ECAT: A CCSS-Based Tool for Enterprise-level System Configuring Automation and Assessment

Bin Wu and Andy Ju An Wang

***Abstract*—The Common Configuration Scoring System (CCSS) is a set of metrics to evaluate the security level of the severity of software security configuration issues. It is time consuming to generate a CCSS score for a computer system as it requires a large amount of manual operations to perform the evaluation on a machine. As a consequence, it is not practical for a system administrator to evaluate all the machines on an enterprise network one by one with CCSS metrics. This paper proposes a new approach to evaluate security configuration issues at enterprise level. Our solution provides a centralized management framework to remotely monitor and assess the security scores of individual machines on the network. Finally, we provide a set of well defined metrics to evaluate the security influence of the configuration issues at enterprise level. Experiments on a small e-commerce company have demonstrated the great potential of our solution and prototype tool.**

***Index Terms*—ECAT, Enterprise-level Security, Security Metrics, Configuration Evaluation, CCSS.**

I. Introduction

The Common Configuration Scoring System (CCSS) [1] is a set of measures of the severity of software security configuration issues proposed by the National Institute of Standards and Technology (NIST) [2]. CCSS is derived from CVSS [3], which is used to measure the severity of vulnerabilities due to software flaws. We can use CCSS to assist organizations in making sound decisions as to how security configuration issues should be addressed and can provide data to be used in quantitative assessment of the overall security posture of a system.

Though CCSS is sound and useful to evaluate a single configuration setting, it becomes tedious and requires large amount of manual operations when attempting to evaluate all the configurations for a computing environment. For instance, there are 590 recommendations at present for windows XP existing in CCE [12] repository, most of them providing general descriptions and technical mechanisms about various configurations. For users, they need to retrieve configuration data from their machines manually and utilize CCE [12] recommendation to calculate a final score for each of those configurations. In addition, CCSS cannot evaluate the overall configuration score for a software product as a whole, let alone for a machine. Moreover, after a user spends a large amount of time evaluating one machine, they still need to spend the same amount of time on another machine to make an evaluation

Now we would like to extend the scope of configuration scoring from individual machine level to an enterprise level, which involves hundreds of different IT resources distributed in different zones of the enterprise network. It is impossible for security professionals to go over all the machines, examining and scoring those configurations manually. A scoring algorithm must be designed to evaluate the overall configuration score of an enterprise. This paper presents a dashboard solution for security administrators of a large enterprise to monitor and assess the configuration security of its enterprise information system in a timely fashion.

In this paper, we firstly provide a semi-automatic approach to assist the evaluation of a configuration. It extracts configuration information from a machine automatically and helps security administrators assign the value of base metrics defined in CCSS [1] to calculate the configuration score. After evaluating a single machine, we will save the manual operation (e.g., assigned base metrics value) and generate a script for the future use of evaluating other machines which may have the same functionality and similar configuration. Moreover, our methods utilize client/server structure to allow centralized management of security configuration. From the server machine, security managers can access and evaluate the security configuration score of other machines existing on the enterprise network, which saves large amount of effort for the overall evaluation for an enterprise.

We also present a new model-based approach to evaluate the overall configuration security of an enterprise in this paper. We firstly construct an enterprise's IT topology model, assigning different weights and interests to all business goals and resources. Then we propose an algorithm to evaluate the security score of a single machine/software product. Finally, we produce a normalized configuration security score for the enterprise based on a well-defined metric formula.

As the implementation of our methodology, we created ECAT (Enterprise-level Configuration Assessment Tool), a semi-automated tool for enterprise configuration management. It provides a user interface to model the enterprise-level IT topology and the functionalities of semi-automatically computing overall configuration security score for an enterprise based on the constructed model.

The remainder of this paper is organized as follows. Section 2 investigates some related works. Section 3 presents the semi-automated approach for assisting the evaluation of a single configuration. Section 4 presents our model of enterprise-level IT topology and section 5 describes our metrics formulas to calculate the overall configuration security score. In section 6, we demonstrate the experiment results of our prototype tool. In the final section, we conclude briefly and discuss further research directions.

Manuscript received July 19, 2011; revised August 10, 2011.

Bin WU and Andy Ju An WANG, Southern Polytechnic State University, GA, USA

II. Related Work

Existing literature such as [4][5][6] provides different models to describe enterprise security. In particular, [6] provides a formal enterprise level model of security used for canonical representation, identification of components that need to be measured. Shi, Fuqian and Xu, Hongbiao and Wang, Haining [4] provides another modeling methodology to manage the network security in an enterprise. However, both of them did not provide any methodology to measure the security level of an enterprise.

