UNIVERSITY OF SANTIAGO DE COMPOSTELA


ESCOLA TÉCNICA SUPERIOR DE ENXEÑARÍA

# Improvements in IDS: adding functionality to Wazuh

*Autor:*
**Andrés Santiago Gómez Vidal**

*Directores:*
**Purificación Cariñena Amigo**
**Andrés Tarascó Acuña**

## Computer Engineering Degree

**February 2019**

Final degree project presented at the Escola Técnica Superior de Enxeñaría of the University of Santiago de Compostela to obtain the Degree in Computer Engineering


**Ms. Purificación Cariñena Amigo**, Professor Computing Science and Artificial Intelligence at the University of Santiago de Compostela and **Mr. Andrés Tarascó Acuña**, Managing Director at Tarlogic Security S.L.

STATE:

That the present report entitled *Improvements in IDS: adding functionality to Wazuh* written by **Andrés Santiago Gómez Vidal** in order to obtain the ECTS corresponding to the final degree project of the Computer Engineering degree was conducted under our direction in the department of Computer Science and Artificial Intelligence of the University of Santiago de Compostela.

For the purpose to be duly recorded, this document was signed in Santiago de Compostela on February TODO, 2019:

| The director, | The codirector, | The student, |
|---|---|---|
| (Purificación Cariñena Amigo) | (Andrés Tarascó Acuña) | (Andrés Santiago Gómez Vidal) |

# Index

# List of Figures

# List of Tables

# Chapter 1

# Introduction

This project was made in collaboration with the cybersecurity company Tarlogic SL, even though I am not a member of Tarlogic and have never worked with them in the past. It is key to note my lack of expierence in cybersecurity (on a professional level) because is the reason for the bad planning estimations and the limit of the scope.

## 1.1 Motivation

Cybersecurity nowadays is very complex and there are many subfields and expert tools, to the point it could be argued that is imposible to master. In this project we put ourselves in the shoes of an administrator of an enterprise system that wants to improve the security by detecting intrusions.

Cybersecurity measures can be applied in multiple layers of the system, each with different tools, objectives, advantages and cost. In general the security of a system has the next parts:

1. **Firewall**: Control the inbound/outbound connections, on the **network layer**. In our case its objective is to reduce the amount of inbound connections, reducing the chance of intrusion.

2. **IPS**: Intrusion Prevention System to minimize the chance of intrusions, on the **program layer**.

3. **IDS**: Intrusion Detection System to mitigate the damage of intrusions, on the **program layer**.

The next table shows a **simplified** flow on how the information is processed by the security layers and methods.

| **Layer** | Network | Program | |
|---|---|---|---|
| **Method** | Firewall | IPS | IDS |
| **Measures** | Prevent | Prevent | Mitigate |

Direction of the data flow $\longrightarrow$

We focus on IDS because it is less explored than IPS or Firewall and due to the advance in gathering and processing of data in the last years IDS has become much more viable and reliable.

IDSs are systems specialized in detection and are different from antivirus or antimalware because they usually focus on prevention, however prevention and detection are often meshed together because both are deeply related. There are some cases where a sytem specialized in detection offers some kind of mitigation functionality or a sytem specialized in prevention offers some kind of detection functionality.

It is important to note that in cybersecurity the tendency is for the attack to be created first and later some kind of measures, not necessarily by the same teams as they tend to be specialized in each rol. Nowadays there are lots of different attacks, so many that their detection could be almost imposible one by one, but most of them can be detected because they share patterns. If we can determine the patterns of an attack and code a way to detect them we can detect the threat.
IDS work by analysing the key information available (programs, logs, network, etc) to determine if there has been an intrusion in the system. The details of the process vary with each IDS but in general they work like an expert system:

- The source of the data is the system.
- The alerts are set by certain rules when they match.
- Rules do not need to throw an alert and there can be dependencies, allowing complex analysis without false positives.

OSSEC is an HIDS (Host-based IDS) solution with detection based on rules and decoders. Both rules and decoders can be defined with numerous options and support dependencies and regular expressions.

- The decoders format the data for the rules.
- The rules determine there is a threat if the conditions are met.

**OSSEC** stands for **O**pen **S**ource HIDS **SEC**urity and is interesting because the next qualities[1]:

- **Widely Used**: OSSEC is a growing project, with more than 5,000 downloads per month on average. It is being used by ISPs, universities, governments and even large corporate data centers as their main HIDS solution. In addition to being deployed as an HIDS, it is commonly used strictly as a log analysis tool, monitoring and analyzing firewalls, IDSs, web servers and authentication logs.

- **Scalable**: Because it is an HIDS and it uses agents.
- **Multi-platform**: GNU/Linux, Windows, Mac OS and Solaris. This is important because most professional services are on GNU/Linux or Windows, but it is important to note that rules can only work in one operating system.
- **Free**: OSSEC is a free software and will remain so in the future; you can redistribute it and/or modify it under the terms of the GNU General Public License (version 2) as published by the FSF – Free Software Foundation.
- **Open source**: The code is open, so you can read and debug it all you want.

Furthermore it offers functionality like[2]:

- **Rootkits detection**: This type of malware usually replaces or changes existing operating system components in order to alter the behavior of the system. Rootkits can hide other processes, files or network connections like itself.
- **File integrity monitoring**: To detect access or changes to sensitive data.
- **Agents**: Each monitored host can either install the agent or use an agentless agent[3] (but still needs to install OSSEC, so it is a bit pointless).

OSSEC is interesting for this project because it offers a reliable way to use an already done and thoroughly tested IDS and enhance it to our needs without much work. To even ease more this we will use Wazuh, a fork of OSSEC.

## 1.2 Objectives

The main objective is to improve intrusion detection in IDS. This can be accomplished in several ways:

- Adding or changing functionality of an already existing technologie.

  Coding on core or additions.

  Configuration or input of the program.
- Develop a new technologie or tools that result in a different detection system.

As mentioned before in this project we will use OSSEC through Wazuh to code rules and decoders, without the need to change any code of the program itself, which means this project can focus directly on detection without the need to create a detection system. Of course if in later stages of the project it would to be found that is convenient to modify the detection system itself it could be considered depending on the importance, the progress and the remaining time of the project.

## 1.3 Structure of this document

This document has TODO chapters:

- In **chapter 1**
- In **chapter 2**
- In **chapter 3**
- In **chapter 4**
- In **chapter 5**
- In **chapter 6**
- In **chapter 7**

# Chapter 2

# OSSEC and Wazuh

## 2.1 Introduction

Wazuh is different than base OSSEC in that it adds functionality (a RESTFul API and rules and decoders) and is easier to install (it uses the ELK stack to gather and preprocess data, while OSSEC leaves that to freedom of the user). Wazuh too is open source and free software.

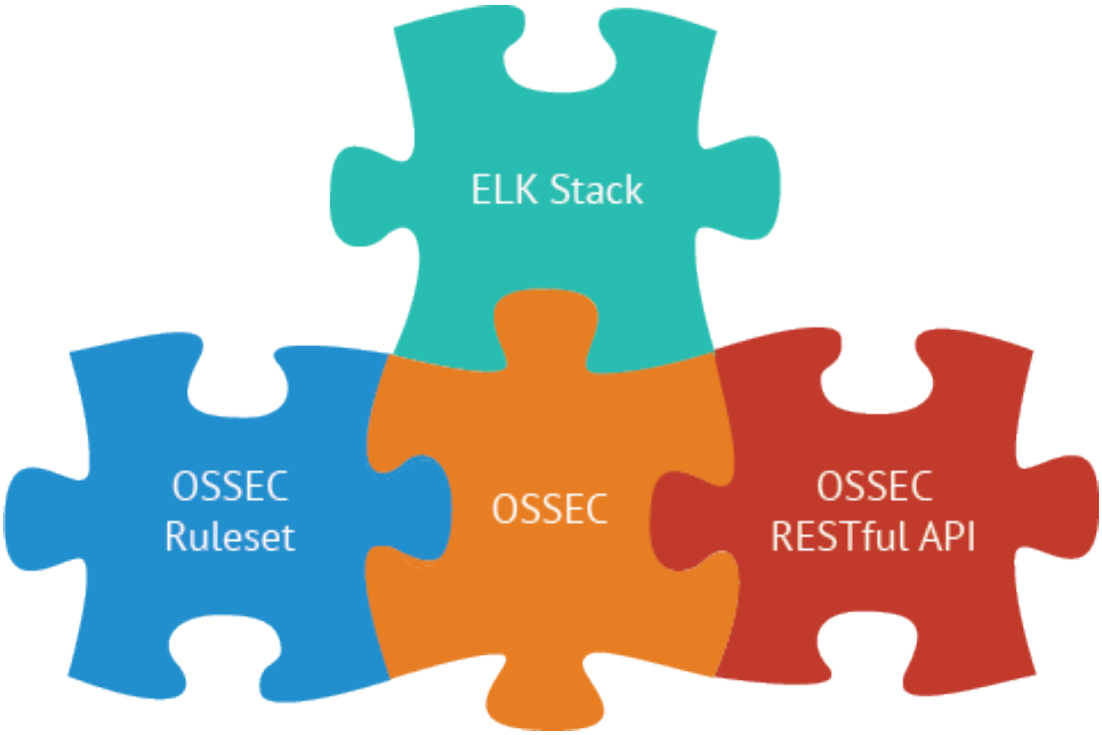



Figure 2.1: The different technologies that form Wazuh[4]

The most interesting qualities of Wazuh for this project are[5]:

- **Rootkits detection**: Rootkits are commonly used after an attack has suceeded to use the computer of the victim leaving no traces.
- **File integrity monitoring**: It can provide detection of intrusions and can be used to comply with GDPR (General Data Protection Regulation).
- **Scalability and multi-platform**: This means that the work on this project could really be used in real work environments.
- **Configuration management**: The configuration is managed by the Wazuh server (Wazuh manager) and the agents can be grouped, allowing custom, grupal or global gathering and detection for each agent.

