Security automation and optimization using HP-NA

Florian Ecard, Master SNE Student, UvA

January 2015

University of Amsterdam

Abstract

As networks keep expanding throughout time, their management is becoming more and more complex. HP Network Automation (HP-NA) is a software that allows a much easier management of a large network by permitting management centralization, keeping track of the configuration's modifications and who made them. It can also define security policies and more over.

The first aim of this project has been to automatically gather new CVE vulnerabilities which had then to be chosen depending on the specific equipment and requirements of the company.
Once this step was addressed, a second objective has been to automate the checking of the devices' security configurations.
A third aim is to change the default authentication certificate of HP-NA, because every purchased HP-NA has the exact same default and self-signed certificate.
Finally, checking the SSH keys performance has also been aimed as a bonus.

The first three objectives have been achieved while the fourth one couldn't be because of a lack of time.

Acknowledgements

I would like to thank my supervisor Olivier Willm for allowing me to do my research project with this company. It has been a wonderful experience.

I also would like to particularly thank Guillaume Demandier for supporting me all along this project and helping me every time I was in need for it.
I am also grateful to the whole team for integrating me so well with the group and for providing help as much as they could do.

Finally, I would like to thank the SNE master for allowing this internship.

Contents

1 Introduction

Networks keep expanding throughout time and their management is becoming more and more complex. In order to manage them properly, it is possible to hire network administrators or use automation tools. For example, it is sometimes needed to run a similar command into several hundreds machines, maybe even thousands. In this case, it could be a waste of time and money to enter them manually. As a solution, a few tools are available, allowing this single command to be executed in every wanted device in only a few clicks. These tools can also provide many other options.

HP Network Automation (HP-NA) is an HP proprietary software that aims to do such network automations. It is allowing a much easier management of a large infrastructure by allowing management centralization, keeping track of the configuration's modifications and who made these changes. It standardizes the configurations using regular expressions in scripts. In addition, its policies can test the devices configuration's compliances within the organization's network.

In this research project, the first and biggest aim is to gather new CVE (Common Vulnerabilities and Exposures) vulnerabilities using the HP-NA software and one of its modules, the Live Network connector (LNc). A CVE is a database that gathers known vulnerabilities along with their solutions or workaround.
These retrieved vulnerabilities are then to be filtered and chosen depending on the specific equipment and requirements of the company. The found problems along with their solutions have to be automatically passed on to the network administrators so that they can be manually addressed.
By doing so and if the administrators fix the given problems, this can help an organization to efficiently improve their security regarding known flaws for their specific equipment.
The outcome of this method describes how more secure a company could be using it, by defining how better it could perform in a penetration testing environment at a vulnerability scan level.

Once this step has been addressed, a second objective is to automate the checking of the device's security configurations. Such verifications concern for example ACLs (Access Control Lists), authentication configurations. Indeed, it is needed to be sure at any time that all the equipments are safely configured and that they have the according behaviour. The integrity of these configurations also has to be checked (one cannot check several hundreds of devices manually).

The third aim is to change the default authentication certificate of the HP-NA. Because every purchased HP-NA has the exact same certificate, an intruder would then be able to easily spoof the identity of the HP-NA and have access to a big part of the company's network. A new certificate has thus to be generated.

This report will first introduce the related work that has already been done concerning this subject. The methodology will be defined in a third part and the results obtained will then be discussed and analyzed in another chapter. Finally, a conclusion will provide the outcome and limitations of this research as well as an answer to the research questions.

1.1 Research question

1. Evaluate the "HP Network Automation" (HP-NA) software capabilities to audit the network equipments configurations about potential security issues.

   Depending on the newest CVEs, define specific policies to be implemented.

   Enforce company specific security configuration rules. Make a proof of concept.

2. Can the security configuration of equipments be automatically checked?

3. How does it fit in an overall network security improvement process?

1.2 Ethics

There are not many ethical issues with this project: no hacking is performed.
However, for security reasons, the name of the company as well as its exact equipment and network topology will not be disclosed. Similar and general schemes will be given instead. This choice has been made by the organization in order to allow a full transparency on how to reproduce this work for any large network. If the real topology and name of organization are provided, the flaws found (if any) could be used against the organization.

2 Literature Review

This section introduces similar tools to HP-NA & HP-LNc . They will be described in-depth and compared with HP-NA in the following sections.
This project being particularly precise, practical and concerning a specific product, only a small quantity of related work has been found.

2.1 HP-NA & HP-LNc

HP-NA

The proprietary software that has been used during this project is the HP-NA version 9.22. It has been installed into a Linux Redhat 5.3 server.
HP-NA allows a company to manage its large network in a much easier way than previously available. It does so by centralizing its administration and allowing to modify the network topology through a GUI (Graphical User Interface). It also permits to easily show what devices are modified, how they are modified, by whom and why.
The specification of predefined standard can be used to implement policies and thus increase the security at a network level.

Several providers are supported by the HP-NA such as Cisco, Nortel, F5 and Extreme. The support of all these devices can thus be done via a single tool.
In order to communicate with all the equipment of the network, a CLI is used in order to access these devices via SSH. SCP can also be used in a similar manner. And if none of them are available, TFTP can be used in order to transfer configuration files.

HP-NA being proprietary and needing a licence, one has to pay in order to obtain it. For an organization containing about 350 servers, the cost to manage such an environment is of about 250,000 euros. This amount of money is huge and should be taken into consideration when an automation tool is chosen.
More information can be found on the HP-NA web page [6].

According to the CVE details website [5], HP-NA could be exploited from a few CVE flaws from 2011. 3 were detected at that time and then one per year, each being different with one another. They were various and could go from SQL injection to information gain. However, HP is working on it actively to reduce these flaws.
A limit that has been found about the HP-NA is that even if it supports a lot of vendors, it doesn't support Windows server management. This would be a problem if there are a lot of Windows servers within the infrastructure. To overcome this limit, The Windows Desired State Configuration can be used (regardless to security) in order to manage a Windows environment.

HP-LNc

The version used of the HP-LNc is 3.40. This tool is integrated within a lot of HP software products. It is used in order to dynamically update the content of a product by, for example in this case, download the latest CVE vulnerabilities. The content actually depends on the core automation products used by the organization (NA / SA / CA).
It has an extensible and powerful architecture allowing to safely download the content from the internet and optionally import it in the HP-NA. It can do so when downloading by using the cutting-edge secure hash technology.

Its aim in this project is to download and import every week the new CVE vulnerabilities into the HP-NA to allow the administrators to have an up-to-date vulnerability database. If the LNc is installed within the HP-NA server, then it can be run from its GUI as an external task. It thus allows the LNc to be configured from that GUI.
The LNc web page provided this information and more can be found in there [4].

