Facultat d'Informàtica de Barcelona
Grau en engineyeria Informática
Departament d'enginyeria de Computadors
Treball final de grau

Office 365, Azure AD, and Exchange Online audit automation

Author: David Soldevila Puigbí
Director: Davide Careglio
Co-director: Rafael Estevan

January 20, 2022

Abstract

The present document will study the update of Ackcent's Office 365, Exchange Online, and Azure AD audit services. Dividing it into three main steps, being the first one migration to a server-less solution of an email audit infrastructure aimed to dynamically review the configuration by sending emails with different threat indicators. The objective for the first part is to make an independent platform to audit without dependencies on third-party infrastructure providers.

The second part will be to upgrade a security control list with new checks to review the configuration deeply. Finally, the possibility of automatizing the control analysis will be investigated using third-party tools and custom implementations. The second and third stages of the project will have objectives to improve the quality of the controls being analyzed for the audit and reduce the amount of time spent by the security architects to perform an audit.

Contents

List of Tables

List of Figures

1 Context

1.1 Contextualization

Ackcent cybersecurity is a company born in Barcelona 7 years ago. I started working there about two and a half years ago. I first started as a member of SOC (Security Operation Center) deploying, implementing, and tuning an Endpoint Detection and Response (EDR) solution into a multinational company with more than seven thousand devices distributed around the globe. After my first project, I joined the Security Architecture team where I learn about security on on-premise, cloud, and hybrid infrastructures.

Security Architecture is the smallest team on Ackcent, it is comprised of five employees, one being the CISO[1] of the company, which implies that he has reduced the amount of time to work on external projects, one engineer that is fully dedicated to an external project with a large company, and finally, the other three members, among whom I am included, are available for short projects. Each member can handle multiple projects at once. A short project can be a one-shot, like audits or third party solutions implementations, or service deployments, like deploying the EDR solution or setting up a SIEM[2] and connecting the solutions to our SOC team.

Auditing cloud infrastructures became one of our main kinds of projects. The different types of audits were assigned to a different engineer in order to improve it and then share the knowledge that was obtained with the rest of the team. For this project we will be focused on to the Office 365 (office applications and services), Azure Active Directory (user management service), and Exchagne Online (mail exchange service).

[1] Chief Information Security Officer

[2] Security Information and Event Management

1.2 Basic concepts for the audits

1.2.1 Audit documents

The audit consists of two different documents. The exercise is divided in order to prove that the configuration is affecting email discrimination.

- Static test audit: this document consists of a review of different controls in various categories, DNS[3] records[4], Microsoft 365 controls[5], and protection mechanisms, and Office 365[6] controls and protection mechanisms. Each section has different points that are evaluated following internal rules, CIS[7] benchmarks[8], and Microsoft recommendations.

- Dynamic test audit: this document consists of a result show of sending SPAM emails, phishing emails, and attachments to every published exchange server. The possible results for each email are, “Received”, “Received as junk”, “Received with blocked attachment” or “Not received”. Each test is sent to all published exchange servers, even if it is not from Microsoft, although they are not reviewed in the static audit. This part is aimed to view how the configuration applied is affecting the end-user service and how attackers can exploit the breaches.

The documents will have inforamtion about Office 365 and Exchange Online service and user configuration that can effect the usage of these tools. The audit will not deal with general Azure configuration or similar services from other platforms.

[3] Domain Name System, is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network [1]

[4] DNS Record: is a public list of instructions that provides information about a domain.

[5] Office 365, is a web portal to access to Microsoft 365 applications.

[6] Microsoft 365, is a set of office applications.

[7] Center for Internet Security, `https://www.cisecurity.org/`

[8] CIS benchmarks: public and open documents that provides a guideline to secure your infrastructure and services.

1.2.2 Stakeholders

The project is aimed at different stakeholders, from the customer IT team to the Security Architecture team. The project will enhance the results and will suppose less work time for the SA team.

- Security Architecture team: the SA team will be able to perform audits with more accuracy, without the human error factor, and deliver the final document in less time. This model will propitiate the proactivity to implement new controls to improve the quality of the audit.
- Customer IT team: the IT team will receive an audit with more detail giving an extensive look at the configuration of his infrastructure. They are also be beneficiated of the shortest delivery time.
- Customer Admin team: the admin team, who is interested in the general view of the company and how the internal mechanisms are working, will be able to view a simple and legible summary with charts that will represent the overview of the configuration.

2 Justification

2.1 Manual audit issues

On average, an audit requires around 25 hours of effort for every infrastructure. This together with the increased amount of infrastructures and the low availability from the architects, results in delays of two weeks per audit thus affecting other projects.

The first action, before planning the acutal assesment, was to minimize the effort was to automatize the execution for some of the controls using PowerSHell[9] command. The output of th ecommand were saved in a plain text file. After the execution the architecet had to analyze the data obtained and compose the audit docuemnt.

2.2 Proposed solutions

With those changes, the audit effort was reduced by more than 7 hours. However, the reduction was not enough. The team decided to evaluate if there was a way to implement an automatized software to get a file with the controls, actual configuration, and recommendations. I checked four different options to implement it, Prisma Cloud, Azucar, Cloud Sploit, and custom software.

2.2.1 Prisma Cloud

From Palo Alto Networks, Prisma Cloud is multiplatform visibility, compliance, and governance tool. It is intended to keep your infrastructure monitored for a long period. That implies that the licenses are not available for one-shot projects and one license cannot be used in multiple customers.

However, it might solve many of our issues, the price of the solution was considered

[9] PowerShell: is a cross-platform task automation solution made up of a command-line shell, a scripting language, and a configuration management framework. PowerShell runs on Windows, Linux, and macOS[2]

a critical problem for the sales team. The price and the contracts with Palo Alto are confidential so it cannot be expose.

2.2.2 Azucar and Cloud Sploit

On the other hand, Azucar and Cloud Sploit, one from the NCC group and the other from Aqua. Both of them being free and open source. Azucar is intended to track the security of your Azure assets. However it is focused on a general Azure audit and is not focused on Office 365, Exchange Online, Azure Active Directory, and many of the controls that are crucial for us are not implemented.

The last tool put under scrutiny was Cloud Sploit, a multi-platform tool that is made to detect security risks in cloud infrastructures and is capable of returning a list of misconfigurations. As their expose on this official website CloudSploit started as an open-source project to help developers find security misconfigurations in cloud accounts [3]. As Azucar, cloud sploit has the same disadvantages.

2.2.3 Custom Script

The last option I studied was full custom automation, extending the existing script with more controls and building a PDF with a markup language.

This solution beat others, most of them free and open-source, because of the flexibility of choosing the wanted controls and the text that follows it. It is more secure as you use Single Sign-On capability and official connection methods instead of API keys that can be leaked easily. Simplicity and maintenance are easier because it is only intended for our audits, has no external dependencies, and has no license cost and no license limitations.

3 Scope and objectives

3.1 Objectives

The main objective is to reduce the amount of effort associated with an audit. In order to make it possible I am going to implement a script that will generate a PDF file with a description of the control, the actual configuration, and the recommendations.

Our project has 3 major objectives:

1. Black Hat server migration: migrate the server responsible for sending the emails for the dynamic test audit to a non-server-dependent solution[10].

2. Update of the controls list to include the new Azure capabilities and new directives. Also documenting the requirements and processes for the new controls.

3. Create the program that queries the information and generates a document with control description, control results, and control mitigation (if it is required).

Down below, in the task section, I will define with more granularity the way that the objectives will be divided in order to be easier to track and accomplish.

3.2 Scope

The scope is the configuration for Office 365 and Exchange Online in addition we are going to consult the configuration of the users that is related with the authentication, risk management and app registration.

As it was mentioned before, this document will not deal with general Azure audits or audits from other platforms, it only include audits to email and communication services in

[10] Non-server-dependent solution: is an implementation that implies no external servers will be needed to perform the execution. That the service will require all the software installed at the host computer and the operability will be independent of alien machines.

Microsoft Azure. However, some of the solutions and ideas that are mentioned are being applied with other types of audits or other kinds of projects that are being developed by the architecture team or other Ackcent teams.

3.3 Requirements

The requirements for the script that will retrieve the information and generate the markup document will be a system with PowerShell and ExchangeOnlineManagement[11] and MSOnline[12] modules installed.

The requirement for the audits is to have global reader access to the audited account. It must be a global reader to run some of the audit commands, like Get-OrganizationConfig, that retrieve information from the global scope.

3.4 Risks

The most remarkable associated risk is time management. Since this project must be collated with other projects and the development of the present document, it is possible that the established deadlines can be delayed.

The risk of a system failure. It is always possible to has technical problems and lose a part, or the total, of the project because all the information is stored in devices that can fail. Even if the information is stored in the cloud there is an associated risk of data loss. It will discussed how this kind of risk will be reduced.

The risk of not obtaining good or coherent results is ever-present. I know that a technologies failure, unexpected difficulties for certain features, or unexpected change of requirements can affect the development of the project.

[11] More information on: `https://www.powershellgallery.com/packages/ExchangeOnlineManagement/2.0.5`

[12] More information on: `https://www.powershellgallery.com/packages/MSOnline/1.1.183.57`

Finally, I want to highlight the possibility of incidentals. We have suffered a global pandemic, floods, volcano eruptions, earthquakes, etc. We must be aware of these kinds of risks and try to have a robust workflow to not interrupt our work, even if it is from a company or studies.

4 Methodology and rigour

The complexity comes out from the large number of controls that must be implemented and that implementation must be flexible to embrace every possible configuration. Some other part, as the limited dedication and the SLA and governance compliance of the technical resources, are points that adds complexity to the project.

To implement the most reliable solution, with the highest efficiency and following the tradition of Ackcent work methodology, I will get help from the SCRUM methodology, supervised by our Scrum Manager, Óscar García. It is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum is a lightweight framework that helps people, teams, and organizations generate value through adaptive solutions for complex problems [4].

To scheduling of the tasks and keep track of the done and pending tasks will be done by using a physical diary and a notepad to take notes. It has the risk of being lost, but it is what I am used to.

As it was mentioned before the project following up is made weekly as every project form architecture. With this follow-up methodology we ensure that, even if I am the only developer of the project, other members of the team can contribute by giving ideas, support, and possible implementations. Also, by sharing the problems and discoveries with other everyone can help with his point of view.

5 Tasks

5.1 Summary table

The actual section will reduce all the tasks and sub-objectives in a table with the task and sub-task identifier, the name of the task, and the effort needed to complete it in hours.

Task	Sub-task	Name	Hours of effort
BHM	1.1	Study the network requirements	2
BHM	1.2	Study the system requirements	2
BHM	1.3	Search for a platform that accomplish the requirements	4
BHM	1.4	Deploy a solution	10
BHM	1.5	Test the solution to check it works as before the migration	5
BHM	1.6	Publish the solution to the Architecture Library	1
CU	2.1	Check actual controls and identify the shortcomings	5
CU	2.2	Review the CIS benchmark and include wanted controls	10
CU	2.3	Update and reorganize the control list	10
SD	3.1	Develop start up and connection module	10
SD	3.2	Develop start up and connection module	15
SD	3.3	Develop application permissions controls module	15
SD	3.4	Develop data management controls module	15
SD	3.5	Develop email and exchange online security controlsmodule	15
SD	3.6	Develop auditing controls module	15
SD	3.7	Develop storage controls module	15
SD	3.8	Develop mobile device management controls module	15
SD	3.9	Documentation	5
		Total	161

Table 1: Summary of tasks
Own compilation

5.2 Task description

5.2.1 Black hat server migration (BHM)

The first point to accomplish is to migrate the service that is used to send the emails for the dynamic audit. Because the server where it is hosted is going to be decommissioned this task is going to be the first. The main obstruction is that the solution must run on the workstation without the requirements of the system and network.

