ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

ITIN
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

"DISEÑO E IMPLEMENTACIÓN DE BOTS PARA AUTOMATIZAR TAREAS DE BÚSQUEDA Y ANÁLISIS DE VULNERABILIDADES EN SISTEMAS WEB"

Autores:
Jordy Javier Quinatoa Medina
Julio Javier Villares Jimenez

Director:
Ing. Germán Rodríguez Mgtr.

CONTENIDO:

CAPÍTULO I - INTRODUCCIÓN

- Introducción
- Antecedentes
- Planteamiento del Problema
- Justificación e Importancia
- Sistemas de Objetivos
- Alcance

CAPÍTULO II - MARCO TEÓRICO

- ¿Qué son los sistemas web?
- Vulnerabilidades y amenazas web
- Cross-Site Scripting (XSS)
- Robos de cookies
- ¿Qué es RPA?
- ¿Qué son los sistemas CAPTCHA?

CAPÍTULO III - METODOLOGÍA

- Determinación de la herramienta RPA
- Análisis de vulnerabilidad normal
- Desarrollo del bot para sistemas CAPCTHA
- Desarrollo del bot para XSS

CAPÍTULO IV - RESULTADOS

- Resultados de las soluciones CAPTCHA
- Resultados del bot de CAPTCHA
- Resultados del bot de XSS
- Resultados de la comparación de las herramientas RPA

CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES

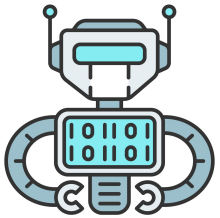
CAPÍTULO I
INTRODUCCIÓN

INTRODUCCIÓN

Muchas empresas y usuarios se han visto en la necesidad de implementar sistemas informáticos basados en una arquitectura cliente-servidor debido a su eficacia y facilidad de acceso. En la mayoría de casos, resulta beneficioso la implementación de sistemas web, sin embargo, pueden estar expuestos a ataques perpetrados por ciberdelincuentes preparados y con sólidos conocimientos.

El uso de sistemas automatizados (bots) se ha intensificado en la integración de los procesos de las compañías y actividades cotidianas de las personas, existen bots que pueden ejecutar desde tareas sencillas hasta manejar toda la lógica de negocio de una empresa. Missouri Enterprise menciona que una organización al integrar bots en sus procesos, puede reducir los costos de un 24% al 50%, cumplir con el tiempo de ciclo de procesos desde el 30% a un 50%

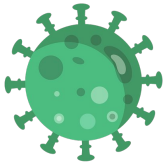
En ese contexto, en este trabajo se desarrollaron bots para automatizar las tareas de búsqueda y análisis de vulnerabilidades en sistemas web, por medio de la utilización de herramientas RPA



RPA bot

ANTECEDENTES

- Según el Informe de Riesgos Globales 2020 del Foro Económico Mundial, menciona que cada minuto existen 100 intentos de explotación de vulnerabilidades a diferentes sistemas informáticos.



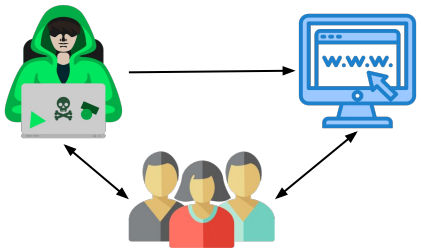
- A raíz de la pandemia del COVID19, los delitos informáticos han aumentado en un 600% a nivel mundial.

- Según el portal Small Business Trends (2020), menciona que de los ataques informáticos de todo el mundo, el 43% fueron dirigidos a pequeñas empresas, de los cuales, solo el 14% lograron mitigar el problema de manera efectiva, mientras que el 29% restante tuvo déficit en la mitigación de riesgos del SI.

- Según el informe de la Asociación Ecuatoriana de Ciberseguridad (2020), menciona que de 100 empresas encuestadas en el Ecuador el 20% gestionan de manera adecuada la seguridad de la información, mientras que el 80% de empresas restantes poseen un manejo inadecuado de la ciberseguridad.

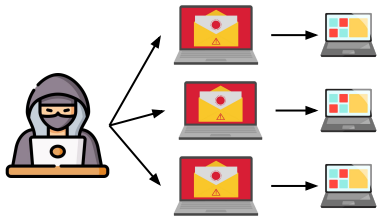
ANTECEDENTES

Principales ataques cibernéticos a sistemas web empresariales se tiene:



TIPO XSS

Según OWASP Top 10 (2021), menciona que los ataques de tipo Cross-site Scripting se encuentran englobados en la categoría de ataques de “Inyección” que ocupa la tercera posición en el top 10 de las vulnerabilidades web, con un índice de incidencia medio de 3,37%.



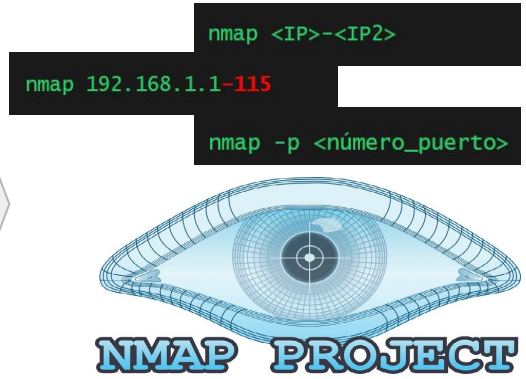
Denegación Servicios (DDoS)

Según una investigación realizada por China y Reino Unido (2018), presentó un modelo de IA capaz de vulnerar los captchas basados en textos de las principales páginas web como Wikipedia, eBay y Microsoft resolviendo el captcha en 0.05 segundos.

El aporte de DARPA, ha generado ideas enfocadas a la creación de sistemas de razonamiento cibernético capaces de detectar vulnerabilidades en los sistemas informáticos.

PLANTEAMIENTO DEL PROBLEMA

Problema 1:
Un inconveniente en realización de la búsqueda de vulnerabilidades dentro de los sitios web, es que se utilizan herramientas que requieren la intervención humana y procesos manuales (en ocasiones repetitivos) para obtener la información y establecer un reporte, lo que afecta el tiempo para dar una respuesta óptima a la vulnerabilidad existente en el sistema web.



Problema 2:
Un inconveniente en los sistemas web es el filtrado de información a través del uso de métodos de inyección de código malicioso. Un elemento que los atacantes optan por vulnerar son las cookies de los sistemas web, ya que son archivos que pueden llegar a contener información sensible de un usuario

Problema 3:
Los sistemas CAPTCHAs permiten contrarrestar los ataques de bots maliciosos, pero hoy en día la implementación de tecnologías como la Inteligencia Artificial y la automatización robótica de procesos, ha hecho que los sistemas CAPTCHAs sean muy fácilmente vulnerados.

JUSTIFICACIÓN E IMPORTANCIA

Justificación 1: Para mejorar la velocidad de detección de las vulnerabilidades web haciendo uso de bots..

Justificación 2: Para ofrecer un modelo de protección de la información del usuario o empresa ante ataques de tipo Cross-site Scripting (XSS).

Justificación 3: Para demostrar la vulnerabilidad en los sistemas CAPTCHAs y que futuras investigaciones se enfoquen en complementar su seguridad

OBJETIVO GENERAL

Diseñar e implementar bots para automatizar tareas de búsqueda y análisis de vulnerabilidades en sistemas web.

