Vulnerability Analysis For Evaluating Quality of Protection of Security Policies

Muhammad Abedin[†], Syeda Nessa[†], Ehab Al-Shaer[‡], Latifur Khan[†]
[†]Department of Computer Science, The University of Texas at Dallas
[‡]School of Computer Science, Telecommunications and Information Systems, DePaul University
maa056000@utdallas.edu, skn051000@utdallas.edu, ehab@cs.depaul.edu, lkhan@utdallas.edu

ABSTRACT

Evaluation of security policies, specifically access control policies, plays an important part in securing the network by ensuring that policies are correct and consistent. Quality of protection (QoP) of a policy depends on a number of factors. Thus it is desirable to have one unified score based on these factors to judge the quality of the policy and to compare policies. In this context, we present our method of calculating a metric based on a number of factors like the vulnerabilities present in the system, vulnerability history of the services and their exposure to the network, and traffic patterns. We measure the existing vulnerability by combining the severity scores of the vulnerabilities present in the system. We mine the National Vulnerability Database, NVD, provided by NIST, to find the vulnerability history of the services running on the system, and from the frequency and severity of the past vulnerabilities, we measure the historical vulnerability of the policy using a decay factor. In both cases, we take into account the exposure of the service to the network and the traffic volume handled by the service. Finally, we combine these scores into one unified score – the Policy Security Score.

Categories and Subject Descriptors: C.2.0 [Computer-Communication Networks]: General[Security and protection]; K.6.5 [Management of Computing and Information Systems]: Security and Protection

General Terms: Security.

Keywords: Security, policy, evaluation, metric.

1. INTRODUCTION

Evaluation of security policies is a very important part of keeping networks secure. To keep up with constantly increasing security threats, security policies like access control rules in firewall, IPSec, IDS, IPS, etc. also have to be kept up to date. Whenever a policy is modified due to changes in requirement, it has to be re-evaluated to ensure that the required security levels are still met. Our research aims at providing a solution to this problem by presenting a method of analyzing and assigning scores to security policies. Using this metric, we can judge whether a change to a policy resulted in a better or worse policy, as well as compare policies.

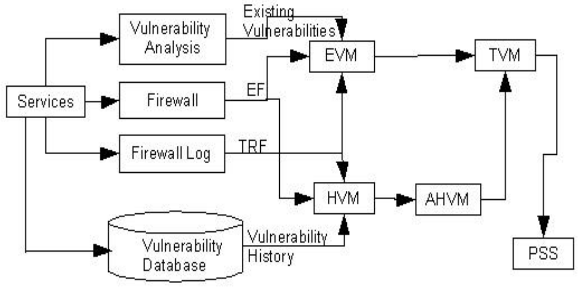

Figure 1: Flow-diagram of calculating the Policy Security Score, PSS. EF is Exposure factor, TRF is Traffic Rate Factor, EVM is Existing Vulnerability Measure, HVM is Historical Vulnerability Measure, AHVM is Aggregated Historical Vulnerability Measure, TVM is Total Vulnerability Measure.

Quite a number of factors contribute to the quality of a policy. In this paper, we focus on the most important of these factors –existing vulnerabilities, vulnerability history of the services exposed by the policy, exposure of the services to the network, and traffic hit ratio for the services. First, we determine the set of vulnerabilities present in the system, and measure the severity of the threat posed by these vulnerabilities. Second, we determine the set of services exposed by the policy to the network. From the vulnerability history of these services, we measure how vulnerability-prone they are, based on the frequency and severity of their past vulnerabilities, and compute one historical vulnerability measure for the policy. We also consider the magnitude of exposure and the volume of network traffic handled by the services. The policy vulnerability measure will be the combination of the present and historical vulnerability measures. Higher vulnerability scores indicate poorly secure policies, and so we define the Policy Security Score as a decreasing function of the policy vulnerability measure. We illustrate this approach in Fig. 1.

Though there has been a lot of work in the direction of verification of security policy, for example in [5, 3, 4, 2, 1], there has little research in evaluation and comparison of

QoP'06, October 30, 2006, Alexandria, Virginia, USA.
Copyright 2006 ACM 1-59593-553-3/06/0010 ...$5.00.

policies. In this paper, we present our method of computing the Policy Security Score and develop the necessary mathematical framework and equations in Sect. 2, and in Sect. 3, we discuss our conclusions from this research, and present some directions for future research.

2. POLICY SECURITY SCORE

A policy can span a number of systems and defines which services are accessible to and from the extenral network. Thus a policy can be completely characterized by the systems and the services it exposes. Services can be running with latest updates and patches, and hence display no present vulnerabilities, yet if they are historically prone to have critical vulnerabilities exposed periodically, one cannot expect the policy to be very secure in the long run. Again, if the services are historically very stable, but are not kept updated, we are again left with an insecure system. Thus, it is clear that both present and historical vulnerabilities are important for having a secure system. Also we need to consider the network exposure and traffic volume factors in developing the metric. Intuitively, wide exposure to the network increases the probability of attack on the service. Also, the probability of attacks increase with the volume of the traffic that a service handles. Based on these factors, we want to compute a single score, Policy Security Score (PSS), with the following principles:

1. The value of PSS for a system will be assigned in the range $[0, 10]$.
2. A system with no present vulnerabilities and using secure and stable services will have the highest PSS.
3. PSS will be a decreasing function of the combined risk factor of the present and historical vulnerabilities discovered in the system considering the network traffic factors.

