Implementing Automated Vulnerability Assessment in Large Network Environments

J. Duplančić
Department of Professional Services and IP Business Solutions
T-Com, T-HT Croatian Telecom
Draškovićeva 26, HR-10000 Zagreb, Croatia
Phone: (+385-1) 4913 800, E-mail: jerko.duplancic@t-com.hr

M. Golub
Department of Electronics, Microelectronics, Computer and Intelligent Systems
Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, HR-10000 Zagreb, Croatia
Phone: (+385-1) 6129 967, E-mail: marin.golub@fer.hr

Abstract – Vulnerability assessment (VA) tools are usually GUI applications requiring human interaction which is not often desirable while performing tests in large network environments. Effectiveness of this setup can be limited due to many reasons such as: both configuration time and duration of assessment are both increased, test traffic passing inter-network boundaries can be mistakenly identified by IDS as DoS attack, intervening firewalls may block or silently drop test traffic causing incomplete and erroneous test results, local tests performed on individual host may require setup and management of additional security credentials. Lightweight software solution based on automation, distribution and code portability is developed to address majority of "GUI setup" problems and to provide higher degree of automation. Additionally, integration with SIM solution is provided and results are evaluated.

I. INTRODUCTION

Today's large network environments are dynamic, complex, heterogeneous, administratively scattered systems, comprised of many different technologies, devices, applications, servers, protocols and people, thus introducing new challenges into a security process. Given the size and complexity of such environments, vulnerabilities and corresponding exploits are likely to appear. Without the specific solution that takes into account the size of such environment, validating security requirements can be a semi-accurate and time consuming task. In case there is not an effective approach to this problem it could be very difficult to maintain acceptable level of security. The implementation of the components of the security process such as prevention and protection, detection and monitoring, and response, all depend on accurate information about the state of the environment and its resources. Therefore, it is crucial to have well established methods to keep up with security.

The process which detects, identifies and quantifies vulnerabilities is called vulnerability assessment (VA). The purpose of VA is to address risks created by vulnerabilities before these vulnerabilities can be exploited by an adversary. In TCP/IP network-centric real-world practice, the main component of VA is usually called the vulnerability scan – an authorized and planned process trying to exploit known vulnerabilities in order to determine the state of security of target networks, applications and hosts. To rephrase it, the main objective of VA is the audit process launched against network environment (its systems, devices and applications) which compares current security state with security level defined in organizational security requirements. VA reveals potential issues and feeds the results into security compliance systems such as security policy and patch management for final evaluation and possible re-evaluation of security procedures already in place. It is obvious that in large networks with hundreds of heterogeneous resources verifying the conformance of each and every resource can be an extremely difficult task to accomplish.

It is important to know that VA – as a part of a much broader security process – is not limited to network environments only. It can be applied to any area of security process, be it an organization, network, or any other area of interest.

The intent of this paper is to try to identify problems found while applying VA to large TCP/IP networks and to offer a possible solution in order to minimize the impact. Eventually, it may lead to an improved implementation of organizational security and its requirements.

The main characteristics of the proposed solution are automation, distributed operation and deployment, centralized management, source code portability (for easier multi-platform installation and configuration), and integration with other security solutions.

II. NETWORK VULNERABILITY ASSESSMENT IN PRACTICE

To understand better the practical aspect of VA it is important to give a more detailed explanation of the steps VA is comprised of. From the real-world network-assessment perspective these steps refer to the methods of usage of VA software based tools, commonly called *vulnerability scan* tools [1], and other actions VA auditor has to perform while conducting assessment. VA steps are, as follows:

1) Preparation: Involves the people who authorize VA coverage, the setup of monitoring systems [2] in case vulnerability scan goes out of control (e.g. the crash of server or router), gathering information about the specifics of environment, reviewing security policies and requirements, establishing general mindset about VA

targets such as data-center network, user workstations, external or internal firewall, payment gateway – credit card payment server, router itself, specific operating system, application or remote service. During the preparation, auditor also has to define the attack source location – it is the point (or points) located inside a network or at network perimeter - chosen for launching vulnerability scan against network targets. For example, it can be a wireless access point, dial-up connection, external firewall, user workstation subnet, etc. Therefore, from attack source location's point of view, vulnerability scan is often said to be internal or external. Due to its distributed deployment nature the proposed solution is transparent to this qualification since it is able do conduct vulnerability scan from many attack locations at once, actually mixing internal and external attack source locations, if needed.