OBJETIVOS ESPECÍFICOS

Diseñar e Implementar BOTS para ejecutar tareas de web scraping de páginas web vulnerables a ataques de tipo Cross-Site Scripting (XSS).

Diseñar e Implementar BOTS para ejecutar tareas de web scraping de páginas web vulnerables a ataques de tipo Cross-Site Scripting (XSS).

ALCANCE

El alcance del proyecto se centra en demostrar que las páginas web son vulnerables a ataques XSS por medio del análisis de sus cookies; y la vulnerabilidad de los sistemas CAPTCHA basado en imágenes, por medio de ataques perpetrados por un bot.

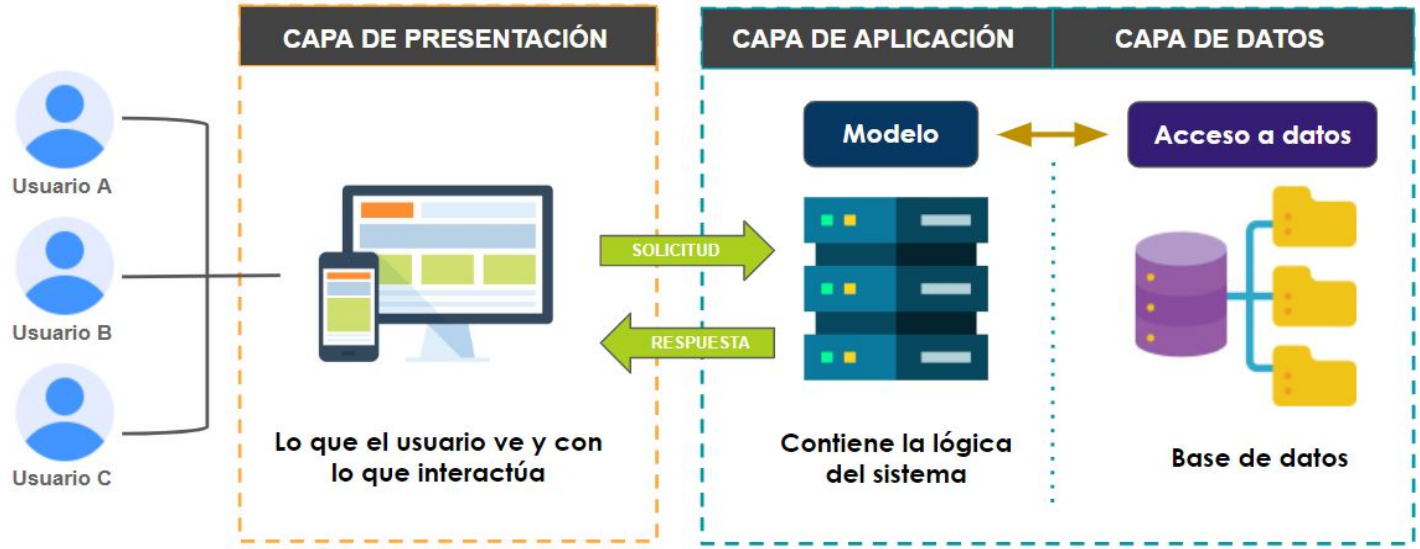
CAPÍTULO II
MARCO TEÓRICO

MARCO TEÓRICO

¿QUÉ SON LOS SISTEMAS WEB?

Según Maldonado (2016), se denominan sistemas web a todas aquellas aplicaciones en las cuales los usuarios pueden acceder a través del internet.

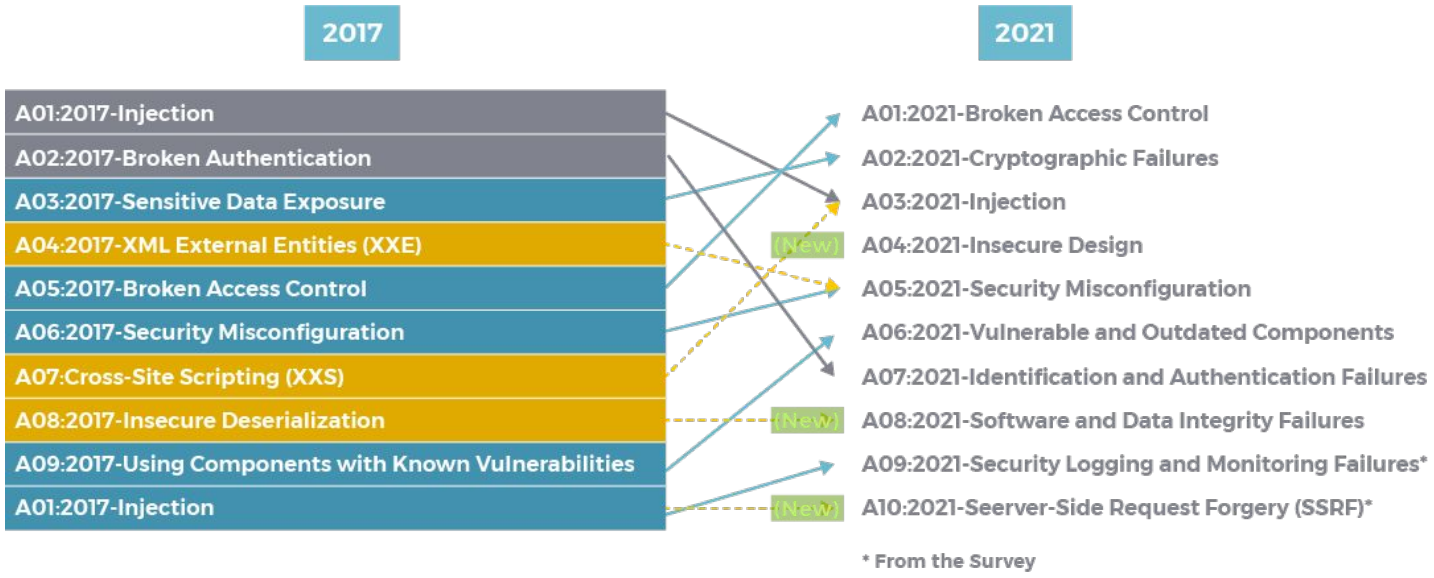
COMPONENTES DE UN SISTEMA WEB Y SU ARQUITECTURA



MARCO TEÓRICO

VULNERABILIDADES Y AMENAZAS DE SISTEMAS WEB

Principales amenazas web según OWASP Top 10 2021



ATAQUES MÁS COMUNES DE LOS SISTEMAS WEB

- Cross-site scripting (XSS).
- Inyección SQL (SQLi).
- Falsificación de solicitudes entre sitios (CSRF).
- Ataques DDoS.