- **Multiple sources of data**: The scanned data can be from logs, output of commands or databases.
- **Active response**: An automated remediation to security violations and threats, to mitigate more the posible damage. For example to stop the Internet connection to isolate a compromised system.
- **Improved ruleset**: This reduces the workload of this project, as it can serve as guidance and complement some of the rules and decoders that this project intends to create or modify.
- **Open source, free and easy to contribute to**: This is optional but nice, as it offers a chance to an unexperienced student to contribute in a real and useful project. The project is hosted on Github and Google Groups. In this project the contribution would be to the ruleset[6] and not to the core of Wazuh[7] or the documentation[8].

The RESTFul API interacts using OSSEC commands and would be interesting if this project were related to a tool issuing queries to Wazuh, but this is not the case. Anyway it is still something valuable to have as these kind of tools are very common nowadays.

## 2.2 Wazuh architecture

A Wazuh setup has the next components[9]:

- Wazuh server: Runs the Wazuh manager, API and Filebeat (Filebeat is only necessary in distributed architecture). It collects and analyzes data from deployed agents.
- ELK stack: It reads, parses, indexes, and stores alert data generated by the Wazuh server. The ELK stack is flexible, highly configurable and very used in big data.
- Wazuh agent: Runs on the monitored host, collecting system log and configuration data and detecting intrusions and anomalies. It talks with the Wazuh server to which it forwards collected data for further analysis.

The main difference with the architecture of OSSEC is the ELK stack, because OSSEC leaves the choice of tools for these functions to the user. ELK stands for the combination of:

- Elasticsearch: Gets the data and allows search queries and analysis.

- Logstash: Transforms the data to the desired format. This step can make alike data from different log and output formats, trivializing the decoders work.
- Kibana: Shows the data in a web browser, with graphs and options like grouping and time interval. This is often easier than to write commands to scan the OSSEC log in the Wazuh server, as the data of interest tends to stay the same.

There are two posible architectures for this setup: having the ELK stack in the same machine that the Wazuh server (singlehost) or in a separated one (distributed). Each has advantages and disadvantages and in this project we will use the singlehost because in our case it does not really matter and is easier to set up and more efficient.

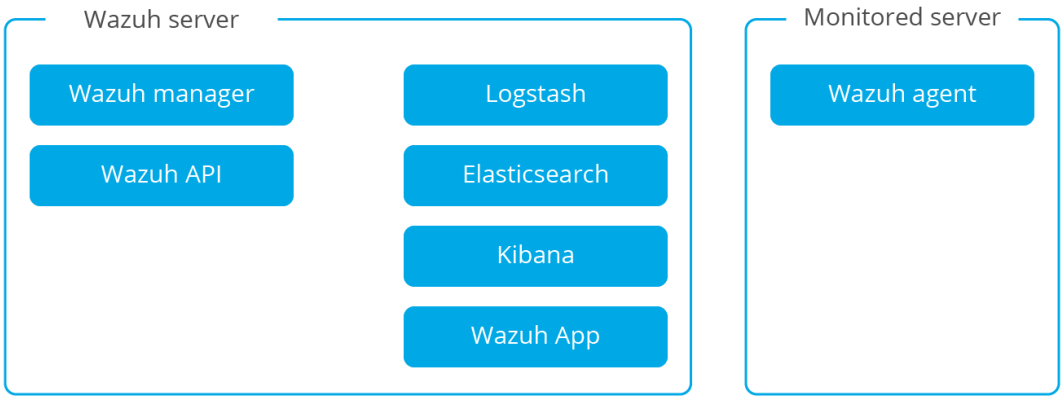



Figure 2.2: Singlehost architecture

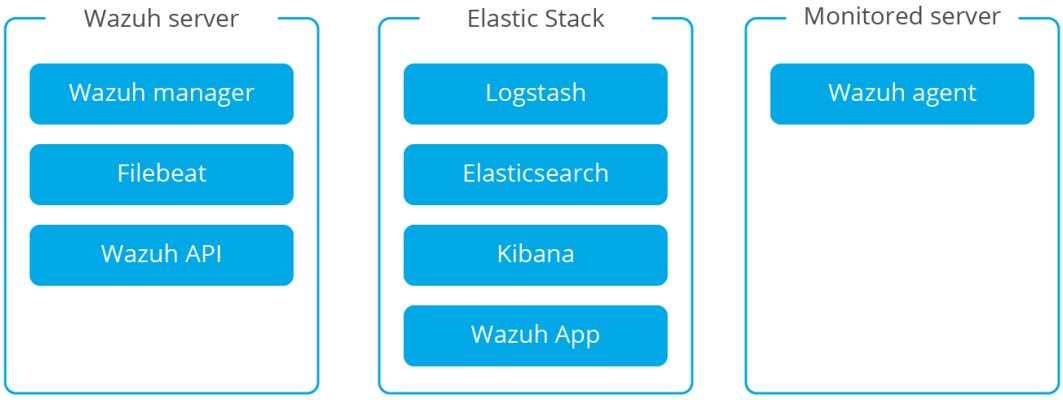



Figure 2.3: Distributed architecture

## 2.3 Rules and decoders

# Chapter 3

# Requirements

The requirement specification is a full description of the software the project is to develop.
PMBOK[10] states that requirements are conditions or capabilities that a product must meet to satisfy the contract. The requirements expose the needs of the client, which have to be accomplished to finish the project successfully. In this project the requirements will be fullfilled in multiple stages along the project.
Note that the client in this case is Tarlogic even if the product is a contribution to an open source project.

This specification contains:

- **Use cases**: Functionalities that the software will provide.

- **Requirements**: Depending of their type they can describe features, data, relations, properties or any details necessary to explain the system without ambiguity, in a way it can be easily understood.

In this project the functional requirements are not included because they can be considered a redundant version of the use cases, because both describe the same functionalities. Uses cases were chosen over functional requirements because they were considered to be easier to understand and have greater detail. If this project had the need of a very complex requirement specification it would be interesting to have both, as each could help to understand the other better, but in this project the specification should be quite simple.

## 3.1 Use cases

A use case is a description of all the ways an end-user wants to "use" a system. These "uses" are like requests of the system, and use cases describe what that system does in response to such requests. In other words, use cases describe the conversation between a system and its user(s), known as actors. Although the system is usually automated (such as an Order system), use cases also apply to equipment, devices, or business processes.[11]

### 3.1.1 Use cases actors

The actors are entities external to the system that interact with it. They can be other systems, persons or even time.

### 3.1.2 Use cases list

## 3.2 Requirements analysis

### 3.2.1 Non functional requirements

### 3.2.2 Functional requirements

As mentioned before these are omited because of the redundancy with use cases.

### 3.2.3 Domain requirements

# Chapter 4

# Project management

## 4.1 Risk management

### 4.1.1 Risk metrics

| Chances of the risk happening | Probability |
|---|---|
| ≥80% | High |
| Between 30% and 80% | Medium |
| ≤30% | Low |

Table 4.1: Probability classification of risks

| Resource in Place / Effort / Cost | Impact |
|---|---|
| ≥20% | High |
| Between 10% and 20% | Medium |
| ≤10% | Low |

Table 4.2: Impact classification of risks

| **Exposition** | | Probability | | |
|---|---|---|---|---|
| | | **High** | **Medium** | **Low** |
| Impact | **High** | High | High | Medium |
| | **Medium** | High | Medium | Low |
| | **Low** | Medium | Low | Low |

Table 4.3: Method of calculation of exposition based on probability and impact

### 4.1.2 Risk types

### 4.1.3 Risk identification

| Identifier | Name |
|---|---|
| R-01 | Optimist planning, "best case" (instead of a realistic "expected case") |
| R-02 | Bad requirement specification |
| R-03 | Design errors |
| R-04 | Lack of key information from sources |
| R-05 | Lack of feedback or support from the security consultants of Tarlogic |
| R-06 | The learning curve of some technologies is larger than expected |
| R-07 | The unexplained parts of the project take more time than expected |
| R-08 | Can not access source material |
| R-09 | Unexpected changes to any of the software used in the project |
| R-10 | Loss of work |
| R-11 | Wrong management of the project's configuration |
| R-12 | A delay in one task leads to cascading delays in the dependent tasks |
| R-13 | The student can not find a way to code the detection of a certain occurrence |
| R-14 | The quality of the product is not enough |
| R-15 | Sickness or overwork |
| R-16 | Performance issues |
| R-17 | Unnecessary work |
| R-18 | Optional requirements delay the project |

