Azer manus presentation

There are plenty of alternatives when it comes to secure file exchange services. The most popular being Google Drive and Dropbox, which both have security features such as encryption and multi-factor authentication. These are however accessed through public domains, that is, over the internet. This means that given a malicious actor that has somehow acquired all necessary credentials, they would be able to access someone's file-sharing account and steal sensitive or proprietary information. A better alternative to those two popular websites is to use a locally hosted file-sharing server, which besides requiring credentials for the service in question, also requires a direct connection to the network that hosts said server. Thus, it introduces another layer of security when compared to Google Drive or Dropbox.

One such file-sharing self-hosted server is provided by Nextcloud, which besides file-sharing also facilitates communication in the form of chats and social networks, time- and employee management systems, and a multitude of other useful applications. As per the topology that we will soon show and introduce, it can only be accessed if a client is connected to the Stockholm network, either directly or through an indirect connection.

This server is hosted on a virtual machine in an Ubuntu OS, on a laptop belonging to an arbitrary group member. The Nextcloud server is hosted using an apache web server, which is software that provides the necessities to host and access the web page by web browsers. MariaDB, a mySQL-like framework acted as a database management system, and the web page was set up using predefined php and apt scripts. All other website functionality that I might have left out was added or created by tweaking settings from the website GUI.

For data in transit, as in files being uploaded or downloaded to and from the server respectively, Nextcloud uses TLS encryption to ensure confidentiality. Not only is the data in transit encrypted, all other communication between client and server is, through the use of HTTPS, TLS encrypted as well. Any client that tries to access the server on the HTTP port 80 will be routed to the HTTPS port, 443. Data at rest, as in what is stored on the server, can optionally be encrypted with AES. We have however chosen not to opt for encryption as we deemed it unnecessary for the scope of this course.

To run HTTPS, the server requires a valid certificate, which is generated and provided by our internal ACME certificate authority, located on the Stockholm network. As the certificate is self-signed, client web browsers will display warnings when connecting to the server. These certificates however might as well have been issued by a more trusted certificate authority, and the server would have worked all the same. We just decided that acquiring such a certificate would be too much of a hassle. Our own self signed was much easier to procure and enough for demonstrative purposes.

Another layer of security is added by having the server administrator account being the only one that has the privileges to create or add another user. Imagine that a malicious actor somehow gains access to the Stockholm network, they would still not be able to access Nextcloud without credentials, unless they have somehow stolen these as well. These credentials are only given to new employees upon request from the administrator. Upon creating an account, users are presented with a QR code that they are required to register with a mobile authenticator application, which from then on acts as a second set of credentials in a two-factor authentication system. This ensures that credentials are worthless if stolen, as you need an associated second layer of authentication to use the server. Once logged in, users will be able to download files shared by other users, as well as upload their own.

Other Security

OPN sense has its own integrated intrusion detection system, which is a security mechanism that detects and prevents unauthorized access or malicious activity on a network. Like all other IDS:s, it achieves this by searching for abnormal patterns that are indicative of a non-law-abiding-user. We overlooked other popular alternatives such as SNORT for instance as OPNSenses own IDS was easier and less time-consuming to implement.

We have unfortunately not been able to implement clogging prevention, as in a countermeasure against Distributed Denial of Service attacks, which can overload a network or server with a flood of traffic, rendering it unavailable to users. As this is something that we do not have, the availability of the network could be attacked and ultimately taken down.