematical Socieity, vol. 56, no. 5, pp. 586–599, 2009.

[20] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," in IEEE Symposium on Security and Privacy, 2012, pp. 95–109.
[21] S. Shin, G. Gu, A. L. N. Reddy, and C. P. Lee, "A largescale empirical study of conficker," IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 676–690, 2012.

[22] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, “A multifaceted approach to understanding the botnet phenomenon,” in Internet Measurement Conference, 2006, pp. 41–52.
[23] A. J. Ganesh, L. Massouli´e, and D. F. Towsley, “The effect of network topology on the spread of epidemics,” in INFOCOM, 2005, pp. 1455–1466.
[24] J. Omic, A. Orda, and P. V. Mieghem, “Protecting against network infections: A game theoretic perspective,” in INFOCOM’09, 2009.
[25] R. L. Axtell, “Zipf distribution of u.s. firm sizes,” Science, vol. 293, 2001.
[26] M. Mitzenmacher, “A brief history of generative models for power law and lognornal distributions,” Internet Mathematics, vol. 1, 2004.
[27] M. Newman, Networks, An Introduction. Oxford University Press, 2010.
[28] Z. K. Silagadze, “Citations and the zipf-mandelbrot’s law,” Complex Systems, vol. 11, pp. 487–499, 1997.
[29] M. E. J. Newman, “Power laws, pareto distributions and zipf’s law,” Contemporary Physics, vol. 46, pp. 323–351, December 2005.

DOI: 10.15680/IJIRCCE.2015. 0311250