Benj VDB

Superviser: Philippe Dekimpe

Internship tutor: Christopher Forbes

03/04/2017

**Vasco Data Security NV**  Kon. Astridlaan 164, 1780 Wemmel

Internship Report

Table of Contents

[VASCO Data Security NV 2](#_Toc482883807)

[What is Vasco? 2](#_Toc482883808)

[The Framework 2](#_Toc482883809)

[What is the SDTF? 2](#_Toc482883810)

[How does AWS fit into all of this? 3](#_Toc482883811)

[The Internship 3](#_Toc482883812)

[Goals 3](#_Toc482883813)

[Tasks 4](#_Toc482883814)

[Week by week resume 4](#_Toc482883815)

[Holdups 5](#_Toc482883816)

[Week 1 5](#_Toc482883817)

[Week 2 6](#_Toc482883818)

[Week 3 6](#_Toc482883819)

[Week 4 6](#_Toc482883820)

[Week 5 7](#_Toc482883821)

[Week 6 7](#_Toc482883822)

[Week 7 7](#_Toc482883823)

[Results 8](#_Toc482883824)

[Personal Experience 9](#_Toc482883825)

[The team 9](#_Toc482883826)

[The project 9](#_Toc482883827)

[The environment 9](#_Toc482883828)

[Conclusion 9](#_Toc482883829)

[What I got from it 9](#_Toc482883830)

[Bibliography 9](#_Toc482883831)

[Appendices 9](#_Toc482883832)

# VASCO Data Security NV

## What is Vasco?

Digipass, identikey, ias

Vasco Data Security NV is a company that specializes in data security as one might assume, offering services such as securing access, managing identities, verifying transactions, simplifying document signing and protecting high value assets and systems. It offers its services to many different sectors such as healthcare, government or even iGaming but it is mostly known for working together with banks. One of its more well-known products is the digipass which is a one-time password generator that is used in eID-readers (DIGIPASS 830 f.e.) that people can use to read their credit cards at home.

IAS is one of its products, it stands for Identikey Authentication Services and it offers secure and seamless access to corporate resources and applications of all kinds, from SSL VPNs to cloud-based apps, supports VASCO’s entire range of authentication and signature solutions, and simplifies authentication management for administrators and users alike.

However, what I mostly worked with was the SDTF. But there will be more explanations about this in the following chapter.

# The Framework

## What is the SDTF?

The STDF (Software Deployment & Testing Framework) is a framework that links the user, Jenkins and vCloud and allows us to easily deploy an environment with ias (IDENTIKEY Authentication Server) installed on it and run test suites in said environments. The way it works currently is that we use Jenkins to, for example, deploy an environment and install ias on the main VM. Jenkins will then use another VM (that we chose) that will serve as a test control host to launch the tests on. In order to deploy and install on an Ubuntu environment, Jenkins will generate the following line of code.

python c:\sdtf\suites\ias\installation\suite.py "3.12.0" "UbuntuS-16.04-x64\_PGSQL\_SSM" AutomatedInstallation --projectName="Sandbox" --testplanName="SDTF\_DEMO" --virt.user="lm-auto-wemmel" --virt.password="\*\*\*\*\*\*\*\*" --traces.user="lm-user" --traces.password="\*\*\*\*\*\*\*\*" --generic\_jenkins\_install\_job 2>&1

c:\sdtf\... refers to the location. In the sdtf there is a suites folder that contains all the test suites, seeing how we only want to deploy and install it goes into the installation subfolder and runs suite.py. (which kind of functions like a main.py, every suite subfolder has one and that is the one they use.) “UbuntuS-16.04-x64\_PGSQL\_SSM” indicates that the TCH will be an Ubuntu.

Once this is done the SDTF will fetch the information about the environment from a list containing all possible environments, stored as IASEnvironments objects. This contains, among other, all the virtual machines needed and the type of virtualization. In this case it would get virtualized on vCloud and thus after a short while we will see a vApp appear on vCloud.

\*vCloud picture\*

Of course this is only to install ias on an environment. The real purpose of the SDTF is to run suites to see if the ias can function correctly on a specific environment, to this effect it can either start from scratch or use existing virtual machines like the ones we would have after a “Deploy and Install” job.