MARCO TEÓRICO

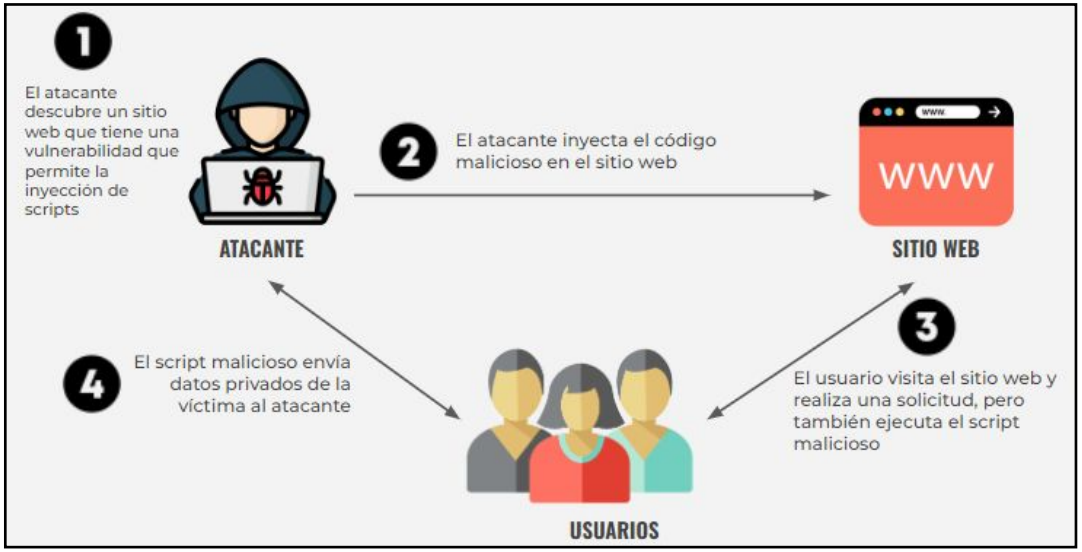
CROSS-SITE SCRIPTING (XSS)

XSS se considera como una vulnerabilidad que aprovecha las fallas de seguridad que permite a los atacantes inyectar scripts ejecutables y maliciosos en una aplicación o sitio web del lado del cliente (Subía, 2018).

Consecuencias de los ataques XSS

- Acceso a las cookies del usuario.
- Robo de credenciales.
- Recopilación de datos personales.
- Redirigir a páginas web maliciosas.
- Acceder al control del equipo de la víctima.
- Cambiar el contenido de un sitio web.
- Ejecutar ataques basados en navegador web.

Funcionamiento de un ataque XSS



MARCO TEÓRICO

Tipos ataques XSS

XSS persistente o almacenado

XSS reflejado

XSS basado en el DOM

MARCO TEÓRICO

Robo de cookies mediante ataques XSS

El atacante por medio de la ejecución de un script del lado del cliente puede obtener acceso de las cookies de la base de datos del navegador (Rodríguez et al., 2019).

Las cookies son archivos de datos que los navegadores y páginas web generan automáticamente en el ordenador del usuario. La mayoría de las cookies contienen información personal del usuario, además permiten una navegación acorde a las preferencias del visitante

TIPOS DE COOKIES

- Cookies de sesión
- Cookies de terceros
- Cookies persistentes
- Cookies seguras
- Cookies propia
- Cookies HttpOnly
- Cookies Zombi

MARCO TEÓRICO

Herramientas para buscar vulnerabilidades en sitios web

Vega	Parsero
WPScan	Burp Suite Community Edition
Joomscan	jSQL injection
Nikto	Websploit

MARCO TEÓRICO

¿QUÉ ES RPA?

La automatización robótica de procesos (RPA) se define como la automatización de procesos manuales de gran volumen, repetitivos, mediante la utilización de robots de software avanzados, también conocidos como "bots".

TIPOS DE RPA

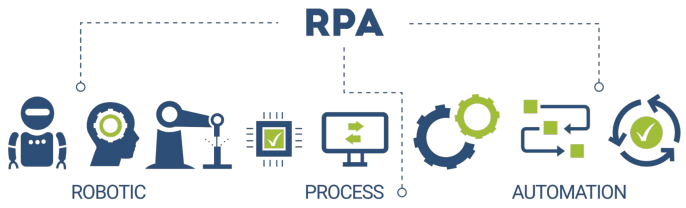
- Automatización RPA asistida
- Automatización RPA no asistida

Beneficios de las RPA

- Reducción de errores.
- Mejora el análisis de datos.
- Ahorro de costes.
- Mejora la productividad del negocio.

HERRAMIENTAS RPA

- UI.Vision
- Robot framework
- TagUI
- Automagica
- UiPath Community Edition



BOTs QUE AUTOMATIZAN

- Software Mayhem
- Mechanical Phish

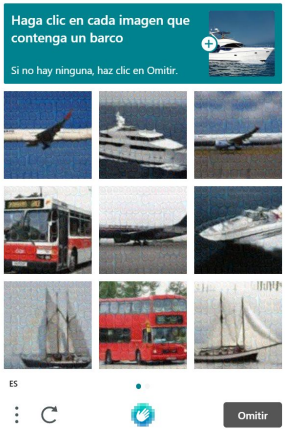
MARCO TEÓRICO

SISTEMAS CAPTCHA

El test de Turing completamente automático y público para diferenciar ordenadores de humanos. Es un desafío que permite determinar si el que intenta ingresar a un sistema es un bot o un humano.

TIPOS DE CAPTCHA

- CAPTCHA basados en texto
- CAPTCHA basados en gráficos
- CAPTCHA basados en audio
- CAPTCHA de problemas matemáticos y lógicos
- CAPTCHA lúdicos
- reCAPTCHA
- hCAPTCHA



MARCO TEÓRICO

ESTADO DE ARTE

La presente Tabla, presente un análisis de los principales estudios enfocados en la detección automática de vulnerabilidades

Código	Título	Link
TR1	Una herramienta inteligente y automatizada de descubrimiento de vulnerabilidades en el WCMS: El estado actual de la web	Link
TR2	Un analizador automático de vulnerabilidades para aplicaciones web	Link
TR3	Ruptura robusta en tiempo real de CAPTCHA de imagen utilizando el modelo Inception v3	Link
TR4	Un ataque de bajo costo contra el sistema hCAPTCHA	Link
TR5	Una extensión del navegador Google Chromium para detectar ataques XSS en sitios web basados en HTML5	Link
TR6	Cookie Scout: Un modelo analítico para la prevención de Cross-Site Scripting (XSS) utilizando un clasificador de cookies	Link

CAPÍTULO III
METODOLOGÍA

METODOLOGíA

Metodología para la Obtención de la Herramienta RPA

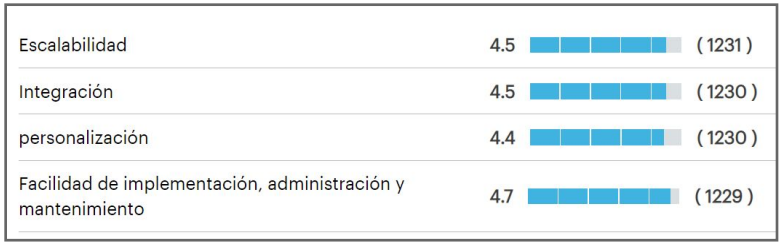
CUADRANTE MÁGICO DE GARTNER (2021)

ONDA DE LÍDERES DE FORRESTER WAVE

METODOLOGíA

Selección del software RPA UiPath en su versión Community Edition

UiPath
Robotic Process Automation



Ediciones de UiPath		
Community	**Enterprise**	
	Enterprise Server	**Enterprise Cloud**
Utilizado por desarrolladores y pequeños equipos que inician su viaje en la automatización.	Utilizado en el despliegue de tareas en empresas y grandes organizaciones.	Utilizado en despliegues empresariales en la nube y empresas de cualquier tamaño.
Versión siempre **GRATIS**	Prueba gratuita por 60 días	Solo por licencia
- 2 módulos Studio para el diseño de la automatización. - 3 robots. - Orquestador alojado en la nube. - Soporte solo por foros y videotutoriales. - Acceso a la academia UiPath.	- Módulos Studios ilimitados para el diseño de la automatización. - Robots ilimitados. - Orquestador on-premises (forma local y servidores de la empresa). - Soporte de primera calidad. - Escala a medida que crece la empresa. - Actualizaciones autogestionadas.	- Módulos Studios ilimitados para el diseño de la automatización. - Robots ilimitados. - Orquestador alojado 100% en la nube. - Soporte de primera calidad. - Escala a medida que crece la empresa. - Actualizaciones constantes, siempre al dia. - Gestión centralizada del acceso de los usuarios.

