PROJECT TITLE: *'SKELETON KEY'*

TEAM*: 'Team'*

OUTLINE BUSINESS CASE & PROJECT PLAN

Version Number: 2.3
Version Date: 14/11/17

VERSION HISTORY

During the project development process, multiple versions of the final proposal have been iterated through. The table below details how each iteration was named as well as dates of when each version was created and who approved it:

Version Number	Implemented By	Revision Date	Approved By	Approval Date	Description of Change
1.0	Ellis	05/09/17	Andrew	05/09/17	Initial Proposal
1.1	Corey & Jonathan	15/09/17	*'Team'*	24/10/17	Catching the Proposal up
1.2	Ellis & Michaela	24/10/17	*'Team'*	26/10/17	Identification of costs and benefits, critical path or chain, risk/issues, executive summary and high level business
1.3	*'Team'*	26/10/17	*'Team'*	31/10/17	Resource Allocation and Time Line, Identification of Critical Path
1.4	*'Team'*	31/10/17	Andrew	1/11/17	
1.4.2	Andrew	01/11/17	*'Team'*	07/11/17	Adding references to linux kernel docs and improve technical research examples
1.4.5	*'Team'*	07/11/17	Andrew	09/11/17	Justification of project approach & project methodology, spelling and grammar fixes
2.0	Michaela, Jonathan & Ellis	11/11/17	Andrew	11/11/17	Redid formatting of numbering and Legal appendixes, added sections 7.0, 7.1 and 7.2. Added Risks to Section 3.1 - Risks / Issues. Finished Copyright section. Added assumptions and constraints - still to finish. Resolved some minor issues. Additional figure to methodology section. Rework of precedence network and associated section. Added Work Breakdown Structure Section to accommodate for the discussion of the framework and modules.

2.1	Andrew, Jonathan	11/11/17			Added framework and module information under product breakdown structures and resolved issues with research summary, fixed formatting issue with page 21 & 22, improved description of R13, started conclusion for preferred solution and expanded upon existing sections. Added minutes and risk matrix diagram to the appendices. Corrected spelling and grammar globally.
2.2	Michaela, Andrew, Jonathan, Ellis	13/11/17	*'Team'*	*'Team'*	Created a Product Breakdown Structure diagram and re-wrote executive summary taking into account all recent changes. Also did minor changes and proof reading. Created a peer assessment sheet for discussion and final draft. Wrote conclusion to the framework and module information. Discussed the risks that were deemed to be of critical importance. Started on the discussion of version control. Fixed figure numbering. Fixed page numbering and contents table. Resolved minor issues and fixed things.
2.3	*'Team'*	14/11/17	*'Team'*	*'Team'*	Final draft - minor changes, Final activity plan, proof reading.

Version History

Version control was implemented by keeping record of previous versions and appropriate labelling. Centralised applications like Google Docs allowed *'Team'* to edit the white paper submission together and ensure consistency throughout the document.

TABLE OF CONTENTS

1 EXECUTIVE SUMMARY

The University of Abertay has requested, in their brief, a physical pen-testing device using a form of microcontroller to enumerate Windows devices through a physical or wireless interface. Our product, *'Skeleton Key'*, not only fulfils this brief but goes above and beyond to provide a tool which not only collects information about a given system but potentially exploits weaknesses present and grants access for the user.

The project is split into two phases - Framework and Modules. These are explained in greater detail later on, however they are what makes the *'Skeleton Key'* possible. The Framework is composed of the basic system functionality: keyboard emulation, command line interface (CLI), file handling and module integration (More detail on this in Section 7.2). The second phase is the modules, the user interacts with the modules through the CLI and can select modules to run such as enumeration.

'Skeleton Key' provides a multitude of benefits for the client. The product itself: minimises errors through automation; reduces time required for testing; and is made using affordable hardware. Furthermore, another benefit is that '*Skeleton Key'* could be used to train individuals on cyber security and provide insight into the importance of securing a system. Training is of great focus to our client as it is an educational institution which wishes to help local business with their cyber security needs. The device could be used to train students in a classroom environment with both programming and cyber security. Finally, the prototype for *'Skeleton Key'* is being produced free of charge and there is no financial cost to the client for the time and experience of *'Team'*. Overall, *'Skeleton Key'* will be an easy to use device which provides many opportunities for cyber security focused learning and development.

'Skeleton Key' is being developed for the University of Abertay, primarily as an educational tool for teaching both students and local businesses about cyber security and programming. The project aligns with the aims and objectives of the client as an academic research project that allows for teaching and learning as well as providing possibility for marketing in future. The client aims to educate and teach society, *'Skeleton Key'* is able to both of these tasks as well as save time and money for the university.

'*Team'* decided in the planning phase to use a SCRUM methodology. The details of why this methodology was chosen are detailed later in Section 7.7. Overall though, it was chosen for the agility it affords to *'Team'* and ability to foresee and mitigate risks such as over-estimate of time to completion. The *'Skeleton Key'* works by using affordable, easily obtainable hardware (Pi Zero) and an object-oriented program language. For example, if supplying for an normal sized class of cyber security students, 30, we would recommend 10 units of *'Skeleton Key'* which would cost £300 approximately.

The project aims to create an compact, portable device that can fulfil all the requirements asked for by the client. Planning the project in advance has helped to

ensure *'Team'* have systematically and comprehensively considered all alternatives and possibilities for the project (This includes hardware, software, risks, ideas and development decisions). Upon connection to a USB port on a computer, power is supplied to *'Skeleton Key'* which begins the boot process. Once '*Skeleton Key'* is booted it may interact with the computer as if it were another form of USB device - as previously specified by the user. The implementation of the product is split into two main phases - Framework and Modules. The Framework is the backbone of the device and required for module operation. It will perform tasks such as keyboard emulation, storage emulation, providing the command line interface (CLI) and more. Modules leverage components of the framework to provide functionality and the user will use the CLI to interact with these modules.

The key performance indicators that will measure the project's success include the end-product itself which will be presented to the client, all software documentation and the final report of the project planning. Performance of the project will be assessed throughout using proven methods of success evaluation such as burndown charts and will be based on the reliability and efficiency of the product as well as if it performs the tasks asked for by the client such as enumeration as well as additional features proposed by *'Team'*. The evidence of the project's success will be in the final white paper which will detail all aspects of the project planning and implementation as well as any issues or incidents.

2. INTRODUCTION

2.1 PURPOSE OF BUSINESS CASE

The purpose of this document is to provide the justification for undertaking this project for the University of Abertay, based on: estimated cost, benefits and risks of *'Skeleton Key'*. The viability of the project will be continuously monitored by *'Team'* throughout the development process but especially as part of both market and technical research. Furthermore, as part of this process our findings will be communicated to our stakeholders to ensure *'Skeleton Key'* meets their expectations.

'Team' has performed required preliminary research to ensure the viability of *'Skeleton Key'*. *'Team'* also believed it was important to establish how the device would interface with a host machine and what was possible in the given development timeframe. Furthermore, time was taken to ensure the architecture chosen met all of the needs of *'Skeleton Keys'*. For further detail on the research *'Team'* carried out for *'Skeleton Key'* please see Section 5.

3. GENERAL PROJECT INFORMATION

Submission Date	14 November 2017
Requested By	University of Abertay - School of Design and Informatics
Client	Dr. Ethan Bayne
Contact Info.	e.bayne@uad.ac.uk
Project Name	*'Skeleton Key'*
Desired Start Date	9 January 2017
Desired End Date	18 April 2017

3.1 PROJECT DESCRIPTION

3.1.1 BUSINESS NEEDS

The University of Abertay was established in 1888 in Dundee, Scotland. As a university Abertay is educational organization whose main business is that of higher learning and academic research. This project was requested to educate students in cyber security threats and can be used to raise awareness surrounding the ever growing cyber security risks that are present in today's society. As a respected, government funded, organization the University of Abertay must conform to all current laws in regard to this area: the Computer Misuse Act of 1990; Copyright, Designs and Patents Act 1988; and the Data Protection Act of 1998. As well, the University must also ensure to abide with all common regulations: fair use, licensing; and liability waivers.

Cyber security is an increasingly important issue in modern society with 46% of all UK businesses identifying at least one cyber security breach in 2017 *(Department for Culture, Media & Sport, 2017).* Meanwhile the average cost of a breach for large businesses in 2017 sits at £19,600 (*Department for Culture, Media & Sport, 2017*). There is a growing need for better cyber security in our businesses and the issue lies with a lack of education in the subject matter. The *'Skeleton Key'* hopes to solve this issue in a safe environment by providing a valid way to learn and experiment with cyber security under the supervision of the University of Abertay members of staff. Our client specialises in higher learning and aims to provide a tool which can be used to teach those less knowledgeable about the dangers their systems can face. However, the *'Skeleton Key'* and *'Team'* are still held to account by UK law and so must conform to regulation such as the *'Computer Misuse Act'* , *'Data Protection Act'* and *'Copyright, Designs and Patents Act'* however this is explained in much more detail later on in Section 7.9.

Since the University of Abertay is an educational institute open to students it is able to provide *'Team'* with the space to work, making use of the 'Hacklab' and 'Netlab' which is considered an organisational resource. These labs can provide *'Team'* with a safe working environment which also provides all the tools necessary to complete development related to *'Skeleton Key'*. The benefit to the investment of these resources will be our end device.

'Team' is connected to the University of Abertay as we consist of a small group of students in our penultimate year, studying a BSc(Hons) Ethical Hacking. We fit into the project as we have experience in penetration testing and with programming languages, along with wider knowledge regarding computing and cyber security. We all have an interest in creating a physical, modular enumeration tool. If successful this project would not just benefit the client but *'Team'* as a group and individuals in our future work and careers.

3.1.2 GOALS/OBJECTIVES

The representative of the client (Abertay University), Dr Ethan Bayne, has requested the creation of a physical penetration testing device that will perform the *'enumeration'* stage of a typical white box penetration test. The physical device itself has been requisitioned by the client to be used within their organization for teaching and educational purposes, and is required to be built on some form of micro-controller.

'Teams' ultimate goal is to develop such a device that meets the given specification at low cost to the client and that is easily obtainable; something that is lacking in the current market of penetration testing hardware today.

Our proposed product *'Skeleton Key'*, built upon the existing micro-controller architecture of a Raspberry Pi Zero W, aims to meet the client's specification and address the client's needs through the use of the Python programming language. By making use of the official Python style guide our product can be easily and efficiently maintained. The advantage provided to the client is that if in the future they require additional functionality it is trivial to implement.

Alongside the required 'enumeration' capabilities, it is both the goal of *'Team'* and in the interest of the client to provide additional functionality to the product through the inclusion of modules. '*Team'* decided upon several other modules that do not exist within any single available package in an attempt to create a truly unique feature set. Refer to *Section 7.2* for more information on the proposed modules that *Skeleton Key* may include.

Enumeration can be defined as *"A process which establishes an active connection to the target hosts to discover potential attack vectors in the system, and the same can be used for further exploitation of the system." (infosecinstitute.com, 2017)*

Moving away from *'Skeleton Key's'* feature set and looking at the bigger picture, *'Skeleton Key'* would also benefit the client due to the the final product being easy to produce and at a low cost. This would allow the client to easily create their own versions of *'Skeleton Key'* by using the pre-existing framework, modules and documentation that will be designed by *'Team'.*

Since the client is a government-funded, educational institute, they may be restricted by tight departmental spending regulations. *'Skeleton Key'* is therefore beneficial for use by the client since it is very cost effective per unit. Not only is *'Skeleton Key'* low cost but also due to the use of pre-existing hardware high availability is ensured. Refer to *Figure 7.6b* for more information on the costing of *Skeleton Key.*

These versions of *'Skeleton Key'* could then be used by the client for educational and teaching purposes, therefore adhering to the client's needs. The *'Skeleton Key's'* could be used to: teach coding via implementing modules to an open ended solution; raise awareness of the importance of cyber security; and educate in both '*blue team'* (defensive) and *'red team'* (offensive) hacking techniques to the students undertaking ethical hacking based programs.

3.1.3 STAKEHOLDERS

The stakeholders for the project are:

1. Subject Specialists - Ethan Bayne
2. The client - University of Abertay, Dundee
3. Module Tutor - Andrea Szymkowiak
4. The development team - *'Team'*

3.1.4 COMMUNICATION

Communication for the project was achieved using the following methods with the respective stakeholders:

Stakeholder	Method of Communication
All	Verbal
Subject Specialist - Ethan Bayne	Email
Module Tutor - Andrea Szymkowiak	Email
Development Team - *'Team'*	Google Drive, Github, Jira, semi-formal messaging clients

Figure 3.1.4a: Stakeholders

'Team' having chosen to use the SCRUM methodology and as part of this involves having daily meetings. However, due to members of *'Team'* living off site daily meetings presented a difficulty therefore a decision was made to hold meetings twice weekly. Please see Appendix D for all minutes of said meetings.

