**INSTITUTO TECNOLÓGICO DE DURANGO**

**Seguridad Web**



**Reporte de Práctica**

**CLICK-JACKING**

**Docente:**

López Quiñones José Roberto

**Alumno:**

Deyanira Idaly Campos Romero 15041193

**Fecha de entrega**: 18 Febrero de 2019

**Objetivo**

Simular un ataque de click jacking , este ataque se simulara haciendo uso de dos páginas HTML/CSS utilizando algunas funciones de javascript dentro de las mismas y simulando dar clic en un botón, pero en realidad se re-dirige a otro lugar, dicho ataque se realizará dentro de clase de seguridad web.

Se realizará de igual manera dicha práctica, solamente utilizando lo ya mencionado anteriormente, en esta ocasión no se utilizaran herramientas más complejas como en prácticas anteriores, solo es necesario tener conocimientos sobre HTML/CSS y javascript.

Un objetivo importante es conocer la finalidad del ataque e informarse sobre cómo protegerse de él, de igual manera el poder ampliar los conocimientos que se tienen sobre algunos conceptos que se utilizan y conocer más sobre este tipo de ataque , el cual suele ser muy usual en la actualidad. Pudiendo ampliar aquella información obtenida a lo largo del semestre.

Ampliando un poco más el objetivo se pretende demostrar cómo es fácil detectar este tipo de ataque una vez que se conoce la definición y objetivos del mismo y como se pueden aprovechar haciendo uso del mismo diversos sitios web.

**Desarrollo**

**Click**-**Jacking**

Consiste en un nuevo método de ataque que se realiza a través del navegador, tratando de engañar al usuario mediante una capa transparente colocada delante de un enlace o cuadro de diálogo. Su objetivo es hacer que el usuario pulse un enlace sin percatarse de ello.

Es importante resaltar que el clickjacking es sólo una nueva técnica para facilitar ataques del tipo CSRF permitiendo engañar al usuario más fácilmente para que haga click. Además aumenta la efectividad de estos ataques evitando contramedidas para CSRF como los tokens aleatorios.