The figure 1 below shows how the CVEs (Common Vulnerabilities and Exposures) are retrieved and added to the server using LNc and HP-NA. It shows that first, Hewlett Packard retrieves the new CVEs and translates them in an adequate format. It then passes them on to the Live Network to publish them. The company's server running LNc will then contact the Live Network in order to retrieve the newest CVEs. Finally, it will trigger a vulnerability check and send an alarm if a device is matching a specific CVE.

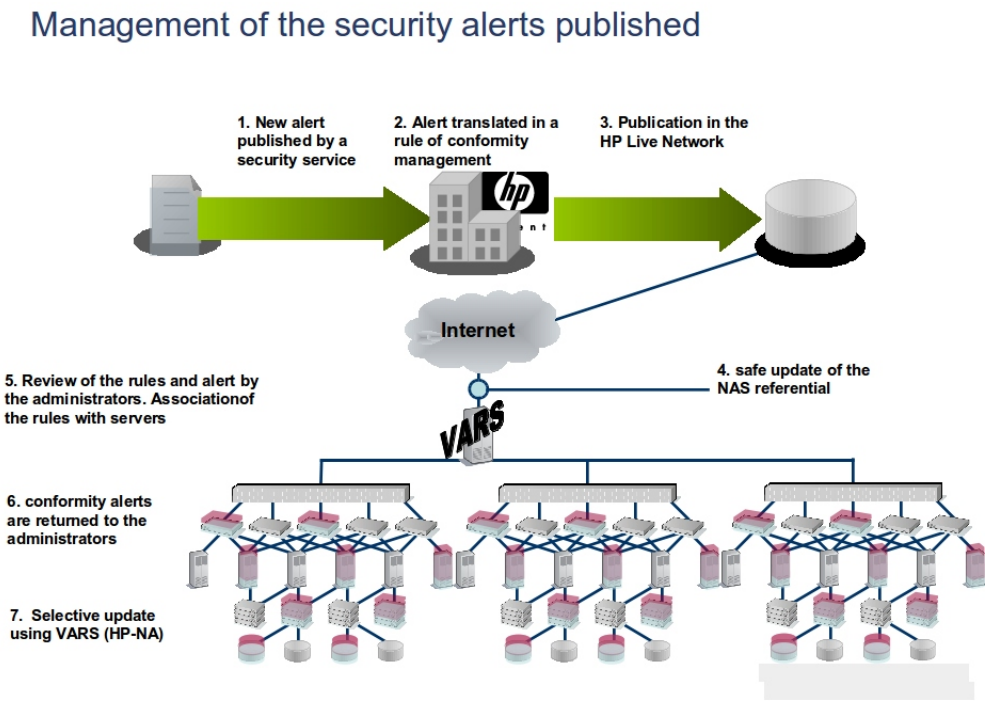

Figure 1: CVEs retrieval

2.2 OpenSource Tools

This section describes open source tools that can meet similar achievements to the HP-NA. However, it must be kept in mind that security speaking, HP-NA performs much better. Indeed, Chef and Puppet are used for network management only (performing similarly to HP-NA) and are thus unable to retrieve CVEs from the web. No open source tools was found to be as complete as HP-NA.
On the other hand, using an open source tool is free. And the code is known to everybody, allowing a community to work together on improving eventual flaws.
Chef and Puppet are made in order to maintain the desired state of a network. Each machine will thus be ensured to be in a correct configuration at any time. HP-NA aims to do the same but is using a graphical interface at its front-end, being more user-friendly.
Chef and Puppet are described in-depth in the following.

2.2.1 Chef

Similarly to HP-NA, Chef aims to manage and scale large networks. It focuses on obtaining a testable, understandable and modifiable network.

In order to automate computing tasks, Chef relies on so called recipes. These recipes are a gathering of instructions allowing one to manage database servers, web servers, load balancers and so on. When putting the recipes together, they define what the infrastructure should look like and how it should be managed.

Chef uses a central management server to manage other entities of the infrastructure. This is done by installing a client on each machine, which will communicate with the management server [10].

2.2.2 Puppet

Puppet is similar to Chef. It is an automation and configuration management tool that keeps its infrastructure in a desired state. It is written in Ruby and is platform independent even though it is mostly used in Linux distributions [9]. Puppet proposes a few automation features like the installation and configuration of Apache, NTP and MySQL. Furthermore, it can manage APT sources, keys, system reboot on Windows and more.

Once Puppet has been installed, every server will have a Puppet agent configured. The management server will be in charge of communicating with the client nodes in order to define the desired state. The following paragraph describes how to enforce the desired state.

The first phase is the creation of a fact collection. In this phase, every client will send data regarding its machine such as hardware and OS information to the management server. In the second phase, the management server compiles all this data in order to create what is called a catalogue. The latter defines what should be the desired state of each machine. Next, the management server will send the catalogue to each client node where each of them will make the necessary changes according to the catalogue. The last phase is the reporting phase, where each agent reports back to the management server which changes have been made. Finally, these reports can be shared with other management servers through the Puppet API [9].

2.3 Proprietary Software

2.3.1 Network Vulnerability Assessment

This software (referred as NVA) is the closest related to the use that has been made of the HP-NA during this project.

This tool allows to make a database that gathers every known vulnerabilities of a company's devices.
These vulnerabilities are tested on the equipment using the tool Foundstone, which is a free MacAfee proprietary product. More information can be found on their website [2].
Each vulnerability is then manually processed in order to know whether or not the equipment is to be patched. This decision is also recorded in that database.

This software can be used in parallel with HP-NA to test a same vulnerability on the same device. Thus ensuring a more coherent and consistent result.

Both this tool and HP-NA don't allow by default to automatically check the newer vulnerabilities on the test equipment. Therefore, Such automation in HP-NA could be used as a base to obtain a similar result with NVA.

2.3.2 Cisco Prime

NVA is similar to the HP-NA in a security way, but Cisco Prime is similar to it in the network management aspect.

This products aims to improve management issues such as the end-user demand for anywhere, the use of intelligent devices in workplaces such as smartphones or business operative to save costs and implement green best practices.

Cisco defines its product as delivering "next-generation management by supporting an intuitive workflow-oriented user experience and integrated lifecycle operations across Cisco architectures, technologies, and networks." [1]

However, this tool is made to be used with Cisco devices, and there is no guaranty of the portability of this system.
Similarly to Chef and Puppet, Cisco Prime performs quite similarly to HP-NA in terms of network management. But it cannot do so in terms of network security.
More information can be found on Cisco's website [1].

2.3.3 HP-SA & HP-CA as a completion

HP-SA and HP-CA are two other core automation features provided by HP. By combining HP-SA, HP-CA and HP-NA, one could obtain a large network that is being managed in a very simple way. Considering that each of these on their own considerably improve the management of a network.

HP-SA stands for Hewlett Packard-Server Automation. As its name states it, both virtual and physical *servers* can be managed, also using a GUI. It provides similar services to the HP-NA except that the servers of the organization are being managed instead of the network level (switches and routers). It is thus managing the infrastructure on an upper OSI layer.
It allows for example to automate the server discovery, software provisioning, application configuration and so on.
for more information about it, see the online website and its administration guide [8] [7].