3.1.5 RISKS/ISSUES

Within the development cycle of any project there are bound to be risks and issues that will arise, affecting not only the client, but also the development team. Some of these risks and issues can be mitigated with appropriate communication and well devised strategies, but there is the possibility that some may be out of the control of the development team i.e. change to scope by the client, acts of God.

'Team' has identified the risks that may occur during the development of *'Skeleton Key'* in order to understand the impacts that they may pose. *'Team'* has also developed a risk matrix diagram to prioritize these risks based upon their likelihood of occurrence. (See Appendix I: Risk Matrix Diagram) From there, mitigations/reductions have been discussed and factored in.

The potential risks identified by *'Team'* and how they potentially could impact the development of *'Skeleton Key'* are detailed in the table below. Note that risks *R7, R8* and *R13* as highlighted in red have been identified as potentially having the highest impact to successful project completion.

R1	Loss of *'Team'* member
	The project is planned for a five person team, if a member has to pull out for whatever reason this workload must be redistributed to other members adding extra pressure to finish on time and to the same high standard.

R2	Member Illness (unable to work)
	The project is planned for a five person team, if a member is unable to work for any length of time this workload must be redistributed to other members adding extra pressure to finish on time and to the same high standard.

R3	Dispute within *'Team'*
	Any arguments or disputes within the team could cause conflict which would disrupt the project potentially causing it become late, go off track or fail altogether. Conflicts could arise inside or outside the project and would still have a detrimental effect on the atmosphere of the members and how they perform.

R4	Member has personal / family issues (unable to work)
	The project is planned for a five person team, if a member is unable to work for any length of time this workload must be redistributed to other members adding extra pressure to finish on time and to the same high standard. Furthermore, traumatic events could have an effect on a members mental state causing them to become unproductive or not perform to the best of their ability.

R5	Damage to current Hardware
	Destruction to current hardware would cause delays on the project as new hardware would have to bought and / or reconfigured. This could cause the project to run over its deadline.

R6	Loss / Corruption of Data
	Loss or corruption of data would cause delays on the project as backups would need to be put into use and old software or coding would need to be updated. This could cause the whole to run over its deadline.

R7	Level of Python comprehension is inadequate
	This risk would require more training and research which may lead to deadlines being missed or tasks taking longer than planned. This, in turn, may lead to the project being late or modules being scrapped due to time constraints.

R8	Modules incompatible with Framework design
	This risk would require *'Team'* revisit the planning of the project and perhaps require rearranging or scrapping certain modules. This may lead to delays or certain features not being present in the final product. As long as the modules were not part of the client specification this would be acceptable, otherwise they would require a meeting with the client to readjust the scope of the project.

R9	Acts of God
	This risk covers incidents like freak storms or disasters which are completely out of the control of the development team and could not have been anticipated. Incidents like this would cause time delays or loss of resources both of which would be to a degree in which the project could not recover and thus cancel the project.

R10	Legal Issues
	Incidents in which the law are broken would call for immediate termination of the project as any form of criminal behaviour is completely unacceptable. Furthermore, all members have a responsibility to report any incidents in which other members or themselves break the law. This may call for immediate removal of said member and could cause delays on the project.

R11	Software Issues requiring Additional Research
	This risk would require more training and research which may lead to deadlines being missed or tasks taking longer than planned. This, in turn, may lead to the project being late or modules being scrapped due to time constraints.

R12	Changes to Scope (by Client)
	The above risk may lead to delays in time as meetings would need to be held to re-establish the scope of the project. Furthermore, additional research would have to take place to judge the viability of the changes requested by the client. This is something the client would need to take into account when requesting changes to the scope as the planning already in place to set for the current scope only.

R13	Over estimated time to completion.
	If the planning stage of the project has resulted in an overestimated time to complete tasks, this will result in less time to spend on subsequent tasks.

R14	Communication Breakdown
	Any communication breakdown within the team could cause disruption within the project potentially causing it become late, go off track or fail altogether. A communication breakdown could have a detrimental effect on the atmosphere of the members and how they perform. It could also lead to work not being down at all, repeated by accident or done incorrectly.

R15	Development technically too difficult
	This risk would require more training and research which may lead to deadlines being missed or tasks taking longer than planned. This, in turn, may lead to the project being late or modules being scrapped due to time constraints.

R16	Feature Creep
	This could cause aspects of the project to go off track or be missed altogether due to work being done on areas that are unnecessary or not part of the client specification. This may lead to the project not meeting the client specifications in time for the deadline and failing.

R17	Real Time Performance Problems
	There is a chance that the software will work but the hardware will take an unrealistic time to complete or perform tasks. This could make the module unviable, depending on the agreed *acceptable time*.

R18	Modules deemed to be too difficult to execute in given timescale
	Due to the complexity of some elements in this project there may be a chance that the creation of modules will be impossible to complete in the given timescale. This could impact development as a finished product may not be delivered.

As previously identified the risks highlighted in red are of significant concern to *'Team'* since:

- **Level of Python comprehension is inadequate:** Each member of *'Team'* has little experience in the use of Python compared to other programming languages. This has been identified to be one of the most impactful risks to successful development as *'Skeleton Key'* relies on Python and without its use in the current design, the project will fail.

- **Modules incompatibility with framework design:** Due to how *'Team'* has decided to create *'Skeleton Key'* by using a framework and module design, each individual module relies heavily on framework elements to function. If a module didn't work with the framework elements that it requires, there is a high chance that the final product will fail to meet the client's specifications. This is because the functionality that the client requires the final product to have relies on these elements so heavily.

- **Over estimated time to completion:** By not running on schedule there is the chance that an unfinished product may be presented to the client. The product may not meet the client's specification if it's not completed in the given development time and would possibly cause the failure of the project. This would in turn impact *'Team'* and its members very negatively.

Factoring in mitigations/reductions during this stage of the project allows *'Team'* to plan ahead and stay on time schedule during the physical development of *'Skeleton Key'*. This is due to there being little or no need to stop development and evaluate potential risks and their mitigations, since they have already been discussed.

Refer to Section 7.10 for a full breakdown of the potential risks that *'Team'* has identified and how they will be successfully mitigated or their impact reduced.

4. HIGH-LEVEL BUSINESS IMPACT

The business processes or functions that will be impacted include the style of teaching as *'Skeleton Key'* acts as a teaching aid for security and programming. The product will also impact the way penetration testing is performed due to increased speed and efficiency . One of the aims of the project is to improve the teaching of ethical hacking by introducing concrete examples to the classroom which results in higher student retention. When the project has been completed, it will also allow for security consultants to focus more on individual or unique issues rather than getting held back with mundane repetitive tasks.

In terms of ongoing operations and future growth, an advantage of the project is its modularity allowing for additional developers the ability to alter and add additional modules as they see fit. Furthermore, the development team may be able to continue adding additional features after the delivery of the project as long as they adhere to the framework requirements. Meanwhile, in terms of the client themselves, university lecturers will be able to add and modify modules on the *'Skeleton Key'* to better reflect the lesson at hand or their personal style of teaching.

The project does not require large volumes of costly hardware - this is one of our key advantages. Our in-house development team will produce the software required by *'Skeleton Key'* and will include user guides to allow ease of use and a deeper understanding into how the tool works as a whole. *'Skeleton Key'* is easy to use due to its simplicity which minimises staff training. As part of the end product, *'Team'* will provide the client with digital copies of all software meaning it can be duplicated onto multiple *'Skeleton Key'* devices.

The performance of the project will be measured throughout using proven development methods of testing and measuring success will be be based on the efficiency and reliability of the product. Furthermore, if the product meets the client specification and our own specification detailed in this document.

'Team' will use a precedence network to visualise the critical path as well as ensure the project is completed on time. Other resources like Gantt charts can be used for time management and to allocate resources. Meanwhile the use of burndown charts will help visualise the current progress on work and ensure that *'Team'* is on track for completion before the deadline.

5. ALTERNATIVES AND ANALYSIS

5.1 MARKET RESEARCH

As proof of a market share to fill, we have looked at what will be *'Skeleton Keys'* competition. To create a successful product, it should be cheaper and more easily obtainable; obtainability will be addressed via ensuring hardware availability. Hardware availability seems to be a big issue within the EU as the products we are comparing to our own are often unavailable; those that are available take a considerable price hike due to import taxes as development is delocalized to the region.

Some of the other products of note that *'Team'* has taken the time to look into, but are not limited to, include:

- USB Armoury - Containing an expansive range of features with the downside of a very high price point and low availability outside of North America. *'Skeleton Key'* intends to provide a similar feature set at a much lower cost with a higher hardware availability.
- Hak5 Rubber Ducky - Capable of acting as a keyboard and performing predetermined actions by an attacker. The *'Skeleton Key'* will be more than capable of replicating all of its capabilities as just one module in the final product.
- Hak5 Bash Bunny - Close to the final product, the *'Skeleton Key'*, the Bash Bunny is a catch all device able to exploit any computer it is configured to attack by the user. Similar to the USB Armoury this product's price point the less than appealing.
- PiKey - By emulating a Network adapter PiKey is capable of stealing user credentials and off loading the password hashes to a cloud service to crack. Once this is successful, the device will then emulate a keyboard and type in the credentials for the attacker logging them in. This product is an open source solution with a similar premise to *'Skeleton Key'* but has a limited scope.
- PiSponder - Achieving the same outcome as PiKey but utilising a different approach. This was chosen to help *'Team'* work toward *'Skeleton keys'* flagship functionality allowing comparison between methods to look to creating an optimised body of code to achieve the same function.
- Poison Tap - By emulating a ethernet connection over USB, Poison Tap can siphon cookies, expose internal routers, and install web backdoors on locked computers. By using this as an example piece *'Skeleton Keys'* development on enumeration modules will be improved.

Finally, similar to other quality open source solutions *'Team'* aims to provide users of *'Skeleton Key'* with documentation for both assisting in the development of new modules and as a user guide. This will expand the usability of our product by providing a way for responsible users with no experience with our product to easily make use of *'Skeleton Key'*.

In conclusion, *'Team'* believes that *'Skeleton Key'* will be a superior product in comparison to many of the currently available alternatives that can be found on the market today. *'Skeleton Key'* performs many of the functions that these alternatives can, but at a severely lower cost and at a greater availability.

5.2 TECHNICAL RESEARCH

VIABILITY

In order to ensure the viability of *'Skeleton Key'* some preliminary research was required. It was important to establish how the device would interface with a host device, and what was possible in the given development timeframe.

Keyboard emulation is perhaps one of the most important components of our framework - many proposed modules rely on it to function. However, existing documentation was sparse, therefore it was considered an excellent starting point for proving viability.

If a choice was made to use Arduino architecture, a large portion of development time would have been required to successfully interface with a host device. As a result different 'drivers' would be necessary for each task we were attempting to complete.

As an example of the technical challenges this represents figure 5.2a below is the keyboard report byte array that would have to be sent to produce what is the equivalent of pressing 'a'. To capitalize this we would have to use a shift modifier as can be seen in figure 5.2b.

Modifier	Reserved	Key 1	Key 2	Key 3	Key 4	Key 5	Key 6
		4					

Figure 5.2a: 'Skeleton Key' - 'a' Key-press

Bit	Modifier
0	Left Control
1	Left Shift
2	Left Alt
3	Left GUI (WINDOWS/SUPER KEY)
4	Right Control
5	Right Shift
6	Right Alt
7	Right GUI (WINDOWS/SUPER KEY)

Figure 5.2b: 'Skeleton Key' - Modifiers

Modifier	Reserved	Key 1	Key 2	Key 3	Key 4	Key 5	Key 6
2		4					

Figure 5.2c: 'Skeleton Key' - 'A' Key-press

Keyboard emulation would have been possible in this manner but unnecessarily complicated - especially when it comes to the language of the keyboard. This method would only stand to inhibit usability and modularity and thus directly conflicts with the goals of Skeleton-Key.

The other method of interfacing over USB is through the use of Linux USB Gadgets. This is a kernel level feature of Linux which provides you drivers which allow your device to act as a specific USB device (depending on the driver selected), for example, G_HID is the '*USB Gadgets Human Interface Device driver*' - or to us the critical component in keyboard emulation. Basic HID handling is done in the kernel, and HID reports can be sent and/or received through I/O on the /dev/hidg* character devices *(kernel.org).* The only drawback of this method is that by default, USB Gadgets is not enabled and thus we will have to do a custom compile of the linux kernel to enable it.

