

Networking - AI3 & AI4 Assignment Reflection Questions

Deakin Fisher

Q1 - Design Sketches in collaboration with System Admin

Due to my circumstances a thorough understanding of concepts was needed. To achieve this I sketched the important concepts/processes that I would integrate into my project.

My first Designs, I brainstormed options, and mapped out two realistic options:

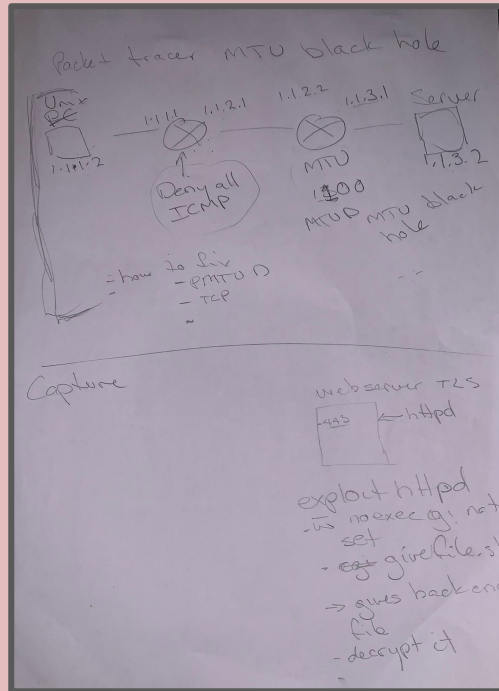
- The first was creating an MTU black-hole using Packet tracer
- The second was exploiting a vulnerability in a web server

After mapping out both, I came to the decision of the second option, due to my prior knowledge in the field, and the ease of implementation.

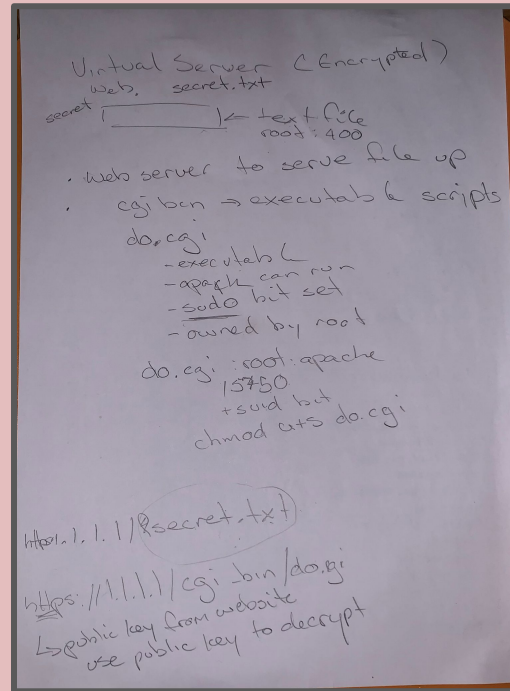
- Through the process, I consulted with a system administrator to help with real-world application and any technical tips.

Iterative Design Process (Initial - Final)

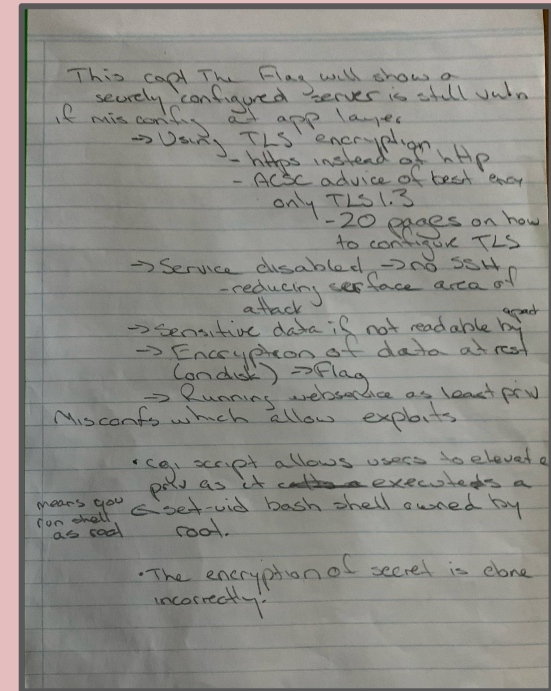
Initial Designs



Emerging Designs



Finalised Design



Further Designing

Plan

- This shows concepts on IT security
 - Information gathering
 - Exploiting a vulnerability
 - the Public/Private key