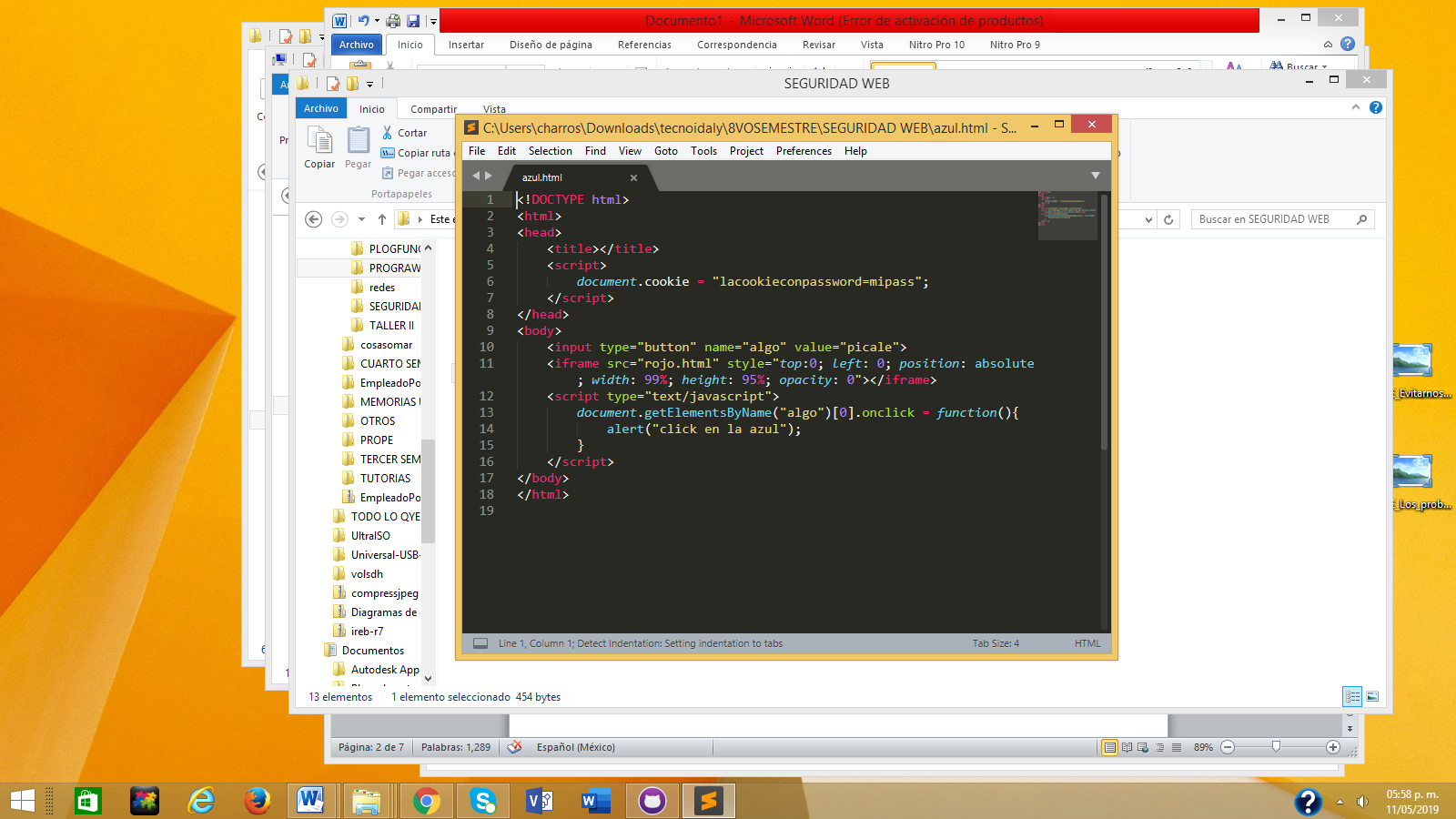
Afecta a todos los navegadores excepto los basados en texto, ya que éstos no implementan las funcionalidades de CSS necesarias para ocultar la capa.

El clickjacking fue descubierto por Jeremiah Grossman y Robert Hansen en 2008. También se le conoce como UI redressing. El concepto es sencillo, y las técnicas para conseguirlo no son difíciles de implementar. Se podría decir que la técnica se basa en un fallo de diseño de HTML y por tanto toda web es "vulnerable" por defecto.

Pasos a seguir:

