**Practica – Pentesting**

**Parte 1**

1. Dado el dominio www.frc.utn.edu.ar, se tiene por objetivo realizar fase de reconocimiento, para ello se realizarán las siguientes tareas:

a. Información de las IP y rangos.

b. Búsqueda en Google de posibles correos electrónicos de usuarios del sistema

c. Información de Whois

d. Registros TXT, MX, servidores DNS

e. Reconocimiento de puertos

f. Reconocimiento de directorios

g. Tecnologías utilizadas

h. Subdominios

Documentar cada paso en un archivo de texto (Word, Libre Docs, etc.)

Herramientas: <https://spice-eyelash-aa0.notion.site/Commands-b3477ebdfb934ae5b1d07f5b5e1eb595>

1. Información de IPs obtenida de <https://synapsint.com/>

Texto

Descripción generada automáticamente

Información de IPs complementaria, incluyendo datos acerca de los DNS extraída de <https://www.robtex.com/dns-lookup/www.frc.utn.edu.ar>

Tabla

Descripción generada automáticamente

Traceroute extraído de <https://centralops.net/co/>

Tabla

Descripción generada automáticamente

1. Posibles correos electrónicos de usuarios

Google dork: *intext:\*@frc.utn.edu.ar*

[ccoggiola@frc.utn.edu.ar](mailto:ccoggiola@frc.utn.edu.ar)

[jsalomone@frc.utn.edu.ar](mailto:jsalomone@frc.utn.edu.ar)

[jcvazquez@frc.utn.edu.ar](mailto:jcvazquez@frc.utn.edu.ar)

[mbartolomeo@frc.utn.edu.ar](mailto:mbartolomeo@frc.utn.edu.ar)

[garaguas@frc.utn.edu.ar](mailto:garaguas@frc.utn.edu.ar)

[cids@sistemas.frc.utn.edu.ar](mailto:cids@sistemas.frc.utn.edu.ar)

[cemetro.utn@frc.utn.edu.ar](mailto:cemetro.utn@frc.utn.edu.ar)

[computos@frc.utn.edu.ar](mailto:computos@frc.utn.edu.ar)

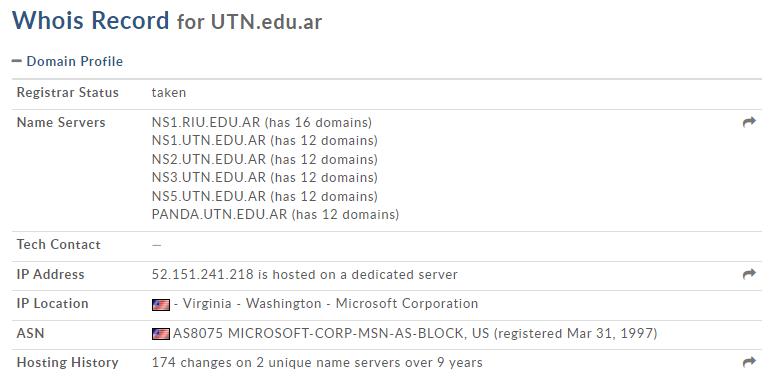
1. Información de whois obtenida en <https://centralops.net/co/>



Texto, Carta

Descripción generada automáticamente

Información complementaria de Whois de <https://whois.domaintools.com/utn.edu.ar>



1. Registros TXT, MX, servidores DNS

Información de servidores DNS extraído de <https://synapsint.com/> Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Información complementaria de DNS, registros MX y registros TXT obtenida de <https://centralops.net/co/>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



