Threat Modelling (Part 1)

1) Assumptions

The first assumption is that the NCA system will [redacted]

The second assumption is the use of an easy way for owners to [redacted] This could be in the form of a mobile app, [redacted] to a database [redacted]

The third assumption is the [redacted] This includes the detection of breaches and fraud – such as ensuring all the necessary payment transactions are made.

The fourth assumption is the use of [redacted] As only certain vehicles can pay to enter the zone, [redacted]

The last assumption is the [redacted]

2) Attack Tree

See attached attack tree (separate PDF).

3) Risk Assessment

The [redacted] model will be followed for a risk assessment procedure, in order to analyse two major threats that endanger the NCA system, as taken from [1, 2]. The [redacted] model is a framework which identifies and categorises security threats; it is an acronym for [redacted] which are each specific categories of threats. This is a type of software focussed threat modelling, focussing on the systems in use.

The risk assessment procedure will give a description of the attack, categorise the attack, identify the vulnerability and countermeasures, assess the likelihood of such an attack occurring and assess the impact of this type of attack. This risk assessment procedure follows the specification from [3].

Design assumption of the NCA system include the fact that the system processes sensitive personal and financial information which could be targeted. Additionally, general assumptions have been made considering the current security in place and the impact of threats under current security standards. Finally, it is assumed that the NCA system is a large system that is likely to attract attackers.

The two threats that this risk assessment will consider are distributed denial of service attacks (DDoS) and data breaches.

Threat: DDoS

Description: A DDoS attack is a malicious attack which overwhelms a target through increasing the normal amount of traffic [4]. This flood of requests can overload a system and prevent it from working or being accessible. Regarding the NCA system, [redacted]

Categorisation: Denial of Service

Vulnerability: The system is unable to [redacted] which must be monitored, and is susceptible to DDoS attack vectors.

Countermeasures: Many layers of protection need to be put in place to prevent the requests from overwhelming the system. Hence, firewalls and other intrusion prevention systems can be implemented. Traffic could also be spread out, across a setup with multiple servers, with load balancers. Additionally, network monitoring and intrusion detection systems should be implemented into the NCA system to better handle the attacks as soon as they occur. These solutions should be regularly assessed and updated.

Likelihood: Likely. As these attacks are now more common.

Impact: Medium. Downtime may occur for a long period of time, [redacted] However appropriate precautions can be put in place to plan ahead for DDoS attacks.

Threat: Data Breach

Description: A data breach is where the confidentiality, availability or integrity of data is affected. Regarding the NCA system, [redacted]

Categorisation: [redacted]

Vulnerability: Sensitive information is stored insecurely, allowing it to be more easily accessible than it should be. There may be [redacted]

Countermeasures: Access controls should be accurately implemented [redacted] Additionally, secure data protection methods should be in place, such as the encryption of data. However, if there was a breach to the NCA system, [redacted]

Likelihood: Very Likely. As attackers are interested when there is sensitive information.

Impact: Bad. There could be many repercussions, [redacted]

References:
[1] Microsoft (2022). *Threats - Microsoft Threat Modeling Tool – Azure*. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats.
[2] Gadouleau, M (2023). *Threat Modelling (Approaches Overview)*, COMP3656: Security Engineering. Durham University.
[3] Gadouleau, M (2023). *Threat Modelling (Risk Modelling)*, COMP3656: Security Engineering. Durham University.
[4] Cloudflare (2022). *What is a DDoS attack?* [online] Cloudflare. Available at: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/.

Certificate Authority and PKI (Part 2)

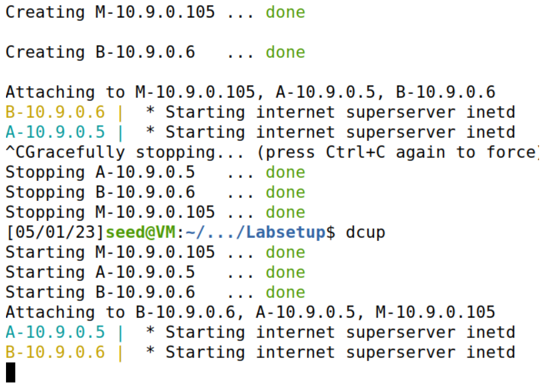
1) PKI Infrastructure

See certificates / public keys

2) Man in the Middle

See the attached code

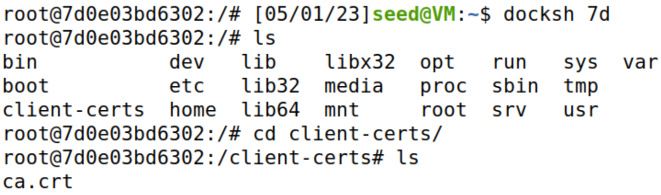
For a successful man-in-the-middle attack, the ca.crt certificate was saved to the client side (system A) in a folder “client-certs” and the server.crt and server.key were saved to the server side (system B).



```
Creating M-10.9.0.105 ... done

Creating B-10.9.0.6   ... done

Attaching to M-10.9.0.105, A-10.9.0.5, B-10.9.0.6
B-10.9.0.6 |  * Starting internet superserver inetd
A-10.9.0.5 |  * Starting internet superserver inetd
^CGracefully stopping... (press Ctrl+C again to force
Stopping A-10.9.0.5   ... done
Stopping B-10.9.0.6   ... done
Stopping M-10.9.0.105 ... done
[05/01/23]seed@VM:~/.../Labsetup$ dcup
Starting M-10.9.0.105 ... done
Starting A-10.9.0.5   ... done
Starting B-10.9.0.6   ... done
Attaching to B-10.9.0.6, A-10.9.0.5, M-10.9.0.105
A-10.9.0.5 |  * Starting internet superserver inetd
B-10.9.0.6 |  * Starting internet superserver inetd
▌
```



```
root@7d0e03bd6302:/# [05/01/23]seed@VM:~$ docksh 7d
root@7d0e03bd6302:/# ls
bin           dev   lib    libx32  opt   run   sys  var
boot          etc   lib32  media   proc  sbin  tmp
client-certs  home  lib64  mnt     root  srv   usr
root@7d0e03bd6302:/# cd client-certs/
root@7d0e03bd6302:/client-certs# ls
ca.crt
```

A socket can be created on the server side which waits for a client’s connection, allowing the exchange of encrypted data securely.

[redacted]

File Integrity (Part 3)

1) File Integrity Code

See the attached code

2) File Integrity Report

Results
The results from the code created show that "message.txt" [redacted]
"messageTwo.txt" [redacted]

Explaining the Code
To carry this out, the package rsa was used, which is a package that allows the implementation of encryption and decryption methods and various other security tasks including the verification of digital signatures. rsa is used here to verify the digital signatures of "message.txt" and "messageTwo.txt" using key cryptography – this is performed through using the RSA decryption algorithm.

Reference to stored files, for analysis, are made. A function is then defined to read in files, however the contents of the messages are not analysed.

The function to analyse the signatures of the messages is defined. This reads in the files. The start of the key data is amended to ensure that it is read by the RSA algorithm correctly. The public key data is then decoded. As described before, the RSA algorithm is then used for authentication to check that the signature is consistent.

This function is then run for both files giving the results above.

Why this Method was Selected
The RSA algorithm generates a public and private key. A file is then signed with a private key to make a digital signature. The public key is then used to verify that the file has not been tampered with. If the signature does not match then the message has been tampered with.

Checking the files this way is better than checking the content of the files. This is since checking the digital signature is more secure and is distinct and unique to file content. The content may stay the same, however a file may have been tampered with however there will always be a different hash value in this case.