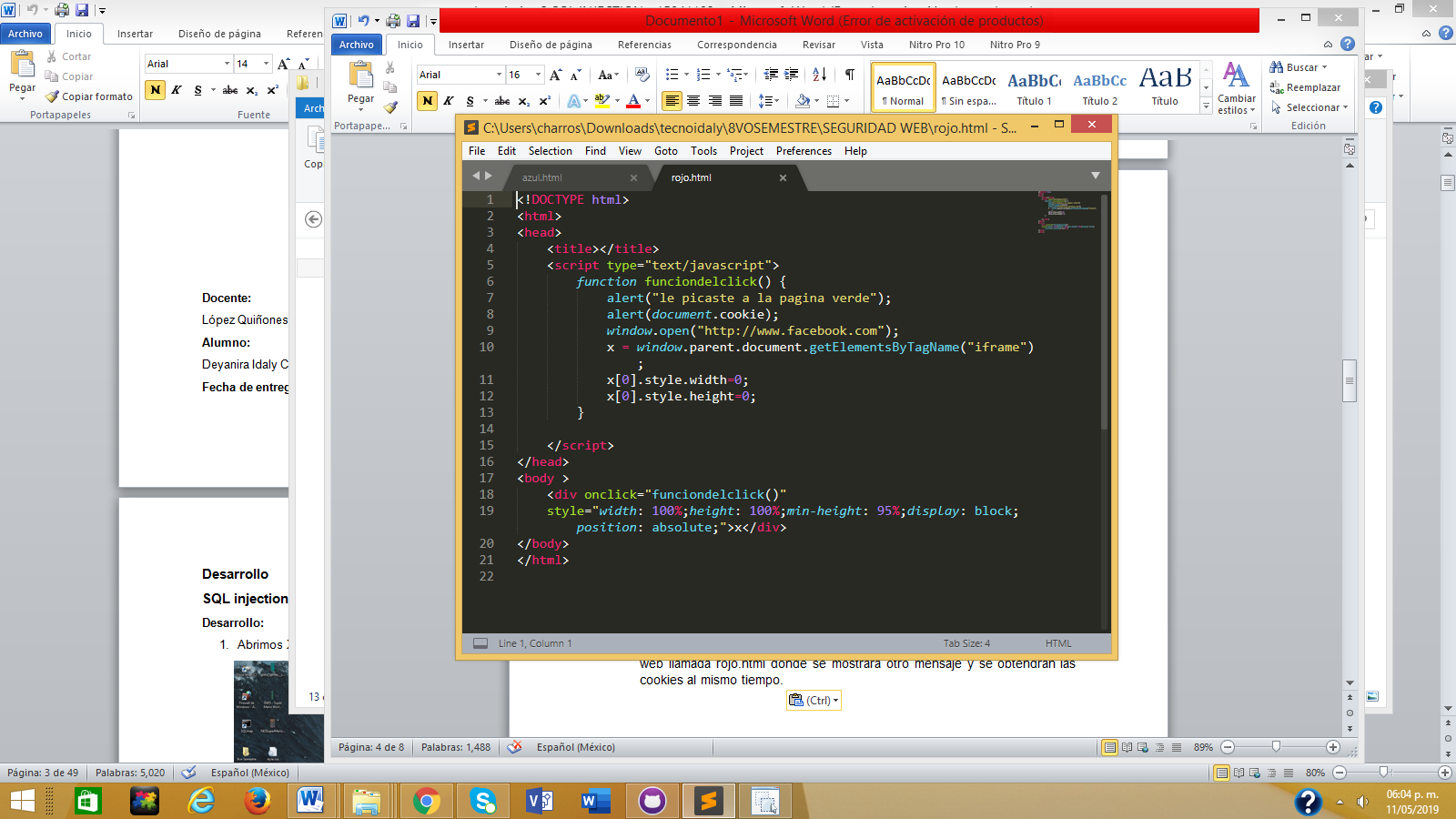
1. Crear una primera página con HTML/CSS y javascript llamada azul.html donde se simule la utilización de un botón, mismo que al dar clic nos deberá mostrar un mensaje diciendo “le picaste a la página azul”. En este html se colocará un iframe que se posicionará en la totalidad de la pantalla y tendrá las propiedades css necesarias para ocultarlo y como una de las funciones de javascript, la obtención de cookies.

La finalidad de dicha página es que al dar clic en dicho botón, como el iframe está “sobre” este botón, por sus propiedades mandará llamar a otra página web llamada rojo.html donde se mostrará otro mensaje y se obtendrán las cookies al mismo tiempo.