We will use a capture the flag to demo this exploit. This will involve the user:

1. Examining HTTP headers to find a possible exploit
2. Calling a misconfigured cgi-script to elevate their privs to get the flag
3. Decrypting the flag using the web server's public key

This capture the flag scenario is an example of "information leakage". This is where an ~~adversary~~ adversary obtains info they are not entitled to. This info can range from:

- what types of servers are ~~host~~ ^{in place} the website, which will then help in the next attack
- User data: e.g. Optus leak which was a simple test website not protected adequately, ext avail

HTTP → Know vulns e.g. Cross-Site Scripting (XSS), frames

Edit

/etc/httpd.conf / httpd.conf
to enable https & cgi

ps

ps

ps -ef | grep Process looking for

Shows running process

All process in full listing

Apache

process

apache

Httpd ~~server~~ is running as ~~apache user~~ ^{apache}

(Apache, root and sometimes httpd is common name)

→ When you run a process as someone

Server details

IP 192.168.64.4/24

fedora is in the wheel group
generate key pair & cert

- installed openssl

" modssl

urlencode

- modified /etc/httpd/conf.d/ssl.conf

- added the cert & key

- added server name

- made a do.cgi

- prints out a file if given name

- make a file for secret

- owned by root

- readable only by root
chmod 600

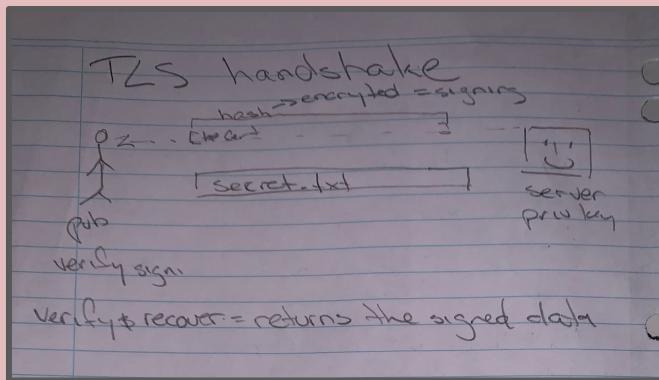
service

- Add https, to Fedora Workstation firewall zone

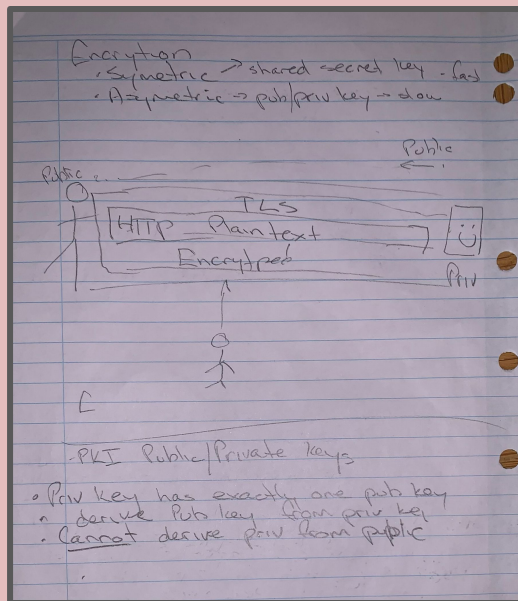
Development of Understanding through Sketches

Developing understanding through sketches was a good way for me to reinforce my knowledge, and learn new important information quickly.

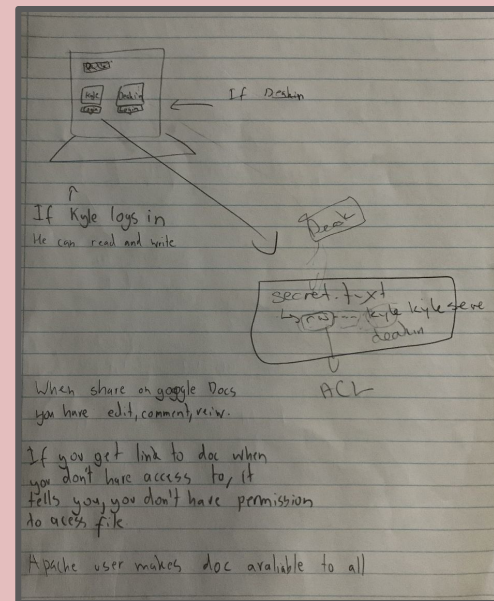
TLS handshakes



Pub/Priv Key Encryption



Apache and Root users



Q2 - Network Vulnerabilities/Exploits

My capture the flag challenge, is used to show the following concepts in IT security


- Information gathering
- Exploiting a vulnerability
- Public/Private Key Misconfiguration

The next slides discuss how this is done.