Table 4.4: List of the risks of the project

### 4.1.4 Risk analysis and planning

| Identifier | **R-001** |
|---|---|
| Name | Optimist planning, "best case" (instead of a realistic "expected case") |
| Description | An optimistic planning at the start of the project does not take into account problems or delays, and so it does not allocate time for them. |
| Negative effects | Could mean the failure of the project if the objectives can not be accomplished in the time left.<br>Cascading delays, because the work done would not fit the planning. |
| Probability | Medium |
| Impact | High |
| Exposition | High |
| Indicator | There are 3 consecutive delays, after the beginning of the project. |
| Prevention: Avoid | Allocate a bit more time than initially expected for each task, in case something goes wrong. |
| Correction: Mitigate | Redo the planning.<br>Reduce the scope of the project, leaving out initially planned optional increments. |

| Identifier | **R-002** |
|---|---|
| Name | Bad requirement specification |
| Description | The requirements specified at the beginning of the project are not specific enough, are not needed or there are new requirements after the beginning of the project. |
| Negative effects | Possible failure of the project if the objectives can not be accomplished in the time left.<br>Wasted time, due to lack of comunication in the requirement specification. |
| Probability | High |
| Impact | High |
| Exposition | High |
| Indicator | There are 3 changes in the requirements specification. |
| Prevention: Mitigate | Confirm that all the requirements have been identified at the beginning of the project.<br>Assure that there is no ambiguity in the requirement specification. |
| Correction: Mitigate | Redo the requirement specification.<br>Rework of related requirements and work based on them, including the need to test the results.<br>Redo the planning.<br>Reduce the scope of the project. |

| Identifier | **R-003** |
|---|---|
| Name | Design errors |
| Description | A design is not enough or is incorrect.<br>This can be found in later stages, when it is clear that the implementation based on the design would not satisfy the requirements. |
| Negative effects | Having to redesign and maybe redo the work based on the design.<br>Minor delays. |
| Probability | Low |
| Impact | Medium |
| Exposition | Low |
| Indicator | There are 3 designs that need rework. |
| Prevention: Mitigate | Use design patterns if needed (this project should have very simple designs, so it is possible that there is no need to use them).<br>Make the design as simple and modular as possible. |
| Correction: Mitigate | Redesign and probably change and test the work based on the design. |

| Identifier | **R-004** |
|---|---|
| Name | Lack of key information from sources |
| Description | Not having key information from articles, documentation or manuals. |
| Negative effects | Minor delays.<br>Loss of quality.<br>Added difficulty, even if the work is done in time.<br>Maybe rework and test the functionality, even completely, to follow the desired procedure. |
| Probability | Medium |
| Impact | Medium |
| Exposition | Medium |
| Indicator | The duration of the study of the attack and the related tools takes 50% than expected. |
| Correction: Mitigate | Ask the security consultants of Tarlogic for specific information.<br>Possibly the need to rework completely some functionality. |

| Identifier | **R-005** |
|---|---|
| Name | Lack of feedback or support from the security consultants of Tarlogic |
| Description | Because I do not know enough of some technical aspects of cibersecurity to solve all the problems in this by myself in time, Tarlogic has promised to help (in a tutoring way) if a problem arises. This help could be critical to solve or get around some of the most complex problems, which probably happen to be critical points, needing to be dealt with to continue working on that stage. |
| Negative effects | Cascading delays. |
| Probability | Medium |
| Impact | Medium |
| Exposition | Medium |
| Indicator | A simple technical question takes more than 2 working days to be answered or a complex question takes more than 7 working days. |
| Prevention: Mitigate | Ask in a clear way and with as many details as possible.<br>Ask during work hours, to ensure they are available. |
| Correction: Mitigate | Redo planning and possibly change the scope. |

| Identifier | **R-006** |
|---|---|
| Name | The learning curve of some technologies is larger than expected |
| Description | This is a critical need because not having enough knowledge can result in an inefficient approach to accomplishing the objectives. |
| Negative effects | Loss of quality.<br>The work is more complicated. |
| Probability | Medium |
| Impact | Medium |
| Exposition | Medium |
| Indicator | The duration of the study of the technologies takes 50% than expected. |
| Correction: Mitigate | Redo planning and possibly change the scope.<br>Ask the security consultants of Tarlogic for specific help.<br>Maybe the need to rework completely some functionality. |

| Identifier | **R-007** |
|---|---|
| Name | The unexplained parts of the project take more time than expected |
| Description | There is not enough specification on what a tasks implies or not enough planning.<br>This means that a part of the project is not understood as it should, and the work done is not what was expected or is not enough, needing more time to finish. |
| Negative effects | Wasted time that should have been easy to avoid.<br>Loss of quality.<br>Could mean the failure of the project if the objectives can not be accomplished in the time left. |
| Probability | Low |
| Impact | High |
| Exposition | Medium |
| Indicator | A task takes 25% more time than expected and when the causes are investigated it is revealed that there were ambiguous descriptions or planning. |
| Prevention: Avoid | Try to detail every part enough, having no obvious ambiguity. |
| Correction: Mitigate | Possible need to redo the specifications.<br>Redo planning and possibly change the scope.<br>Maybe having to redo related work. |

| Identifier | **R-008** |
|---|---|
| Name | Can not access source material |
| Description | All or part of the source material can not be accessed, probably because the only host of the resource is down. |
| Negative effects | In some cases this could mean a delay in a critical task, delaying the whole project for an unknown period of time. |
| Probability | Low |
| Impact | Medium |
| Exposition | Low |
| Indicator | There have been at least 10 failed attempts to download the source material, at least 5 with a computer A in a network X and at least 5 with a computer B in a network Y. |
| Prevention: Avoid | When possible choose the source with the best uptime. |
| Correction: Mitigate | Redo planning and possibly change the scope.<br>Possible need to cut out the part of the project that depends on this source.<br>Maybe find another source or wait to the original source to be accessible again. |

| Identifier | **R-009** |
|---|---|
| Name | Unexpected changes to any of the software used in the project |
| Description | Changes to base software could affect this project directly or indirectly: programs could fail or not work as expected.<br>This could mean any software changes, from simple syntax to API changes.<br>In a project that does not work in a bleeding edge environment, like this, this occurrence should be very rare and even if it were to happen it would have to interfere with the part of the software this project uses, which (as this is not bleeding edge) normally would be backwards compatible. |
| Negative effects | Minor delays. |
| Probability | Low |
| Impact | Low |
| Exposition | Low |
| Indicator | The software is not working as expected due to a change in another software version. |
| Prevention: Mitigate | When possible use software that follow good design guidelines and try to be backwards compatible. |
| Correction: Mitigate | Need to adapt the software to work as expected or remove the related functionalities. |

| Identifier | **R-010** |
|---|---|
| Name | Loss of work |
| Description | Due to a bad configuration management or something else, there is a loss of work related to this project. |
| Negative effects | Need to do again the work already done but lost.<br>Depending of the time needed to recover the work, there could be minor or very big delays, planning, changes to the scope of the project and even its failure. |
| Probability | Low |
| Impact | High |
| Exposition | Medium |
| Indicator | The need to replicate already done work is greater than 30 minutes. |
| Prevention: Mitigate | Take snapshots of key status for each virtual machine.<br>Automate backing up the data and store the copies both in a cloud storage service and in a local disk. |
| Correction: Mitigate | Recover the last backup available of the work.<br>If needed work even outside schedule and in holidays. |

| Identifier | **R-011** |
|---|---|
| Name | Wrong management of the project's configuration |
| Description | The project's configuration is inefficient or lacks work.<br>For example due to unclear changes or taking too long to commit changes. |
| Negative effects | Maybe the failure of the project if the objectives can not be accomplished in the time left.<br>Possibly wrong baselines or identification of the configuration elements.<br>It could be that it takes more time than expected to manage the project.<br>The project suffer delays because the need to redo management work and/or planned tasks. |
| Probability | Medium |
| Impact | High |
| Exposition | High |
| Indicator | There are 3 delays because of the configuration of the project. |
| Prevention: Avoid | The configuration of the project should be just complex enough (whithout ambiguity, to ensure a proper management), but not too much complex (which would be hard to follow).<br>Use of familiar and standard tools, like Git.<br>Optionally use an easier to manage lifecycle.<br>Study of the configuration management done in previous final degree projects, to get a proper idea of its scope and details. |

| Identifier | **R-012** |
|---|---|
| Name | A delay in one task leads to cascading delays in the dependent tasks |
| Description | A task gets delayed and one or more tasks depends on its completion to start, so they get delayed too. |
| Negative effects | Cascading delays. |
| Probability | Medium |
| Impact | Medium |
| Exposition | Medium |
| Indicator | At least 2 tasks are delayed, due to only one of them needing more time. |
| Prevention: Avoid | When planning, avoid task dependencies whenever possible.<br>Optionally use a lifecycle based on increments. |
| Correction: Mitigate | Redo planning and possibly change the scope. |

| Identifier | **R-013** |
|---|---|
| Name | The student can not find a way to code the detection of a certain occurrence |
| Description | It could be that the knowledge of the student is too limited or the problem has too much logical or mathematical difficulty. Another possibility is that the event is impossible to detect with the current technologies. If so, this impossibility could be hard to assure too, due to the complexity of nowadays technology. |
| Negative effects | High difficulty to estimate the time needed to detect the event. Cascading delays. |
| Probability | Low |
| Impact | Low |
| Exposition | Low |
| Indicator | Finding a way to detect the occurrence takes 30% more time than planned. |
| Prevention: Mitigate | Have as much information on the problem as possible, the more detailed the better. |
| Correction: Mitigate | Ask the security consultants of Tarlogic for help.<br>Demonstrate that it is possible to detect it. |

| Identifier | **R-014** |
|---|---|
| Name | The quality of the product is not enough |
| Description | The final result is does not comply the quality standard set for this project. |
| Negative effects | The incorporation to the official repository gets rejected.<br>Redo planning and possibly change the scope.<br>Analysis of the changes needed to improve the quality. |
| Probability | Low |
| Impact | High |
| Exposition | Medium |
| Indicator | Getting 10 suggestions to rework functionality. |
| Prevention: Avoid | Follow design patterns.<br>Follow the design guidelines of the official repository when possible. |
| Correction: Mitigate | Need to redo and test work.<br>Pass some kind of quality control. |

| Identifier | **R-015** |
|---|---|
| Name | Sickness or overwork |
| Description | The health of the student deteriorates to the point it affects the project. |
| Negative effects | Probably the quality of the project drops.<br>Possibly delays, that could be hard to specify their limit.<br>Analysis of the changes needed to improve the quality.<br>In the worst case scenario the project can not continue and fails. |
| Probability | Medium |
| Impact | High |
| Exposition | Medium |
| Indicator | There is an unexpected delay because the functionality is not done but there has not been any important issues that could explain it but there is a clear deterioration of the student health. |
| Prevention: Avoid | Stay healthy by following a regular schedule for work and exercising, that includes multiple rest periods.<br>Optionally maintain a diet. |
| Correction: Mitigate | Go to the doctor and follow any instructions to improve the recovery. |

| Identifier | **R-016** |
|---|---|
| Name | Performance issues |
| Description | The program is too heavy for the environment and takes too much resources, because there are not good enough optimizations or the problems are poorly approached. |
| Negative effects | Minor delays. |
| Probability | Low |
| Impact | Low |
| Exposition | Low |
| Indicator | The program takes 30% more resources that at the beginning of the project. |
| Prevention: Mitigate | If possible use efficient algorithms and check the efficiency after the testing is done for each increment. |
| Correction: Mitigate | Analysis of faster ways to solve the problem.<br>Code and test a faster solution. |