BHM - 1.1 Study the network requirements (2 hours of effort).

The first job to do is to identify the specific network requirements. It includes IP requirements, firewall requirements, the number of network interfaces, and bandwidth.

BHM - 1.2 Study the system requirements (2 hours of effort).

The study of the system requirements will conclude with a list of software that must be compatible with the operating system that will be used to deploy the final solution. In addition, hardware requirements will also be discussed.

BHM - 1.3 Search for a platform that accomplishes the requirements (4 hours of effort).

Some solutions will be listed and compared to get a rank of how compliant are with the requirements. Then the one with more facilities will be the chosen to be developed. This task is dependent on BHM - 1.1 and BHM - 1.2.

BHM - 1.4 Deploy a solution (10 hours of effort).

This task will consist in implement the solution by creating the platform that will be used and installing and deploying all the software and scripts that will be used for the dynamic audits. This task is dependent on BHM - 1.3.

BHM - 1.5 Test the solution to check it works as before the migration (5 hours of effort).

In this part, I will launch a series of executions to test all cases and test that all controls are executed without any issue and the behavior is still the same. This task is dependent on BHM - 1.4.

BHM - 1.6 Publish the solution to the Architecture Library[13] (1 hour of effort).

I last part of this migration will be to share the final solution and the documentation that had been done on the previous tasks with the rest of the team. It will be published at the SA[14] site on SharePoint[15]. This task is dependent on BHM - 1.5.

5.2.2 Control update (CU)

The second step is to review the controls that are implemented and review the new CIS benchmark for Microsoft 365[16]

CU - 2.1 Check actual controls and identify the shortcomings (5 hours of effort).

The actual list was made about one year ago and some of the control implementations are deprecated or have some changed specifications. This first step will consist of reviewing, testing, and repairing, if it is necessary, all the controls that were established.

CU - 2.2 Review the CIS benchmark and include wanted controls (10 hours of effort).

After reviewing the implemented controls and well knowing the audit requirements, I will review the CIS benchmark for Microsoft 365 to implement all controls that I consider relevant to check. This task is dependent on CU - 2.1.

CU - 2.3 Update and reorganize the control list (10 hours of effort).

After identifying the missing controls I will add them to our list. It implies adding a description of it, the method to check it, and the recommended configuration to the

[13] Architecture Library: site that is only accessible for members of Security Architects where the team shares all the documentation and tools.

[14] SA: executive abbreviate used to name Security Architecture team.

[15] SharePoint: Microsoft 365 collaborative platform. In Accent, this is the official method to share information.

[16] CIS benchmark for Office 365: Available on `downloads.cisecurity.org`.

previous list. In addition, it might be necessary to reorganize the controls creating more sections to make the audit more legible. This task is dependent on CU - 2.2.

5.2.3 Script development(SD)

The final part of the project is to implement and merge all different controls into a single script that creates a document with the information of the tenant. It is mandatory to have a modular implementation, the readability, reliability, and maintenance of the platform must improved. I divided the development of the program into eight parts and an extra part for the documentation.

SD - 3.1 Develop start-up and connection module (10 hours of effort).

The first module to implement is the start-up and connection module. It will be the first module that will be launched and it will install and load all PowerShell modules and connect to the services.

SD - 3.2 Develop account and authentication controls module (15 hours of effort).

This module will launch a series of controls that will check the configuration of the account and authentication methods. This task also includes the testing and debugging of the module. The specific controls will be defined in future sections of the document. This task is dependent on SD - 3.1.

SD - 3.3 Develop application permissions controls module (15 hours of effort).

This module will launch a series of controls that will check the configuration of the application permissions settings. This task also includes the testing and debugging of the module. The specific controls will be defined in future sections of the document. This task is dependent on SD - 3.1.

SD - 3.4 Develop data management controls module (15 hours of effort).

This module will launch a series of controls that will check the configuration related to data management. This task also includes the testing and debugging of the module. The specific controls will be defined in future sections of the document. This task is dependent on SD - 3.1.

SD - 3.5 Develop email and exchange online security controls module (15 hours of effort).

This module will launch a series of controls that will check the security configuration of email and exchange online. This task also includes the testing and debugging of the module. The specific controls will be defined in future sections of the document. This task is dependent on SD - 3.1.

SD - 3.6 Develop auditing controls module (15 hours of effort).

This module will launch a series of controls that will check the auditing configuration. This task also includes the testing and debugging of the module. The specific controls will be defined in future sections of the document. This task is dependent on SD - 3.1.

SD - 3.7 Develop storage controls module (15 hours of effort).

This module will launch a series of controls that will check the configuration storage services. This task also includes the testing and debugging of the module. The specific controls will be defined in future sections of the document. This task is dependent on SD - 3.1.

SD - 3.8 Develop mobile device management controls module (15 hours of effort).

The last module will launch a series of controls that will check the mobile device management (MDM) configuration. This task also includes the testing and debugging of the module. The specific controls will be defined in future sections of the document. This task is dependent on SD - 3.1.

SD - 3.9 Documentation (5 hours of effort).

A user-friendly documentation document will be attached to the resulting software to help another member use and generate audits without any trouble. This task will be developed in parallel with the development.

5.3 Gantt charts

In this section, I will present some Gantt charts for the project. The general Gantt chart will represent the entire duration of the project divided by the three major objectives/tasks defined in previous sections of the document.

The concrete Gantt charts will be focused on each task and will be more detailed. Each Gantt chart will have one bar for each sub-task with the dependencies between each one.

5.3.1 General Gantt chart

As it can be seen from the general Gantt chart the first objective is independent of the rest of the project. However, the Control Update must be performed before the Script Development because the program will be defined after checking and updating the current control list.

The estimated duration for the project is 5 months. To prevent possible affectations of time I decided to spread out the invested time in more weeks.

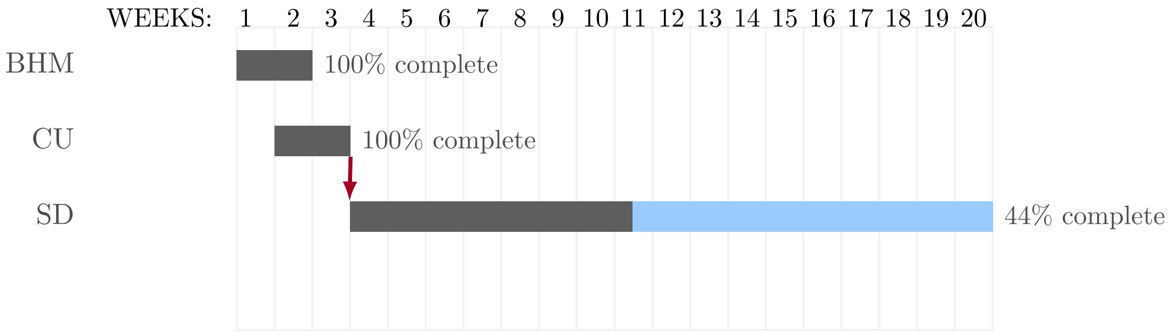


Figure 1: Gantt chart for the general project

Own compilation

5.3.2 Gantt chart for Black Hat Migration

As the chart shows the dependence of the task for the Black Hat server Migration is almost sequential. Excepting from the two first tasks, that will be developed in parallel, the rest of the tasks are only dependent on the previous one.

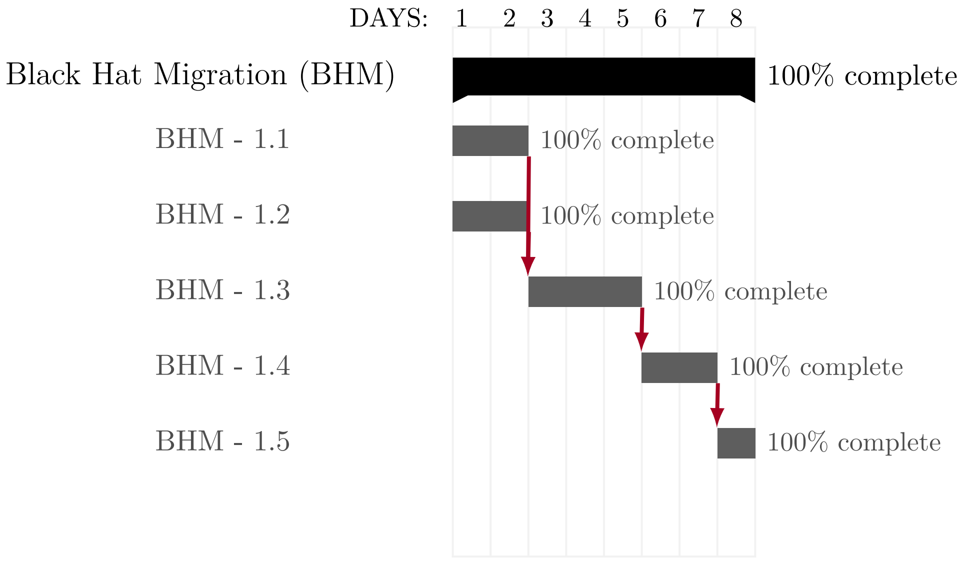


Figure 2: Gantt chart for Black Hat Migration task

Own compilation

5.3.3 Gantt chart for Control Update

Like the first diagram, the sequence of sub-tasks for this task is sequential. Note that the first task will be performed in a really short time. That is because this was been a recurrent topic with the rest of the members of the team.

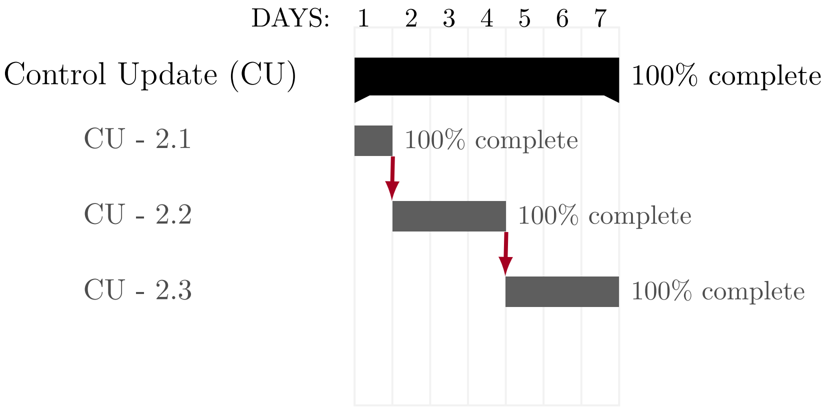


Figure 3: Gantt chart for control update task

Own compilation

5.3.4 Gantt chart for Script Development

As it can be seen in the first diagram, this task is the most extensive in time. However, it is divided into sub-tasks with a similar length as the other sub-tasks. Note that in this sequence of tasks the unique dependence is the first module. Without the first module, any control can be tested, so from the first task there is a dependence arrow to the rest of the tasks. For the last one, documentation, that one will be developed in parallel from other tasks.