Http Headers

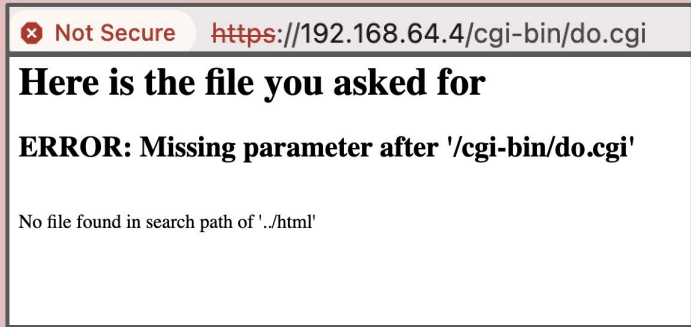
HTTP headers can store valuable information, which can sometimes be misconfigured to store critical information that should have been secured. In this case, the challenge shows two instances of misconfigured HTTP response headers, which hold valuable information.

Accept-Ranges	bytes
Connection	Keep-Alive
Content-Length	2048
Content-Type	text/html; charset=UTF-8
Date	Fri, 20 Jun 2025 12:33:09 GMT
Etag	"800- 6380004e7e76c"
Keep-Alive	timeout=5, max=100
Last-Modified	Fri, 20 Jun 2025 12:29:53 GMT
Server	Apache/2.4.59 (Fedora Linux) OpenSSL/3.0.9
X-Companyurlcompliance	Strict/XML
X-Fileretrieveendpoint	/cgi-bin/do.cgi
X-Uriacl	Not For Bob

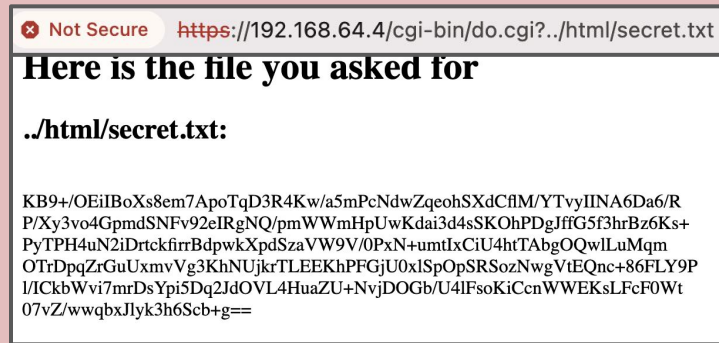
▼ Response Headers <input type="checkbox"/> Raw	
Connection	Keep-Alive 
Content-Type	text/html; charset=UTF-8
Date	Fri, 20 Jun 2025 12:39:50 GMT
Keep-Alive	timeout=5, max=100
Server	Apache/2.4.59 (Fedora Linux) OpenSSL/3.0.9
Transfer-Encoding	chunked
X-Companyurlcompliance	Strict/XML
X-Encryption	WebServer Private Key
X-Fileretrieveendpoint	/cgi-bin/do.cgi
X-Uriacl	Not For Bob

CGI exploits

By calling the CGI script, my challenge allows users to elevate their privileges, as it executes a set-uid bash shell owned by root. This means that the user is able to run the bash shell as root. My challenge shows insecure information extracted due to this exploit.



After ?../html/secret.txt



Private key vulnerability

When data is encrypted with a private key, it presents a red-herring of security. This is because any private key is able to be encrypted with a public key, which are given out by the webserver. This exploit highlights a critical vulnerability within a web-server that is likely done due to human error.

