Super Security
Automated SCAP Security Compliance and Common Vulnerabilities Evaluation

Jordan A. Caraballo-Vega[1], George Rumney[2], John Jasen[2]
[1]*University of Puerto Rico at Humacao, Department of Mathematics*
[2]*High Performance Computing, NASA Goddard Space Flight Center, Mail Code 606.2*

Objectives

It is NASA's responsibility to secure and protect its computer systems and enormous amounts of data. Therefore, in order to enhance the security for the NASA Center for Climate Simulation (NCCS), tools to continuously monitor our High Performance Computing systems were implemented.

Figure 1. NCCS High Performance Computational Environment.

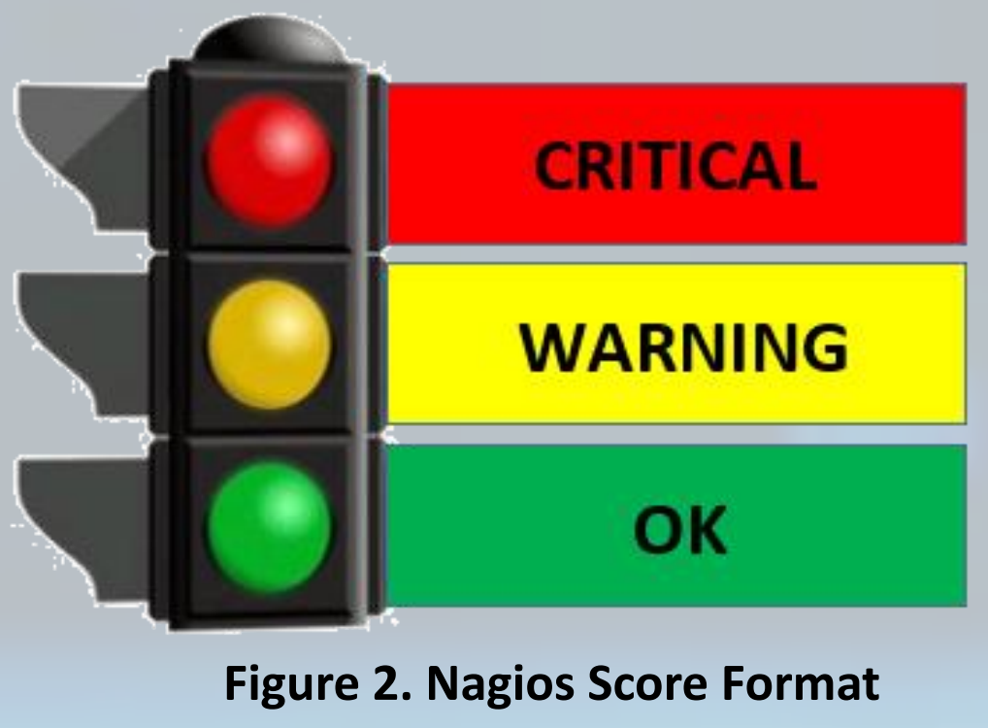
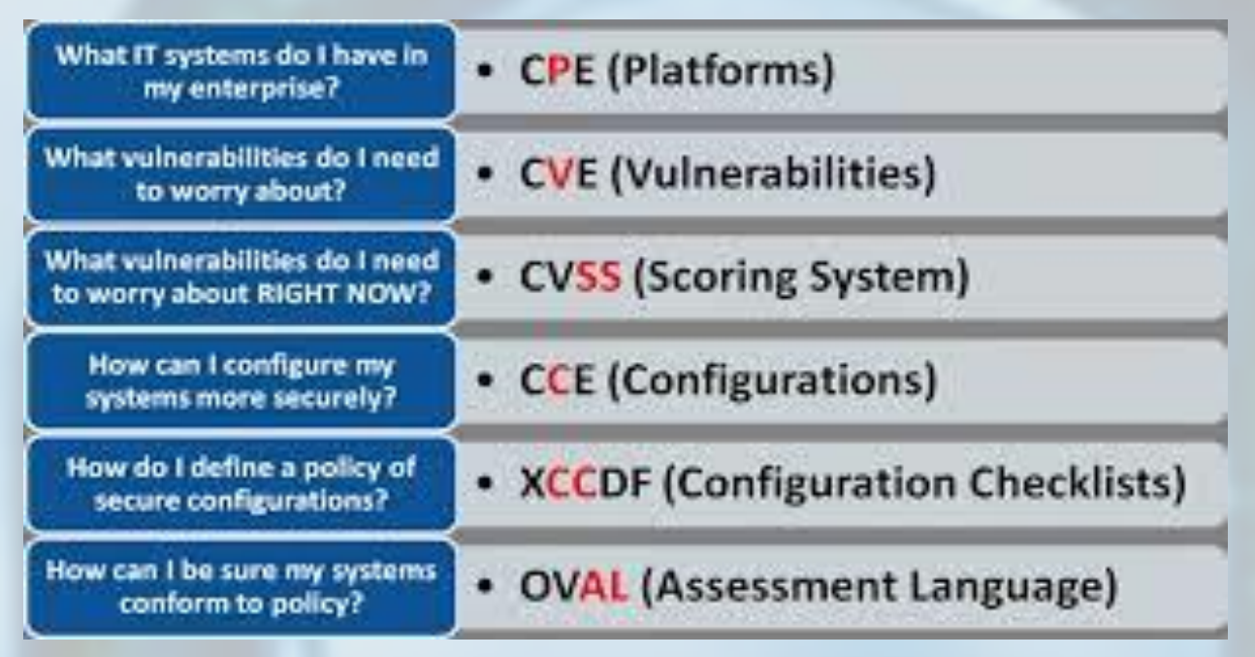


