ANALISIS DE VULNERABILIDADES WEB

OWASP

The Open Web Application Security Project

Todo aquello que asumas que el usuario nunca hará, será lo primero que haga cuando entre a tu Sitio Web

Acerca de mi.

OWASP
The Open Web Application Security Project

- **Jaime Iván Mendoza Ribera**
- **Administrador de WHM Cpanel en COTAS.ltda**
- **Pentesting en IT-Forensic.**
- **Parte del Proyecto THD en Bolivia.**
- **Email: jaime981@hotmail.com**

OWASP
The Open Web Application Security Project

TEMAS

- INTRODUCCION
- TIPOS PAGINAS WEB
- VULNERABILIDAD
- ¿POR QUÉ HACKEAN LOS SISTIOS WEB?
- Y QUE METOLOGIA UTIIZAMOS EN UN PENTESTING
- ALGUNA DE LAS HERRAMIENTAS UTILIZADAS
- VULNERABIIDADES Y LOS FALSOS POSITIVOS....
- REPORTAR VULNERABILIDADES
- XML EXTERNAL ENTITY XXE

OWASP
The Open Web Application Security Project

INTRODUCCIÓN:

Una de las principales preocupaciones cuando estamos al cargo de sistema de información o servicios publicados en internet, es si los mismo se encuentran correctamente protegidos.

Hoy en día como muchos de nosotros tenemos conocimiento que en estos tiempos el Análisis de Aplicaciones Web juega un papel muy importante al hacer una Evaluación de la Seguridad y/o Penetration Testing, ya que esta nos brinda la información adecuada acerca de la aplicación web, como por ejemplo el tipo de Plugin que utiliza, tipos de CMS ya sea Joomla - WordPress u otros.

OWASP
The Open Web Application Security Project

TIPOS PAGINAS WEB

SEGÚN SU CONSTRUCCION

- ESTATICAS
- DINAMICAS

SEGÚN SU TECNOLOGIA

- HTML
- ASP
- PHP
- JSP
- Ruby on Rails

- Personales
- Corporativas
- Comercio
- Portales
- Educación
- Turismo
- Instituciones
- Informativos
- Servicios

REF: http://www.ajaxperu.com/paginas-web/tipos-paginas

OWASP
The Open Web Application Security Project

VULNERABILIDAD.

Definimos Vulnerabilidad como debilidad de cualquier tipo que compromete la seguridad del sistema informático.
Las vulnerabilidades de los sistemas informáticos las podemos agrupar en función de:

Diseño

Debilidad en el diseño de protocolos utilizados en las redes.
Políticas de seguridad deficientes e inexistentes.

Implementación

Errores de programación.
Existencia de "puertas traseras" en los sistemas informáticos.
Descuido de los fabricantes.

Uso

Mala configuración de los sistemas informáticos.
Desconocimiento y falta de sensibilización de los usuarios y de los responsables de informática.
Disponibilidad de herramientas que facilitan los ataques.
Limitación gubernamental de tecnologías de seguridad.

OWASP
The Open Web Application Security Project

Vulnerabilidad del día cero

Se incluyen en este grupo aquellas vulnerabilidades para las cuales no existe una solución "conocida", pero se sabe como explotarla.

Vulnerabilidades conocidas

- **Vulnerabilidad de Inyección de código SQL (sqlinjection)**
- **Vulnerabilidad de Inyección de Commandos**
- **Vulnerabilidad de desbordamiento de buffer.**
- **Vulnerabilidad de Cross Site Scripting (XSS).**
- **Vulnerabilidad de denegación del servicio**
- **Vulnerabilidad de ventanas engañosas (Window Spoofing).**
- **Vulnerabilidad de Xml Entity Externa XXE**

OWASP
The Open Web Application Security Project

¿Por qué Hackean los Sitios Web?

El hackeo de sitios web es un problema mucho más común de lo que la mayoría de la gente se imagina, es algo que a cualquier sitio web le puede pasar y es difícil de evitar aún cuando se tomen medidas para prevenirlo. Lamentablemente el cine ha pintado una imagen totalmente irreal de lo que es un hacker y lo que le hacen a un sitio web, es por eso que cuando le pasa a un sitio común y corriente los dueños siempre se preguntan ¿Por qué hackearon mi sitio web?.