2) Inventory discovery and enumeration: In this step auditor uses manual or automated methods to obtain as much information as possible about devices, operating systems, enabled network protocol stacks, open ports, network topology, versions, applications, services, IP addresses. Inventory discovery is a very important VA step because every target that is not discovered or is falsely identified can cause inconsistencies in overall security picture, effectively causing other VA steps to generate less accurate results. For inventory discovery step to perform correctly it is required to have a method to circumvent detrimental effect of inter-network devices. For example, firewalls, routers, traffic shapers, anti-DoS appliances and active IDS devices, all can influence attack traffic, to render the vulnerability scan useless or, even worse, to provide erroneous results. Again, the proposed solution tends to be much more "immune" to this problem because inter-network attack traffic flow can be relatively easily managed in order not to pass across subnets.

3) Vulnerability scan: A controlled process of vulnerabilities exploitation using predefined known-vulnerabilities library and corresponding automated attacks implemented in software, which exploits these vulnerabilities against target inventory, is called vulnerability scan (through this paper scanning software will be referred to as the "VA agent"). These automated attacks are called attack-plug-ins and are usually implemented in some form of high level computer language [3]. Automation is an important part of the vulnerability scan considering vast amount of resources included in the scan target list. It would be almost impossible to accomplish vulnerability scan on hundreds, thousands or on a even larger amount of targets and not having automation in place. Every decent vulnerability scanner has automation implemented in its core as "default" feature.

Since vulnerability scan relies only on known vulnerabilities in order to attack targets – meaning it requires information about vulnerabilities, publicly announced by vendors for proper operation – it cannot be a good security measure against zero-day exploits. Zero-day exploit indicates a situation when an attacker knows how to exploit vulnerability for which a patch has not yet been published, or the vulnerability itself is still unknown to others (e.g. vendors, security communities etc.). The best measure to mitigate the risk of zero-day exploits is to have effective security procedures and policies established – as a part of a good security process.

The ability to reuse results of previous vulnerability scans is another feature contributing to effectiveness because it significantly reduces both scan time and attack traffic. On the other hand, the use of previous scan results to build network security state may introduce interference from the old or changed network state, leading to a situation where new results become inaccurate. Time between two consecutive full-scans – e.g. scans that don't reuse previous results - has to be determined depending on dynamics of network changes, for example: network with DHCP clients is highly dynamic because the same IP address can have many different devices attached to it in a relatively short period of time. Production network is much less dynamic. It is unlikely to see frequent changes of IP addresses of DNS and incoming mail servers, or database backend, between two consecutive full-scans.

4) Scan results and report generation: The final step of VA is obtaining scan results and report generation based on the organizational role of individual recipients. The results are often stored in a database for easy access and statistics, for export into other security solutions, and for reuse by a new scan.

III. LIMITATIONS OF VULNERABILITY ASSESSMENT IN LARGE TCP/IP NETWORKS

The time and effort needed to accomplish VA steps are directly proportional to the size of the network since the expected number of scan targets usually increases accordingly. In a larger network more devices and applications have to be taken into account in order to have accurate VA results which can lead to increased overall complexity. Without proper architectural and organizational measures, increased VA scope can render results practically useless or, at best, semi-accurate.

As the size of the network increases, the reliability of VA decreases. The reason for this lies in the fact that large networks are often internally physically and logically separated into a number of interconnected autonomous systems – e.g. WAN networks, disaster tolerance redundant networks, workstation subnets – all are good examples of this occurrence. This configuration also results in a scattered administration so that many subnets have their own administration staff and policies. An increased number of subnets decreases the efficacy of VA due to detrimental effect of inter-subnet devices, hosts and servers, or applications carrying and possibly mangling VA attack-traffic flow or even the traffic payload itself. It is not uncommon to find a large amount of attack traffic to be completely blocked on its path. As a result, vulnerability scan generates inaccurate view of security, as shown in Figure 1.

Every VA step can have some or all of the following limitations, depending on actual network and VA setup:

A. Preparation step limitations

- It is harder to gather general information about network intended for scan.
- It is more difficult to contact every staff member involved in VA and to coordinate scan among them.

