Advanced Computer Networks - Second Semester 5784

Lecturers: Barak Gonen, Guy Saadon, Avi Treistman, Zvika Herst

Alternative Assessment – CTF Network

This is an alternative assessment (replacing the final exam) that counts as 50% of the final grade in the course. In this exercise, you will create your unique CTF challenge using the topics examined during the course. The more complex and advanced the challenge, the higher the grade. The CTF can be used in the future for job interviews, so investing your efforts into challenging results is worthwhile.

What are the deliverables?

1. A proposal approved by the lecturer (no score will be given, but a deficient proposal will have to be improved)
2. The starting point of the CTF as it is delivered to participants trying to solve it (such as a file, link, etc.)
3. The solution
4. A document explaining the underlying development, including any code you have used or written

Proposal for approval

Short proposal. The proposal will include the selected topics and explain how they will fit into the CTF. The proposal consists of what needs to be done to resolve the CTF. An example of a development proposal document is in the Appendix. The lecturer must approve your proposal. If necessary, you will need to improve your proposal before starting work.

The starting point of the CTF

Write a background story geared to the participants that "puts them in the mindset" of the CTF. You can incorporate solution-related clues into your story, such as a text to look for later. Within the story will be the starting point of the CTF – such as an image, file, link, etc.

The CTF should include some of the following topics:

- Socket programming (optimized server for single or multi-user clients)
- Using Wireshark
- Using SCAPY
- Access to a server created for the project on a domain you registered
- Python development in Python (consisting of at least a few dozen lines of code)
- HTTP/S protocol
- DNS protocol
- SMTP protocol
- QUIC

- TLS
- TCP or UDP, if used specially, not just as part of the Protocol stackof higher-tiered protocols.
- IP protocol, if used specially, not just as part of the Protocol Stackof protocols at higher levels.
- Encryption
- Performing a MITM vulnerability
- Topics from Operating Systems or Reversing
- Any other topic, to be approved in advance by the lecturer

Clarification regarding additional protocols in networks – Additional protocols can of course be used, but no additional points will be given for using protocols that are already in the Protocol Stack of one of the protocols mentioned. For example, if you created a CTF that includes HTTP, it already includes TCP, IP, and Ethernet.

Grading

I. Background story – 5 points
II. Complexity of the CTF – Scoring will be determined according to the number of topics selected from the list.
    a. 3 Topics – 45
    b. 4 Topics – 55
    c. 5 Topics – 65
    d. 6 topics or more – 75
III. Complete and understandable solution – 10 points
IV. Lecturer evaluation (originality, creativity, investment) – 10 points. Note- The lecturer is entitled to give additional points if exceptional work has been done.

Examples of scoring:

Shuki incorporated five themes into his CTF. Together with the background story and the solution, he is entitled to 80 points. The lecturer assessed his originality, creativity, and efforts for an additional 10 points, giving Shuki a score of 90.

Joash prepared a three-theme CTF. He received five points for the background story. For the solution, he received only five points because the explanation of how to solve one of the parts was flawed and inaccurate. The lecturer estimated Joash's efforts at five. Joash's score is 60.

Daniella created a CTF that included six themes, wrote an exciting and exact background story, and provided a good solution. The lecturer gave her the full score for her efforts and creativity. Danielle's score is 100.

Appendix – Example of a Development Proposal

Written by Barak Gonen, translated by Avi Treistman

The exercise uses seven themes: HTTP, Wireshark, DNS, socket programming, cryptography, PE format, and JSON.

Step 1 - Networks

Capture file. The file will include an HTTP capture which contains an image.

The capture will contain all sorts of hints that something interesting is on the HTTP server, such as the name of the server or the name of the resource requested from it. We will connect these names to the general background story (for example, if the story is about Snow White, then the resource's name will be poisonapple.jpg).

Wireshark has the option to export an image from an HTTP capture. This image will be the product of this step.

Step 2- Integration with Operating System Features

Image Analysis. The outcome of the previous step appears as just an image, but hidden inside of the JPEG file will be an executable file in PE format. Something in the size of the image would imply this. For example, the size will be exactly 1 MB or the visual image will be very small, but the file size will be unreasonably large.

It takes some operating system knowledge to recognize a PE format. Copying the bytes starting with the PE header into a new file and saving it as an executable will allow you to run it.

The code will be a client that will need to go through several steps on networks to solve it.

I. The program will search for a server like blabla.co.il and if it can't find it, print a suitable message. You must set up an HTTP server and link the domain name to the localhost using the *hosts* file in the system directory.
II. The program will print a message if it receives a response whose content type is not JSON.
III. The program prints a message if the information does not contain a field named Success with a True value.

After going through all the steps, the client will print encrypted text.

Step 3 - Cryptography

The text of Step 2 will be encrypted in the Vigenère cipher, and its decrypted text will contain some story along with a URL that leads to a “Success” image.