Following on from the earlier example to write 'A' using linux USB gadgets we would only have to do the following:

```
1    modprobe g_hid
2    "A" | hid-gadget /dev/hidg0 keyboard > /dev/null
```

Figure 5.2d: Using a terminal on 'Skeleton-Key' to type 'A' on the host based on documentation at http://www.linux-usb.org/gadget/

This effectively turns on the appropriate driver for keyboard (g_hid) and then pipes the letter 'A' to the this driver which is using the hidg0 keyboard device.

In conclusion, configuring a build of the linux kernel is substantially easier than writing USB drivers for arduino and thus *'Team'* intends to be going down the route of Linux USB Gadgets.

SOFTWARE

Having completed viability research *'Team'* was able to narrow down suitable platforms. Aiming to maintain a high level of functionality without adding bloatware, *'Team'* decided a minimal but well supported OS would be the best choice. By choosing Raspian Lite, *'Skeleton Key'* would be able to maintain availability and stability which aligns with our goals in creating a open-source operating system.

Raspian Lite (based on Debian) is very similar to Raspian with the exception of the GUI; Lite does not include graphical desktop environment package to save on space and processing power. Choosing this version lowers our hardware requirements without impacting on the user experience; which in turn allows us to select lower cost hardware and pass on that cost reduction to the client.

Skeleton Key will be written in Python 3.x as it provides several distinct benefits:

- Hardware Added on Top (HATs) typically provide python libraries that ease usage; we will require a HAT to provide visual feedback to the user.
- Python has a feature called doc strings which allow our developers to document the usage of their functions and classes as they go, this will ease implementation and testing.
- The Python standard library includes things like internet protocols, string operations, web services tools and operating system interfaces. This reduces length of code to be written significantly.
- Python 3.x (as opposed to 2.x) has "*Formatted string literals*" which will help ensure consistency in our CLI.

HARDWARE

By establishing software requirements *'Team'* were able to narrow down detailed hardware requirements. As Linux kernel features are being relied on to provide the device emulation functionality, a suitable device would also be required to include USB On The Go (USB OTG) - the hardware support required for some aspects of use. USB OTG enables the product to act as a host, allowing other devices to connect to the *'Skeleton Key'*.

When selecting appropriate hardware *'Team'* had to consider the established requirements:

- Support for USB On The Go
- Capable of running linux
- Easily obtainable (and highly available)
- Device size - USB ports are usually close together and a larger device may not fit in beside other devices.

While there are many micro-computers which support linux there is only one kind with the availability, support and documentation that '*Team'* believes is adequate - 'Raspberry Pi'. In order to keep costs low and improve portability '*Team'* decided to use

the Raspberry Pi Zero W. The Pi Zero series is the smallest, cheapest type of Raspberry Pi and the only kind that is capable of USB OTG without any hardware add ons.

By opting for the wireless-enabled version we are able to allow for remote access during operation as opposed to limiting functionality to pre-selected modules. While this incurred an increase in cost '*Team'* believes this is a negligible increase given the improvement to user experience.

In conclusion, *'Skeleton Key'* will be significantly cheaper than its competition at £30 per unit and due to a high hardware availability *'Team'* are able to ensure this price will not rise. Through the use of well documented hardware and software *'Team'* can promise a quality product that will be easy to use and be extensively supported.

6. PREFERED SOLUTION

6.1 PRELIMINARY WORK BREAKDOWN STRUCTURE

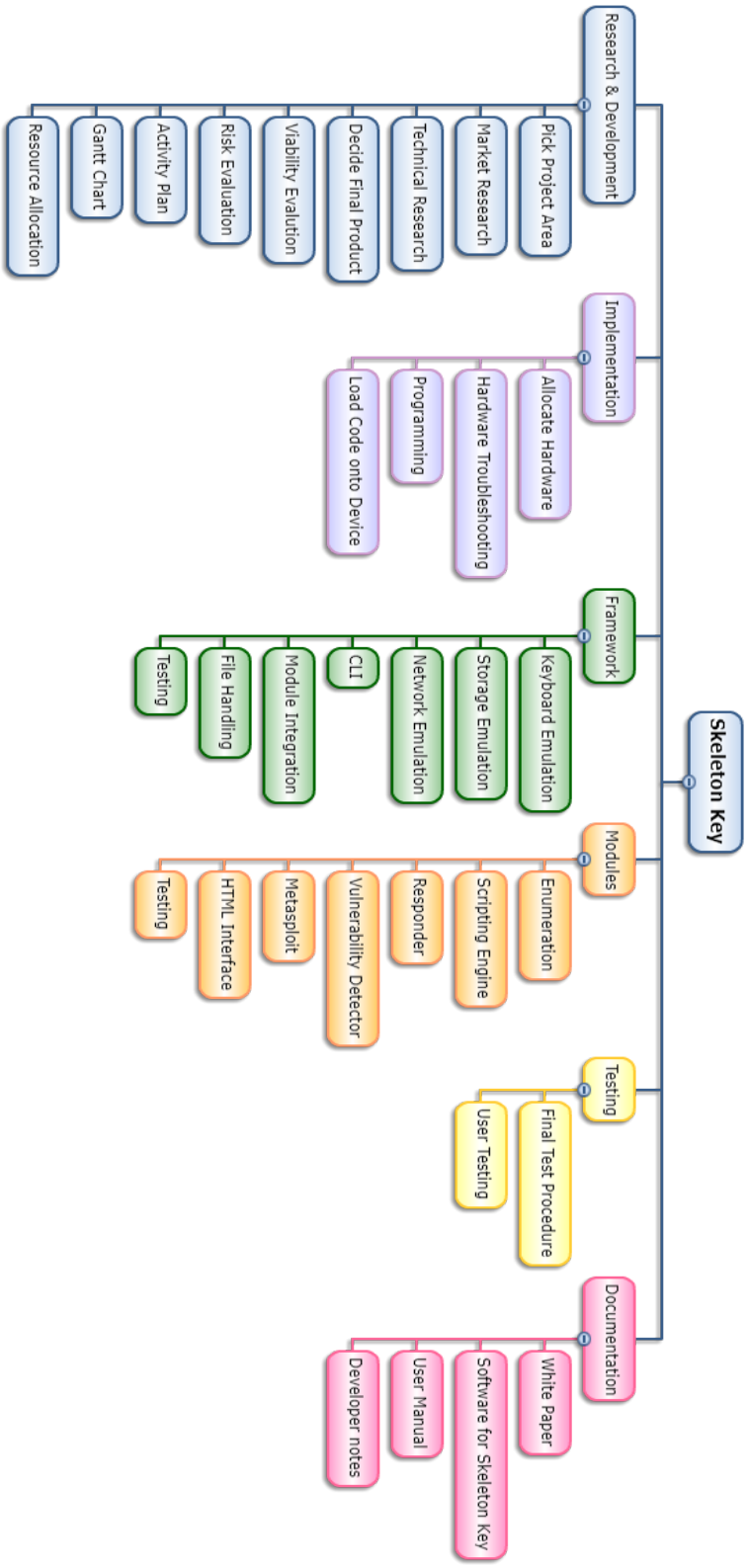


Figure 6.1a: Work breakdown structure diagram

6.2 ASSUMPTIONS AND CONSTRAINTS

Every project involves assumptions and constraints and *'Team'* has made attempts to limit these before development has begun where possible but list below details those that remain. First, constraints will be listed with their appropriate descriptions regarding how each will affect the project's development. Following these will be the assumptions; yet again with descriptions of their scope.

Constraint	Description
Time	A deadline has been set by the client, with a end date of 18/04/18, which means the project must be to a high standard by said date. This also means the project can not expand too far beyond the scope or go off course. Fail-safes must be in place to keep focus and ensure team productivity so that the project is delivered on time.
Current Technology	Though research is always ongoing throughout the project and viability of all modules are checked before being implemented, there is always the chance that issues will arise due to a lack in the technology being used. These issues may lead to *'Team'* creating their own solution.
Personnel	Despite being a five member team there is still a constraint on the workload based on how much work each member of the team can do, within reason. Although members of *'Team'* are dedicated to working hard and producing a high quality product, they are also working on other projects and must take into account their basic needs.

Figure 6.2a: Constraints table

Assumption	Description
Lab Environment	As this is project is for the University of Abertay, Dundee and is being overseen by a representative of said client, the implementation stage of the project will be making use of the Universities on site resources. The assumption is that *'Team'* will have access to the Hacklab (room 4511) for implementation and testing. This is conditional and only applies while the room is not being used to teach classes. Otherwise work will take place on the personal systems belonging to the members of *'Team'*.
Access to Internet	It is relatively safe assumption that *'Team'* will have Internet connection at all times, be this for testing or research. Between the University providing WiFi and the Internet connection members of *'Team'* have at their personal residence. All members will be able to communicate, view work, research and submit work from their personal devices. However, there is a small chance this access could be temporarily unavailable in any of the above locations. This would only cause minor delays to productivity.
Access to Software (VMWare, Visual Studio, Kali Linux, etc.)	*'Team'* requires to use specific pieces of software throughout the development cycle. Each piece of software serves an individual purpose i.e. to allow for code to be written or the creation of a secure test environment. It is therefore assumed that *'Team'* will have access to these pieces of software, as without them the project cannot be efficiently developed.

Figure 6.2b: Assumptions Table

7. PROJECT PLAN

7.1 PRODUCT BREAKDOWN STRUCTURE

There will be three main deliverables at the end of the '*Skeleton Key*' project. These are - the prototype itself, all documentation this includes a user manual, developer notes as well as the final white paper detailing all aspects of the project planning. Finally, *'Team'* will also present the client with digital copies all of software. So that it may be applied to other suitable devices and viewed by the client in more detail for learning purposes.

To visualize this *'Team'* have created a Product Breakdown Structure diagram (seen below) which makes it clear what items will be given to the client at the end of the project and how they relate to one another.

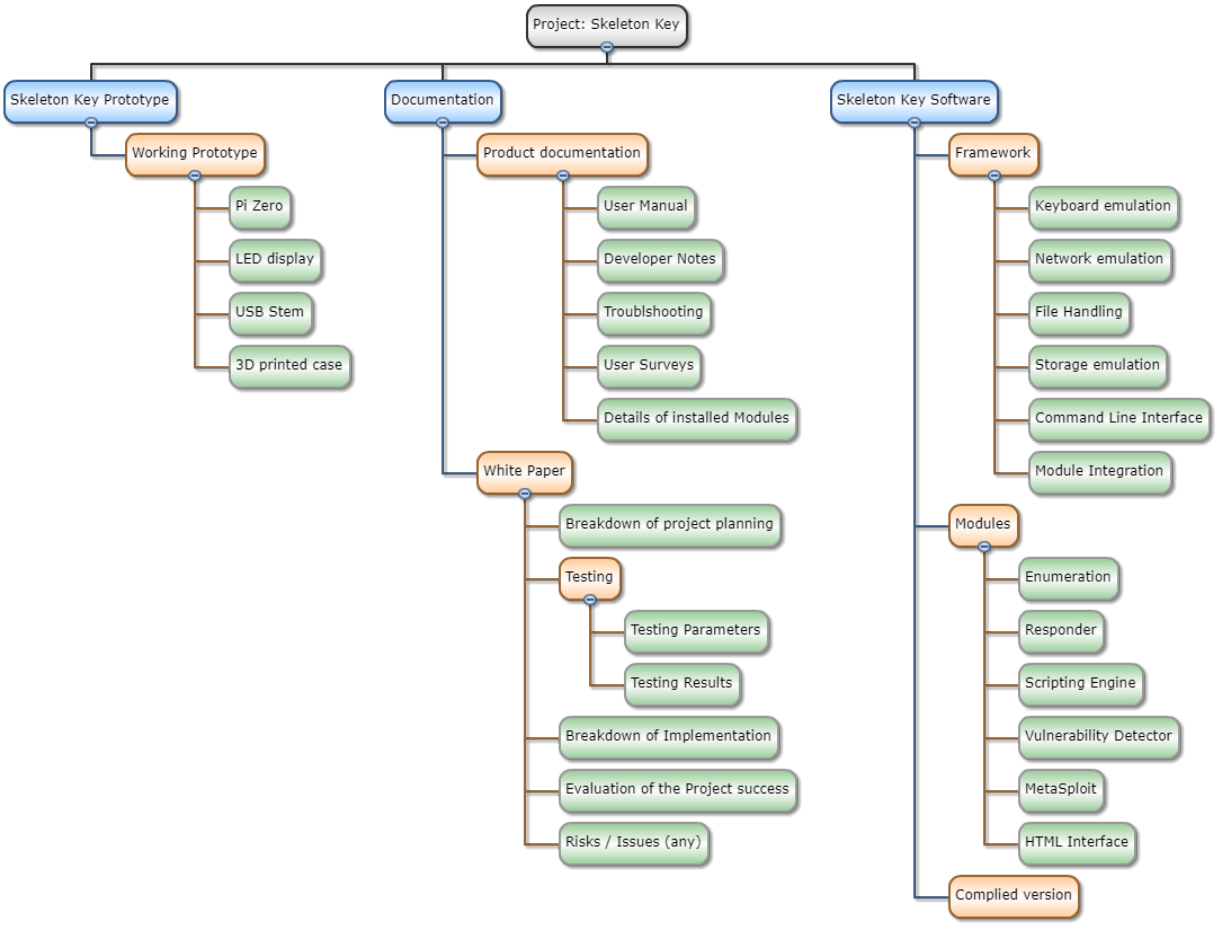


Figure 7.1a: Product breakdown structure diagram

7.2 SKELETON KEY SOFTWARE BREAKDOWN

FRAMEWORK

The framework is effectively the foundation of the entire project - it is required for even basic levels of functionality. The framework is made up of components which perform very specific tasks, these components may be queued to extend functionality however only one component may operate at any one time.