METODOLOGíA

La principal razón por la que seleccionó la opción de Ui.Vision

- Permite la automatización de procesos basándose en macros
- Es ideal para funcionar en los sitios web más complejos.
- Permite la grabación y reproducción de procesos de manera visual.
- Permite realizar pruebas basadas en la utilización de datos con importación de archivos CSV.

Comparativa del IDE Selenium y UI.Vision IDE

Característica	IDE de Selenium	Ui.Vision RPA Selenium IDE
Implementa todos los comandos importantes de Selenium IDE	Sí	Sí
Código abierto	sí (licencia Apache 2.0)	sí (Licencia GNU AGPL 3.0)
Línea de comandos para programar ejecuciones, ejecutar en una cuadrícula.	selenium-ide-runner	interfaz de línea de comandos
Tomar captura de pantalla	Sí	Sí
Tomar captura de pantalla de página completa	No	Sí
Automatizar descargas de archivos	No	Sí
Exportación de scripts a Java	No	No
Pruebas visuales de la interfaz de usuario	No	Sí
Pruebas de elementos de lienzo	No	Si

METODOLOGíA

Análisis de vulnerabilidades web con herramientas convencionales

Parrot OS

http://redisd.org

```
┌──[root@jordy-qm]─[/home/jordy]
└─── #nikto -h http://redisd.org
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          69.174.114.113
+ Target Hostname:    redisd.org
+ Target Port:        80
+ Start Time:         2022-06-26 16:59:27 (GMT-5)
```

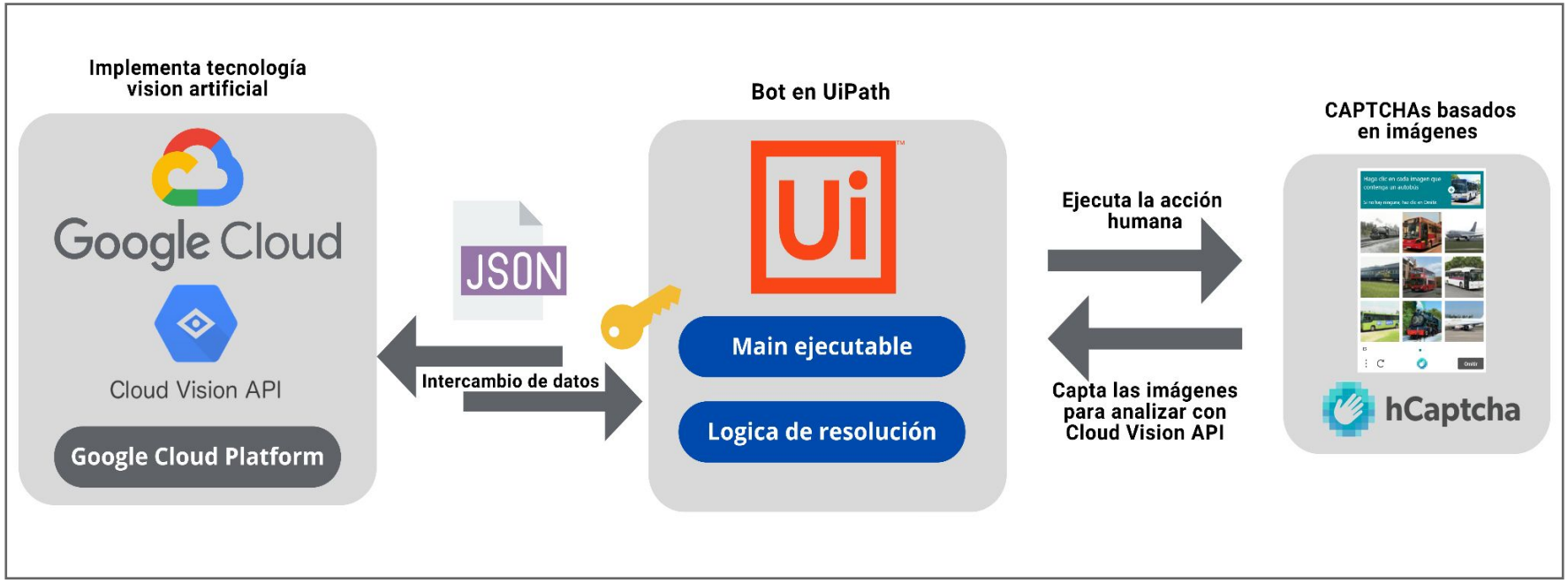
http://pizza.com

```
┌──[✗]─[root@jordy-qm]─[/home/jordy/XSStrike]
└─── #python3 xsstrike.py -u http://pizza.com/?s=test

      XSStrike v3.1.5

[~] Checking for DOM vulnerabilities
[+] Potentially vulnerable objects found
--------------------------------------------------------------------------------
4       document.cookie = "_utms=" + ($(this).attr("data-source") ? $(this).attr("data-source")
: "0RAvNwywfBPp5iVOel_VU0JRqhki1bCK_MfpqqKAh1pvnqbxPPAjas.bwVmpz_eOovvzF4VFTa9PskEgfGtSu_WCWPtyq.JIeUtr5UXMZSepY
hwQHAarxr2nfLazGqC.") + "; path=/";
7       document.cookie = "_utms=" + ($(this).attr("data-source") ? $(this).attr("data-source")
: "0RAvNwywfBPp5iVOel_VU0JRqhki1bCK_MfpqqKAh1pvnqbxPPAjas.bwVmpz_eOovvzF4VFTa9PskEgfGtSu_WCWPtyq.JIeUtr5UXMZSepY
hwQHAarxr2nfLazGqC.") + "; path=/";
3       return decodeURIComponent((new RegExp('[?|&]' + name + '=' + '([^&;]+?)(&|#|;|$)').exec(location.search)
||[,""])[1].replace(/\+/g, '%20'))||null
9   ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga
.js';
--------------------------------------------------------------------------------
[+] WAF Status: Offline
[!] Testing parameter: s
[!] Reflections found: 1
[~] Analysing reflections
[~] Generating payloads
[!] Payloads generated: 3072
```