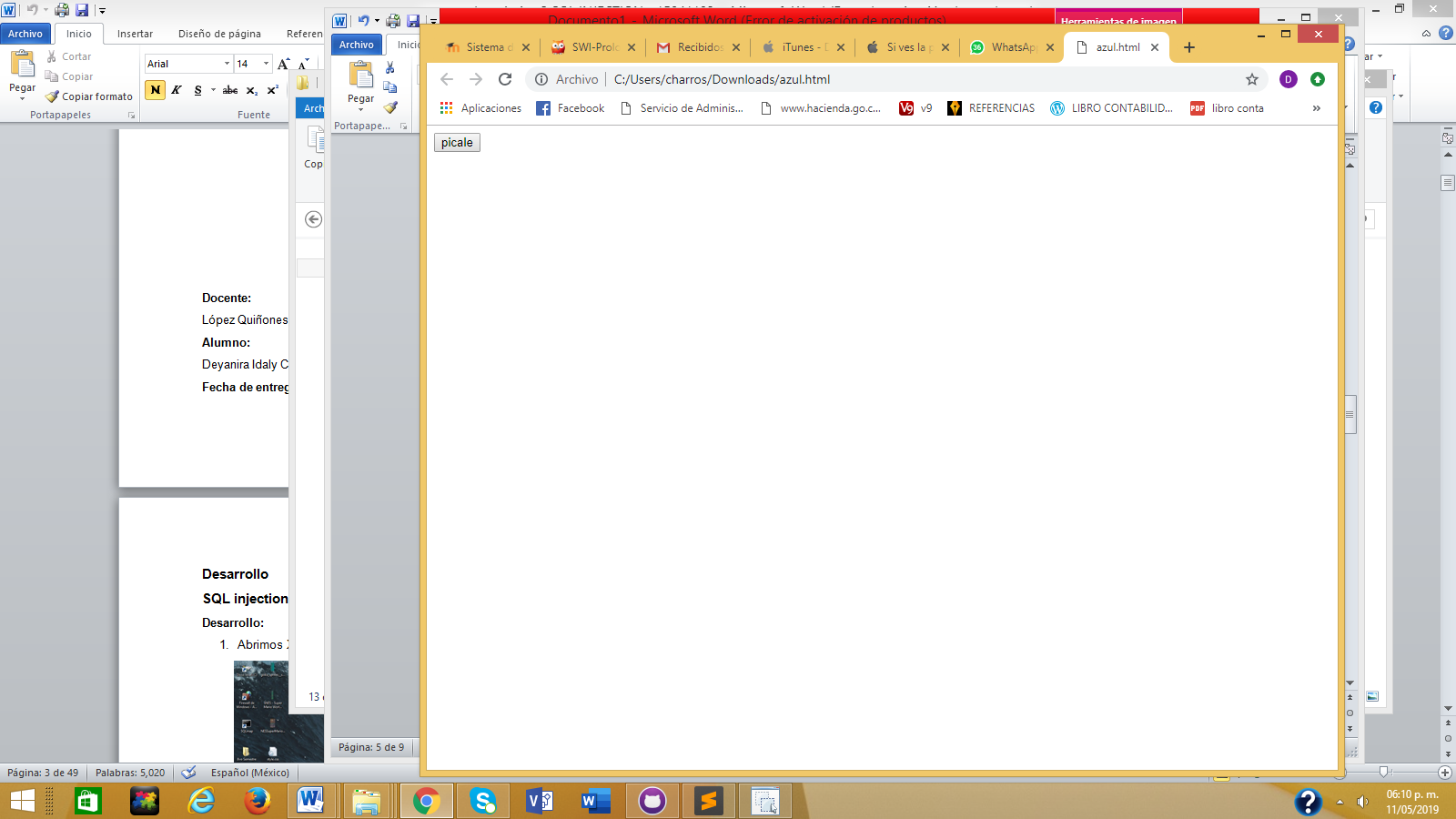


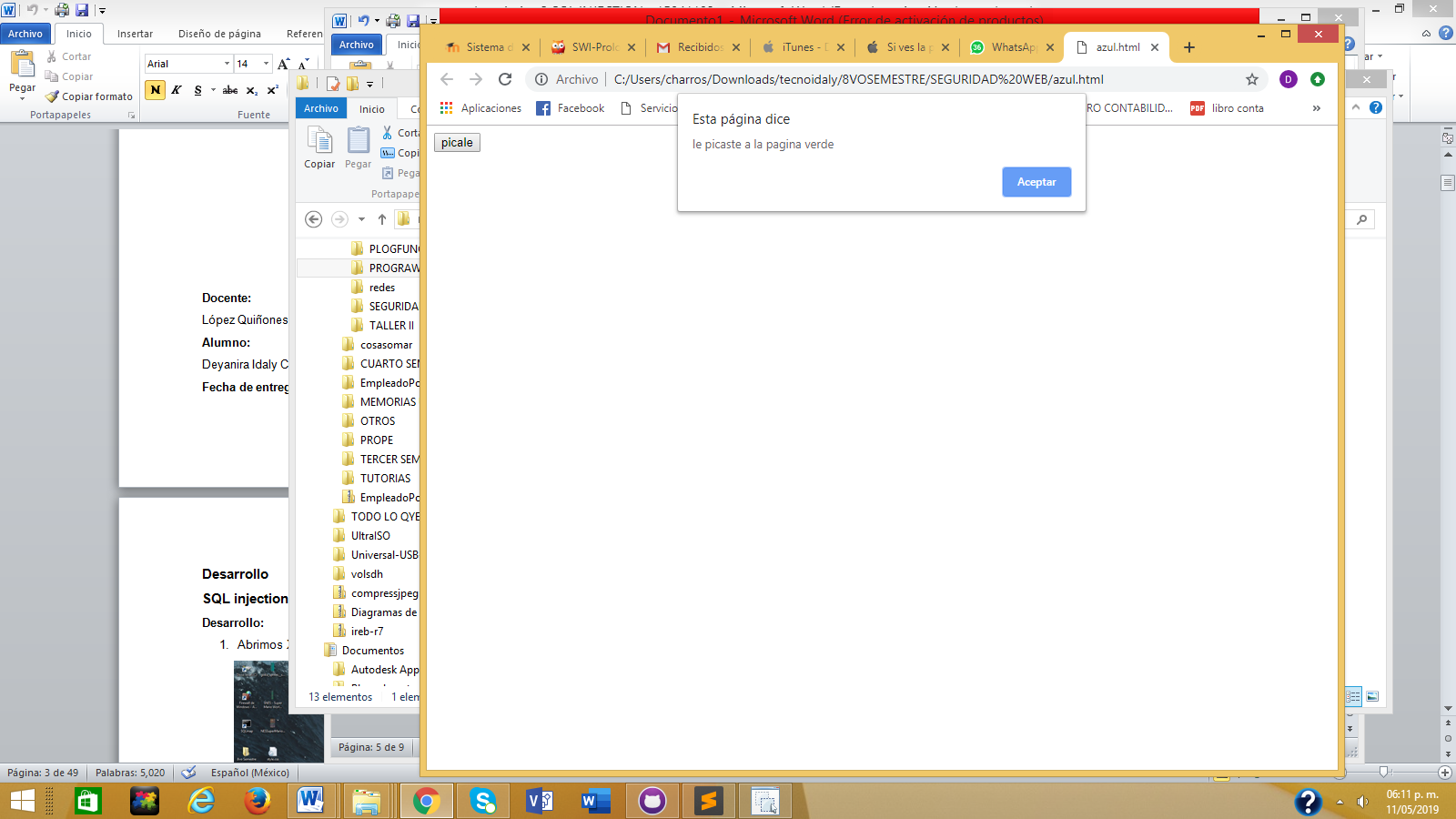
1. Crear una segunda página con HTML/CSS y javascript llamada rojo.html que contenga una función con un alert, la obtención de las cookies y que se redirija a Facebook.com, después de esto el iframe cambie de tamaño a 0 x 0 (se quite , se podría decir). Y dentro del código html, solamente se manda llamar a dicha función y otras propiedades de CSS.

