Pentesting

Adnner Esperilla Ruiz,Jorge Pacora

August 25, 2020

Abstract

The purpose of this presentation is to provide an overview of the application of penetration testing to secure systems administration. As such, the presentation is not overly technical in scope, but covers instead what penetration testing is, what benefits stakeholders in a secure system receive from a test, and how policies can aid or hinder penetration testing. Penetration testing is a specialized security auditing method where a tester simulates an attack on a secured system. The goal of this is not to cause damage, but instead to identify attack surfaces, vulnerabilities, and other security weaknesses from the perspective of an attacker. Such testing can range across all aspects of a system; the areas of computer, operational, personnel, and physical security can all encompass potential weaknesses that a malicious attacker can exploit, and thus a penetration tester may examine. Depending on the organization's priorities, risk assessment, and policies, some of these areas may be out of scope or not deemed as important, so a reduced scope penetration test may be conducted.

I. Introduccion

Podemos defenir que son normalmente un conjunto de "tests de penetración" basados en ataques hacia sistemas informáticos con la intencionalidad de encontrar debilidades y/o vulnerabilidades relativas a la seguridad, pudiendo clasificar más determinar el alcance y la repercusión de las mismas.

II. TITULO

- APLICACION DE PENTESTING

III. AUTORES

- Adnner Esperilla Ruiz
- Pacora

IV. PLANTEAMIENTO DEL PROBLEMA

i. Problema

- La Municipalidad de Pocollay, es una entidad del estado, del orden territorial y a favor de la comunidad, cuyo objetivo es velar y prestar sus servicios públicos con el cumplimiento de un rol fundamental para con la población, ya que como institución pública, autónoma y jurídicamente hablando, puede promover e implementar toda clase de actividades políticas, económicas, sociales y culturales, con la visión de satisfacer las necesidades de la comunidad.
  El ente territorial en su principio de actualización e inmersión tecnológica para todos y cada uno de los procesos que se generan funcionalmente, ha implementado un sistema distribuido de red de computadoras LAN (Local Área Network) la cual permite compartir recursos Software, Hardware e información, elementos que deben contar con confidencialidad, Integralidad y disponibilidad.
  Dicha situación conlleva a una nefasta perdida, robo o destrucción de la información, suplantación de identidad de funcionarios, virus informáticos, fallos en los sistemas de información, mal funcionamiento del Hardware, intermitencia o caída de la red (offnet), Spoffing de DNS, IP o DCHP, denegación del servicio (DoS), ingeniería social, entre otras situaciones críticas que afectan el funcionamiento de una red y

sus servicios.

ii. Justificacion

El desarrollo de esta investigación le permitirá a la Municipio de Pocollay, contar con red de computadoras y sus respectivos servicios de forma segura, permitiendo generar un alto grado de confidencialidad, Integralidad y disponibilidad de la información.
Los análisis de vulnerabilidades o PenTesting permitirán determinar el nivel de seguridad en: un equipo, en la red de equipos LAN (Local Área Network) o WLAN (Wireless local Área Network), aplicaciones Web, Servidores de Información, entre otros, por medio de ataques informáticos simulados idénticos a los que realizaría un Cracker o Black Hat Hacker pero sin poner en riesgo la información o la disponibilidad de los servicios, esto se hace con el fin de encontrar las posibles amenazas o vulnerabilidades en los sistemas informáticos antes de que las descubra un atacante (externo o interno).

iii. Alcance

El proyecto de investigación se desarrollará tomando como punto de partida los elementos o fases de la investigación cualitativa. Determinando un alcance de procedimientos exploratorios, y teniendo en cuenta los diferentes métodos y técnicas propias de cada una de las etapas que se abordaran en el estudio, incluyendo los procedimientos, recolección, procesamiento y análisis de la información, además del seguimiento al cronograma de actividades.

V. OBJETIVOS

i. General

Describir los problemas de seguridad de la red de computadoras en Municipalidad de Pocollay, a través de pruebas de penetración que permitan el mejoramiento continuo de la entidad

ii. Especificos

Realizar un pentesting “prueba de penetración” para la determinar qué tipo de vulnerabilidades presenta la red de computadoras en el Municipio de Pocollay

Identificar los diferentes ataques a los que está expuesta la red de computadoras y sus servicios.

