Laboratorio de DNS Rebinding

1 Introducción

El objetivo de este laboratorio es separado en dos partes: (1) demostrar como funciona el ataque de DNS Rebinding, y (2) ayudar a que los estudiantes obtengan experiencia en el uso de la técnica de DNS Rebinding para atacar dispositivos IoT. En el setup, hemos simulado un dispositivo IoT que puede ser controlado a través de una interfaz web (esto es típico en muchos dispositivos IoT). Muchos de estos dispositivos no tienen mecanismos de seguridad robustos, si los atacantes pueden interactuar directamente con esta interfaz poco segura, podrán hacerse del control de estos dispositivos.

El dispositivo IoT simulado en este laboratorio es un termostato que controla la temperatura de una habitación. Para controlar y setear la temperatura de manera correcta, el cliente necesita interactuar con el servidor IoT. Dado que el dispositivo IoT se encuentra detrás de un firewall, las máquinas por fuera de este no podrán acceder a este dispositivo IoT y por ende no podrán controlar el termostato. Para evadir la protección del firewall, el código del atacante debe estar dentro de la red interna en donde se encuentra el dispositivo. Esto no es difícil. Cada vez que un usuario de una red interna visita el sitio de un atacante, el código del atacante (Código JavaScript) se ejecuta en el navegador del usuario y corre dentro de la red interna que se está protegida. Sin embargo, debido a las protecciones de sandbox implementadas por los navegadores, el código del atacante no podrá interactuar con el dispositivo IoT aunque este se encuentre en la red interna.

El objetivo de este laboratorio es usar el ataque de DNS Rebinding para evadir la protección sandbox, y hacer que el código JavaScript del atacante puede obtener información sensible del dispositivo IoT de manera exitosa luego de esto usar esta información para setear la temperatura del termostate en un valor alto que pueda ser peligroso.

Este laboratorio cubre los siguientes tópicos:

- Setup del servidor DNS
- El Ataque de DNS Rebinding
- Ataques en dispositivos IoT
- Política de Same Origin (Same Origin Policy)

Readings and videos. Para una cobertura más detallada sobre el protocolo DNS y sus ataques puede consultar:

- Capítulo 18 del libro de SEED, *Computer & Internet Security: A Hands-on Approach*, 2nd Edition, by Wenliang Du. Para más detalles `https://www.handsonsecurity.net`.
- Sección 7 del curso de SEED en Udemy, *Internet Security: A Hands-on Approach*, by Wenliang Du. Para más detalles `https://www.handsonsecurity.net/video.html`.

Entorno de Laboratorio. Este laboratorio ha sido testeado en nuestra imagen pre-compilada de una VM con Ubuntu 20.04, que puede ser descargada del sitio oficial de SEED. Sin embargo, la mayoría de nuestros laboratorios pueden ser realizados en la nube para esto Ud. puede leer nuestra guía que explica como crear una VM de SEED en la nube.

2 Background: IoT

Nuestro objetivo a atacar será un dispositivo IoT detrás de un firewall. No podemos acceder directamente a este dispositivo por fuera de la red interna. Nuestra meta será que el usuario corra nuestro código JavaScript, y de esta forma usar el ataque de DNS Rebinding para interactuar con el dispositivo IoT.

Muchos dispostivos IoT vienen con un simple un servidor web corriendo dentro de ellos, de esta forma los usuarios pueden interactuar con el dispositivo a usando APIs web. A menudo estos dispositivos son protegidos por un firewall. Debido a este tipo de protección, muchos equipos de este tipo no implementan un mecanismo de autenticación robusto. Si los atacantes pueden encontrar la forma de interactuar con ellos, pueden comprometer la seguridad de estos dispositivos de manera sencilla.

En este laboratorio vamos a emular un dispositivo IoT vulnerable usando un servidor web muy simple, que servirá dos APIs: `password` y `temperature`. El dispositivo IoT setea la temperatura de una habitación. Para hacerlo, necesitamos enviar request HTTP a la API `temperature`; el request debe de incluir dos datos: la temperatura y el passowrd. El password cambia períodicamente, pero se puede obtener usando `password`. Para setear la temperatura de forma exitosa, los usuarios necesitan obtener primero el password y luego incluir este password en el llamado a la API `temperature`.