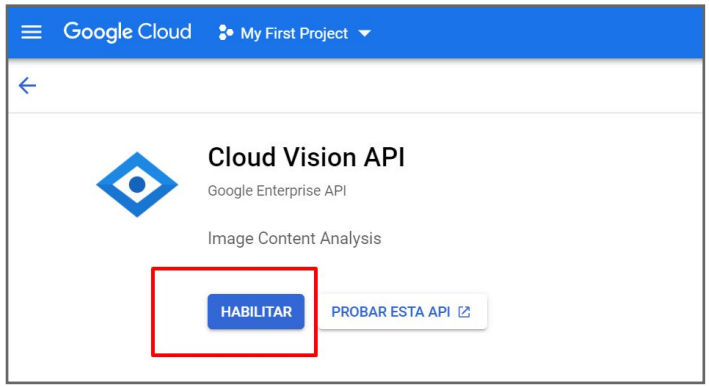
METODOLOGíA

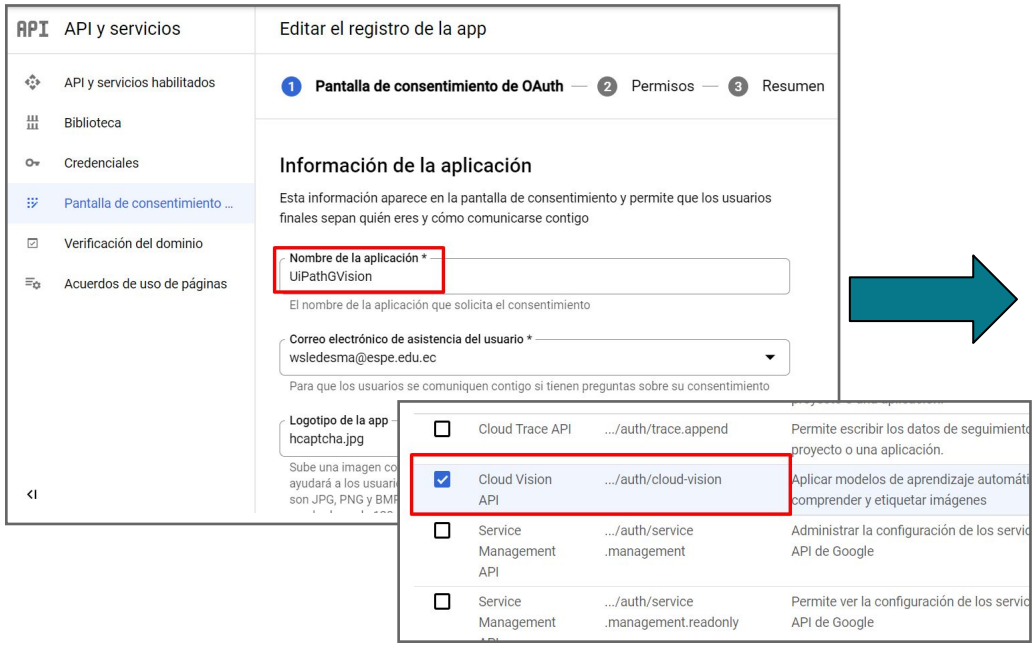
Bot para el análisis de vulnerabilidades en los Sistemas CAPTCHA

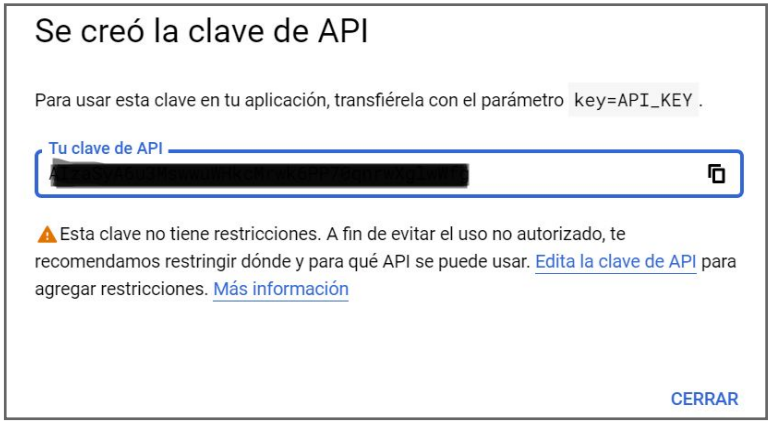
Desarrollo del bot propuesto para vulnerar un sistema CAPTCHA



