Ehsan Toreini
COMP3657
ehsan.toreini@durham.ac.uk

Coursework Assignment
Security Engineering - 2022

deadline: 05, May, 2022

1 Summary of the coursework

In this coursework, you will finish three tasks. In all three, you will be acting in different roles related to an online file sharing service (like Dropbox). This file sharing service (lets call it as ACME Clouds in the rest of this document) is recently founded and considers itself as a potential challenger of the mainstream file sharing services such as OneDrive, Dropbox and Google Drive. Of course, the founders of ACME Clouds are aware they do not have the technical capacity to compete with tech giants such as Google and Microsoft; instead, they claim they guarantee an end–to–end verifiable, privacy preserving and secure file sharing experience to its clients. This way, the customers can be sure their files are stored in the clouds encrypted and they are the sole owner of the secret keys to decrypt the files (fulfilling privacy–preserving property). Also, they can track the storage of their files in the systems and verify its availability, integrity and confidentiality (fulfilling the end–to–end verifiability claims). Please follow the tasks below to ensure the promised bottom–line specifications of the company.
The deadline for submission is on May 5th, 2022. You will submit all the deliverables in a single compressed file (preferably .zip format) through Blackboard. Please be aware that this is an individual submission, so you should not work in groups for the coursework. Any sort of plagiarism and joint submission will be reported. If there are any queries regarding this coursework, please do not hesitate to contact me: ehsan.toreini@durham.ac.uk
Zero Policy: this is an experimental lab to get yourself familiarised with cyber attacks. You are not supposed to perform anything you learned in these tasks in an environment outside the experimental lab setting. This includes your PC, your friends, and any other device in the university CAMPUS or outside. If you do so, you are responsible for the consequences. If you have any questions (or concerns) regarding this, please do not hesitate to email me.

1.1 Task 1: Threat Modelling the ACME Clouds [20 Marks]

In this task, you will act as the security architect of the ACME Clouds. As the winning point of the company relies on security and privacy promises, your role is a essential part of the success of the company. You are tasked to have a comprehensive security analysis of the main competitors of the company and design a secure product. Some features of the cloud storage service are as follows:

- It has iOS, Android and Windows apps for uploading and syncing the local and cloud files
- You can upload files through your web browser as well as the developed apps
- As it is a trial version, the system has “only–invitation” membership plan. Also, the current members can invite three more people to use the systems
- For usability purposes, the system designers decided to use cloud–based NoSQL database services such as MongoDB for their databases
- The server will use cutting edge cryptographic primitives and algorithms to ensure the overall system remains responsive, despite the heavy computations for cryptographic proofs.

You should prepare these items for this task:

1. Make at least 5 further assumptions about the system, together with their justification (limit 400 words).
2. Prepare the attack tree of ACME Clouds. Include at least 15 nodes in the tree. These nodes should be both general threats (such as protocol failure, wiretapping and alike) and scenario–specific ones (such as social engineering emails and insider threats). Submit the tree in 1 page PDF (it is alright the page size is bigger than A4). Make sure the text in the file is readable and in high resolution.
3. Prepare a risk assessment on the two major threats that will endanger the ACME Clouds. Explain the risk assessment procedure and your findings in your research and provide your countermeasures. Remember, you need to provide some design assumptions for your assessment. These assumptions should be aligned to the design choices explained above and your own research on how cloud storage systems

and technologies work. For instance, you can use the threat and vulnerability databases such as NIST National Vulnerability Database[1] for your analysis). The style of the analysis should be technical, rather than verbose. This should be understandable by someone with a good knowledge of the security of the system. Be concise and straight to the point. Make sure your answer to these questions do not exceed 2 A4 pages, including the citations.

Note: Combine the PDF files to gether and submit your answers for task 1 in a single PDF file named as 'Task 1.pdf'.
Note: remember to properly cite the claims in your assumptions, threat tree and risk assessment.