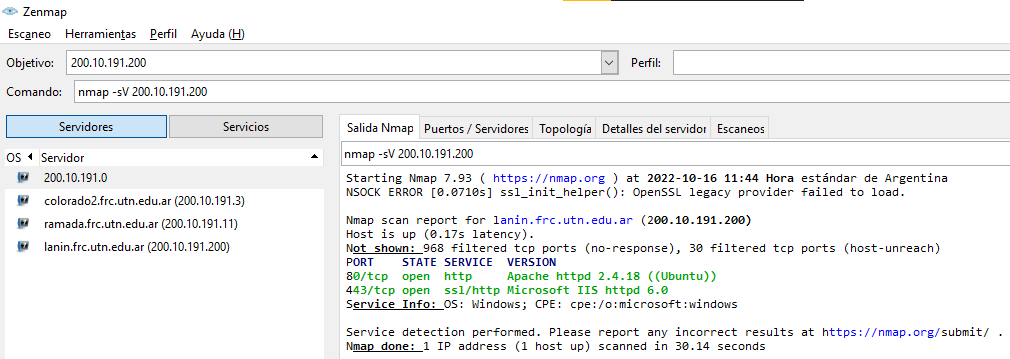
1. Información de puertos extraída de <https://synapsint.com/> 

Escaneo de puertos de <https://centralops.net/co/>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

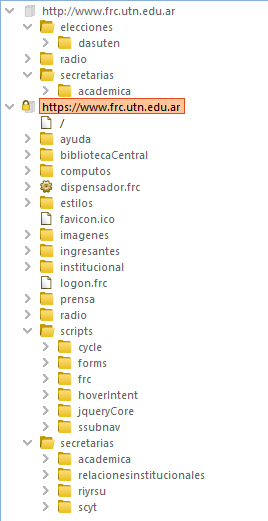
Escaneo de puertos con herramienta nmap



1. Reconocimiento de directorios

Interfaz de usuario gráfica, Texto

Descripción generada automáticamenteNavegador

Burp suite

1. Tecnologías utilizadas

Extraído de <https://synapsint.com/>

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Información complementaria obtenida en <https://w3techs.com/sites/info/utn.edu.ar>

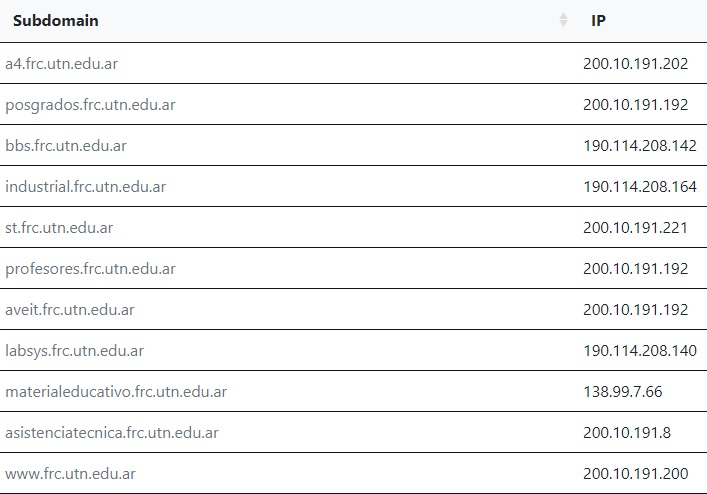
Texto

Descripción generada automáticamente con confianza media

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

h. Subdominios - Algunos subdominios con sus direcciones IP, extraído de <https://synapsint.com/>



Tabla

Descripción generada automáticamente

**Parte 2**

1. Levantar el proyecto de “Pixi” y acceder a localhost.

2. Buscar / retomar las vulnerabilidades.

3. Con **una** de ellas, armar la plantilla de vulnerabilidad, definiendo:

- Título de la vulnerabilidad

- Categoría de vulnerabilidad

- Score CVSS

- Descripción (determinando el impacto)

- Pasos de reproducción

- Capturas de pantalla

- Recomendaciones

**Título de la vulnerabilidad**: Logueo en sitio web, sin conocer la contraseña

**Categoría de vulnerabilidad**: Inyección NoSQL - NoSQLi

**Score CVSS:** 9.1 (Crítica)

Vector string CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

**Descripción (determinando el impacto)**: puedo loguearme en el sitio Pixi sin necesidad de conocer la contraseña de un usuario determinado.