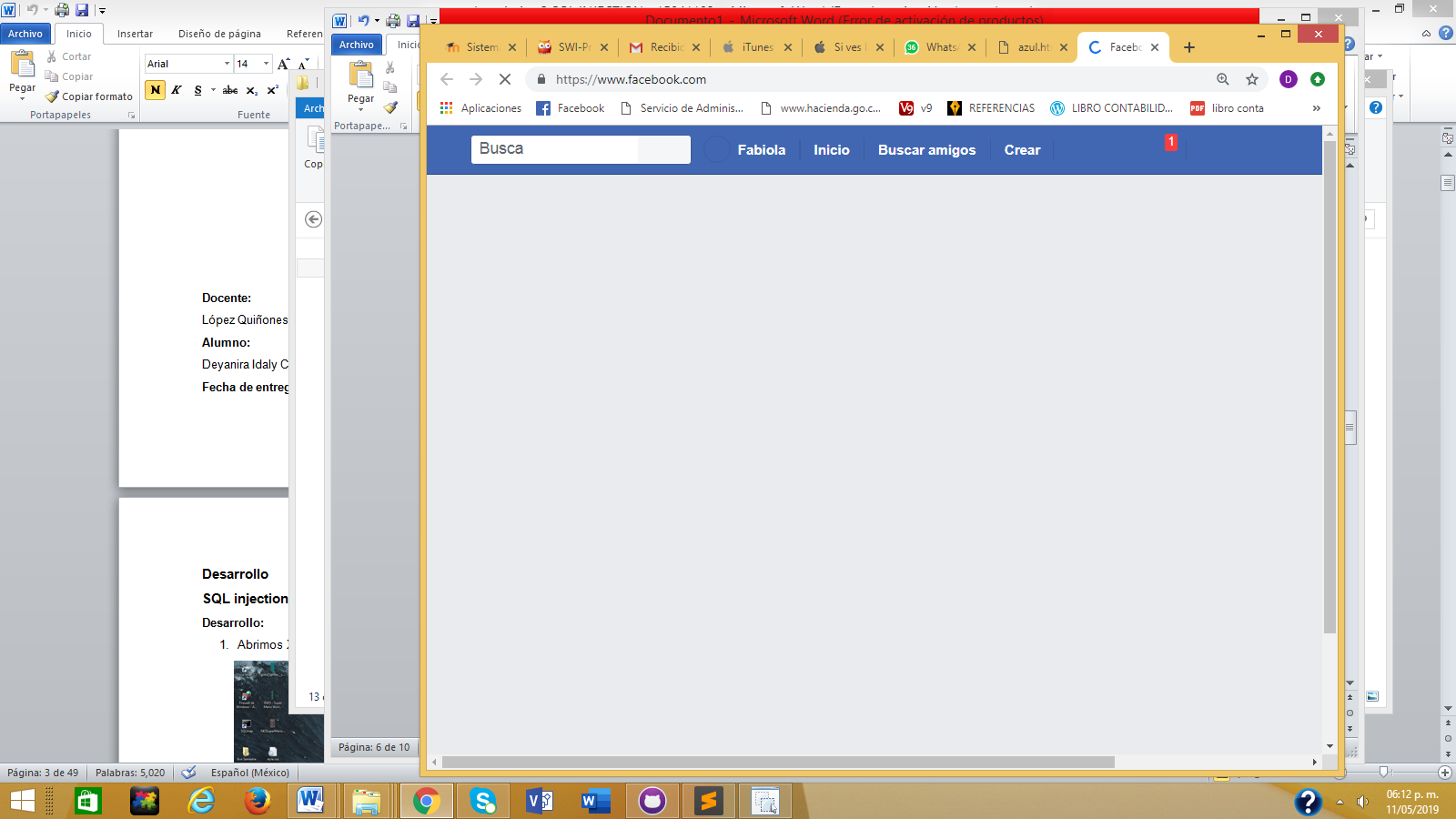
La finalidad de dicha página es que al dar clic en dicho botón de la primer página, como el iframe está “sobre” este botón, por sus propiedades mandará llamar a otra página web llamada rojo.html (esta página) donde se mostrará otro mensaje , se obtendrán las cookies al mismo tiempo y se re direccionará a Facebook, después se “quitara” el iframe y si regresamos a la primer página y volvemos a dar clic en botón ahora si se mostrara el mensaje de “click en la azul” (lo que se debería mostrar en el inicio) y no el engaño que se realizó con el iframe.