In my challenge during reconnaissance, the user will notice the private key is vulnerable:

X-Encryption	WebServer Private Key
X-Fileretrieveendpoint	/cgi-bin/do.cgi

Here is the file you asked for

./html/secret.txt:

```
KB9+/OEilBoXs8em7ApoTqD3R4Kw/a5mPcNdwZqeoHSXdCfIM/YTYvIINA6Da6/R
P/Xy3vo4GpmdSNFv92elRgNQ/pmWWmHpUwKdai3d4sSKOhPDgfG5f3hrBz6Ks+
PyTPH4uN2lDrtekfmrBdpwkXpdSzaVW9V/0PxN+umtIxCIU4htTAbgOQwILuMqgm
OTrDpqZrGuUxmvVg3KhNUjkrTLEEkHPFGjU0xlSpOpSRSozNwgViEQnc+86FLY9P
lICxbWvi7mrDsYpiSDq2ldOVL4HuaZU+NvjDOGbU4lFsoKiCcnWWEKslFcFOWt
07vZ/wwqbXJly3h6Scb+g==
```

They can download the public key, and then use Openssl tools to decrypt and recover the insecure data:

deakinfisher@Deakins-MacBook-Air ~ % openssl s_client -connect 192.168.64.4:443 < /dev/null 2>/dev/null openssl x509 -pubkey -nocert > securum.pub
deakinfisher@Deakins-MacBook-Air ~ % cat securum.pub
-----BEGIN PUBLIC KEY----- MIIBIjANBgqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAlSB0Rko9z1+g6iIehOjL dU0uCqXhj2omaKoF+HSSFug5jZvJnBunCveJ+WMKESLn6xg8DHrov/MoZi7C8fJk GdmHmo4evu85ap/NT1LL5ji2qviiNKJ13p4xxB3qi2s5MySuYOjJhe10wN12sIBg yMk/13q0Xp+AqcPTwZi193sFnk+4I3J1FwNLSwzMST+W7vgEChyqpjLFvGwur5nF gupSztarDJKSRG0EVnLRqPjN+HVhWdg992Y9xZ/A96h5jKwprLHbH5nVX8ijtsp f3VGiJIF6/Q2XnN4gKHojbK4ZHS8aDbqAuuP85mifJV5fNhJYQ4S07WgSRemLn8t UwIDAQAB
-----END PUBLIC KEY-----
deakinfisher@Deakins-MacBook-Air ~ % openssl enc -d -a -in download.txt openssl pkeyutl -verifyrecover -inkey securum.pub -pubin
Super Secret File

Comparison chart - Concepts vs Execution

Concept Shown	Simplification Made
Creating an TLS web service	Used self signed certificate instead of commercial certificate and chain. This was because the identity verification component of the certificate was not the focus of the exercise, examining the public/private key misconfiguration was.
CGI exploit	Very simple CGI exploit script instead of using a more complex exploit. This was because the demonstration needed to be easy to implement.

Throughout the design process and execution of the design, it was imperative to maintain a balance between the realism of the challenge, whilst making it accessible for others. Sometimes, sacrifices would have to be made to maintain this balance.

Q3 - Real World Application - Information Leakage

This capture the flag scenario is an example of 'information leakage'. Information leakage is defined by an adversary or malicious party obtaining information they are not entitled to. In the cyberspace, this information includes;

What types of servers host a website (which will help in future attacks)

Confidential user data/information

Vulnerabilities of a Server that can be exploited



Case Study - Optus Leaks

The optus leaks are a real-world scenario in which user data was leaked due to improper security of an API. The leak demonstrated weakness in the process, using in a test environment, production data without the obfuscation of users' details.

Optus data breach class action launched for millions of Australians caught up in cyber attack