Generar recomendaciones que reduzcan la vulnerabilidad de la red de computadoras

VI. REFERENTES TEORICOS

Disponibilidad: Hace referencia a la capacidad de un sistema que permite realizar consultas en la medida que se requiera de una manera rápida y eficaz por el personal autorizado. También se refiere a la capacidad de que la información pueda ser recuperada en el momento que se necesite.

Confidencialidad: hace referencia a la privacidad de la información, la seguridad informática debe proteger un sistema informático de acceso a la información por parte de personal o programas no autorizados.

Integridad: Es la cualidad que posee un documento o archivo que no ha sido alterado y que además permite comprobar que no se ha producido manipulación alguna en el documento original.

VII. DESARROLLO DE LA PROPUESTA

i. Tecnologias de Informacion

Las Herramientas a Utilizar para realizar esta propuesta de Desarrollo son las siguientes :

Android Studio

ILenguaje Java , Xml

Sqlite

Nmap

Metasploit

ii. Metodología, técnicas usadas

Metodologia Scrum

VIII. CRONOGRAMA

ITEM	FASES	CRONOGRAMA DE ACTIVIDADES															
		MAYO				JUNIO				JULIO				AGOSTO			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	I	X	X	X	X												
2	II			X	X	X											
3	III					X	X	X	X	X	X	X					
4	IV									X	X	X	X				
5	V											X	X	X	X		
6	VI														X	X	X

Fase I: Identificación de las diferentes fuentes de información que permitirán ampliar la perspectiva del conocimiento a aplicar.

Fase II: Análisis y clasificación de la información según su género, origen y categoría.

Fase III: Selección y aplicación de las herramientas de Pentesting "Prueba de Penetración" para determinar las vulnerabilidades de la red de computadoras y sus servicios.

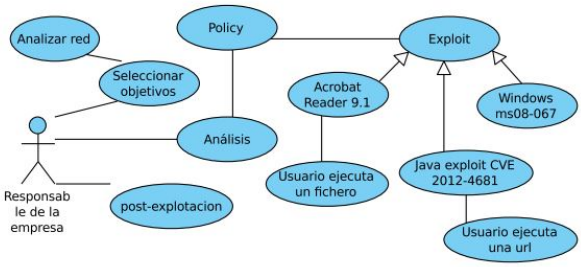
Fase IV: Análisis de resultados obtenidos de la red de computadores y sus servicios e identificación de las vulnerabilidades

Fase V: Aplicación de medidas correctivas y sugerencias para mitigar las vulnerabilidades de la red de computadora y sus servicios.

Fase VI: Conclusiones y elaboración de un Documento final.

IX. CASOS DE USO

Podemoss ver que solo existe un actor, el responsable, que es quien debe ejecutar el software cuya primera acción es buscar los dispositivos conectados dentro de la red donde se ejecuta el programa, para así poder seleccionar los objetivos sobre los que realizar los análisis. A continuación, debe seleccionar un análisis ya creado y elegir las políticas de las que consta dicho análisis. Vemos que las políticas están formadas por diferentes exploits, que pueden tener diferentes comportamientos. Una vez realizado el análisis, y en el caso de haber establecido una sesión con un objetivo, podrá ejecutar el módulo de post-explotación para realizar cualquier acción sobre el objetivo.



X. DIAGRAMA DE ARQUITECTURA

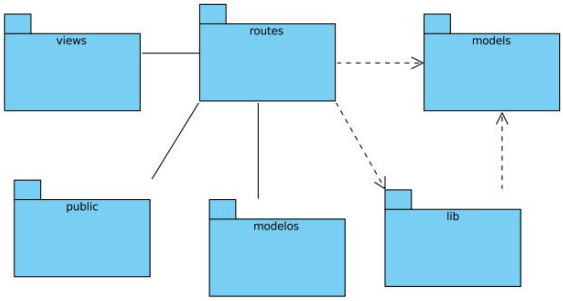
Para realizar más sencillo el desarrollo de la aplicación, se ha elegido el sistema de arquitectura por capas, un patrón que separa los datos y la lógica de una aplicación de la interfaz. Como principal ventaja, cabe destacar que el desarrollo se puede realizar independientemente de qué parte se amplíe, por lo que encaja perfectamente en el desarrollo que se ha planificado. También es ideal debido a la metodología elegida, ya que es un desarrollo incremental. En el modelo multicapa, se suele encontrar tres niveles: presentación, desarrollo y datos.

