International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

A SYSTEM FOR COLLECTING SECURITY ALERT AND DIFFUSING CUSTOMIZED SECURITY BULLETINS

Ebot Ebot Enaw
University of Yaounde I, Cameroon
National Advanced School of Engineering
Department of Computer Sciences

Abstract

With the increasing rate of Internet usage around the world and its omnipresence in almost all aspects of daily life, preventing cyberattacks is becoming critical for governments, private companies, NGO as they need to protect their information system assets notably by collecting alerts related to latest security vulnerabilities so as to diligently take appropriate measures to mitigate them.

This paper presents a system developed to collect and classify security alerts from Incident Response Team or vendors like Microsoft, Cisco and diffuse security bulletins to IT managers of public administrations, customized to their specific organizational and technical environment. Our article is structured as follows: section 1 introduces the topic, section 2 presents some research papers related to our topic, section 3 states the problem, section 4 presents the CVE dictionary, section 5 presents CVSS, section 6 presents CPE, section 7 presents CVRF, section 8 presents OVAL, section 9 presents our solution, section 10 presents some results.

Keywords: *vulnerability, risk, security bulletin.*

1 Introduction

Our society is increasingly dependent on ICT and Internet to assist us in almost every aspect of daily life. This makes ICT assets very precious for governments and private companies. Since it is well established that the best way to protect an asset is to prevent and anticipate on attacks that can target it, all vulnerabilities have to be identified and patched in advance so that a hacker cannot exploit them to attack the asset. Although many mechanisms have been put in place to deal with security alerts namely CVE, CVSS, CPE, they don’t provide a practical way to collect, classify and diffuse security alerts. The aim of this paper is to provide governments especially those of developing countries with a methodology and practical clues to collect, classify and diffuse to their IT staff, security bulletins customized to their respective environments.

2 Related work

Some research have been done on topics related to this issue namely [1] which by leveraging the practices of system engineering to functionally decompose security management practices and identifying the basic functions and activities that need to be done and then getting appropriate technology to support the functions and activities, proposes an architecture made up of four main blocks namely the standardized enumerations of the common concepts that need to be shared, languages for encoding high-fidelity information about how to find common concepts and exchanging that information between humans and tools, sharing information through repositories of content and uniformity of adoption achieved through branding and vetting programs to encourage the tools, interactions, and content remain standardized and conformant.

[2] that disparages the lack of harmonization in vulnerability report and security bulletin, proposes a common and consistent framework called Common Vulnerability Reporting Framework (CVRF) which is an XML-based language that will enable different stakeholders across different organizations to share critical security-related information in a single format, speeding up information exchange and digestion.

[3] in a bid to compare security level of open and close source software, proposes a methodology and metrics namely the mean time between vulnerability disclosure. Their methodology consists of selecting a sample of popular closed and open software that serve the same purpose and with a sufficiently large set of vulnerability data available, analyze them regarding their vulnerabilities, as published in the National Vulnerability Database (NVD) of the National Institute of Standards and Technology (NIST) in terms of the number of vulnerabilities, the disclosure rate, the development of disclosure over time, and the severity of vulnerabilities.

Their studies reveal that (a) the mean time between vulnerability disclosures was lower for open source software in half of the cases, while the other cases show no differences, (b) in contrast to literature assumption, 14 out of 17 software packages showed a significant linear or piecewise linear correlation between time and the number of published vulnerabilities, and (c) regarding the severity of vulnerabilities, no significant differences were found between open source and closed source.

[4] proposed a dictionary of known information security vulnerabilities and exposures called CVE (Common Vulnerability Enumeration), [5] proposed a free and open standard called CVSS (Common Vulnerability Scoring System) for assessing, communicating the characteristics and impacts of IT vulnerabilities. [6] proposes a framework called OVAL (Open Vulnerability and Assessment Language) that can be used to specify automated security tests.