By Ben Knight and staff

Information Technology Industry

Fri 21 Apr 2023

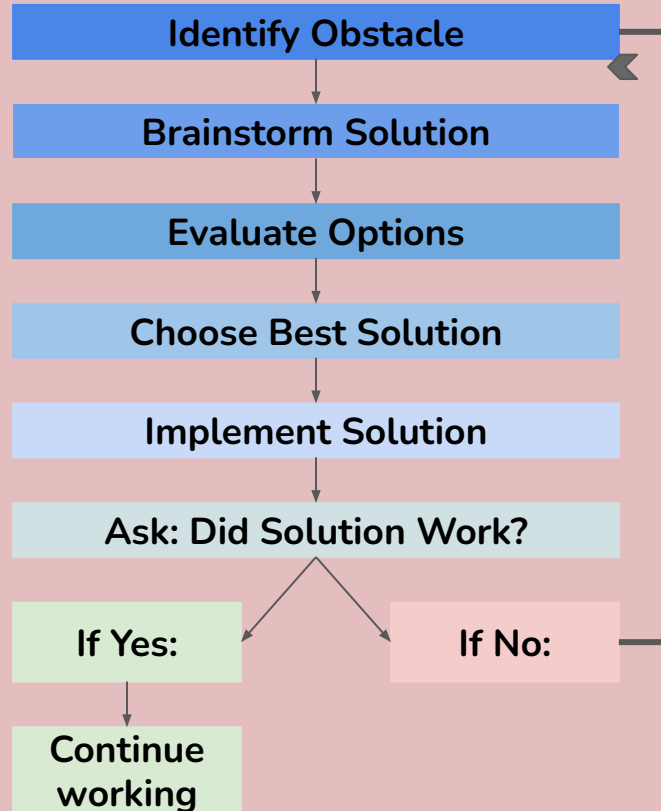


The Optus hack exposed the data of almost 10 million Australians. (AAP/Bianca De Marchi)

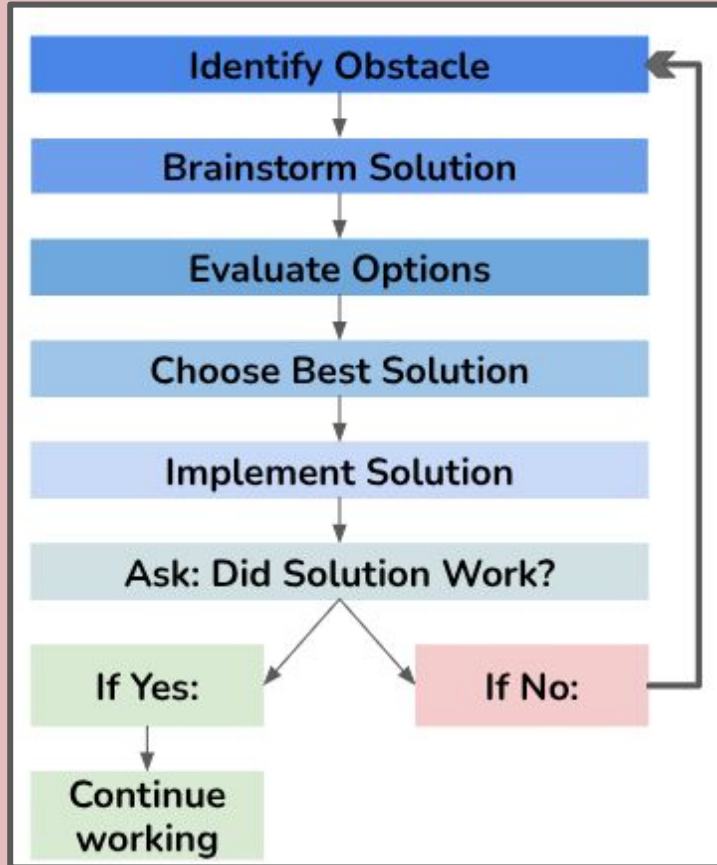
Q4 -Troubleshooting and Problem Solving

Model to overcome challenges

This is a self-made model which I will apply to an challenges/obstacles, to effectively troubleshoot the issue and resolve it