HP-CA stands for HP-Client Automation. This software concentrates on the management of end-to-end persistent devices, being either mobile or fixed and virtual or physical. It mostly reports on hardware and OSs uncommon activities. It allows thus to be compliant with complex and specific networks regardless of the specificity of it.
This software can reduce the costs on initiatives such as virtualization, cloud, migration and more.
Refer to the HP-CA web page for more information [3].

Could similar achievements to the HP-NA be met?
The objective concerning the automatic retrieval of CVEs using HP-LNc could be adapted to both of them (CA and SA), thus improving their security awareness too. They would obviously be concerned by other CVEs than the network automation software.
The integrity checking of the configuration should also be possible in these two. However, no information was found about this possibility and the assumption that it could be possible is thus to be verified.
Concerning the modification of the certificate for safety purposes, it appears that both the HP-SA and HP-CA certificates are default self-signed certificates. They should thus be replaced in a similar manner that it is described for the HP-NA.

3 Methodology

To achieve the goals described in the research questions, several methods have been used. Each of them was using the HP-NA and is defined in-depth in the following.

3.1 HP-LNc installation and customization

This part explains how the first objective has been achieved.
The aim is to retrieve the vulnerabilities and describe how they are gathered. It is also made sure that they are not already existing nor duplicated.

In order to do so, a way had to be found using the HP-NA. After further investigation, it has been discovered that a module is available in HP-NA, called HP Live Network connector (HP-LNc). It allows to retrieve the vulnerabilities depending on the contract level (Depending on how much one pays). It will then display them in a software vulnerability report. The company's contract level concerning this feature is the highest.
It has been possible to install the LNc in the HP-NA from the SSH interface. The LNc is needed in order to allow the communication with the HP-Live Network (HP-LN) via the HTTPS interface.
It is reminded that the HP-LN gathers all the CVEs in an HP remote location, and the HP-LNc retrieves the wanted ones from within the company 1.

Details on how to install and configure the LNc are provided in a documentation that has been created for the company. It also describes the problems encountered in order to avoid any similar issue in the future. This documentation is available in the Appendix A 5.1.

The HP-LNc can be configured by selecting the device family for which it will retrieve the CVEs. These are called "streams", several are available and are shown in the figure below 2.

```
[root@tlssbcloud26 /tech/cloud/lnc/lnc/bin]# ./live-network-connector list-streams
Product         Service         Stream (stream name)
_______________ _______________ ____________________________________________
nas             security        vc_legacy (security.vc_legacy)
nas             security        vc_juniper (security.vc_juniper)
nas             security        vc_hp (security.vc_hp)
nas             security        vc_nortel (security.vc_nortel)
nas             security        vc_cisco (security.vc_cisco)
nas             security        vc_f5 (security.vc_f5)
nas             security        vc_check_point (security.vc_check_point)
nas             security        vc_cisco_diag (security.vc_cisco_diag)
nas             security        cc_pci_cisco (security.cc_pci_cisco)
nas             content         na_drivers (content.na_drivers)
```

Figure 2: List the available streams in the HP-LNc

Configuration
To correctly configure the LNc, each configuration has to be entered through a CLI and not via a text editor. Indeed, HP-NA doesn't support this method of configuration.
In this project, Cisco has been the chosen device type to configure the LNc.

Once the family devices (Cisco) to be submitted to a vulnerability check are specified, two usernames have to be provided to the LNc.
Indeed, one HP username is used to allow the LNc to securely download the content from the HP website. The second username is the administrator's username of the company. It is needed to allow the LNc to import the CVEs from the LNc within the company.
This difference between usernames has to be well configured. Otherwise, the Live Network connector will not work.

These steps done, the LNc should be running and the first CVEs should have been retrieved. The next step shouod be to find a way to automate these CVE retrievals. There are two possible scenarios, depending on the LNc installation location. If the LNc is installed in the same server as HP-NA, it could be configured from the HP-NA itself. But if it isn't, then the CLI should be used and extra parameters should be given to provide access to the HP-NA. Making the installation more difficult.

Being installed on the same machine, the HP-NA allows to run the LNc as an external task. Once launched from the GUI, the external task fields have to be fulfilled.
An example of configuration is shown in the next figure 3.

Automate the vulnerability scan
Now that the CVEs are automatically retrieved on a weekly basis, there still is one thing missing to create a fully automated vulnerability check. Indeed, these new CVE vulnerabilities should be automatically filtered and then tested onto the according equipment only.

Doing so has not been possible during this project because it would have needed the development of a Perl Script. In addition, this objective is the first of a list of four and Perl is not bieng proficiently managed by the researcher. It has thus been decided to come back to this task later if the other objectives were achieved on time, which hasn't been the case. This task is thus left as a future work.
Perl is a relatively easy to use and user friendly language, using it as an API (Appication Programming Interface) is common for a software. For an experienced Perl programmer, creating such a script should be doable in a few working days.

The following describes some steps that could be used as a base to create such a script:
A first step would be to take some test devices (gathered by OS type) and add them in new groups, each representing a device family. Next, retrieve the newest vulnerabilities according to the device family (i.e. Cisco NXOS) should be done. Having these two, automatically use the vulnerability to scan the according equipment should be implemented. The final aim would be to alarm the administrator by displaying the devices that are reported vulnerable.