EN LA API DE GOOGLE CLOUD



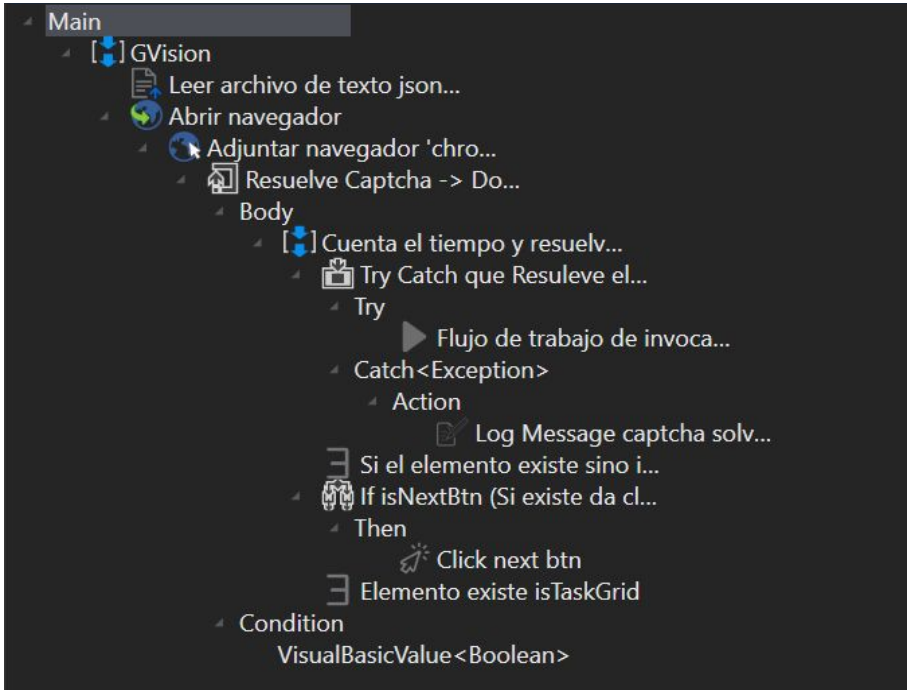




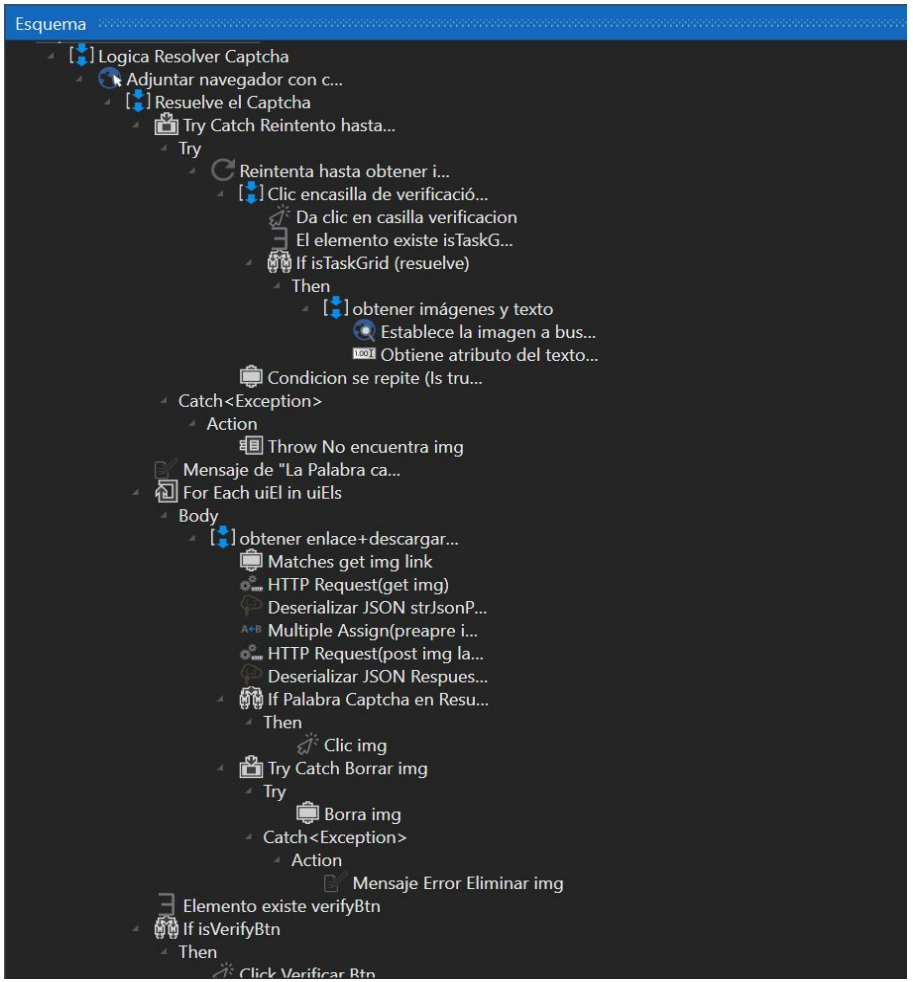
METODOLOGíA

EN LA API DE GOOGLE CLOUD

Desarrollo del *Main.xaml*

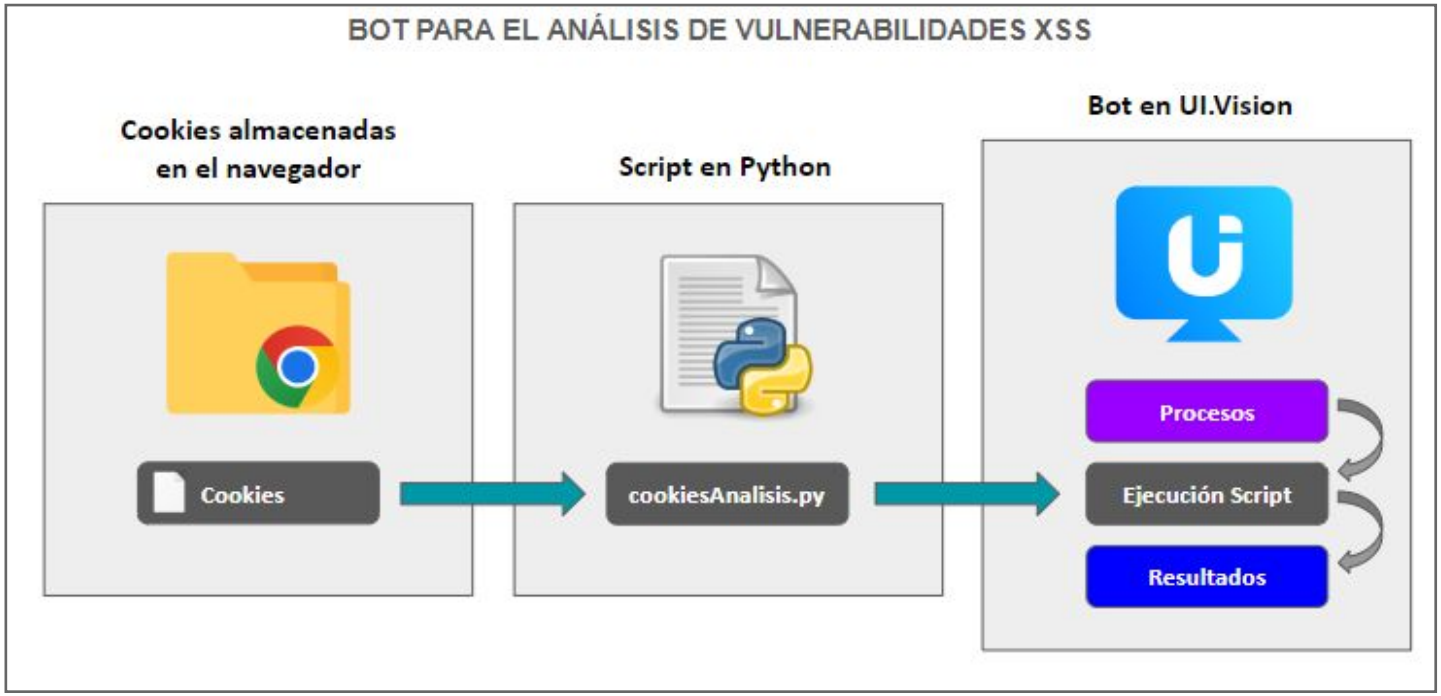


Desarrollo del *CatpchaSolver.xaml*



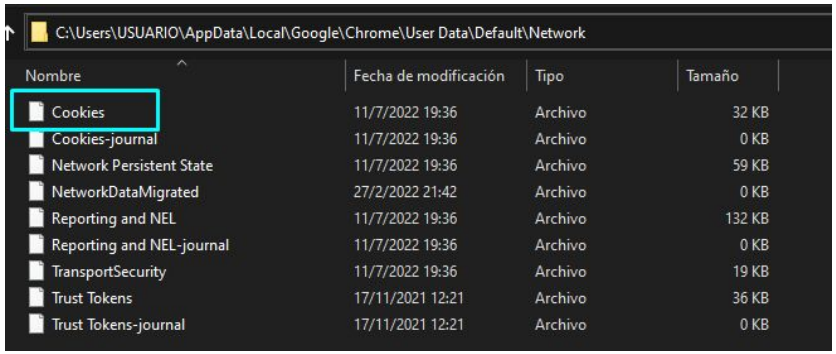
METODOLOGíA

Bot para el análisis de vulnerabilidades de tipo Cross-Site Scripting (XSS)

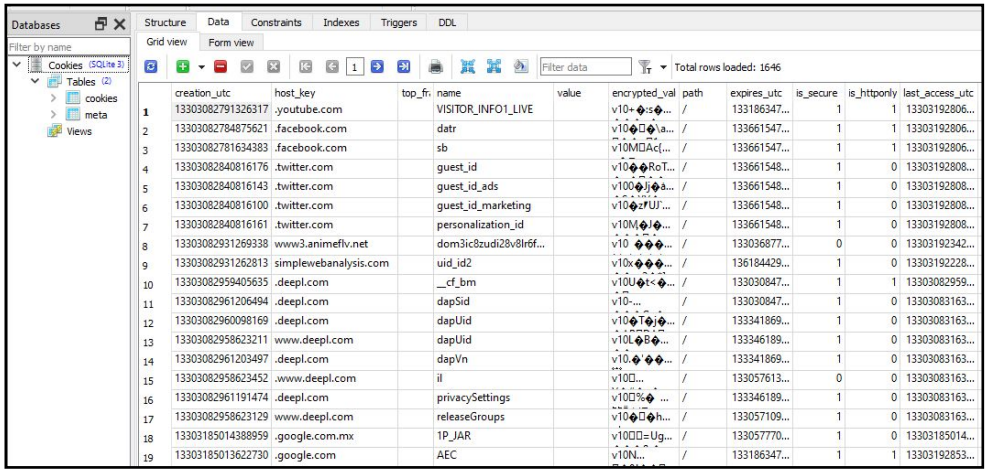


METODOLOGíA

Ubicación del archivo “Cookies”