Tristemente este problema es muy común y es difícil de evitar. Sin embargo existen muchas medidas de seguridad que se pueden seguir para tratar de minimizar el riesgo

- IDS/IPS
- WAF
- Hardening PHP.INI / Servidor Web
- Contraseña Seguras
- Revisa Actualizaciones
- Escanea en Busca de Fallos
- Programación Segura conociendo OWASP

OWASP
The Open Web Application Security Project

Y QUE METOLOGIA UTIIZAMOS EN UN PENTESTING..

- Reconocimiento
- Escaneo
- Enumeración
- Acceso
- Mantenimiento

Este no es un criterio único, pueden exisitir otras metodologías

OWASP
The Open Web Application Security Project

SON FIABLES LOS ANALIZADORES AUTOMATICOS ??

Por escáner Web se entiende cualquier programa con la capacidad de analizar la seguridad de una aplicación o pagina web

Últimamente, la tendencia en materia de seguridad es emplear , cada vez mas , herramientas para facilitar la ardua tarea de verificar el nivel de seguridad real de una aplicación web.

VENTAJAS

Reducción de costes
Disponibilidad y Automatización
No son habilidades especificas de Seguridad

DESVENTAJAS

Gran cantidad de Falsos Positivos
Imposibilidad de encontrar 0 Day en diseño
Problemas con vulnerabilidad de aplicaciones conocidas.

OWASP
The Open Web Application Security Project

ALGUNA DE LAS HERRAMIENTAS MAS UTILIZADAS ...

- Nmap
- CMS identificacion
- BlidElephant
- WhatWeb
- CMS explorer
- Waffit
- UA-Tester
- Xssed
- Revhosts
- Dirbuster
- Web Crawler
- Nessus
- Vega
- w3af

- Joomscan
- Sqlmap
- Nikto
- Fimap
- TheHarvester
- Uniscan
- Wapiti
- Weevely
- Havij
- WPscan
- DPscan
- SSh shadon security scanner
- Foca
- Bjommla

OWASP
The Open Web Application Security Project

VULNERABIIDADES Y LOS FALSOS POSITIVOS....

En el Análisis de Vulnerabilidades es muy frecuente que se realicen varios test / scanner con diferentes herramientas para llevar a cabo una eliminación de falsos positivos (especialmente durante Auditorias de Seguridad a aplicación WEB). Una vez tienes el resultado, se verifican manualmente los resultados con objeto de ofrecer un informe con el mínimo número de falsos positivos posibles.

proyecto Open Source (WAVSEP) que se encarga precisamente de evaluar herramientas de seguridad, aplicando varias pruebas sobre entornos vulnerables.

El ultimo informe [WAVSEP 2013/2014], se analizan múltiples herramientas de seguridad tanto comerciales como open source;

			WIVET	SQLi	RXSS	LFI	RFI	Redirect	Backup
4		W3AF	Accuracy	19%	35.29%	37.88%	57.48%	16.67%	63.33%	22.83%
			False Positive		30.0%	0.0%	12.5%	16.67%	11.11%	0.0%

Audit Features	Input Vectors	WebApp Scanner	Flash Scanner	CGI Scanner	WebService Scanner
23	8	✔	✘	✔	✘

Consultant	Enterprise	Any
Seat/Year	Seat/Year	Website/Year
0.0$	0.0$	0.0$
Seat/Perpetual	Seat/Perpetual	Website/Perpetual
0.0$	0.0$	0.0$

Curiosamente, w3af no sale muy bien parado en éste análisis, y si además, puedo confirmar que no detecto una vulnerabilidad del tipo XSS, lo que me lleva a descartar la herramienta y buscar otra que sea open source

			WIVET	SQLi	RXSS	LFI	RFI	Redirect	Backup
19		Vega	Accuracy	50%	100.0%	100.0%	94.12%	100.0%	✘	✘
			False Positive		20.0%	0.0%	62.5%	0.0%	✘	✘

Audit Features	Input Vectors	WebApp Scanner	Flash Scanner	CGI Scanner	WebService Scanner
11	3	✔	✘	✘	✘