3 Research problem

Since to attack a system, an attacker has to find a vulnerability on that system and exploit it, every IT administrator has to discover all vulnerabilities inherent to its assets and patch them as fast as possible. Given that in Cameroon, most IT managers of public administrations don't have the necessary cybersecurity skills and capabilities in terms of crawling and sorting out vulnerabilities repository of vendors or other CIRT nor evaluating the risk of these vulnerabilities in their specific environments coupled with the low and unsteady nature of Internet bandwidth. It is very difficult for IT managers to collect security alerts from vendors, or other CIRT and apply patches. It is in a bid to address these shortcomings that we developed a system that centralizes security alerts that are collected automatically or manually from several sources, classify them by risk level and targeted assets and then diffuse those security alerts to IT managers through the web, email and even SMS for high risk vulnerabilities based on the assets of their Information system. Thus, IT managers don't have to worry anymore about crawling multiple vulnerability repository available on the Internet and analyzing the risk level of these vulnerabilities. They only receive security alerts related to their assets, with their corresponding risk evaluation customized to the specificities of their IT environment, from our system. To develop the application we used concepts like CVSS, CVE, CPE and CVRF that will be explained in the subsequent sections.

4 CVE

CVE is a dictionary of publicly known information security vulnerabilities and exposures managed by the **MITRE** organization.

CVE assigns a unique identifier to each vulnerability called CVE-ID or CVE names or CVE number or CVEs. In this paper we will use the terminology CVE-ID.

This identifier enables data exchange between security products and provides a baseline index point for evaluating coverage of tools and services.

The syntax of this identifier is as follows:

CVE prefix + Year + Arbitrary digits where Year represents the year when the ID were assigned to a particular vulnerability, and “arbitrary digits” represents any digits assigned to the vulnerability.

CVE-ID are assigned by CNA (CVE Numbering Authority). CNA request CVE-ID pool from MITRE and uses the pool to assign CVE-ID numbers to researchers and information technology vendors for inclusion in first-time public announcements of new vulnerabilities.

CVE has become very popular as it has been adopted by many entities including vendors, Computer Incident Response Team and vulnerability assessment service.

5 CVSS

The Common Vulnerability Scoring System is a free and open standard under the custodial of Forum of Incident Response and Security Teams (FIRST) that provides an open framework for communicating the characteristics and impacts of IT vulnerabilities. CVSS is composed of three metric groups: Base, Temporal, and Environmental, each consisting of a set of metrics.

The base metric group represents the intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments. This contains six metrics: Access Vector, Access Complexity, Authentication that capture how the vulnerability is accessed and whether or not extra conditions are required to exploit it and Confidentiality impact, Integrity impact, Availability impact which measure how a vulnerability, if exploited, will directly affect an IT asset, where the impacts are independently defined as the degree of loss of confidentiality, integrity, and availability.

The temporal metric group represents the characteristics of a vulnerability that change over time but not among user environments. It contains three metrics: exploitability, remediation level and report confidence.

The environmental metric group represents the characteristics of a vulnerability that are relevant and unique to a particular user's environment. This group contains five metrics: collateral damage potential, target distribution, confidentiality requirement, integrity requirement and availability requirement.

Generally, the base and temporal metrics are specified by vulnerability bulletin analysts, security product vendors, or application vendors because they typically have better information about the characteristics of a vulnerability than do users. The environmental metrics, however, are specified by users because they are best able to assess the potential impact of a vulnerability within their own environments.

This framework has been adopted by many entities namely: vulnerability bulletin providers, software application vendors and researchers.

Normally, temporal and environmental groups are not mandatory in CVSS, however since our system is aimed to provide customized security bulletins to public administrations with contextualize risk level assessment, these two groups will be mandatory in our application.

6 CPE

CPE is a structured naming scheme for information technology systems, software, and packages based on the generic syntax of Uniform Resource Identifiers (URI) and managed by the NIST (US National Institute for Standards and Technology). CPE includes a formal name format, a method for checking names against a system, and a description format for binding text and tests to a name. The CPE dictionary that contains a repository of software, systems and packages is provided in XML format and is available to the general public. The XML schema is constantly updated so as to improve it. The latest release of this schema is 2.3.