There are some researches utilizing the enormous security data from NIST [2] to evaluate the security score of a software product or a machine. Wang, J., Wang, H., et. al., [7] provides a set of security metrics to rank attacks based on vulnerability analysis. Wang, J. and Guo, M. [8] proposes a novel methodology of using Bayesian networks to automating the categorization of software security vulnerabilities based on standardized vulnerability data. In [9] we proposed a set of well defined metrics to evaluate the overall vulnerability score of an enterprise.

In [10], Roy H. and Suvda Myagmar addresses secure configuration of reconfigurable radio systems such as in software defined radio. Chen, Huoping and Hariri, Salim [11] present a set of metrics to evaluate the dynamic configuration techniques. To the best of our knowledge, there is no existing research focusing on the scoring of IT configuration issues at an enterprise level.

III. Assisting Configuration Evaluation

A. Example of CCSS

As mentioned in section I, CCSS can help evaluate the severity level of a configuration issue. This section provides a process example of how CCSS would be used for evaluation. The issue in the example is from the Common Configuration Enumeration (CCE) [12], which provides unique identifiers to system configuration issues for operation systems and applications.

Considering the issue CCE-2776-3 in Windows XP, for instance, the following information in Table 1is provided.

In a CCE issue, the description defines the recommendation of a configuration. The parameters defined the possible values of this configuration and the technical mechanism suggests how and where you can retrieve the configuration setting from your machine.

To evaluate this issue using CCSS, security administrators must firstly explore the registration table (the path is suggested by the technical mechanism) manually and then assign the base metrics value according to the retrieved data. We assume the automatic logon is allowed in this computer, which means that anyone with physical access to the computer could boot it and be logged on with the user’s stored credentials, thus gaining unauthorized access as that user. With this assumption, security professionals can assign the value of exploitability metrics and impact metrics to generate the base score. The scoring formula can be found in [1].

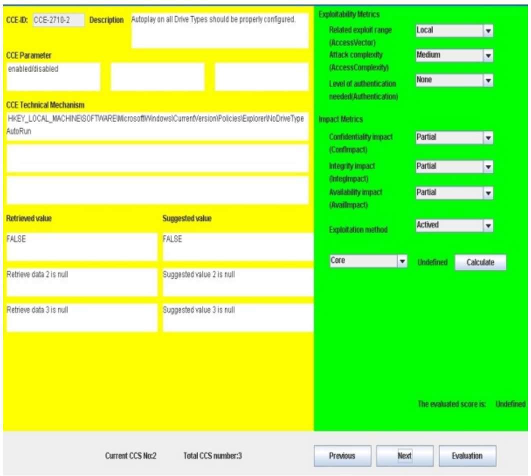
Table 1: An example of CCE

CCE-ID	CCE-2776-3
Description	Automatic Logon should be properly configured
Parameters	(1) enabled/ disabled
Technical Mechanism	(1)HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon\AutoAdminLogon

Noting that there are 590 configuration issues currently for Window XP defined by CCE [12] and many more in Unix-like systems or other applications. It is impossible for them to look up the CCE repository one by one, retrieving configuration settings from a machine manually and then further thinking about the possible situation to assign the values of metrics to get the final score. Moreover, after a long time and tedious working on a single machine, they have to move to another machine and then repeat the same process.

B. Accelerating Configuration Evaluation

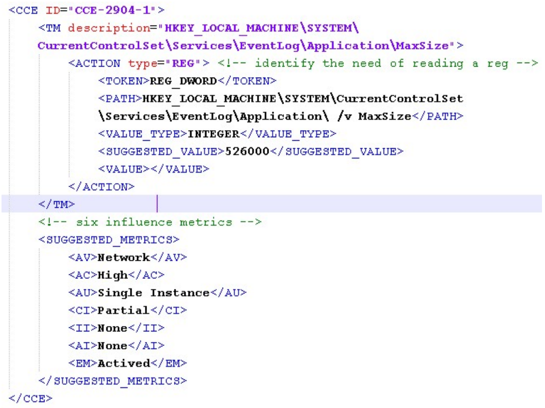
To help improve this situation, we developed ECAT to provide a semi-automatic approach for evaluating the configuration security of an enterprise. Fig1 (at the end of the paper) demonstrates the GUI of our prototype tool.



