IJIRMET ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 6, June 2017

Firewall Policy Testing And Optimization

[1]K.Rohit, [2]S.Afzal, [3]T.Manoranjitham
[1] [2] [3]Department Of Computer Science Engineering,SRM University
[1]merohitchandra@gmail.com, [2]afzalsheikh002@gmail.com, [3]manoranjitham.t@ktr.srmuniv.ac.in

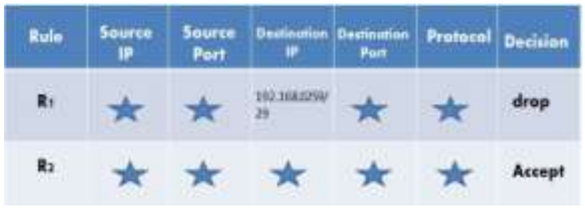
Abstract: ***Firewall is basically a security solution designed so as to protect the LAN(Private Network) and give half access to public users (DMZ).It basically monitors the packets that are going in and out of the network with the intention to block the malicious and undesirable traffic with an assumption that it might harm the network.The Firewall is basically commanded in the form of policies that are basically in the form of input given by the network administrators and henceforth are present and maintained in the rulebase irrespective of whether the firewall is software based or hardware based.To help ensure the effectiveness of the firewall it is necessary to optimize the firewall policy base and to test before deploying the firewall in the network.***

Keywords: ***Firewall Policy,Testing ,Cumulative efficiency.***

I. INTRODUCTION

Before we optimize the policy base of the firewall it is important to know the ill effects that unoptimized policy base can have on the network.As we know that any object oriented firewall follows top to bottom approach when it comes to reading the rules in the policy base,let's say that we have placed stealth rule (any traffic hitting the gateway is dropped) above the admin access rule which states that administrators access to gateway via https ,port no 443 and ssh port no 22,then the intention of admin policy will be overridden by the stealth rule thereby blocking the admins access to the gateway to perform diagnostic operations
.Another example of the ill effects that an unoptimized firewall brings along can be due to redundant policies(mentioning the same policy more than once in the rulebase) and it leads to undesired consumption of RAM and Memory which further leads to drop down in the throughput of the firewall thereby increasing the latency of the packet,due to which firewall fails to leverage its capabilities in terms of performance and hardware.

There is a way by which we can optimize and test the firewall before its deployment in the network.The testing is generally done to determine the efficiency of the firewall and based on which further optimization can be performed. In order to test the efficiency of the firewall test packets have to be generated and the test packets can be generated using tools such as ostinato and capsa colasoft packet builder.In order to further test the efficiency of the firewall it is important to generate normal packet and faulty packets
,packets are generally made faulty by creating malicious payload intentionally or by creating checksum errors intentionally (Integrity Error).Tools such as Bluecoat packet shaper or symantec can be used for faulty packet creation but the drawback would be costing,to avoid costing issues colasoft packet builder can be used to both generate the packet and create faulty packet.



Rule	Source IP	Source Port	Destination IP	Destination Port	Protocol	Decision
R_1	★	★	192.168.0250/ 29	★	★	drop
R_2	★	★	★	★	★	Accept

Fig 1 Sample Firewall Policy

II. EXISTING SYSTEM

The current system talks about how fault can detected by intentionally adding the faulty packets and evaluation of the fault ,which can be done by using the structural coverage mechanism,which includes predicate coverage,rule coverage and clause coverage criterions.

III. PROPOSED SYSTEM

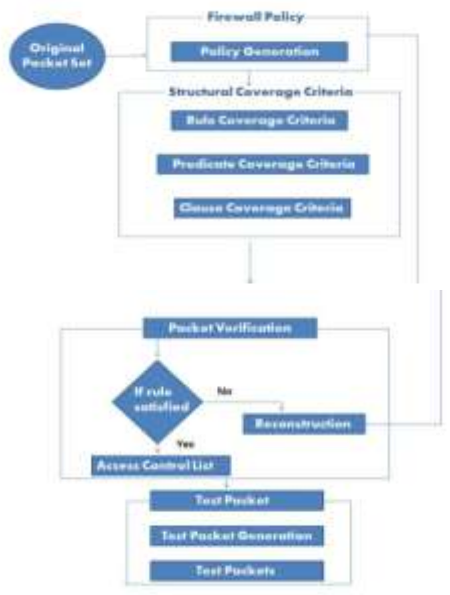
The optimization of the firewall policies is a critical part as 80% of the firewall level faults are due to misconfigurations done by the administrator.The presence of redundant policies can shadow many legitimate policies making them non functional.Hence using the greedy algorithm,which is more of a technique ,the redundant policies can be removed,thereby conserving resources of the firewall to a great extent and enhancing the security.