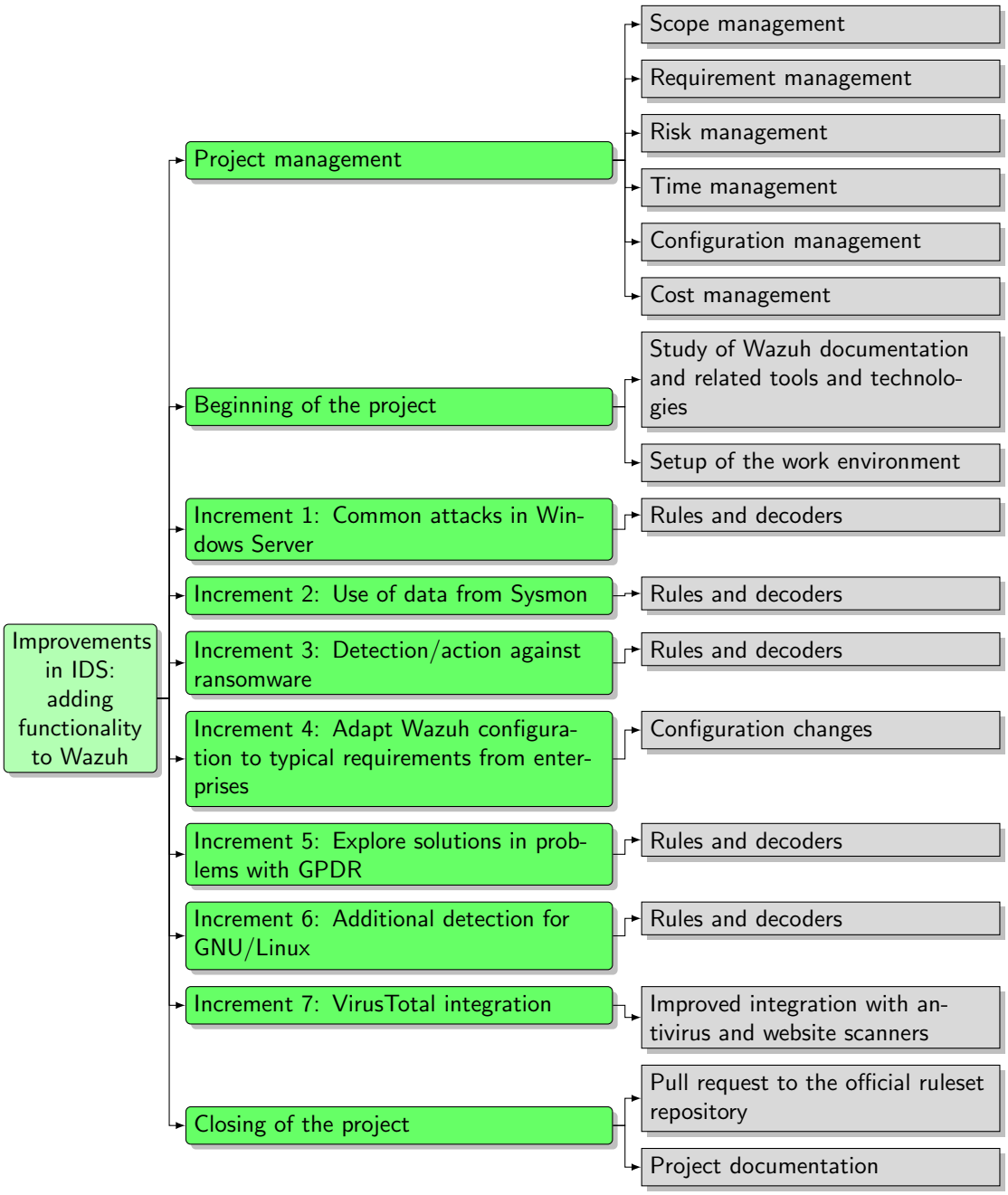
| Identifier | **R-017** |
|---|---|
| Name | Unnecessary work |
| Description | Resources are wasted in work that latter is not used.<br>This could happen because multiple reasons, like wrong assumptions or balancing of the remaining time of the project. |
| Negative effects | Minor delays. |
| Probability | Low |
| Impact | Low |
| Exposition | Low |
| Indicator | There is at least one functionality not necessary or useful for any requirement. |
| Prevention: Avoid | In the design stage make sure that everything is really needed. |
| Correction: Mitigate | Evaluate again if the work planned is really needed. |

| Identifier | **R-018** |
|---|---|
| Name | Optional requirements delay the project |
| Description | Optional requirements get too much time or are treated as vital. |
| Negative effects | The task related to these requirements get too much resources.<br>Vital requirements get less resources, making the project loss value. |
| Probability | Low |
| Impact | Low |
| Exposition | Low |
| Indicator | There is at least one functionality from an optional requirement, when the project is behind schedule and there are vital requirements not yet accomplished. |
| Prevention: Avoid | The optional requirements are planned as optional: they are only done if there is enough time left. |
| Correction: Mitigate | Redo the planning. |

### 4.1.5 Risk supervision

## 4.2 Planning

### 4.2.1 Initial WBS




## WBS dictionary:

1. **Project management**

    a **Scope management**: Scope explanation, set the restrictions of the project and determine what is going to be turned in at the end of the project.

   b **Requirement management**: Analysis, requirement specification and probably a traceability matrix.

   c **Risk management**: Identification, analysis, classification, planning and supervision of risks.

   d **Time management**: Planning (initial and real), any planning changes and necessary measures.

   e **Configuration management**: Documentation on the management of changes and control version.

   f **Cost management**: Cost estimation (direct and indirect) of software, hardware and resources.

2. **Beginning of the project**

   a **Study of Wazuh documentation and related tools and technologies**: Is the base for multiple aspects of the project and if it is done correctly it can mean less hours in related work.

   b **Setup of the work environment**: Installation and basic configuration of the virtual machines of the project, like having a functional Wazuh environment.

3. **Increment 1**

   a **Rules and decoders**: The objective is to be able to detect common attacks in Windows Server (specifically 2016 and 2019), but it should be backwards compatible and depending on the difficulty it could be worth to ensure support for Windows 10 Pro too. This rules are the final product of this increment, which probably will need more time than any other increment, because its heavy study and testing.

4. **Increment 2**

   a **Rules and decoders**: It will need a preliminary study of Sysmon and the ways to use its data to improve detection in certain situations. It is possible that this increment will modify rules and decoders of the previous one.

5. **Increment 3**

   a **Rules and decoders**: This increment tries to produce rules and decoders to detect ransomware and launch alerts and maybe actions against the attack, like rollback to a previous backup or try to stop the attack from repeating in a short period of time.

6. **Increment 4**

   a **Configuration changes**: Adapt Wazuh to the typical requirements from enterprises. This means that an enterprise could choose from a set of templates, with different security profiles.

7. **Increment 5**

   a **Rules and decoders**: Most should be focused on detecting changes on the protected files. Part of this increment should be the investigation on normal problems of these technologies and recent innovations and solutions.

8. **Increment 6**

   a **Rules and decoders**: There would be preliminary study to do, but the increment should be about expanding the already done work in the field, probably focusing in services and security technologies like SELinux or AppArmor.

9. **Increment 7**

   a **Improved integration with antivirus and website scanners**: The idea is to improve the detection as much as possible with the help of VirusTotal malware scanners, which is updated consistently and so it would mean a consistently updated detection for a system with Wazuh without the need to write new rules and decoders. Obviously there is a difference in the scope and objectives of these technologies, which can be redundant, but this could be certainly interesting in some cases.

10. **Closing of the project**

    a **Pull request to the official ruleset repository**: There is a fundamental need to investigate the correct way to organize the the forked repository for a pull request to an official repository like this. In any case the status of the fork should be checked before and there should be a high amount of commits and use a different branch for each functionality, allowing an easier way to select what to admit or not in the official repository.

    b **Project documentation**: The memory and presentation of the project and whatever other documentation if necessary.

### 4.2.2 Initial planning

The tasks marked in red are essential to the project, meanwhile the ones marked in cyan are considered optional and only will be done if there is enough time

left. The tasks marked in yellow are normal, and they are used when there is no need to distinguish between essential and optional.

The next Gantt diagram shows the initial planning, from the draft proposal (31/10/2018) to the end of the project (TODO/02/2019).
Furthermore the last two weeks are marked with a grey overlay to mark that there are only about 17 weeks before the due date of this project (in February). This difference is because the estimation of the tasks was made by the student and so it is not reliable, which means that it could be optimistic or pessimist. Thus the need to either reduce tasks or have more that there were expected to fit.