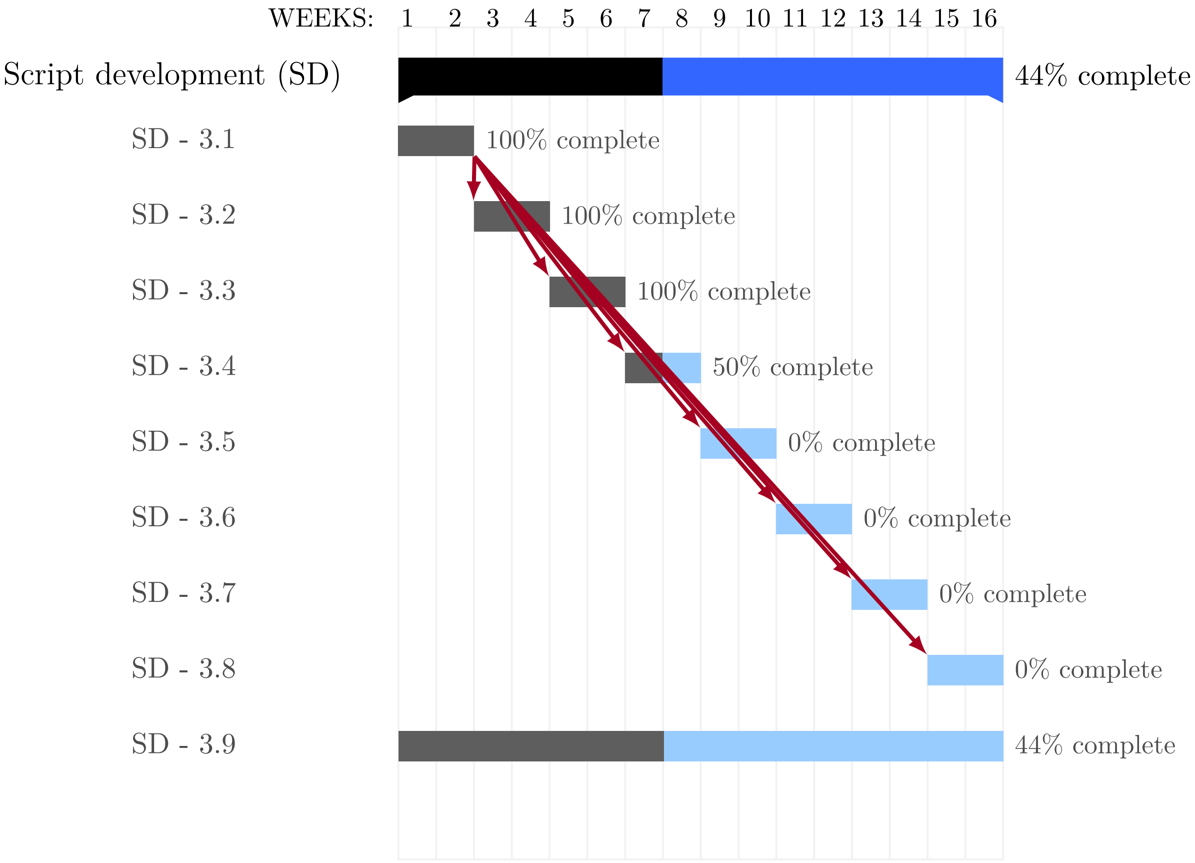


Figure 4: Gantt chart for Black Hat Migration task

Own compilation

5.3.5 Gantt chart conclusions

The Gantt chart is distributed in weeks as it represents the starting and ending weeks for each assignment and it also lets us a view of how much dedication will be designated in a concrete task or sub-task. Taking the entire Black Hat Migration task as an example, the effort in hours is 20 hours, but it is distribute along two weeks, with 40 work hours each, the dedication will be:

$$dedication = \frac{e}{w} = \frac{20}{2 * 40} = 0.25 \rightarrow 25\%$$

Being e as the amount of time dedicated to a task and w as the number of work hours.

5.4 Human and material resources for tasks

For each part, I will have as implicit material resource our workstation and the software that is provided.

5.4.1 Black hat server migration resources

The human resources for this task will be minimal. The effort is twenty hours, the lowest in compression with the other two tasks. No additional help is planned to be necessary from other departments nor IT actions will be needed.

This part of the project is the one that will have more material resources. It will end with the exportation of a platform that was hosted on a virtual server on Amazon Web Services (AWS). The possible will be a machine that can run the audit scripts, it can be from a self-hosted virtual machine from a dedicated physical computer.

5.4.2 Control update

The human resources for this task will be 25 hours from my availability and it also include the time that will be used in the weekly follow-up meetings, we estimate it will be an hour for the entire team, it will be an increase of 5 hours. The result is 30 work hours of security architecture.

The material resource that is needed for this part is only the CIS benchmark, which is free and open.

5.4.3 Script development

The development of the script will be the one demanding most human resources. I estimate that it will take around 115 hours to finish it, including the development, and the testing. Additionally, I estimate 5 more hours to write and prepare the documentation that will follow the program.

No additional material resources will be needed for this part.

6 Risk management

As I explained before some risks can affect the development of the present project. Now I am going to comment one by one on the mitigation plan.

6.1 Time management

In my opinion, the most probable risk is the lack of time. To prevent it I distributed the time assigned in twenty weeks, eight more than the first estimation. With this method, I am going to ensure that I will have enough time to develop the project and write the actual document.

It is possible that the project will be continued under development after the delivery of this report. In that case, the document will explain the implemented part and how it was developed and will propose an extension of the plan.

6.2 System failures

A system failure can imply data losses. To minimize the risk of not being able to restore lost documents the following methods will be applied:

Code and internal documentation will be stored at the working laptop and synchronized with One Drive. With this redundancy, I can ensure that if the physical disc of the laptop fails I can restore the document from the cloud. If the cloud is not available, even if I don't have an internet connection, the service is down or is affected by an incident, the offline copy of the document will still be available.

Actual document will be stored in my laptop and will be synchronized with Overleaf[17] As the same way as the previous method the document has an offline copy and it is also available from the web application.

[17] Overleaf: is an online platform to write documents in LaTeX.

In both cases, if the online service and the physical disc fails at the same time the data will be lost. However, the probability of experimenting with this is extremely low.

6.3 Solution incorrectness

The possibility of obtaining unwanted results that do not accomplish the needs. Despite learning from your failures is the best way to improve yourself I have some mitigations to ensure good results and a working implementation. Before starting this project I had been investigating on this topic and I saw specifications and command-line instructions to review the controls using similar mechanisms to the ones expected for this project.

6.4 Accidents

The chance of an impedance of developing our regular work has been proved that is more present than we used to believe. In order to mitigate that, we have some services that can help to work correctly from everywhere and, if it is necessary, with any device. From VPN[18] to an EDR helps us to maintain the security even if the device is not enrolled with the infrastructure.

6.5 Material and services

I will include here the likelihood of needing more resources, like a new computer or online services. It is planned to develop the solution to dispense online services so it is improbable that I am going to need it. However, the physical resources such as laptops or peripherals are managed by the IT team and any special need can be solved. The Gantt chart is distributed in weeks as it represents the starting and ending weeks for each charge and it also let us a view of how much dedication will be designated in a concrete project and

[18] VPN: Virtual Private Network, is an encrypted connection over the Internet from a device to a network. The encrypted connection helps ensure that sensitive data is safely transmitted [5].

labor. Taking the entire Black Hat Migration project as an example, the effort in hours is 20 hours, but it is distribute along two weeks, with 40 work hours each, the dedication will be:

$$dedication = \frac{e}{w} = \frac{20}{2 * 40} = 0.25 \rightarrow 25\%$$

Being e as the amount of time dedicated to a task and w as the number of work hours.

6.6 Deadline

The deadline for the practical part of the project is the third week of November. Two additional weeks were added to ensure I have enough time to solve problems in case of I did not get correct results or stoppers effect the development. If the results gotten at the deadline are not solid the project will be extended until I get a suitable solution.

The deadline for the actual document is 20 of December. That will give us about one month to compile information about the project and the documentation.

7 Budget

7.1 Costs

7.1.1 Expenses identification

Here I am going to discuss the cost of the project. It will involve the material costs, all from the computer equipment to the online platforms, and staff costs.

OpEx is really significant for this project. It involves more hours of work and services as Microsoft 365[19], Windows, and Azure AD.

Resource	Cost
Microsoft 365 licence	16,90 €/user per month
Microsoft Windows Pro 10 for Workstations licence	19.20 €/user per month
Azure AD Premium P2 licence	7,59 €/user per month
Security Architect staff	12.43 €/hour per worker

Table 2: OpEx for this project
Own compilation

In this section, I will not take care of supplies because the actual working model is teleworking. However, to manage the new work model the company had to upgrade the Microsoft 365 E3 subscription to an Azure AD P2, implying a migration from the on-premise active directory to the Azure Active Directory.

The software for the development used in this project is free and open-source, so it does not bear a cost.

7.1.2 Expenses estimation

I must add to the cost estimation of the invested time of Security Architecture follow-up meets used to talk about this project. I estimate that was about 5 minutes per week on

[19] Microsoft 365 license includes office software, the operating system license is not included in the pricing.

average.

Task	Description	Time/Unites	Cost	Subtotal
OpEx	Microsoft 365 licence	3 months per 1 user	16,90 €	50,7 €
OpEx	Microsoft Windows Pro 10 for Workstations licence	3 months per 1 user	19,20€ €	27,6 €
OpEx	Azure AD Premium P2 licence	3 months per 1 user	7,59 €	22,77 €
SA follow-up	Security Architect staff salary	4 member, 5 minutes 4 times per month, 3 months	12.91 €/hour	68,85€
BHM 1-1	Security Architect staff salary	2 hours	12.91 €/hour	25.82 €
BHM 1-2	Security Architect staff salary	2 hours	12.91 €/hour	25.82 €
BHM 1-3	Security Architect staff salary	4 hours	12.91 €/hour	51,64 €
BHM 1-4	Security Architect staff salary	10 hours	12.91 €/hour	129.10 €
BHM 1-5	Security Architect staff salary	5 hours	12.91 €/hour	64.55 €
BHM 1-6	Security Architect staff salary	1 hours	12.91 €/hour	12.91 €
CU 2-1	Security Architect staff salary	5 hours	12.91 €/hour	64.55 €
CU 2-2	Security Architect staff salary	10 hours	12.91 €/hour	129.10 €
CU 2-3	Security Architect staff salary	10 hours	12.91 €/hour	129.10 €
SD 3-1	Security Architect staff salary	10 hours	12.91 €/hour	129.10 €
SD 3-2	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-3	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-4	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-5	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-6	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-7	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-8	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-9	Security Architect staff salary	5 hours	12.91 €/hour	64.55 €
Total				2351.71 €

Table 3: Expenses estimation
Own compilation

7.2 Contingencies

We are aware of the possibility of setbacks for any kind of project. To mitigate it some time ago Ackcent Admin department made a study of the completed projects. The result was a table with different percentages depending on: the type of the project, the number of employees, and the estimated effort.