IV. STRUCTURAL COVERAGE CRITERIA

The structural coverage criteria covers for three possible dimensions of the policy base by using
Rule coverage:
Rule coverage is the coverage of each rule in the firewall policy base at least once in such a way that it is examined to be true at least once.
Predicate Coverage:
Predicate Coverage is the coverage of each rule in the firewall policy base in such a way that it is examined to be true and false at least once,this ensures that decision of each rule is both satisfied and not satisfied at least once. **Clause Coverage:**
Clause coverage is the coverage of each rule in the firewall policy base in such a way that all the fields in a rule are examined to true and false at least once ,this ensures all the fields in each rule are covered.
These 3 criteria for coverage ensure the correct specification of all the elements of the firewall.

V. FAULT DETECTION USING STRUCTURAL COVERAGE

Policy testers can choose any of the above mentioned 4 ways of generating packets to test the rules in a firewall policy base to ensure the correct specification of all the elements of the firewall.Not to get drifted from our main objective which is to ensure the detection of faults in the policy base by maintaining the equivalent coverage criteria.A flaw in a rules outcome is determined only by the coverage of the rule as well as the verification of its outcome.The packet generation set chosen will aid us in detecting such faults easily and increment the resistance of the policy against such flaws.Similarly a packet set with high coverage of structural entities ensures high rate of detection of flaws in the firewall elements.So basically all the dimensions of the firewall policy base are covered to ensure its correctness.



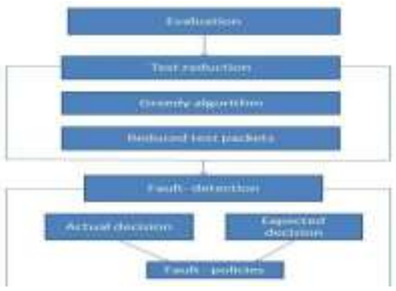


Fig 2 Framework

VI. FRAMEWORK

In this section the framework for testing firewall policy base is presented,figure …. Shows the our brief overview to our approach.Three Phases are included in the framework Packet Generation,Packet Reduction and Fault Detection.
In the Packet Generation Phase critical analysis of the firewall policy base is performed to perform packet generation which ensures the coverage in all the dimensions(Rules,Predicates,Clauses),four different ways of generating packets have been proposed.In the Phase of Test Reduction its component cuts short the number of
packets based on the criteria for coverage by only including the packet that will advocate the increment the coverage of policy measurement during the examination.In the phase of fault detection ,the policy testers perform manual
inspection on the actual decisions and expected decisions.The actual decisions are the resultant decisions of the generated packet against the firewall policy base being tested) .The consistency of the actual decision and the expected decision is put to test.If any inconsistent decisions are found it is concluded that a fault is determined in the policy. A)Packet Generation

As The process of manual generation of packets for testing the firewall policy base is cumbersome,4 ways to automatically generate packets have been proposed,all this is done to ensure the coverage of the firewall policy base in all dimensions for the correctness at the miniscule
level.Test packets are generated for inspecting the correctness for the firewall rules .This section comprises of the 4 ways of packet generation Sending the packets in bulk,minimal fixed generation,maximum fixed generation
,The flexibility of the tester.The main difference between the second approach and the third approach is that between scope wise minimum and maximum constraints.The second and third approach are based on the the values generated randomly within the solutions obtained by solving the conditions,The fourth one generates test packets based on limit values within the range solved by the condition solving.

Sending the packets in bulk: In this mode of packet generation the packets sent to the firewall junction are generated randomly in large numbers,this is a measure of the throughput of the firewall.This approach is fairly straightforward.While this technique does not require the actual rule in firewall policy base itself in the phase of test generation and can generate a large number of packets,it is not effective to achieve the coverage of the firewall policy base in all the dimensions with the packets generated to test the firewall policy base.The randomness of this approach is the cause for the small number of fields covered in comparision to the total fields present in the firewall policy base being tested.