7 CVRF

Common Vulnerability Reporting Framework is an XML-based language designed by the Industry Consortium for the Advancement of Security on Internet (ICASI), that enables different stakeholders across different organizations to share critical security-related information in a single format, speeding up information exchange and processing. The most recent version CVRF 1.1 was released in 2012 and overcomes issues found in the previous version including the duplication of XML when product and vulnerability tree were merged. For purposes of efficiency and conformance to standards, our system generates the security bulletin in CVRF format before converting them to more readable formats ie .doc and pdf.

8 OVAL

Open Vulnerability and Assessment Language (OVAL) provides a way to standardize how to assess and report upon the machine state of computer systems. OVAL includes a language to encode system details, and an assortment of content repositories held throughout the community which in turn enables interoperability among security tools and services.

With OVAL, tests can be automated and specified in a precise language to check for the presence of a particular vulnerability on an asset (software/system), a particular configuration on an asset, the presence of a specific patch which can be useful for vulnerability management solutions and compliance verification.

9 Our solution

8.1 Methodology

The methodology used for our system consists of five main steps:

1. Collect security alerts from vendors or incident response team
2. Collect information related to software/hardware of public administrations as well as those of their designated cybersecurity focal point
3. Evaluate the risk level of vulnerabilities collected with regards to the specificities of public administration IT environments
4. For each vulnerability propose recommendations to fix it
5. Associate vulnerabilities to IT assets
6. Elaborate a customized security bulletin for each administration to fit their IT assets ;
7. Diffuse those security bulletins to public administrations focal point.

Following this methodology and as depicted in the figure below, we came up with an architecture that comprises four main modules namely Vulnerability Collector, IT asset collector, Security bulletin manager, and Dispatcher which will be described in the subsequent sections. The architecture of our system is depicted in the figure below:

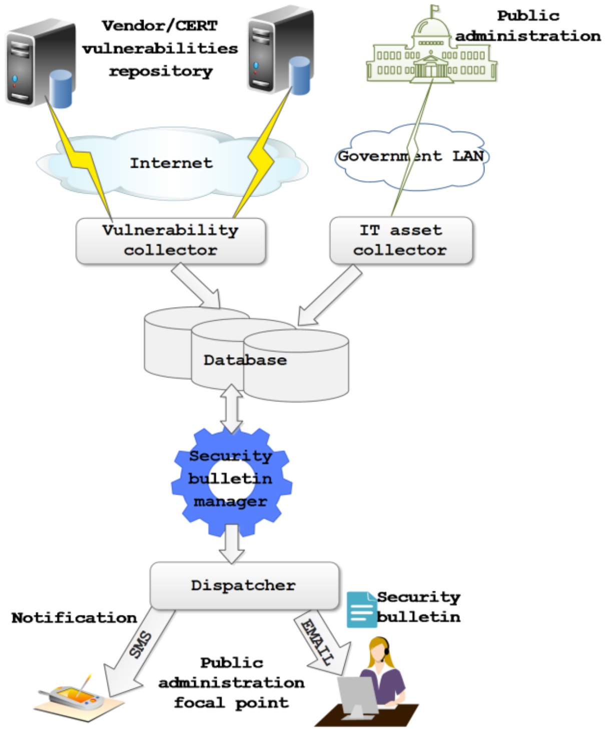

Figure 1: Architecture of the system

8.2 Presentation of the environment

8.2.1 Vulnerability Collector

This module maps to the first step of our methodology. It collects rough vulnerabilities from vendors and incident response team and structures them in a specific format that is stored in a database. This format has several fields presented in the table below:

Table 1: vulnerability data structure

N°	Field	Description
1	Vulnerability ID	It is a code that identifies the vulnerability in the system
2	CVE ID	It is the CVE ID assigned by MITRE or any CNA
3	Source	It is the vendor or researcher or incident response team that

		published the vulnerability
4	Description	It is a description of the vulnerability
5	Affected assets	ID of products that are subjected to that vulnerability
6	Date of publication	Date on which that vulnerability was published by the source
7	CVSS base vector	It is the base vector of the vulnerability
8	CVSS temporal vector	It is the temporal vector of the vulnerability
9	CVSS environmental vector	It is the environmental vector of the vulnerability. Initially this value is not assigned as it depends on the environment of the vulnerable asset. Its value will be defined by the security bulletin manager module.
10	Patch	Guidelines to follow in order to fix the vulnerability

This module collects vulnerabilities through two ways:

- since most of the vendors or incident response teams provide vulnerabilities through RSS feeds or any other XML format, this module automatically downloads them and structures them according to the aforementioned format ;
- For vendors or incident response teams that do not present vulnerabilities in a structured format like XML, this module provides a Graphic User Interface (GUI) whereby a system administrator can manually fill the information about these vulnerabilities.

For vulnerabilities whose patches have been released, the patches are downloaded and stored in a local repository. Links to the patch in that repository are automatically mapped to the patch section of their corresponding vulnerability. This module also provides a console whereby for each vulnerability the system manager will give detailed guidelines on how to fix the vulnerability or contain it in case of zero day.

8.2.2 IT asset collector

This module maps to the second step of our methodology. It collects three types of data about information system of public administration:

- Characteristics of the assets (software/system/hardware) of the information system
- Description of the architecture of the IT system and network, the relations between the constituencies (hardware/software) as well as the core business of the company so as to be able to have a good understanding of the environment and get an objective evaluation of the risk level of the vulnerabilities within that particular environment
- Information about the designated cybersecurity focal point of the public administration

Asset's data once collected are structured in the following format

Table 2:Asset database structure

N°	Field	Description
1	Asset ID	It is a string identifying that particular software in the system
2	Name	It is the name of the asset
3	Description	It is a brief description of the asset
4	CPE ID	It is the CPE ID allocated to the software
5	Editor	It is the manufacturer of the software
6	Release	It is the release number of the software
7	End of support date	It is the date by which the software will no longer be supported by its editor
8	Confidentiality level	It expresses the requirement in terms of confidentiality with respect to the environment of the asset and the core business of the company
9	Availability level	It expresses the requirement in terms of availability with respect to the environment of the asset and the core business of the company
10	Integrity level	It expresses the requirement in terms of integrity with respect to the environment of the asset and the core business of the company

It is worth mentioning that the values of confidentiality level, availability level and integrity level are assigned by the system administrator based on the analysis of the whole IT environment as well as the core business of the administration. These values are therefore not fixed, they can be modified as the IT architecture or the core business change. A threshold risk value should also be defined for critical asset so that when a vulnerability with a risk level that hits that value is found, an alert is immediately sent to the focal point through SMS.

The focal point's data structure is outlined in the table below:

Table 3: focal point's data structure

N°	Field	Description
1	Name	Name of the designated focal point
2	Administration ID	ID assigned to the administration to which the focal point belongs
3	Position	Position occupied by the designated focal point in his

		administration
4	Phone number	Phone number of the focal point. It is through this phone number that alerts might be sent to him through sms
5	Email	Email address of the focal point through which alert or security bulletin will be sent to him

8.2.3 Security bulletin manager

This module encompasses the steps 3,4,5,6 of our methodology.

This module associates vulnerabilities to assets indexed in our database and evaluates their CVSS score in their specific environments. When vulnerabilities are collected and structured in the format presented in the last section, this module finds all assets indexed in the system that are prone to that vulnerability and calculates the CVSS score of each vulnerability in the context of these assets based on the values reported by the source as well as the IT environment of the administration that owns the asset and its core business which are expressed in the availability, integrity and confidentiality values of the asset.