In the following sections, we describe the calculation of the PSS.

2.1 Calculating the Existing Vulnerability Measure

Security can be easily compromised by the vulnerabilities present in the services. We compute the Existing Vulnerability Measure, EVM, for a policy, based on the severity of the vulnerabilities present. There can be two types of vulnerabilities – those that have known solutions or patches, but present as the solutions or patches have not been applied, and those that do not have any known solution, including those that have been discovered very recently, and have not been fully analyzed or understood yet, posing a greater security risk than those with known solutions.

In developing the equation for the EVM, our goal is to design an equation for the score such that it will be at least as large as the weighted sum of the highest vulnerability scores present in the system in each class, and the score will increase with the contribution of the risk factors of the other vulnerabilities. For a set of numbers $x_1, x_2, \ldots, x_n$, we define the *arithmetic mean* as $\frac{1}{n}\sum_{i=1}^{n} x_i$, *geometric mean* as $\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$ and *exponential average* as $\ln \sum_{i=1}^{n} e^{x_i}$. As an example, if the data is 1, 3, 4, 8, the arithmetic mean would be 4, the geometric mean would be 3.13, and the exponential average would be 8.026. Thus, arithmetic and geometric mean does not meet our goal since they all result in a quantity that is always less than the maximum, whereas the exponential average produces a value that is always at least as great as the maximum value in the data, and hence we choose the exponential average.

Let $EV(A)$ be the set of vulnerabilities that currently exist in the system A, and $SS(v)$ be the severity score of a vulnerability v. We can divide $EV(A)$ into two sets – $EV_S(A)$ contains all the vulnerabilities that have existing solutions, and $EV_U(A)$ contains those without existing solutions. We take the exponential average of the severity scores of the vulnerabilities present in the two sets separately and take the weighted sum of the two averages to calculate the EVM. Mathematically,

$$EVM(A) = \alpha_1 . \ln \sum_{v_i \in EV_S(A)} e^{SS(v_i)} + \alpha_2 . \ln \sum_{v_j \in EV_U(A)} e^{SS(v_j)} \tag{1}$$

Here, the weight factors α_1 and α_2 can be used to model the difference in security risks posed by the two classes of vulnerabilities.

2.2 Calculating the Historical Vulnerability Measure

In calculating the Historical Vulnerability Measure, HVM, we first need to determine the HVM of different services using information from standard public vulnerability databases. Let $HV(S)$ be the set of vulnerabilities of the service S. We divide $HV(S)$ into three groups – $HV_H(S)$, $HV_M(S)$ and $HV_L(S)$ for vulnerabilities that pose high, medium and low risks to the system. We need to assign the highest weight to the vulnerabilities in the set $HV_H(S)$, and progressively lower weights to those in $HV_M(S)$ and $HV_L(S)$. In evaluating a service, the vulnerabilities discovered a long time ago should carry smaller weight, because with time these would be analyzed, understood and patched. Hence, we apply an exponential decay function decreasing with the age of the vulnerability to the severity scores. In computing the HVM of individual services, we sum up the decayed scores in each class, and take their weighted sum. Since this sum can be quite large, we take its natural logarithm to bring it to more manageable magnitude. The equation for HVM of service S, $HVM(S)$ is as follows:

$$HVM(S) = \ln \sum_{X \in \{H,M,L\}} w_X . \sum_{v_i \in HV_X(S)} SS(v_i).e^{-\beta Age(v_i)} \tag{2}$$

In the exponential decaying factor applied to the severity scores of the vulnerabilities, $e^{-\beta Age(v)}$, the parameter β controls how fast the factor decays with age.

We conducted an experiment to evaluate the HVM score with the hypothesis that if service A has a higher HVM than service B, then in the next period of time, service A will display a higher number of vulnerabilities than B. Based on the National Vulnerability Database (NVD) published by National Institute of Science and Technology (NIST) updated at 08/04/2006 and available at `http://nvd.nist.gov/download.cfm`, we calculated the HVM of all the products in the database. The NVD contains vulnerabilities published since 1999. We used vulnerability data upto 12/31/2005 to compute the HVM of the services, and used the rest of the data from 01/01/2006 onwards to validate the result. We varied β so that the decay function falls to 0.1 in 0.5, 1 , 1.5 and 2 years, and observed the best accuracy for the first