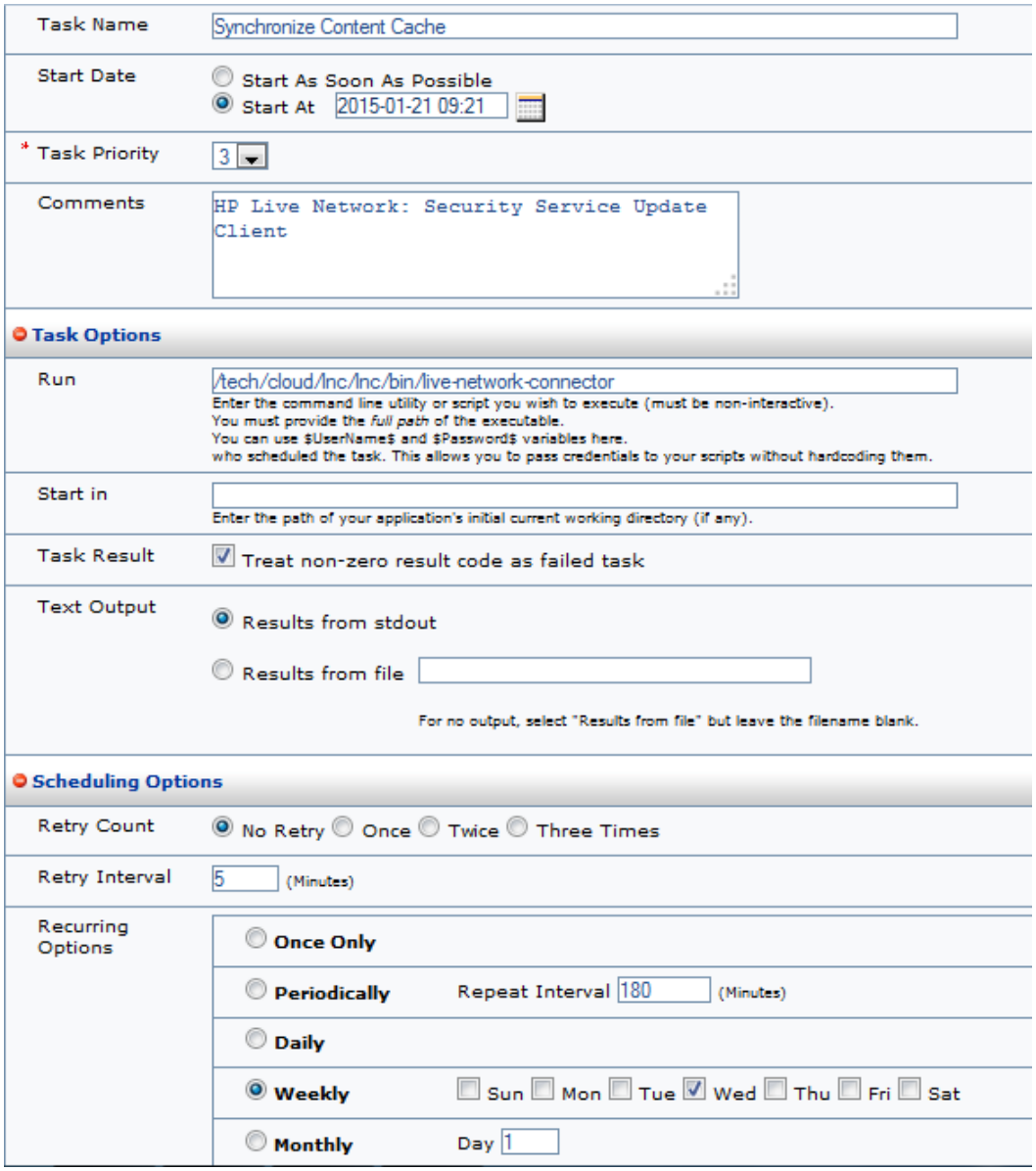

Figure 3: Example of recurrent task to retrieve the CVEs

The figure 3 shows the different fields asked to run an external task. The task name, start date, priority (1 to 5) and comment have to be defined first. The "run" field should contain the absolute path used in order to launch the LNc from the server. The task result should be checked, because it gives an error otherwise. The text output defines how is the output wanted to be checked. The weekly schedule that has then to be chosen, in this case it is Wednesday, because new CVEs are released on Tuesdays.

3.2 Automate the security configuration checks

This section describes how it has been possible to automatically check the integrity of Cisco security configurations. It will thus be made sure that the configuration of the equipment has not been modified. Only the HP-NA is used to do so, without its module HP-LNc.

Each device has a type of operating system. In the organisation's network, only Cisco IOS and Cisco NXOS (Nexus) have been investigated. These two OSs represent over 2000 Cisco devices within the company. However, the integrity check is only executed on devices situated where the project has been done. About 60 devices are then concerned but the created rules can be adapted to the whole infrastructure if it is needed.
Two different OSs being checked, two set of policies have been created (one per OS) along with two sample groups. A so called group is used to gather a set of devices to which a set of policies will be applied. For NXOS devices, the name of the group is "Test NXOS security integrity". This group contains a single Nexus 5000, it is the only type of switches available in the company with such operating system, no more devices are thus needed to ensure that the created policies are operational.
Concerning IOS devices, the group is named "Test IOS security integrity". Two different machines are being checked: the Catalyst 4500 and 2960.

In order to know what elements of the device's configuration should be checked, it is necessary to separate security configurations from network configurations.
Once these two are distinguished, the different rules should be made to ensure that the current security configuration is the expected one.

Different set of rules can be defined within a Cisco device, the ones that were estimated to be the most important are described below. Why it is important to be sure of their integrity is also explained:

- ACLs are an essential part of the security configuration. Indeed, they authorize access for a specific range of IP addresses and to drop non-authorized connections. If modified, an intruder could be granted access via the corrupted device.
- The "features" are functions that can only be enabled in a Nexus switch. They are defined before the configuration itself to avoid unnecessary lines of code. For instance, Telnet could be disabled from there, avoiding to go through the whole configuration of the device. If an intruder could modify these, a service having a known flaw could be enabled and exploited.
- The AAA (Authentication, Authorization and Accounting) is mainly responsible of the authentication. If an attacker changes these parameters, the AAA server could be used to authenticate in any machine.
- The NTP (Network Time Protocol) is running the network's synchronisation. Maliciously changing its parameters could cause all the switches to synchronize themselves again with the network. In a worst case scenario, this could crash the whole network.
- The username that can access a device is also an important feature to be checked. Indeed, it provides direct access to several critical equipments.
- The VTY parameters are enabling protocols such as Telnet and SSH. It is well known that Telnet is not safe any more and it must be ensured that SSH is enabled instead.
- The logging facilities must also be well protected. Indeed, if they are deleted or moved to an unknown location, it becomes impossible to troubleshoot a problem.

All these reasons show that it must be ensured that these parameters haven't been modified. The way to do so for each OS is described in the following.
The rules can be defined using regular expressions or not, in order to be more flexible. The set of rules should be written in Cisco commands, a script will then compare whether the desired configuration matches the actual one.

Once the rules have been created to check the integrity of these two Cisco OSs, a way had to be found to associate them with their corresponding OS.
To do so, Each set of rules (concerning NTP, ACLs etc.) has been added to the two policy sets (one per OS). These set of policies will associate the compliance of the rules with the according group of devices. each of this group contains the test devices of the concerned OS (one contains the Nexus 5000, the other the Catalysts 4500 and 2960).
An illustration is shown in the figure 4 below. It describes the association between a group of devices and a set of rules.

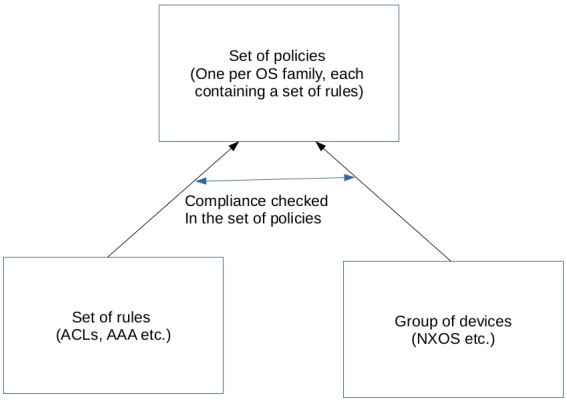

Figure 4: Association of the rules with their corresponding OS

In order to automate these two policy checking (otherwise an administrator would have to run the policies himself), a new task needs to be created. It can be done in a similar way as in the first aim, as shown in the figure 3.