Consultant	Enterprise	Any
Seat/Year	Seat/Year	Website/Year
0.0$	0.0$	0.0$
Seat/Perpetual	Seat/Perpetual	Website/Perpetual
0.0$	0.0$	0.0$

En la puntuación (score) del brenchmark, VEGA (la herramienta que había utilizado y que ha detectado dicha vulnerabilidad XSS), obtiene muy buen resultado (0% falsos positivos en XSS reflected), si a esto le unimos que w3af obtenía también algunos otros falsos positivos.

wapiti, una herramienta de línea de comando escrita en python, múltiples formatos de salida y que además obtiene mejor puntuación en el brechmark que w3af.

14		Wapiti		WIVET	SQLi	RXSS	LFI	RFI	Redirect	Backup	Consultant	Enterprise	Any
			Accuracy	44%	100.0%	66.67%	51.47%	57.41%	✘	4.35%	Seat/Year	Seat/Year	Website/Year
			False Positive		20.0%	42.86%	12.5%	0.0%	✘	100.0%	*0.0$*	*0.0$*	*0.0$*
											Seat/Perpetual	Seat/Perpetual	Website/Perpetual
											0.0$	*0.0$*	*0.0$*

Audit Features	Input Vectors	WebApp Scanner	Flash Scanner	CGI Scanner	WebService Scanner
15	*3*	✔	✔	✘	✘

Conclusion Al final ninguna herramienta es la “ poderosa” .

OWASP
The Open Web Application Security Project

GRAMPUS **Project nace para todos aquellos usuarios que necesitan automatizar sus procesos en auditorias web. Como sabemos la recopilación de información a la hora de realizar un ataque a un objetivo es esencial y a veces este proceso puede ser muy largo y pesado.**

ANUBIS, herramienta de ayuda en auditorías de seguridad, en el footprinting y el fingerprinting

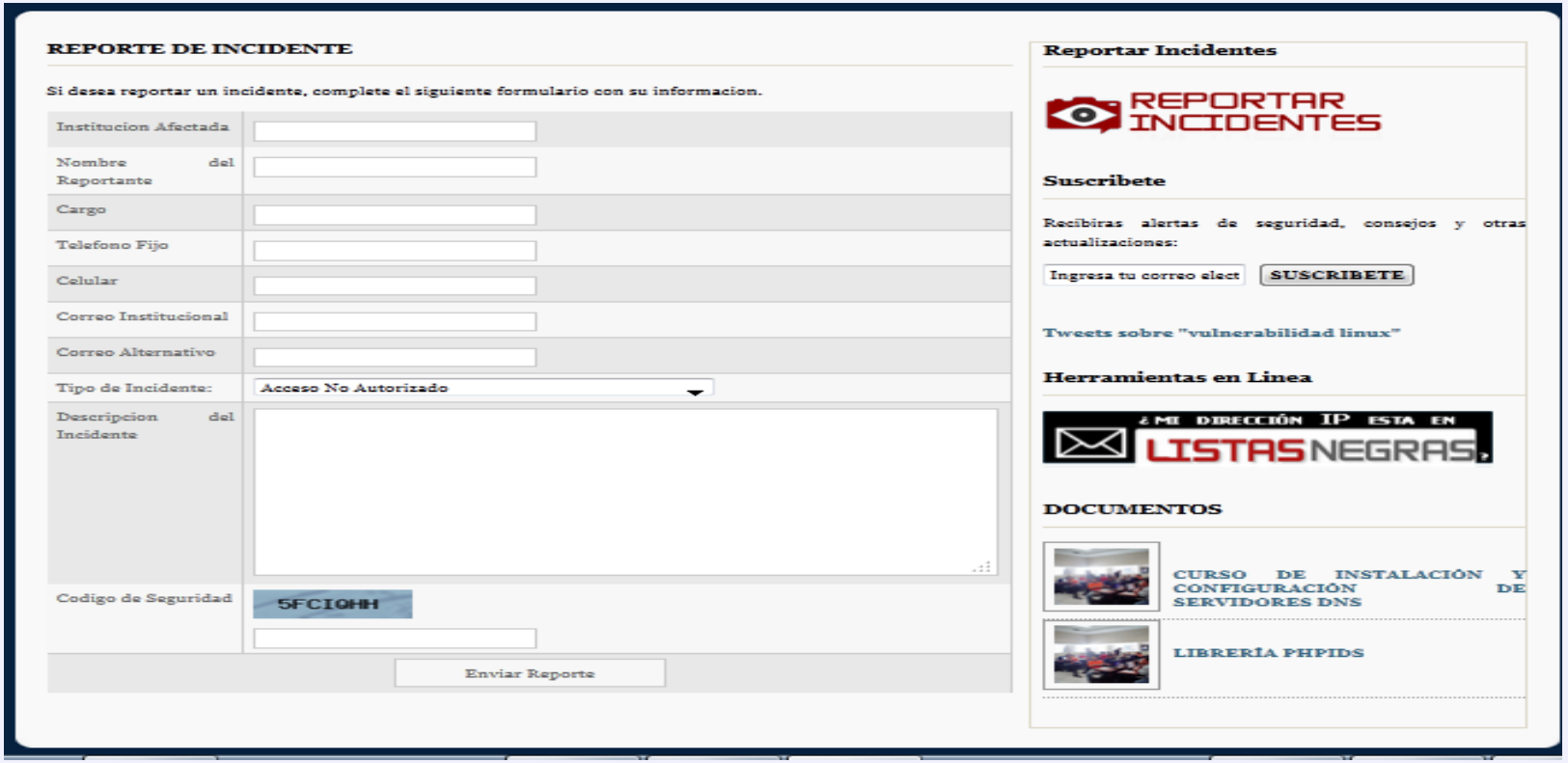
WEB SORROW **que anda en búsqueda de información sensible que nos permita encontrar algunas cosas interesantes que a los desarrolladores que instalaron se les haya pasado así también detectar la versión de un CMS, identificación y enumeración como su principal objetivo.**