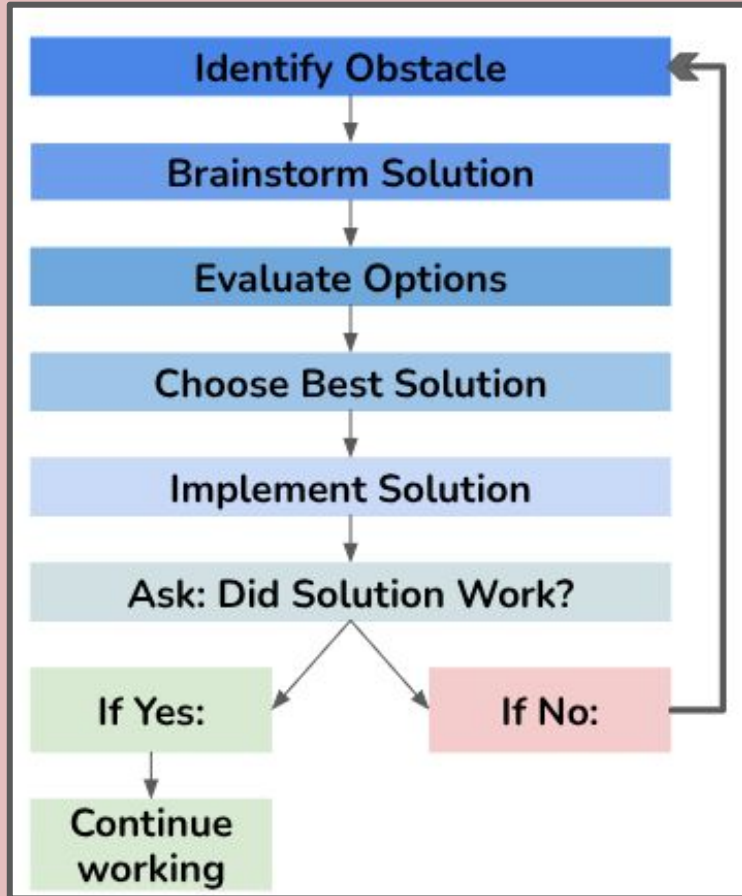


Problem 1- Non-compliance of do-CGI



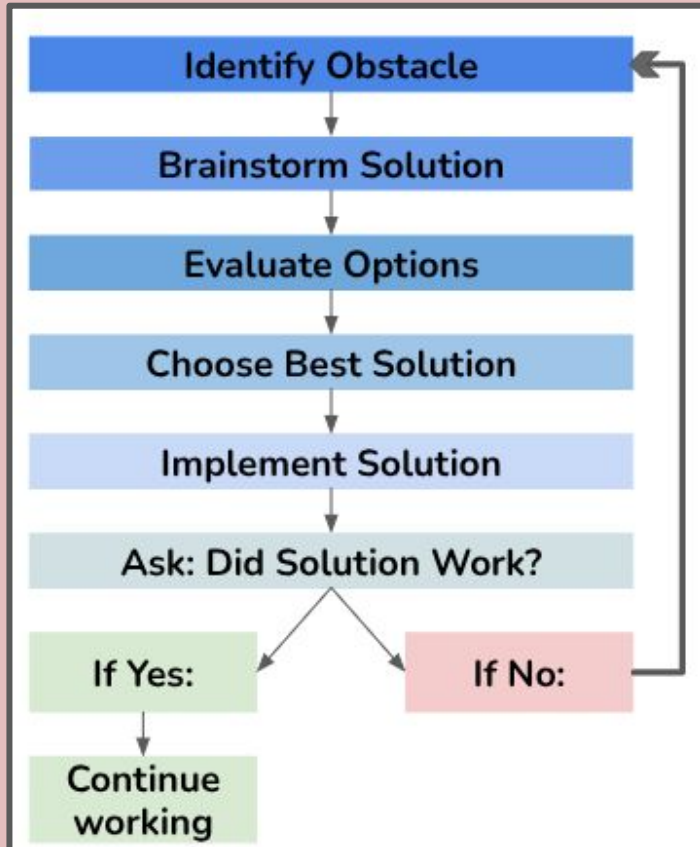
- **Obstacle:** I could not get do-CGI to elevate its' privileges
- **Brainstorm/Evaluation:** It Uses a /bin/bash shell in script in do-CGI. Googled 'why does my cgi bash script not work'. Answer said server stops bash running setuid script to prevent this type of exploitation. A possible solution was to copy bash shell to another name to get around this. Another was to call the setuid script from another non-setuid script
- **Chosen Solution:** Followed first answer, and made a setuid version of bash called bash-insecure and used this in the do.cgi script.
- **Outcome:** Was successful in fixing the problem and continued working.

Problem 2 - Unable to Encrypt on Private Key



- **Obstacle:** When trying to encrypt my file 'secret.txt' with the private key, the tools I was using would not let me do this. OpenSSL tried to encrypt secret.txt with private key but tools would not let me as it can be accessed with the public key.
- **Brainstorm/Evaluation:** I had googled searched, and had learnt about signing. I learnt that I shouldn't use a hashing algorithm at all, but instead use a different openssl pkeyutl command flag "verifyandrecover".
- **Chosen Solution:** changed command flags to pkeyutl, and swapped out hashing algorithm and used verifyandrecover option.
- **Outcome:** I was then successful in encrypting my file with the private key despite the obstacles and continued working.