Figure 2. Nagios Score Format

Our aim is to: compare the efficiency of different assessment tools, keep a time trending record of the results, identify specific rules that need to be addressed urgently, and expand the use of this automated SCAP through all NCCS operational systems.

Background

SCAP – protocol developed by NIST designed to audit and assess a target system with a defined set of configuration requirements and rules. The two main components developed during this work are the Baseline and Vulnerabilities audits.



What IT systems do I have in my enterprise?	• CPE (Platforms)
What vulnerabilities do I need to worry about?	• CVE (Vulnerabilities)
What vulnerabilities do I need to worry about RIGHT NOW?	• CVSS (Scoring System)
How can I configure my systems more securely?	• CCE (Configurations)
How do I define a policy of secure configurations?	• XCCDF (Configuration Checklists)
How can I be sure my systems conform to policy?	• OVAL (Assessment Language)

Figure 3. Acronyms of files used to execute SCAP

Common Configuration Enumeration (CCE) - unique identifiers to security-related system configuration issues used in baseline audits.

Common Vulnerabilities and Exposures (CVE)- unique identifiers that enable data exchange between security products; specifically cybersecurity vulnerabilities.

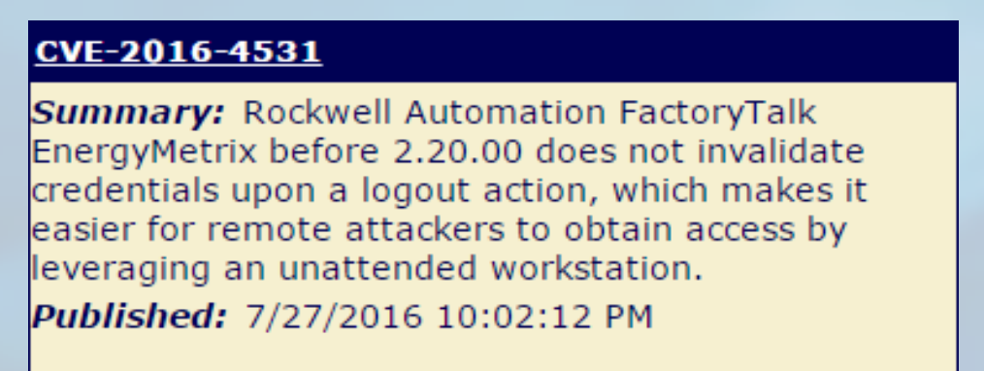

CVE-2016-4531
Summary: Rockwell Automation FactoryTalk EnergyMetrix before 2.20.00 does not invalidate credentials upon a logout action, which makes it easier for remote attackers to obtain access by leveraging an unattended workstation.
Published: 7/27/2016 10:02:12 PM

Figure 4. Example of an html CVE description

They both include:
- Identifier Number – specific rule id CCE-2715-1/ CVE-1999-0067
- Description – human readable description
- References – points sections of the documents

Extensible Configuration Checklist Description Format (XCCDF) - a structured collection of security configuration rules for some set of target systems.