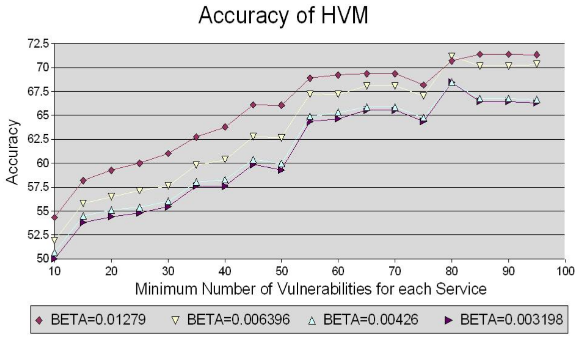

Figure 2: Accuracy of the HVM for different values of β and minimum number of vulnerabilities.

case. As this is a historical measure, better results should be found if there are more history of vulnerabilities for the services. Here, we first chose only those services with at least 10 vulnerabilities in their lifetimes, and gradually increased this lower limit to 100, and observed that the accuracy does increase with the lower limit. The graph in Fig. 2 presents the results of this experiment. Using Eqn. 2, we can compute the historical vulnerability measure of any service. In order to evaluate the aggregated HVM of a system, we take the set of services exposed to the network by the policy, and combine their historical service vulnerability scores. If the set of such services in a system A is $SERVICES(A)$, then the aggregated historical vulnerability measure of system A, $AHVM(A)$ is calculated as

$$AHVM(A) = \ln\left(\sum_{s_i \in SERVICES(A)} e^{HVM(s_i)}\right) \quad (3)$$

Like Eqn. 1, this equation is designed to be dominated by the highest HVM of the services exposed by the policy. The score will at least be equal to the highest HVM, and will increase with the HVM's of the other services.

2.3 Calculation of Policy Security Score

For a system A, we can combine $EVM(A)$ and $AHVM(A)$ into one total vulnerability measure of the system, $TVM(A)$, as the weighted sum of $EVM(A)$ and $AHVM(A)$:

$$TVM(A) = \eta_1.EVM(A) + \eta_2.AHVM(A) \quad (4)$$

η_1 and η_2 define the weight given to each type of vulnerability measure. We define the Policy Security Score of the system A, $PSS(A)$, as:

$$PSS(A) = 10e^{-\gamma TVM(A)} \quad (5)$$

This will assign the score of 10 to a system with vulnerability score of 0. The score assigned by this equation will be a monotonically decreasing function of the vulnerability score of the system. The parameter γ provides control over how fast the policy score decreases with the risk factor.

2.4 Considering Network Factors

We define the Exposure Factor, EF, of a service S as a ratio between the logarithm of the product of the number of IP addresses and ports served by S, $IP(S)$ and $PORTS(S)$, and the logarithm of the product of the total number of IP addresses and ports, 2^{32} and 2^{16}. As exposure to the network will magnify the risk, but not being exposed to the network will not mitigate the risk, we add 1 to this quantity:

$$EF(S) = 1 + \frac{\log_2(IP(S).PORTS(S))}{\log_2(2^{32}.2^{16})} \quad (6)$$

The probability of a service's getting attacked increases with the volume of traffic it handles. We can model this factor by defining the Traffic Rate Factor, TRF, of a service S as:

$$TRF(S) = 1 + \frac{\text{Traffic volume of } S}{\text{Total traffic volume}} \quad (7)$$

These two factors need to be considered when using the severity score of a vulnerability by multiplying the severity score in the database by the EF and TRF to take into account for these factors in all calculations.

3. CONCLUSION AND FUTURE WORKS

We have presented an approach of assigning scores to security policies based on a number of factors – past and present vulnerabilities, traffic patterns and exposure to network. This is an important step in the direction of automated security policy evaluation that can eventually lead to fully automated evaluation tools to assist administrators in securing the network. The addition of appropriate reporting can help administrators in paying attention to where improvement is most needed and give a guideline of how to change it for better.

There are a number of directions for pursuing future research on this track. An important improvement would be to mine multiple standard vulnerability databases and aggregate the data found in them. The major challenge will be to combine the severity metrics used in different databases in a meaningful way. The security of a network depends upon a number of issues besides service vulnerabilities, like physical, operational and procedural security, strength of passwords and change frequency, etc. The policy score can be made more accurate if these factors can also be incorporated into the PSS. Another very useful and challenging direction would be the mining of the vulnerability databases for useful patterns of vulnerabilities. In future we can mine this data for prediction purposes, like calculating the probabilities of future attacks given the current state of the policy.

4. REFERENCES

[1] A. Atzeni and A. Lioy. Why to adopt a security metric? a little survey. In *QoP-2005: Quality of Protection workshop*, September 2005.

[2] A. Atzeni, A. Lioy, and L. Tamburino. A generic overall framework for network security evaluation. In *Congresso Annuale AICA 2005*, pages 605–615, October 2005.

[3] A. El-Atawy, K. Ibrahim, H. Hamed, and E. Al-Shaer. Policy segmentation for intelligent firewall testing. In *1st Workshop on Secure Network Protocols (NPSec 2005)*, November 2005.

[4] M. Frantzen, F. Kerschbaum, S. Fahmy, and E. Schultz. A framework for understanding vulnerabilities in firewalls using a dataflow model of firewall internals. *Computers and Security*, 20(3):263–270, May 2001.

[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.