Problem 3 - Binary File Complexity Issues



- **Obstacle:** I couldn't serve out the encoded file as binary, as it was too hard for the user to work out.
- **Brainstorm/Evaluation:** I read suggestions that if it was a binary file and it had to be downloaded or convert it on the server to a text file using an encoding algorithm like base64 or UUencoding so it will present as a string rather than something unreadable.
- **Chosen Solution:** Decided to use string as the task was already hard to do, and binary file would not have made it immediately obvious what to do with it.
- **Outcome:** Overcome challenge, and was able to balance realism with difficulty

Q5 - Contribution and Management of the Project - Introduction

Due to my unique circumstances, I had to complete the project with a significantly shorter time frame than my peers.

I also had to complete the assignment on my own, with no group collaboration.

Contribution and Management of the Project

Day	Milestone	Tasks	Evidence to Capture
Day 1	Planning & Setup	<ul style="list-style-type: none"> - Read VM challenge brief - Draft initial file discovery strategy -Consult with system administrators and let them help design initial processes and CTF processes. -What real world application can the system admin give which I can apply into my CTF 	<ul style="list-style-type: none"> - Outline creation of Virtual Machine using fedora - Sketch initial plans
Day 2	Initial Scripting & Testing	<ul style="list-style-type: none"> - Download and install UTM for mac and run fedora as VM -Begin writing scripts 	<ul style="list-style-type: none"> - Code version 1 (commented) - Screenshot of command-line test results - Notes on bugs/misfires
Day 3	Continue scripts	Continue on scripts, making sure to stick to the guidelines given the time frame	<ul style="list-style-type: none"> - Save bash scripts, make sure to write down any and all initial problems
Day 4	Secrets File	- complete the secrets file and make sure that the script runs it adequately and is able to generate unique checksum each	<ul style="list-style-type: none"> - Possible table comparing secure/insecure configuration - Design process, save bash script, begin

Due to my short timeframe, creating a concise and ultra-specific plan, helped me integrate my ideas effectively and create my project within the given time frame.

On either side, are my schedules and to-do list which outline the tasks that need to be completed, and the evidence that I needed to systematically capture in order maximise the efficiency of how I worked. By doing this, and further compiling it into a github cookbook, it allowed for me to easily complete sections of the assignment such as the slides (problem-solving and overcoming obstacles), due to my documentation of the process.

Due to this rigid and specific scheduling, my comparison of my original goals to my final outcome, remained extremely similar, and somewhat surpassed my expectations:

Project Aim

To simulate a real-world example of exploiting vulnerabilities within a webserver, using a range of OpenSSL tools, to decrypt and extract secure information. It aims to simulate a multi-step process, demonstrating skills that are both grounded in realism, and within the scope of possibility at a college level.



My challenge is a capture-the-flag-like task that combines:

common authentication flaws
misconfigurations with insecure web-services
Multi-layer encoding and decoding
File analysis and weak permissions

to immerse the user in an example of real-life web service vulnerability exploitation.

		time	mental notes for walkthrough (Day 7).
Day 5	Real-World Mapping & Research	<ul style="list-style-type: none"> - Find 1–2 real incidents of information leakage - Connect challenge to enterprise and ask system administrator -Create model to overcome obstacles for slides (reflection question) - Compile list of all problems 	<ul style="list-style-type: none"> - Diagram or markdown linking your challenge to real systems - Any notes made when consulting with the system administrator.
Day 6	Documentation & Evidence	<ul style="list-style-type: none"> - begin and complete markdown doc, and information for challenge in cookbook - Create general script for voiceovers/walkthroughs of the code for tomorrow(MAKE SURE TO KEEP THEM) - Show each obstacle/challenge during the course of scripting and outline it with adherence to the previously stated model. 	<ul style="list-style-type: none"> - Markdown file with structure: Introduction, Tools, Execution, Reflections - Screenshots, diagrams, comments
Day 7	Create Videos Finalise Everything	<ul style="list-style-type: none"> - Record short voiceover reflecting on process of creating secret flag using script - Record Voiceover of the creation of the bash scripts. 	<ul style="list-style-type: none"> - Deployment video file - 5–6 reflection slides for class or submission - Finalise slides

Self Assessment

Strengths	Weaknesses
<ul style="list-style-type: none">- Managed time well, given short time frame- Maintained strict adherence to my plan- Flexible in learning new Ideas- Adapted to problems well	<ul style="list-style-type: none">- Limited knowledge of content meant longer time spent learning- Over-complication of code meant going down 'rabbit holes'- Set goals that though achieved, have negative effects afterwards- Incorporation of AI3 could have been better