**Video script**

The service I will be presenting is a network security toolkit which provides dns and network scanning tools accessible through both a GUI an an API.

Our workflow moves across navbar in the right direction as we go from submitting scan request, to viewing running scans, to managing scan reports, all done via asynchronous updates

Right now, I am going to install the dependencies and start the server.

* + Npm install
  + Npm start

**During install -** While the terminal is powerful, junior analysts usually are unable to harness it’s complete potential which is why we provide a GUI.

Our simplistic forms allow you to specify what scan parameters you want and are responsive and both client-side and server-side.

Take our network scanner for example. We pick which ever parameters we would like and all we need to do is submit. It’s that simple

We provide concurrent scan execution, so we let’s setup up scans simultaneously.

Looking at the running scans overview, We use cards to provide an intuitive way of visualizing different scans where each one is color coded to a scan type. For each card we can click on it to view more information about the scan – for example here we can see the scan parameters

**Report Management** – Once your scans have finished, the reports are presented in the same card system for simplicity.

Let’s look at this on completed scans from previous sessions

Our application is friendly to all kinds of devices including mobile. Take our report system for example, our tables and cards are responsive and furthermore we edit the text based on viewport size to maintain a clean design.

If you’re not pleased with the results of a scan, then you can always delete it and execute another one later on.

Moreover our application adheres to aria standards which allows users with disabilities to use software such as screen readers successfully with our site.

**Moving on to our API** For those with more advanced scripting knowledge our API documentation provides all the necessary knowledge that you need to interact with our system. This involves showing the different URL paths including their optional parameters, the valid format for the request, as well as many different examples. Through our API you can unlock additional customizability and functionality through operational parameters

**Video Script Notes:**

* Demonstration of how to start the program
* Demonstration of the functionality:
  + What functionality do we want to show?

This will be assessed using:

* Structure
* Visual Presentation
* Audio explanation

Client-side functionality

* User Experience (UX): clean layout and minimal clicks/entry required
* App complexity: entities can be listed and edited
* 'Single page' style: asynchronous updates

Server-side functionality

* More than one entity type, with relationships
* REST API provides each entity with appropriate GET/POST methods
* Installs with npm install
* Starts with npm start