Open Vulnerability and Assessment Language (OVAL)- an effort to standardize how to assess and report upon the machine state of computer systems.

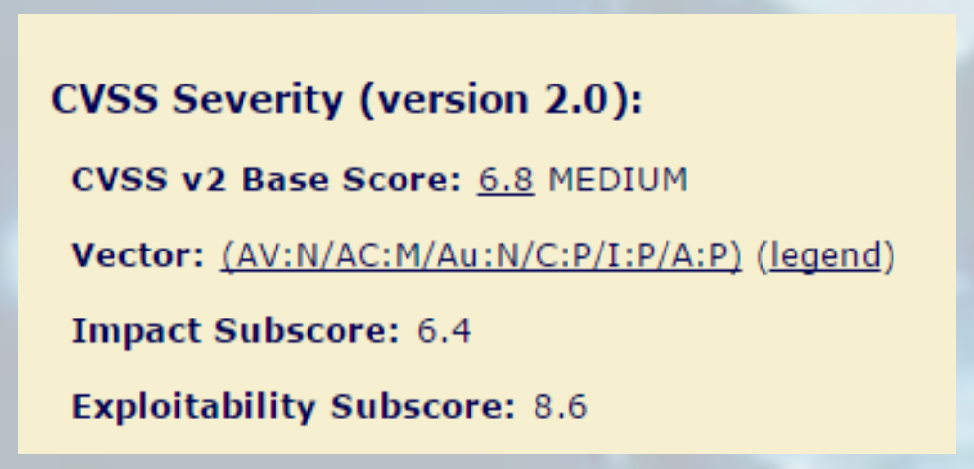

CVSS Severity (version 2.0):
CVSS v2 Base Score: 6.8 MEDIUM
Vector: (AV:N/AC:M/Au:N/C:P/I:P/A:P) (legend)
Impact Subscore: 6.4
Exploitability Subscore: 8.6

Figure 5. Example of an html CVSS description

Common Vulnerability Scoring System (CVSS)- a quantitative model to ensure repeatable accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the scores.

OpenSCAP CIS-CAT

OpenSCAP is an auditing tool that utilizes the Extensible Configuration Checklist Description Format (XCCDF). It is based on a framework of libraries to improve the accessibility of SCAP and enhance the usability of the information it represents.

CIS-CAT tool to perform assessments of target systems according to CIS Benchmark rules. By discovering any lack of conformance to CIS Benchmarks, CIS-CAT offers a powerful tool for analyzing and monitoring the effectiveness of internal security processes.

OpenSCAP	CIS-CAT
Not supported in all distributions	Needs Java to run
Faster	Slower
Lacks of Benchmark files	Does not have severities information
Open source	Not free

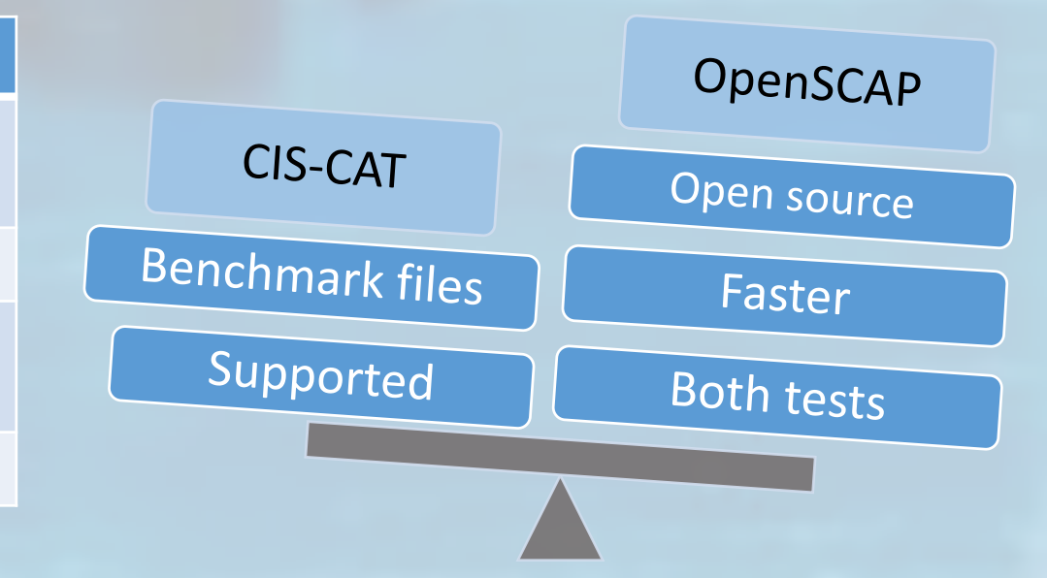


Figure 6 and 7. Comparison between OpenSCAP vs. CIS-CAT

Software

Perl scripts were written in order to execute and automate the audit process. Bash scripts were implemented in order to report to Nagios, while Python scripts were developed to produce the graphs.

