ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

Firewall policy anomaly detection and resolution

Ms. R.V.Darade, Prof P.B. Kumbharkar
Department of Computer Engineering, SCOE,Sudumbare,Pune

Abstract: Security of all private networks in businesses and institutions is achieved by firewall. Firewall provides protection by the quality of policy configured. Lack of Systematic analysis mechanism and Tools, Complex firewall configuration makes designing and managing firewall policies difficult. With help of segmentation rule, anomaly management framework is designed for accurate detection and effective resolution of anomalies. Using this technique, packets of network can be divided into set of disjoint packet space segments. Every segment is associated with unique set of firewall rules which specify an overlap relation among all firewall rules which could be conflicting or redundant. Flexible conflict resolution method is provided which has many resolution strategies for risk assessment of protected networks and its policy definition. Firewall logs are maintained by using association rule mining on these logs to find frequent logs, which in turned filtered to find malicious packets. Apriori algorithm is used to find frequent element from above logs. In each round, it computes the support for all candidate-item-sets. Candidate-item-sets with frequency above the minimum support parameter are selected at the end of each round; these frequent item-sets of round are used in the next round to construct candidate -item-sets. The algorithm halts when item-sets with desired frequency not found .

Index Terms— Anomaly management, Firewall, policy firewall log.

I. INTRODUCTION

AS one of essential elements in network and information system security, firewalls have been widely deployed in defending suspicious traffic and unauthorized access to Internet-based enterprises. Sitting on the border between a private network and the public Internet, a firewall examines all incoming and outgoing packets based on security rules. To implement a security policy in a firewall, system administrators define a set of filtering rules that are derived from the organizational network security requirements. Firewall policy management is a challenging task due to the complexity and interdependency of policy rules. This is further exacerbated by the continuous evolution of network and system environments. For instance, Al-Shaer and Hamed reported that their firewall policies contain anomalies even though several administrators including nine experts maintained those policies. In addition, Wool recently inspected firewall policies collected from different organizations and indicated that all examined firewall policies have security flaws. The process of configuring a firewall is tedious and error prone. Therefore, effective mechanisms and tools for policy management are crucial to the success of firewalls.

Recently, policy anomaly detection has received a great deal of attention Corresponding policy analysis tools, such as Firewall Policy Advisor and FIREMAN, with the goal of detecting policy anomalies have been introduced. Firewall Policy Advisor only has the capability of detecting pair wise anomalies in firewall rules. FIREMAN can detect anomalies among multiple rules by analyzing the relationships between one rule and the collections of packet spaces derived from all preceding rules. However, FIREMAN also has limitations in detecting anomalies. For each firewall rule, FIREMAN only examines all preceding rules but ignores all subsequent rules when performing anomaly analysis. In addition, each analysis result from FIREMAN can only show that there is a misconfiguration between one rule and its preceding rules, but cannot accurately indicate all rules involved in an anomaly.

On the other hand, due to the complex nature of policy anomalies, system administrators are often faced with a more challenging problem in resolving anomalies, in particular, resolving policy conflicts. An intuitive means for a system administrator to resolve policy conflicts is to remove all conflicts by modifying the conflicting rules. However, changing the conflicting rules is significantly difficult, even impossible, in practice from many aspects. First, the number of conflicts in a firewall is potentially large, since a firewall policy may consist of thousands of rules, which are often logically entangled with each other. Second, policy conflicts are often very complicated. One rule may conflict with multiple other rules, and one conflict may be associated with several rules. Besides, firewall policies deployed on a network are often maintained by more than one administrator, and an enterprise firewall may contain legacy rules that are designed by different administrators. Thus, without a priori knowledge on the administrators' intentions, changing rules will affect the rules' semantics and may not resolve conflicts correctly. Furthermore, in some cases, a system administrator may intentionally introduce certain overlaps in firewall rules knowing that only the first rule is important. In reality, this is a commonly used technique to exclude specific parts from a certain action, and the proper use of this technique could result in a fewer number of compact rules. In this case, conflicts are packet monitor or firewall not an error, but intended, which would not be necessary to be changed.