Users won't necessarily see the framework, as the tasks each component performs won't seem to do much unless it is being leveraged by a module.

The components that make up the framework are as follows:

- Keyboard Emulation
  - Used to pass keyboard input to the host machine without requiring user interaction.
- Network Emulation
  - Allows emulation of a network interface; either physical or wireless
- Command Line Interface (CLI)
  - Required for user interaction
  - Allows for display of information
  - Allows for user input
  - Standardises display of module output
- Storage Emulation
  - Emulates attached storage
  - Allows for insertion and extraction of data
- File Handling
  - Enables the creation, movement and deletion of files on the device
- Module Manager
  - Facilitates the loading and unloading of modules
  - Required for module integration

MODULES

By piggybacking off of the functionality that each framework component provides individual modules each providing a unique function to *'Skeleton Key'* can be created.

As discussed in *Section 6.2* it has been agreed with the client that the functionality of *'Skeleton Key'* will exceed the mandated specification by the development and inclusion of other modules aside enumeration. These modules will allow for '*Skeleton Key'* to be used as a more well-rounded penetration testing device.

The tasks performed by the modules themselves will be developed around pre-existing open source tools as it proved infeasible for *'Team'* to create their own tools in the timescale that the client has provided.

Each module will require the use of a combination of framework components to operate. Creating modules in this way allows for the recycling of resources thus keeping *'Skeleton Key's'* storage requirements to a minimum.

Modules can be queued just like the framework components and are the only parts of *'Skeleton Key'* that are visible to the user whilst the device is being operated. Each module can be queued using the CLI arming the device before insertion to a target computer.

An example of how the framework and modules are proposed to work together is demonstrated in Figure 7.2a. The diagram itself represents how a 'Scripting Engine' module would be used to execute a 'Ducky script'. The framework elements required for the execution of this task are highlighted in green.

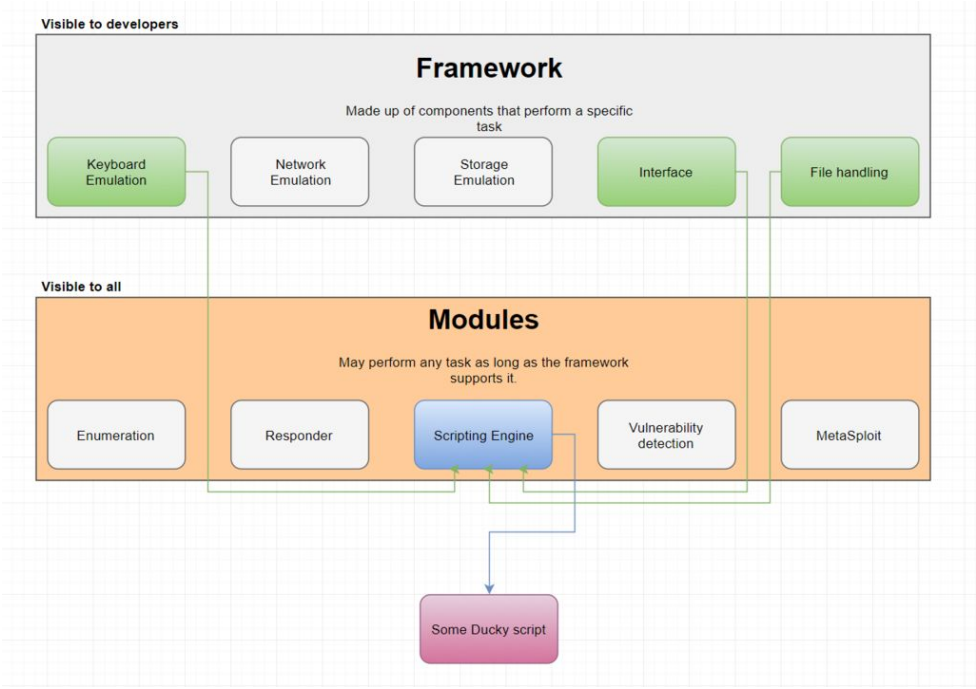


Figure 7.2a: Example of module reliance on frame work. Taken from 'Team's' presentation that took place on the 19th of October

After considerable discussion between '*Team'* and Dr Ethan Bayne a list of modules that were deemed to be of interest to both parties were drawn up. The following information details the what these modules are and the functionality that they will provide to *'Skeleton Key'* if included.

- **Enumeration:**
  - Use of open source tools to provide information regarding the host machine and any information that can be obtained from the connected network. This can be achieved over USB interface but also over wireless if required.

- **Scripting Engine:**
  - Allows for *'Skeleton Key'* to run user made scripts.
  - The *'Scripting Engine'* will make use of '*Hak5's Ducky Script*'. By using this known standard it simplifies the creation of new scripts and allows for the use of already existing Ducky Scripts.

- **Responder:**
  - Makes use of *'Spiderlabs, Responder'* to allow *'Skeleton Key'* to capture password hashes from a computer system by acting as a network interface.
  - *Responder'* captures password hashes by allowing the device to answer specific NetBIOS Name Service queries originating from a target and then steals the hash when the target attempts to authenticate.
  - The final goal of the *'Responder'* module is to give *'Skeleton Key'* the functionality to unlock a computer system without user intervention. If password hashes are obtained they are cracked and then by making use of keyboard emulation the acquired user credentials are typed into the locked machine permitting the user access.

- **Metasploit:**
  - The Metasploit penetration testing framework allows for a wide range of exploitation capabilities to be easily added to the feature set of *'Skeleton Key'*.
  - Metasploit will through information obtained from the *'enumeration'* module allows *'Skeleton Key'* to exploit not only the host machine but also the network at large.
  - The addition of exploitation capabilities through Metasploit to *'Skeleton Key'* adds extremely valuable functionality to the device while also increasing the uniqueness of its feature set.

- **HTML Interface:**
  - Currently it is planned for *'Skeleton Key'* to be operated using a command line style interface when arming the device with module payload(s).
  - To simplify this process and make it more appealing to the eye an HTML interface module has been proposed.
  - This new style of interface would be accessed via a web browser and would allow for a graphical interface be be presented to the user rather than that of a purely text based design.

- **Vulnerability Detection:**
  - Basing a *'Skeleton Key'* module on a vulnerability scanner/detector has proved to be of interest to *'Team'*.

- The addition of such a module would permit *'Skeleton Key'* to scan a target and determine if any vulnerabilities could be used to exploit the device in question.
- If implemented this module would be based around open source vulnerability detectors/scanners such as OpenVAS or Qualys Freescan.

Due to the complexity of this project '*Team'* agreed with the client that not all of the modules discussed may be included initially with *'Skeleton Key'.* Each proposed module has therefore been assigned to a category that allows *'Team'* to prioritize module development to ensure that the client specification will be met. These categories are:

Client Requirements:

Modules that the client require to be included by the end of development.

Extended Development:

Will be implemented providing that the client required modules have been successfully constructed, said modules perform as intended and the remaining timescale allows for it.

Developer Interest:

Modules that may not be viable for either timescale restraint or technical complexity reasons but members of *'Team'* have interest in their inclusion.

The inclusion of these modules are of less importance to both *'Team'* and the client. Their inclusion will be assessed on the viability of the module(s) confirmed through research and if available timescale allows for additional development.

Figure 7.2b denotes the categories that each proposed module falls under.

Figure 7.2b: Module listings. Taken from Client presentation that took place on the 19th of October

By using this framework and module design to develop *'Skeleton Key'* it allows *'Team'* to create a truly unique product unlike anything on the commercial market today.

The modular aspect of this design allows for the product to be used in a variety of situations as modules can be switched out on the fly due to the queuing process. This lets *'Skeleton Key'* perform efficiently since the device won't have to be restarted every time a new module is used during operation.

This style of design also allows the client to create their own versions of *'Skeleton Key'* by developing new modules. The implementation of new modules makes use of existing framework functionality reducing the difficulty to implement said modules. The creation of these modules could be used as the basis for an educational project within the client's institute promoting and teaching the use of Python.

7.3 ACTIVITY PLAN

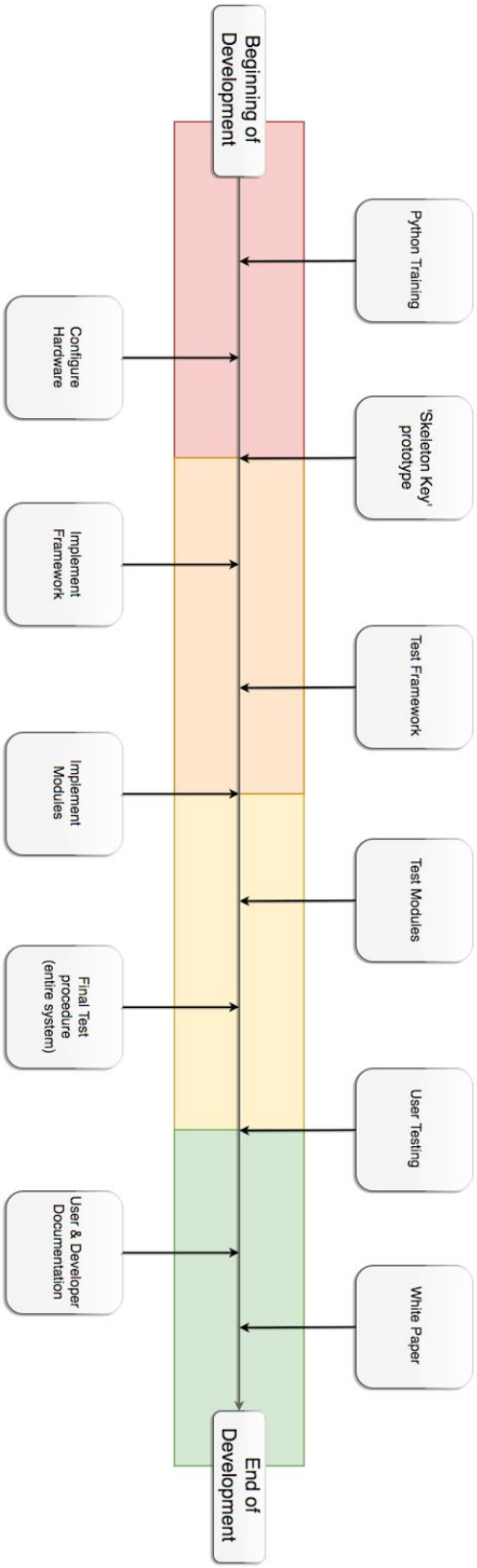


Figure 7.3a: Activity Plan

7.4 RESOURCE ALLOCATION AND TIMELINE

For the purpose of identifying the how critical each component of the project is, the following diagrams were produced. Each component is given an importance and a difficulty between one and ten. The criticality of the component is then given a score by dividing the importance by the cost. If at any point the importance is ten then the criticality score automatically becomes ten as these represent the most crucial components of the project.

As previously discussed, the framework as a whole is a critical component of the project, however, it was decided that a criticality score should still be assessed for components of the framework to allow for the focusing of development on the highest priority components first.

Component	Importance	Difficulty (Cost)	Score	Time Estimate
Keyboard Emulation	10	3	10.00	3 Days
Storage Emulation	6	4	1.50	2 Days
Interface	8	3	2.67	2 Days
Module Integration	10	8	10.00	5 Days
Network Emulation	6	3	2.00	3 Days
File Handling	7	1	7.00	1 Days

Figure 7.4a: V/C & time estimate of Framework taken from 'Team' planning

For the assessment of modules both *'Framework'* and *'Enumeration'* have been assessed with a criticality score of 10; *'Framework'* due to the fact without it the rest of the project will cease to function and *'Enumeration'* because it is a flat requirement of the brief.