CCE-ID: CCE-2710-2
Description: Autoplay on all Drive Types should be properly configured.
CCE Parameter: enabled/disabled
CCE Technical Mechanism: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\NoDriveType AutoRun
Retrieved value: FALSE
Suggested value: FALSE
Retrieve data 2 is null
Suggested value 2 is null
Retrieve data 3 is null
Suggested value 3 is null

Exploitability Metrics
Related exploit range (AccessVector): Local
Attack complexity (AccessComplexity): Medium
Level of authentication needed(Authentication): None
Impact Metrics
Confidentiality impact (ConfImpact): Partial
Integrity impact (IntegImpact): Partial
Availability impact (AvailImpact): Partial
Exploitation method: Actived
Core Undefined
The evaluated score is: Undefined
Current CCS No:2 Total CCS number:3

Fig. 1.The prototype tool, ECAT for configuration security evaluation



```
<CCE ID="CCE-2904-1">
    <TM description="HKEY_LOCAL_MACHINE\SYSTEM\
    CurrentControlSet\Services\EventLog\Application\MaxSize">
        <ACTION type="REG"> <!-- identify the need of reading a reg -->
            <TOKEN>REG_DWORD</TOKEN>
            <PATH>HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
            \Services\EventLog\Application\ /v MaxSize</PATH>
            <VALUE_TYPE>INTEGER</VALUE_TYPE>
            <SUGGESTED_VALUE>526000</SUGGESTED_VALUE>
            <VALUE></VALUE>
        </ACTION>
    </TM>
    <!-- six influence metrics -->
    <SUGGESTED_METRICS>
        <AV>Network</AV>
        <AC>High</AC>
        <AU>Single Instance</AU>
        <CI>Partial</CI>
        <II>None</II>
        <AI>None</AI>
        <EM>Actived</EM>
    </SUGGESTED_METRICS>
</CCE>
```

Fig. 2: An example of test case

Our tool firstly loads the CCE repository and then displays the content of one issue on the left. From Fig 1, we see that for CCE-2710-2, the description, parameters and technical mechanism is presented automatically for users to look up. The next step is to retrieve configuration setting from the registration table manually. To reduce the manual effort, we implemented a reusable test case repository which could be loaded into our tool for automatically retrieving configuration settings from the machine. Fig 2 demonstrates a test case sample in the repository.

Fig 2 defines the test case used for CCE-2904-1, the <Action> tag instructs ECAT to retrieve configuration from the registration table. It also defines the location of path and the suggested value. In this way, ECAT can automatically retrieve required configuration settings and display them in the GUI. In addition, we provide suggested metrics values for each test case to assist security administrators assigning metrics values according to the configuration setting. This suggestion is based on the assumption that the value retrieved from the registration table is the same as the suggested value.

C. *Centralized Configuration Evaluation*

The approach in Section 3.B helps security administrators reduce the effort of examining all the configuration issues in a single machine. However, there are usually hundreds of machines on an enterprise network and it is impossible for them to repeat the same process on every machine.

To improve this, ECAT implements the client/server architecture which allows security administrators accessing and evaluating configuration settings from a central server. Moreover, since most machines in a network zone may play the same role and have the same functionalities, the evaluation result attained from one of them can be reused on different machines. Our tool will generate a script encapsulating the manual operations and the configuration settings on that machine into the repository. It can be imported into our tool when evaluating another new computer. In this way, when security administrators remotely evaluate the configuration scores of other machines, if the retrieved configuration settings are the same as previous example, it could automatically assign proper metrics value which users assigned before. This will significantly reduce the evaluating time of similar computers.

IV. Enterprise IT Topology Model

In this section we firstly describe the model of enterprise vulnerability topology for calculating the overall vulnerability score of an enterprise. Four principles are applied in our modeling method:

First, an enterprise is modeled as a collection of business goals. These business goals from a tree of business goals with each node a business goal associated with a different interest (weight). The root business goal must be the top of the business existence. For each business goal, it may have multiple children business goals.

Each leaf business goal utilizes a number of IT resources to reach its goal. A resource can contribute to one or more business goals. For a pair of resource and business goal, a weight is used to measure the importance of the resource contributing to that business object.

Each leaf business goal should have a vulnerability score using the transferred CCSS/CVSS base metrics. This quantitative score describes the characteristics and impacts on that business goal when it becomes unsecure due to its internal defects and external threats.