Since the policy conflicts in firewalls always exist and are hard to be eliminated, a practical resolution method is to identify which rule involved in a conflict situation should take precedence when multiple conflicting rules (with different actions) can filter a particular network packet simultaneously. To resolve policy conflicts, a firewall typically implements a first-match resolution mechanism based on the order of rules. In this way, each packet processed by the firewall is mapped to the decision of the first rule that the packet matches. However, applying the first-match strategy to cope with policy conflicts has limitations. When a conflict occurs in a firewall, the existing first matching rule may not be a desired rule that should take precedence with respect to conflict resolution. In particular, the existing first matching rule may perform opposite action to the rule which should be considered to take precedence. This situation can cause severe network breaches such as permitting harmful packets to sneak into a private network, or dropping legal traffic which in turn could encumber the availability and utility of network services. Obviously, it is necessary to seek a way to bridge a gap between conflict detection and conflict resolution with the first-match mechanism in firewalls.

II. LITERATURE SURVEY

This section will introduce us with the previous system and its analysis. The process of configuring a firewall is tedious and error prone. Therefore, effective mechanisms and tools for policy management are crucial to the success of firewalls.
Corresponding policy analysis tools, such as Firewall Policy Advisor and FIREMAN, with the goal of detecting policy anomalies have been introduced. Firewall Policy Advisor only has the capability of detecting pair wise anomalies in firewall rules. FIREMAN can detect anomalies among multiple rules by analyzing the relationships between one rule and the collections of packet spaces derived from all preceding rules. However, FIREMAN also has limitations in detecting anomalies. For each firewall rule, FIREMAN only examines all preceding rules but ignores all subsequent rules when performing anomaly analysis. In addition, each analysis result from FIREMAN can only show that there is a misconfiguration between one rule and its preceding rules, but cannot accurately indicate all rules involved in an anomaly.
LUMETA AND FANG:

Allow user queries for the purpose of analysis and management of firewall policies. Essentially, they introduced lightweight firewall testing tools but could not provide a comprehensive examination of policy misconfigurations.

Solution By Al-Shaer and Hamed :
Designed a tool called Firewall Policy Advisor to detect pairwise anomalies in firewall rules.
Yuan et al. presented FIREMAN, a toolkit to check for misconfigurations in firewall policies through static analysis.

FAME:
As we discussed previously, our tool, FAME, overcomes the limitations of those tools by conducting a complete anomaly detection and providing more accurate anomaly diagnosis information. In particular, the key distinction of FAME is its capability to perform an effective conflict resolution, which has been ruled out in other firewall policy analysis.
Hari et al:
He Provided an algorithm for detecting and resolving conflicts in a general packet filter. However, they can only detect a specific correlation conflict, and resolve the conflict by adding a resolving filter, which is not suitable for resolving conflicts identified recently in firewall policies. There are several interfaces that have been developed to assist users in creating and manipulating security policies. Expandable Grid is a tool for viewing and authoring access control policies . The representation in Expandable Grids is a matrix with subjects shown along the rows, resources shown along the columns, and effective accesses for the combinations of subjects and resources in the matrix cells.
The SPARCLE Policy Workbench allows policy authors to construct policies in a natural language interface, which are in turn translated into machine-readable policies . Even though these tools are useful for authoring access control policies, they cannot effectively represent the results of policy analysis for firewalls.

III. IMPLEMENTATION

Proposed System:
In this system, we represent a novel anomaly management framework for firewalls based on a rule-based segmentation technique to facilitate not only more accurate anomaly detection but also effective anomaly resolution.
Based on this technique, a network packet space defined by a firewall policy can be divided into a set of disjoint packet space segments. Each segment associated with a unique set of firewall rules accurately indicates an overlap relation (either conflicting or redundant) among those rules. We also introduce a flexible conflict resolution method to enable a fine-grained conflict resolution with the help of several effective resolution strategies with respect to the risk assessment of protected networks and the intention of policy definition.
We will maintain firewall logs and will apply association rule mining on logs to find frequent logs. We will use these frequent logs to filter malicious packets. To find out frequent elements from firewall log, we will use Apriori algorithm.
The standard algorithm for discovering frequent item-sets is the Apriori algorithm. Apriori computes in each round the support for all candidate -item-sets. At the end of each round, the item-support parameter are selected. The frequent item-sets of round are used in the next round to construct candidate - item-sets. The algorithm stops when no -item-sets with frequency above the minimum support are found.

```
Apriori:
Ck: Candidate item set of size k
Lk : frequent item set of size k
L1 = {frequent items};
For (k = 1; Lk !=Ø; k++)
Do begin
Ck+1 = candidates generated from Lk;
For each transaction t in database does
Increment the count of all candidates in Ck+1 that are contained in t
Lk+1 = candidates in Ck+1 with minimum support
End
```

