Towards Autonomic Risk-aware Security Configuration

Mohammad Salim Ahmed†, Ehab Al-Shaer‡, Mohamed Mahmoud Taibah‡, Muhammad Abedin†, Latifur Khan†
†Department of Computer Science, The University of Texas at Dallas
‡School of Computer Science, Telecommunications and Information Systems, DePaul University
salimahmed@utdallas.edu, ehab@cs.depaul.edu, mtaibah@cs.depaul.edu,
arshad@student.utdallas.edu, lkhan@utdallas.edu

***Abstract*—Security of a network depends on a number of dynamically changing factors. These include emergence of new vulnerabilities and threats, policy structure and network traffic. Due to the dynamic nature of these factors, identifying security metrics that measure objectively the quality of security configuration pose a major challenge. Moreover, this evaluation must be done dynamically to handle real time changes in the threat toward the network.**

In this paper, we extend our security metric framework [2] that identifies and quantifies objectively the most significant security risk factors, which include existing vulnerabilities, historical trend of vulnerabilities of remotely accessible services, prediction of potential vulnerabilities for any general network service and their estimated severity and finally propagation of an attack within the network. We have implemented this framework as a user-friendly tool called *Risk based prOactive seCurity cOnfiguration maNAger (ROCONA)* and showed how this tool simplifies security configuration management using risk measurement and mitigation.

I. Introduction

A network is a collection of systems that provide various services. Risk evaluation of each of these services can help in identifying services posing higher risk and thereby calling for extra attention. If the risk of a system or a service can be quantified in such a way, then existing aggregating methods can be used to evaluate the security of the entire network.

Our framework and user-friendly implementation for network security evaluation quantitatively measures the security of a network based on the risk of having a successful attack and the risk of this attack being propagated within the network. As can be seen in Fig. 1, we have modeled our framework as a combination of two parts. The first part measures the security level of the services within the network based on vulnerability analysis. In the second part, the degree of penetration or impact of successful attacks and the risk due to traffic destined for unused address space are measured from a network policy perspective. Service risk components and the unused address space exposure, together give us the threat likelihood and at the same time the attack propagation provides us with the risk impact to the network of interest. When the risk impact is combined with the cost of the damage, we get the the total risk. The framwork of this paper is an extension of our previous work [2].

The effectiveness of a security metric depends on the security measurement techniques and tools that enable network administrators to analyze and evaluate network security. Our proposed tool, based on our framework, can help in comparing security policies to determine which policy is more secure. It is also possible to judge the effect of a change to the policy by comparing the security metrics before and after the change. This framework and its implementation is also an important step toward adaptive security systems in which networks can be evaluated and automatically hardened accordingly by constantly monitoring dynamic changes in the network and service vulnerabilities. We used the Java Programming Language to implement these metrics in one graphical user-interface called *ROCONA*.

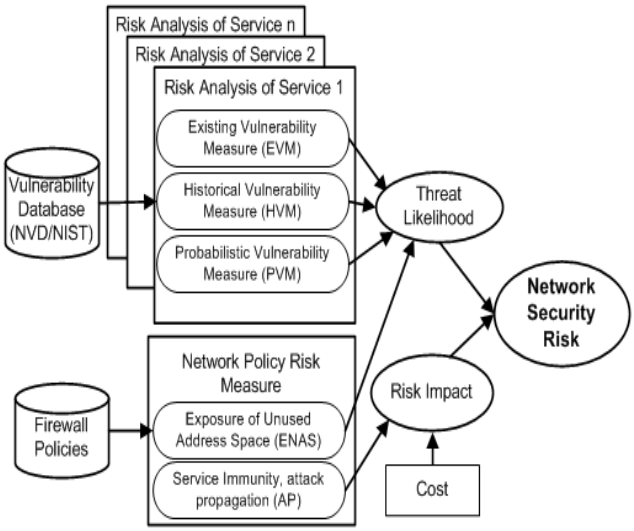

Fig. 1. Network risk measurement framework

The organization of the paper is as follows. First, we discuss related works in Sect. II. The service risk analysis and network policy analysis have been discussed in Sect. III and Sect. IV respectively. Then, we present our implementation of the framework in Sect. V and some experimentental results in Sect. VI. Finally, we present our conclusion in Sect. VII.