Minimal Fixed Generation : Fixed number of packets are sent to the firewall junction resulting in the calculation of the accuracy and fault ratio.
Ex:10 packets are sent to the firewall junction out of which
5 will be correctly formed packets and 5 will be faulty packets ,lets say 4 packets are dropped out of the expected
5 faulty packets.In this case the Accuracy of the firewall will be: ⅘ The Fault will be ⅕.
In general emphasis is laid on the coverage of fields that have not been covered previously, the effect of the other rules in the policy is not taken into account in this method.The 2 major limitations are brought by this technique ,the packets generated may not succeed to cover the fields due to overlapping (i.e fields that can be satisfied by the same packet).Secondly determination of whether a field could be covered in advance is not possible .This is because some rules in the firewall policy

base are overridden by other rules and are never evaluated.In cases like there is no presence of any prerequisite criteria to make a decision whether additional packets are to be generated or bring the test to a halt.

Maximum Fixed Generation: In the case of maximum fixed generation maximum number of finite packets are sent to the firewall junction and the results of the policy base are evaluated by calculating the Accuracy and Fault Rate of the firewall.
Ex:50 packets are sent to the firewall junction out of which
25 are genuine correctly formed and the rest are malformed
,out of which 23 packets are only blocked .Upon evaluation the Accuracy and Fault Rate of the firewall are found out to be
Accuracy:23/25
Fault Rate:2/25

This way considers the influence of other rules ,covering fields in a rule requires that the evaluation of the preceding rules to be false.This technique is useful for the generation of packets with high coverage capability by considering the effect of the preceding rules of the rule under test ,however this way requires more time due to the complexity involved in solving the conditions than the previous two ways.

Flexibility Of The Tester:(change to be made ,include the limit symbol)
This mode is suitable for testers who require flexibility to choose the number of packets that can be sent.Mathematical Approach is followed by using limits over defined intervals,the lower limit would specify the start point for the minimum value of each field and the
upper limit would specify the maximum value of each field, to be sent which are in accordance with the firewall element's range of values.Upon fault injection the flawed policy behaves within the range limits of a field instead of some other values.More specifically packets are generated based on the range values to cover true and false clause fields.The Accuracy and the fault ratio are calculated accordingly .

Throughout these four modes of Packet Generation The Terms Accuracy And Fault are calculated using the following formula.
Accuracy:Ratio of the actual number of Dropped Packets to the total number of Faulty Packets .
Fault:Number of expected Dropped Packets minus Number of actual Dropped Packets to the total number of faulty packets.
Cumulative Effeciency(CE):After the firewall passess the above mentioned four tests,the Cumulative Efficiency is calculated by summing the efficiencies of the 4 ways of packet generation.
CE=(Efficiency of Way 1+Efficiency of Way 2+Efficiency of Way 3+Efficiency of Way 4)/4
Ex:
After the fw pass these 4 tests ,cumulative efficiency(CE) will be based on fault of way 1 +fault of way2 +fault of way 3 +fault of way 4 =(20+20+15+8)/4=13.25

B)TEST REDUCTION

It is cumbersome for testers of the firewall policy base to perform manual inspection on a set of packets ,which is a packet with its evaluated decision.Hence reducing the size of the set of packets for inspection is of utmost importance
,however without compromising on the capacity to detect flaws.The packet set size can be reduced without altering its level of dimensional coverage.
Each set of packets is evaluated against the firewall policy base,A Greedy Algorithm is used that advocates the removal of a packet from a set of packets provided the packet will not increment the coverage metrics achieved by the previous packets evaluated in set.

C)FAULT DETECTION CAPABILITY

The emphasis of any process under test is to detect faults
,in this paper we propose to dig deep into the relation between the dimensional coverage that a packet set achieves upon testing the firewall policy base and the packets ability to detect flaws.Mutation Testing is applied for the measurement of the capacity of the packet to detect faults.