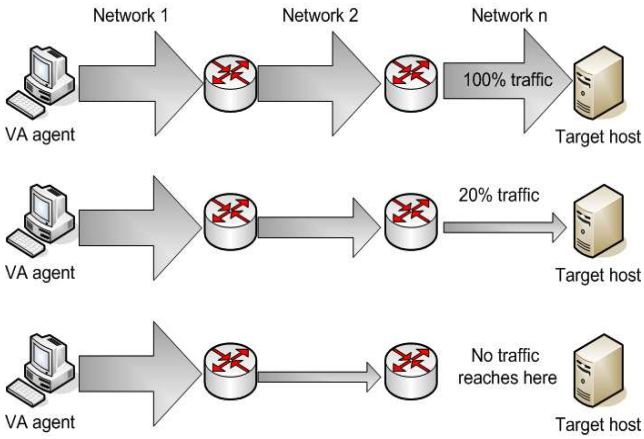

Figure 1. Detrimental effects of inter-network devices to attack traffic.

- Setup of scan logistics becomes more complex because configuration of monitoring system becomes more difficult to setup. There is also a possibility for deployment of additional monitoring systems along attack traffic paths.
- Attack traffic paths are more difficult to determine due to limitations imposed by business logic or local security policies every individual subnet or administrative domain can have in place.

B. Inventory discovery step limitations

- Number and influence of inter-network devices becomes much greater.
- Increased VA scope limits effectiveness of identification of known discovered devices and applications. Additionally, a chance to miss detecting undocumented resources or hidden subnets largely increases.
- In practice, inventory discovery refers to software tool with focus on GUI interface ("Graphical User Interface"), to make the operation easier. While it can be good for human to interact with such an interface, setup and configuration of new scans become boring and error prone as number of vulnerabilities increases (usually on daily basis), and when scanned resources tend to frequently relocate or change their role.
- Network protocols timeouts additionally extend scan time. UDP port scan is particularly sensitive to this problem because inter-network devices are required to allow both UDP and specific ICMP packets to pass through, which may be not the case.

C. Vulnerability scan limitations

- The limiting factor is the influence of increased number of inter-network and other security devices.
- Due to a highly intrusive characteristic of the attack traffic, it is largely affected by network security-aware devices. Intrusion detection systems may trigger alerts and obfuscate any real attacks from being detected during vulnerability scan; intrusion prevention and anti-DoS systems may falsely block the attack traffic; inter-network and host based firewalls may block "non-standard" traffic effectively allowing pass only for traffic destined for open ports of known services belonging to a particular subnet, blocking the rest of the attack traffic. All security devices may need to be tuned not to block the attack traffic but it can be a tremendous task to accomplish, considering the number of both involved people and devices in the process.
- Traffic rate limiters may control the rate of the attack traffic flow thus increasing the time required for vulnerability scan to finish.
- In practice, vulnerability scan refers to GUI software tool, although it is not uncommon for inventory discovery and vulnerability scan (and even report generation) to be implemented in single software tool. In large networks this type of implementation may not be adequate, because accurate vulnerability scan results depend on distributed attack source locations..
- OSI Layer 2 access lists (VACL) [4], configured on network switches, may block the attack traffic destined to highly accessed servers. Highly accessed servers are expected to be more suspect to intrusions. Despite the fact that security level of these servers is stricter, in case of possible break-in, they must not become the stepping stone for further attacks.
- False-positive results of vulnerability scan tend to increase, which may lead to constant reviewing and missing the real from the fake alerts.
- Some attack plug-ins can disrupt the operation of critical production servers requiring an auditor to carefully design attack configuration templates.
- Very intrusive scans may panic end users. To avoid this situation a good practice is to inform everyone affected about the scan schedule and the scope.

D. Scan results and report management limitations

- Scan results are affected by accumulated errors in all previously described steps.
- As the size of network increases, both the number of recipients and the number of reports increase. Management and distribution of reports among multiple recipients become more complex. In order to generate a particular report it is important to have mechanism that keeps track of scan targets, types of scan and the recipients.

E. Other limitations

- Additional expenses for maintenance and problem-fixing.
- Software bugs.

VA in large networks may require a considerable amount of time and resources for its operation, making it almost impossible to achieve any reasonable performance and accuracy in a timely manner. It may be difficult for a VA agent to launch all selected attack plug-ins using only a single network location as the source of the attack traffic. The impact of the previously described detrimental effects can block or disrupt the attack traffic. Also, there is a high possibility to gather different inventory information and scan results while conducting VA from two or more topologically different network locations targeting the same

resources. If this problem fails to be considered it may result in a distorted view of the network security state.