In Fig 3, ◇ means the relationship between a business goal and its sub business goals. ◆ Means the relationship between a business goal and the resources it utilizes. From this figure, it is evident that the root business goal is the Company node. It has three business goals: e-commerce, goods transportation and Internal IT system. Also e-commerce has two sub business goals: Online selling and Data backup. Two servers are used for reaching the goal of online selling: server.id1 and server.id2.

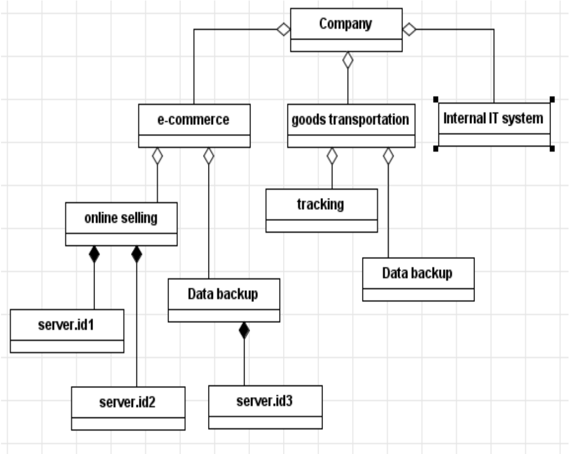


Fig. 3: Part of IT topology of an e-commerce company

Then we need to determine the interest (weight) of a business goal related to its parent business goal. The same as resource, we need to determine the weight of a resource for a business object. For instance, interest value of e-commerce is 10 (the range is from 0.0 to10.0) while internal IT system only has an interest of 6. In this way it becomes straightforward to understand the importance of a business goal/resource in relation to the whole enterprise. This weight tree could be very useful when calculating the vulnerability impact of a specific resource to the whole enterprise.

We will discuss the calculation of the business goal/resource vulnerability score in section 5.

V. Enterprise-level Evaluation

In this section, we propose a method for producing a normalized configuration security score of an enterprise based on the set-up model. We firstly calculate the configuration score of a single machine and then extend the scope to the whole enterprise.

A. *Single Machine Level Evaluation*

Now assuming we have got across the evaluation steps in section 3.B, and attained a list of evaluated scores of different configuration settings. The configurations range from different kind of applications, such as Apache Server, Firefox, Windows XP and Mysql. Using CCSS, the evaluated scores presents the direct impact to those applications which use the configurations. It's always true that for a computer running HTTP server, the configurations of Apache server and operating system have much strong influence to the functionalities of this computer compared to the configuration of a browser. In order to evaluate the configuration security score of a single machine which installs multiple applications, we must identify the influence level for different configurations.

Here we introduce the 'influence level' to identify the impact of a configuration setting. For any configuration issue, it can be core, important, related or unrelated to this machine. Now consider a computer has n evaluated configuration scores (s_1,s_2… s_n) and the influence level set assigned to them is (l_1,l_2⋯ l_n). We can then compute the security score of any machine using formula (1) below：

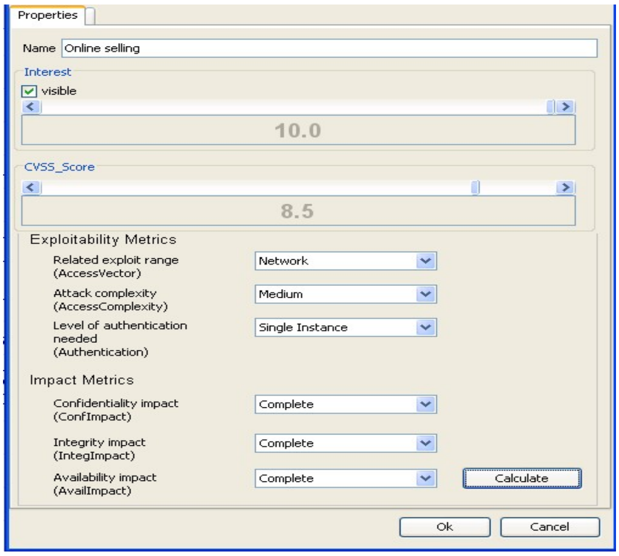
$$S = \max\{s_i \times l_i, i = n\} \quad (1)$$

In formula (1), l_i depends on the influence level of that configuration. Now we have *core*= 1.0, *important* =0.7, *related* =0.4 and *unrelated* =0.0. This value may be changed according to the future experiments. After attaining the score for a single machine, we can then compute the enterprise level configuration score.