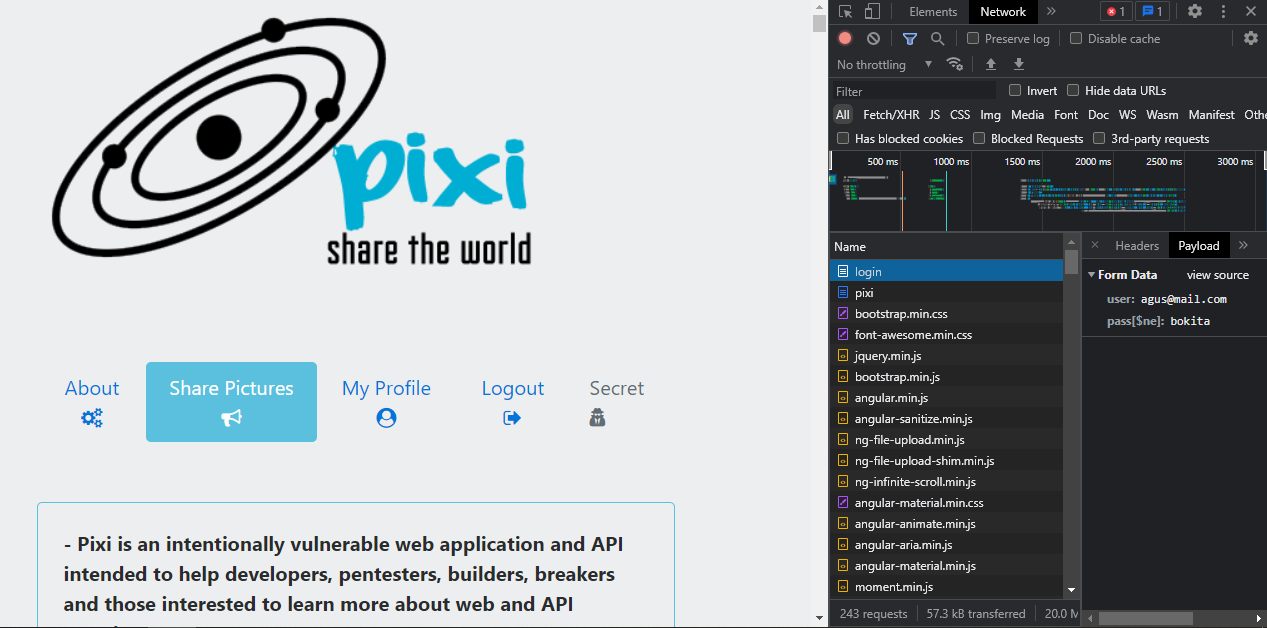
**Pasos de reproducción:**

1. Ingreso al sitio Pixi, e inspecciono el elemento correspondiente al text box de contraseña.
2. Modifico el name del input type password para verificar si el backend procesa lo que le estoy mandando

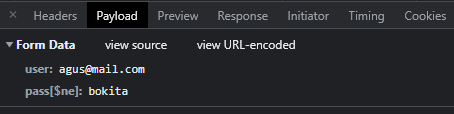


De esta forma veremos si el backend toma el operador $ne para validar que la clave sea not equal al valor ingresado en el input

1. Indico un nombre de usuario existente, en este caso [agus@mail.com](mailto:agus@mail.com), y en la password ingresamos un string cualquiera, como ser “bokita”. Cualquier valor serviría, inclusive dejarlo en blanco.
2. Seleccionamos Login y podremos loguearnos



Si vemos el payload de la request



**Recomendaciones**:

Implementación de mejoras en el código:

* Sanitización del campo password, es decir, validar que el parámetro sea exactamente un string y no un array.
* Otra alternativa, es validando el parámetro con la función con filter\_input()