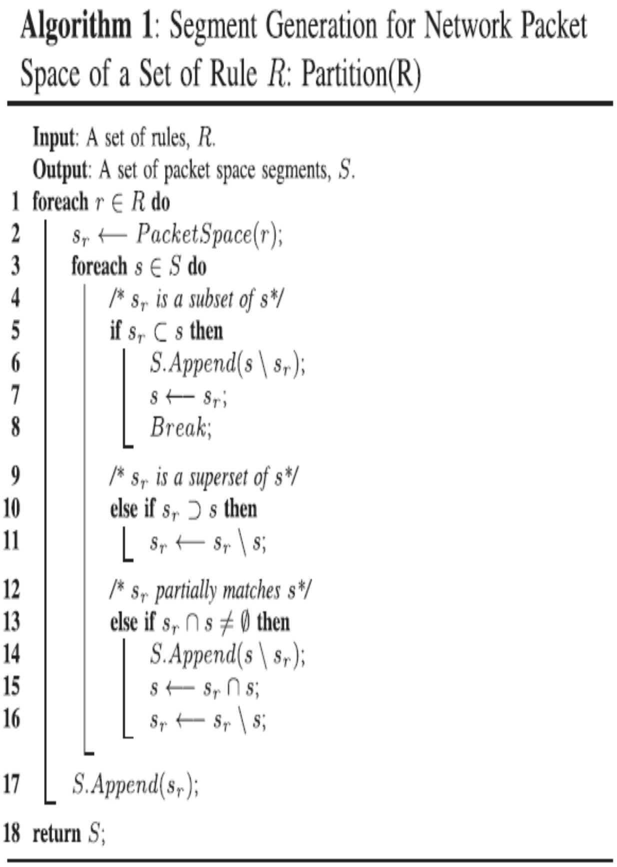
Return Up Lk.

Design :

Steps of proposed Algorithm:

1. Conflicting segment identification and correlation
2. Action constraint generation
3. Conflicting rule reordering
4. Overlap correlation
5. Property assignment
6. Redundancy identification and elimination

Segment Generation Algorithm:



Algorithm 1: Segment Generation for Network Packet Space of a Set of Rule R: Partition(R)

```
Input: A set of rules, $R$.
Output: A set of packet space segments, $S$.
foreach $r \in R$ do
    $s_r \longleftarrow PacketSpace(r)$;
    foreach $s \in S$ do
        /* $s_r$ is a subset of $s$*/
        if $s_r \subset s$ then
            $S.Append(s \setminus s_r)$;
            $s \longleftarrow s_r$;
            $Break$;
        /* $s_r$ is a superset of $s$*/
        else if $s_r \supset s$ then
            $s_r \longleftarrow s_r \setminus s$;
        /* $s_r$ partially matches $s$*/
        else if $s_r \cap s \neq \emptyset$ then
            $S.Append(s \setminus s_r)$;
            $s \longleftarrow s_r \cap s$;
            $s_r \longleftarrow s_r \setminus s$;
    $S.Append(s_r)$;
return $S$;
```

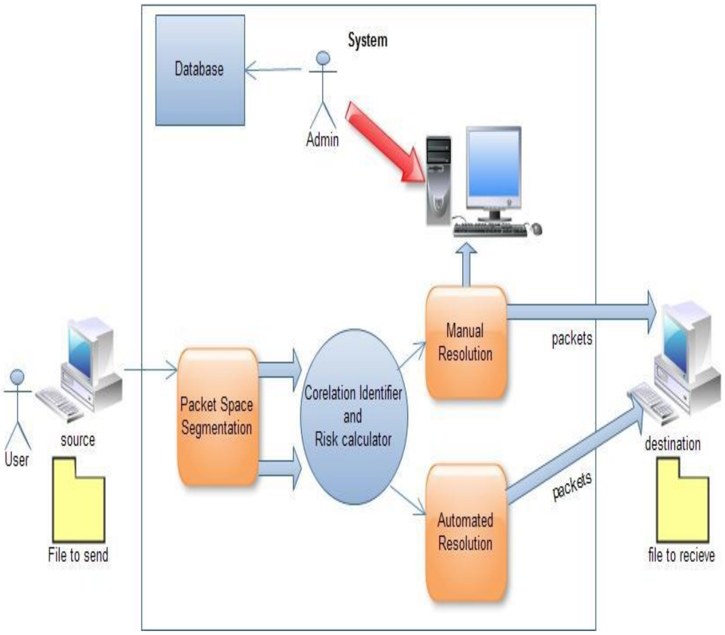
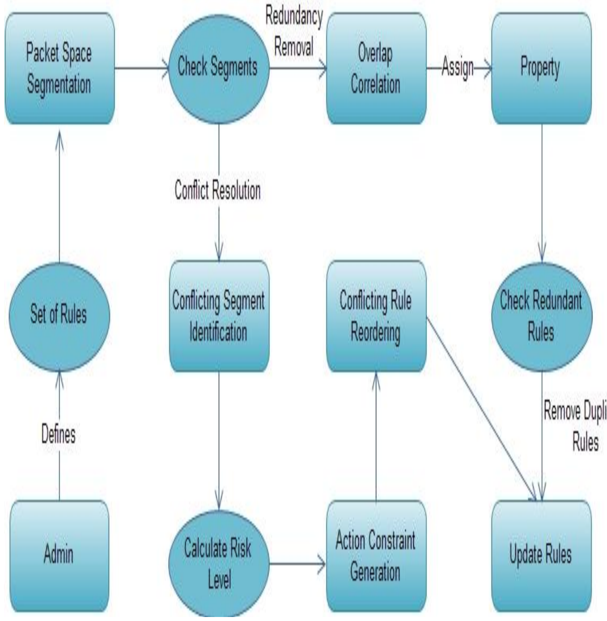