I personally got to work a lot with an Ubuntu environment (without a part of its back-end due to configuration problems but that would only be a problem for make complicated test suites AKA regressions) and I would run build intakes on it. When doing so the SDTF would first ping the back-ends and ping STAF on the main VM. Once done it would check the content of the back-end and then create the necessary users, if they already exist it would delete them all before recreating them.

## How does AWS fit into all of this?

Vasco has been thinking about making the jump to the cloud instead of continuing to use vCloud. While the decision has not been made yet I have been tasked with making a proof of concept to give them an idea of the problems we could encounter when using AWS. A couple other of employees were also busy working with AWS in preparation, whether it was testing out AWS Lambda and see what it could offer to the company or trying to make an organization in AWS (to limit the permissions of users in a certain branch for example). In the long term the SDTF would be adapted to work for AWS mainly and vCloud would not be used any longer.

# The Internship

## Goals

As said previously my goal was to create a proof of concept that would help Vasco to make a decision.

This involved being able to create a virtual environment that would be able to be used by the sdtf, then adapt that environment so that it could work on the cloud and finally save it for deployment. Once done I would need to find a way to ensure that build intakes can be run against said environment like they would on vCloud and finally, I would need to ensure that regressions can also be run.

In case there were to be any time left, an additional objective was to containerize the environments with Docker. This objective however was not reached.

## Tasks

Originally my first task was to create an environment on AWS that would later be used as a template when deploying virtual machines for testing purposes. This would involve getting started with AWS, launching an instance, connecting to it and configuring it. However after many holdups it was decided to import the existing virtual machines used for testing (The Identikey of the Ubuntu environment). This way I would be able to concentrate on the problems that were caused by the VMs being on the cloud instead of always focusing on problems due to missing packages. (IAS can’t open an xtemp terminal in Ubuntu, are there packages missing, is the local variable not set correctly, is the PATH missing something, etc..). Naturally, importing existing VMs could come with it’s own problems. One of them being IP being fixed according to the VM, which existed within a certain IP range on vCloud, while the IPs would be randomly distributed on AWS.

After this my main goal was to get build intake running (install IAS + verify installed correctly?). This would be easier than running a regression/specific suites (all test suites) because only the main VM was needed for this and not the backends.

## Week by week resume

Before starting on the project I had been told that it was going to be quite difficult and that 7 weeks wasn’t exactly a lot of time and, indeed, this project was way more problematic then what I originally thought.

When starting the internship I spent my first week just reading code and trying to understand how vCloud, Jenkins, and the SDTF were related and what was going on inside of the VMs. Luckily I got a lot of help from my colleagues and vCloud gave me the ability to see what was going on inside of those VMs which helped me understand how they functioned and how they were configured. It was also during the first week that I had to make a task list in order to know what I would be doing during the following weeks and what my tasks were. One of those tasks for example was the recreation of the virtual machines, which I was not expecting because I originally thought I would be using the existing ones.

During the second week I finally started working with AWS (Amazon Web Services), before that I had been unable to use it because the account had not been created yet. Once created however, I got to test out the different functionalities of AWS, create instances (both linux and windows server), connecting to them, assigning security groups, replacing volumes, creating AMIs and so on and so forth. I created the instances for the Ubuntu environment which included an Ubuntu 16.04 instance with staf installed (Identikey), an OpenSUSE instance (Tivoli) and a Windows Server 2012 R2 instance (DC).

However after trying to launch an installation there were obvious problems. After talking with the tutor I understood there was a lot more necessary to make an environment “SDTF-enabled”. After bringing up my issues during the weekly meeting a coworker mentioned a file containing the necessary packages, programs, configurations to make an instance SDTF-enabled.

On the third week I would use the environments I made to try and install IAS on them, but it wouln’t even ping to them. I tried to understand why it wouldn’t be recognized but later it was discover that the company firewall closed off certain ports and did so as long as the Ethernet cable was connected even if I wasn’t using the LAN connection. The third week was also the week I discovered that the Tivoli was called like that because it was supposed to have IBM Tivoli Directory Services installed on it even though it wasn’t mentioned in the configuration files.