1.2 Task 2 - White Hacker [40 Marks]

In this task, you are hired as a white hacker. In the first day of your job, you want to assess if the client–app is vulnerable against Man–in–the–Middle attacks. Thus, you implement a ARP poisoning attack to check the network packets. In this attack, you exploit the ARP packets to poison the ARP tables in the client computer so all packets will be redirected to your computer instead of the ACME server. This is a powerful Man-in-the-Middle attack in which you practically have full power over unencrypted network traffic, i.e. you can choose to *sniff* or *spoof* the packets.
You will use the *"scapy"* (a Python package) to implement your attacks. Use the attached python code template as the reference for your implementation (named mitm.py); however, you can change the template in your preferred implementation. Implement the attack on host-M (as it is assumed to be the attacker's device).
Store these packets in separate pcap files named as the name of the task and step (for example, for the first step, file name will be "Task 2 - Step 1.pcap").

1. Capture the first 100 network packets transferred in the communication.
2. Capture all network packets containing plain text password and username.
3. Capture all HTTP packets (responses) containing images.
4. Capture and modify all Telnet packets and replace each typed character in the communication to 'R'.

Note: more detail on ARP poisoning attack is found here: https://www.tutorialspoint.com/ethical_hacking/ethical_hacking_arp_poisoning.htm
Note: Please use the virtual machine and containers available in the coursework folder on Blackboard Ultra. The instructions on how to set–up your containers are available to you in a separate file and the IPs are already set for your system. Use the same VM as your at-home practical.

1.3 Task 3 - PKI infrastructure [40 Marks]

In task 3, you will create a PKI infrastructure for ACMEclouds. ACME Cloud company wants to get a public key certificate from our CA. You are responsible to get that certificate and verify if it works well. For simplicity, you create digital certificates without going to pay any commercial CA. You should become a root CA yourself, and then use this CA to issue certificate for anyone (including ACMEclouds servers).
You are also allowed to register the certificate in a combination including your own name. Therefore, the name used in the ACMEClouds server certificate must contain *ACME Clouds, your last name and the current year*. The registered URL will be "www.acmeclouds.com".
Name the CA's public-key certificate and private key as "ca.crt" and "ca.key". Also, the server's public-key certificate and private key should be named as "server.crt" and "server.key".
After you have generated your own certificate authority and the certificates for server, you will be implementing a secure channel between server and client (in presence of a powerful Man-in-the-middle). You should use system A as client, and system B as server. Store the CA's certificate in the "./client-certs"folder on the client device (A) and use it for your handshake requests. Use the python packages*"socket"* and *"ssl"* for your implementation (other packages are not allowed to be used).
Note: You can use openSSL (https://www.openssl.org/) to generate the certificates. In order to use OpenSSL to create certificates, you should have a configuration file. The configuration file usually has an extension.cnf. It is used by three OpenSSL commands: ca, req and x509. The manual page of openssl.conf can be found from online resources. By default, OpenSSL use the configuration file from /usr/lib/ssl/openssl.cnf. If you need to make changes to this file, you will copy it into our implementation directory, and instruct OpenSSL to use this copy instead. Include the configuration file in your submission. Also, include the openSSL command you used for generation of CA and server certificates in a single PDF file.

[1] https://nvd.nist.gov/

Note: You need to generate a self-signed certificate for our CA. This means that this CA is totally trusted, and its certificate will serve as the root certificate. Use the following configurations for your CA: RSA uses 4096 key length, use sha256 for hashing and certificates will be valid for 3650 days (10 years). If required, you can set the CA password as 'dees'.

1. write all the openSSL commands used for generation of the certificate in a PDF file attached to your submission. Briefly explain the purpose of the command and what happens if you do not execute them.
2. Implement a simple TLS server on host-B in a file named as server.py. Use the certificates that you generated before.(you are allowed to use python packages for SSL, i.e. ssl and socket packages)
3. Implement a handshake request from client on host-A to the TLS server (name the file as client.py)
4. Transfer an encrypted message between host-A and host-B (include the code in client.py)
5. Capture the TCP data transmission between host-A and host-B by the Man-in-the-middle (host-M). The captured (encrypted) message should be stored in another pcap file name "Task 3 - encrypted.pcap"

2 Timeline and Project Marking Guidelines

2.1 Project Timeline

The deadline for submission is on May 5th, 2022. You will submit the three tasks in one single compressed file (preferably in zip format). Use your student ID as the name of the zip file. The threat tree and risk assessment should be in pdf file format. Your final project submission should have the following items:

- Task 1 in PDF file format
- Task 2 and Task 3 in a separate folder, include the python source codes (mitm.py, client.py, server.py), the openSSL configuration file, the openSSL commands PDF file and "ca.crt", "ca.key", "server.crt" and "server.key"

2.2 Marking Guideline

Threat Tree. [20 marks]
You submitted essay will be marked based on the quality of writing, relevance of the citations, depth of discussions on the topics above and the overall correctness, fluency, clarity and delivery.