Presentación: En la capa de presentación es aquella donde se presenta el sistema al usuario y donde se ejecutan las acciones y se visualiza la información desde el servidor, como se puede ver en el diagrama de paquetes

Desarrollo: Es aquel código que recibe las peticiones que solicita el cliente y mediante el cual se envían las respuestas. Esta capa se comunica con la capa de presentación y es mediante esta comunicación interna cómo se envía la información al cliente. En nuestro software, son clases de gestión todas aquellas almacenadas en el directorio routes.

Datos : Es la capa por la cual se recuperan los datos desde la BD (en este caso Mon43 goDB). Cada clase está formada normalmente por los métodos Create, Read, Update y Delete (CRUD) para manipular los objetos con la BD:

create (crear), read (leer), update (actualizar) y delete (borrar)



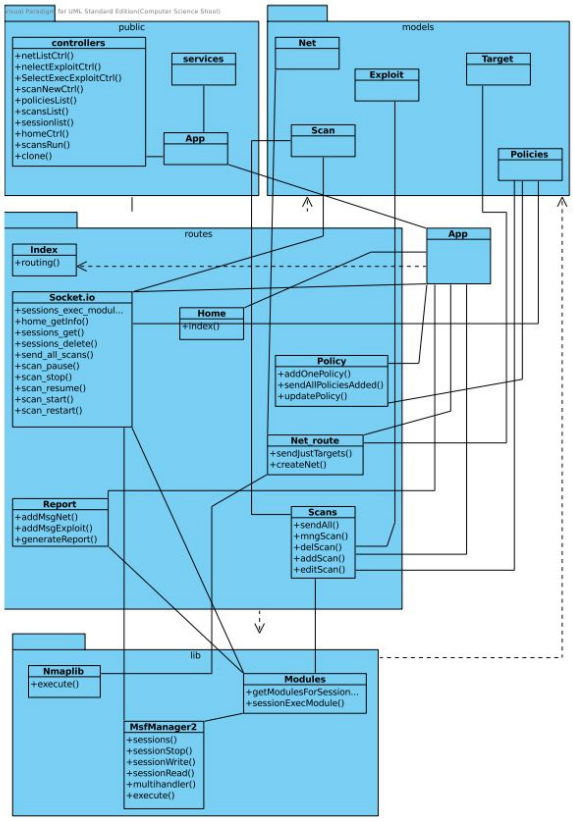
Routers : Se corresponde con la capa de gestión o desarrollo y contiene las clases que gestionan las peticiones de Express, correspondientes a las solicitudes que hace el cliente. La organización de este paquete se ha llevado a cabo creando una clase para cada tipo de petición. Así, las peticiones sobre análisis las gestiona la clase Scan, por ejemplo. Este paquete depende de models (la capa de datos) para crear nuevas instancias de los datos o recuperarlos para manipularlos. Por otro lado, libs contiene las bibliotecas que gestionan los datos. Además, tiene relación con views y public ya que es la capa intermedia entre la capa de presentación y datos, y en esos paquetes se encuentran las diferentes clases del entorno gráfico.

Models: En este paquete se encuentran las clases correspondientes a la capa de datos. Concretamente cada archivo es un model de uno o varios schemas que modelan los datos de la aplicación correspondiente a la collection (lo correspondiente a una tabla en la BD) y además de los métodos CRUD, permite añadir funciones para ejecutarse en la instancia. Más adelante, se detalla cómo facilita mongoose un entorno claro y sencillo con los modelos o models.

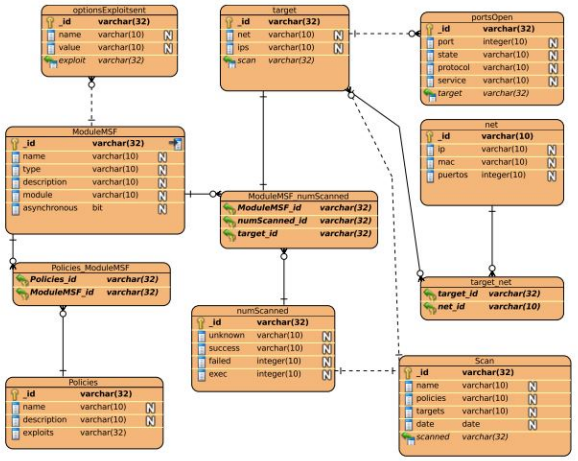
Lib: Forman parte de la capa de desarrollo ya que son clases que son utilizadas para aquellas del paquete routes, como se ha mencionado anteriormente.