**Ejemplificación**







**¿Cómo protegerse?**

**Usuarios**

* **Para usuarios de Firefox**: Es recomendable incluir el complemento NoScript en su última versión, en la cual incorpora una nueva característica llamada ClearClick, cuya finalidad está centrada en impedir que el usuario se convierta en víctima del ClickJacking. Ayuda a interaccionar con dichos elementos permitiendo evaluar el enlace de destino, viendo el texto o la imagen con el enlace real, y el enlace que se estaba anteponiendo (ClickJacking), pudiendo bloquear incluso el ataque de forma permanente para ese sitio.
* **Usuarios de Opera:**
* Deshabilitar todas las opciones en Herramientas -> Avanzado -> Contenidos.
* Escribir "opera:config" en la barra de direcciones. Buscar "Extensions" y deshabilitar "iFrames".
* **Usuarios de navegadores en modo texto (Links, Lynx, w3m...):** No afectados por la vulnerabilidad.
* **Usuario de otros navegadores:** Deshabilitar scripts, plugins y JavaScript en el navegador. Aunque no se necesita para llevar a cabo el ataque, sí lo favorece notablemente.
* **Para todos los usuarios de Adobe Flash:** Para prevenir ser víctima de Clickjacking, también se puede cambiar la configuración de Adobe Flash de la siguiente manera:
* Se accede a las Global Privacy Settings del Adobe Flash Player Settings Manager en la siguiente URL: <http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager02.html>
* Seleccionar la opción Always deny.
* Confirmar.
* Otra opción es deshabilitar la característica de modificación de Hardware, manualmente en el archivo mms.cfg, cambiando el valor “AVHardwareDisable” de 0 a 1. Con esto el administrador del complemento flash no volverá a preguntar si se quiere habilitar la cámara web y el micrófono sino que denegará su uso, a menos que se esté visitando un sitio de confianza (lista blanca). De este modo se presupones que cualquier página que solicite el cambio de configuración de Adobe Flash intenta realizar un ataque de ClickJacking.
* antivirus que detectan en el navegador comportamientos "extraños" de este tipo, y les han asignado firmas. Por ejemplo Avira, lo detecta como HTML/Infected.WebPage.Gen2.

**Desarrollador**

* Dotar a la página de código que evite que la página sea renderizada dentro de un iframe (http://www.quirksmode.org/js/framebust.html). Nota: esta práctica puede no ser recomendable desde un punto de vista de accesibilidad/compatibilidad.
* añadir al protocolo HTTP una cabecera llamada X-Frame-Options. Las páginas que envíen estas cabeceras al navegador, pretenden protegerse de aparecer en un iframe. Se propusieron varios métodos más o menos flexibles para intentar ayudar a las que legítimamente necesitaran incluirse dentro de un iframe. Sus valores son:
  + DENY, el navegador evita que la página sea renderizada si está contenida dentro de un iframe
  + SAMEORIGIN, la página solo puede ser mostrara en un frame que provenga del mismo origen que la propia página.
  + ALLOW-FROM uri, el navegador bloqueará la renderización sólo si el origen de nivel superior está en un contexto de navegación que es diferente al valor uri proporcionado en la directiva

Los usuarios deben confiar en que las páginas las envíen, y que su navegador las interprete correctamente. Esto último hace tiempo que ya lo implementan la mayoría. Por ejemplo, Chrome desde su versión 4.1.249.1042, Firefox desde 3.6.9, Internet Explorer desde la 8, Opera desde la 10.5...

* Enviar los encabezados de respuesta de directivas de marco de ancestros de política de seguridad de contenido (CSP) adecuados que indican al navegador que no permita el encuadre desde otros dominios.
* Emplear código defensivo en la interfaz de usuario para garantizar que el marco actual sea la ventana de mayor nivel.
* Utilizar código JavaScript en la página web, de manera que se esté frustrando la mayoría de los intentos de clickjacking así como otros tipos de ataque que se basan en la utilización de frames dentro de una página web.
* Instalar una herramienta que brinde seguridad a las aplicaciones web aun cuando los equipos de TI no estén activos. Una excelente opción en este caso es la solución Tenable.io. Capaz de proteger tanto aplicaciones web, como activos en la nube y contenedores informáticos.