Nagios is a passive check service that provides monitoring of all mission-critical infrastructure components including applications, services, operating systems, network protocols, systems metrics, and network infrastructure. The data sent to Nagios include the score and the pass/failure of priority rules (identified by the administrator of the system), and trend analysis of results over time.

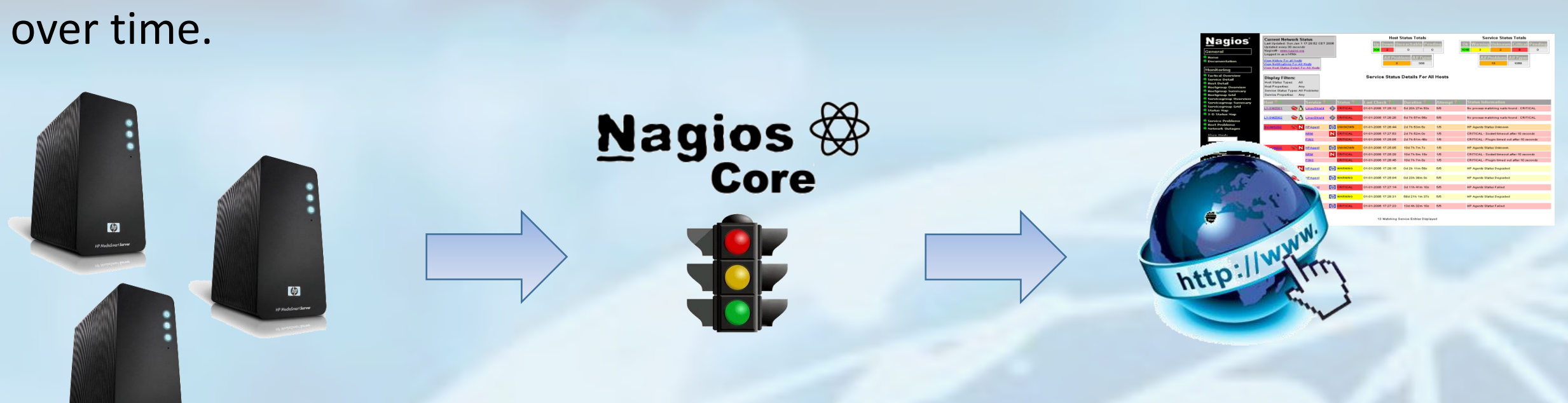


Figure 8. Diagram of Nagios infrastructure. At first, servers send the info to be processed by Nagios. Date is processed and then delivered to the internet interface