OWASP
The Open Web Application Security Project

REPORTAR VULNERABILIDADES ??

contactando al propietario del Whois. ó

http://www.csirt.gob.bo/reportar.php



OWASP
The Open Web Application Security Project

Xml eXternal Entity XXE

OWASP
The Open Web Application Security Project

Formatos Open XML y extensiones de nombres de archivo

A partir de 2007 Microsoft Office system, Microsoft Office usa los formatos de archivo basados en XML, como .docx, .xlsx y .pptx. Estos formatos y las extensiones de los nombres de archivo se aplican a Microsoft Word, Microsoft Excel y Microsoft PowerPoint. En este artículo, se describen las principales ventajas del formato, así como las extensiones de los nombres de archivo, y se explica cómo compartir archivos de Office con personas que usan versiones anteriores de Office.

OWASP
The Open Web Application Security Project

VENTAJAS ????

- Archivos Compactos
- Mejoras en la recuperacion de archivos dañados
- Mayor privacidad y mas control de la informacion personal
- Mejor integracion e interoperabilidad de los datos
- Mayor facilidad para la deteccion de documentos que continen macros
- Fácilmente procesable tanto por humanos como por software.
- Separa radicalmente la información o el contenido de su presentación o formato.
- Diseñado para ser utilizado en cualquier lenguaje o alfabeto.
- Su análisis sintáctico es fácil debido a las estrictas reglas que rigen la composición de un documento.

OWASP
The Open Web Application Security Project

Desventajas !!!

La posibilidad de construir sistemas acordes a nuestras necesidades para el intercambio de datos podría llevarnos a la proliferación de versiones incompatibles y si esto llegase a suceder, entonces la solución que plantea el XML ante la búsqueda de intercambio universal de información, lo llevaría a su opuesto; en vez de unificar todo un lenguaje, nos encontraríamos con lenguajes muy específicos y cada vez más alejados de la "universalidad".

-sintaxis XML es redundante o grande en relación a las representaciones binarias de datos similares.
---La redundancia puede afectar la eficiencia de aplicación a través de mayores de almacenamiento, transmisión y costes de
---sintaxis XML es demasiado prolijo en relación con otra alternativa "basada en texto con los formatos de transmisión de datos

OWASP
The Open Web Application Security Project

Jugando con XXE (Xml eXternal Entity)

XXE (Xml eXternal Entity) es un tipo de vulnerabilidad con el que hace mucho tiempo tenía ganas de jugar. En este post voy a explicar lo más básico de esta vuln con el propósito de entender su funcionamiento.