Lastly the third week was also the week were the whole QA team went to the AWSome Days Brussels to learn more about AWS and how it could benefit the company.

On the fourth week I looked into IBM Tivoli Directory Services and how to install it but there wasn’t a lot about how to install on OpenSUSE it and even less info about how to install it on OpenSUSE without a GUI. I got into contact with ITrequest and asked for help with the configuration and they referred me to two people who had worked on it. One of the responses I got about the necessary configurations needed told me that the VM they were currently using was made years ago and came from the Vasco team in Brisbane. That and the fact that a certificate of purchase may be needed to download IBM Tivoli Directory Services led to the decision that I would continue without Tivoli.

I got to start an installation and it appeared (midway through the installation) that there were a lot of packages missing. After downloading said packages there would be another error saying the instance wasn’t able to open an xterm terminal. I asked around but it was quite a strange error and even messaging ITrequest for a possible list of packages and necessary software didn’t bear any fruit. At that point it was decided it would be better to import the existing VM to AWS using S3. That would have its cost but it would allow me to do something other than configurations which I seemingly had spent most of my internship on. I went and imported the existing Identikey and started working with it the following week.

On the fifth week things started going at a faster pace, and I got past the installation part. The DC that I had configured myself seemed to work fine seeing as the framework could recognize it and was able to create users on it. However when it came to the actual test it would launch a web browser and try to navigate it but it was unable to and would crash. After asking around it seemed that this may be caused to the current version of selenium which was in charge of automating the browser based tests. I was also around then that I understood about how virtual environments worked because the SDTF used the packages from the virtual environment and not the ones I had installed through pip. After this the build-intake would be able to go though a decent amount of test cases before crashing.

## Holdups

### Week 1

* Mostly reading code, biggest holdup was my comprehension of the SDTF
* Couldn’t register as an employee properly -> contacted IT support, didn’t seem to work, got register manually.

### Week 2

* Couldn’t install STAF properly -> Needed to add java to environmental variable (PATH)
* Couldn’t connect to Windows instance -> Tried forcing a static IP on an AWS instance with a random IP on every boot (Static external IP only possible by purchasing Elastic IPs)

### Week 3

* Couldn’t ping to VMs -> Tried pinging through important ports before they were used to see if the firewalls allowed it, ports in use could be pinged (Linux: port 22 for SSH ok but port 389 for LDAP not ok).
* Couldn’t ping STAF (pinging done with STAF functions) -> STAFProc needed to be running for pinging to be possible, needed to add STAF/STAFProc to PATH + pinging needed to be done while in a virtual environment
* Trouble installing LDAP on SUSE Linux environment -> SUSE Linux (despite being called an LDAPMachine) needed to use IBM Tivoli Directory Services as a backend and not OpenLDAP.

### Week 4

* Couldn’t find IBM Tivoli Directory Services for SUSE Linux (no GUI) -> Contacted IT support in order to speak with whoever configured it originally -> Got in contact with 2 coworkers who worked on the VMs -> Got told the VMs were configured years ago by the team in Brisbane

🡪After discussion with internship tutor, decided to work without Tivoli. Build intake possible without the back-ends

* Still cannot ping STAF even though STAFProc is activated on Identikey, firewall deactivated on VM and AWS security group wide open -> \*
* Can only connect to a windows VM through RDP at random -> \*

For both \* -> Vasco WIFI and Vasco WLAN had different security measures. I had been told to use the WIFI and I did but the laptop still had the Ethernet cable connected and so it is possible traffic was split between WIFI and WLAN making successful connections random.

* Couldn’t install IAS on a machine even though the SDTF copied the install files into the VM, migrating to the N.Virginia region as suggested did not help (but reduces costs of deployment)-> installed missing packages in Identikey
* Manual IAS update on Identikey possible but result in an error -> possible that $TERM was not defined, installation cannot open an xterm console -> decided to import the Identikey VM
* Imported VM can’t communicate with SDTF -> change network configurations

### Week 5

* Can’t start build intake, needs to create users (in LDAP) -> use the other back-end (DC), works
* Browser is launched but code cannot use hover function (in order to simulate a mouse) -> needed a different version of selenium
* Different version of selenium doesn’t seem to do anything -> Doesn’t use the selenium on the computer but the selenium on the virtual environment that we work in.