For this project, we are going to apply an extra 10% of the budget.

7.3 Incidentals

At least, we have to take care of the possible cost of the alternative plans to mitigate incidents. Each possible incident has a probability associated. Keep in mind that part of this section is shared with other projects and it will not be charged to the actual project. In fact, in Ackcent since the global pandemic, it has been charged in a special project. The shared costs are listed in the first part of the table, separated by two horizontal lines.

Incident	Probability	Cost
Laptop	10%	1.793,53 €
Peripherals	30%	50 €
Increment of dedication for BHM	30%	92.95 €
Increment of dedication for CU	30%	96.83 €
Increment of dedication for SD	40%	464.76 €
Total		654.54 €

Table 4: Incidentals
Own compilation

7.4 Total cost estimation

The cost estimation for the project is represented in the following table:

Task	Description	Time/Unites	Cost	Subtotal
CapEx	Microsoft 365 licence	3 months per 1 user	16,90 €	50,7 €
CapEx	Microsoft Windows Pro 10 for Workstations licence	3 months per 1 user	19,20€ €	27,6 €
CapEx	Azure AD Premium P2 licence	3 months per 1 user	7,59 €	22,77 €
SA follow-up	Security Architect staff salary	4 member, 5 minutes 4 times per month, 3 months	12.91 €/hour	68,85€
BHM 1-1	Security Architect staff salary	2 hours	12.91 €/hour	25.82 €
BHM 1-2	Security Architect staff salary	2 hours	12.91 €/hour	25.82 €
BHM 1-3	Security Architect staff salary	4 hours	12.91 €/hour	51,64 €
BHM 1-4	Security Architect staff salary	10 hours	12.91 €/hour	129.10 €
BHM 1-5	Security Architect staff salary	5 hours	12.91 €/hour	64.55 €
BHM 1-6	Security Architect staff salary	1 hours	12.91 €/hour	12.91 €
CU 2-1	Security Architect staff salary	5 hours	12.91 €/hour	64.55 €
CU 2-2	Security Architect staff salary	10 hours	12.91 €/hour	129.10 €
CU 2-3	Security Architect staff salary	10 hours	12.91 €/hour	129.10 €
SD 3-1	Security Architect staff salary	10 hours	12.91 €/hour	129.10 €
SD 3-2	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-3	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-4	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-5	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-6	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-7	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-8	Security Architect staff salary	15 hours	12.91 €/hour	193.65 €
SD 3-9	Security Architect staff salary	5 hours	12.91 €/hour	64.55 €
Contingencies	10% of the estimated cost			235.17 €
Incidentals	Incidental costs showed at Table 5			654.54 €
Total				3241.42 €

Table 5: Total cost estimation
Own compilation

7.5 Amortisation

The amortization for the project in hours will be calculated by dividing the development time of the project, calculated at previous sections, by the average elapsed time for an audit made with the first methodology.

$a_t = \frac{t_p}{t_a} = \frac{161}{35} = 4.6$

Being t_p the development time of the project and t_a the average time for an audit made with the first methodology. a_t is the amortization index, in other words, the number of audits that will be worth the time of the project.

As it can be seen, the time invested in this project will be amortized by doing five audits. By taking the number of audits made in 2021, the number of hours must be amortized.

In terms of expenses, at first, an audit would cost us about 35 hours of work, around 15 hours of Amazon Web Services EC2[20] t2.small[21] on Europe Ireland data center[22] computing time. The cost for a manual audit will be: $p = a_p + s_p = 35h*12.91/h + 15h*0,02/h = 466.87$ being a_p the price for the Security Architecture hours and s_p the price for the AWS EC2 t2.small hours. Using the new software the price for one audit will be: $p = p_p/n + a_p = 2351.71/n + 2h * 12.91/h = 2351.71/n + 25.82$ being p_p the project price, n the number of audits made with this system and a_p the price for the Security Architecture hours.

Using the previous equation, after 3 audits the cost of the project will be amortized. The entire price of the laptop, peripherals, and Microsoft licenses to this project charged the, so in reality, it will be amortized with fewer audits.

The amortization of the laptop and the peripherals will be represented for the 3 months, the estimated elapsed time for the project. The computing tools will be amortized as EPI[23].

The annual amortisation is defined with the following formula:

$a = \frac{c-s}{u_s}$

Where a is the amortization for one year, c is the initial cost, s is the salvage cost and u_s is the useful life. I estimate a cost for the laptop and the peripherals of 1300€ and

[20] Amazon Web Services Elastic Compute Cloud (AWS EC2): is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers [6]

[21] t2.small: virtual server with 2 vCPUs, 2 GB of RAM with a price of 0.0208$ (0,02€) per hour

[22] Europe Ireland data center: one of the data centers owned by Amazon that gives service AWS platform.

[23] Equipo de Procesamiento de Información: Information Processing Tools.

salvage of 300€, mostly from the reusable components of the computer. The useful life for a computer in Ackcent is estimated at 3 years. The applied equation for our case will be:

$a = \frac{1300-300}{3} = 333.33$€/year

In conclusion, for one year the IT tools will be amortized 333.33€, reduced to the ideal 161 hours of project duration, the amortization will be 6.13 €.

7.6 Management control

The Project Manager Office (PMO) will perform the management control for this project, like any other job in Ackcent. They will be the managers of the initiative and will take control of the expended hours, money costs, and future amortizations.

The regular workflow is:

- Kick-off meriting: all members related with the plan will meet, data will be shared, from the tasks to the budget. After this session, the project is officially started.
- Time imputation: every day, each employee imputes time to the assigned task, and the PMO verifies the work and the dedication for every proposal.
- Expenses verification: periodically PMO checks the used time and budget assigned. If something does not work as was planned an interview will be made with the employees and discuss the fluctuations.
- Project close: after finishing the job and presenting all the documentation the assignment will be closed and launched to production.

8 Sustainability

After taking part in the EDINSOST survey I have realized that I knew many of the points, and I figured out many other strategies and concepts to improve sustainability.

8.1 Cost reduction

The cost will be decreased to the level that will only take from two to four hours of work of one employee. The used electric energy will be reduced about nine times, so the expenses in terms of supplies will be much lower. In the environmental impact social improvements, I will reflect the consequences of that reduction to the respective fields.

8.2 Useful life

The useful life for this software is as large as the company decides. It is fully upgradeable with new features and controls. The simple modification and patching capabilities of the code allow us to adapt the solution to future updates and new features with low effort and maintain the solution for a long period.

8.3 Environmental impact

By performing the proposed changes, the power consumption of the audit infrastructure have minimized by switching from a server that was permanently running, not only for these audits, it has more projects that will be migrated to serverless solutions, and decreasing the employees the time and energy.

In addition to the reduction of energy consumption. In Spain, 41.72% of the energy was made from burning combustibles in 2019, as INE[24] reports [7]. This reduction of energy, applied on the biggest scale, can have a great impact on the greenhouse effect.

[24] Instituto Nacional de Estadistica: Spanish national statistics center.

9 Social dimension

On the other hand, the actual project is private, and it will not be open to everyone due to internal policies. The best way to improve society in computer science, in my opinion, is through open-source solutions. Respecting and encouraging the ten open-source rules:

- Free Redistribution.
- Source Code.
- Derived Works.
- Integrity of The Author's Source Code.
- No Discrimination Against Persons or Groups.
- No Discrimination Against Fields of Endeavor.
- Distribution of License.
- License Must Not Be Specific to a Product.
- License Must Not Restrict Other Software.
- License Must Be Technology-Neutral [8].

9.1 Personal growth

To me, this project means applying meanly all the knowledge acquired during the two and a half years working in Ackcent and a great part of the knowledge gotten from the university. It also opened many doors about software and solutions which I had not heard about.

9.2 Social improvements

It can imply that some of the established goals will be harder to reach. Despite the private facet of the solution, in Ackcent we are faced to improve the security for the customer.

We have seen many security gaps, many of them affecting the end-user by exposing their private lives. We want to offer reliable and fast responses to incidents, protect infrastructures from threats, and help with the security compliance of data owners.

9.3 Real needs

For the actual project, the need we have detected is the number of infrastructures exposed to the internet with misconfigurations that can become a gap where an attacker can get into and leaks information.

By giving the agility needed to this kind of audits we can ensure a quick and effective response to our customers. Helping them to prevent breaches, secure the products, and become a more privacy-friendly company.

10 Black Hat Server migration

The first task is orientated on the dynamic audit service migration to a server-less solution.

10.1 Original Platform

The first implementation for the email testing service was a bundle of scripts made with PHP[25]. The function of the scripts was to send some emails with spam and fishing content and some other emails with attachments, some of them being goodware and others being malware, in both cases the executables were packaged in different ways, like zipped, zipped with a password, zipped multiple times, embedded in a macro, etc.

After the summer of 2020 where I took the majority of the audits, I translated and improved the PHP scripts to bash scripts. This was made because some of the email servers were changed and nobody knew about PHP in the Security Architect department. The easiest solution was to translate and try to simplify the scripts to some bash scripts to automate the sends using swaks[26].

10.2 Study of requirements

10.2.1 Network requirements

I have to take care of two kinds of connection, the first one is the user interaction. The user can connect to the server using ssh. The network requirements for the ssh connection are really simple, you only need to make the server reachable for the client, open port 22, and have an ssh server. It will be discussed later if an ssh communication will be needed.

The second consideration will be the behavior of the scripts and how SWAKS and SMTP works. The first network operation is to access translate the MX server to an IP

[25] PHP: general-purpose scripting-language focused to web development

[26] SWAKS (Swiss Army Knife for SMTP): program for SMTP (Simple Mail Transfer Protocol) testing

address, this is made using the DNS (Domain Name System) protocol. After knowing where it has to send the information the program builds an email, SWAKS sends the email to the MX server using SMTP.

The internet speed is not crucial. The average email with text is no bigger than a few kilobytes and the test with attachments does not suppose a high band-with stress, expert for one case, the “big attachment file test”. This test is made to know if large files can be sent by email, regularly this send fails because the exchange server from the customer has a lower threshold than the size of the file. Size now I did not see any infrastructure with a threshold that allowed us to send the “big file”.

The summary of network requirements will be ssh server, if it is needed, resolving addresses with DNS protocol, and scenting email using SMTP protocol. The network requirements for the scripts are not high demanding and do not suppose any relevant obstacle for the migration.

10.2.2 System requirements

Our system must run a distribution of GNU/Linux. For our intention we do not need any graphical interface, the only need is to run the bash scripts. It must also be a lightweight distribution to a fast boot and low hardware requirements. Git is also will need for the version control and synchronization.

10.3 Choosing a platform

The first requirement, exposed at the introduction, was to have our platform out of the cloud and do not use a server. Because it uses more energy and resources that are not mandatory for our purpose. So the first decision was to implement the new solution using a virtualization platform or containers. Finally, I decided to use a virtualization platform because I have more knowledge and experience in that than with containers.