Mutation Testing on policies,flaws injection is performed on the unaltered firewall policy base and its flawed version is created.These flaws of injection can be something as simple as an not correct decision of a rule to something complex such as configuration errors involving various rules.Mutation Testing is built on the understanding that if
a policy is flawed then an equivalent set of altered policies can be put to an end if only and if the packet generated is also capable of detecting that flaw.The number of altered policies generated are equal to the rule count or the clause count depending on the nature of alteration.
Upon evaluations of the same packet on the original policy and its altered version ,if different decisions are produced then the packet is capable for detecting the flaw in the altered version of policy in the firewall policy base.It is to be noted that a change in the syntax of the policy cannot assure a semantic change ,what this means is a policy can
have two altered versions which function similarly,in which case the packet cannot be put to an end .So a change in semantic of the policy under test should be done,an existing tool which analyzes the potential impact brought out by a change in the policy in the firewall policy base ,the purpose of this tool is to calculate the repercussions to look out for a change in the functionality ,so this tool would analyze the differences between similar altered versions of the original policy in the firewall policy base.

VII. IMPLEMENTATION

The codebase is built on top of Java,comprises of four modules Generation of packets ,Evaluation of packets,Reduction of packets ,Generation of altered versions of the original policy.For generation of packets based on minimal fixed generation the packet generator selects random values within the field range.For packet generation based on Maximum fixed generation a Theorem Prover called Z3 was utilized,what this does is finds numeric solutions which are translated to packets,if a solution does not exist then the output is
unsolvable.Generation of packets based on the flexibility of the tester the packet builder selects values which are within the range of each field,redundancy in the firewall policy base was removed by utilizing an existing tool.

Packet evaluation a engine to emulate evaluation of packet by firewall policy in the firewall policy base.Rules are parsed and stored in the form of a List.When a packet is being evaluated,the simulation searched for the first rule that can be applied and the rules evaluated decision is the output (accept or drop),comparison of evaluated decisions is taken care of as well.

VIII. EXPERIMENTS

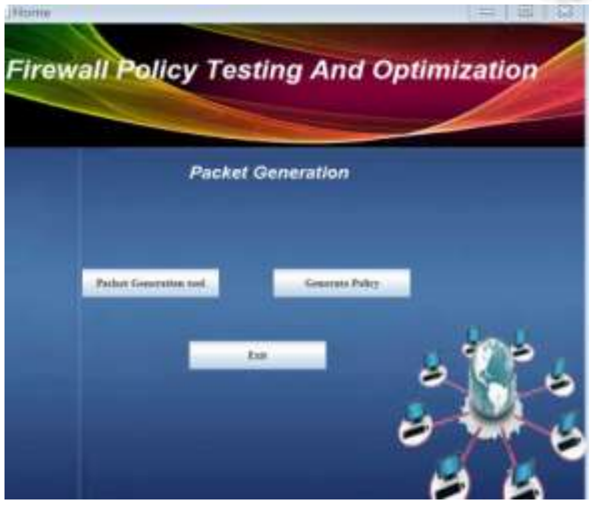
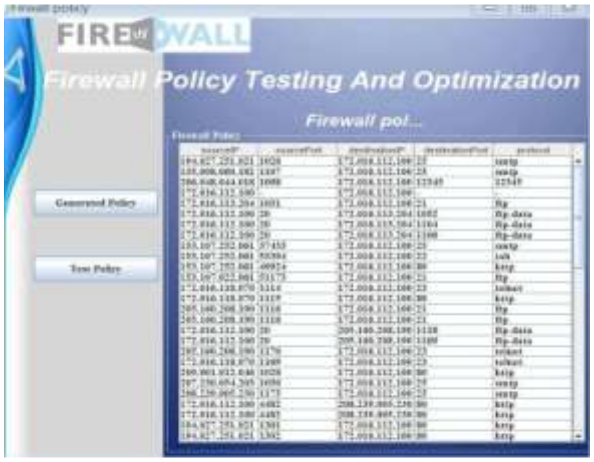
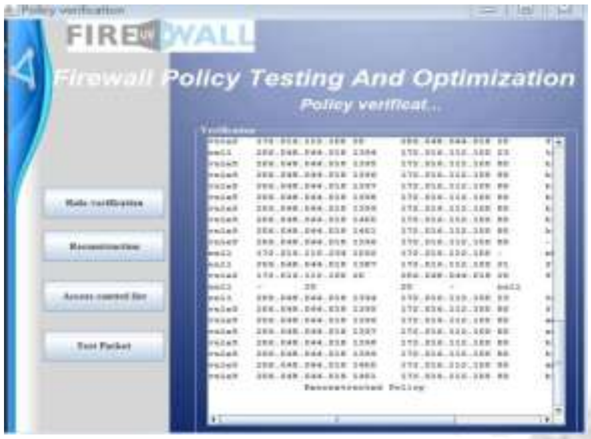