El password no fue pensando para propósitos de autenticación; este es usado para evadir el ataque de Cross-Site Request Forgery (CSRF). Sin esta protección, un ataque de Cross-Site Request Forgery (CSRF) es suficiente; no habría necesidad de usar un ataque de tipo DNS Rebinding. Para simplificar un poco todo, hemos harcodeado el password; en los sistemas reales, el password se regenera periódicamente.

3 Setup del Entorno de Laboratorio usando Contenedores

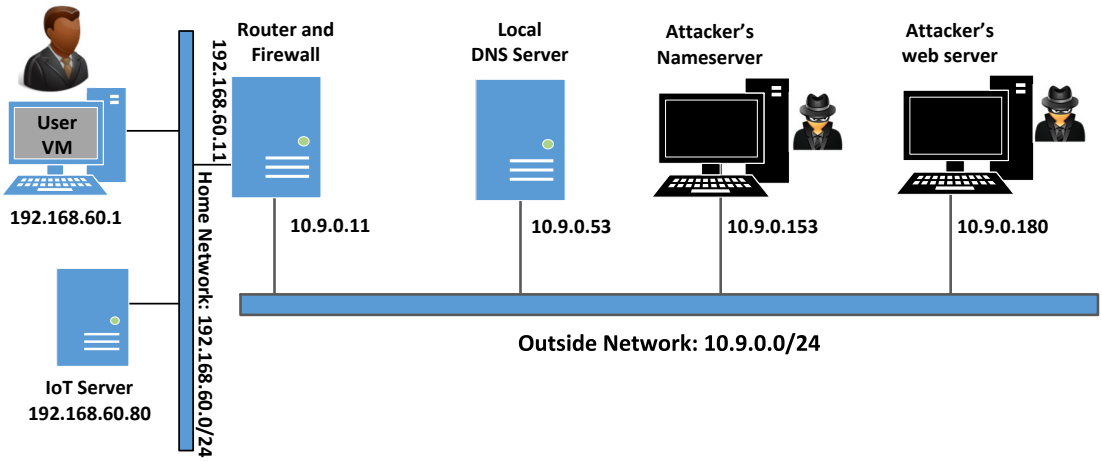

Figure 1: Setup del Entorno de Laboratorio

En este laboratorio, usaremos seis máquinas. El setup del entorno del entorno de laboratorio es ilustrado en la Figura 1. Sólo la máquina del usuario será una Máquina Virtual, el resto serán contenedores. En el

setup, tenemos dos redes, una red hogareña y otra red externa. La red hogareña simula una red típica de una casa. La máquina de usuario y los servicios IoT están conectados a esta red, que es protegida por un firewall en un contenedor router. El firewall bloquea todo el tráfico a `192.168.60.80`. De esta forma, las máquinas externas no pueden acceder al dispositivo. También hemos hecho un setup para un servidor NAT en el router, de esta forma las maquinas en la red hogareña pueden salir a Internet.

La segunda red simula el mundo externo. Aparte del router, existen tres contenedores conectados a esta red, uno es el servidor de DNS local, y los otros dos serán el nameserver de ataque y el web server. El atacante posee el dominio `attacker32.com`, que está hosteado en el contenedor del nameserver. El web server hostea el sitio malicioso que es usado para el ataque.

3.1 Setup del Contenedor y sus Comandos

Para empezar a preparar el contenedor, deberá descargarse el archivo `Labsetup.zip` ubicado en el laboratorio correspondiente dentro del sitio web oficial y copiarlo dentro de la Máquina Virtual prevista por SEED. Una vez descargado deberá descomprimirlo y entrar dentro del directorio `Labsetup` donde encontrará el archivo `docker-compose.yml` que servirá para setear el entorno de laboratorio. Para una información más detallada sobre el archivo `Dockerfile` y otros archivos relacionados, puede encontrarla dentro del Manual de Usuario del laboratorio en uso, en el sitio web oficial de SEED.

Si esta es su primera experiencia haciendo el setup del laboratorio usando contenedores es recomendable que lea el manual anteriormente mencionado.