🡪Changing to selenium 2.53.6 also made the SSL check pass automatically, speeding up the tests

* Couldn’t launch multiple of the same environments at once -> previous code relied on vCloud raising an error if two environments had the same name, AWS doesn’t -> manually raise error
* Could only install IAS if network was configured manually -> automated network configuration (using STAF)

### Week 6

* During the create\_custom\_report test the test failed and the error indicated that a command violated the foreign key contraints -> that was supposed to happen, create\_custom\_report would generate those errors but would in normal circumstances be ignored if we were to run the test in testlink (instead of running it in debug mode).
* Couldn’t install correctly anymore -> first installation succeeded due to luck, all the others failed because of a timeout seeing as how installing IAS on an AWS instance takes more time.
* Random errors during webserver tests -> During those tests, the code takes control of the mouse and keyboard and doing anything else (like clicking or writing) while it’s busy has a chance of generating errors.
* Can’t create users with the default DC -> forgot to save changes -> DC password needed to be changed
* Launching the script when not in debug mode would take way longer due to the errors generated because there was no FTP server to send logs to -> created an AWS instance hosting an ftp server.
* Couldn’t use ftplib functions even though the ftp server was available -> FTP server needed to be used on active mode even though it was used in passive mode exclusively up until now

### Week 7

* Couldn’t access FTP server after booting the instance cointaining the server -> on top of changing IP’s in both the code and the vsftpd.conf file for it to fit with the IP of the instance, the vsftpd service has to launch on boot.
* Installation stops working before installing Vasco Web Administration Services -> wrong number of arguments submitted at the installation (even though this part is automated) -> There was a comment in a tcl script but in that language comments can only be placed if the # is the first symbol of a line, because it followed a normal line of code it was considered as an argument
* Couldn’t launch a windows instance with the SDTF -> had to follow a naming convention that was different for windows. (main instance had to be called W2K12R2ESSDC instead of Identikey)
* Could only communicate with STAF after logging in manually -> Autologgon took care of logging in, but it was still configured with the original instance password.
* Cannot create iasadminuser on Windows ->
* Cannot fill in domain name during installation on Windows -> iasadminuser necessary?

## Results

The project was quite ambitious and might’ve honestly been too much for me. However when discussing it with the internship tutor before the decision to take me in as an intern was made, they told me they were fully aware that all objectives may not be reached.

And indeed the project was quite difficult, understanding how the framework worked and how to debug it took some time and there were a lot of concepts I had to get introduced to (like a virtual environment for example). However after discussing with the tutor about the importance of asking for help instead of wasting time trying to understand how the code works on my own I started building up a decent pace and would usually be able to resolve issues in a day. After that I also started gaining more confidence in what I was doing and in turn it became easier for me to discuss problems with my coworkers.

In the end I managed to achieve two out of the 3 goals I had, manage to have the SDTF deploy an environment and install IAS correctly on said environment, and run a build intake on said environment.

I was, however, unable to run a regression on an environment mainly because there was only 1 week left by the time I was done with build intakes and rushing to try and get regressions to work might not have been worth it, especially considering I was missing one of the back-ends on the environment I was using.

Instead I spent the last week trying to redo build intakes but this time on a Windows environment, it came with its own problems of course but the considering the time it took to get build intakes on Ubuntu to work, I wasn’t too unhappy about how far I got into the it. And seeing how this VM had a Graphical User Interface, I even got introduced to AutoIt which basically takes to place of tcl to automate the installation process, so it was quite nice to be introduced to another part of the SDTF this late into the internship.

# Personal Experience

## The team

## The project

## The environment

During this internship I got to work a lot with Linux, the cloud, virtual machines and automatisation (tcl, selenium, AutoIt were used to automate different parts and I got to see how they work) and I feel like I learned a lot

# Conclusion

## What I got from it

# Bibliography

* https://www.vasco.com/about-vasco/index.html

# Appendices

LOG