Method

Baseline Audit

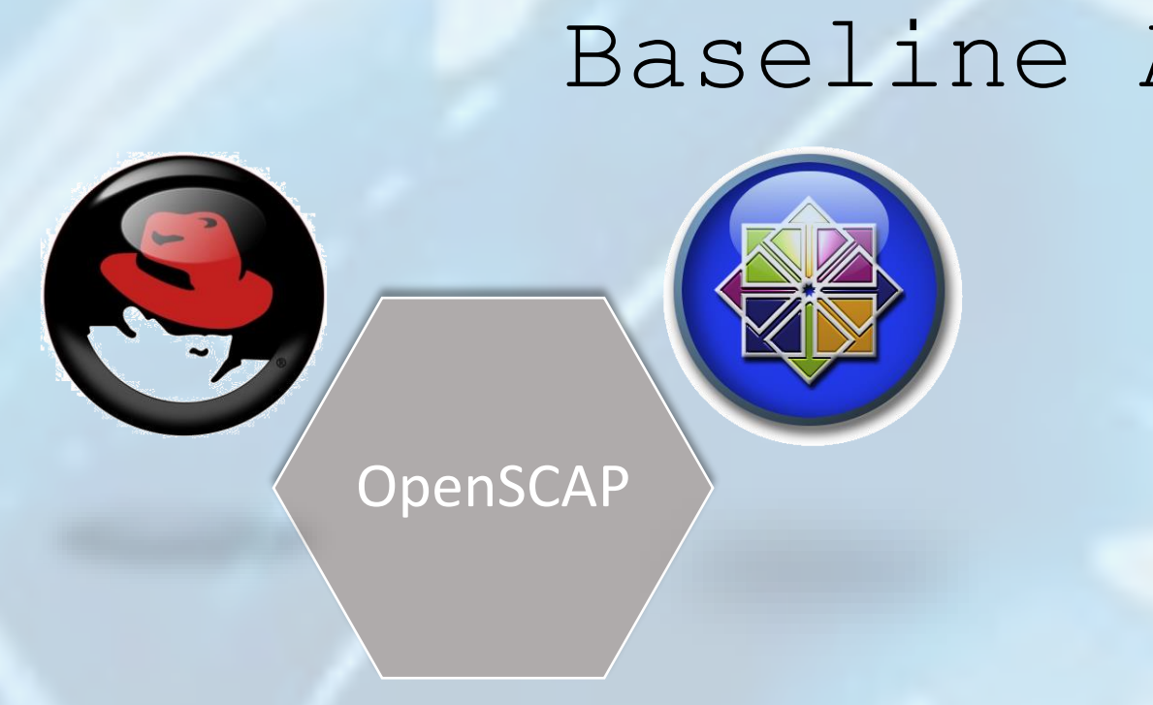


Figure 9. Operating systems supported by OpenSCAP based on the availability of benchmark files: CentOS and RHEL.

For Baseline tests OpenSCAP supports RHEL 6/7 and CentOS 6/7. On the other hand, CIS-Cat tool supports SLES 11/12, CentOS 6/7, RHEL 6/7, FreeBSD, Ubuntu 14/16, Solaris and Debian 8.

Figure 10. Operating systems supported by CIS-CAT from upper center to right: SLES, CentOS, Debian, FreeBSD, Solaris, Ubuntu, and RHEL.

Vulnerabilities Audit

Figure 11. Operating systems logos of found OVAL files: SLES, Debian, Solaris, and Ubuntu. CentOS uses RHEL modified feed

OpenSCAP is the auditing tool used to asses the vulnerability profiles for every operating system used in this work. OVAL files for RHEL, SLES, Ubuntu, and Solaris were found. Debian files are very recent, so further tests need to be performed in order to prove their accuracy. After the audit is done, severities (from low to high) are taken from the resulting xml, or the online CVSS principal page.

RHEL, SLES, Debian, and Ubuntu files work properly without any changes. However, to audit CentOS, RHEL feed must be heavily modified. Previous work done by Summer Intern Graham Mosley prove that this is possible by replacing the file with the corresponding CentOS version, changing the platform information, and signing keys to the CentOS equivalent on RHEL feed.