A continuación, se muestran los comandos más usados en Docker y Compose. Debido a que estos comandos serán usados con mucha frecuencia, hemos creados un conjunto de alias para los mismos, ubicados en del archivo `.bashrc` dentro de la Máquina Virtual provista por SEED (Ubuntu 20.04)

```
$ docker-compose build  # Build the container image
$ docker-compose up     # Start the container
$ docker-compose down   # Shut down the container

// Aliases for the Compose commands above
$ dcbuild       # Alias for: docker-compose build
$ dcup          # Alias for: docker-compose up
$ dcdown        # Alias for: docker-compose down
```

Dado que todos los contenedores estarán corriendo en un segundo plano. Necesitamos correr comandos para interactuar con los mismos, una de las operaciones fundamentales es obtener una shell en el contenedor. Para este propósito usaremos "`docker ps`" para encontrar el ID del contenedor deseado y ingresaremos "`docker exec`" para correr una shell en ese contenedor. Hemos creado un alias para ello dentro del archivo `.bashrc`

```
$ dockps        // Alias for: docker ps --format "{{.ID}}  {{.Names}}"
$ docksh <id>   // Alias for: docker exec -it <id> /bin/bash

// The following example shows how to get a shell inside hostC
$ dockps
b1004832e275  hostA-10.9.0.5
0af4ea7a3e2e  hostB-10.9.0.6
9652715c8e0a  hostC-10.9.0.7

$ docksh 96
root@9652715c8e0a:/#
```

```
// Note: If a docker command requires a container ID, you do not need to
//       type the entire ID string. Typing the first few characters will
//       be sufficient, as long as they are unique among all the containers.
```

En caso de problemas configurando el entorno, por favor consulte la sección "Common Problems" en el manual ofrecido por SEED.

3.2 Configurar la Máquina Virtual del Usuario

Necesitamos proporcionar una configuración adicional en la máquina virtual del usuario.

Paso 1.Reducir el tiempo de expiración de la caché DNS del Firefox. Con el objetivo de reducir la carga en los servidores DNS y optimizar los tiempos de respuestas, el navegador Firefox cachea los resultados de los DNS. Por defecto el tiempo de expiración de esta caché es de 60 segundos. Eso quiere decir que nuestro ataque de DNS Rebinding necesita esperar al menos 60 segundos. Para hacer todo mas sencillo, hemos reducido este tiempo a 10 segundos o menos. Para esto en la barra de navegación escriba `about:config`. Despues de hacer click y pasar la página de advertencia, veremos una lista de preferencias con sus nombres y valores correspondientes. Busque `dnsCache` encuentre la entrada y cambie su valor:

```
network.dnsCacheExpiration: cambie el valor a 10 (el valor por defecto es 60)
```

Después de hacer este cambio, debe de reiniciar el navegador, de otra forma no tomará efecto.

Paso 2. Cambiar `/etc/hosts`. Necesitamos agregar la siguiente entrada en el archivo `/etc/hosts`. Usaremos `www.seedIoT32.com` como el nombre para el servidor IoT. Su dirección IP es `192.168.60.80`. Necesitamos tener privilegios de superusuario para modificar este archivo (usando `sudo`):

```
192.168.60.80  www.seedIoT32.com
```

Mientras este editando este archivo, verifique si existe alguna entrada que contenga `attacker32.com`. Si está presente borrela.

Estamos listos para testear el servidor IoT. Ingreser en su navegador la siguiente URL usando la máquina virtual de usuario. Si está todo configurado correctamente, deberíamos de ver un termostato. También podemos cambiar la temperatura usando la barra de sliding. Por favor provea una captura de pantalla en su informe del laboratorio.

```
http://www.seedIoT32.com
```

Paso 3. Servidor de DNS local. Necesitamos que la máquina virtual del usuario use un servidor de DNS local particular. Esto se logra cambiando el archivo de configuración (`/etc/resolv.conf`) en la máquina del usuario de manera tal que se agregará la dirección IP del contenedor como la primera entrada `nameserver` dentro de este archivo, es decir, este servidor será usado como el servidor de DNS primario. Desafortunadamente nuestra Máquina Virtual usa DHCP para obtener los parámetros de la configuración de red, tales como la dirección IP, el servidor de DNS local, etc. Los clientes DHCP sobreescribirán el archivo `/etc/resolv.conf` con su la información provista por el servidor DHCP.

Una forma de evitar esto es agregar la siguiente entrada dentro del archivo `/etc/resolvconf/resolv.conf.d/head` (`10.9.0.53` es la dirección IP de nuestro servidor de DNS local que hemos configurado en el setup):

```
nameserver 10.9.0.53
```

El contenido del archivo head será antepuesto al que se usa en la generación dinámica por el DHCP. Después de hacer este cambio necesitamos correr el siguiente comando para que tome efecto.

```
$ sudo resolvconf -u
```

3.3 Probando el Setup del Laboratorio.

Después de configurar la máquina virtual del usuario, use el comando `dig` para obtener la dirección IP de `www.attacker32.com` y `ns.attacker32.com`. Debería de obtener `10.9.0.180` y `10.9.0.153` respectivamente. Si ud. no obtiene estas direcciones su entorno de laboratorio no fue configurado correctamente.