II. Related Work

The area of security policy evaluation has seen significant research. Evaluation of VPN, Firewall and Firewall security policies include [3], [5]. Attack graph [4], [8] is another technique that has gained interest recently for assessing the risks associated with network exploits. But the modeling and analysis in this approach is highly complex and costly. Sahinoglu et al. propose a framework in [9] for calculating existing risk. They consider present vulnerabilities in terms of threat represented as probability of exploiting a vulnerability and also the lack of counter-measures.

978-1-4244-2066-7/08/$25.00 ©2008 IEEE

Attack surface of a network is another strategy which has been focused in order to measure security. Mandhata et al. in [6] have tried to find the attack surface from the attackability of a system. In [7] Pamula propose a security metric based on the weakest adversary (i.e. the least amount of effort required to make an attack successful).

But all these work do not represent the total picture. They predominantly try to find existing risk without addressing how risky the system will be in the near future or how policy structure or network traffic would impact on security. They also do not provide a tool that aids in proactive security management.

A detailed investigation of measuring the existing vulnerability, historical trends, probabilistic risk and attack propagation have been anlyzed in our previous work [2].

III. Network Service Risk Analysis

In this section, we describe in brief the method of our vulnerability analysis for service risk measurement that has been used in our tool. This analysis comprises of Existing Vulnerability Measure, Historical Vulnerability Measure and Probabilistic Vulnerability Measure. Details about each of these measures can be found in our previous work [2].

A. Existing Vulnerability Measure

When a vulnerability is discovered, it takes time before a patch is introduced for it. During that time the network and services are vulnerable to outside attack. The *Existing Vulnerability Measure (EVM)* measures this risk. EVM has been studied and formalized in our previous work [1].

We use the *exponential average* to quantify the worst case scenario so that the score is always at least as great as the maximum vulnerability value in the data.

B. Historical Vulnerability Measure

Using the vulnerability history of a service, the *Historical Vulnerability Measure (HVM)* measures how vulnerability prone a given service has been in the past. Considering both the frequency and recency of the vulnerabilities, we combine the severity scores of past vulnerabilities so that a service with a high frequency of vulnerabilities in the near past has a high HVM.

We apply an exponential decay function of the age of the vulnerability. In computing the HVM of individual services, we sum up the decayed scores in each class, and take their weighted sum. Finally, we take its natural logarithm to bring it to a more manageable magnitude. This equation is designed to be dominated by the highest HVM of the services exposed by the policy [2]. We take the exponential average of all the HVMs so that the score will be at least equal to the highest HVM, and will increase with the HVM's of the other services.

C. Probabilistic Vulnerability Measure

Using the vulnerability history of a service, we can calculate the probability of at least one new vulnerability being published in a given period of time. From the vulnerability history, we can also compute the expected severity of the vulnerabilities exposed in the next period of time [2].

We define *Expected Risk (ER)* for a service as the product of the probability of at least one new vulnerability affecting the service in the next period of time and the expected severity of the vulnerabilities. We can compare two policies using this measure – a higher value of the measure will indicate a higher chance of getting vulnerable in the near future. For each service, we determine the probability of a new vulnerability appearing within the next time interval, T, from the list of interarrival times. We also determine the expected severity of that vulnerability from the probability distribution of the vulnerabilities in the past for that service. We calculate ER using these two measures. This ER is then used to get the PVM of that particular service by taking the exponential average. The formalization of this measure is given in [2].

EVM, HVM and PVM based risk mitigation: As can be seen from our definition of EVM, it indicates whether there are existing vulnerabilities present in the system of interest. In order to minimize risk present in the system, $ROCONA$ follows these steps – (1) Finds out which vulnerabilities have solutions and alerts the user with the the list of patches available for install, (2) Services having higher risk value than user defined threshold are listed to the user with the options: (i) Block the service completely (ii) Limit the traffic toward the service by inserting firewall rules (iii) Minimize traffic by inserting new rules in IDS (iv) Place the service in DMZ area and (v) Manual. (3) Recalculate EVM to show score for updated settings of the system. The system administrator can use the tool to calculate the EVM score of other deployable services to find EVM scores of them and make a cost-benefit analysis to decide on changing the service providing software.