Fig. 1. System Architecture



We are assuming server as a router in our system which observes all the traffic in the network.

Input: {Set of Rules}

Output: {Conflict rules reordered and redundant rules removed}

Success: {Conflict resolution and Redundancy removal}

Failure: {Presence of conflicting and redundant rules}Fig. 2 shows process flow.

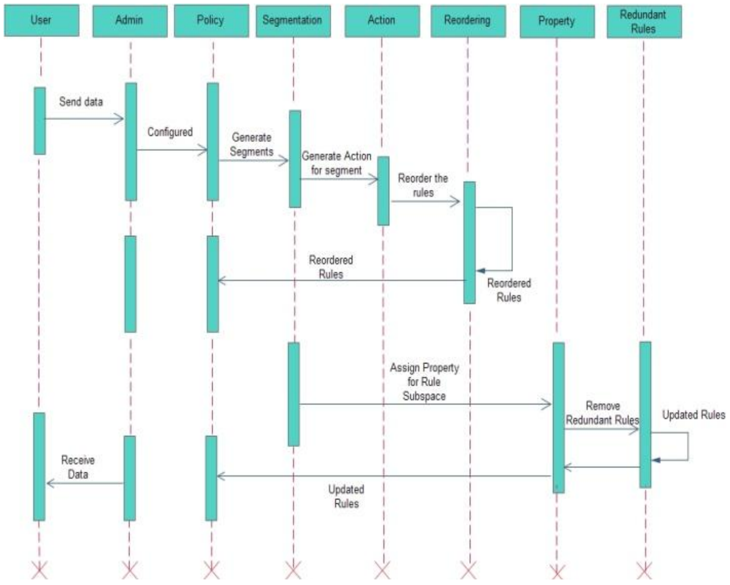


Fig. 2 Process flow.

IV. STATICTICAL MODEL

1] U is main set of users like u1, u2, u3….
U = {u1, u2, u3…….}
2] A is main set of Administrators like a1, a2, a3….
A = {a1, a2, a3…….}
3] R is main set of rules like r1, r2, r3…
R = {r1, r2, r3…….}
4] S is main set of segments like s1, s2, s3….
S = {s1, s2, s3………}
5] C is main set of conflicting segments like c1, c2, c3….
C = {c1, c2, c3………}
6] Identify the processes as P.
P = {Set of processes}
P = {P1, P2, P3……}
If(conflict detection and resolution)then
P1 = {e1, e2, e3}
Where
{e1=i|i is to check conflicting segment identification and correlation}
{e2=j|j is to generate action constraint}
{e3=k|k is to update conflicting rule reordering}

If (user redundancy discovery and removal) then
P1 = {e1, e2, e3}
Where
{e1=i|i is to find overlap segment and correlation}
{e2=j|j is to assign property}
{e3=k|k is to Redundancy identification and elimination}
We can Calculate Risk value for vulnerability as
Risk Value= (CVSS Base Score) *(Importance Value)

Calculate Risk level as

$$RL(cs) = \frac{\sum_{v \in V(cs)} (CVSS(v) \times IV(s))}{\alpha \times |V(cs)|},$$

V. CONCLUSION AND FUTURE WORK

We will propose a novel anomaly management framework that facilitates systematic detection and resolution of firewall policy anomalies. A rule-based segmentation mechanism and a grid-based representation technique were introduced to achieve the goal of effective and efficient anomaly analysis. In addition, we have described a proof-of-concept implementation of our anomaly management environment called FAME and demonstrated that our proposed anomaly analysis methodology is practical and helpful for system administrators to enable an assurable network management. We will maintain firewall logs and will apply association rule mining on logs to find frequent logs. We will use these frequent logs to filter malicious packets. To find out frequent elements from firewall log, we will use Apriori algorithm.