General Procedure

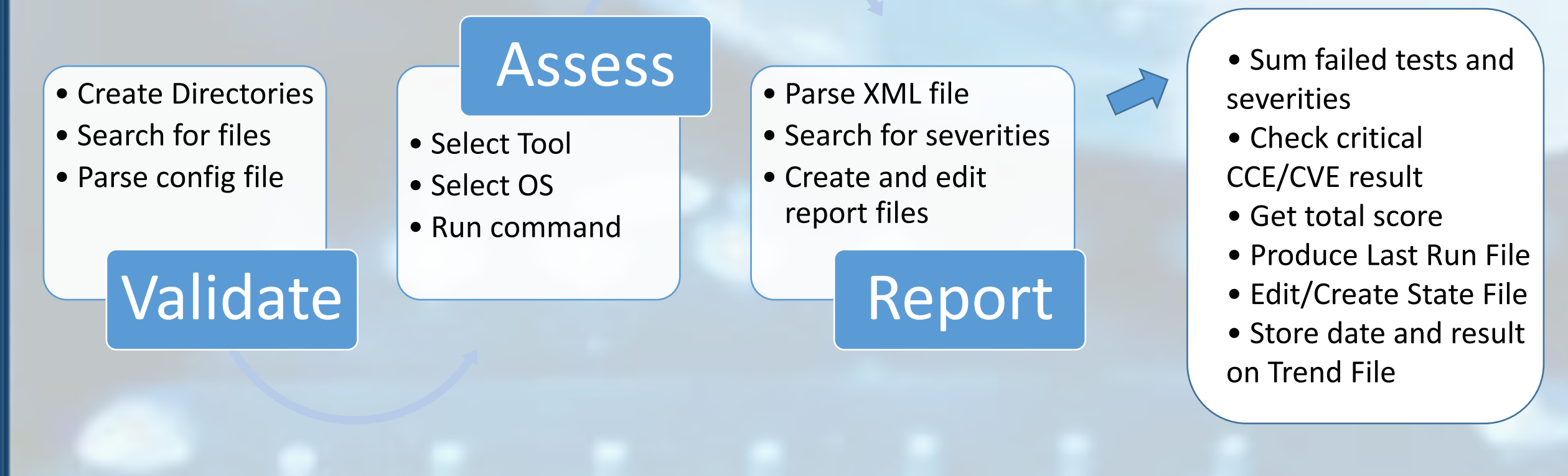


Figure 12. Diagram of script procedures

Method

Nagios Procedure

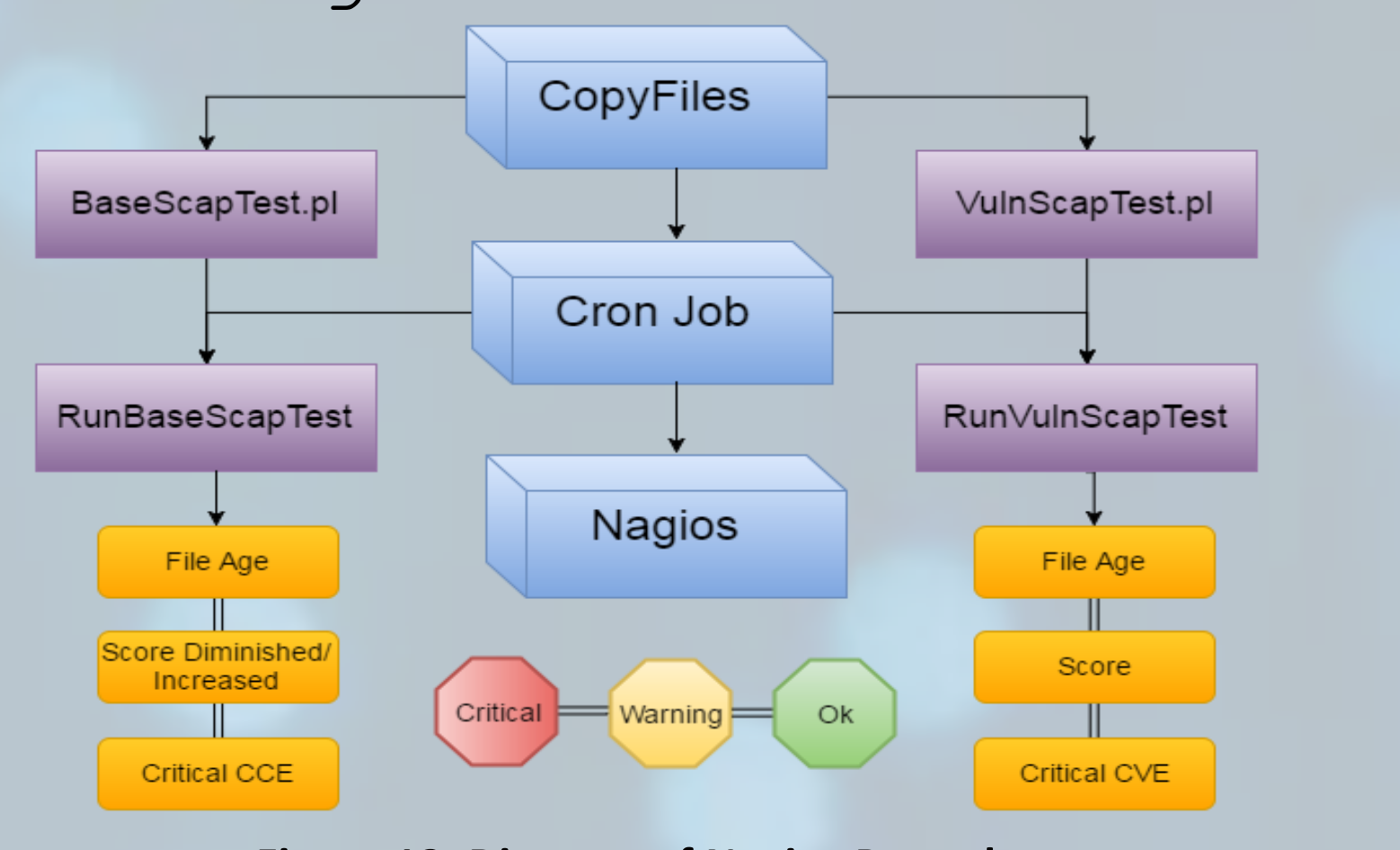


Figure 13. Diagram of Nagios Procedure

A bash script named CopyFiles is executed in order to locate files in their respective folders. This will set in place the cron job that was previously configured to assess the system daily. After this is done, a check script is executed to verify the results of the file age function, score, and identified critical rules. Scores are reported to the Nagios interface.

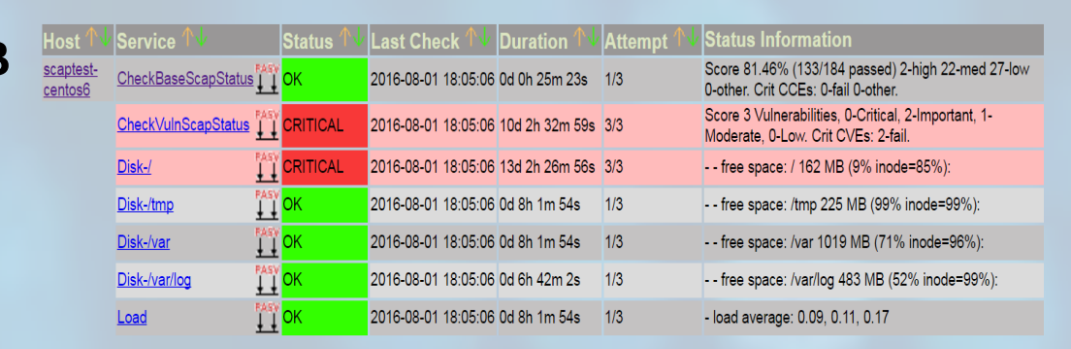
Results & Conclusions

Operating Systems Tested

A

	Baseline	Vulnerabilities
CentOS	OpenSCAP & CIS-CAT	OpenSCAP
RHEL	OpenSCAP & CIS-CAT	OpenSCAP
SLES	CIS-CAT	OpenSCAP
Debian	CIS-CAT	Needs to be tested.
Ubuntu	CIS-CAT	OpenSCAP
FreeBSD	Not found.	Needs to be tested.
Solaris	CIS-CAT	Needs to be tested.
Windows	Needs to be implemented.	Not found.

B