Unlike EVM, HVM score gives us the historical profile of a software and PVM score gives us the latent risk toward the system. Therefore, a software with low HVM and PVM score providing the same service should be preferred. Our implemented tool performs the following steps to mitigate the HVM and PVM risk: (1) Calculate the HVM and PVM scores of the services (2) Compare the scores to user defined threshold vaules. (3) If scores are above the threshold then strengthen layered protection with options just like in case of EVM (4) Recalculate the HVM and PVM scores of the services (5) If scores still above the threshold, then show recommendations to the system administrator. Recommendations include (i) Isolation of the service using Virtual LANs (ii) Increase weight (i.e. importance) to the alerts originating from these services even if false alarms increase. (iii) Propose the use of both *Host based IDS (HIDS)* and *Network based IDS (NIDS)*, if both of them are not present and (iv) Replace the service providing software with a different one. The administrators can use our tool to measure the HVM and PVM scores of similar service providing softwares from the NVD database and choose the best possible solution. But just like EVM based risk mitigation, a cost-benefit analysis must precede such a decision making.

IV. Network Policy Risk Analysis

The network policies determine the exposer of the network to outside world as well as the extensiveness of an attack on the network (i.e. how widespread the attack is). The *Attack Propagation (AP)* and *Exposure of Unused Address Spaces*

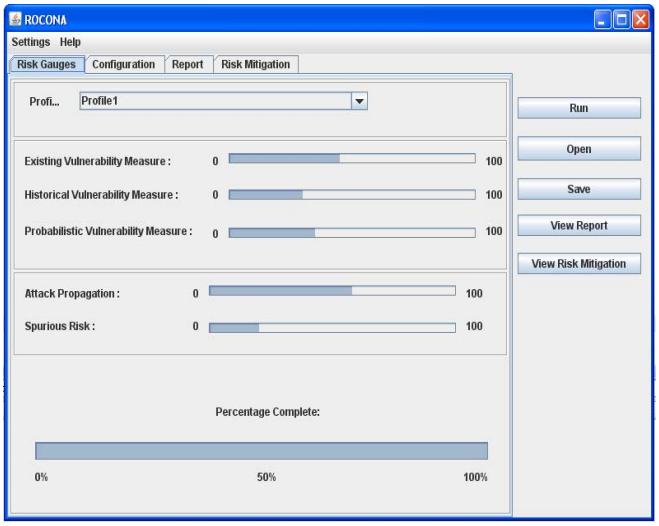

Fig. 2. ROCONA Risk Gauges

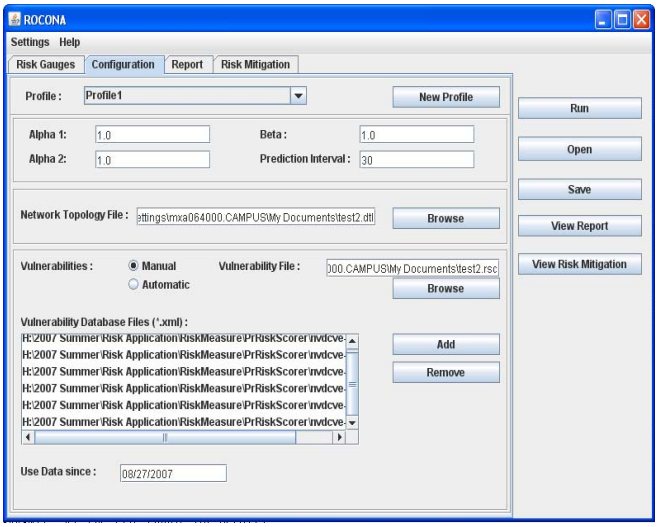

Fig. 3. ROCONA Configuration

(ENAS) are the two factors that we have used to quantify risk with respect to network security policies.

A. Attack Propagation Metric

Attack propagation (AP) provides us with an indication of how widespread an attack can be. The *Attack Propagation* measure assesses how difficult it is for an attacker to propagate an attack through a network, using service vulnerabilities as well as security policy vulnerabilities.

1) The Attack Immunity of a Service: For our analysis, we define a measure, I_s, that assesses the attack immunity of a given service, s, to vulnerabilities based on that service's EVM and HVM. I_s is directly calculated from the combined vulnerability measure of a service s.