XXE es un fallo que se produce en aplicaciones que hacen uso de "parsers" XML. Es decir aplicaciones que reciben como entrada un documento XML y para procesarlo hacen uso de alguna librería de parseo como LibXML, Xerces, MiniDOM, etc. El atacante entonces puede enviar un documento XML especialmente manipulado para conseguir que el parser XML divulgue información del sistema, consuma recursos en exceso, ejecute comandos u otras formas de explotación.

OWASP
The Open Web Application Security Project

Documentos XML válidos y el DTD

Se dice que un documento XML está bien formado cuando cumple con la estructura definida por el estándar XML: que incluya la especificación de versión, que tenga un único nodo raíz, que cada tag esté correctamente cerrado, etc. Además, se dice que un documento XML es válido, cuando además de estar bien formado cumple con las reglas definidas por el DTD (u otro mecanismo de validación).

Una DTD es un documento que define la estructura de un documento XML: los elementos, atributos, entidades, notaciones, etc, que pueden aparecer, el orden y el número de veces que pueden aparecer, cuáles pueden ser hijos de cuáles, etc. El procesador XML utiliza la DTD para verificar si un documento es válido, es decir, si el documento cumple las reglas del DTD.

OWASP
The Open Web Application Security Project

ESTRUCTURA BASICA DE UN XML

```
<?xml version="1.0" encoding="utf-8"?>
<root>
   <alumno>
      <nombre>Juan</nombre>
      <apellido>Perez</apellido>
      <codigo>1234</codigo>
   </alumno>
</root>
```

OWASP
The Open Web Application Security Project

El documento anterior es un XML bien formado, pero para que sea válido debemos especificar un DTD contra el cual se validará la estructura del XML. El DTD sería algo como esto:

```
<!DOCTYPE root [
<!ELEMENT root (alumno)>
<!ELEMENT alumno (nombre, apellido, codigo)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT apellido (#PCDATA)>
<!ELEMENT codigo (#PCDATA)>
]>
```

Con este DTD indicamos que el nodo raiz es "root", que el nodo "root" tiene un subnodo "alumno", que el nodo "alumno" tiene subnodos "nombre", "apellido" y "codigo" y finalmente que los nodos "nombre", "apellido" y "codigo" contienen datos (es decir no tienen subnodos).

OWASP
The Open Web Application Security Project

Finalmente nuestro documento XML quedará así:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE root [
<!ELEMENT root (alumno)>
<!ELEMENT alumno (nombre, apellido, codigo)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT apellido (#PCDATA)>
<!ELEMENT codigo (#PCDATA)>
]>
<root>
   <alumno>
      <nombre>Juan</nombre>
      <apellido>Perez</apellido>
      <codigo>1234</codigo>
   </alumno>
</root>
```

OWASP
The Open Web Application Security Project

Cuando un parser procese nuestro XML encontrará el DTD y procederá a verificar si la estructura del documento concuerda con las reglas del DTD para concluir si el XML es válido o no lo es.

Entidades XML

Los DTD también nos permiten definir entidades XML (XML Entity). Las entidades XML son "alias" que se substituyen por otro valor previamente definido cada vez que aparecen en el documento XML. Para comprenderlo mejor piensen en la códificación de ciertos carácteres en HTML:

CARACTER	CODIFICACIÓN
©	&copy;
<	&lt;
>	&gt;
&	&amp;

De forma similar, el DTD nos permite definir nuestras propias entidades y usarlas en el documento XML. Por ejemplo:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE root [
<!ELEMENT root (data)>
<!ELEMENT data (#PCDATA)>
<!ENTITY ejemplo "Este es un ejemplo de entidad...">
]>
<root>
   <data>&ejemplo;</data>
</root>
```

En el ejemplo definimos la entidad "ejemplo" con el valor "Este es un ejemplo de entidad...". Luego insertamos la entidad dentro del nodo "data". Si posteriormente le pedimos al parser XML el valor del nodo "data" nos devolverá "Este es un ejemplo de entidad...".