Host	Service	Status	Last Check	Duration	Attempt	Status Information
scaptest-centos6	CheckBaseScapStatus	OK	2016-08-01 18:05:06	0d 0h 25m 23s	1/3	Score 81.46% (133/184 passed) 2-high 22-med 27-low 0-other. Crit CCEs: 0-fail 0-other.
	CheckVulnScapStatus	CRITICAL	2016-08-01 18:05:06	10d 2h 32m 59s	3/3	Score 3 Vulnerabilities, 0-Critical, 2-Important, 1-Moderate, 0-Low. Crit CVEs: 2-fail.
	Disk-/	CRITICAL	2016-08-01 18:05:06	13d 2h 26m 56s	3/3	- - free space: / 162 MB (9% inode=85%):
	Disk-/tmp	OK	2016-08-01 18:05:06	0d 8h 1m 54s	1/3	- - free space: /tmp 225 MB (99% inode=99%):
	Disk-/var	OK	2016-08-01 18:05:06	0d 8h 1m 54s	1/3	- - free space: /var 1019 MB (71% inode=96%):
	Disk-/var/log	OK	2016-08-01 18:05:06	0d 6h 42m 2s	1/3	- - free space: /var/log 483 MB (52% inode=99%):
	Load	OK	2016-08-01 18:05:06	0d 8h 1m 54s	1/3	- load average: 0.09, 0.11, 0.17

Figure 14 (A) Table summarizes the stage of the work for each OS. Green for ready, yellow for on going, red for further needs. Figure 14 (B) shows the Nagios HTML interface.