To extract a measure of security for a given network, we map the network of interest to a *Service Connectivity Graph (SCG)*. In the SCG, each arch between two hosts is labeled with the corresponding attack immunity measure.A path through the network having a low combined historical immunity value represents a security issue.

For each node, to find how difficult it is for an attacker to compromise all the hosts within reach from it, we build a minimum spanning tree for that node for its segment of the SCG. The weight of this minimum spanning tree will represent how vulnerable this segment is to an attack. We define *Service Breach Effect* to be the weight of this tree. It actually denotes the damage possible through a node. Details about this measure can be found in [2].

2) AP based risk mitigation: This metric indicates the penetrability of the network. Therefore, if this metric has a high value, it indicates that the network should be partitioned to minimize communications within the network and propagation of an attack. The possible risk mitigation measures that can be taken include (1) Network redesigning (introducing virtual LANs) (2) Rearrange the network so that machines having equivalent risk are in the same partition. (3) Increase the number of enforcement points (Firewalls, IDS) (4) Increase the security around the hot spots in the network and (5) Strengthen MAC sensitivity labels, DAC file permission sets, access control lists and roles or user profiles.

B. Exposure of Unused Address Spaces (ENAS)

Policies should not allow spurious traffic, i.e., traffic destined to *unused* IP addresses or port numbers, to flow inside the network, because this spurious traffic has the potential to consume bandwidth and cause DDoS attacks. In our previous work [3], [5], we show how to identify automatically the rules in the security policy that allow for spurious traffic, and once we identify them, we can readily compute the spurious residual risk for the policy.

In order to accurately estimate the risk of the spurious traffic, we must consider how much spurious traffic can reach the internal network, what is the ratio of the spurious traffic to the total capacity and what is the average available bandwidth used in each internal subnet. Assuming that maximum per-flow bandwidth allowed by the firewall policer is M_{l_i} Mbps for link l_i of capacity C_{l_i}, and F_{l_i} is the set of all spurious flows passing through link l_i, we can estimate the *Residual Capacity (RC)* for link l_i as follows:

$$RC(l_i) = 1 - \frac{\sum_{f_j \in F_{l_i}} M_{l_i}}{C_{l_i}} \quad (1)$$

The residual capacity has a range $[0, 1]$. We can now calculate the *Spurious Risk (SPR)* of host d and then sum this measure for all the hosts in network A to get the SPR for the entire network :

$$SPR(A) = \sum_{d \in N} (c(d) \times \max_{l_i \in L}(1 - RC(l_i))) \quad (2)$$

Where $c(d)$ is the weight (i.e. importance or cost) associated with the host d in the network with a range $[0, 1]$ and L is the set of links connected to host d. We take the minimum of all the Residual Capacity associated with host d to reflect the maximum amount of spurious traffic entering the network and therefore measuring the worst case scenario. This is the spurious residual risk after considering the use of per-flow traffic policing as a counter-measure and assuming that each of the hosts within the network have a different cost value associated with each of them.

SPR based risk mitigation: High score for this metric indicates that the firewall rules and policies require fine tuning. When the score of this measure rises beyond the threshold level, *ROCONA* recommends additional firewall

rules to the firewall. The allowable IP addresses and ports need to be checked thoroughly so that no IP address or port is allowed unintentionally. This metric also gives us indication where to put the expensive security measures to minimize cost and maximize security. To assist in switch configuration outside each partition is another feature of our tool.

V. ROCONA Tool Implementation

To simplify the network risk measurement and mitigation, we have implemented a tool called *Risk based proactive security configuration manager (ROCONA)*. We have used Java Programming Language for its implementation. The tool can run as a daemon process and therefore provide system administrators with periodic risk updates. Five risk scores are provided for components that have already been described. The measures are provided as risk gauges. The scores are shown as values between 0 and 100. The users can setup different profiles for each run of the risk measurement. The options that a user can configure include parameters of EVM, HVM and PVM. Also the network topology file describing the interconnections of nodes within the network and the vulnerability database (.xml) files can also be configured. All these configuration options are shown in Figure3. After the tool completes its run, it provides the system administrator with the measurement process in details. All the details can be seen in the *Details* tab. Using all the gauges, *ROCONA* also provides risk mitigation strategies (i.e. which service needs to patched, how rigorous firewall policies should be, etc. already described with the individual measures) in the *Risk Mitigation* tab. All this information can be stored for future use. It will be made available for evaluation in the web in near future when it becomes a stable release.