I decided to not use the physical machine due to the different operating systems we can install on our computer. Although SWAKS is available for Mac, GNU/Linux, BSD, and Windows, the installation methods are really different between the operating systems, and bash is not natively supported in Windows.

10.4 Deploy the solution

For our solution I will use the following software:

- Vagrant
- Virtual Box
- Ubuntu 21.10
- SWAKS
- Git
- Bash
- SSH

I defined a virtual machine using Vagrant and chose Virtual Box as the manager. The VM has specified the official Ubuntu 21.20 image as a base for our system. It has the basic hardware and network specification, one gigabyte of ram, one assigned CPU core, and a full internet connection.

The software specifications are guaranteed by the base Ubuntu image, it contains bash and ssh by default and vagrant launch commands. I had specified to update the system and install git and SWAKS in the vagrant file. With these commands, I can ensure every time every member of the team will have the latest version of the software and will not have out-of-date packages.

The distribution of the scripts will be also done using launch commands and git. I uploaded the receipts to the internal GitLab repository, only accessible if you are connected to the internal VPN. Vagrant will clone the GitLab repository and place it in the user directory.

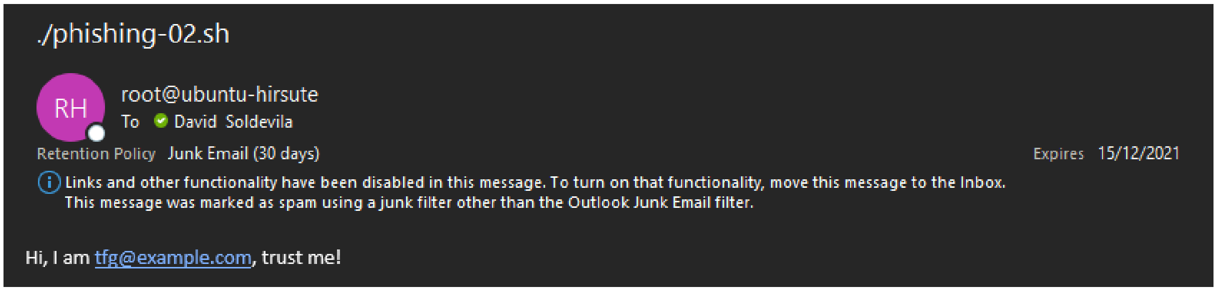
10.5 Testing the implementation

To test the new implementation I performed a bundle of execution from the new implementation and the previous server. If the results were equal and the new platform does not report any error the migration would be successful.

After checking the output from every send and observing how some of the emails were received I concluded the migration was successful. As you can see at the following output, email can be sent and these emails are received in the user inbox.

```
root@ubuntu-hirsute:/home/vagrant/O365DynamicTests# ./phishing-02.sh
### Phishing02 ###
=== Trying ackcent-com.mail.protection.outlook.com:25...
=== Connected to ackcent-com.mail.protection.outlook.com.
<-  220 -------------- Microsoft ESMTP MAIL
    Service ready at Mon, 15 Nov 2021 08:44:35 +0000
 -> EHLO ubuntu-hirsute
<-  250--------------- Hello [34.242.186.64]
<-  250-SIZE 157286400
<-  250-PIPELINING
<-  250-DSN
<-  250-ENHANCEDSTATUSCODES
<-  250-STARTTLS
<-  250-8BITMIME
<-  250-BINARYMIME
<-  250-CHUNKING
<-  250 SMTPUTF8
 -> MAIL FROM:<root@ubuntu-hirsute>
<-  250 2.1.0 Sender OK
 -> RCPT TO:<dsoldevila@ackcent.com>
<-  250 2.1.5 Recipient OK
 -> DATA
<-  354 Start mail input; end with <CRLF>.<CRLF>
 -> Date: Mon, 15 Nov 2021 08:44:34 +0000
 -> To: dsoldevila@ackcent.com
 -> From: root@ubuntu-hirsute
 -> Subject: ./phishing-02.sh
 -> Message-Id: <------------------------------------>
 -> X-Mailer: swaks v20201014.0 jetmore.org/john/code/swaks/
 ->
 -> Hi, I am tfg@example.com, trust me!
 ->
 ->
 -> .
<-  250 2.6.0 <------------------------------------>
    [InternalId=68238440419617,
    Hostname=---------------------------------------]
    8105 bytes in 0.116, 67.997 KB/sec Queued mail for delivery
 -> QUIT
<-  221 2.0.0 Service closing transmission channel
=== Connection closed with remote host.
```

Figure 5: Output for one of the testing script
Own compilation



./phishing-02.sh

root@ubuntu-hirsute
To David Soldevila
Retention Policy Junk Email (30 days) Expires 15/12/2021

Links and other functionality have been disabled in this message. To turn on that functionality, move this message to the Inbox.
This message was marked as spam using a junk filter other than the Outlook Junk Email filter.

Hi, I am tfg@example.com, trust me!

Figure 6: Phishing email sent with the testing tool
Own compilation

10.6 Aduit docuemnt

This document had no changes from the original version. However, I will explain the document and how the explained test will be explained.

It will contain an abstract of the tests explaining which kind of emails will be sent. This part is standard for every audit.

The second section is the security checks made with the results from the tests. First of all, the MX servers for the naked domain and will be identified as long as the MX servers for the [domain].onmicrosoft.com.

After this first check, four tables are presented with the email tests identified by a description and classified as “Received”, “Received as junk” or “Not received”. Each table has information about a different kind of test, the first one is for spam, the second one is for phishing and finally, the third and fourth tables are for the emails with attachments, one for the non-malicious attachments, and the last one for the malicious files.

If there are more than one MX servers we will add a column to every table for each exchange server.

Finally, we annex evidences for every failed test. An evidence consists of a screenshot of the received email.

Additionally, if an extra investigation is needed we are going to annex all the information after the first annex.

10.7 Conclusions

I concluded the migration with a solution that does not imply external infrastructure, does not use resources that are not mandatory, and, in addition, it contributes to a more secure and reliable way of sending test emails for our audits.

I also could deploy our new solution by using only free and open-source software. Keeping the software expenses at zero and decreasing the infrastructure cost due to the server-less solution.

The result report for the dynamic part of the audit will not be implemented using an automation script. I considered that it will not be worth spending time implementing a system that connects to a mailbox, queries emails, get the wanted, check if the email has been accepted as regular mail, is received to the junk folder, or is not accepted. In addition, the dynamic email tests controls must have a screenshot to demonstrate the status. Having in mind the tasks to do for the report automation I do not consider implementing the document automation.

11 Control update

One of the worries for the team was to not cover a good amount of configuration while auditing. We were taking care of the most crucial controls from the Azure mailing infrastructure, however, some of the less relevant controls were not being checked and, even if there are not as relevant as the already considered configurations, it can let an open attack vector or cause a data leak.

In order to upgrade our audits, I decided to check and improve or add controls to our delivery report. But first of all, I have to check what we have and what we are missing.

11.1 Control review

The second task to do is review the implemented controls, detect the shortcomings, and consult other sources to implement the deepest and usable audit checks.

The first audit developed years ago, did not accomplish our expectations of completeness for a while. As a team, we created a task to improve it. To identify the shortcoming I have to know what are we are checking and then, knowing what we want to control, review other solutions and knowledge bases to get a new list of controls to check the entire security configuration.

11.1.1 Check identification

The audit was divided in three sections:

- Status of published email management infrastructure.

  In this section, I checked the status of the published DNS records, MX records, SPF records[27], DKIM [28], and DMARC [29].

- Security status of Microsoft 365 controls and protection mechanisms.

  In the actual section I had included controls from multi-factor authentication, user permissions, Azure AD Identity Protection [30], password policy, authentication policy, applications policies, and Lockbox [31] configuration.

- Office 365 security controls and protection mechanisms.

  I included Connection filter configuration, SPAM filter configuration, Office 365 ATP [32] configuration, audit log configuration, forwarding rules, and Exchange Online policies for mobile devices.

[27] Sender Policy Framework (SPF): SPF identifies which mail servers are allowed to send mail on your behalf [9].

[28] DomainKeys Identified Mail (DKIM): DKIM lets you add a digital signature to outbound email messages in the message header. When you configure DKIM, you authorize your domain to associate, or sign, its name to an email message using cryptographic authentication [10]

[29] Domain-based Message Authentication, Reporting, and Conformance (DMARC): is an email authentication, policy, and reporting protocol. It builds on the widely deployed SPF and DKIM protocols, adding linkage to the author (‘From") domain name, published policies for recipient handling of authentication failures, and reporting from receivers to senders, to improve and monitor the protection of the domain from fraudulent email [11].

[30] Azure AD Identity Protection: Identity Protection uses the learnings Microsoft has acquired from their position in organizations with Azure AD, the consumer space with Microsoft Accounts, and in gaming with Xbox to protect your users. Microsoft analyses 6.5 trillion signals per day to identify and protect customers from threats. [12]

[31] Lockbox: Most operations, support, and troubleshooting performed by Microsoft personnel and sub-processors do not require access to customer data. In those rare circumstances where such access is required, Customer Lockbox for Microsoft Azure provides an interface for customers to review and approve or reject customer data access requests. It is used in cases where a Microsoft engineer needs to access customer data, whether in response to a customer-initiated support ticket or a problem identified by Microsoft. [13]

[32] Office 365 Advanced Threat Protection (APT): Microsoft Defender for Office 365 is a cloud-based email filtering service that helps protect your organization against advanced threats to email and collaboration tools, like phishing, business email compromise, and malware attacks. Defender for Office 365 also provides investigation, hunting, and remediation capabilities to help security teams efficiently identify, prioritize, investigate, and respond to threats [14].

11.1.2 Shortcomings

I have detected some missing controls based on the experience obtained and knowing the scope of the audit.

First of all, I consider that we did not get the entire Office 365 ATP configuration. It is capable of scanning links and documents. There are more other detecting engines, however, I considered the rest of the features out of the scope for our audit. You can check the entire list of features at `https://docs.microsoft.com/en-us/office365/servicedescriptions/office-365-advanced-threat-protection-service-description`.

Since now the security team did not consider SharePoint in the scope of the audit. However, I consider the file sharing service must be included in the audit. It is one of the main ways to share information and it is included on the basic pack of the Office 365 service pack.

The mobile devices policy must be updated and I think providing a base template to establish a correct policy could help our customer to maintain secure their infrastructure.

11.2 CIS benchmark review

The last revision of the benchmark consists in eighty-six different controls that check the entire Microsoft 365 service. Some of them are included in the controls checked for us in the previous control list. However, some of them are not being audited and are interesting to review and will be included in the list.

The utilized benchmark revision will be CIS Microsoft 365 Foundations Benchmark v1.3.0 available on `downloads.cisecurity.org`.

11.2.1 Account and authentication controls

This part of the audit was fully implemented and no additional controls will be incorporated.

11.2.2 Application permissions

This part of the audit was partially implemented, the following controls will be implemented among the previous controls:

- Ensure O365 ATP SafeLinks for Office Applications is Enabled.
- Ensure Office 365 SharePoint infected files are disallowed for download.
- Ensure users installing Outlook add-ins is not allowed

11.2.3 Data management

This part of the audit was fully implemented and no additional controls will be incorporated.