Figure 4.1: Planning simplification

The rest of the Gantt diagrams are organized in days, for a more detailed planning.
It is important to note that these plannings could change during the project, either because controlled measures or any unexpected reason.
The order they are implemented could change too and that is the reason because these diagrams have not a set date for start and end, yet.

In other words, they could be described as the models for the final Gantt diagrams.

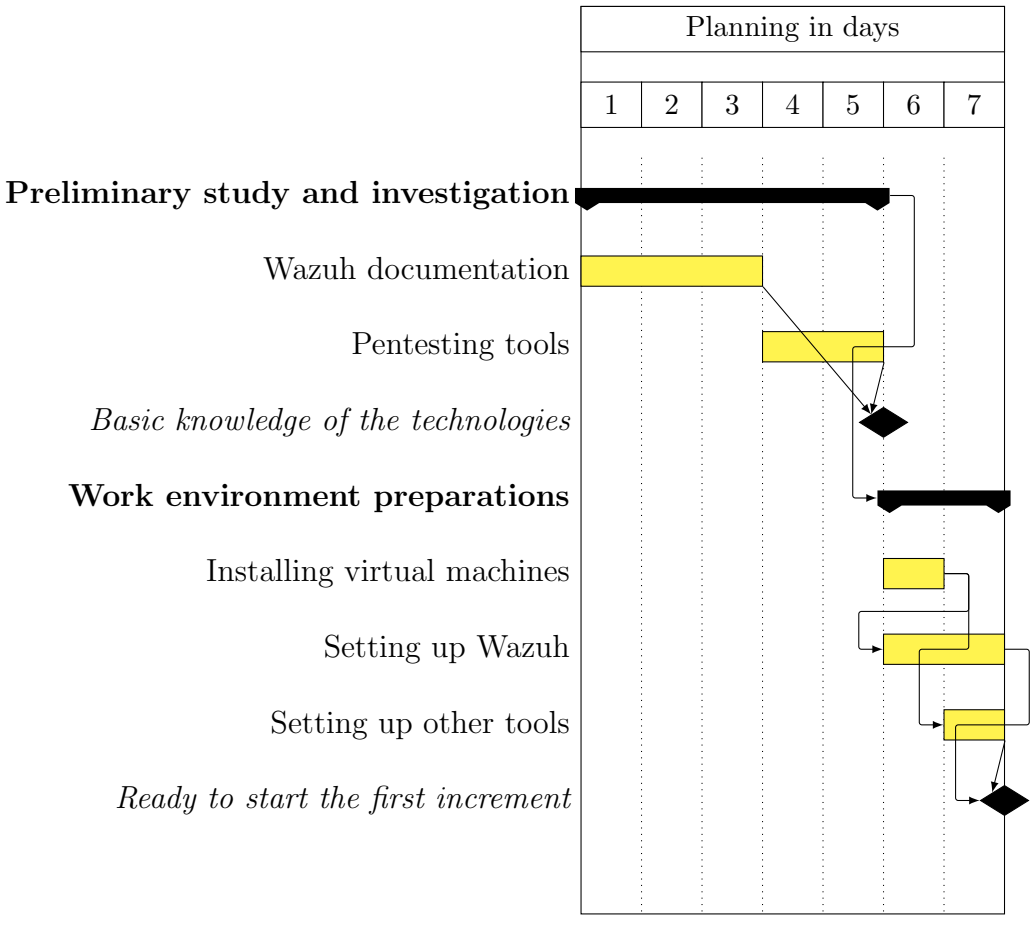