It is obvious that only when VA solution includes some sort of automation, distributed operation and centralized management, it can help to reduce the majority of described limitations. The primary objective in design of the proposed solution is based on that assumption, and extended with code portability.

IV. SYSTEM DESIGN AND IMPLEMENTATION

The intent to design and implement the proposed solution is not to develop a brand new vulnerability scan tool, but on the contrary, to reuse existing popular tools and to extend them with automation, distribution, and centralized management. The decision is not to modify the source code of the scan tool itself, even if it could be possible. This way, the solution can easily keep track with the future official versions of the scan tool, without spending any effort in patching and porting the scan tool to platforms it is intended to run on, be it closed- or open-source scan tool. Only the solution's own wrapper code that provides interface to the scan tool has to be carefully examined and changed in case the types of input or output parameters of the scan tool change. Also, with this approach the existing attack plug-in library is reused, thus contributing to quick deploy of the solution throughout the network. Graphical user interface is used only when creating attack templates, otherwise the solution operates autonomously.

For the abovementioned purpose, the popular vulnerability scanner Nessus [5] is used. In the context of the solution, the vulnerability scanner and the wrapper software are together referred to as the VA agent.

The proposed solution design consists of five major components, as follows:

A. Automation

Automation saves time and obfuscates operational complexity, but any good automation also has to have alerting systems and other fail-safe measures in case of the system failure. Automation capabilities of the proposed solution refer to:

- Attack plug-ins are updated automatically from the official plug-in repository and then automatically distributed to VA agents.
- Vulnerability scans are triggered by hosting system's scheduler (e.g. *cron* daemon).
- Timestamped results are archived in a local file system for later reuse and are also stored into external relational database.
- The report is generated according to a particular recipient filter and immediately delivered to the recipient by the mail system.
- The automation module operation is monitored by an external monitoring solution [6].

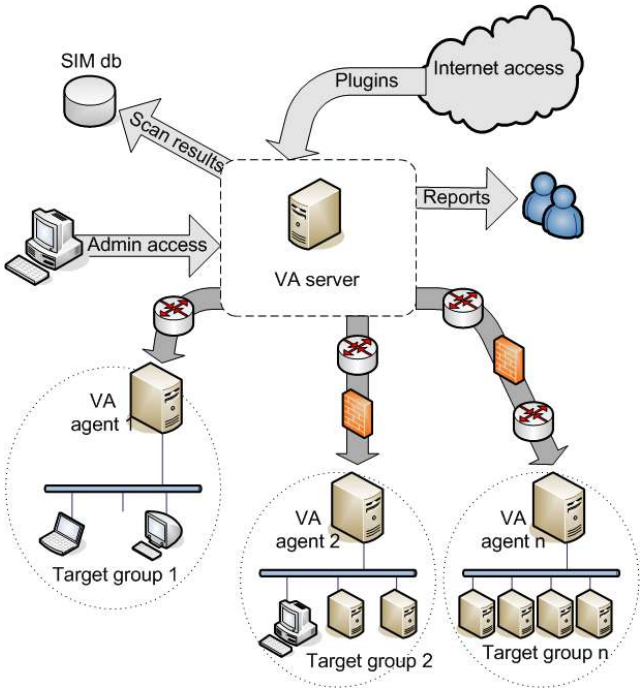

Figure 2. System design of the proposed solution.

B. Distributed operation

Distributed operation of the solution provides a better network coverage, because the central VA server handles multiple VA agents distributed amongst different network locations at the same time, i.e. perimeter, subnets, and hosts, as shown in Figure 2. The impact of inter-network devices is considerably reduced because only the command traffic from the VA server to VA agents has to pass through inter-network devices. The attack traffic between a VA agent and its targets mainly remains unaffected because it is local to the VA agent's subnet.

Though, there is a small possibility that local host firewalls and VACLs have detrimental impact on the attack traffic. This situation can be avoided by deploying VA agent at target host itself, thus bypassing the obstacles, but the implementation may require administrative approval.