Esta técnica está relacionada con una vulnerabilidad de Adobe Flash (CVE-2008-4503) debido a que su plugin para los navegadores permite que el usuario cambie sin enterarse las preferencias de seguridad y privacidad si se coloca un objeto de este tipo en la capa frontal transparente, modificando por ejemplo las preferencias de acceso a la cámara web o al micrófono del usuario.

Estas son algunas de las técnicas que utilizan los hackers para realizar *click jacking*:

* Forzar seguidores no solicitados en Twitter u otras redes sociales.
* Crear automáticamente actualizaciones de estado compartiendo enlaces en redes sociales como Facebook o Tuenti.
* Cambiar la configuración de cuentas privadas en redes sociales haciéndolas públicas y visibles a todo el mundo.
* Engañar a los usuarios de manera que se active su cámara y el micrófono de su ordenador a través de una ventana emergente flash.
* Los hackers también se aprovechan de aquellas campañas publicitarias que pagan por cada clic o por cada impresión del anuncio en sus páginas web. Para forzar clics e impresiones utilizan, entre otros, el malware conocido como *DnsChanger* a través del cual los usuarios son redirigidos hacia redes de anuncios infectadas, haciendo que los hackers ganen dinero sin merecerlo y abriendo los ordenadores a posibles infecciones todavía más serias.

**Conclusiones**

En la actualidad existen una gran cantidad de ataques dentro de los sitios web, uno de los ataques que se consideran relativamente nuevos es el click-jacking, este ataque trata de engañar al usuario mediante capas transparentes utilizando propiedades css, para que el usuario al dar clic en un sitio en particular se re dirija a otro sin darse cuenta y de esta manera poder obtener infinidad de información sobre él.

Es un ataque muy común y efectivo, si podría decirse así, porque no es fácil detectarlo a simple vista y la mayoría de los usuarios que navega por internet suele toparse con ellos.

A veces se puede hacer mal uso de esta vulnerabilidad para el usuario, como cambiar contraseñas, obtener sesiones, publicidad, etc.

Existen gran cantidad de recomendaciones para que este tipo de ataques sean evitados como las mencionadas anteriormente, esto es muy eficiente si se aplican de forma correcta.

Con esta práctica una vez más se puede comprender que es muy fácil ser engañado hoy en día en la red, pero también es fácil estar informado y conocer sobre el tema para protegerse, todo esto sirve para ampliar conocimientos y sobre todo obtener nuevos que nos permitan seguir creciendo en el ámbito de la seguridad.

**Bibliografía**

Incibe-cert.es(S.F).avisos de seguridad sobre clickjacking. Recuperado el 9 de mayo del 2019 de :<https://www.incibe-cert.es/alerta-temprana/avisos-seguridad/clickjacking>

Blog.elevenpaths.com(Diciembre 9,2018).Ejemplos y defensas contra el clickjacking. Recuperado el 9 de mayo del 2019 de: <https://blog.elevenpaths.com/2013/10/algunos-ejemplos-y-defensas-contra-el.html>

Backtrackacademy.com(S.F). Que es el clickjacking. Recuperado el 9 de mayo del 2019 de :<https://backtrackacademy.com/articulo/que-es-el-clickjacking>

Registrodominiosinternet.es(S.F).Clickjacking que es y como protegerse. Recuperado el 9 de mayo del 2019 de : <https://www.registrodominiosinternet.es/2012/05/clickjacking-que-es-como-protegerse.html>

Gb-advisors.com(S.F). Clicjacking. Ui redressing. Recuperado el 9 de mayo del 2019 de :<https://www.gb-advisors.com/es/clickjacking-ui-redressing/>