B. *Security Score of a business goal*

In our model, we evaluate the security score of leaf business goal. For a parent business goal, the security score equals to the maximum security score among its children. Both CVSS and CCSS have the same base metrics formulas with different meaning to compute the security score. In our methodology, we transfer the calculation of base metrics in CCSS/CVSS to compute the security score of a business goal.

The base metrics of CVSS captures the characteristics of vulnerability that are constant with time and across user environments. Similarly in the measurement of an IT product's security, the Access Vector, Access Complexity and Authentication metrics capture how the security existed in a business goal is accessed and whether or not extra conditions are required to exploit it. The three impact metrics, confidentiality, integrity and availability metrics measure how a security, if exploited, will directly affect that business goal. Fig 4 is a sample interface scoring of business goal online selling.



Properties

Name Online selling

Interest

☑ visible

10.0

CVSS_Score

8.5

Exploitability Metrics

Related exploit range (AccessVector): Network

Attack complexity (AccessComplexity): Medium

Level of authentication needed (Authentication): Single Instance

Impact Metrics

Confidentiality impact (ConfImpact): Complete

Integrity impact (IntegImpact): Complete

Availability impact (AvailImpact): Complete

Fig..4: Using transferred CVSS to evaluate the vulnerability of Online selling business goal.

C. *Computing the Weight Tree*

The usage of a weight tree is to determine the importance of a resource to the whole enterprise. For instance, in an e-commerce company, the server used for online selling is far more important than a personal PC used by an employee. It is impractical for a company to adjust the configuration issues immediately for all machines because of the cost of maintenance procedure. Computing the weight tree could help security administrators focus on the most important resources and delay those not so important.

In Section IV, we have already modeled the enterprise IT topology. Each business goal has an interest (weight) value related to its parent and each resource has a weight value related to the business goal it contributes. Now we use formula (2) to calculate the weight of a resource to the whole enterprise.

$$W_i = W_p \times \frac{w_i}{\sum_{i=1}^{m} w_i}, (1 < i < m) \quad (2)$$

In formula (2), W_p is the weight of i's parent, *m* is the children number of p. The weight of root is 10. Formula (2) iterates from the root to a resource node to calculate the overall weight of the resource to the whole enterprise.

D. *Configuration Score of an Enterprise*

Finally, we calculate the overall configuration security score of an enterprise. Assume a leaf business goal has security score sb (calculated by base metrics in CVSS/CCSS) and it has n resources weighted $(wr_1, wr_2 \dots, wr_n)$ and the configuration scores are $(sr_1, sr_2 \dots, sr_n)$. The contributed security score for that business goal is:

$$s = \sum_{i=1}^{n} sb \times wr_i \times sr_i \quad (3)$$

Then we sum up all leaf business goals and normalize the score into (0-100).

$$es = \sum_{i=1}^{m} s_i \times 10 \quad (4)$$

As is the overall security score of an enterprise in terms of its configuration well-being.

VI. Experiment Demonstration

In this section, we model a small E-commerce company's IT topology and calculate the overall security score of that company. We then construct the testing environment using lab resources. We currently have constructed 20 test cases in our repository for semi-automation. The sample model is presented in Fig 5 (at the end of the paper) and all resource entities have already been scored using ECAT. The result of enterprise vulnerability analysis of is shown in Fig 6.

From Fig 5, we see that 4 servers play the most important roles in this company. Also they contribute most of the vulnerability factors. The company's configuration security score is 25.7, which implies that this company is in good security state, with relatively little configuration problem.

The ECAT tool provides the following functionalities to help security administrators monitor and manage enterprise's configuration security in many different ways.

It provides a simple but efficient way to model the enterprise IT topology that helps security administrators understand the relationship of business goals and resources, the weight of different resources and the vulnerability of a business goal (Fig 5).

It can be an accelerator for evaluating large number of configuration issues on multiple machines by constructing more test cases and analysis result reuse.

It can be an assistant tool that helps security administrators determine the priority level of different resources requiring configuration setting change. For example, both e-commerce.server.id1and e-commerce.server.id2 are out of secure states because of poor configuration. However, only one server can be patched first due to the limited resource of security professionals. By examining the enterprise IT topology and the influence factors of two servers, it is straightforward to discover that e-commerce.server.id1 should be updated first because it has higher weight and security impact.