OWASP
The Open Web Application Security Project

Las entidades pueden ser de dos tipos: internas o externas. Las entidades internas son como la que vimos en el ejemplo anterior, su valor se define en el mismo documento XML. Por otra parte, las entidades externas son aquellas cuyo valor se encuentra en un recurso externo (osea otro archivo). En este caso la definición de la entidad incluirá una URL o URI con la referencia al recurso externo. Veamos:

```
<!ENTITY externa SYSTEM "otroarchivo.xml">
```

En el ejemplo se define la entidad "externa" que hace referencia al archivo "otroarchivo.xml". El parser comprenderá entonces que cada vez que en el documento XML aparezca &externa; deberá insertar en esa posición el contenido del archivo "otroarchivo.xml".

OWASP
The Open Web Application Security Project

Un investigador hackea Facebook con un documento de Word

OWASP

The Open Web Application Security Project


https://www.facebook.com/careers/search?q=&location=buenos-aires
Buscar
Busca personas, lugares y cosas
Jaime Ivan
Inicio 20+
Careers at Facebook
Search Jobs
Buenos Aires
Busques en Buenos Aires xD
Careers
By Team
By Location
University
Facebook University for Business
Facebook University for Engineers
Resultados de la búsqueda
Su búsqueda para encontró 5 posibles vacantes abiertas
Descripción
Localización.
Small Business Marketing Communications Associate
Marketing
Buenos Aires, Argentina
Creative Strategist
Marketing
Buenos Aires, Argentina
Communications Manager
Communications & Public Policy
Buenos Aires, Argentina
Client Solutions Manager
Sales & Business Development
Buenos Aires, Argentina
SMB Account Manager, University Grad
Buenos Aires, Argentina

OWASP

The Open Web Application Security Project

Careers at Facebook

Search Jobs | Menlo Park

Communications & Public Policy

Communications Manager

Buenos Aires, Argentina

Facebook was built to help people connect and share, and over the last decade our tools have played a critical part in changing how people around the world communicate with one another. With over a billion people using the service and more than fifty offices around the globe, a career at Facebook offers countless ways to make an impact in a fast growing organization.