C. Centralized management

The control of the VA operation becomes easier when using the centralized VA server for managing VA agents, plug-in updates, report archiving and launching of the scheduled vulnerability scans. The VA server is located at a single location in a network and it controls the operation of every deployed VA agent. All changes made to the VA server configuration are automatically reflected on the overall VA operation. Here again, from the VA server to VA agents only legal command traffic passes across inter-network devices and therefore remains undisrupted. This greatly contributes to the accuracy of scan results because the attack traffic reaches its targets directly, i.e. the VA agent is on the same local subnet as targets. Thus, no additional configuration of inter-network devices is necessary.

D. Portability

To ensure a multi-platform code portability of the VA server and the wrapper code, the solution is written in an interpreted, high-level, object-oriented programming language – Python [7], and some portions of the code are written in UNIX shell scripts. The code was successfully tested on several brands of UNIX and its derivatives, i.e. FreeBSD, NetBSD, OpenBSD, Debian, Solaris, and Tru64. Also, the effort to run and test the code on the other platforms, i.e. Interix, Fedora Core, and Red Hat Enterprise Linux, is underway.

The portability of the vulnerability scanner and the software it depends on is achieved using Nessus 2.x version, published under GPL license. The only reason for using the free 2.x version of Nessus is because it is supported under the NetBSD Packages Collection (*pkgsrc*) framework [8] for building third-party software on UNIX-like systems. In this case, pkgsrc framework provided uniform installation and patch management of Nessus and its dependencies across the heterogeneous network environment. The result of using pkgsrc is a highly portable VA agent which can be deployed on already existing servers, or hosts within a network.

The new version of Nessus (3.x) is closed-source but it is still free for personal use and currently runs only on FreeBSD and Linux-kernel based operating systems, and is not currently considered for integration with the solution.

E. Integration with SIM system

The solution can parse and export the vulnerability results to an external database for further processing by other security solutions. For this purpose, a database schema for a SIM solution [9] (Security Information Management) is used.

V. RESULTS

While performing the scan to a particular target, it is important to know that results vary greatly when attack source location or number of VA agents change, due to the impact of different network environment characteristics, as previously discussed. The results for the same target tend to become more comprehensive and more accurate as the attack source location topologically moves closer to the target or the number of distributed VA agents increases. The presented solution tries to eliminate the need for manual setup, deployment, and the manual control of multiple vulnerability scans.

To estimate effectiveness of the solution, several vulnerability scans are performed. Vulnerability scan configuration parameters (i.e. selected plug-ins, protocol timeout values, attack traffic rate, time of day when launching scan, and others) remain the same throughout all scans.

The scanned network environment includes four class C networks consisting of 28 subnets total, 336 IP-enabled devices (i.e. hosts, servers, network equipment, etc.), and 715 services. The distributions of devices and services among the networks are shown in each corresponding table. To simplify the presentation of the results, the distribution among subnets is not given. Vulnerabilities are introduced by intentionally installing outdated and unpatched service software.

The first scan is launched from the single attack-source location external to the network, using the single VA agent. This scan is expected to be under the most detrimental impact of inter-network devices, as shown in Table I.

The results of the first scan reveal an interesting situation: the external vulnerability scan discovered 3 visible networks out of 4, but the majority of subnets and corresponding devices and services remained undiscovered. It is a common setup for networks to have the perimeter preventive measures deployed (firewalls, active IDS, etc.). While an auditor usually knows internal topology of a network, an adversary cannot do that very easily (however, auditors sometimes perform so called "blind scans", thus mimicking the behavior of a real adversary).

The second scan is launched from the single attack-source location, this time internal to the network, using the single VA agent. The results from Table II show how the visibility of the network and the efficacy of scan may increase as location changes. Here, the impact of inter-network devices still affects the attack traffic, but at a lower rate. It is important to know that in this case the scope of the vulnerability scan only theoretically remains the same because a different location may be affected by different inter-network devices and thus may reveal new portions of the network (e.g. workstations can access internal HTTP proxy but outsiders cannot do that). Therefore, the picture of the network, i.e. topology and visible resources, depends heavily on the actual location the network is being "watched" from.

The final scan is launched from multiple attack-source locations, internal to the network, using distributed VA agents in the majority of subnets. The results from Table III reveal additional portions of the network. In this scan, the picture of the network is probably the most accurate as can possibly be. Yet, not every host or service is guaranteed to be discovered, because local host firewalls or VACLs may block the attack traffic. If the administrative policies and the operating system platform permit the installation of the VA agent directly to a host, it is possible to completely bypass any firewall access lists. The only requirement is to have non-blocked command traffic from the VA server to the VA agent. The time needed to conduct the scan is decreased, because the impact of stateful firewalls dropping packets of the blocked attack traffic is decreased.