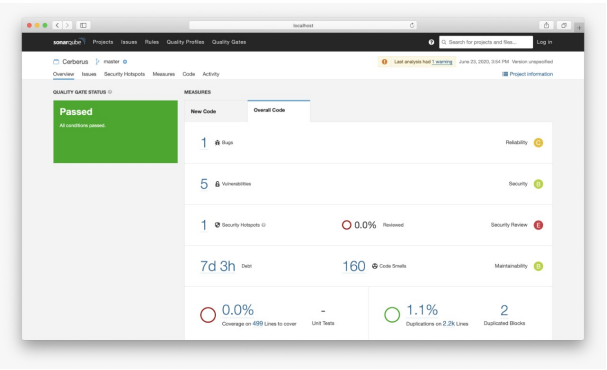
Public : Contiene los ficheros HTML, Javascript y demás que son servidos por el servidor web express y que verá el cliente. Obviamente, corresponde a la capa de presentación.

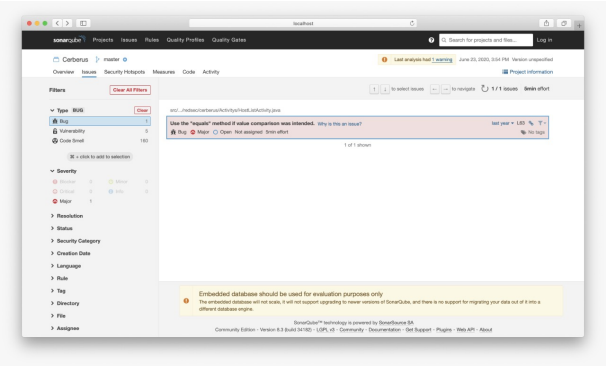
XI. DIAGRAMA DE CLASES



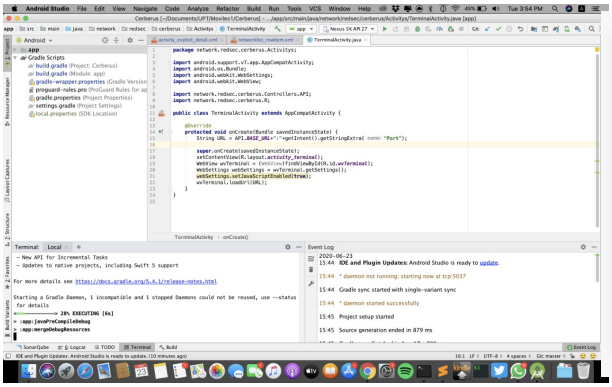
Para la visualización en detalle del diseño de la base de datos, se ha creado otro diagrama de entidad-relación.