Ahora podemos testear el sitio del atacante. Ingrese la siguiente URL en la máquina virtual del usuario y debería de poder ingresar al sitio del atacante. Por favor incluya una captura de pantalla en su informe del laboratorio.

```
http://www.attacker32.com
```

Note. Puede que se haya usado el mismo hostname `www.attacker32.com` en otros laboratorios de SEED, por lo cual es probable que ese hostname pueda haber quedado mapeado en una dirección IP diferente. Si ud. al entrar a la URL no ve el sitio del atacante indicado, por favor chequee el archivo `/etc/hosts` y borre cualquier entrada que contenga `attacker32.com`.

4 Lanzando el Ataque en el Dispositivo IoT

Estamos listos para lanzar el ataque en el dispositivo IoT. Para una mejor comprensión de parte de los estudiantes en como funciona el ataque, hemos fragmentado el ataque en varios pasos incrementales.

4.1 Tarea 1. Entendiendo la Protección de la Política de Same-Origin

En esta tarea, haremos algunos experimentos para entender la protección otorgada por la política de same-origin (Same Origin Policy), este mecanismo se encuentra implementado en los navegadores. En la Máquina Virtual del Usuario, navegaremos a las siguientes tres URLs. Es mejor que estas tres páginas se abran en tres ventanas diferentes de Firefox (en vez de tres tabs diferentes), para así poder visualizarlas por completo.

```
URL 1:  http://www.seedIoT32.com
URL 2:  http://www.seedIoT32.com/change
URL 3:  http://www.attacker32.com/change
```

En la primera página veremos la temperatura actual que está seteada en el termostato (vea la Figura 2.a); esta obtiene el valor de la temperatura del servidor IoT cada un segundo. Esta página debe estar siempre visible, de estas forma podremos observar la temperatura seteada en el termostato. La segunda y tercer página son iguales (vea la Figura 2.b), excepto que una viene del servidor de IoT y la otra viene del servidor del atacante. Cuando clickeamos en el botón en ambas páginas, un request será enviado al servidor IoT para setear la temperatura del termostato. Se supone que debemos de setear la setear la temperatura a `99` grados Celsius.

Haga click en el botón de la segunda y tercer página, y describa su observación. ¿Cuál es la página que permite setear de forma exitosa la temperatura del termostato? Explique el porque. Para encontrar la razón, siga la siguiente secuencia del menú de Firefox. Se abrirá una ventana de consola y desplegará mensajes de

Figure 2: Las páginas de las tres URLs

error si es que hay alguno/s. Pista: la razón se relaciona a la política de same-origin implementada por los navegadores. Por favor explique porque esta política hace que una de las páginas falle.

```
Web Developer -> Web Console
```

4.2 Tarea 2. Evadiendo la Protección de la Política de Same Origin

Basados en la tarea anterior, pareciera imposible establecer el valor de la temperatura del termostato desde la página del atacante, debido a la protección de la política de same-origin implementada por el navegador. El objetivo de esta tarea es evadir dicha protección, para así poder setear la temperatura desde esta página.

La principal idea en orden para evadir la protección de same-origin viene del hecho que esta política esta reforzada por el hostname y no en la dirección IP, por lo tanto mientras usemos `www.attacker32.com` como URL, estamos cumpliendo con la política SOP, lo que no significa que estamos restringidos para comunicarnos con el servidor web `www.attacker32.com`.

Antes que el usuario desde su navegador envíe un request hacia `www.attacker32.com`, el navegador necesita saber la dirección IP de `www.attacker32.com`. Una petición DNS será enviada desde la máquina del usuario. Si la dirección IP no está cacheada en el servidor de DNS local, una petición DNS será enviada al nameserver de `attacker32.com` que está controlado por el atacante. Por lo que el atacante puede decidir que poner en la respuesta.