11.2.4 Email and Exchange online security

The proposed controls from the CIS benchmark that are not being checked at the first audit report are related to the ATP module. However, I considered adding them to the list. It is possible to do not have ATP services available depending on your subscription plan.

The included controls will be:

- Ensure the Advanced Threat Protection Safe Links policy is enabled.

11.2.5 Auditing

This section was fully implemented at the first version of the audit excepting the controls that check if the administrators of the infrastructure review the generated logs.

I will include the controls related to the log reviewing. However, I considered having an automatic log reviewing platform, a SIEM, with alerts and continuous monitoring is better to review periodically the majority of the controls. Our punctuation will be, from worst to better, not reviewing, hand reviewing, and automated review.

11.2.6 Storage

No additional storage controls will be implemented.

11.2.7 Mobile device Management

Reviewing the controls listed by the CIS does not apply to our audit scope.

The implemented control related to the actual section will be to check the Mobile Device Management (MDM) with our template policy.

11.3 Microsoft Recommendations

In this new version, I will take care of the Microsoft recommendations from Microsoft Secure Score[33]. The information given by this resource will be used to complete the document and give to the customer an initial point to start working on his security.

Although the information from the secure score was used in the first version of the audit, it was used as an appendix. Just used as a reminder that the customer IT team can use that tool to review the automatic security recommendations from Microsoft and the data was not used to perform any recommendation or to improve the security review.

[33] Microsoft Secure Score: is a measurement of an organization's security posture, with a higher number indicating more improvement actions taken. It can be found at `https://security.microsoft.com/securescore` in the Microsoft 365 Defender portal[15]

11.4 Control list update

The final control list will contain 45 controls. In a comparison of the 11 points that were applied at the previous version of the audit.

However, it is difficult to compare the new format and the old format because of the structure of the document. Before the actual revision, the structure of the audit was more client-focused, making it more customized for each infrastructure. Some of the controls were ignored if the customer subscription includes the audited service or not. For example, the free and the included version for Office 365 of Azure AD, which don't have Identity Protection, this control was excluded or marked as "Not Applies". However, the new version will be marked as it has to be improved, and recommended to upgrade the Active Directory subscription to protect the user's accounts.

11.5 Conclusions

With the control update performed comparing our previous controls with the CIS benchmark and Microsoft recommendations I got a new list of controls that are more granular, easy, and fast to check, and will provide a more widespread snapshot of the tenant configuration.

In addition, for the next section, this new control list will be more convenient because of the simplest way of checking it. I tried to reduce every control to a binary answer.

12 Script development

Finally, I am ready to implement a script to run the controls, write the output configuration, and add the description headers and the mitigation for every control specified in the list.

The main strategy to make this program possible will be to reduce the controls to a binary check. Sometimes this simplification will not be possible. For example, checking the MX DNS records will not be automated and will be made by hand. In fact, this configuration is usually discussed with the customer to know if it is configured as it was intended. However, the grand part of the controls will be reduced to compare the output to expected value, and depending on the result of the check will be positive or negative.

Using this method the "mitigations and recommendations" will be easies to implement. The average mitigation and recommendations will be a text exposing the wanted configuration and explaining the reasons for it.

This process is expected to generate a markdown file that will be reviewed by a security architect, completed and corrected, if any error is detected, and compiled into a PDF using any compilation tool like Visual Studio Code extensions, for example Markdown PDF, or websites, such as `www.markdowntopdf.com`, installed software, or any other solution preferred for the user.

The action made to implement this part will be:

- Reduce the controls to binary checks.
- Implement the comparison code section.
- Implement the "description" and "mitigations and recomendations" sections.

12.1 Style of the docuemnt

To make a presentable document I needed a way of specifying a PDF document that was simple, easy to understand, and easy to implement. I will have to be able to show multiple levels of titles, have the feature of monospaced text boxes, display images, draw tables, and, most important, must be able to be exported to a PDF file.

I thought of three possible options, HTML[34], LaTeX[35], and MarkDown[36]. Those three options have the document element needed for this document, can be exported to a PDF file, however, HTML has no native support to be exported as a PDF file, and are relatively simple to write.

The first I ruled out was HTML tables and style are not as simpler as I need to implement the automation simply. If HTML is used I would have to develop a CSS style and tables will require many lines and more invested time on it. In addition, MarkDown is compatible with HTML. Markdown documents can handle inline HTML sections and CSS style modifications and almost every static HTML document can be used with markdown.

Finally, I decided to write the raw document with MarkDown. An advantage, it has the simplest syntax. Without knowledge of it, you can understand how it works in a few minutes. The tables are clear to specify and the default style is acceptable and it can be customized to match the internal style rules with simple CSS modifications. Moreover, the PDF export for MarkDown is easier and has fewer software requirements than LaTex.

Options that involve PowerShell PDF modules or other languages were initially not

[34] HyperText Markup Language (HTML): is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript [16].

[35] LaTeX: is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. LaTeX is the de facto standard for the communication and publication of scientific documents [17].

[36] Markdown: is a lightweight markup language that you can use to add formatting elements to plaintext text documents. Created by John Gruber in 2004, Markdown is now one of the world's most popular markup languages [18].

contemplated because I have no bits of knowledge about them. I would have to invest time and effort to learn about a technology that is will give me similar or exact results that the known solutions.

12.2 Develop the start-up and connection module

This part of the program will load the needed PowerShell modules and connect our shell to the azure cloud. In order to implement this part, I will reuse the code made for the previous implementation of the audit, the same commands are also reliable for this part. However, some of the new controls need other modules and additional connection requirements.

12.2.1 Exchange online connection

To connect to the exchange online session, I will use the ExchangeOnlineManagement module. The Exchange Online PowerShell V2 module (abbreviated as the EXO V2 module) uses modern authentication and works with multi-factor authentication (MFA) for connecting to all Exchange-related PowerShell environments in Microsoft 365: Exchange Online PowerShell, Security and Compliance PowerShell, and standalone Exchange Online Protection (EOP) PowerShell [19].

The requirements for the user that will audit Exchange Online will be to have Global Reader permissions because the organization configuration is only visible for global users.

This part will be reused from the previous implementation.

```
Import-Module ExchangeOnlineManagement
Import-Module MSOnline
Connect-ExchangeOnline
Connect-MSOLSercie
```

Figure 7: Authentication for Exchange online
Sources: https://docs.microsoft.com/en-us/powershell/module/exchange/connect-exchangeonline?view=exchange-ps and https://docs.microsoft.com/en-us/powershell/module/msonline/connect-msolservice?view=azureadps-1.0

12.2.2 SharePoint connection

In order to connect to the SharePoint Online service, the SharePointOnlinePowerShell module will be used. The SharePoint Online Management Shell is a Windows PowerShell module that you can use to manage SharePoint settings at the organization level and site collection level [20].

It is possible that infrastructure does not have this service. In this case, the output for the controls that will depend on this service will be erased.

The requirements for the user that will audit the SharePoint Online service will be to have SharePoint reader permissions, a regular Global Reader user will be not able to read the SharePoint Online configuration.

This connection will be implemented as it is specified in the official Microsoft documentation.

```
Import-Module SharePointOnlinePowerShell
Connect-SPOService -Url $SPOurl
```

Figure 8: Authentication for SharePoint online
Source: https://docs.microsoft.com/en-us/powershell/module/sharepoint-online/connect-sposervice?view=sharepoint-ps

12.3 Control structure

I have implemented all the controls using three kinds of templates. One is used for the controls that use information from a command and automatically write the configuration and the remediation and configurations, another type for the manual controls, which must be analyzed by a human, and the controls that will be analyzed from a customer answer, this kind of controls will be named as “quiz controls”.

12.3.1 General structure

The general structure of the code for each control is the following:

- Title: A short string that identifies the control
- Description: Text that describes the scope of the control, the possible affectations of the configuration.
- Command or question reviewed: It outputs the command used or the question that is being used to perform the control review.
- Status of the command: Output of the command or question-answer that is being used to perform the control review.

12.3.2 Automatic controls

The automatic controls have some extra code lines. The automatic recommendations and mitigations writing needs extra code. It must have code to parse the command output and instructions to perform the output evaluation.

The parse code will have the following format:

```
$res1 = [command] | Select-Object -ExpandProperty [property]
$res2 = [command] | Select-Object -ExpandProperty [property]
...
.\check\[control_id].ps1 $res1 $res2 ...
        | Out-File output.md -Append -Froce
```

Figure 9: Parsing code for an automatic control.
Own compilation

It will be as much as necessary "res" input variables for each control. The average control requires only one input, however, some of them need more of them, because it will take considerations of different points in a configuration state.

The control evalutaion code will have the following foramt:

```
$Success1 = [Text]
$Success2 = [Text]
...
$Fail1 = [Text]
$Fail2 = [Text]
...
If ([logical comparison]) {
    Write-Ouput $Success1
}
Else{
    Write-Output $Fail1
}
...
```

Figure 10: Control evaluation for an automatic control.
Own compilation

This piece of code will perform logical comparisons of the inputs and the expected configurations. Depending on the comparison results, the script will write the message of success or failure depending on the input parameter. Each script can have more than one set of comparisons, depending on the control objective and how it is implemented.

12.3.3 Manual controls

While implementing, I have encountered that one of the controls can only be implemented consulting the output of a command by a human, concretely DNS MX records. Since the DNS configuration has not an standard good configuration.

Knowing the possibility of having some extra intermediate protection solutions, like spam filters, malware protection, etc. I have decided to not implement an output parse and automatic control evaluation.

12.3.4 Quiz controls

- Architecture Quiz: This file has five controls that need to be reviewed at a specific section of the Azure portal. The file has a control description, a link to the configuration web page, and a field to fill with the control state.

- Customer Quiz: This file has 12 controls related to the reviewing of the Office 365 logs and reports. The customer must answer yes or no if they review the information at least weekly or it is automated, and they can add a comment.

- Microsoft Secure Score: Although it is not a quiz the format is very similar. Each of the controls is implemented by the Secure Score. Each one has an identifier, a control description, and a status, among other information that is not relevant for our audits.

The code that implements the parse of this kind of controls is:

```
$res = $[file] | Where-Object 'id' -eq '1'
    | Select-Object -Property 'Status'
```

Figure 11: Control parse for a quiz control.
Own compilation

The evaluation code has the same format as automatic control.

12.4 Testing

In order to test the controls implemented, I ran a series of tests checking to test the command execution, the logical test for the command output, and the markdown writing.

12.4.1 Command testing

The test for this part was limited because I could only test using the Ackcent test account. In addition, this is also limited because I could only have read access at the security configuration and this configuration cannot be changed because the testing infrastructure is being used to test other projects.

The tests for this part consisted in run the command and ensuring the command reports the information with the wanted format. The markdown output had to be in the human-readable format and the output that have to be redirected to the input of the compression had to be the row output, without format enrichment.

12.4.2 Logical comparison testing

To test the logical comparison code I made two different types of tests. One using the output of the command as it was a regular execution and other sending the other possible outputs of the command were extracted from the Microsoft documentation available at `https://docs.microsoft.com/en-us/powershell/module/sharepoint-online/?view=`

sharepoint-ps, https://docs.microsoft.com/en-us/powershell/module/msonline/?view=azureadps-1.0, and https://docs.microsoft.com/en-us/powershell/module/sharepoint-online/?view=sharepoint-ps.