The following sections will describe how the creation of the integrity checks have been done in a practical manner. Both Cisco IOS and Cisco NXOS operating systems will be described. The following parts are specific to the organization and have been designed for this particular infrastructure. If this research is used as a base to secure a network, the created rules should be adapted to the network's specifications.

3.2.1 Cisco IOS

For Cisco IOS operating systems, the AAA authentication integrity should be checked using regular expressions. Indeed, a line is present in some devices but isn't in others and thus has to be made optional.
Another rule concerns the ACLs integrity check. In order to check the ACL concerning the static OSPF, a regex (regular expression) is used because the final line could be either a "!" or an expression.
Concerning the SNMP ACLs, no regex have to be used. Indeed, being sure that all the lines aren't changed and that their order is correct is sufficient.

The telnet ACL is different, its policy is based on block recognition. This kind of policy finds every block of a configuration file that is starting and ending by a particular expression. What should be in between these blocks is thus defined in a separate field. In this case, it is checked that SSH is used by controlling that each line finishes with a 22 and not with a 23 (respectively referring to the SSH and Telnet port).

The logging facility could be detected in two ways, depending on the equipment. It could be done by either detecting the first line as being a "!", or as being a "deny ip any any" from another ACL. The same applies to find the last line of the facility.
Simply checking the configuration lines about the logging could have been done, but if lines were added to this part, simply checking them couldn't allow to see that.

The integrity of the NTP has also been checked using regex. It has been used to ensure that the clock period is any number of 8 figures and that other lines of codes were not modified.

The usernames can be defined in two different ways, either using the combination "secret 5 [30 charac passwd]" or "password 7 [16 charac passwd]". An example of regular expression to distinguish between these two is shown below:
secret |password 5 |7 ([\S]30)|([A-Z,0-9]16)

3.2.2 Cisco NXOS

An example of some policy rules that could be created via the HP-NA GUI is shown in the figure 5 below:

Rule Name	Rule Type	Device Family	Importance	Description	Actions
AAA	Configuration	Cisco NXOS	High	Checks if all the authentication rules haven't been modified	View & Edit | Delete
features	Configuration	Cisco NXOS	Medium	Check the integrity of the features	View & Edit | Delete
ntp	Configuration	Cisco NXOS	Medium	checking the NTP integrity	View & Edit | Delete

Figure 5: Test NXOS integrity

The AAA, and ACLs rules being very similar to the ones described in the previous part 3.2.1, how they should be created is not defined again.

As explained earlier, the "features" are specific to this OS. Being static, no regular expression should be needed to ensure their integrity. The same applies for NTP and VTY configurations.
The username rule is similar to the IOS OSs but is simpler. Indeed, there is only one kind of password introduction and no regular expressions should be needed.

3.3 Renew certificate

This objective has been achieved based on the HP-NA administration guide [11] and user-guide [12].

SSL certificates are used to allow a secure connection between a server and a web browser. By simply using a default certificate, one could create a similar certificate pretending to be Alice whereas its real identity is Trudy.
In order to prevent such thing to happen, a Certificate Authority (CA) can be used. This authority is a third party delivering digital certificates certifying the ownership of a public key. It thus allows both sides to safely verify each other.

HP-NA uses a default and known certificate without a third party CA. This means that even if HP-NA is only accessible from the inside of a company's network, if an intruder manages to get in, he or she could act as being HP-NA and thus retrieve usernames and passwords from users.
In addition, every time the HP-NA is accessed with this default certificate, a pop up attests that the navigation isn't private (due to the non-use of a CA). So every time a network administrator accesses it, he has to accept the pop up and thus could be under attack.

This flaw should to be fixed as it could cause serious problems in an organization's network. Generate a new certificate using one a CA should be done. It ensures that the HP-NA actually is itself and that no error appear when accessing it. If an attack on the certificate occurs then this very problem (navigation not private) will show up and this time the network administrator will be aware that a problem is really happening and could react accordingly.

A detailed documentation has been created and is available in Appendix 2 5.1. If it is wanted to create a new CA-signed certificate, this documentation explains the steps to follow to achieve it.

It has been possible to regenerate a certificate by connecting to the HP-NA via the SSH interface. The default directories that contains the certificates are called *truecontrol.keystore* and *truecontrol.truststore*. The ".keystore" contains the server certificate in itself and the ".truststore" contains the chain of trust, corresponding to the CA. Before any new certificate generation, a backup of the existing one should be done for safety.
A new asymmetric RSA key should then be generated (2048 bits is great compared to the actual technology), it is by default valid for a year. During generation, some values are asked such as the name of the organization (it has to be the Fully Qualified Domain Name (FQDN) of the server), the country and so on. Only the field concerning the FQDN has to be exact, the others are only informational.

Once the key is created, it is used to create a CSR (Certificate Signing Request), which is to be sent to the CA. The certificate authority will then sign the CSR and send back the according certificates, marked as ".crt". At least two certificates are returned, one is for the ".keystore" file and the other for the ".truststore".
If more are sent, the others are used to create the chain of trust for the ".truststore". If this is the case, the certificates forming the chain have to be concatenated into a single file, only two files are then still remaining. Each certificate should be put one after the other, where the first one has to be the root certificate. An example of such concatenation is shown in the figure shown below 6.

```
-----BEGIN CERTIFICATE-----
MIIDwTCCAqmgAwIBAgIILv8bQwmWlIgwDQYJKoZIhvcNAQELBQAwUTEnMCUGA1UE

          ......................................

tF0PKanTyDv6991mL5V24c9yOc7+nAc85QCA4SavCj7sWMzlag9UT1k4y0xwQhE4
B2zAs9Y=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDwDCCAqigAwIBAgIILMtac9ga7LwwDQYJKoZIhvcNAQEFBQAwTzElMCMGA1UE

          ......................................

56q+HebIUu/NylTTjYkCgwRKeiR4dyrKHHtG9VlZWkAy0jsTOigRW1QGz2i1kquP
EGl9ew==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDuzCCAqOgAwIBAgIIERHJkhIZEncwDQYJKoZIhvcNAQELBQAwQzEZMBcGA1UE

          ......................................

m3NefVCLUaZb56w1Cxrwz4hWLEXzYwaE2jW1y7BW4mF14f3CrKDTmdxlNxHKljI=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDtDCCApygAwIBAgIIPKnlhYlkiv4wDQYJKoZIhvcNAQELBQAwTzElMCMGA1UE

          ......................................

Ur7xirmeAMTB15l02Shu2fGdMr1RMPIv/86SMWTDLoy5bbIOHNCp/A==
-----END CERTIFICATE-----
```

Figure 6: Concatenated certificates

The certificates should then be imported in the ".truststore" and ".keystore". The certificates should thus be ready to use and a restart of the service should make everything work. To be sure that the new certificate is effective, it is just needed to go to the URL of the server and check if an error appears. If there are none, then it means that the certificate is effective. To double check it, click on the lock next to the "HTTPS" in the URL and verify that the information about the certificate is true.