TABLE I
VULNERABILITY SCAN RESULTS FROM THE SINGLE EXTERNAL LOCATION

Discovered resources	Net 1	Net 2	Net 3	Net 4
Subnets	1/6	0/20	1/1	1/1
Devices	12/78	0/95	35/148	12/15
Services	4/142	0/273	30/317	13/19
Vulnerabilities	8	0	5	27

TABLE II
VULNERABILITY SCAN RESULTS FROM THE SINGLE INTERNAL LOCATION

Discovered resources	Net 1	Net 2	Net 3	Net 4
Subnets	5/6	8/20	1/1	1/1
Devices	61/78	43/95	148/148	15/15
Services	120/142	138/273	206/317	19/19
Vulnerabilities	39	56	19	27

TABLE III
VULNERABILITY SCAN RESULTS FROM THE MULTIPLE DISTRIBUTED LOCATIONS

Discovered resources	Net 1	Net 2	Net 3	Net 4
Subnets	6/6	18/20	1/1	1/1
Devices	75/78	81/95	148/148	15/15
Services	136/142	217/273	264/317	19/19
Vulnerabilities	41	75	33	27

In this case, the timeouts of the TCP/IP protocol suite [10] have no considerable effect on duration of the scan.

The rate of false positive results depends on accuracy of the attack plug-ins, and discussion about it is out of the scope of this paper.

VI. LIMITATIONS

Although the solution is successfully tested in lab networks, it is still in a development phase. The main limitations are, as follows:

1) VA server is a single-point-of-failure: If the VA server becomes unavailable, none of VA agents can launch the scan, and in the case of crash of the VA server during the scan, the results are lost. To avoid this problem, the integration with a HA system [11] (High Availability) may be considered.

2) VA agent lacks fully automated operation mode: Current version of the VA agent doesn't have a mechanism to automatically retrieve a complete template, configuration, and attack schedule from the VA server, in order to survive network changes or outages while conducting the scan. Currently, it just executes commands received from the VA server.

3) No built-in secure communication between the VA server and VA agents: The solution can utilize the existing encryption module implemented into the vulnerability scanner for encrypting the command traffic between the VA server and VA agents, but the solution itself doesn't have any built-in encryption, thus making it difficult to use other types of vulnerability scanners not having encryption already integrated.

VII. CONCLUSION

The vulnerability assessment is a challenging and comprehensive task. In large network environments it is almost impossible to obtain accurate security state of the vast number of heterogeneous systems. By implementing automation, distribution, centralized management and portability, the proposed solution extends existing technologies, and adapts itself according to the characteristics of the environment. While definitely not without its limitations, the solution can help make the vulnerability assessment easier. But, more work is needed to develop a more stable release of the solution, and to avoid the introduction of new vulnerabilities into the environment. For mission-critical environments, integration of HA and more automated mode of operation is mandatory.

REFERENCES

[1] J. Foristal, G. Shipley, "Vulnerability Assessment Scanners", Network Computing, 2001., available from: http://www.networkcomputing.com/1201/1201f1b1.html

[2] R. Bejtlich, "The Tao of Network Security Monitoring", Addison-Wesley, ISBN 0321246772, 2004.

[3] R. Deraison, "The Nessus Attack Scripting Language Reference Guide", 2000. available from: http://www.virtualblueness.net/nasl.html

[4] "Securing Networks with Private VLANs and VACLs", Cisco Systems, Inc., Document ID: 10601, 2005. available from: http://www.cisco.com/

[5] Nessus Vulnerability Scanner, available from: http://www.nessus.org/

[6] Service Monitoring Daemon, available from: http://www.kernel.org/software/mon/

[7] Python Programming Language, available from: http://www.python.org/

[8] NetBSD Packages Collection, available from: http://www.pkgsrc.org/

[9] Open Source Security Information Management, available from: http://www.ossim.net/

[10] W. R. Stevens, "TCP/IP Illustrated, Volume 1", 7th Edition, ISBN 0201633469, p. 77-81, p. 297-337, *Addison-Wesley Inc.,* USA, March 1996.

[11] A. L. Robertson, "The Evolution of The Linux-HA Project", *UKUUG LISA/Winter Conference High-Availability and Reliability*, Bournemouth, UK, 25-26 February, 2004.