Fig 3 Firewall Policy Testing And Optimization



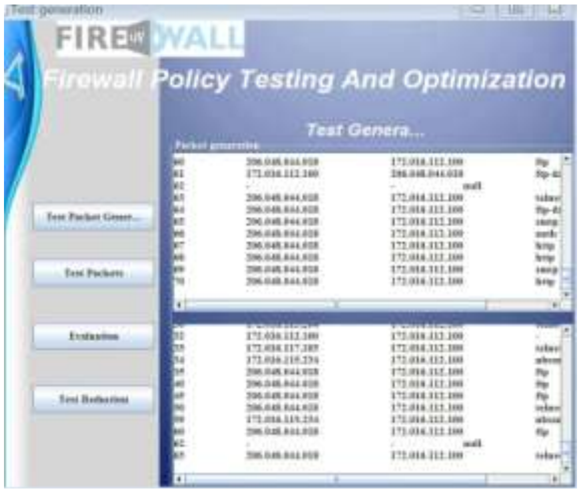
[illegible]

Fig 4 Firewall Policy Generation And Testing



[illegible]

Fig 5 Policy Verification



[illegible]

Fig 6 Test Generation

[illegible]

Fig 7 Test Reduction

[illegible]

Fig 8 Fault Detection

IX. CONCLUSION

We have devised techniques to optimize policies in the firewall policy base.Three different structural coverage criteria(Rules,Clauses,Predicates) have been defined and Four ways to generate packets have been developed sending packets in bulk,minimal fixed generation,maximum fixed generation,flexibility of the tester.In the future we intend to convert this into a framework so that it can be leveraged and integrated into the firewalls .

REFERENCES

[1] S. W. Lodin and C. L. Schuba, “Firewalls fend off invasions from the net,” IEEE Spectrum, vol. 35, no. 2, pp. 26–34, 1998.
[2] A. Wool, “A quantitative study of firewall configuration errors,” Computer, vol. 37, no. 6, pp. 62–67, 2004.
[3] E. Al-Shaer and H. Hamed, “Discovery of policy anomalies in distributed firewalls,” in Proc. 2004 IEEE Conf. on Communications, pp. 2605–2616.
[4] L. Yuan, J. Mai, Z. Su, H. Chen, C.-N. Chuah, and P. Mohapatra, “FIREMAN: a toolkit for FIREwall Modeling and ANalysis,” in Proc. 2006 IEEE Symposium on Security and Privacy, pp. 199–213.
[5] M. R. Lyu and L. K. Y. Lau, “Firewall security:
policies, testing and performance evaluation,” in Proc.
2000 International

Conference on Computer Systems and Applications, pp. 116–121.
[6] A. X. Liu, M. G. Gouda, H. H. Ma, and A. H. Ngu, "Non-intrusive testing of firewalls," in Proc. 2004 International
Computer Engineering Conference, pp. 196–201. [7] D. Hoffman and K. Yoo, "Blowtorch: a framework for firewall test automation," in Proc. 2005 IEEE/ACM international

Conference on Automated Software Engineering, pp.
96–103.
[8] H. Zhu, P. A. V. Hall, and J. H. R. May, "Software unit test coverage and adequacy," ACM Comput. Surv., vol.
29, no. 4, pp.
366–427, 1997.
[9] R. A. DeMillo, R. J. Lipton, and F. G. Sayward, "Hints on test data selection: help for the practicing programmer," IEEE
Computer, vol. 11, no. 4, pp. 34–41, 1978.
[10] P. Ammann, J. Offutt, and H. Huang, "Coverage criteria for logical expressions," in Proc. 2003
International Symposium on
Software Reliability Engineering, pp. 99–107.
[11] A. X. Liu and M. G. Gouda, "Complete redundancy detection in firewalls," in Proc 2005 Annual IFIP Conference on Data
and Applications Security, pp. 196–209. [12] E. Martin, T. Xie, and T. Yu, "Defining and measuring policy coverage in testing access control policies," in Proc. 2006 International Conference on Information and Communications Security, pp. 139–158.