Secondly this module designs a customized security bulletin for each administration that encompasses only security alerts that target their assets only, with the estimation of their risk level within that particular environment and guidelines to fix the vulnerability. The security bulletins are generated first in an XML format derived from the template of CVRF and later converted to RTF and pdf. Depending on his technical skills, the focal point can display the security bulletin either in the executive summary form where only specific fields are presented or in the detail form where all CVRF fields are presents. A sample of the security bulletin of a typical public administration in the executive summary form is presented in Annex1. In order to provide focal points with an easy way to compare the risk level of vulnerabilities that are listed in the security bulletin, we assigned a single value to each vulnerability which is evaluated as the norm of the vector whose coordinate are the base, temporal and environmental scores through the formulae:

$$\text{Norm} = \sqrt{(bscore)^2 + (tscore)^2 + (escore)^2}$$

Where *bscore* represents the CVSS base score, *tscore* represents the CVSS temporal score and *escore* represents the CVSS environmental score.

8.2.4 Dispatcher

This module maps to the last step of our methodology. It sends security bulletins to focal points by email. It also provides an interface whereby once authenticated with his credentials, the focal point can access all security bulletins and alerts related to the information system of his administration. This interface is very important in the Cameroon context where Internet bandwidth in public administration is low and unsteady but a government LAN exists. So, thanks to this interface, public administrations can access their security bulletins in the local network with a good bandwidth.

In case of vulnerabilities with high risk levels that target critical assets of an administration, this module notifies the focal point of the administration concerned by SMS so that he can immediately log into his account to get the link to the patch or the procedure to contain or mitigate the vulnerability.

8.3 Technical environment

To develop our solution we used:

- A server with the following characteristics: 16 GB RAM, 1To Hard disk, Intel Xeon 1.87Ghz *16
- **Netbeans 7.3**: Netbeans is a popular free IDE (Integrated Development Environment) that supports many languages like Java and PHP ;
- **Java** and **Tomcat 7.0.34.0** ;

10 Results

For the purposes of this paper, we will present some data gathered from a security bulletin designed by our system for a particular administration whose name will not be revealed for security purposes. This security bulletin has fifteen vulnerabilities of three types: hardware, applications and operating system. Their ID and their CVSS value are presented in the table below:

Table 4: Sample vulnerabilities data

Id	Base score	Temporal Score	Environmental Score	Norm
OPERATING SYSTEM				
CVE-2014-2241	6.8	6.8	9.1	13.24
CVE-2014-0004	6.9	6.9	8.5	12.94
CVE-2014-0647	2.1	2.1	6.6	7.24
CVE-2013-4711	4.3	4.3	7.7	9.81
CVE-2014-0899	6.5	6.5	8.8	12.72
APPLICATION				
CVE-2014-1297	5	3.3	0.9	6.06
CVE-2014-0324	9.3	6.2	1.4	11.26
CVE-2014-1715	7.5	5	1.1	9.08
CVE-2013-2192	3.2	2.1	0.5	3.86
CVE-2014-1514	9.3	6.2	1.4	11.26
HARDWARE				
CVE-2014-1467	5	3.3	0.9	6.06
CVE-2013-3689	7.8	5.2	1.1	9.44
CVE-2013-3690	6.8	4.5	1	8.21
CVE-2013-6976	6.8	4.5	1	8.21
CVE-2013-5669	7.8	5.2	1.1	9.44

The figure below depicts a pictorial representation of the data presented in the table above.