VI. Experimentation and Evaluation

We experimented for HVM and PVM using publicly available vulnerability databases. we used the National Vulnerability Database (NVD) published by National Institute of Science and Technology (NIST). All the vulnerabilities are stored using the standard CVE (Common Vulnerabilities and Exposures) name. For each vulnerability, the NVD provides the products and versions affected, descriptions, impacts, cross-references, solutions, loss types, vulnerability types, the severity class and score, etc. We have used the database snapshot updated at 04/05/2007. In our evaluation process, we divided the data into training sets and test sets and tested our metric on a large number of services and random policies. Using the vulnerability publishing dates we performed validation of HVM and the interarrival times and severity information was used to measure the PVM. Details of these experiments can be found in [2].

a) Validation of Spurious Risk (SPR): In order to validate the *Spurious Risk*, we use a risk model. In this model, we assume that the DDoS attackability of a host is proportional to the expertise of the attacker and inversely proportional to link capacity C_{l_i}, as can be seen in real life scenarios. We used this model to perform simulations for three cases. In the first case, most of the attackers were inexperienced. In the second case, majority of attackers had moderate experience whereas in the third simulation

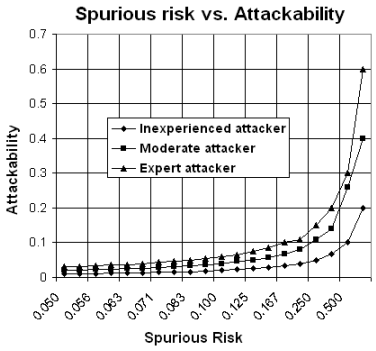

Fig. 4. Spurious Risk Vs. Attackability graph for different attacker levels

attackers were mostly experts. Finally, we plot attackability with increasing spurious risk in Figure 4(c). It shows clear increasing trend of attackability with respect to *Spurious Risk*.

VII. Conclusions

As network security is of utmost importance for an organization, an autonomic risk-aware configuration manager will be highly effective in assessing the protection of the current policy, and justifying consequent decisions to strengthen security. In this paper, we present a system that quantitatively evaluates security policies based on several important factors that have profound effect on the security of a network. Our proposed tool is useful not only for administrators to evaluate policy/network changes and, take timely and judicious decisions, but also for enabling adaptive security systems based on vulnerability and network changes.

Acknowledgments

The authors would like to thank Syeda Nessa of The University of Texas at Dallas for her help with the formalization and experiments making this work possible.

References

[1] M. Abedin, S. Nessa, E. Al-Shaer, and L. Khan. Vulnerability analysis for evaluating quality of protection of security policies. In *2nd ACM CCS Workshop on Quality of Protection*, Alexandria, Virginia, October 2006.

[2] M. S. Ahmed, E. Al-Shaer, and M. M. T. L. Khan. A novel quantitative approach for measuring network security. In *Press, IEEE Infocom Miniconference*, Phoenix, AZ, April 2008.

[3] E. Al-Shaer and H. Hamed. Discovery of policy anomalies in distributed firewalls. In *Proceedings of IEEE INFOCOM'04*, March 2004.

[4] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 217–224, New York, NY, USA, 2002. ACM Press.

[5] H. Hamed, E. Al-Shaer, and W. Marrero. Modeling and verification of ipsec and vpn security policies. In *Proceedings of IEEE ICNP'2005*, November 2005.

[6] P. Manadhata and J. Wing. An attack surface metric. In *First Workshop on Security Metrics*, Vancouver, BC, August 2006.

[7] J. Pamula, P. Ammann, S. Jajodia, and V. Swarup. A weakest-adversary security metric for network configuration security analysis. In *ACM 2nd Workshop on Quality of Protection 2006*, Alexandria, VA, October 2006.

[8] C. Phillips and L. P. Swiler. A graph-based system for network-vulnerability analysis. In *NSPW '98: Proceedings of the 1998 workshop on New security paradigms*, pages 71–79, New York, NY, USA, 1998. ACM Press.

[9] M. Sahinoglu. Security meter: A practical decision-tree model to quantify risk. In *IEEE Security and Privacy*, June 2005.