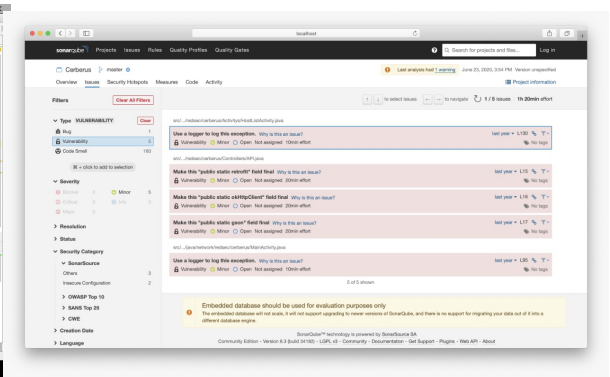


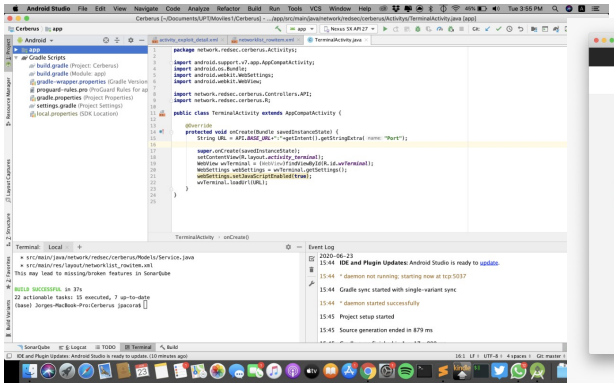


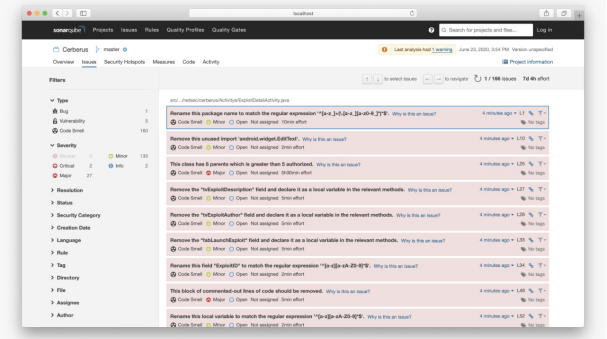


XII. ANALISIS DE SONARQUBE









//

XIII. RESULTADO DE PRUEBAS IMPLEMENTADAS

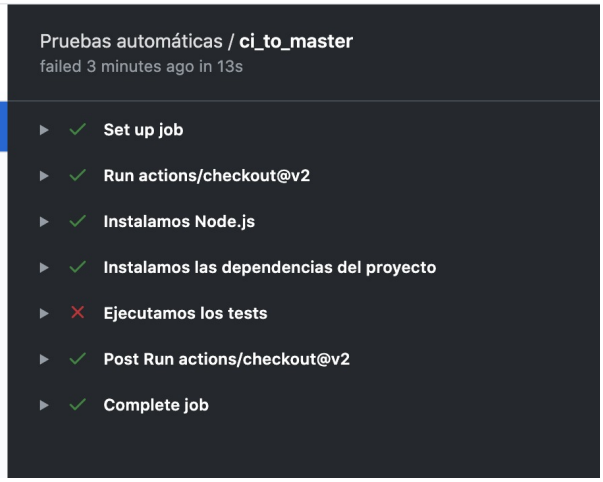
i. CONFIGURACION

```
> api@0.0.0 test /Users/jpacora/Documents/RedSec/secapi/API
> mocha --timeout 50000 ./tests/api.test.js


  Pruebas unitarias para la API
GET /api/Challenge 200 3.778 ms - 17
    ✓ El Challenge debe retornar estado 200 (OK)
GET /api/Interfaces 200 1.065 ms - 4109
    ✓ Las interfaces de red debe contener el campo "address"
GET /api/ExploitDB/Search?q=win 200 27.753 ms - 367963
    ✓ Buscar una vulnerabilidad en la DB
GET /api/ExploitDB/61 200 1.376 ms - 2908
    ✓ Obtener la vulnerabilidad con ID 61
[WiFi] Scan started
GET /api/WiFi/Start 200 0.814 ms - 20
    ✓ El WiFi debe poder iniciarse
GET /api/WiFi/Stop 200 0.620 ms - 20
    ✓ El WiFi debe poder pararse
GET /api/WiFi/Scan 200 0.578 ms - 2
    ✓ El scaneo del WiFi debe ser accesible
GET /api/TerminalHUB 200 0.759 ms - 2
    ✓ Obtener la lista de sesiones de TerminalHUB
GET /api/TerminalHUB/New 200 127.428 ms - -
    ✓ Crear una nueva sesión de TerminalHUB (129ms)


  9 passing (212ms)
```

```yaml
name: Pruebas automáticas
on:
    push:
      branches: [ master ]
jobs:
  ci_to_master:
    runs-on: ubuntu-latest
    defaults:
      run:
        working-directory: ./API
    steps:
      - uses: actions/checkout@v2
      - name: Instalamos Node.js
        uses: actions/setup-node@v1
        with:
          node-version: 10
      - name: Instalamos las dependencias del proyecto
        run: npm ci
      - name: Ejecutamos los tests
        run: npm test
```

ii. SE CORRE LA PRUEBA

XIV. PRUEBAS EN GIT HUB ACTIONS



iii. ACTIVAMOS MODULOS

```yaml
name: Pruebas automáticas
on:
    push:
      branches: [ master ]
jobs:
  ci_to_master:
    runs-on: ubuntu-latest
    defaults:
      run:
        working-directory: ./API
    steps:
      - uses: actions/checkout@v2
        with:
          submodules: true
      - name: Instalamos Node.js
        uses: actions/setup-node@v1
        with:
          node-version: 10
      - name: Instalamos las dependencias del proyecto
        run: npm ci
      - name: Ejecutamos los tests
        run: npm test
```

iv. SE CORRE LA PRUEBA DE NUEVO