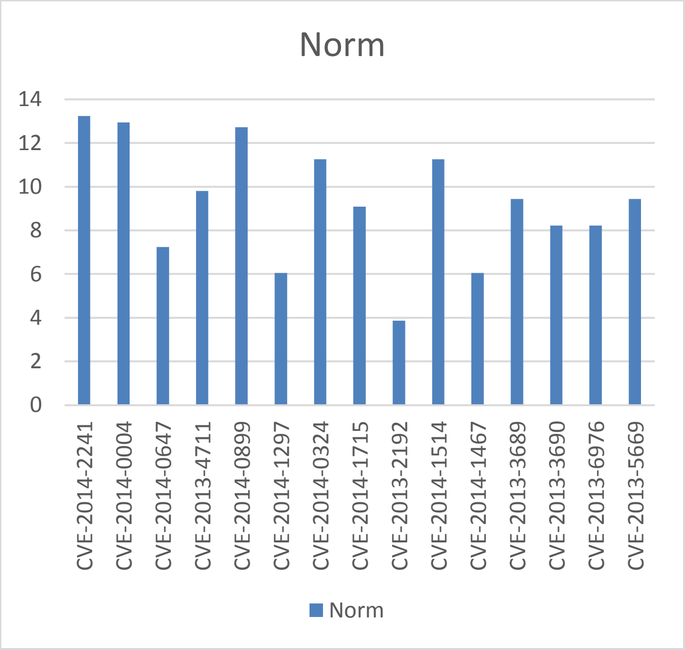

Figure 1: Vulnerabilities risk values

According to the data above, operating system vulnerabilities have the highest risk value: their average risk value is 11,2 while the average risk value of application vulnerabilities is 8,30 and that of hardware vulnerabilities is 8,27.

It should be noted that operating systems fall under two categories: Servers and PCs , with PCs representing 95%. The PC category comprises mostly Microsoft windows, which for the most part are unlicensed and as such can not be updated online resulting in the high risk value.

11 Conclusion and future work

Since hackers exploit vulnerabilities to attack systems and due to the ever growing importance of ICT in our daily life, IT managers have to identify vulnerabilities their assets are prone to and mitigate them before attackers exploit them, to gain unauthorized access to their systems. Vulnerability alerts are scattered across many repositories on the Internet and are usually presented in an unclassified way making it impossible for IT administrators to filter only vulnerabilities that relate to their IT assets. Moreover, the evaluated risk level of these vulnerabilities presented in the aforementioned repositories doesn't take into account the specificities of the environment of each information system and there is no way for an IT administrator to be notified instantly when a high risk vulnerability targeting one of his assets is published. In an effort to address these concerns, we developed a methodology and a system for the dissemination of customized security bulletins to public administrations. The system collects and centralizes security alerts from different sources (vendors and incident response team) as well as their countermeasures, parses and structures them in a specific format. It also collects informations about IT materials (hardware/software) of public administrations, matches vulnerabilities collected to materials and evaluates their risk level in their specific environments so as to prioritize the vulnerabilities. It then builds customized security bulletins for each administration with vulnerabilities that relate to their IT assets only, with their corresponding risk level and sends it to its focal point through email, web or SMS. The system offers many advantages namely: it provides a single repository of vulnerabilities stemming from differents sources, it provides a way to follow-up the IT assets of public administrations, it provides a means to notify an administrator in real time when a high risk vulnerability targeting one of his asset has been identified.

Future work can include leveraging the OVAL framework to permit the follow-up of the implementation of countermeasures by IT administrator.

Annex

Annex 1: Sample of security bulletin

Table 5: Sample security bulletin

Security bulletin N° 0014 of Ministry of …………………………………… **Date: 02nd April 2014**				
N°	**Assets referred**	**description**	**Risk** **(**base Temporal Environment Norm**)**	**Resolution**
CVE-2014-2241	Canonical Ubuntu Linux 13.10	The (1) cf2_initLocalRegionBuffer and (2) cf2_initGlobalRegionBuffer functions in cff/cf2ft.c in FreeType before 2.5.3 do not properly check if a subroutine exists, which allows remote attackers to cause a denial of service (assertion failure), as demonstrated by a crafted ttf file.	6.8 5.2 1.7 8.72	The problem can be corrected by updating your system to the following package version: **Ubuntu 13.10:** libfreetype6 2.4.12-0ubuntu1.1

