Exercises for Network Security
9. Private Communication

Emmanouil Vasilomanolakis & Carsten Baum, DTU

March 15, 2024

Exercise 1. (**What Does A Webserver Learn About You**)

In this exercise, we will see what information is leaked to a webserver whenever we access a page.

1. Identify what HTTP headers are part of your web browser's usual HTTP request. For that (if you use Chrome or Firefox) open the "Inspect" window, access a website of your choice and look at the HTTP headers your browser generates when accessing the page.
2. If you don't have it already, install an anti-tracking plugin such as Ghostery. Then access e.g. `dr.dk`, `dtu.dk` and `nytimes.com` (or your favorite websites) and check how many trackers they have and what they belong to. Can you identify which companies or websites belong to the respective trackers?

Exercise 2. (**Hiding Data In Plain Sight**)

In this exercise, we will learn how to secretly communicate data without using cryptography! For this, we rely on an ancient technique called Steganography.

1. Read up in Ross Anderson's Security Engineering book about Steganography. In particular, sections 22.4, 22.4.1, 22.4.2 of `https://www.cl.cam.ac.uk/~rja14/Papers/SEv2-c22.pdf` might be helpful.
2. Install *steghide*, a classical tool for steganography. If you use Debian or a version thereof, you can simply use apt to install it. On Windows, you can get it in WSL as easily.
3. Compare the files dtu-logo.jpg and embedded.jpg visually. Do you see differences? Then try to extract the secret message from embedded.jpg using steghide, using the passphrase "netsec".

Exercise 3. (**Secretly Getting Data Out**)

Assume you are a journalist and your friend Alice works for EvilCorp Inc. She desparately wants to give you information about their recent evil deeds, but unfortunately she cannot move data out of EvilCorp physically as USB ports are locked down etc.

1. Given the techniques you have learned so far in the course, in particular those in this lecture, devise a strategy for Alice to send the documents to you digitally.
2. Assume that EvilCorp also disallows the use of e-mail encryption and scans all e-mails or files

getting out of their company networks for company secrets. They also seem to be able to check if steganography was applied to files. Develop a strategy for how Alice could hide information in the HTTP headers when accessing a website, e.g. with the information you can find here `https://www.rfc-editor.org/rfc/rfc9110.html`.

Exercise 4. (Using Tor)

In this exercise, we are trying out the Tor system. For this, we use the so-called Tor Browser which is a bundle of Firefox and Tor that you don't have to configure yourself.

1. Download the Tor Browser and use for fun. Do you notice any difference in response behavior for sites you know? Try to download a file of reasonable size and look at download speeds.
2. You likely googled for the Tor Browser, opened the first result, then downloaded and executed the file. What kind of attacks could a malicious actor which controls the network have done so that you don't download the correct executable?
3. The Tor project allows you to verify that the downloaded file is indeed correct. Follow the instructions on `https://support.torproject.org/tbb/how-to-verify-signature/`. Which assumptions does their verification procedure make and how could it be undermined?

Exercise 5. (Signal Key Agreement)

In the lecture, we learned that the Signal Key Agreement protocol is a bit more complicated than regular Diffie-Hellman key exchange between two parties. In particular, it works as follows:

1. Alice creates an identity key pair $(pk_{IK,A}, sk_{IK,A})$, a signed key pair (pk_S, sk_S), a signature $\sigma \leftarrow Sign(pk_S, sk_{IK,A})$ as well as one-time keys $(pk_{OK,1}, sk_{OK,1}, \ldots, pk_{OK,n}, sk_{OK,n})$. She uploads $pk_{IK,A}, pk_S, \sigma, pk_{OK,1}, \ldots, pk_{OK,n}$ to Signal's server.
2. Bob also creates an identity key pair $(pk_{IK,B}, sk_{IK,B})$ and sends $pk_{IK,B}$ to the server.
3. To send a message, Bob first creates a fresh session key by downloading Alice's information from Signal, checks σ and creates an ephemeral key (pk_{EK}, sk_{EK}).
4. Then, Bob creates the session key from computing Diffie-Hellman key agreement individually on
   - $sk_{IK,B}, pk_S$
   - $sk_{EK}, pk_{IK,A}$
   - sk_{EK}, pk_S
   - $sk_{EK}, pk_{OK,1}$

   and hashes the outcomes to obtain the session key k.
5. He sends his message, encrypted under k, to Alice, together with pk_{EK}. She downloads Bob's $pk_{IK,B}$ from Signal's server and rederives k by computing DH key agreements on
   - $pk_{IK,B}, sk_S$
   - $pk_{EK}, sk_{IK,A}$
   - pk_{EK}, sk_S

- $pk_{EK}, sk_{OK,1}$

and hashing the outcome. After having decrypted the message, Alice throws away $pk_{OK,1}, sk_{OK,1}$.

We will now look into which attacks are possible if only a subset of these keys go into deriving k.

1. If k is only derived from $pk_{IK,A}, sk_{IK,B}$ what happens to the key? In particular, what if Bob tries to open multiple sessions to Alice?
2. If only the pair $sk_{EK}, pk_{IK,A}$ is used by Bob to derive the key k, what kind of attacks are possible? In particular, what does Alice know about the sender of the message?
3. If Bob derives the key using the pairs $(sk_{EK}, pk_{IK,A})$, $(sk_{IK,B}, pk_S)$ and (sk_{EK}, pk_S) while checking σ, what attacks are there? In particular, what if an attacker gets hold of $sk_{IK,A}, sk_S$ and looks at messages in the past or in the future received by Alice?
4. If Bob derives the key k from $sk_{EK}, pk_{OK,1}$ only, what attacks could the Signal server perform? In particular, what if Alice and Bob check that they have each other's correct identity key pairs $pk_{IK,A}, pk_{IK,B}$ by comparing hashes?