Future Scope:

Our future work includes usability studies to evaluate functionalities and system requirements of our policy visualization approach with subject matter experts. Also, we would like to extend our anomaly analysis approach to handle distributed firewalls. Moreover, we would explore how our anomaly management framework and visualization approach can be applied to other types of access control policies.

Results of Practical Work:

Our conflict resolution mechanism introduces that an action constraint is assigned to each conflicting rule. An action constraint for a conflicting rule defines a desired action (either Allow or Deny) that the firewall policy should take when any packet within the conflicting segment comes to the firewall. Then, to resolve a conflict, we only assure that the action taken for each packet within the conflicting rule can satisfy the corresponding action constraint.

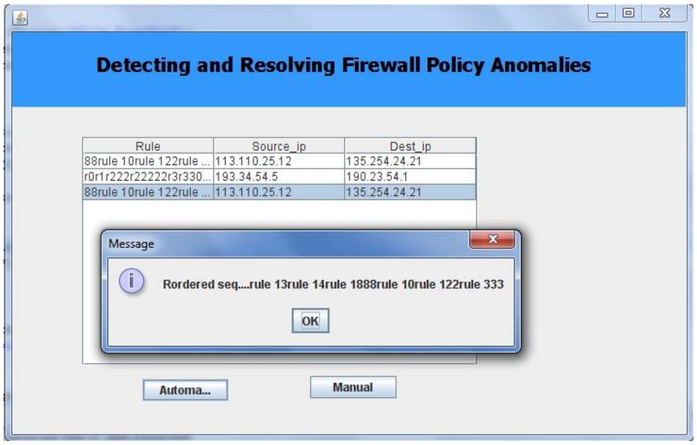
We are going to how resolution rate increases by using our system the conflict resolution of existing methods like first match are very less compared to out method so that resolution rate means number of conflicts resolved with corresponding policy.

In general, when conflicts in a policy are resolved, the risk value of the resolved policy should be reduced and the so at the end we will see that risk value of our policy is reduced compared to existing methods.

Also result of our system would show that there would be increased number of redundant rules identified compared to existing system hence if more number of redundant rules are identified more we will be able to remove it and the systems efficiency increases.

Results :

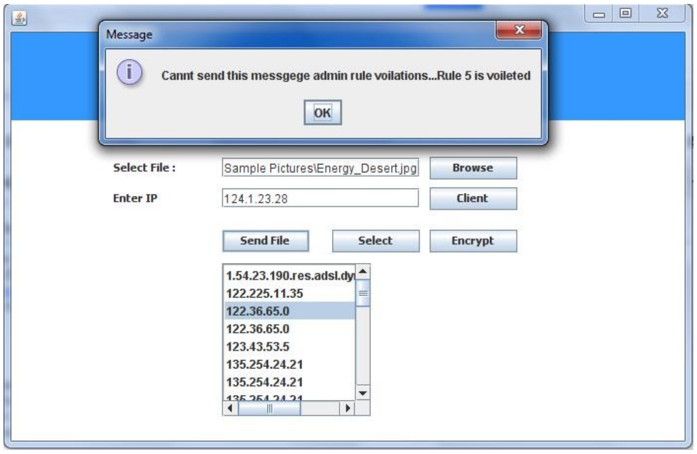
1.Rule Reordering:



Rule	Source_ip	Dest_ip
88rule 10rule 122rule ...	113.110.25.12	135.254.24.21
r0r1r222r22222r3r330...	193.34.54.5	190.23.54.1
88rule 10rule 122rule ...	113.110.25.12	135.254.24.21