Figure 4.2: "Beginning of the project" planning

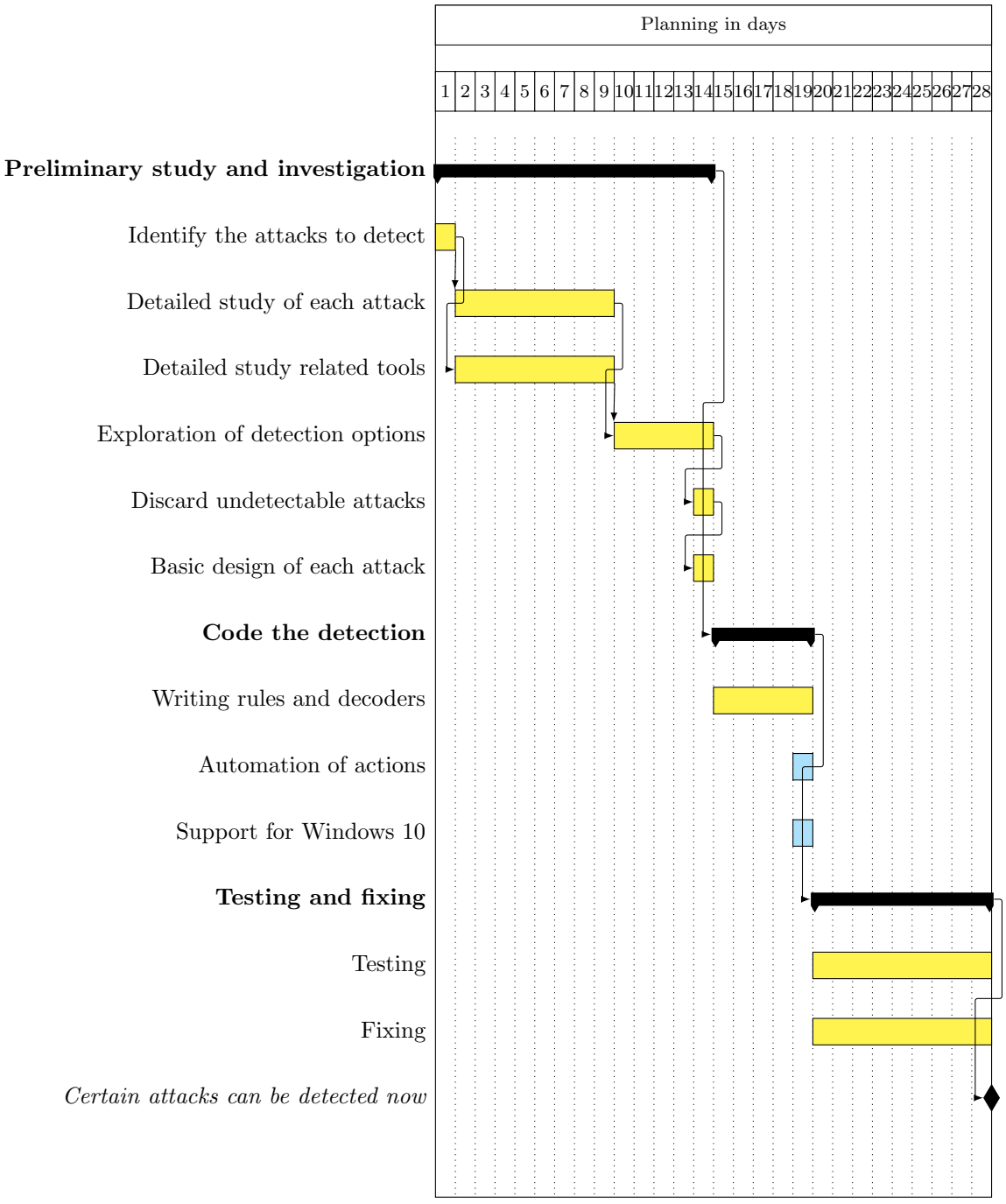



Figure 4.3: "Increment 1: Common attacks in Windows Server" planning

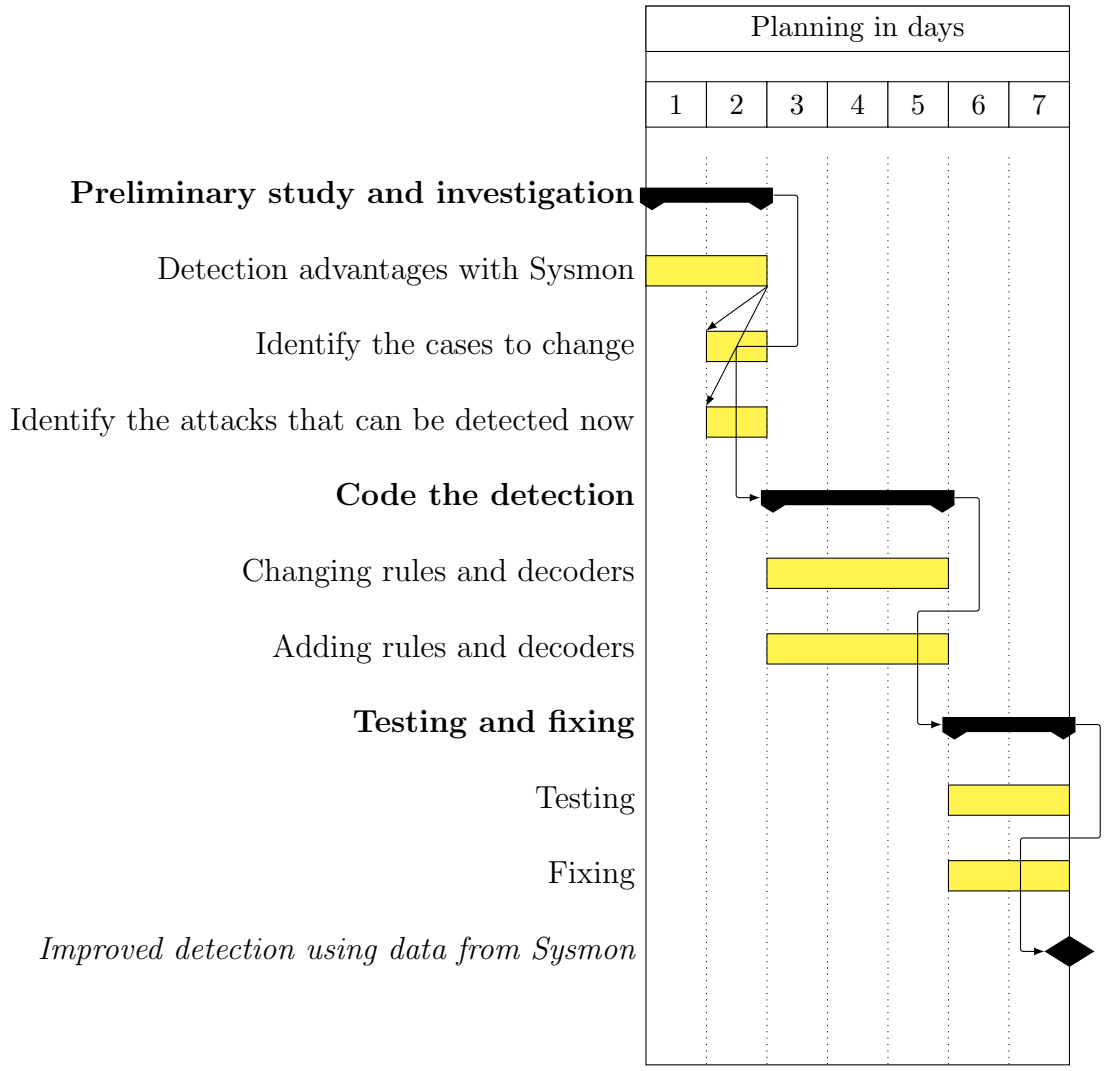



Figure 4.4: "Increment 2: Use of data from Sysmon" planning

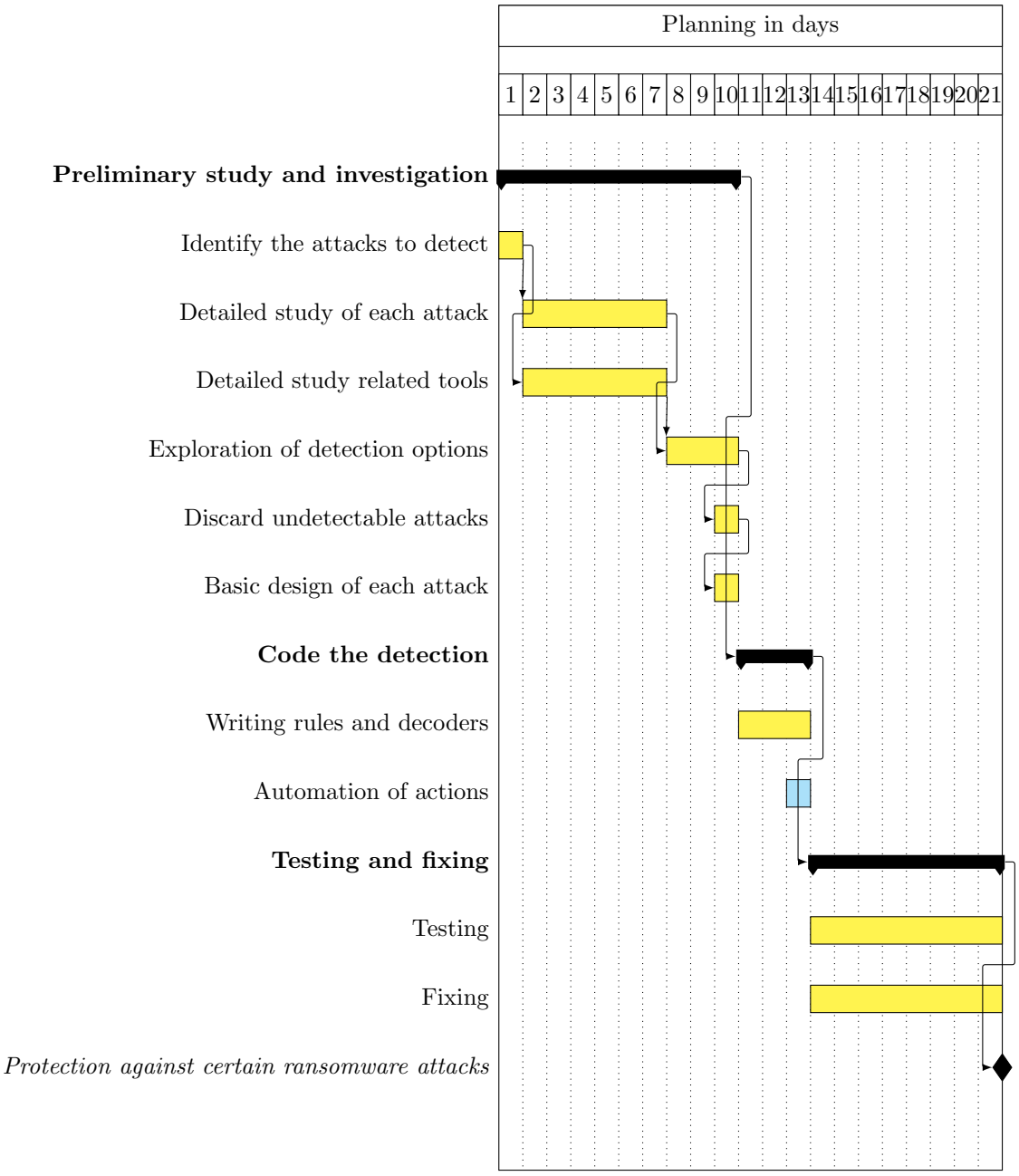



Figure 4.5: "Increment 3: Detection/action against ransomware" planning

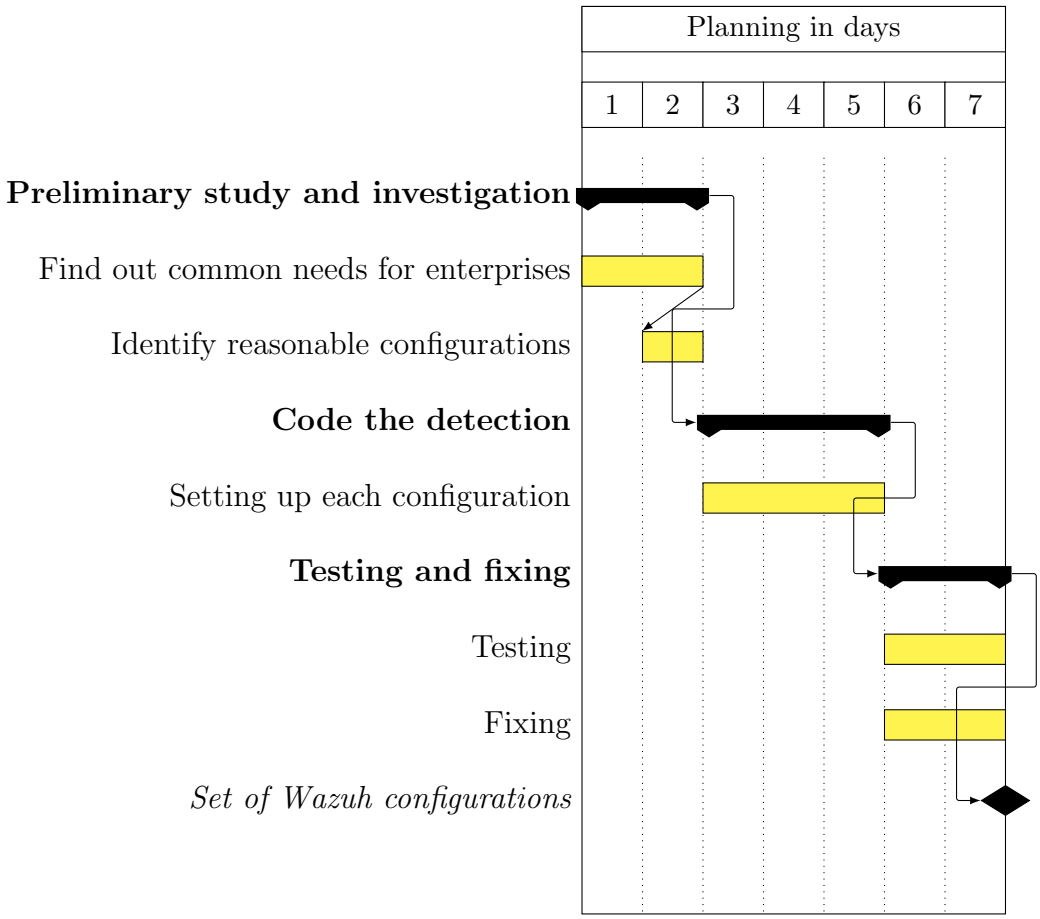



Figure 4.6: "Increment 4: Adapt Wazuh configuration to typical requirements from enterprises" planning