Contenido del archivo “Cookies”

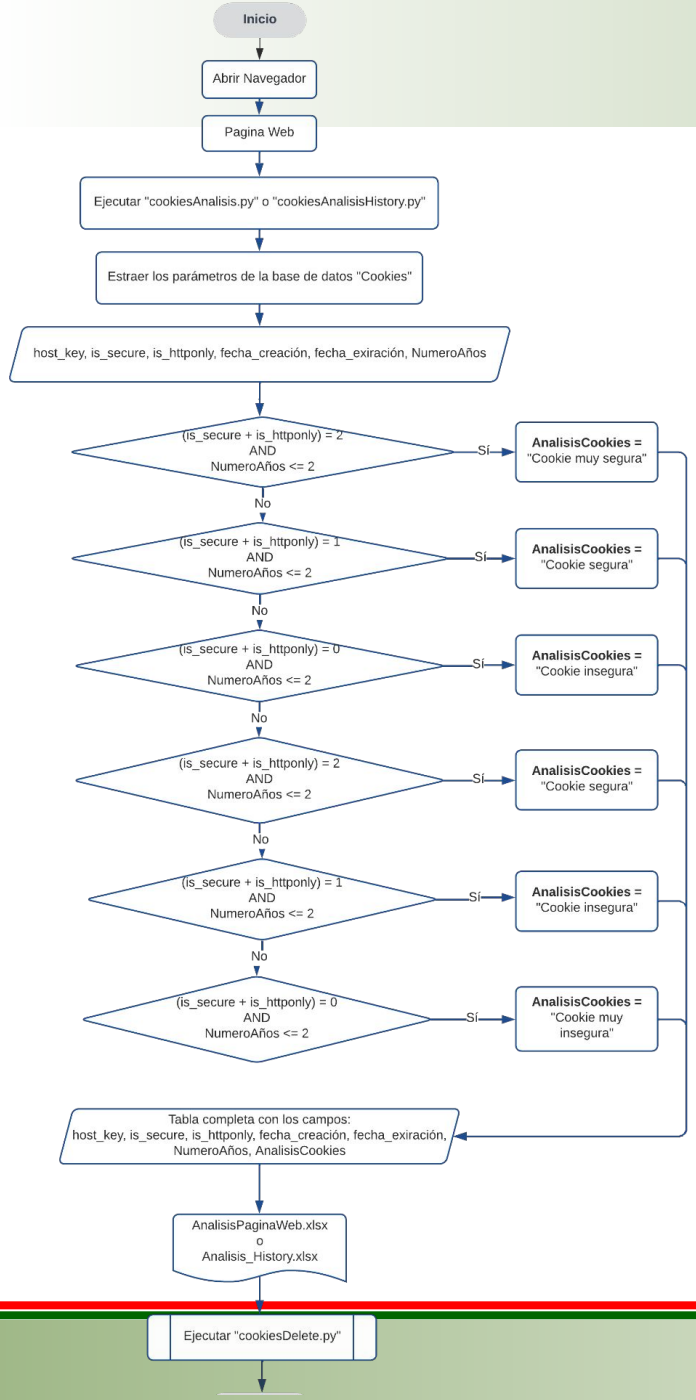


METODOLOGíA

Variables identificadas para el diseño del análisis del bot.

Campo	Detalle
host_key	Representa el dominio o página web de la cookie generada por la página web visitada por el usuario
creation_utc	Corresponde a la fecha y hora en qué se creó la cookie. Este dato es almacenado en formato Unix epoch, el cual es el número de segundos que han transcurrido desde el 1 de enero de 1601.
expires_utc	Corresponde a la fecha y hora de caducidad de la cookie. Al igual que creation_utc, este se encuentra en formato Unix epoch.
is_secure	El valor [0] en el atributo is_secure, representa que el navegador puede enviar la cookie a través de un canal o conexión HTTP. Lo cual permite que los atacantes accedan a las cookies.
	El valor [1] en el atributo is_secure, representa que la cookie se utiliza a través de HTTPS y se transmite de forma segura en texto cifrado, protegiendo la confidencialidad de la cookie.
is_httponly	El valor [0] en el atributo is_httponly, indica que la cookie no está protegida ante ataques XSS por lo que existe la posibilidad de acceder a sus datos.
	El valor [1] en un atributo is_httponly, prohíbe que se acceda a la cookie a través de scripts del lado del cliente, en específico con la propiedad *document.cookie*.

METODOLOGíA



Modelo en Python para el análisis del archivo “Cookies”

Análisis para determinar si una cookie es segura o insegura a ataques XSS

is_httponly	is_secure	num_years	Cookie muy segura	Cookie segura	Cookie insegura	Cookie muy insegura
1	1	<= 2	x			
1	0	<= 2		x		
0	1	<= 2		x		
0	0	<= 2			x	
1	1	> 2		x		
1	0	> 2			x	
0	1	> 2			x	
0	0	> 2				x

METODOLOGíA

cookiesAnalisis.py

```
#Consulta para determinar el analisis final----------------------------------------------------------------------------------------------------
print('\n================================')
print('  ANALISIS FINAL')
print('================================\n')
print('| DOMINIO | HTTPONLY | SECURE | FECHA CREACION | FECHA EXPIRACION | DURACION (ANIOS)| ANALISIS EN BASE DE LAS COOKIES |')

conv_fcrToYears = "strftime('%J', date(creation_utc/1000000 + (strftime('%s', '1601-01-01')), 'unixepoch', 'localtime'))"
conv_fexToYears = "strftime('%J', date(expires_utc/1000000 + (strftime('%s', '1601-01-01')), 'unixepoch', 'localtime'))"
num_years = "round((("+conv_fexToYears+" - "+conv_fcrToYears+")/365),4)"

sqlanalisis = """SELECT
                    host_key AS DominioCookie, is_secure AS Segura, is_httponly AS HttpOnly,
                    """+fecha_creacion+""" AS FechaCreacion,
                    """+fecha_expiracion+""" AS FechaExpiracion,
                    (CASE
                        WHEN """+num_years+""" >= 0
                            THEN """+num_years+"""
                        ELSE 'No Deff'
                    END) AS NumeroAños,
                    (CASE
                        WHEN (is_secure + is_httponly) = 2 and """+num_years+""" =< 2
                            THEN 'Cookie muy segura'
                        WHEN (is_secure + is_httponly) = 1 and """+num_years+""" =< 2
                            THEN 'Cookie segura'
                        WHEN (is_secure + is_httponly) = 0 and """+num_years+""" =< 2
                            THEN 'Cookie insegura'
                        WHEN (is_secure + is_httponly) = 2 and """+num_years+""" > 2
                            THEN 'Cookie segura'
                        WHEN (is_secure + is_httponly) = 1 and """+num_years+""" > 2
                            THEN 'Cookie insegura'
                        WHEN (is_secure + is_httponly) = 0 and """+num_years+""" > 2
                            THEN 'Cookie muy insegura'
                    END) AS AnalisisCookies
                FROM
                    cookies"""

cur.execute(sqlanalisis)
rows2 = cur.fetchall()

for row2 in rows2:
    print(row2)
```

