

secu-AUDIT: CONTINUOUS COMPUTER SECURITY AUDITING EXPERIENCES

Urko Zurutuza, Roberto Uribeetxeberria, Jesus Lizarraga, Iñaki Velez de Mendizabal

Computer Science Department, Mondragon University
Mondragon, Gipuzkoa, (Spain)
{uzurutuza,ruribeetxeberria,jlizarraga,ivelez}@eps.mondragon.edu

Abstract

In this paper, we present the experience obtained by the application of a methodology designed in Mondragon University. In this experience, three different security tools were combined: Nessus, Snort and Nagios. This way, the security administrator and even the enterprise management can easily analyse the organizations information system's security level in real time. This can be accomplished just by means of checking its representation and assess the risk that involves any change of the level. The ability to measure the current state of the security is essential to continue improving the safeguard of our information. secu-AUDIT, seeks to define and analyse a methodology for the realisation of continuous audits of network security in organisations. This allows a proactive position regarding to security issues as one can be aware of the level acquired as well as the level required. The paper gives a brief overview of security metrics, discusses how the metrics are obtained in order to measure the security level and provides an example of carrying out a continuous audit.

Keywords

Computer security, continuous auditing, security auditing, security metrics.

1. INTRODUCTION

As organisations' information systems grow, they must assume that their risk rises at the same time. It gets more difficult to update, maintain and verify the proper state and configuration of the network systems. On the other hand, a great number of vulnerabilities are reported for several systems every year. Unpatched vulnerabilities allow outsider and insider attackers to break into the systems and generate security incidents. In fact, these incidents are growing in an exponential way as shown in figure 1.

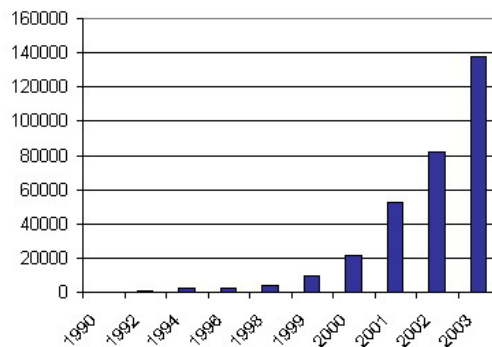


Fig 1. Security incidents reported by CERT [1]

Computer security audit is an important issue in the computer security field. Computer auditing measures how the confidentiality, availability and integrity of an organization's information are assured [2]. An information security audit is one of the best ways to determine the security level of an organization's information system.

Measuring our organisations security level has always been a difficult challenge for auditors and system administrators. Current audit techniques allow obtaining a snapshot of the state of the security in a given moment. When an audit is accomplished, some security deviations are detected, and it is in that precise moment when corrective measures are taken to increase the security level. Nevertheless, the period

between two audits is not evaluated and important events may happen during this time. These events can significantly decrease the security level of the systems making successful attacks possible as the system may not be secure any longer.

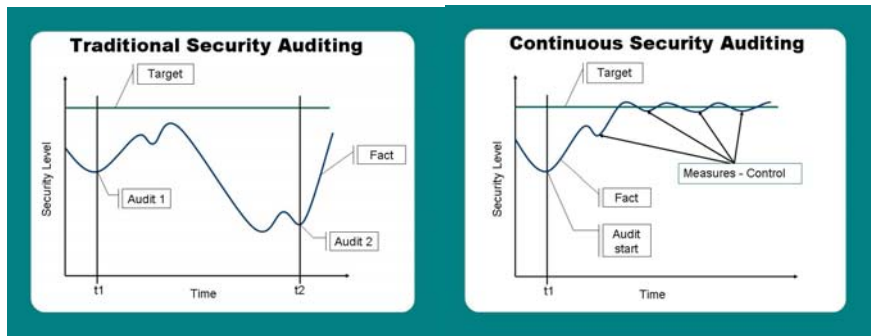


Fig 2. Traditional security auditing Vs. continuous security auditing

While the concept of continuous auditing (CA) [3] is over a decade old, the rapid advances in technology have now made it feasible [4] [5]. Up to the moment, CA had only a financial meaning. In a computer security context, the process of CA consists on measuring and controlling the security level, performing continuous monitoring in time and taking proofreaders actions if necessary. This ongoing process establishes a method that allows organisations to control and to monitor the security level in real time to be able to take the appropriate measures in case a deviation occurs.