Modules	Importance	Difficulty (Cost)	Score	Time Estimate
Framework	10	9	10.00	16 Days
Enumeration	10	1	10.00	1 Days
Responder	5	6	0.83	3 Days
Scripting Engine	9	7	1.29	3 Days
Vulnerability Detection	5	8	0.63	5 Days
MetaSploit	6	2	3.00	1 Days
HTML Interface	2	6	0.33	5 Days

Figure 7.4b: V/C & Time estimate of modules taken from 'Teaming' planning

While time estimation is present in these diagrams it is not factored into the criticality score as time has little bearing on the criticality of components.

For the purpose of timeline management the following Gantt chart was produced.

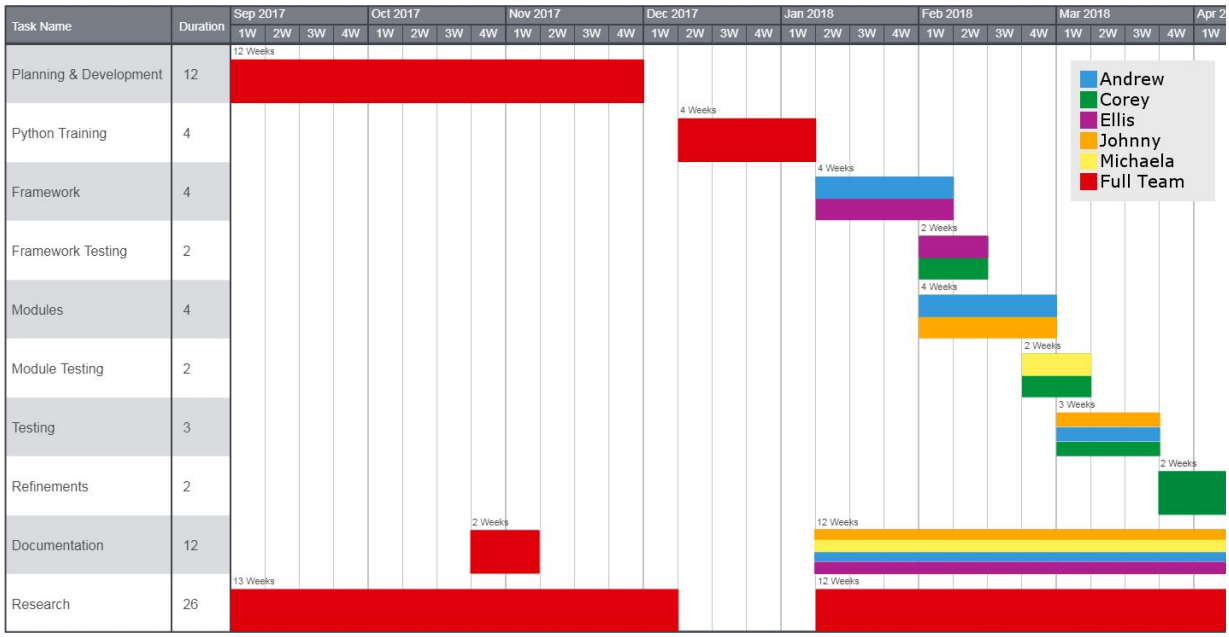


Figure 7.4c: Gantt chart of 'Skeleton Key' Project

The Gantt chart provides *'Team'* with an excellent overview of the project timescale and, by providing each member of the team with their own colour, gives an easily readable visualisation of work breakdown. The chart also allows for exceptional progress tracking on account of the timeline present at the top of the chart.

7.5 IDENTIFICATION OF CRITICAL PATH

To assist *'Team'* in defining priority of the work flow, the precedence network below was produced. In this diagram the orange components represent the critical path while the purple bars represents the beginning of each stage.

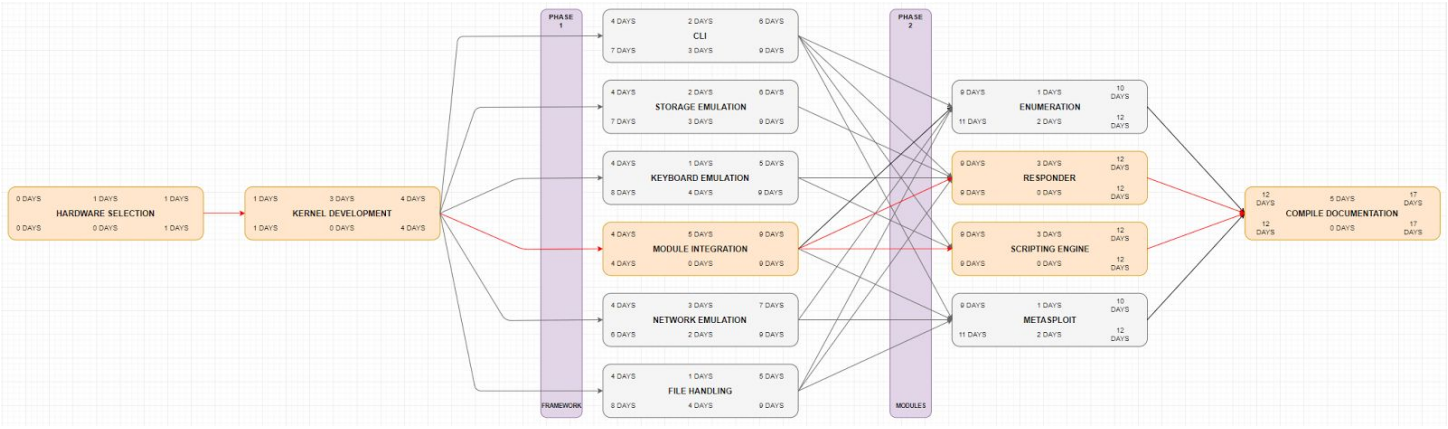


Figure 7.5a: The Precedence Network

The entire framework itself represents a critical path due to the fact that without a functional framework, development of the second phase can not begin. However, prior to the framework development, both hardware selection and kernel development must be completed, and therefore also represent a critical path. The most critical component of the framework has been identified as the module integration as estimates suggest that integration will require the most development time to complete.

During the module stage the critical path becomes more complex due to having both *'responder'* and the *'scripting engine'* together representing the critical path. Simply because both modules require the most time commitment. Following the module stage *'Compile Documentation'* becomes the critical path as it is the final component and under its umbrella it covers all client documentation including the requested white paper and user guides.

By identifying a critical path through the use of a precedence network *'Team'* can estimate the overall time taken to develop *'Skeleton Key'*. By using this information *'Team'* can focus its development on the most critical components ensuring delivery within the deadline.

7.6 IDENTIFICATION OF COSTS AND BENEFITS

Costs and benefits make up a large part of project development as knowing in detail how much the project is worth is incredibly important to the client. The benefits can also decide whether a project continues such as when benefits are monumentally larger than the costs.

Therefore, the development team spent a lot of time considering how *'Skeleton Key'* benefits the client and what possible costs there might be. Below is a breakdown of both and a more detailed description of what each means in terms of the project.

Cost	Description
Time to research	The time needed to research the project was integrated as part of the development phase and was necessary to ensure the viability of the project. Research will happen throughout the project to allow all issues to be addressed with the large amount of possible knowledge on the subject. Allowing a much more successful *'Skeleton Key'* which demonstrates the best of our ability.
Development time	Through discussion within the development team, we have approximated that the project will take one month of active development. This takes into account outside commitments and additional modules assessments that each member of *'Team'* is working on during the 2017-2018 academic year.
Hardware costs	By researching the current market, *'Team'* have been able to not only identify a gap in the market but find a better alternative. However, this does come with a small cost. *'Team'* has approximated £30 for hardware costs per unit

Figure 7.6a: Cost Table

Benefit	Description
Little Training	Very little training is required to use the *'Skeleton Key'* as the majority of processes are automated and minimise user input, thus producing a reduction in human errors.
Gather information quickly	The *'Skeleton Key'* is able to gather large amounts of information quickly.
Cost effective	Since *'Team'* has chosen to use hardware with a smaller price tag, *'Skeleton Key'* provides a cheaper alternative to what is currently available on the market today.
Very low cost for client	Product is offered at £30 including both the physical device and the supporting documentation.
Teaching Aid	*'Skeleton Key'* has the ability to act as a cost-effective teaching aid in the University classroom due to the low cost of £30 per unit. With an average class size of roughly 30-40 approximately 10 *'Skeleton Keys'* at a cost of £300 would allow for a full class to learn about: programming, computer security and, furthermore, provide students with the opportunity to test portable devices.

Figure 7.6b: Benefits Table

7.7 JUSTIFICATION OF PROJECT APPROACH & PROJECT METHODOLOGY

It is important for any project to use a coherent development strategy and for this reason *'Team'* spent a fair amount of time deliberating which methodology to use for *'Skeleton Key'*. Several methodologies were investigated and a decision was quickly made to use SCRUM for the development of the framework due to the following advantages:

- Each component of the framework can be developed as a SCRUM sprint
- SCRUM improves development agility by allowing for team members to be allocated to different sprints.
- Any issues identified can be raised at regular team meeting allowing for easy solutions.
- SCRUM burndown charts allow for excellent time management by showing the amount of remaining work against the time remaining.
- The agility inherent with SCRUM allows *'Team'* to react to and ultimately resolve risks as they are identified
- Component testing can be integrated as part of each sprint improving efficiency.

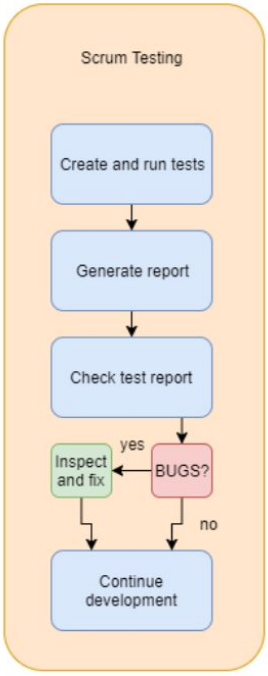


Figure 7.7a: Example of Scrum Integrated Testing

However, for the development of the modules an incremental methodology was initially considered as:

- Incremental development lends itself to numerous small modules.
- Small components can be tested more easily and quickly.
- Incremental development affords exceptional flexibility in regards to changing scope.

After additional consideration the decision was made to use SCRUM methodology for module development also as:

- By managing sprints appropriately all benefits from an incremental approach are retained.
- *'Team'* acquired an educational licence for *'Jira',* a tool for managing SCRUM development allowing for easier project control.
- Maintaining the same development model across phases reduces complexity.

Overall a SCRUM development model is preferred for many reasons. With correctly allocated sprints SCRUM provides exceptional developmental agility and by integrating component testing with the development as part of the sprint quality assurance is streamlined. SCRUM meetings maintain an excellent level of communication between all members of *'Team'* and rapid resolution of issues as they are raised. It also must be mentioned that by acquiring access to *'Jira'* SCRUM development was eased for *'Team'* which further increased its attractiveness as a development model.

7.8 VERSION CONTROL AND REQUIREMENTS CHANGES

With the development of *'Skeleton Key', 'Team'* will have to enforce measures to keep track of the different iterations of both code and documentation. This will help to account for the progress that each member has contributed to the project as well as letting the client monitor progress of the project if they so require.

Whilst coding the framework and individual modules *'Team'* will make use of the open source platform Github to account for version control. Using this platform allows for multiple users to contribute to the Python code at the same time. This is ideal as not all members of *'Team'* will work from the same location. Github also allows for the restoration of old versions of code. This feature will aid *'Team'* when it comes to debugging as the access to previous versions allow a timeline to be drawn up, pinpointing where bugs have been introduced.

In terms of the documentation, *'Team'* has already and will continue to use Google Docs to maintain its versioning. Google Docs is an ideal platform due to its intuitive design and ability to export documents to a variety of different formats.

Keeping the aspects of the project centralized and online using these platforms reduces the viability of data becoming corrupted, the last thing that both the client and *'Team'* would want to occur.

7.9 QUALITY ASSURANCE AND TEST PLAN

To test the quality of our project we intend to run multiple series of testing throughout the development phase in our secure lab where we are authorised users. This is to ensure reliability, consistency and efficiency as well as to stay within the law. This will be based on the project brief and will fulfil the client's requirements. However, we will also check the usability of *'Skeleton Key'* by conducting a user survey and use paired programming to ensure the readability and maintainability of our code.

Using a well-known standard was discussed as part of the planning and since it was decided this project was, at this stage, academic. It seemed unnecessary to weigh the project down with the bureaucracy of international standards. However, marketing the *'Skeleton Key'* would most definitely require a standard. Firstly, there is the ISO Standard which, *"... ensures that materials, products, processes and services are fit for their purpose." (ISO, 2017)*

This organization is composed of representatives from various national standards organisations and has set over twenty thousand standards worldwide covering all aspects of life. This standard would make a good structural start for our product, but would be strict and difficult to achieve.