It can happen that it doesn't work the first time or that the HTTPS in the URL is marked red rather than green. This would be because the root certificate provided by the CA isn't yet validated in the hosts desktops. To avoid waiting that it gets validated, the administrator using HP-NA should add the root certificate to the certificates trusted by the web browser by modifying the browser's parameters.

4 Results and discussion

This chapter will first remind the research questions and added objectives of this project. The results obtained will then be discussed in order to evaluate their efficiency. Finally, the problem encountered during the project will be defined.

4.1 Research question

1. Evaluate the "HP Network Automation" (HP-NA) software capabilities to audit the network equipments configurations about potential security issues. By using the editor supplied policies and defining specific policies from the CVE database descriptions. Enforce company specific security configuration rules. Make a proof of concepts of this.
2. How to automatically check the security configurations of equipments?
3. How does it fit in an overall network security improvement process?

Two other objectives were added during the first week of the research project.
First, the creation of a new CA signed certificate has been done in order to provide a CA trusted certificate. The second objective added was to check the length of the SSH keys and be sure that they are regenerate at each use.

4.2 Results & Discussion

HP-NA being an automation tool, it is by default not set to improve the security of a network. The first aim of this research has been to find how to extend to the maximum the HP-NA security capabilities. The expected results were to detect and display the newest CVE vulnerabilities in an automatic manner, and only for the desired equipment.
This aim has been a success by using the HP-LNc to retrieve the CVEs and then using the HP-NA to do so in an automatic manner. A way to optimize a network's security using HP-NA has thus been found and a documentation is available for reproducibility.
However, this vulnerability check automation does not protect a network from every vulnerabilities. Indeed, there are many vulnerabilities for which no workaround or solutions are known. In addition there are always flaws that are unknown of the majority and that an attacker could use to attack a network. HP-NA would not be useful in these cases.

The second objective has been to create and automate a configuration integrity check for different devices. Indeed, a company should always be aware if a device's state has changed. This objective has been a success by creating policies in the HP-NA. These policies would check the configuration of specifics Cisco equipments. Different operating systems could also be checked by adapting the methods described to the specific OS features.
The results were verified by modifying the security configuration of the sample devices. A check had then been performed and successfully generated an alert if it was not policy compliant. The image shown in the next page 7 shows a case where a policy is in compliance with the selected devices.
This achievement is thus improving a network's security by ensuring the integrity of the security configurations of devices. Both an attack and a human mistake can now be detected using these automated integrity check.

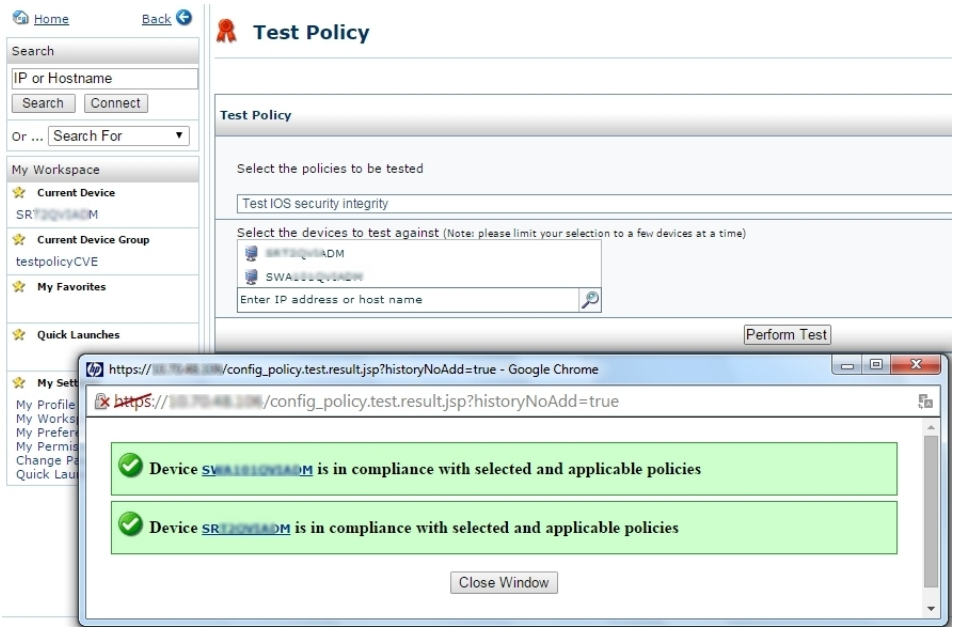

Figure 7: Test of the policies

A third objective has been to generate a CA-signed certificate for HP-NA. This has been done to improve the HP-NA's software security itself. By doing so, a theft of HP-NA's identity could be detected and then reduce the chances that someone could take control over it.
If this wasn't achieved, it would still be very easy for an intruder to gain access to the HP-NA. The administrators would not even be able to detect when such thing occurs because of the persistent pop-up showing "Connection not trusted" at login time.
This solution only allows an administrator to be aware that when such a pop up is now appearing, something wrong may happen. If this person is not careful enough, then the attack described could be successful anyway. The efficiency of such an improvement is thus depending on the employees proficiency and carefulness.

The final aim that was given as a bonus during this project hasn't been tried. Indeed, no time was left to look into the area of SSH keys. No further explanations can thus be provided about this objective and should be investigated as a future work.

in order to answer the last research question, the findings of this project have allowed to have a more secure network. However, these improvements are useful only if an intruder manages to get into the organization's network. Indeed, these improvements are only internal but could protect a lot of things in case of attack. It also allows to detect much more quickly an unattended modification of the network.

4.3 Problems encountered

Throughout this project, a lot of problems had to be solved.
First, it took a week before all the access (four different accounts were needed) to the equipments was granted, significantly delaying the time to be operational on those devices.

4.3.1 External problems

There are two HP-NA installed, one of them is used for testing and the other one for the operational network. The whole project was based on the HP-NA test which had just been upgraded to a newer version still tested by the employees. Several people were thus implementing different things on the same server.
The first problem encountered is that an engineer changed the port to connect to the HP-NA. A bad manipulation was done and the HP-NA wasn't accessible for a few hours.
Also, the update to the same version as the test server in the production environment was made during this project. It switched off the access to both servers for about a day.

4.3.2 Research related problems

HP-LNc
When the LNc was launched in order to put the module up-to-date with the latest CVEs. However, even when it was stated to be up-to-date, it contained at best CVEs from 2011. This problem was due to a bad configuration. Indeed, the settings were added by hand using "vim" whereas the live network connector CLI has to be used instead.