This paper describes the results and benefits of applying secu-AUDIT, our methodology for continuous computer security auditing presented in [6]. The next section describes the methodology and gives a brief overview of security metrics. Section 3 presents details about the implementation and measurements obtained from our prototype. Finally, further work and conclusions are outlined.

2. secu-AUDIT METHODOLOGY

The CA model starts on an initial meditation of what do we need to measure in order to get an overall security state. In this new method, it is necessary to select automatically measurable metrics and the tools that will provide those measurements. Once we collect the security metrics in a database, we suggest a formula based on a statistical analysis of the behaviour of each metric.

2.1 Security metrics

We understand security metrics as a uniform monitoring method and an objective way to document our organisation's security attitude. Security metrics are required to understand the current state of security and its improvement in time. It is necessary to establish control points in order to evaluate the state of the effectiveness of the security. This will improve the protection level. The metrics will permit us to work on security as a dynamic process and not as a final product.

In order to calculate the security level, a set of metrics that allow to carry out the measures and the pursuit of the security level in situ and in a completely automatic way has been identified (which could be expandable). Measuring the security metrics, it is possible to determine their influence in the audit process and see whether a continuous improvement is obtained or not.

This way, starting from the data that are capable to catch some security and network management tools, we suggested 6 different measures:

- Number of high/medium/low risk vulnerabilities
- Number of intrusions
- Time of down servers
- Time of down services

These variables are in their simpler form, and they will be modified later for an easier analysis.

It seems logical to evaluate the interaction of the variables with the environment. The security level can differ from big organisations to smaller ones, even if the measure of a variable is the same. For example, it would not be unusual to have a lot of login attempts, intrusions or viruses in a big organisation, and this would give us a certain security level. On the other hand, if a small organisation had the same values, the security level value should be much lower. To solve this problem, the measures have been transformed into metrics that are independent from environment.

2.2 A case study: number of vulnerabilities

Taking the number of vulnerabilities as an example, we realise that it was necessary to divide the measure into two metrics; in one hand the total number of vulnerabilities per host, and in the other, the number of different types of vulnerabilities. The reason of dividing the metric is that, for example, we can find a great number of vulnerabilities in an organisation, but most of them can be the same one (in different hosts). In other words, it is like one key that opens several doors with the same bolt. Instead, finding few vulnerabilities but all of them different could be even more dangerous, and an attacker could have much more possibilities of breaking into the system (several keys that open several doors).

Now, these metrics stop being environment dependant, and they would be valid for companies of any size.

In order to define more exactly our metrics, their behavior have been reflected in a graphical way. We consider a continuous model as a base for their behavior: the bigger the independent variable is, the lower will be the dependent one (security level). After that, it is only necessary to translate the behaviors into a mathematical form, trying to reflect them in the simplest form possible (for example polynomial). Continuing with the previous example, the result would be the following:

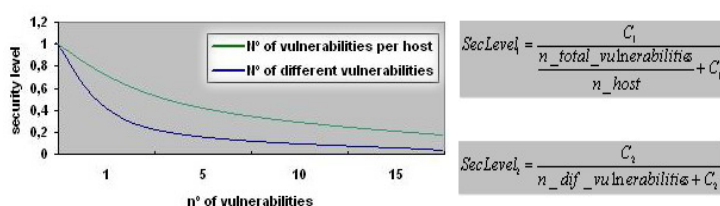


Fig 3. Behavior of the vulnerabilities, and formulas that represent the behaviors

being C_1 the number of vulnerabilities that make security level decrease down to 50%. This way, each company administrator can establish what number of vulnerabilities can tolerate in his system (the tolerance to risk may differ from one organisation to other).