Facebook is seeking an experienced Communications Manager to lead the

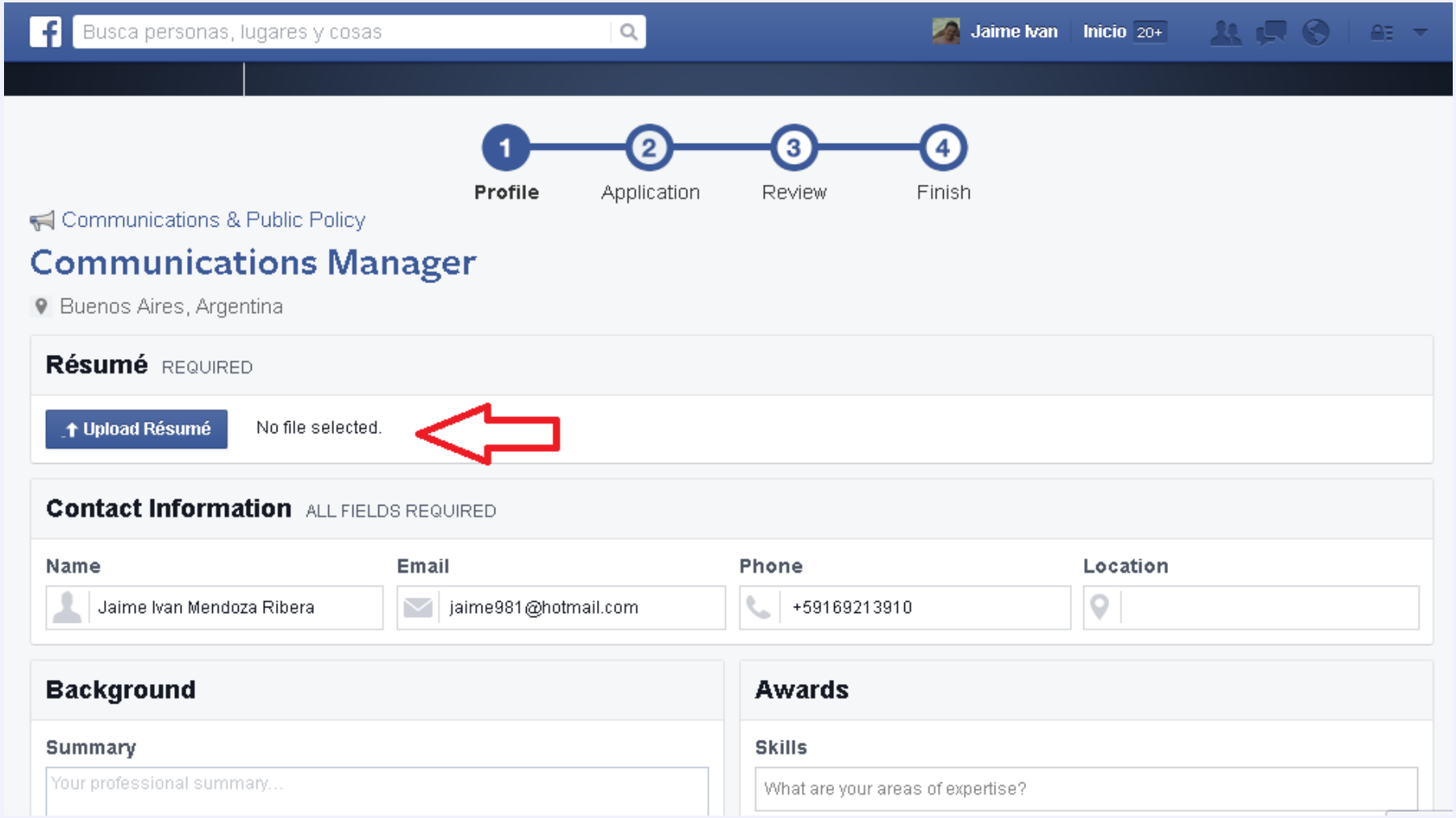
Enviar solicitud

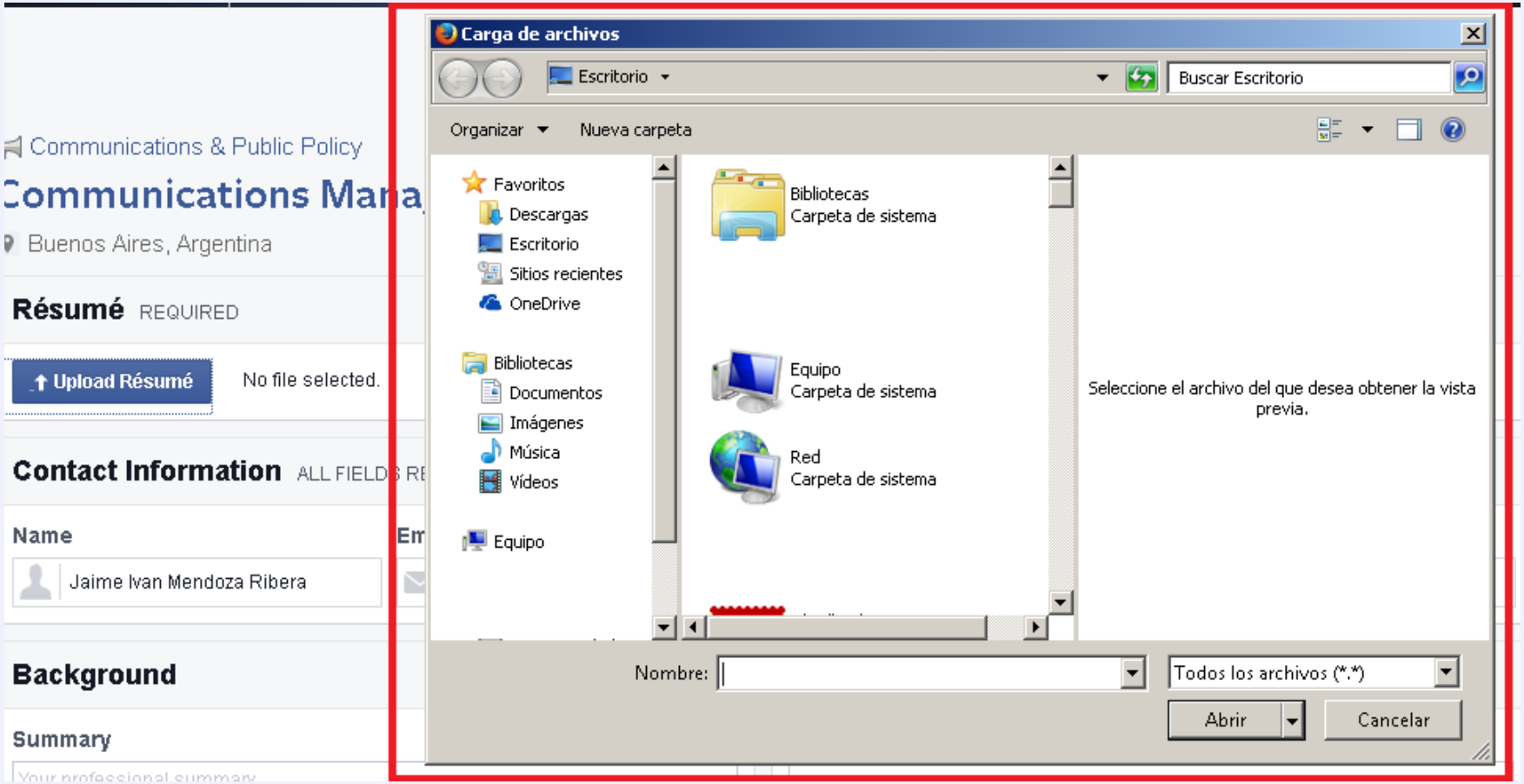
Solo puedes enviar tres solicitudes.