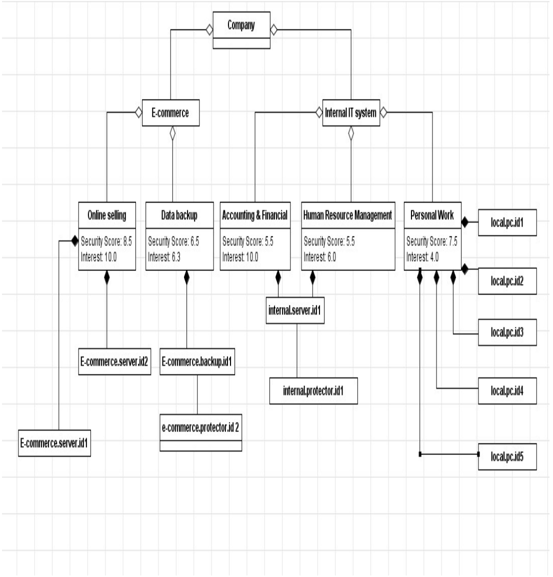
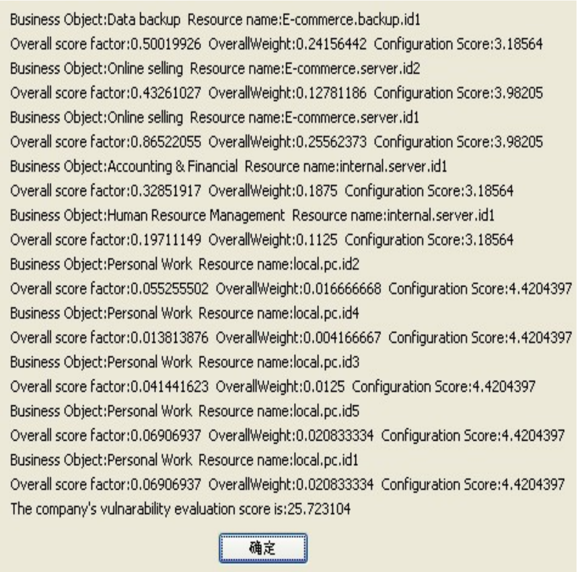


Fig.5: A simple E-commerce company IT topology model



```
Business Object:Data backup  Resource name:E-commerce.backup.id1
Overall score factor:0.50019926  OverallWeight:0.24156442  Configuration Score:3.18564
Business Object:Online selling  Resource name:E-commerce.server.id2
Overall score factor:0.43261027  OverallWeight:0.12781186  Configuration Score:3.98205
Business Object:Online selling  Resource name:E-commerce.server.id1
Overall score factor:0.86522055  OverallWeight:0.25562373  Configuration Score:3.98205
Business Object:Accounting & Financial  Resource name:internal.server.id1
Overall score factor:0.32851917  OverallWeight:0.1875  Configuration Score:3.18564
Business Object:Human Resource Management  Resource name:internal.server.id1
Overall score factor:0.19711149  OverallWeight:0.1125  Configuration Score:3.18564
Business Object:Personal Work  Resource name:local.pc.id2
Overall score factor:0.055255502  OverallWeight:0.016666668  Configuration Score:4.4204397
Business Object:Personal Work  Resource name:local.pc.id4
Overall score factor:0.013813876  OverallWeight:0.004166667  Configuration Score:4.4204397
Business Object:Personal Work  Resource name:local.pc.id3
Overall score factor:0.041441623  OverallWeight:0.0125  Configuration Score:4.4204397
Business Object:Personal Work  Resource name:local.pc.id5
Overall score factor:0.06906937  OverallWeight:0.020833334  Configuration Score:4.4204397
Business Object:Personal Work  Resource name:local.pc.id1
Overall score factor:0.06906937  OverallWeight:0.020833334  Configuration Score:4.4204397
The company's vulnarability evaluation score is:25.723104
```

Fig.6: The overall analysis of enterprise vulnerability

Our solution provides a dashboard solution that semi-automatically measures the enterprise configuration security in different scope, from a single computer to an entire enterprise (Figure 6). For instance, due to an error configuration of Apache server installed e-commerce.server.id1, the security score of MySQL becomes 9.0 (previously it was 4.8). The overall vulnerability score increases to 29.19 and the configuration score of that server increases to 7.2, which alerts the security administrators to take actions to make adjustment.