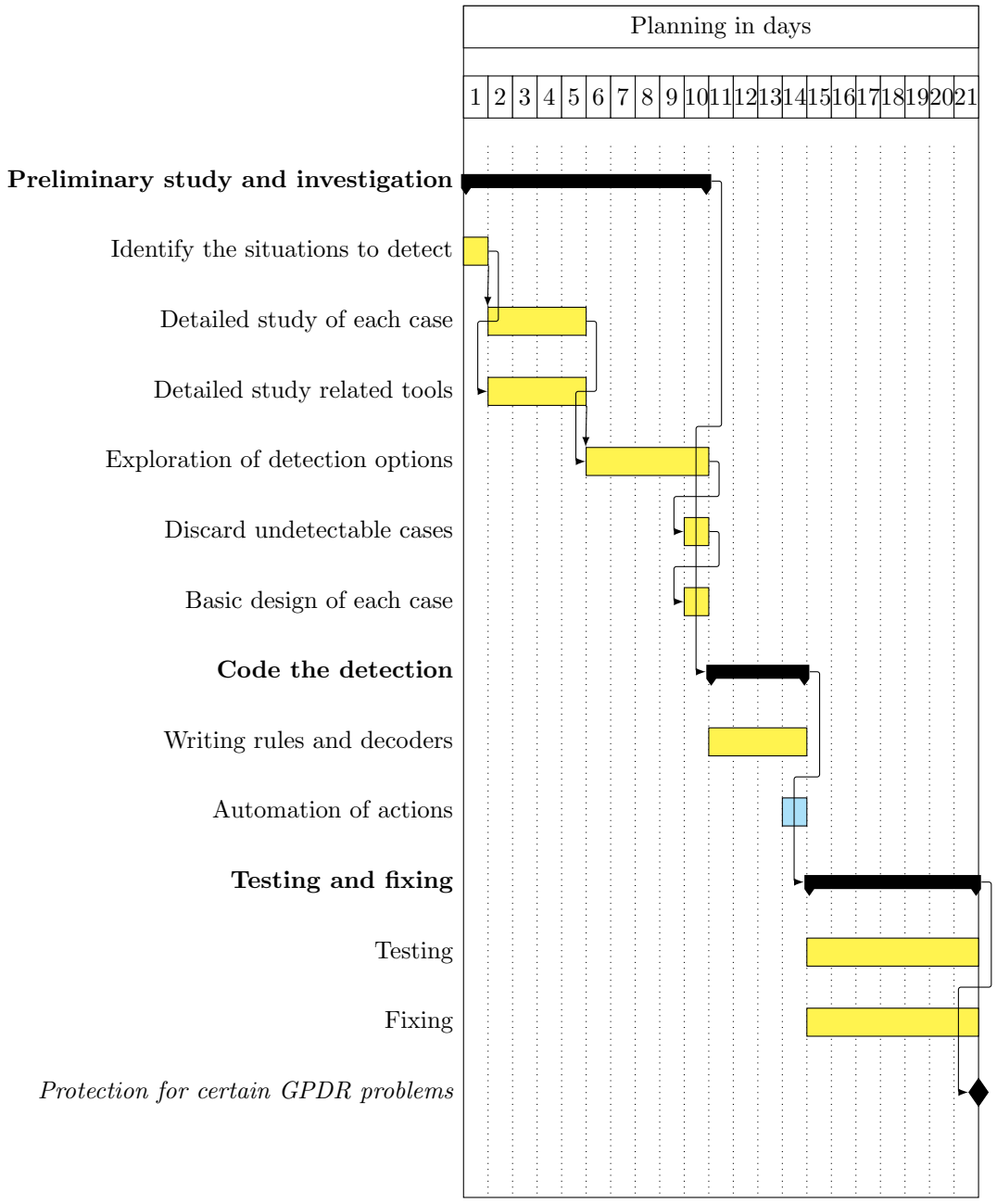



Figure 4.7: "Increment 5: Explore solutions in problems with GPDR" planning

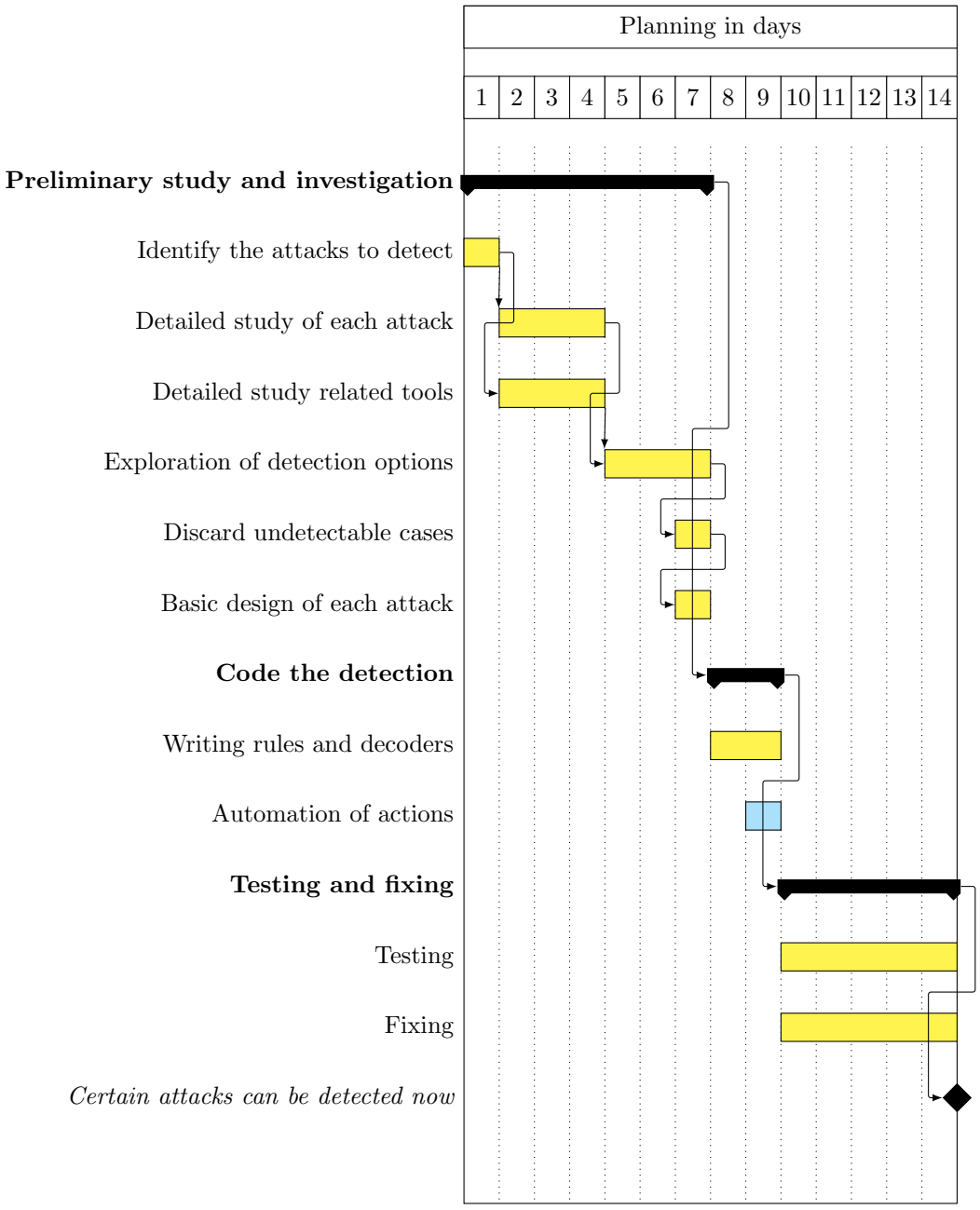



Figure 4.8: "Increment 6: Additional detection for GNU/Linux" planning

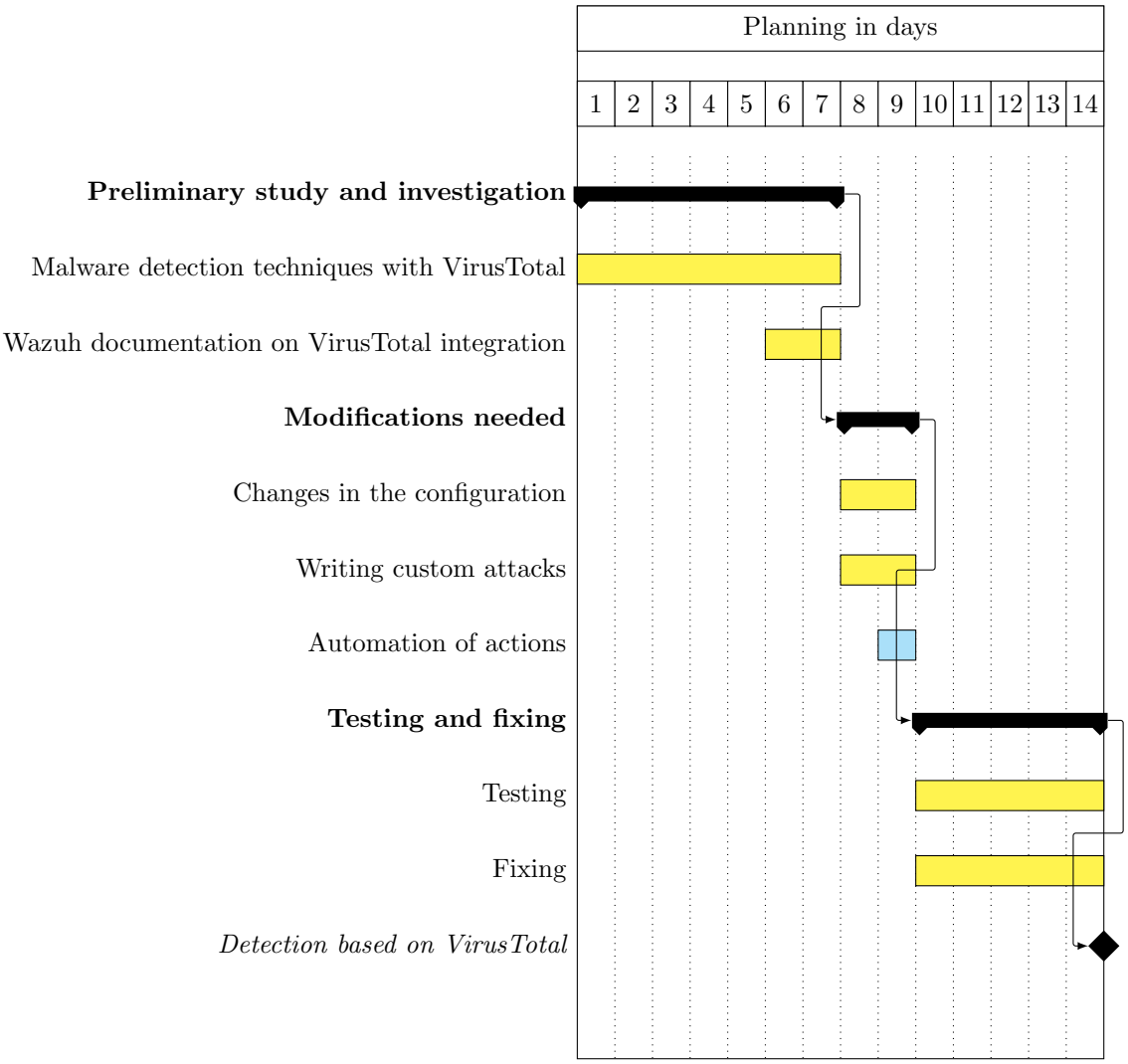



Figure 4.9: "Increment 7: VirusTotal integration" planning

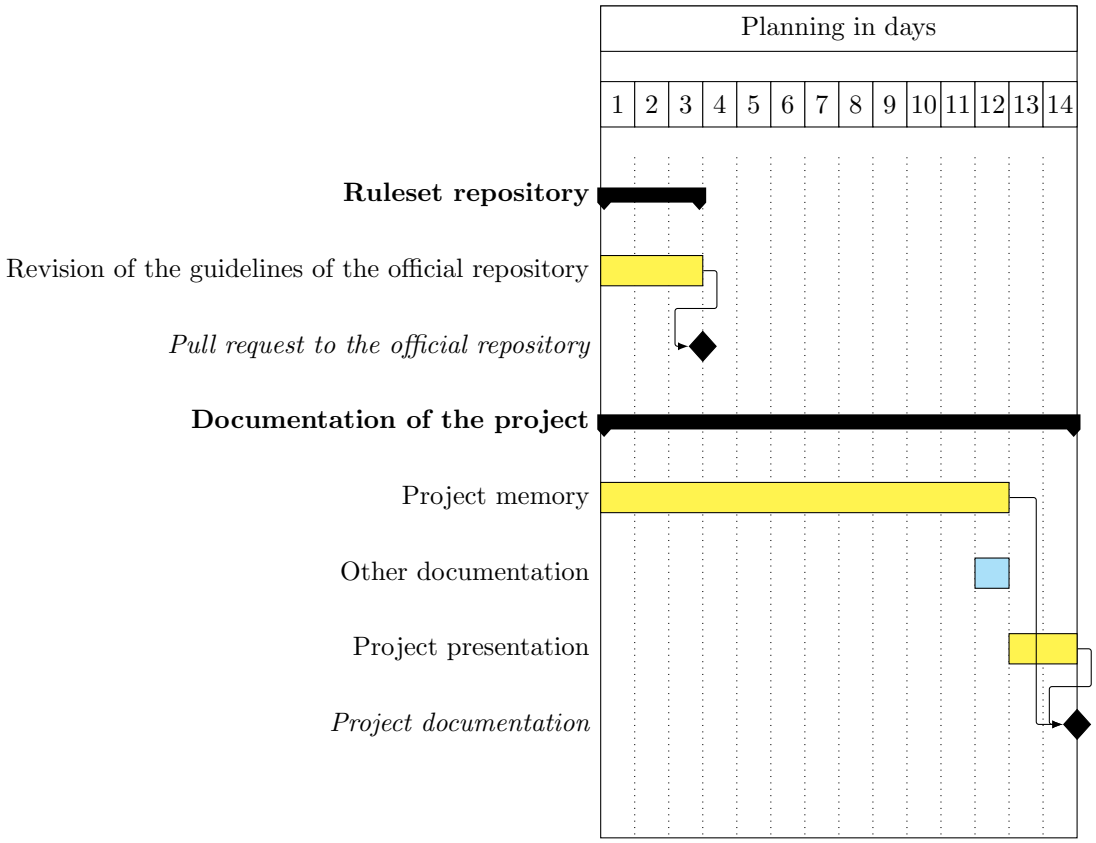



Figure 4.10: "Closing of the project" planning