12.4.3 Markdown writing test

For the final test, I run the entire code for every control. The main objective for this test was to view the final results written in markdown. As the markdown syntax is really simple and it was specified inline there were no problems with the document creation.

12.5 Shortcomings

The technical development of the project has almost no shortcomings or problems. However, in the beginning, the permission assignment for the user that has been used to test the controls supposed some work to try to minimize the permissions assigned. The result was that the minimum permissions to review all the controls implemented are Global Reader. I would like to use more strict roles like Security Reader. The first role assignment was Security Reader and it caused some issues with a test that uses Get-OrganizationConfig. These cmdlets needs read permissions that are not included in the Security Reader role.

The most significant shortcoming was caused by a risk mentioned, the highest priority of other projects. In my case was an incident. A huge company was attacked and during some months I had to attend the securitization of the infrastructure and help to restore services and production. This workload added to the other customer projects ended with a delay for the tasks that caused to do not have the implementation for the administrative report for the delivery of the actual document.

Because of the lack of time I had no opportunity to implement the automatized executive report. To remediate this situation I will present a workaround that will consist of a manual guided way to write the report.

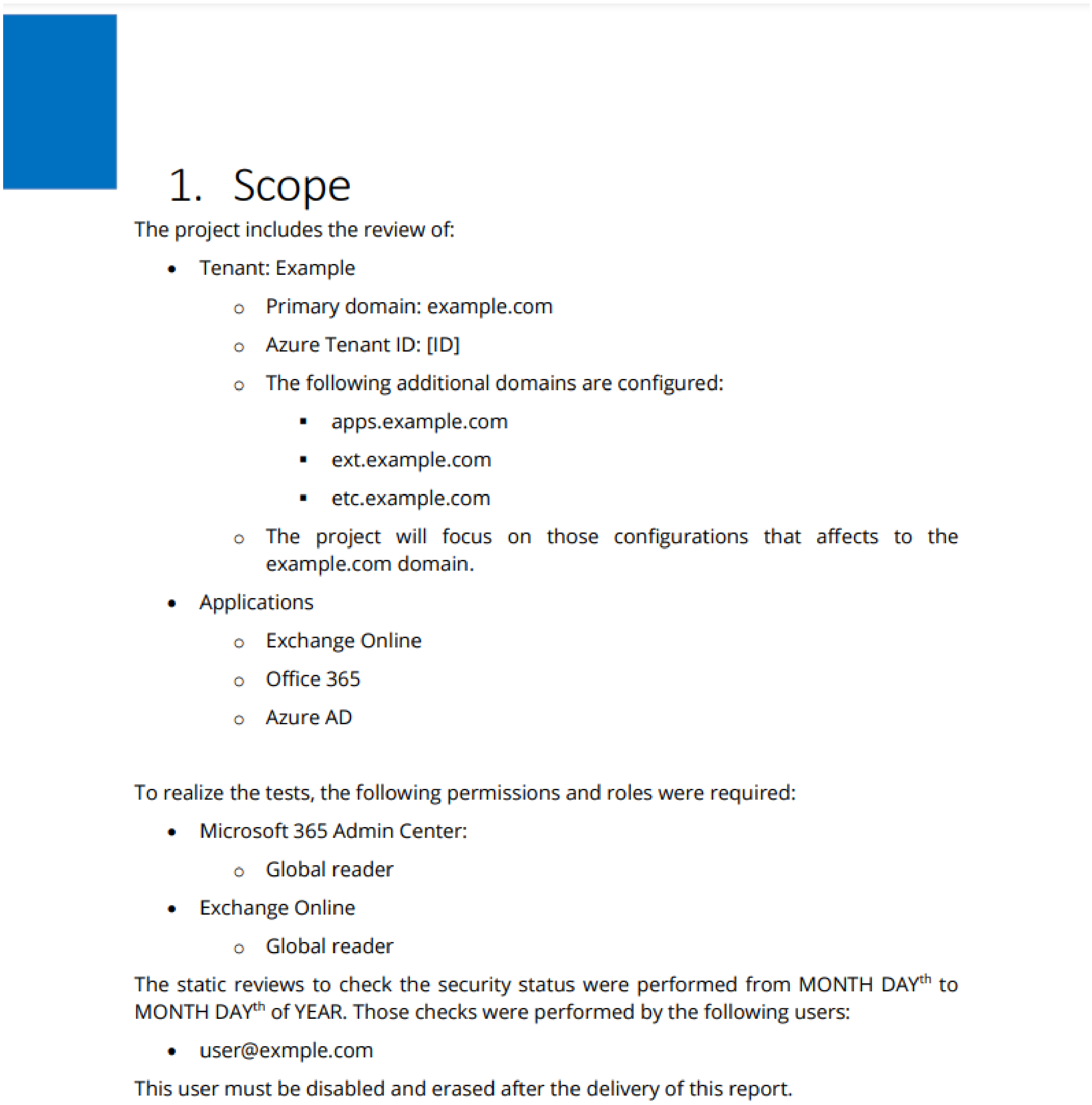
12.6 Executive report

The executive report is the part that will contain the relevant information, used to present the results with no technical details. It provides information in a clear and readable way. With plots representing the overall state of the controls, the main points to solve, and a custom text summarizing the actions performed, the scope, and the purpose of the audit. It is useful for the directives and to have a general view of the infrastructure status.

The scope will contain the information about the tenant that will be audited as:

- Tenant Name
- Primary domain
- Azure Tenant ID
- Additional domains
- Audited applications and services
- User used to audit
- Date and time of the audit performance

In the following image is the template for the scope:



1. Scope

The project includes the review of:

- Tenant: Example
  - Primary domain: example.com
  - Azure Tenant ID: [ID]
  - The following additional domains are configured:
    - apps.example.com
    - ext.example.com
    - etc.example.com
  - The project will focus on those configurations that affects to the example.com domain.
- Applications
  - Exchange Online
  - Office 365
  - Azure AD

To realize the tests, the following permissions and roles were required:

- Microsoft 365 Admin Center:
  - Global reader
- Exchange Online
  - Global reader

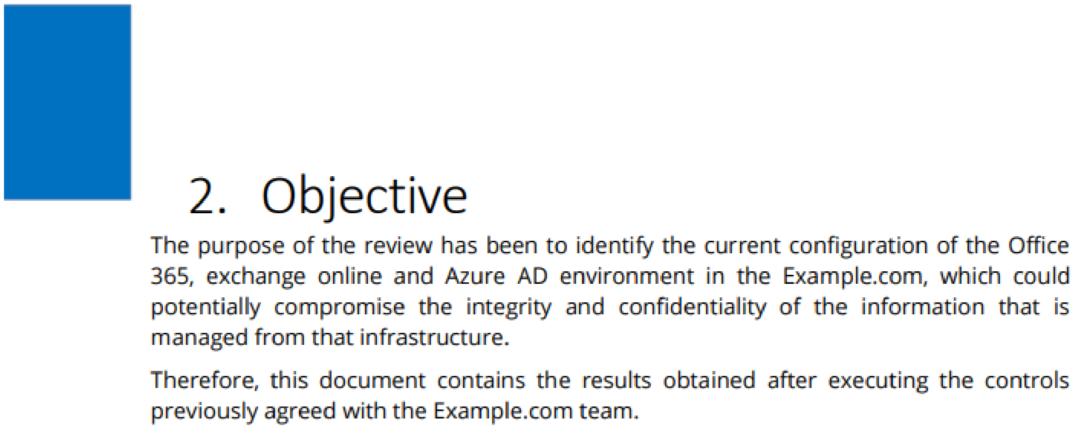
The static reviews to check the security status were performed from MONTH DAYth to MONTH DAYth of YEAR. Those checks were performed by the following users:

- user@exmple.com

This user must be disabled and erased after the delivery of this report.

Figure 12: Executive report scope section
Own compilation - Static control executive report

The objective section will define the objective of the docuemnt. In general, the objective of the audit is always the same. This is the objective section for our audits:



2. Objective

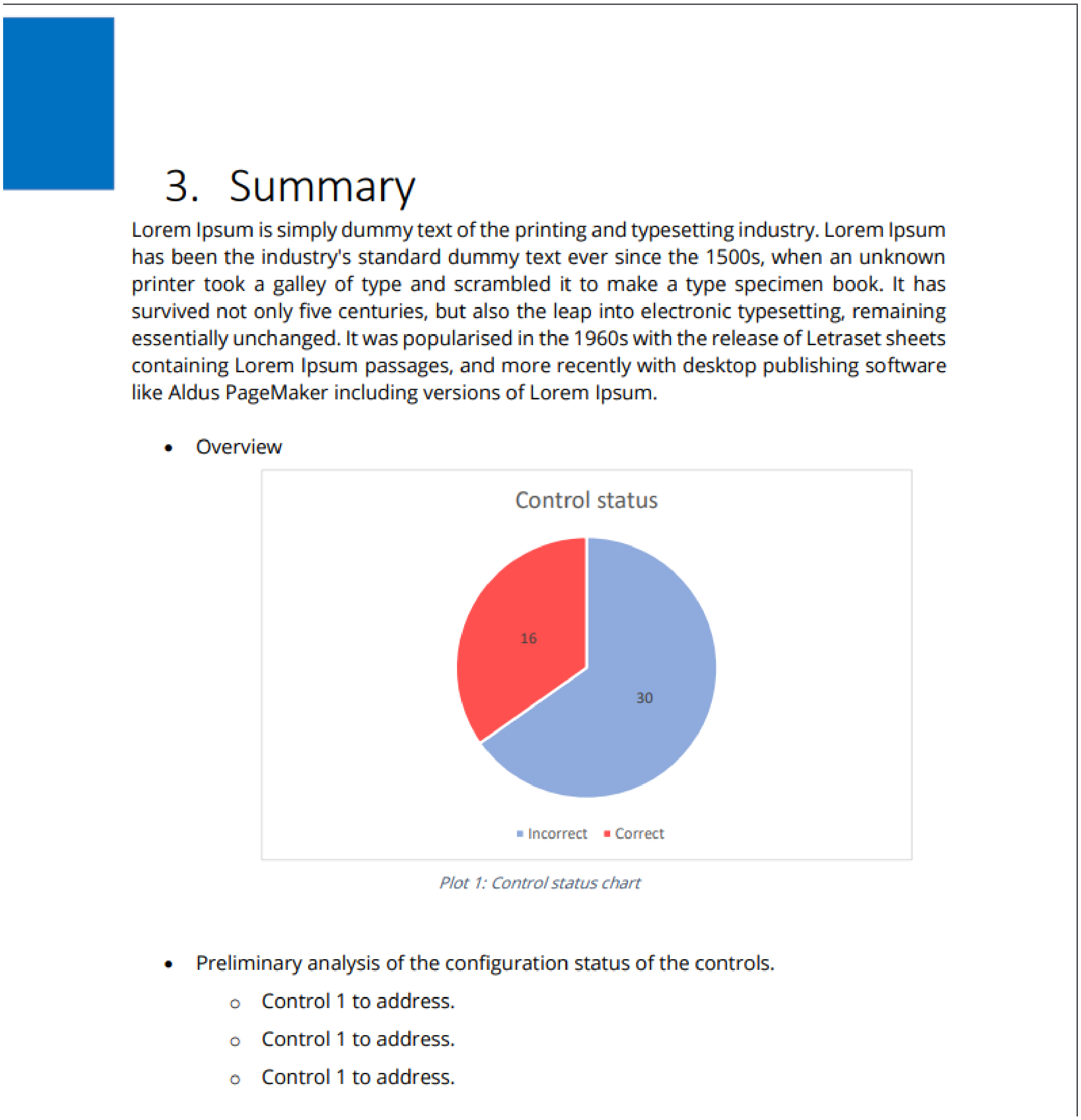
The purpose of the review has been to identify the current configuration of the Office 365, exchange online and Azure AD environment in the Example.com, which could potentially compromise the integrity and confidentiality of the information that is managed from that infrastructure.