VII. Conclusion and Discussion

This paper presents a model-based semi-automated solution to quantify the enterprise level configuration security levels. As an implementation of our methods, ECAT demonstrates the following strength: 1) It provides a user interface to model the enterprise IT topology; 2) It accelerates the evaluation of a batch of configuration issues and allows the reuse of previous analysis result; 3) It quantitatively measures the overall security score of an enterprise. Our experiment on an e-commerce company has demonstrated the great potential of this tool.

There are a number of research topics that deserve further efforts. First, our tool now only supports retrieving configuration settings from registration tables. How to attain information from different kinds of configuration files requires more delicate research. Second, we would like to increase the test case repository to provide better recommendation and convenience to security administrators. Finally, the metric formulas of calculating the overall configuration security score could be improved after more experiments with real enterprise data.

Acknowledgment

This paper is based upon work supported by the National Science Foundation under Grant No. 0722157 and 0941900. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would like to express their gratitude to anonymous reviewers of this paper for their thoughtful comments and suggestions.

References

[1] Common Configuration Scoring System, http://csrc.nist.gov/publications/nistir/ir7502/nistir-7502_CCSS.pdf.

[2] NIST, http://www.nist.gov/index.html.

[3] Mell Peter and Scarfone Karen and Romanosky Sasha.Common Vulnerability Scoring System.IEE Security and Privary, 4(6):pp. 85-89, 2006.

[4] Shi, Fuqian and Xu, Hongbiao and Wang, Haining. A Representative Management Model of Network Security in Enterprise Informatization. Proceedings of the 2008 International Conference on Information Management, volume 2: pp. 304-307, 2008

[5] Zhang, Zonghua and Nat-Abdesselam, Farid and Lin, Xiaodong and Ho, Pin-Han. A model-based semi-quantitative approach for evaluating security of enterprise networks. Proceedings of the 2008 ACM symposium on Applied computing, pp. 1069-1074, 2008.

[6] Anderson, Evan and Choobineh, Joobin and Grimaila, Michael R. An Enterprise Level Security Requirements Specification Model. Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 186.3--, 2005

[7] Wang, Ju An and Wang, Hao and Guo, Minzhe and Zhou, Linfeng and Camargo, Jairo. Ranking Attacks Based on Vulnerability Analysis. Proceedings of the 2010 43rd Hawaii International Conference on System Sciences, pp. 1-10, 2010

[8] Wang, Ju An and Guo, Minzhe. Vulnerability categorization using Bayesian networks. Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research,29:1—29:4, 2010.

[9] Bin WU, Andy Ju An WANG. EVMAT: An OVAL and NVD Based Enterprise Vulnerability Modeling and Assessment Tool, In Proceedings of ACMSE, Kennesaw, GA, USA, March 24-25, 2011.

[10] Suvda Myagmar and Roy H. Campbell. *Secure Configuration for Software Defined Radio*. Ph.D. Dissertation. University of Illinois at Urbana-Champaign, Champaign, IL, USA, 2008.

[11] Huoping Chen and Salim Hariri. An evaluation scheme of adaptive configuration techniques. *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*, 493-496. NY, USA, 2007.

[12] Common Configuration Enumeartion. http://cce.mitre.org/

Bin WU was born in Fuzhou, China in 1988. He received his master degree of Information Technology at Southern Polytechnic State University, Marietta, GA, USA in 2011. His major research interests include Cloud computing, Computer Graphics and Enterprise Information System Security.

He published two conference papers focused on the security evaluation of an Enterprise's information system. Currently he works as application developer in Atlanta, USA.

Bin WU is a ACM student member He received the certificate of system analyst in China. Now he is working hard developing a full-set security metrics focus on every aspects of Enterprise security.

Andy Ju An WANG is the chair of Dept. Information Technology at Southern Polytechnic State University. His research interests include: Information Security, Embedded Software Engineering, Component-Based Development, Computer Game Design and Implementation, Software Reuse and Metrics, Web Services, and Computer Science Education. From August 2001 to June 2004, his research work was funded by Yamacraw Research Initiative, which is now changed to GEDC, Georgia Electronic Design Center. From 2004 to 2009, his research has been funded by the National Science Foundation (NSF), Microsoft Research, Cyber Object Corp., and SPSU.

He published three books as Component-Oriented Programming, Information Security Practice and EAS' 03