Rordered seq....rule 13rule 14rule 1888rule 10rule 122rule 333

2.Policy Violation



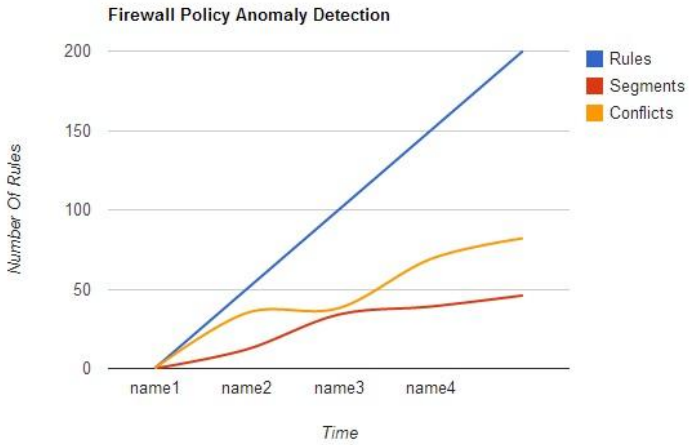
Cannt send this messgege admin rule voilations...Rule 5 is voileted

Select File : Sample Pictures\Energy_Desert.jpg

Enter IP 124.1.23.28

1.54.23.190.res.adsl.dy
122.225.11.35
122.36.65.0
122.36.65.0
123.43.53.5
135.254.24.21
135.254.24.21

Graphs :



ACKNOWLEDGEMENT

Our heartfelt thanks go to Siddhant College of Engineering for providing a strong platform to develop our skills and capabilities. We would like to thank to our guide & respected teachers for their constant support and motivation for us. Last but not least, we would like to thanks all those who directly or indirectly help us in presenting the paper.

References

[1]. E. Al-Shaer and H. Hamed, "Discovery of Policy Anomalies in Distributed Firewalls," IEEE INFOCOM '04, vol. 4, pp. 2605-2616, 2004Security Applications Conference (ACSAC), 2008.

[2]. D. E. Denning. A lattice model of secure information flow. *Commun. ACM, 19:236–243, May 1976.*

[3]. D. E. Denning and P. J. Denning. Certification of programs for secure information flow. *Commun. ACM, 20:504–513, July, 1977.*

[4]. W. Cui, R. H. Katz, andW. tian Tan. Design and Implementation of an extrusion-based break-in detector for personal computers. *In ACSAC, pages 361–370. IEEE Computer Society, 2005.*

[5]. M. G. Jaatun, J. Jensen, H. Vegge, F. M. Halvorsen, and R. W. Nergard. Fools download where angels fear to tread. *IEEE Security & Privacy, 7(2):83–86, 2009.*

[6]. H. Xiong, P. Malhotra, D. Stefan, C. Wu, and D. Yao. Userassisted host-based detection of outbound malware traffic. *In Proceedings of International Conference on Information and Communications Security (ICICS), December 2009.*

[7]. R. Gummadi, H. Balakrishnan, P. Maniatis, and S Ratnasamy. Not-a-Bot: Improving service availability in the face of botnet attacks. *In Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NDSI), 2009.*

[8]. M. Christodorescu, S. Jha, and C. Kruegel. Mining specifications of malicious behavior. *In ESEC-FSE '07: Proceedings of the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering, pages 5–14, New York, NY, USA, 2007. ACM.*

[9]. A. Srivastava and J. Giffin. Tamper-resistant, Application-aware blocking of malicious network connections. In RAID '08: *Proceedings of the 11th international symposium on Recent Advances in Intrusion Detection, pages 39–58, Berlin, Heidelberg, 2008. Springer-Verlag*

[10]. S. Garriss, R. C´aceres, S. Berger, R. Sailer, L. van Doorn, and X. Zhang. Trustworthy and personalized Computing on public kiosks. In MobiSys '08: *Proceeding of the 6th international conference on Mobile systems, applications, and services, pages 199–210, New York, NY, USA, 2008. ACM.*

[11]. A. Baliga, P. Kamat, and L. Iftode. Lurking in the shadows: identifying systemic threats to kernel data. *In IEEE Symposium on Security and Privacy, pages 246–251. IEEE Computer Society, 2007.*

[12]. J. Wei, B. D. Payne, J. Giffin, and C. Pu. Soft-timer driven transient kernel control flow attacks and defense. *In ACSAC '08: Proceedings of the 200 Annual Computer Security Applications Conference, pages 97–107, Washington, DC, USA, 2008. IEEE Computer Society.*

[13]. Z. Wang, X. Jiang, W. Cui, and X. Wang. Countering persistent kernel rootkits through systematic hook discovery. *In RAID '08: Proceedings of the 11th international symposium on Recent Advances in Intrusion Detection, pages 21–38, Berlin, Heidelberg, 2008. Springer-Verlag.*