The second important issue that arose when looking at standards was the legal implications of our product. Under the *'Computer Misuse Act 1990'* is it an offence to make, supply or obtain, *"... articles for use in offence under Section 1, 3 and 3ZA,"* those sections are detailed in Appendix F for reference. This means that marketing a tool like the *'Skeleton Key'* in its finished form is illegal. However, since this project is academic and is for educational purposes only an exception to is made CMA under academic exclusion.

Furthermore, all testing will take place in a secure computer lab where all members of *'Team'* are authorised users. This avoids any legal implications as the tool will not be released to or tested on the general public. In addition to this, in accordance with the *'Data Protection Act 1998'*, all information and credentials will be held on an encrypted drive and disposed off at the end of the project responsibly by a member of *'Team'*. See Appendix G for a detailed view of the *'Data Protection Act 1998'.*

Finally, there is the matter of the copyright which is covered by the *'Copyright, Designs and Patents Act 1988'* . During the project some modules will be implemented using existing libraries or open-source code. This is so no time is wasted during the project by re-writing already existing programs. Every item of code or library that is not written by *'Team'* will be credited. Under normal circumstances '*Team'* would require written permission of the copyright owners for whatever items we use, but since this project is for educational purposes we are exempt from this. This detailed on the *copyrightuser* website where it states:

*"... there are **circumstances when works can be used without seeking the copyright holder's permission**. These are known as copyright exceptions. They include fair dealing for quotation, news, reporting, **education,** private study and parody."* (copyrightuser, 2017)

As explained above, the *'Skeleton Key'* will be tested at multiple stages of development. We will measure how many times the product successfully gathers information, what information, and if it gains access to the desired system. The optimum percentage of successful attempts would ideally be 75%. This is compared with a known baseline from each individual module. Furthermore, it would be useful to conduct a survey of user opinions that sample the *'Skeleton Key'* to gain an idea of the usability of the product.

7.10 RISK MANAGEMENT PLAN

As mentioned in *Section 3.1.5, 'Team'* discussed the potential risks that may occur during the development of '*Skeleton Key'.* The table below denotes information regarding all of risks that were identified as well as the corresponding mitigation or reduction that will be deployed if required.

No.	Risk	Mitigation	Reduction of Impact
R1	**Loss of *'Team'* member**	Regular meetings and communication, discussion of holidays	Redistribute workload to compensate for lost member
R2	**Member Illness (in which they are unable to work)**		Redistribute workload to compensate for lost member
R3	**Dispute within *'Team'***	Regular meetings and communication, decision not to argue or allow disputes to take root.	Resolve conflict through discussion using proven methods.
R4	**Member had personal issues (in which they are unable to work)**		Redistribute workload to compensate for lost member
R5	**Damage to current hardware**	Keep spare hardware and backup all work on centralised application.	Use backups to bring spare hardware up to date and avoid downtime on the project
R6	**Loss / Corruption of Data**	Regular back-ups, centralised work using applications such as Slack and Google Docs to avoid loss of data	Use backups to update work and avoid downtime on the project.

R7	**Level of Python comprehension is inadequate**	All members research Python, paired programming	More research and training of Python, re-evaluation of project design.
R8	**Modules incompatible with framework design**	Extensive research through the project. Possible module and framework re-design	More research and training, re-evaluation of project design.
R9	**Acts of God**		Be prepared for events to happen which are out with the control of the team and were not foreseen.
R10	**Legal Issues**	All members have a working understanding of regulations and the laws which apply to the project	
R11	**Software issues requiring additional research**	Identify quality research resources. Group research sessions	More research and training, re-evaluation of project design.
R12	**Changes to scope (by client)**	Maintain communication with client, using email and face-to-face discussion to avoid miscommunication.	More research and training, re-evaluation of project design.
R13	**Over estimated time to completion**	Continual re-evaluation of time estimates	
R14	**Communication breakdown**	Set minimum number of meetings per week as two	Resolved through group discussion.
R15	**Development technically too difficult**	*'Team'* training	More research and training, re-evaluation of project design.
R16	**Feature creep**	Ensure members stick to planned features and development plan	Re-focus the team during meetings and alter resource allocation back to development plan
R17	**Real time performance issues**	Weekly / Bi-weekly code reviews	
R18	**Modules deemed to be too difficult**	Re-evaluate module necessity	More research and training, re-evaluation of project design.

	to execute in given timescale		

Figure 7.10a: Areas of Risk with Mitigation/Reduction

8. REFERENCES

Department of Culture, Media & Sport. 2017. Cyber Security Breaches 2017: General Business Findings. Available from: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/609187/Cyber_Security_Breaches_Survey_2017_infographic_general_business_findings.pdf [Accessed 7 November 2017]

Department of Culture, Media & Sport. 2017. Cyber Security Breaches 2017: Medium / Large Business Findings. Available from: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/609189/Cyber_Security_Breaches_Survey_2017_infographic_medium_large_business_findings.pdf [Accessed 7 November 2017]

International Organization for Standardization. 2017. International Standards. Available from: https://www.iso.org/standards.html [Accessed 7 November 2017]

Linux-usb.org. (2005). Linux-USB Gadget API Framework. [online] Available at: http://www.linux-usb.org/gadget/ [Accessed 14 Oct. 2017].

Kernel.org. (2017). Cite a Website - Cite This For Me. [online] Available at: https://www.kernel.org/doc/Documentation/usb/gadget_hid.txt [Accessed 5 Oct. 2017].

InfoSec Resources. (2017). What is Enumeration?. [online] Available at: http://resources.infosecinstitute.com/what-is-enumeration/#gref [Accessed 8 Oct. 2017].

CopyrightUser. (2017). Getting Permission - Copyright. [online] Available at: http://www.copyrightuser.org/faqs/question-11/ [Accessed 14 Nov. 2017].

9. BIBLIOGRAPHY

Wikipedia. 2017. *International Organisation for Standards.* Available from: https://en.wikipedia.org/wiki/International_Organization_for_Standardization [Accessed 7 November 2017]

Wikipedia. 2017. *SCRUM (Software Development).* Available from: *https://en.wikipedia.org/wiki/Scrum_(software_development)* [Accessed 9 November 2017]

Stack Overflow. 2017. What is the difference between scrum and agile development. Available from: https://stackoverflow.com/questions/11469358/what-is-the-difference-between-scrum-and-agile-development [Accessed 9 November]

USB Armoury. 2017. The USB armory from Inverse Path. Available from: https://inversepath.com/usbarmory

Hak5 Rubber Ducky. 2017. USB RUBBER DUCKY. Available from: https://hakshop.com/products/usb-rubber-ducky-deluxe

Hak5 Bash Bunny. 2017. BASH BUNNY. Available from: https://hakshop.com/products/bash-bunny

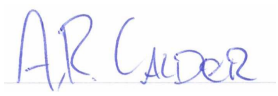
PiKey. 2017. SecurityJon/PiKey. Available from: https://github.com/SecurityJon/PiKey

PiSponder. 2017. dee-oh-double-gee/pisponder. Available from: https://github.com/dee-oh-double-gee/pisponder

Poison Tap. 2017. PoisonTap - siphons cookies, exposes internal router & installs web backdoor on locked computers. Available from: https://samy.pl/poisontap/

APPENDIX A: BUSINESS CASE APPROVAL

The undersigned acknowledge that they have reviewed the ***'Skeleton Key'* Business Case and Project Plan** and agree with the information presented within this document. Changes to this document will be coordinated with, and approved by, the undersigned, or their designated representatives.

Signature:  Date: 14/11/17

Print Name: Andrew Calder

Title: Mr

Role: Team Leader

Signature:  Date: 14/11/17

Print Name: Corey Forbes

Title: Mr

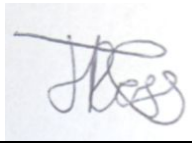
Role: Market Research

Signature:  Date: 13/11/17

Print Name: Ellis Richmond

Title: Mr

Role: Technical Research

Signature:  Date: 13/11/17

Print Name: Jonathan Ross

Title: Mr

Role: Lead on Quality Assurance and Documentation

Signature: Michaela Stewart Date: 13/11/17

Print Name: Michaela Stewart

Title: Miss

Role: Editor, Information Assurance and Legal Adviser

Signature: Date:

Print Name: Ethan Bayne

Title: Dr.

Role: Subject Specialist and Client

Signature: Date:

Print Name: Andrea Szymkowiak

Title: Dr.

Role: Module Tutor and Client

APPENDIX B: KEY TERMS

The following table provides definitions and explanations for terms and acronyms relevant to the content presented within this document.

Term	Definition
'Skeleton Key'	Portable, physical enumeration and exploitation tool made using a microcontroller by *'Team'* in 2017-2018.
CMA	Computer Misuse Act 1990
DPA	Data Protection Act 1998
USB OTG	Universal Serial Bus On The Go - allows *'Skeleton Key'* to act as host
HID	Human Interface Device, defines devices that provide user interaction
CLI	Command line interface
HAT	Hardware Added on Top - provides additional functionality

APPENDIX C: TEAM RULES AND ROLES

RULES

1. Confirmation of absences prior to meetings (Apologies).
2. Maintain communication.
    a. Do not ignore messages.
    b. Ensure *'Team'* is informed of current tasking.
3. Ensure balance of workload.
4. Ensure useful minutes are taken at meetings.
5. Inform members of alterations of objectives and rules.
6. Use of multiple back ups

ROLES

1. Andrew: Project Manager
2. Corey: Market Research
3. Ellis: Technical Research
4. Jonathan: Lead on Quality Assurance and Documentation
5. Michaela: Editor, Information Assurance and Legal Adviser

APPENDIX D: MINUTES & REFLECTION

WEEK 1

WEEK 1: Tuesday 5th/9th

Meeting Information

Objective:	Initial Meeting
Date: 05/09/2017	Location: 4511
Time: 1000-1300	Meeting type: Initial Meeting

Apologies

None

Approval of minutes

No previous formal meetings to note/approve

Agenda

1. Formation of team and networking
2. Team Building exercise
3. Defining and confirming roles of team
4. Deciding on project objective
5. Work towards a complete understand of the Project Proposal template

Decisions

1. Formed Team
2. Team built via exercise
3. Creation of semi-formal communication channels
4. Defined project objective and task
5. Member role creation and approval
6. Collectively emailed confirmation of team and roles to Andrea

WEEK 1: Thursday 7th/9th (Meeting 1)

Meeting Information

Objective:	Follow up and Continue Initialization
Date: 07/09/2017	Location: 4511
Time: 1020-1130	Meeting type: Preparation Meeting

Apologies

None

Approval of minutes

Approved previous minutes

Agenda

1. Defining team rules
2. Work towards completing Section 7 of project proposal

Decisions

1. Team rules discussed and confirmed.
2. Completed Steps 1 – 3 and partial completion of step 4 of Section 7 in the template
3. Internal project title was decided upon as "*Steve*"

WEEK 1: Thursday 7th/9th (Meeting 2)

Meeting Information

Objective:	External Meeting to primarily discuss the addition of a new team member
Date: 07/09/2017	Location: Andrew's Flat
Time: 1720-1800	Meeting type: Casual/External Meeting

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Discussion on the addition of a new member to "Team"
    a. *Benefits and Drawbacks*
    b. *Agreement on the project details on a greater level*
2. Discussion on modules and OS to be used on the Pi
3. Establishing a slack channel for communication between members
4. Change of working project "title"

Decisions

1. It was agreed that the pros outweighed the cons when adding a new member to "Team"
    a. *Pros: Minimal Scope Increase, Balance of Workload, Potential Difficulty Decrease in Semester 1.*
    b. *Cons: Potential Difficulty Increase in Semester 2, Expected Standards Increase Across the Board.*
2. Agreement on Rapsian Lite as the base OS for the Raspberry Pi
3. Agreement on the use of PiKey, Psexec and Script reading modules.
4. Slack channel to be established indefinitely
5. Working Project Title updated to "Skeletal Steve"

WEEK 2

WEEK 2: Monday 11th/9th

Meeting Information

Objective:	Introduction of New Member to *"Team"*
Date: 11/09/2017	Location: Andrew's Flat
Time: 2010-2100	Meeting type: Casual/External Meeting

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Introduction of new member to *"Team"*
2. Discussion on new member's role
3. Rundown of project specification to new member
4. Discussion on the first "Primary" module for *"Skeletal Steve"*
5. Basic discussion on Client Pitch
6. Discussion on rules to new member
7. Re iteration on the importance of CMA & DPA

Decisions

1. New member's role hasn't been confirmed, to be established at a later date
2. Agreement that “ducky style" scripting will be the project's primary module
3. Agreement that the client pitch will include different levels of technical complexity
    a. *ie. Advanced pseudo code in the pitch's appendices*