Criteria/ Level	Absent	Weak	Medium	Good	Excellent
Structure	No explicit structure on most parts i.e. (description, likelihood, impact, vulnerability)	Identification of a few elements of the expected structure.	Structure containing most elements, with some being slightly wrong.	All elements are given and appropriate.	Structure is given following a professional style, with reference to standards and databases (e.g. CVE).
Fit to scenario	General answer not taking into account the actual scenario	A hint to the scenario	Partial fit, adapted to the scenario, but with no explicit originality	A mix of partially fit and some originality	Original answer, using the scenario very well
Risk assessment	No actual assessment (no likelihood, no impact)	Assessment given on likelihood and/or impact but justification is missing	Assessment and proper justification is given on one either likelihood or impact	Assessment and proper justification is given on one both likelihood and impact	Assessment given and motivated by the example
Clarity and formatting	No emphasis, hard to read.	Some form of structure, but hard to follow	Partial emphasis and structuring	proper sectioning for each requirement and clear solution	Very clear solution, with a good emphasis on the most important points, use of underlined, bold, etc.
Threat tree	No tree	A tree without sufficient nodes	A coherent tree with at least 15 nodes, covering only general threats/attacks	A coherent tree with at least 15 nodes, covering both general and scenario-specific threats/attacks	A clear tree with proper categorisations of threats and attacks in different layers and meaningful connections between the nodes

Implementation. [80 marks]
The code and project proposal will be marked as follows:

- Implementation
  - Does your project work?
  - Correct implementation of the ARP poisoning attack
  - Correct chain of openSSL commands
  - Correct implementation of the PKI infrastructure.
  - Correct implementation of a TLS communication
  - Correct commands for generation of server certificate
  - Correct pcap output
- Sophistication and appropriateness of the solution
  - How well have you applied the relevant theory to the problem?
  - How hackish is your implementation, or is it robust and well-designed?
  - Have you just cited and pasted code, or is their evidence of comprehension with further study and novel design extending beyond the lecture materials?

3 Frequently Asked Questions

It is strongly recommended to read these common questions and answers carefully.
I found code online which looks similar to what I need. Can I use it?
Yes, but you must cite the code in both the written report and in the comments at the top of the code. As a common practice in any software development, you first try to search and make sure you are not the first one who is trying to make it work. However, it is one of my tasks to make sure you are doing something original. So, please adapt the code and make sure you have cited it. Otherwise, it is very likely that you get caught (see the sub-mission Plagiarism and Collusion section on DUO to read about the tools used to detect this). This incurs a very severe departmental penalty.
Isn't the best strategy to just copy the state-of-the-art? Yes, if you notice from the list of suggested projects, you are already working on the state–of–the–art solutions in fair AI. If you know the literature, you will find the most reputable research in the field of fairness and algorithmic bias belongs to a conference called (ACM FAT, which recently got rebranded as ACM FAccT). So, keep looking in their accepted paper list to find the most recent developments in the field.
I'm struggling and feeling overwhelmed by all of this. The tasks are too complicated and I don't know where to begin.
If you get errors, read them slowly, Google them. When you're confident enough, try to implement something a little bit more complicated. Do a slow, step–by–step implementation approach.
My writing is not as good, will it make my essay mark automatically low?
Not really, I understand for the majority of students (including myself), English is not their first language. Therefore, instead of focusing on using complicated words, try to stay as simple as you can in your writing. My first advice to everyone is to write simple. Avoid complicated sentences, words or grammatical combinations. Focus on the quality of discussions rather than making your essay look fancy. I would also recommend using online grammar check tools (such as Grammarly) to polish any mistakes. Also, you can always borrow technical words or some discussions from the papers you want to cite (just remember to cite them, then rephrase them in your wording to avoid plagiarism).
Am I allowed to use python packages other than the ones suggested in this coursework?
No. There might be a brilliant package that does exactly what the tasks suggests; however, the marking of the implementation tasks will be automatic and the code runs based on these packages. If you have any further queries or confusions, do not hesitate to contact me.
Does the page limit include references?
Short answer to this question is: Yes