### 4.2.3 Real planning

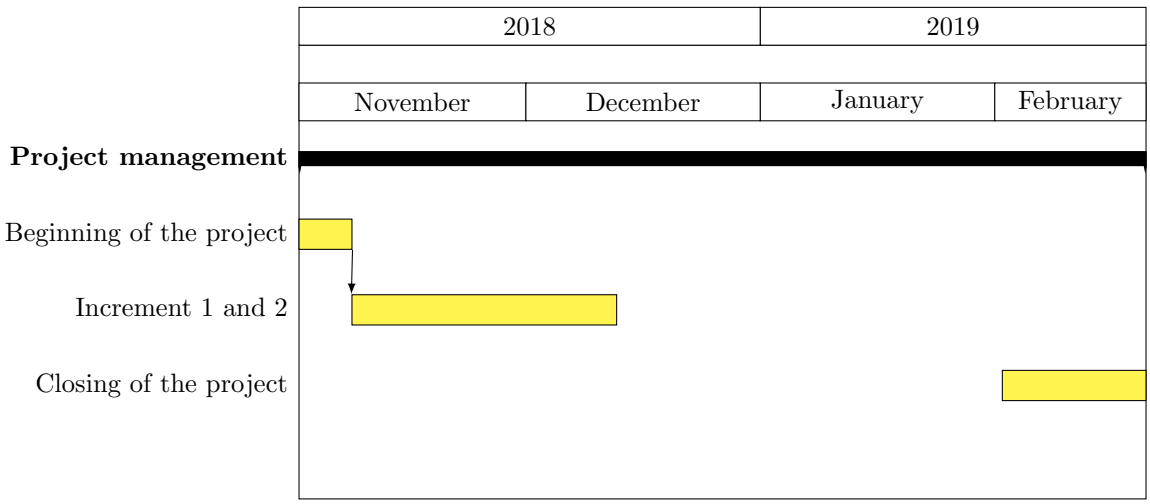



Figure 4.11: Planning simplification

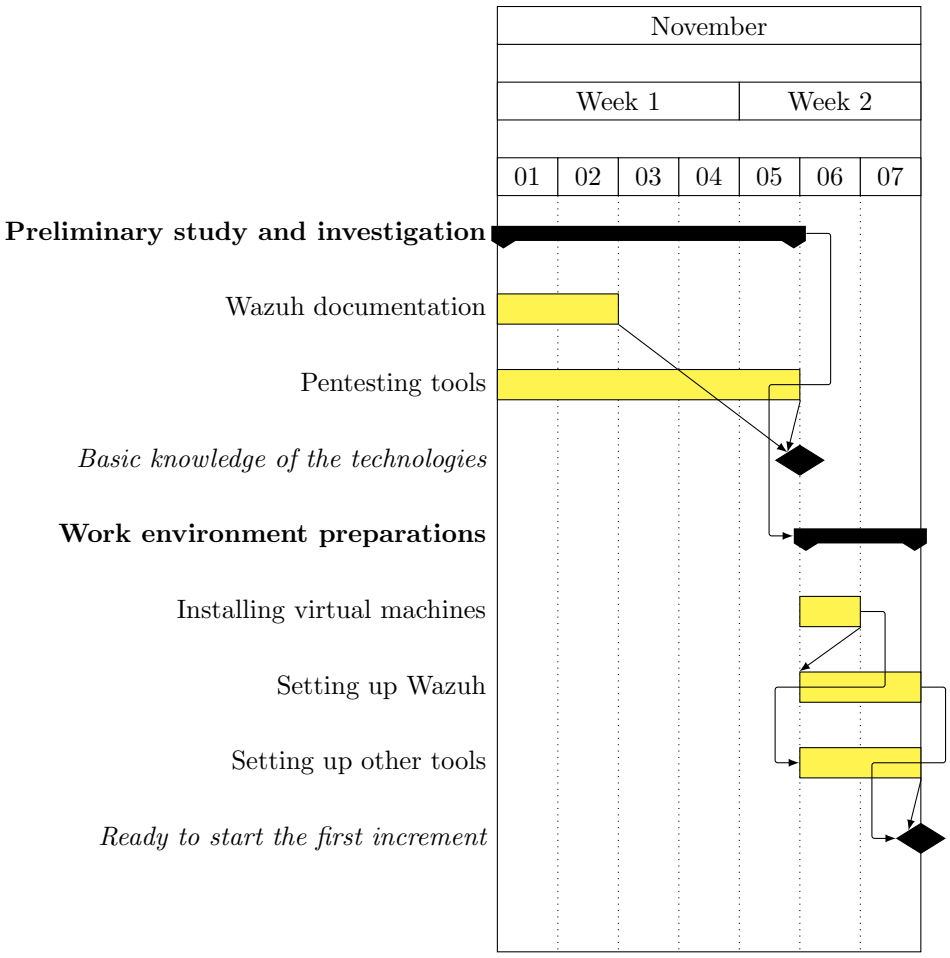



Figure 4.12: "Beginning of the project" planning

# Chapter 5

# Technologies and tools

## 5.1 Development technologies and tools

## 5.2 Pentesting technologies and tools

## 5.3 Documentation technologies and tools

## 5.4 Other technologies and tools