Paso 1: Modificar el código JavaScript. En el servidor del atacante, el código JavaScript que se encuentra en la página `www.attacker32.com/change` está dentro del archivo: `/app/rebind_server/templates/js/change.js`. Dado que está página viene del servidor `www.attacker32.com`, de acuerdo a la política de same-origin, solamente puede interactuar con el mismo servidor. Necesitamos cambiar la primera línea del código de `http://www.seediot32.com` por la siguiente (en el contenedor hemos instalado un editor llamado `nano`):

```
let url_prefix = 'http://www.attacker32.com'
```

Después de haber hecho este cambio, debe reiniciar el servidor web en el contenedor del atacante (vea el comando mas abajo). Después de esto debe de ir a la página de la máquina virtual de usuario, refrescarla y hacer click en el botón nuevamente: ¿El error aún permanece? Por favor explique su observación.

```
$ docker ps
...
78359039627a  attacker-www-10.9.0.180

$ docker container restart 7835
```

Paso 2: Conduciendo el ataque de DNS Rebinding. Nuestro código JavaScript envía la petición a `www.attacker32.com`, es decir, la petición regresará hacia el servidor web del atacante. Eso no es lo que queremos; lo que queremos es que las peticiones vayan hacia el servidor IoT. Esto se puede lograr usando la técnica de DNS Rebinding. Primero mapeamos `www.attacker32.com` con la dirección IP del servidor web del atacante, así el usuario puede obtener la página desde `http://www.attacker32.com/change`. Antes de hacer click en el botón dentro de la página, remapeamos el hostname `www.attacker32.com` con la dirección IP del servidor IoT, lo que hará que la petición lanzada por el botón vaya hacia el servidor IoT. Eso es exactamente lo que queremos.

Para cambiar el mapeo DNS, los estudiantes pueden modificar el archivo de zona `zone_attacker32.com` dentro del contenedor del nameserver del atacante. El archivo zona se encuentra dentro del directorio `/etc/bind`. A continuación se muestra el contenido del archivo de zona. La primera entrada es el valor por defecto del Time-To-Live (`TTL`) (en segundos) para la respuesta, este especifíca cuanto tiempo se almacena la respuesta en caché. Este valor puede que necesite ser modificado. Contenido del archivo de zona:

```
$TTL 1000
@       IN      SOA   ns.attacker32.com. admin.attacker32.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS    ns.attacker32.com.

@       IN      A     10.9.0.22
www     IN      A     10.9.0.22
ns      IN      A     10.9.0.21
*       IN      A     10.9.0.22
```

Después de haber hecho los cambios en el archivo de zona, corra el siguiente comando para recargar los datos de la zona en el nameserver.

```
# rndc reload attacker32.com
```

Debido a que en las tareas anteriores hemos hecho algunas operaciones con los DNS, el mapeo DNS para `www.attacker32.com` ha sido cacheado por el servidor de DNS local, este tiene un tiempo de expiración de 1000 segundos o más. Para acortar la espera, los estudiantes pueden limpiar la caché usando el siguiente comando (en el servidor de DNS local). Sin embargo, esto puede hacerse antes de empezar el ataque. Una vez que el ataque comienza, los estudiantes no deberán de tocar el servidor de DNS local.

```
// Do it on the local DNS server container
# rndc flush
```

Si los dos pasos descriptos en esta tarea son realizados de forma correcta, al clickear el botón en la página `change` de `www.attacker32.com` debería de permitir cambiar la temperatura del termostato exitosamente. Por favor incluya la evidencia en su informe del laboratorio que demuestre que esto funcionó de manera correcta.

4.3 Tarea 3. Lanzar el Ataque

En la tarea anterior, el usuario debe de hacer click en el botón para setear la temperatura en un valor peligrosamente alto. Obviamente, los usuarios no harían eso. En esta tarea, necesitamos hacer esto de forma automática. Hemos creado una página web para este propósito. Se puede acceder usando la siguiente URL:

```
http://www.attacker32.com
```

Una vez que haya cargada esta página dentro de la máquina virtual del usuario, debería de poder ver una página con un timer, que desde 10 a 0. Una vez que este en 0, el código JavaScript enviará una petición para setear la temperatura a `http://www.attacker32.com` y reseteará el timer a 10. Los estudiantes necesitan usar la técnica de DNS Rebinding, para que cuando el timer este en 0, la temperatura del termostato sea 88 grados Celsius.

5 Informe del Laboratorio

Debe enviar un informe de laboratorio detallado, con capturas de pantalla, para describir lo que ha hecho y lo que ha observado. También debe proporcionar una explicación a las observaciones que sean interesantes o sorprendentes. Enumere también los fragmentos de código más importantes seguidos de una explicación. No recibirán créditos aquellos fragmentos de códigos que no sean explicados.

Agradecimientos

Este documento ha sido traducido al Español por Facundo Fontana