WEEK 2: Thursday 12th/9th

Meeting Information

Objective:	Creation of V/C	
Date: 12/09/2017		Location: 4511
Time: 1000-1310		Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Creation of V/C style diagram in Google sheets which will be used to aid '*Team*' in determining the importance of each component of the project
2. Further discussion regarding the content of the presentation

Decisions

1. Ellis will head the creation and editing of the V/C style diagram

WEEK 2: Thursday 14th/9th

Meeting Information

Objective:	Further development of V/C
Date: 14/09/2017	Location: 4511
Time: 1000-1310	Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Further development to V/C to add timescales to each component of the project
    a. *These times may be subject to change*
2. Re-evaluation of project name
3. Research by all members regarding information to be included in the presentation

Decisions

1. Project title has been confirmed as *'Skeleton Key'* and all other working titles will be dropped
2. Set up a meeting with Andrea for next session to discuss the development of the V/C and timescales so far

WEEK 3

WEEK 3: Tuesday 19th/9th

Meeting Information

Objective:	Re-evaluating time estimates
Date: 19/09/2017	Location: 4511
Time: 1000-1255	Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Discussion with Andrea on regarding time estimates
    a. *Basis around estimate style has been confirmed to be ok*
2. Re-evaluation of V/C on Google Docs
    a. *Change of existing fields*
3. Discussion on what happens if a module falls through/doesn't work
    a. If a major module fails/over runs, then re-evaluate priorities for optional modules
4. Work on presentation
5. Watch PiKey Youtube video
6. Discuss preliminary research methods
7. Discuss presentation content and format with Ethan

Decisions

1. Digital version of timescale diagram will be created for presentation
    a. *Corey has volunteered to undertake the task*
2. Constant re-evaluation of V/C to ensure the project is going to plan
3. "*Team*" to watch Pi-Key Bsides London presentation:
    a. *Book out library pod*
    b. *Watch during informal meeting at "Team HQ – Nelson Street"*
    c. *Henceforth Nelson Street (meeting location) will be known as "Team HQ"*
4. It has been agreed that links in *"Reading List"* channel of Slack all members of "*Team*" will start to read links by 21/09/2017
    a. If any other research material has been found, encourage other *"Team"* members to submit links to Slack

5. Confirmed that "*Phase 1"* will use a SCRUM methodology, whilst "*Phase 2"* will use incremental (SUBJECT TO CHANGE)
6. Terminology of project details have been discussed and confirmed:
    a. *Phase 1: Framework*
    b. *Phase 2: Modules*
        i. *Components are the individual parts of each phase*
7. Appropriate amendments have been made to the presentation thanks to Ethan's input

WEEK 3: Thursday 21th/9th

Meeting Information

Objective:	Creation of diagrams from practical and discussion of research
Date: 21/09/2017	Location: 4511
Time: 0900-1155	Meeting type: Practical

Apologies

Corey (no notice given)

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Creation of precedence network by Ellis
2. Discussion of research that each member has done
    a. *viability of framework confirmed by Andrew*
    b. Modules that will and will not work
3. Discussion on languages to be used
4. Re-evaluation of V/C due to new research
5. Creation of Gantt chart by Michaela

Decisions

1. Removal of C++ as a language being used (subject to change)
    a. *Due to the ease of swapping driver modules on the fly in Python, thanks to PiKey documentation*
2. Look into the development of custom kernels to enable ALL modules

WEEK 4

WEEK 4: Tuesday 26th/9th

Meeting Information

Objective:	Risk Management
Date: 26/09/2017	Location: 4511
Time: 1000-1215	Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Evaluation of risks
    a. *Creation of risk matrix by Michaela using info from presentation and proposal plan (work in progress)*
    b. *Consider mitigations to risks that have arisen from the risk matrix*
2. Creation of submission checklist with dates
    a. *Delegate each member of "Team" work based on checklist*
3. Discussion on centralized location for document sharing on Slack
    a. *Moving resources created by Jonathan and Corey from the Messenger chat*
4. Consideration of tasks that could be delegated
    a. *Assignment of supervisors to individual tasks (list created in Google Docs)*
5. Discussion with Andrea on precedence network
    a. *Re-identification of critical path*

Decisions

1. Agreed to meet at least 3 times a week to work on the project
2. Finish presentation on either Thursday or Friday
    a. *Not all members may be present, so video conferencing may be used*
3. Discuss presentation submission with Andrea tomorrow
4. Created "Code", "General" and "Research" channels on slack

WEEK 4: Thursday 28th/9th

Meeting Information

Objective:	Critical Chains and Resource Allocation
Date: 28/09/2017	Location: 4511
Time: 0900-1220	Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Completion of Precedence Network by Ellis
2. Completion of Risk Matrix by Jonathan
3. Completion of Risk Matrix Graph
4. Rest of "*Team"* to work on individual sections of Deliverable 2
5. Discussion of 3rd party equivalents to *"Skeleton Key*"
6. Summarisation of market research thus far

Decisions

1. Decided that Deliverable 1 will be completed tomorrow morning (29/09/2017) at *"Team HQ"*
2. Graphs, charts and tabular information will be completed this session in an attempt to focus more time on Deliverable 1 and other commitments
3. Corey has agreed to work on Risk Reduction
4. *"Team"* has decided to use ducky style scripting instead of implementing their own
    a. *This is going to be done so 3rd party developers can easily write modules*
5. For Deliverable 2 on Google Docs, each member of *'Team'* will use a different colour of text
    a. *Andrew: Light Red*
    b. *Michaela: Baby Blue*
    c. *Corey: Light Purple*
    d. *Ellis: Royal Purple*
    e. *Jonathan: Green*

WEEK 5

WEEK 5: Tuesday 3rd/10th

Meeting Information

Objective:	Work on Presentation/General Project Discussion
Date: 03/10/2017	Location: 4511
Time: 1000-1200	Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Discuss messages from communication channels
    a. *Enumeration ideas learnt from alumni on Saturday*
2. Research into Enumeration alternatives/add-ons to NMAP to build an advanced enumeration tool
    a. *Ellis is looking into "Enum4Linux" and its switches*
    b. *Michaela has drawn up a diagram to visualize enumeration possibilities*
3. Corey contacts Gerry regarding the 3D printing of a case for *"Skeleton Key"* within the university
4. Discussion with Ethan
    a. *Show presentation and discuss possible improvements*
        i. *Review and Reflection" has been determined to be inner group workings (Confirm with Andrea)*

Decisions

1. Decided that every *"Team"* member will know general information about every slide
    a. *Each member will then be given specific slides to learn more in depth and talk about*
        i. *Take into consideration potential questions*
            1. *Colour code potential questions, so "Team" will know whose expertise is better to answer the question.*
        ii. *Finalize presentation for Thursday*
        iii. *Allocate slides*
            1. *Rehearse for Monday*

WEEK 5: Thursday 5th/10th

Meeting Information

Objective:	Work on Proposal/General Project Discussion
Date: 05/10/2017	Location: 4511
Time: 0900-1240	Meeting type: Practical

Apologies

Corey – Late

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Assess the quality of the project via Practical 10 activity
2. Work on Project Proposal
    a. *Each "Team" member will be assigned an individual Section to work on during today's session.*
3. Discuss/confirm with Andrea what to contain within the "Review and Reflection" slide for the presentation.
    a. *As discussed with Ethan, it has been confirmed that the slide should contain information regarding the inner workings of "Team"*
4. General discussion regarding the legal aspects of "Skeleton Key" being an open source project
    a. *Concern about "Script Kiddies" using the tool*
5. Ellis digs further into Enumeration tools
    a. *Reading documentation on "Arp-Scan" via Kali Linux CLI*

Decisions

1. During the testing phase, "Team" has agreed that a 75% success rate will be baseline that we will be happy with.

WEEK 6

WEEK 6: Tuesday 5th/10th

Meeting Information

Objective:	Work on Presentation
Date: 10/10/2017	Location: 4511
Time: 1000-1230	Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Each individual member on team is working on their slide allocations for the presentation
    a. Allocation has been created by Michaela via a Google Doc
2. Discussion on methodology:
    a. Whether "Incremental" will actually be appropriate during "Phase 2"
3. Discussion with Andrea:
    a. Whether precise date for completion is subject to change
        i. i.e. "30 days" for execution
4. Discussion with Ethan:
    a. Re-iteration of previous point to get a 2nd opinion
5. Acquisition of Educational Jira licence
    a. Andrew has acquired Jira server license on behalf of team
    b. Whether facilities provided by Jira make "scrum" more viable for both phases

Decisions

1. "Testing" slides within "Phase 1" and "Phase 2" have been dropped due to repetition
    a. *Jonathan has now been allocated the Testing slides for the "Project Overview" section*
2. Stick with current methodology until team management software (Jira) is up and running on server at *"Team HQ"*
3. Number of slides in the presentation to be reduced to ensure *"Team"* doesn't over-run during the presentation

WEEK 6: Thursday 12th/10th

Meeting Information

Objective:	Polishing and rehearsal of presentation
Date: 12/10/2017	Location: 4511
Time: 0900-1255	Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Polish presentation
2. Each "Team" member to write notes for their slides
3. Adjust "roles" so that Michaela's "role" can be defined
4. Practice presentation
    a. *Run through each individual "Team" member's slides and go over notes together*
5. Visit Gerry to collect 3D printed cased
    a. Discussion on case and how to secure the Pi and the lid of the case
6. Members informed of Jira
    a. Each member has their own account
    b. To be used with other version tracking e.g github
    c. only accessible at *"Team HQ"*

Decisions

1. Amendment to roles:
    a. *Jonathan: Lead on Documentation and Minutes*
    b. *Michaela: Editor*
2. Due to "Ethical Hacking" being cancelled this afternoon, it has been decided to run through the presentation this afternoon

WEEK 7

No meetings took place during week 7 (16/10 to 22/10) as it was feedback week. However, '*Team*' did present their client pitch on Thursday of this week (19/10).

WEEK 8

WEEK 8: Tuesday 24th/10th

Meeting Information

Objective:	Work on Project Proposal
Date: 24/10/2017	Location: 4511
Time: 1045-1245	Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Continuation of work on the project proposal
2. Jonathan created a new version of the precedence network using draw.io
    a. *Help was provided by Ellis as he created the initial precedence network*

Decisions

1. Team has decided that clearer, more presentable version of the precedence network was required for the proposal, as the current version wasn't up to standard visually.
2. Amendment to roles:
    a. *Michaela: Editor, Information Assurance and Legal Adviser*

WEEK 8: Thursday 26th/10th

Meeting Information

Objective:	Work on Project Proposal
Date: 26/10/2017	Location: 4511
Time: 0930-1240	Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Continuation of work on the project proposal
    a. *Discussion occurred regarding the formatting of sections and where certain information should go*

Decisions

1. The addition of a new Section to the proposal has been confirmed
    a. *This Section will allow for the discussion of modules and framework*

WEEK 9

WEEK 9: Tuesday 31th/10th

Meeting Information

Objective:	Work on Project Proposal
Date: 31/10/2017	Location: 4511
Time: 1000-1300	Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Continuation of work on the project proposal
    a. *Each member of 'Team' is peer assessing another member's work on the Google Docs version of the proposal, and making alterations were required.*
2. Michaela is creating a more visually appealing version of the Gantt chart for inclusion in the proposal

Decisions

None

WEEK 9: Thursday 2th/11th

Meeting Information

Objective:	Work on Project Proposal
Date: 02/11/2017	Location: 4511
Time: 0900-1240	Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Continuation of work on the project proposal

Decisions

None

WEEK 10

WEEK 10: Tuesday 7th/11th

Meeting Information

Objective:	Work on Project Proposal
Date: 07/11/2017	Location: 4511
Time: 0945-1315	Meeting type: Practical

Apologies

1. Michaela:
    a. *Opted to stay home to accompany her sick boyfriend*
    b. *Michaela has stated she will still work during this allotted session from home*
    c. *The rest of 'Team' were informed of this so no penalty will occur*

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Continuation of work on the project proposal
2. Corey and Jonathan to find and ask Andrea/Ethan questions about the proposal
    a. *Neither Andrea or Ethan could be found*
3. Corey to email Andrea to set up a meeting to discuss the project

Decisions

1. It has been decided that a meeting will be arranged with Andrea for Thursday (9^{th}) to run through the proposal and answer any questions *'Team'* has.

WEEK 10: Thursday 9th/11th

Meeting Information

Objective:	Work on Proposal	
Date: 09/11/2017		Location: 4511
Time: 0900-1245		Meeting type: Practical

Apologies

None

Approval of minutes

Previous minutes have been recorded and saved

Agenda

1. Work on project proposal: Fix grammatical errors
2. Prepare questions to ask Andrea
3. Meet with Andrea to discuss the project proposal

Decisions

1. All members of *'Team'* are to resolve issues that Andrea raised. This will begin tonight (9^{th})

APPENDIX E: PEER ASSESSMENT

As part of the peer assessment, *'Team'* made a peer assessment table and had a group discussion on the 14th of November 2017. During this discussion, members of *'Team'* were able to voice any issues they had and reflect on their work and the work of other members.