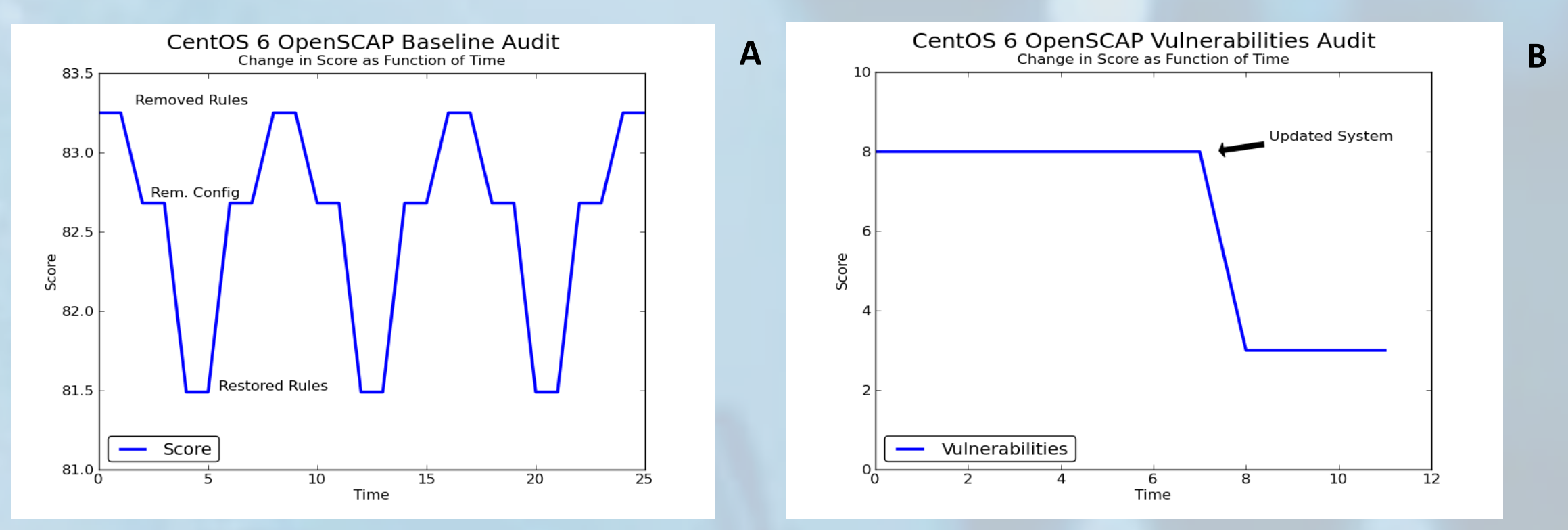


Figure 15 (A) shows the behavior of Centos 6 Baseline audit with and without auditing rules; while Figure 15 (B) represents a Vulnerabilities audit before and after updating the system. The higher the score in the baseline audit, the safer the system is. However, vulnerabilities audits state that the higher the result, the more vulnerable the system will be.

As it is shown in the graphs, scores can change dramatically, and for that reason systems need to be assessed continuously. Therefore, this method will facilitate systems administrators effort to maintain their server secure and monitored through time.

Future Work

Future work will include the implementation of a database (PostgreSQL) to organize the view of all the NCCS High Performance Computing systems behavior, and also the expansion of this method to a wider environment within the NCCS. Currently we are manually installing the package of scripts in each operating system; so next step would be developing Puppet rules in order to generalize the installation method of the scripts depending on the environment.

References

- National Institute of Standards and Technology (2009). The Security Content Automation Protocol (SCAP). Retrieved on June , 2016 from https://scap.nist.gov/
- National Vulnerability Database. Retrieved on June, 2016 from https://nvd.nist.gov/
- Open Vulnerability and Assessment Language. Retrieved on June, 2016 from http://oval.mitre.org/
- OpenSCAP. Retrieved on June, 2016 from https://www.open-scap.org/
- Center for Internet Security. Retrieved on June, 2016 from https://benchmarks.cisecurity.org/

Acknowledgments

- NASA Minority University Research and Education Scholarship
- Thanks to George Rumney (mentor), John E. Jasen (mentor), Bennett Samowich, Maximiliano Guillen, and Zed Pobre from the NASA Center for Climate Simulations, for their continuous help during this work.
- Special thanks to Julie Rivera (Puerto Rican's Advisor) and Stephen Shinn (Chief Financial Officer) for their assistance.