Once this had been corrected, a new update was tried by enabling Cisco devices, the result gave a new bunch of errors. The most significant one was stating that the connection to the localhost through the API was impossible.
It was notified that two different logins had to be used. One connecting to the HP database and one connecting to the HP-NA server. By providing the correct credentials, the localhost connectivity problem has been solved.

However, the import of CVEs in the server came up again, stating unable to continue with import. The solution found was that the username provided in the configuration was 'only' an expert and a root account was needed, this problem was definitely solved by switching this account with a root user.
The LNc was now 100% working and more than a hundred newer CVEs were downloaded.

Certificates
Concerning the certificates, the HP documentation that explains how to change it is very badly explained (see [11], p.97) some problems thus arose.

In addition, there has been two problems encountered due to a human mistake. The first one is when asking for a CA-signed certificate, the most important parameter is the FQDN of the server. The *www* was added by mistake in front of the FQDN when generating the CSR. Once this had been found and a new CSR created, the problem was solved.
Also, the order of adding of the certificates in the ".truststore" file wasn't correctly done (nor specified in the HP documentation). Indeed, the root certificate should be first.

5 Conclusion and Future work

In this project, out of the four objectives that were provided, three have been achieved.
The automatic retrieval of the new CVEs concerning Cisco has been done using the HP-NA and its module, the LNc (Live Network connector).
Automatically check the devices' security configuration integrity has also been achieved. This aim has been done using HP-NA only, once the configuration that wanted to be checked and the policies had been defined.
Changing the self-signed certificate present by default in the HP-NA to a CA-signed certificate has also been done.
Concerning the SSH bonus objective, it hasn't been possible to achieve it.

This research allowed to define the best practices to use the HP-NA security capabilities. The report contains guidelines on how to implement these improvements and what are the key points not to be missed in this process.
If all the proposed improvements were to be implemented, the internal security of a network could be widely expanded.
It is possible that an attack is successful on device configurations or on HP-NA. However, the improvements brought would also allow the security team to detect the attack much faster than before.
Even if the system is now more secure, the HP-NA doesn't stand as a full security mechanism but only as a part of a security solution. Many flaws remain unknown and the fact of having an infrastructure that is fully CVE compliant doesn't mean that the network is secured.

It is still possible to improve the obtained results. Some of the possible improvements are defined in the following.

5.1 Future work

As a future work, by using the HP-NA API in Perl, a script could be written so that once the CVEs are retrieved, all of them would be checked on the concerned equipment. Once this done, only the devices reported vulnerable should be returned to the administrators.

If this first improvement was to be achieved, then it would be possible to compare the results returned by it, with the other tools such as NVA (Network Vulnerability Assessment). Doing so would allow to ensure that both tools provide the same new vulnerabilities and that the same devices are found to be vulnerable.

A last improvement would be to do the SSH key checking and automate it if needed. Which means to verify for instance the key length of the keys, or be sure that they are regenerated at each connection and further on.

List of Figures

References

[1] Cisco prime. http://www.cisco.com/c/en/us/products/cloud-systems-management/prime.html. Accessed: January 2015.

[2] Foundstone. http://www.foundstone.com/. Accessed: January 2015.

[3] Hp client automation page. http://www8.hp.com/us/en/software-solutions/client-automation-management-software/. Accessed: January 2015.

[4] Hp live network connector page. https://hpln.hp.com/group/hp-live-network-connector. Accessed: January 2015.

[5] Hp-na cve flaws. http://www.cvedetails.com/product/20717/HP-Network-Automation.html?vendor_id=10. Accessed: April 2015.

[6] Hp network automation page. http://www8.hp.com/us/en/software-solutions/network-automation/. Accessed: January 2015.

[7] Hp server automation guide. http://www.blackstylus.com/portfolio/SA_9.0_Overview_and_Architecture_Guide.pdf. Accessed: January 2015.

[8] Hp server automation page. http://www8.hp.com/nl/nl/software-solutions/server-automation-software/. Accessed: January 2015.

[9] Puppet labs. https://docs.puppetlabs.com. Accessed: April 2015.

[10] What is chef? https://www.chef.io/chef/. Accessed: April 2015.

[11] HP. Hp network automation software: Administration guide. April 2014.

[12] HP. Hp network automation software: User guide. May 2013.

Appendixes

Appendix A

Configuring the Live Network connector (LNc)

Before starting the configuration, some criterias must be verified:
- Being **root** is mandatory
- If a version of the LNc is already installed, it must be entirely deleted.
- The port 443 must be enabled because LNc communicates with HP-LN through HTTPS.
- The environment variable PYTHON_HOME must not be set.
- If LNc is used for Software Automation (SA), the DET (DCML Exchange Tool) must be installed via /*opt/opsware/cbt.*

The installation can be done by following the HP-LN documentation's steps. These will thus not be described here, only how the configuration will be.
Once installed, the configuration can start. The one explained below is only valid for Linux and Solaris Operating Systems.

Configuration

In the following commands, *install_directory* refers to the LNc location, for example: */tech/cloud/lnc.*

The configuration file is located at: *install_directory/lnc/etc/live-network-connector.conf*
The log file is located at: *install_directory/lnc/log/live-network-connector.log*

From now on, the location from which the commands are launched is considered to be: *install_directory/lnc/bin/.*

WARNING: The files that have to be modified during the configuration must not be edited by hand (for instance using vim or nano) but by using the LNc command *write-config*. This command works as follow, where *XXX* represent the configuration to apply (replace by - -help for more information):
./live-network-connector write-config XXX

In order to correctly configure the LNc, the type of product must be specified in the configuration file. In this case, the module is NA (Network Automation), what must be added to the "product" variable is thus "nas".
./live-network-connector write-config --product=nas

Next, by fulfilling the variables *nas_user* and *nas_pass*, the username and password of the NA administrator will be provided to the LNc. The HP account must also be specified as shown below.
The field *nas_root* must contain the path where the HP-NA has been installed:
./live-network-connector write-config --setting=nas.nas_user=NAUSER
./live-network-connector write-config --setting=nas.nas_pass=NAPWD
./live-network-connector write-config --username=HPUSER --password=HPPWD
./live-network-connector write-config --setting=nas.nas_root=HPNAINSTALLDIR

The brand of the devices used by the company as well as the chosen driver are then specified. In this case it concerns Cisco and F5 as well as the NA driver:
./live-network-connector write-config --stream=security.vc_cisco --enable
./live-network-connector write-config --stream=security.vc_f5 --enable
./live-network-connector write-config --stream=content.na_drivers –enable

Some basic commands to verify the configuration

./live-network-connector read-config	# Shows the general configuration
./live-network-connector list-streams	# Shows available device families (Cisco. Juniper..)
./live-network-connector list-locales	# Shows enabled device families
./live-network-connector list-status	# Shows the status of enabled device families
./live-network-connector	# launch the retrieval of CVEs

Automating the CVEs retrieval

Now that the configuration is done and that the above commands have been launched, the CVEs should have been retrieved, the LNc's recurrence must then be programmed.
Being located on the same server as the HP-NA, it is possible to execute the LNc as an external application of the HP-NA itself. The following describes how to do so.