People you might know who work at Facebook.

Otros puestos en Corporate Communications

OWASP
The Open Web Application Security Project




Communications & Public Policy
Communications Mana
Buenos Aires, Argentina
Résumé REQUIRED
Upload Résumé
No file selected.
Contact Information
Name
Jaime Ivan Mendoza Ribera
Background
Summary
Carga de archivos
Escritorio
Buscar Escritorio
Organizar
Nueva carpeta
Favoritos
Descargas
Escritorio
Sitios recientes
OneDrive
Bibliotecas
Documentos
Imágenes
Música
Vídeos
Equipo
Bibliotecas
Carpeta de sistema
Equipo
Carpeta de sistema
Red
Carpeta de sistema
Seleccione el archivo del que desea obtener la vista previa.
Nombre:
Todos los archivos (*.*)
Abrir
Cancelar

OWASP
The Open Web Application Security Project

> “El error ya ha sido solucionado y Facebook ha recompensado al hacker con 5.300 euros”

Por eso Ramadán cogió un .docx cualquiera y lo descomprimió (utilizando la herramienta 7zip) con la intención de acceder a su código y modificarlo. Concretamente, cambió una línea de código para ordenar a ese documento de Word que siempre, estuviera donde estuviera, se comunicara con un fichero gemelo alojado en el ordenador del investigador.

Explotando Xml eXternal Entity

Suficiente teoría... ahora la acción!!

OWASP
The Open Web Application Security Project

Referencias:

http://www.trecebits.com/2015/03/20/un-investigador-hackea-facebook-con-un-documento-de-word/
http://www.pandasecurity.com/spain/mediacenter/redes-sociales/cuidado-con-facebook-un-investigador-ha-logrado-hackearlo-con-un-documento-de-word/
http://www.mclibre.org/consultar/xml/lecciones/xml_dtd.html
http://alguienenlafisi.blogspot.com/2014/03/jugando-con-xxe-xml-external-entity.html
http://fiery-owl.blogspot.com/2014/03/xxe-online.html
https://support.office.com/es-mx/article/Introducci%C3%B3n-a-las-nuevas-extensiones-de-nombres-de-archivo-y-a-formatos-XML-de-Office-eca81dcb-5626-4e5b-8362-524d13ae4ec1?ui=es-ES&rs=es-MX&ad=MX

http://mamaquieroserpentester.blogspot.com/2013/12/fases-de-un-pentesting.html
https://hackersenlared.wordpress.com/category/capacitacion/que-es-un-pentest/