Therefore, this document contains the results obtained after executing the controls previously agreed with the Example.com team.

Figure 13: Executive report objective section
Own compilation - Static control executive report

Finally, the document will have a summary section. The architect will write a few lines about the results of the audit and the most important failed controls and the mail recommendations to implement. Ideally, the summary will not have more than 10 lines. This last part will not be automated, so it is meant to be a custom and unique text for every customer. I will be written applying the knowledge of the infrastructure and the needs that the customer has and only the architect knows. This knowledge can come from other projects or conversations so it will be almost impossible to automatize.

It will also contain a pie chart with the passed and failed controls. It will give a visual view of the infrastructure status. The chart will be composed of the pie, an index for the passed and failed controls, and a legend to identify the information.



3. Summary

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

- Overview

Plot 1: Control status chart

- Preliminary analysis of the configuration status of the controls.
  - Control 1 to address.
  - Control 1 to address.
  - Control 1 to address.

Figure 14: Executive report summary section
Own compilation - Static control summary report

12.7 Time savings

With the program working and tested, I made an audit of Ackcent's test infrastructure. This test will be used to determine the effort reduction and it will be shown by the reduction of the hours used to elaborate the final document. However, it will not be a realistic test, the test Azure tenant is not a production environment and some of the services are disabled or configured with non-standard parameters. From users without MFA to services registrations out of the main Azure account or intended breaches that are known and used by the red team to test offensive tools.

After running the execution of the script it generated a markdown file with the results for the 45 controls. After reviewing the result, completing it with the necessary information, and generating by hand the executive report I ended with two documents, the complete generated report with all the controls and detailed information and the report with the explanation of the document, the plot condensing the state of the configuration and the conclusions.

The time used to make the audit was 5 hours, one more hour than the planned ideal estimation made at the beginning of the project. Nevertheless, it is the first time it is implemented, with no practice and no experience of how it was presented and how to elaborate the executive report. Compared with a load of the previous workflow, 35 hours at least, the actual way of auditing implies a time save of:

$Time\ reduction = \frac{T_{initial} - T_{final}}{T_{initial}} = \frac{35-5}{35} = 0,8571 \rightarrow 85,71\%$

With this method, we save at least 30 hours or 85,75% of the time used to audit an infrastructure.

12.8 Conclusions

Knowing the time we are going to save, the confidence of applying all the controls specified, with an standardized method, and ensuring the architects will not be focused to make a repetitive and monotone work will let us focus in more fine details I can conclude that the script does the intended work as fine as it was meant.

Of course, it has some inconveniences, it has to be reviewed manually by an expert to ensure the information and the recommendations were correctly processed and displayed. In one of the controls it has to be written by hand and the executive report was not automatized. These missing features implied more human dedication. Those missing features will be addressed in a future review of the script.

The plan for this project is to implement the missing features and extend them with new and updated controls if some change is made in Azure Active Directory, Office 365 of Exchange Online.

13 Final conclusions

13.1 Project conclusions

Since the beginning of the project, the objective was to make a reliable and fast way to perform an audit for Office 365, Exchange Online, and Azure AD. Those sets of projects take us about 35 hours of effort by manually reviewing the configuration and analyzing the results to make a complete report. However, the architecture team had detected some faults and lackings. In addition, we started an initiative for moving services hosted at AWS to server-less solutions.

This last part will also help reduce the ecologic impact of Ackcent's activity. Knowing that moving one service to a more efficient platform is not very significant, every saving is a step forward to a green approach of the industry. Furthermore, we expect to lessen the energy waste and compute time because of the dedication reduction.

Following this ideology, my first task was to migrate the infrastructure needed for the dynamic audit to an on-demand, server-less, and trustworthy platform. To accomplish it, I recreated the original virtual server with vagrant, virtual box, and git, acquiring a solution with few dependencies. The obtained framework could perform the same work with the advantage of having more than one instance running simultaneously with no additional cost, Independence of support and, smooth update process. The experience and the results I got with the new virtual machine environment were the same as the previous solution. The migration concluded as a great success.

The second project was to update the controls audited to the infrastructure and implement the previous and new controls using an automatic solution. The first step was to review the current control list identifying its lackings. After that assignment, I completed the control catalog with the recommendations from CIS and Microsoft. Customizing the new checks and adapting the original pool I ended with a draft of 45 controls to implement.

The last step of the audit improvement was to implement the previous rules. The implementation had the following requirements: checks had to be automatized, had to report a human-readable document, had to be updatable, and it had to be customizable. With all of these needs in mind, we studied some available solutions. Some of the software was open-source, others were products from networking companies. After analyzing, comparing, and debating with the entire architecture department the decision was to implement a custom script using PowerShell. The program should investigate all the controls I had specified. An automatized administrative report was planned to be implemented. I would have the basic information of the audit and a quick overview of it.

The result of the implementation was a script that connects to Microsoft account with global reader permissions, prints out the information needed to analyze the configuration and writes a markdown file with a brief description of the security control, the command used to get the information, the information reported by the command, the stat of the configuration, and, if needed, recommendations and remediations to secure the infrastructure.

While developing, I have defined three kinds of controls. Automatic, using a command output, semi-automatic using a CSV, and manual. However, the most reliable and optimal type of control is the fully automatic kind, some of the information is not accessible by the command line or just comes from the customer. The manual controls: in this strategy, as mentioned previously, only one particular check cannot be automated. Manual controls are the ones that cannot be determined with a binary output, and many configurations are reasonable.

The automatic administrative report has not been implemented due to setbacks. The last trimester of 2021 has been strongly affected by a huge campaign of ransomware attacks. I was directly involved with incident response. The workload to restore the infrastructure and securing the endpoints took me the entire workday (and extra hours) for two weeks, and after those two weeks I had to spend about 75% of the workday solving related issues, and the other 25% of the day was committed to other customer projects that needed to be finished before holidays. The unexpected workload on that project turned into a lack of time to implement the final stages of the automation script.

Nevertheless, to mitigate the missing part of the project I made a template with the sections and items needed. The administrative report template has a scope section, an objective advertisement, and an audit summary. Every part will be written by hand by the on-charged architect.

The overview of the entire project, despite the missing feature, is the update of the audit was a successful project. Ending with a new way to perform the mail sending for the dynamic audit and an automation script to perform the static audit.

13.2 Planning compression

The initial planning had the goal of completing the project before December. The last month of the year is usually busier because of the last time projects. A significant part of that kind of project is infrastructure audits. Nevertheless, before closing the project, and when only the final part job left, an incident affected a big customer. This stepback had an impact on the planning.

The objective was to end the project officially by the third week of November and operate with the new solution from December on. The actual planning is meant to implement the residual part for the second quarter of 2022.

The cost of the project has not changed. The number of hours assigned to the task is the same, plus the price per hour did not change.

13.3 Personal conclusions

From my personal experience, the update of the audit service was an opportunity to grow my expertise in cloud services and script development. My programing experience with PowerShell was nearly null. Previously, I was used to programming with Linux Shell scripts. By now, I know how PowerShell modules works and the syntax of the scripting language. In addition, I discovered the use of the PowerShell pipes and how they streams the data to the following command.

Aside from the security and system knowledge related to my regular job, it was the first project related to software development.

From the point of view of project management, I have acquired experience of how to plan the expenses for a project and administrating S.M.A.R.T. methodology.

References

[1] W. contributors. Dns. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Domain_Name_System&oldid=1046611458

[2] Microsoft, “What is powershell?” https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.2.

[3] Aqua. Cloud sploit open source project. [Online]. Available: https://cloudsploit.com/opensource

[4] Scrum.org. Scrum. [Online]. Available: https://www.scrum.org/resources/what-is-scrum

[5] Cisco. Vpn. [Online]. Available: https://www.cisco.com/c/en/us/products/security/vpn-endpoint-security-clients/what-is-vpn.html

[6] A. W. Services. Amazon web services ec2. [Online]. Available: https://aws.amazon.com/ec2/?ec2-whats-new.sort-by=item.additionalFields.postDateTime&ec2-whats-new.sort-order=desc

[7] E. y. N. e. M. d. E. S.G. de Prospectiva, “Estadística de la industira de la energía eléctrica,” https://energia.gob.es/balances/Publicaciones/ElectricasAnuales/ElectricasAnuales_20192021/2019/Resumen%20de%20datos%202019/Industria_energia_electrica_anual_2019.pdf.

[8] O. S. Initiative. The open source definition. [Online]. Available: https://opensource.org/osd

[9] Microsoft, “Set up spf to help prevent spoofing,” https://docs.microsoft.com/en-us/microsoft-365/security/office-365-security/

set-up-spf-in-office-365-to-help-prevent-spoofing?view=o365-worldwide#what-does-spf-email-authentication-actually-do.

[10] ——, “Use dkim to validate outbound email sent from your custom domain,” https://docs.microsoft.com/en-us/microsoft-365/security/office-365-security/use-dkim-to-validate-outbound-email?view=o365-worldwide#how-dkim-works-better-than-spf-alone-to-prevent-malicious-spoofing.

[11] DMARC.org, “What is dmarc?” https://dmarc.org/.

[12] Microsoft, “What is identity protection?” https://docs.microsoft.com/en-us/azure/active-directory/identity-protection/overview-identity-protection.

[13] ——, “Customer lockbox for microsoft azure,” https://docs.microsoft.com/en-us/azure/security/fundamentals/customer-lockbox-overview.

[14] ——, “Microsoft defender for office 365 service description,” https://docs.microsoft.com/en-us/office365/servicedescriptions/office-365-advanced-threat-protection-service-description.

[15] ——, “Microsoft secure score,” https://docs.microsoft.com/en-us/microsoft-365/security/defender/microsoft-secure-score?view=o365-worldwide.

[16] W. contributors, “Html,” https://en.wikipedia.org/wiki/HTML.

[17] L. project team, “Latex – a document preparation system,” https://www.latex-project.org/.

[18] M. Cone, “What is markdown?” https://www.markdownguide.org/getting-started/.

[19] Microsoft, “Exchange online powershell,” https://docs.microsoft.com/en-us/powershell/exchange/connect-to-exchange-online-powershell?view=exchange-ps.

[20] ——, “Intro to sharepoint online management shell,” https://docs.microsoft.com/en-us/powershell/sharepoint/sharepoint-online/introduction-sharepoint-online-management-shell?view=sharepoint-ps.