2.3 Global security level

Now, it is necessary to link these metrics with the calculation of the level of global security. The global security level depends on a function involving all metrics (and time): $level = f(m1, m2, \dots, mi, t)$

Based on a study accomplished by the Computer Sciences Corporation [7], the impact of each metric within the total security of the system has been evaluated. The threats or metrics that can be automatically obtained have been extracted from this study. Then, it has been calculated the weighting or ratio scale of the impact to include in the final formula. Table 1 shows how the study has been interpreted:

	Metric	Impact	%	Tool
1	Number of high risk vulnerabilities per host	***	10.34	Nessus
2	Number of medium risk vulnerabilities per host	**	6.9	Nessus
3	Number of low risk vulnerabilities per host	*	3.44	Nessus
4	Number of different kind of high risk vulnerabilities	***	10.34	Nessus
5	Number of different kind of medium risk vulnerabilities	**	6.9	Nessus
6	Number of different kind of low risk vulnerabilities	*	3.44	Nessus
7	Number of intrusions	****	13.8	Snort
8	Time of down servers	****	13.8	Nagios
9	Time of down services	****	13.8	Nagios
Total			100	

Table1. Behavior of the vulnerabilities, and formulas that represent the behaviors

Finally, linking these metrics, their behavior within the security and their impact over the total security, the formula that will measure the security level in any given moment is obtained:

$$\begin{aligned}
 \text{Level} = & \left(\frac{C_1}{T_Down_Servers + C_1} \right) * 13.8 + \left(\frac{C_2}{T_Down_Services + C_2} \right) * 13.8 + \left(\frac{C_3}{N_Intrusions + C_3} \right) * 13.8 + \\
 & + \left(\frac{C_4}{\frac{N_tot_Vul_h}{N_Hosts} + C_4} \right) * 10.34 + \left(\frac{C_5}{\frac{N_tot_Vul_m}{N_Hosts} + C_5} \right) * 6.9 + \left(\frac{C_6}{\frac{N_tot_Vul_l}{N_Hosts} + C_6} \right) * 3.44 + \\
 & + \left(\frac{C_7}{\frac{N_dif_Vul_h}{N_dif_Vul_h} + C_7} \right) * 10.34 + \left(\frac{C_8}{\frac{N_dif_Vul_m}{N_dif_Vul_m} + C_8} \right) * 6.9 + \left(\frac{C_9}{\frac{N_dif_Vul_l}{N_dif_Vul_l} + C_9} \right) * 3.44
 \end{aligned}$$

Fig 4. Global security formula

being Ci the constants that make security level decrease down to 50% for each variable (defined by the system administrator).

3. IMPLEMENTATION

We implemented a prototype for the continuous security auditing called “munix”. This tool audits the system in a continuous and automatic way and updates periodically its data bases of vulnerabilities directly from Internet. The idea of “munix” is to obtain the information related to security from the maximum quantity of tools or information sources in order to measure the security level as accurately as possible. “munix” is not limited to only discover which are the vulnerabilities of our organisation’s IT system (as many auditors do), but also detects intrusions and monitors the most significant servers and services. Based on a Linux system, the first approach integrates three known security tools available under GNU General Public License [8]: Nessus vulnerability scanner [9], Snort Intrusion Detection System [10] (rule-based system [11]) and Nagios monitoring package [12]. Each of these tools store its results in a database. We developed some Perl scripts in order to extract the metrics, and store them in our secu-AUDIT database, this way historical data can be queried. Finally, using PHP technologies we plot the security level of each day in a graphical manner.

We monitored one network segment of the local area network of the University for one month. This subnet contained 180 hosts, most of them Windows 2000 based workstations and servers. In the third week of the experiment some security patches were installed and the overall security level raised significantly. The result can be seen in the figure 5:

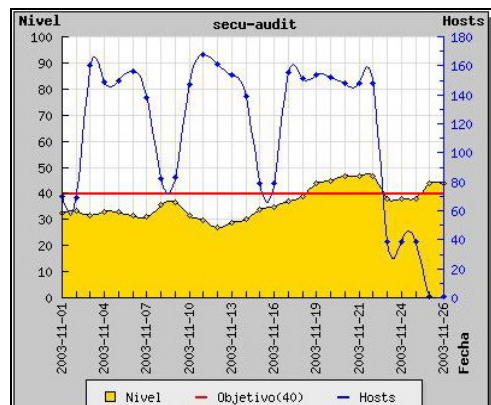


Fig 5. Security level evolution graph

The shaded area is the security level in a 1-100 scale for each day. The straight line represents the security level target we want. And finally we have added a dotted line that shows how many hosts were evaluated at that moment. It can be seen that during the four weekends (when the number of hosts decreased down to 60-80) the security level did not experiment a major change. This proves that the organisations size is independent from our methodology.

4. CONCLUSIONS AND FURTHER WORK

Continuous security auditing is possible with little effort by means of combining existing security tools. Although this type of combined system is not enough by itself, it is a good starting point to make our site a safer place. The system that has been implemented can be improved adding more security metrics so more aspects of the security of our company will be covered, and this will give higher reliability to the system.

Measuring security effectiveness is a challenging enterprise. None of the metrics can be used productively without understanding the relative importance of system security for the organisation's mission.

An improvement on the global security is achieved when a patch is installed on an organisation. Thanks to this unique system, it is possible to minimise the efforts by only choosing the most significative ones.

Future work will extend continuous auditing methodology increasing the number of security metrics evaluated. On the other hand, secu-AUDIT can be used for the risk assessment. Traditionally, risk assessment methodologies are based upon a simplistic model of risk which identifies threats and the vulnerabilities they exploit to affect a security breach. Countermeasures which mitigate the threat/vulnerability pairs are identified. Loss due to a security breach is calculated based on the probability of the threat overcoming the countermeasure and creating the breach [13].

The security CA model can be extremely helpful to asses those risks. It is useful for evaluating the effectiveness of each countermeasure against each threat/vulnerability pair, analysing the impact of the overall security level after any action is taken. The next step to be taken during this research is how to obtain the cost of improving the security level of each metric. This will depend on the technology and resources used for that meaning.

References

- [1] URL: http://www.cert.org/stats/cert_stats.html
- [2] Hayes, B., "Conducting a Security Audit: An Introductory Overview", *NebraskaCERT Conference 2004 on Computer Security and Information Assurance*, Omaha, NE USA, August 3-5, 2004
- [3] Canadian Institute of Chartered Accountants (CICA), "Research Report on Continuous Auditing", Toronto, 1999
- [4] Kogan, A., Sudit, E.F., and Vasarhelyi, M. A., "Continuous Online Auditing: A Program of Research", *Journal of Information Systems, Section of the American Accounting Association*, 1999, Vol. 13, No. 2, pp. 87-103
- [5] Groomer, K., Murthy, U., "Continuous Auditing of Database Applications: An Embedded Audit Module Approach", *Journal of Information Systems, Section of the American Accounting Association*, Spring 1989, pp. 53-69
- [6] Zurutuza, U., Uribeetxeberria, R., Lizarraga, J., and Velez de Mendizabal, I., "A Methodology for Continuous Computer Security Auditing", *Proceedings of the IADIS International Conference e-Society 2004*, 16-19 July, Avila, Spain, 2004, pp. 1024-1027
- [7] Computer Science Corporation, CSC Global Information Security Services. Security value metrics. URL: http://www.csc.com/aboutus/lef/mds69_off/uploads/Enterprise_Info_Risk_Management.pdf, Page 8.
- [8] URL: <http://www.gnu.org/copyleft/gpl.txt>
- [9] URL: <http://www.nessus.org/>
- [10] URL: <http://www.snort.org/>
- [11] Caswell, B., Beale, J., Foster, J.C., and Faircloth, J., *Snort 2.0 Intrusion Detection*, Syngress, USA, 2003
- [12] URL: <http://www.nagios.org/>
- [13] Drake, D.L., Morse, K.L., "The security-specific eight stage risk assessment methodology", *Proceedings of 17th NIST-NCSC National Computer Security Conference*, Baltimore, USA, 1994, pp. 441-450