From the HP-NA GUI, in Tasks → New tasks → Run external application.
A new window will be displayed, called "New task / Template", HPLN will be started from there.
The fields must be fulfilled as follow (those not being specified here should be let the way they are):

- Task name → Synchronize Content Cache
- Start date → Select next Wednesday morning (vulnerabilities released on Tuesdays)
- Comments → HP Live Network: Security Alert Service Update Client.
- Run → /tech/cloud/lnc/lnc/bin/live-network-connector
- Task Result → Check "Treat non-zero result code as failed task"
- Retry count → Can be set to whatever value, Once is suggested.
- Recurring option → Select Weekly and Wednesday.
- Task Completed notification can be chosen to be sent by email or not at all, it depends on the preferences of the administrator.

Appendix B

Adding a CA-signed certificate for HP-NA

NAHOME represents the HP-NA installation directory.

- First, cd into *NAHOME/server/ext/jboss/server/default/conf* via the HP-NA SSH interface.
It is in this folder that the certificate files *.keystore et .truststore are located.*

- A new key must be generated in the *.keystore* file (create a back up). The following command is used to do so (size of 2048 bits using RSA, valid 1 year, the alias must the same all along the manipulation):
NAHOME/jre/bin/keytool -genkey -keyalg RSA -keysize 2048 -validity 365 -alias nacacert -keystore truecontrol.keystore

The default password (which should be modified) for HP-NA is: *sentinel.*
To do so in the next certificate generation, the file *truecontrol.keystore* should be erased. They will be automatically regenerated by the above command and a new password will be asked.

- Parameters will be asked in fields and must be fulfilled. It is not mandatory to provide the exact data, but it is recommended. The only parameter that must correct is the first one, "First and last name", corresponding to the FQDN (Fully Qualified Domain Name) of the HP-NA:

First name and Last name (FQDN): *var-rct.COMPANYDNS.COUNTRYCODE*
COMPANYDNS refers to the domain name of the organization. COUNTRYCODE are the two letters of the country, below the root (e.g. fr, nl)
C: *FR*
OU: Organization *Unit*
O: *ORGANIZATION*
L: *CITY*
ST: *France*
To validate, type "yes". When the password is asked (for the keystore) type enter to keep the actual one or type a new password if wanting to have a different one.

- Now that the file .keystore is modified, the CSR (Certificate Signing Request) must be created so that the CA (Certification of Authority) can sign it. To do so, type:
NAHOME/jre/bin/keytool -certreq -alias nacacert -file narequest.csr -keystore truecontrol.keystore

Provide the output to the CA, which will return a *.crt* file.

- Several certificates will be provided, one will be the server signed certificate and the others will be the chain of trust containing the root certificate and others.

- Now, the certificates must be made operational. To do so, stay positioned in:
NAHOME/server/ext/jboss/server/default/conf

A new backup for the files *truecontrol.keystore* and *truecontrol.truststore* must be done because if an error occur, operations can be started from there without having to ask again the CA to sign new ones (which can take a long time). Next, import the received certificates in the current directory (*scp* can be used if the certificates are in a different machine).
Do not copy-paste them. Indeed, some data could be lost during such process.

- Now that the certicates are imported and the *keystore* files saved in a back up, the certificates of the chain of trust must be concatenated with one another, in our case the chain of trust will have 3 certificates. The root certificate must be on top of the chain.
Concerning the server signed certificate, it must contain all the certificates. To do so, the chain of trust must be added (*vim* or another text editor can be used) below the server certificate in the following order: root certificate, technical and finaly internal.
Two certificates will then be left: The server signed and the chain of trust.

An example of chain of trust and server signed certificate concatenation is shown below:
cat CAEAirFrance-KLMRootCA.cer CAETechnicalCA.cer CAEInternalInfrastructureCA.cacert.cer >> vars-rct.XXX.fr.cer
cat CAEAirFrance-KLMRootCA.cer CAETechnicalCA.cer CAEInternalInfrastructureCA.cacert.cer > chainoftrust.cer

An example of *.keystore* file after concatenation is shown in the image below:

It is seen in this figure that the certificates are positionned one after another.
The *.trustore* file will only contain the three bottom ones, the top one being the server signed certificate.

- Once the certificates accordingly created, the alias from the *.keystore* ("sentinel") must be deleted:
NAHOME/jre/bin/keytool -export -alias sentinel -file sentinel_from_truecontrol_keystore.cer -keystore truecontrol.keystore
NAHOME/jre/bin/keytool -delete -alias sentinel -keystore truecontrol.keystore

The alias may have changed throughout time, one can check it using the command:
NAHOME/jre/bin/keytool -list -keystore truecontrol.keystore

- The final certificates can now be imported in the *truecontrol.keystore* and *truecontrol.truststore*,:
NAHOME/jre/bin/keytool -import -trustcacerts -alias nacacert -file vars-rct.XXX.fr.cer -keystore truecontrol.keystore
NAHOME/jre/bin/keytool -import -trustcacerts -alias nacacert -file Concatenated.cer -keystore truecontrol.truststore

- The new certificates are now operational. To enable a faster navigation within the HP-NA, the file *NAHOME/jre/adjustable_options.rcx* can be modified (optional) by adding the following line (wherever in the file but not in the last one because it has to be fixed):
<option name="startup/precompile/http.prefix">https://vars-rct.XXX.fr/</option>

- The truecontrol service must now be restarted to take changes into consideration:
/etc/init.d/truecontrol restart

- A message signaling a webpage not trusted could still appear at this time. Indeed, all the host's browsers may not be updated to this specific root certificate. The next update in the company will make it available.
To avoid waiting that long, every HP-NA user (all of them being probably administrators) should add manually the root certificate to the browser to avoid having this warning.
An example of how to do so using Mozilla is shown in the following: *Open menu* → *Preferences* → *Advanced* → *Certificates* → *View certificate.*
On the new popup in the tab *authorities* → *Import* → Browse to the root certificate and import it.

By doing so, no warning will be shown at connection.
If a warning appears again, then the administrator will be able to suspect a certificate obfuscation.

TROUBLE SHOOTING

- Be very careful when creating the CSR (Certificate Signing Request). Indeed if the "First and Last name" are badly fulfilled, then the certificate will not work.
As an example, the first request had been made for www.vars-rct.XXX.fr which didn't work because the good request was concerning vars-rct.XXX.fr .

- Be also careful when manipulating the files *truecontrol.keystore* and *truecontrol.truststore*. If one of them gets accidentally overwritten with a wrong one (e.g. if a mistake is done, the backup files should be used and an error could happen very quickly), then everything has to be started again from scratch because the key will not be corresponding anymore.

- The order of certificate in the keystores must be respected. If it isn't the case, an error "keytool error: java.lang.Exception: Failed to establish chain from reply" will occur during the import of the certificates.