In the table, median was used to gain an idea of best mark to award each member. Median was chosen due to its statistical significance. However, the average mark was very close in value.

Attribute/Name	Andrew	Corey	Ellis	Jonathan	Michaela
Attends group meetings regularly and arrives on time	4.0	3.5	4.5	4.5	4.5
Contributes meaningfully to group discussions and work	4.5	4.0	3.5	4.0	4.5
Demonstrates a cooperative and supportive attitude	4.5	4.0	4.0	4.5	4.0
Prepares work in a quailty manner	4.5	3.5	4.5	4.0	4.5
Completes group assignments on time	4.5	4.0	4.0	4.5	4.5
Contributes significantly to the success of the project	4.5	3.5	4.0	4.5	4.5
Helped others with their work when needed	4.5	4.0	4.5	4.0	4.0
Worked well with other group members	4.5	3.5	4.0	4.0	4.5
Median	**4.5**	**3.8**	**4.0**	**4.3**	**4.5**
Grade	**A+**	**A**	**A**	**A+**	**A+**

Name (print) for each team member	**<u>Average</u> mark the group assigned to each team member for their <u>contribution to the team project</u> (select from 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5)**
Andrew Calder	4.5
Corey Forbes	4.0
Ellis Richmond	4.0
Jonathan Ross	4.5
Michaela Stewart	4.5

APPENDIX F: COMPUTER MISUSE ACT 1990

1 Unauthorised access to computer material.

(1) A person is guilty of an offence if—

(a) he causes a computer to perform any function with intent to secure access to any program or data held in any computer [**F1**, or to enable any such access to be secured] ;

(b) the access he intends to secure [**F2**, or to enable to be secured,] is unauthorised; and

(c) he knows at the time when he causes the computer to perform the function that that is the case.

(2) The intent a person has to have to commit an offence under this Section need not be directed at—

(a) any particular program or data;

(b) a program or data of any particular kind; or (c)a program or data held in any particular computer.

(3) A person guilty of an offence under this Section shall be liable—

(a) on summary conviction in England and Wales, to imprisonment for a term not exceeding 12 months or to a fine not exceeding the statutory maximum or to both;

(b) on summary conviction in Scotland, to imprisonment for a term not exceeding [**F4**12] months or to a fine not exceeding the statutory maximum or to both;

(c) on conviction on indictment, to imprisonment for a term not exceeding two years or to a fine or to both.]

3 Unauthorised acts with intent to impair, or with recklessness as to impairing, operation of computer, etc.

(1) A person is guilty of an offence if—

(a) he does any unauthorised act in relation to a computer;

(b) at the time when he does the act he knows that it is unauthorised; and (c)either subsection (2) or subsection (3) below applies.

(2) This subsection applies if the person intends by doing the act—

(a) to impair the operation of any computer;

(b) to prevent or hinder access to any program or data held in any computer; [**F2**or]

(c) to impair the operation of any such program or the reliability of any such data; [**F3** or

(d) to enable any of the things mentioned in paragraphs (a) to (c) above to be done.]]

(3) This subsection applies if the person is reckless as to whether the act will do any of the things mentioned in paragraphs (a) [**F4**to (d)][**F4**to (c)] of subsection (2) above.

(4) The intention referred to in subsection (2) above, or the recklessness referred to in subsection (3) above, need not relate to—

(a) any particular computer;
(b) any particular program or data; or
(c) a program or data of any particular kind.

(5) In this section—
(a) a reference to doing an act includes a reference to causing an act to be done;
(b) "act" includes a series of acts;
(c) a reference to impairing, preventing or hindering something includes a reference to doing so temporarily.

(6) A person guilty of an offence under this Section shall be liable—
(a) on summary conviction in England and Wales, to imprisonment for a term not exceeding 12 months or to a fine not exceeding the statutory maximum or to both;
(b) on summary conviction in Scotland, to imprisonment for a term not exceeding [**F5**12] months or to a fine not exceeding the statutory maximum or to both;
(c) on conviction on indictment, to imprisonment for a term not exceeding ten years or to a fine or to both.

3ZA Unauthorised acts causing, or creating risk of, serious damage

(1) A person is guilty of an offence if—
(a) the person does any unauthorised act in relation to a computer;
(b) at the time of doing the act the person knows that it is unauthorised;
(c) the act causes, or creates a significant risk of, serious damage of a material kind; and
(d) the person intends by doing the act to cause serious damage of a material kind or is reckless as to whether such damage is caused.

(2) Damage is of a "material kind" for the purposes of this Section if it is—
(a) damage to human welfare in any place;
(b) damage to the environment of any place;
(c) damage to the economy of any country; or
(d) damage to the national security of any country.

(3) For the purposes of subsection (2)(a) an act causes damage to human welfare only if it causes—
(a) loss to human life;
(b) human illness or injury;
(c) disruption of a supply of money, food, water, energy or fuel;
(d) disruption of a system of communication;
(e) disruption of facilities for transport; or
(f) disruption of services relating to health.

(4) It is immaterial for the purposes of subsection (2) whether or not an act causing damage— (a)does so directly; (b)is the only or main cause of the damage.

(5) In this section—
(a) a reference to doing an act includes a reference to causing an act to be done;

(b) “act” includes a series of acts;
(c) a reference to a country includes a reference to a territory, and to any place in, or part or region of, a country or territory.

(6) A person guilty of an offence under this Section is (unless subsection (7) applies) liable, on conviction on indictment, to imprisonment for a term not exceeding 14 years, or to a fine, or to both.

(7) Where an offence under this Section is committed as a result of an act causing or creating a significant risk of—
(a)serious damage to human welfare of the kind mentioned in subsection (3)(a) or (3)(b), or (b)serious damage to national security,

a person guilty of the offence is liable, on conviction on indictment, to imprisonment for life, or to a fine, or to both.]

APPENDIX G: DATA PROTECTION ACT 1998

The following Section provided a short rundown on the principles of the Data Protection Act 1998. This is to provide a point of reference when reading the related material detailed in this report. Please see https://www.legislation.gov.uk/ukpga/1998/29/contents for full details

(1) Personal data shall be processed fairly and lawfully and, in particular, shall not be processed unless –
(a) at least one of the conditions in Schedule 2 is met, and
(b) in the case of sensitive personal data, at least one of the conditions in Schedule 3 is also met.

(2) Personal data shall be obtained only for one or more specified and lawful purposes, and shall not be further processed in any manner incompatible with that purpose or those purposes.

(3) Personal data shall be adequate, relevant and not excessive in relation to the purpose or purposes for which they are processed.

(4) Personal data shall be accurate and, where necessary, kept up to date.

(5) Personal data processed for any purpose or purposes shall not be kept for longer than is necessary for that purpose or those purposes.

(6) Personal data shall be processed in accordance with the rights of data subjects under this Act.

(7) Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data.

(8) Personal data shall not be transferred to a country or territory outside the European Economic Area unless that country or territory ensures an adequate level of protection for the rights and freedoms of data subjects in relation to the processing of personal data.

APPENDIX H: COPYRIGHT, DESIGNS & PATENTS ACT 1988

Below is an extract from the *'Copyright, Designs and Patents Act 1988'.* For the full document, please refer to: https://www.legislation.gov.uk/ukpga/1988/48

50A Back up copies.

(1) It is not an infringement of copyright for a lawful user of a copy of a computer program to make any back up copy of it which it is necessary for him to have for the purposes of his lawful use.

(2) For the purposes of this Section and sections 50B [**F3**, 50BA] and 50C a person is a lawful user of a computer program if (whether under a licence to do any acts restricted by the copyright in the program or otherwise), he has a right to use the program.

(3) Where an act is permitted under this section, it is irrelevant whether or not there exists any term or condition in an agreement which purports to prohibit or restrict the act (such terms being, by virtue of Section 296A, void).

50B Decompilation.

(1) It is not an infringement of copyright for a lawful user of a copy of a computer program expressed in a low level language—

(a) to convert it into a version expressed in a higher level language, or

(b) incidentally in the course of so converting the program, to copy it, (that is, to "decompile" it), provided that the conditions in subsection (2) are met.

(2) The conditions are that—

(a) it is necessary to decompile the program to obtain the information necessary to create an independent program which can be operated with the program decompiled or with another program ("the permitted objective"); and

(b) the information so obtained is not used for any purpose other than the permitted objective.

(3) In particular, the conditions in subsection (2) are not met if the lawful user—

(a) has readily available to him the information necessary to achieve the permitted objective;

(b) does not confine the decompiling to such acts as are necessary to achieve the permitted objective;

(c) supplies the information obtained by the decompiling to any person to whom it is not necessary to supply it in order to achieve the permitted objective; or

(d) uses the information to create a program which is substantially similar in its expression to the program decompiled or to do any act restricted by copyright.

(4) Where an act is permitted under this section, it is irrelevant whether or not there exists any term or condition in an agreement which purports to prohibit or restrict the act (such terms being, by virtue of Section 296A, void).

50BA Observing, studying and testing of computer programs

(1) It is not an infringement of copyright for a lawful user of a copy of a computer program to observe, study or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program if he does so while performing any of the acts of loading, displaying, running, transmitting or storing the program which he is entitled to do.

(2) Where an act is permitted under this section, it is irrelevant whether or not there exists any term or condition in an agreement which purports to prohibit or restrict the act (such terms being, by virtue of Section 296A, void).]

50C Other acts permitted to lawful users.

(1) It is not an infringement of copyright for a lawful user of a copy of a computer program to copy or adapt it, provided that the copying or adapting—

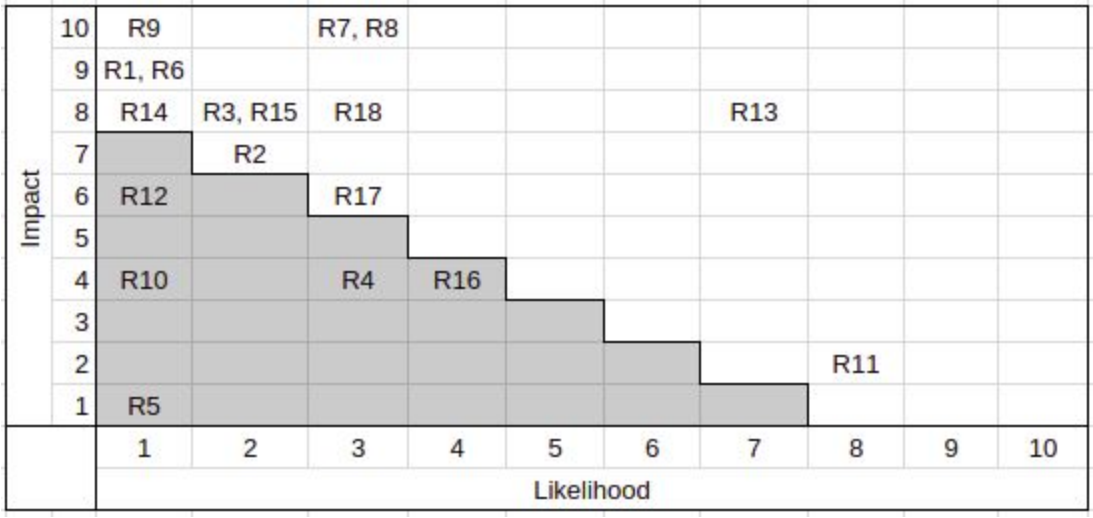
(a) is necessary for his lawful use; and

(b)is not prohibited under any term or condition of an agreement regulating the circumstances in which his use is lawful.

(2) It may, in particular, be necessary for the lawful use of a computer program to copy it or adapt it for the purpose of correcting errors in it.

(3) This Section does not apply to any copying or adapting permitted under [section 50A, 50B or 50BA].]

APPENDIX I: RISK MATRIX DIAGRAM



RISK:

- R1 - Loss of 'Team member'
- R2 - Member Illness (in which they are unable to work)
- R3 - Dispute within 'Team'
- R4 - Member had personal / family issues (in which they are unable to work)
- R5- Damage to current hardware
- R6 - Loss / Corruption of Data
- R7 - Level of Python comprehension is inadequate
- R8- Modules incompatible with framework design
- R9- Acts of God
- R10- Legal Issues
- R11- Software issues requiring additional research
- R12 - Changes to scope (by client)
- R13 - Over estimated time to completion
- R14 - Communication breakdown
- R15 - Development technically too difficult
- R16 - Feature creep
- R17 - Real time performance issues
- R18 - Modules deemed to be too difficult to execute in given timescale