CVE-2014-0004	Canonical Ubuntu Linux 13.10	Stack-based buffer overflow in udisks before 1.0.5 and 2.x before 2.1.3 allows local users to cause a denial of service (crash) and possibly execute arbitrary code via a long mount point.	6.9 5.3 1.73 8.86	The problem can be corrected by updating your system to the following package version: **Ubuntu 13.10:** udisks2 2.1.0-4ubuntu0.1 udisks 1.0.4-8ubuntu1.1 **Ubuntu 12.10:** udisks2 2.0.0-1ubuntu1.1 udisks 1.0.4-6ubuntu0.1 **Ubuntu 12.04 LTS:** udisks 1.0.4-5ubuntu2.2
CVE-2014-0647	Apple iPhone OS 7.1	The Starbucks 2.6.1 application for iOS stores sensitive information in plaintext in the Crashlytics log file (/Library/Caches/com.crashlytics.data/com.starbucks.mystarbucks/session.clslog), which allows attackers to discover usernames, passwords, and e-mail addresses via an application that reads session.clslog.	2.1 1.6 1 2.82	It has been reported that this issue has been fixed. Upgrade to version 2.6.2, or higher, to address this vulnerability.
CVE-2013-4711	AccelaTech BizSearch 3.2 on Linux Kernel	Cross-site scripting (XSS) vulnerability in Accela BizSearch 3.2 on Linux and Solaris allows remote attackers to inject arbitrary web script or HTML via unspecified vectors.	4.3 3.3 1.3 5.57	Apply the patch according to the information provided by the developer or distributor.
CVE-2014-0899	aix	ftpd in IBM AIX 7.1.1 before SP10 and 7.1.2 before SP5, when a Workload Partition (aka WPAR) for AIX 5.2 or 5.3 is used, allows remote authenticated users to bypass intended permission settings and modify arbitrary files via FTP commands.	6.5 5 1.6 8.35	IBM has assigned APARs to this problem: http://www.ibm.com/support/docview.wss?uid=isg1IV51420 http://www.ibm.com/support/docview.wss?uid=isg1IV51421 Fixes are available. The fixes can be downloaded via ftp from: ftp://aix.software.ibm.com/aix/efixes/security/wparcre_fix.tar

ISSN:2319-7900

References

[1] Robert A.Martin, "Making security measurable and manageable" in *Military Communication Conference* 2009.

[2] MikeSchiffman, "The Common Vulnerability Reporting Framework,"*whitepaper*, 2011.

[3] MikeSchiffman,"The Missing manual: CVRF 1.1 ,"*whitepaper*, 2012.

[4] MITRE, "Common Vulnerability Enumeration" Technical report, cve.mitre.org, 2014.

[5] US National Institute of Standards and Technology, "The Common Vulnerability Scoring System", nvd.nist.gov.

[6] MITRE, "Open Vulnerability and Assessment language" , oval.mitre.org.

[7] MITRE, "Common Platform Enumeration," *cpe.mitre.org..*

[8] Guido Schryen, "Security of open source and closed source software: An empirical comparison of published vulnerability" AMCIS, 2009.

Biography

Dr. EBOT EBOT ENAW obtained his B.Eng hons degree from Liverpool University in Electronic Engineering in 1989. He later obtained an M.Eng degree in Telecommunication Engineering from The University of Manchester England in 1991. He returned home where he was recruited in the University of Yaounde I, as an assistant lecturer. He pursued his university studies and obtained a PhD in Computer Sciences from the National Advanced School of Engineering of the University of Yaounde I, where he is currently a senior lecturer. His area of specialization include: computer network security, cryptography and formal specification and verification; theorem proving and model checking. He has published some research articles in international journals namely *Formal model of a group key agreement protocol. Journal of Computational Technologies, 7(4):4–20, 2002*. In 2006 he was appointed Director General of the National Agency for Information and Communication Technologies Cameroon, a position he occupies till date. Major activities of the agency include amongst others: securing the Cameroon cyberspace through three key services: Computer Incidence Response Team (CIRT), Public Key Infrastructure (PKI) and Computer Security Audits.

Dr. EBOT EBOT ENAW may be reached at ebotenaw@yahoo.com