Consulta SQLite con todos los parámetros que analizan las cookies en base a las condiciones.

cookiesDelete.py

analizador > cookiesDelete.py > ...

```
1  import sqlite3 as sql
2
3  con = sql.connect('./Cookies')
4  cur = con.cursor()
5
6  cur.execute('DELETE FROM cookies')
7  con.commit()
8
9  con.close()
```

```
97  #Enviar los resultados a un excel
98  analisis = pd.read_sql(sql = sqlanalisis , con = con)
99  nombre = dom.replace('://','')
100 analisis.to_excel("C:\\Users\\JORDY\\Desktop\\AnalisisXSS\\AnalsisPaginasWeb\\analisis_("+nombre+").xlsx", index=False)
101 #----------------------------------------------------------------------------------------------------
102
103 con.close()
```

METODOLOGíA

Desarrollo del bot en el software UI.Vision

Test_XSS Save Record Step Play Macro

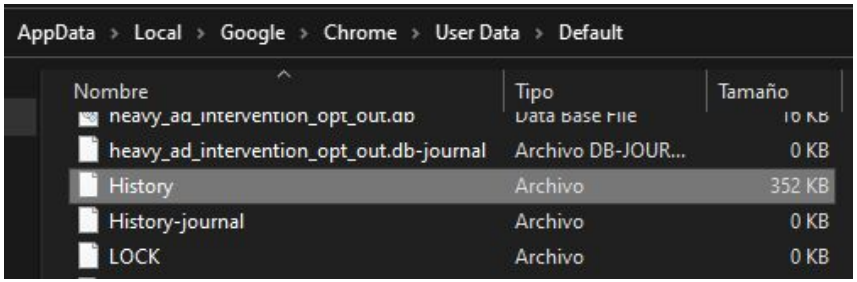
Table View | Source View (JSON) | Desktop mode

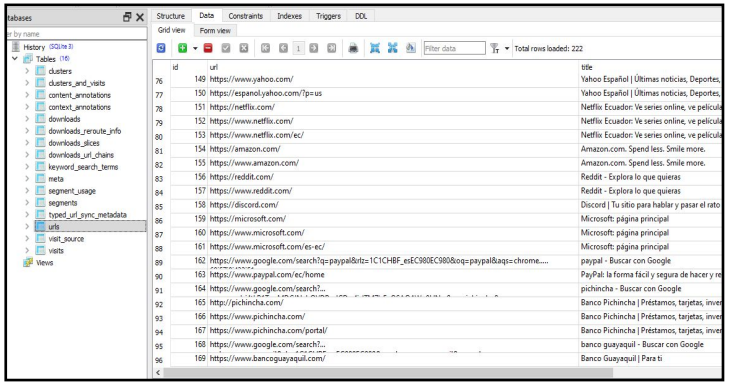
	Command	Target	Value	Ops
1	csvReadArray	SitiosWeb.csv	myCSV	
2	store	0	i	
3	while_v2	${i} <=1		
4	XClick	chrome_dpi_96.png	#doubleclick	
5	pause	4000		
6	XType	${myCSV[${i}]}		
7	XType	${KEY_ENTER}		
8	pause	8000		
9	XClick	Salir_dpi_96.png	#left	
10	XRun	C:\Users\JORDY\Desktop\abrir.vbs		
11	pause	2000		
12	XType	python cookiesListar.py		
13	XType	${KEY_ENTER}		
14	pause	2000		
15	XType	${myCSV[${i}]}		
16	XType	${KEY_ENTER}		
17	pause	4000		
18	XType	python cookiesDelete.py		
19	XType	${KEY_ENTER}		
20	pause	2000		
21	XType	${KEY_WIN+KEY_DOWN}		
22	executeScript_Sandbox	return Number (${i}) + 1;	i	
23	end			

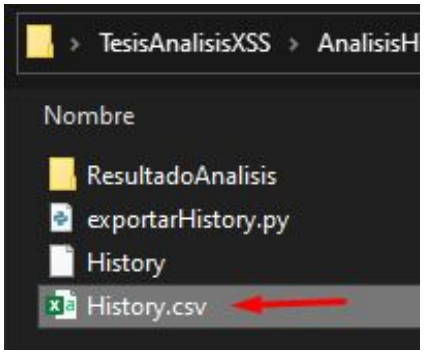
Una característica de UI.Vision es que el proceso de automatización ya sea web o de escritorio está compuesto por un conjunto de pasos que se ejecutan de manera secuencial.

METODOLOGíA

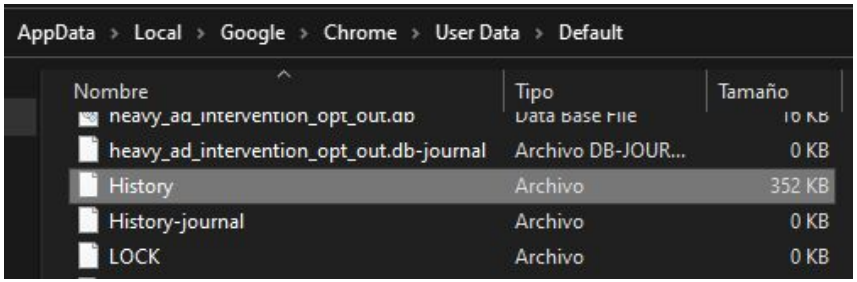
Análisis del historial de visitas de un usuario en Chrome

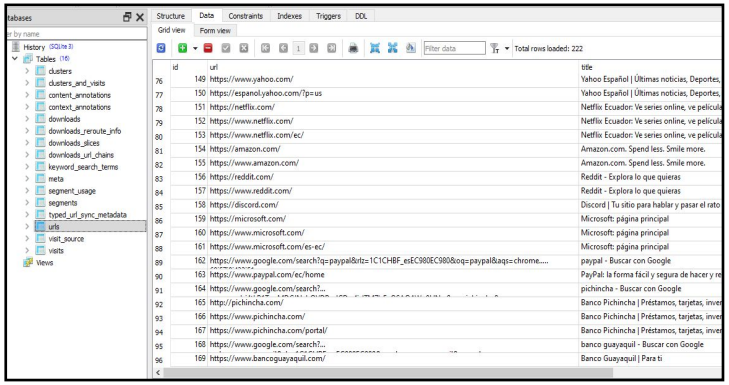


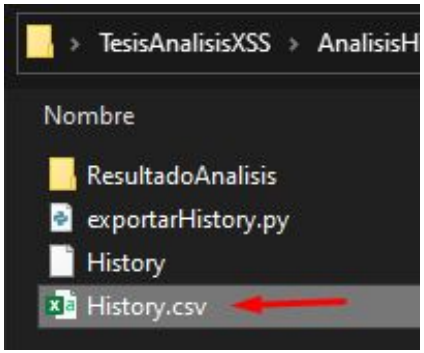




METODOLOGíA

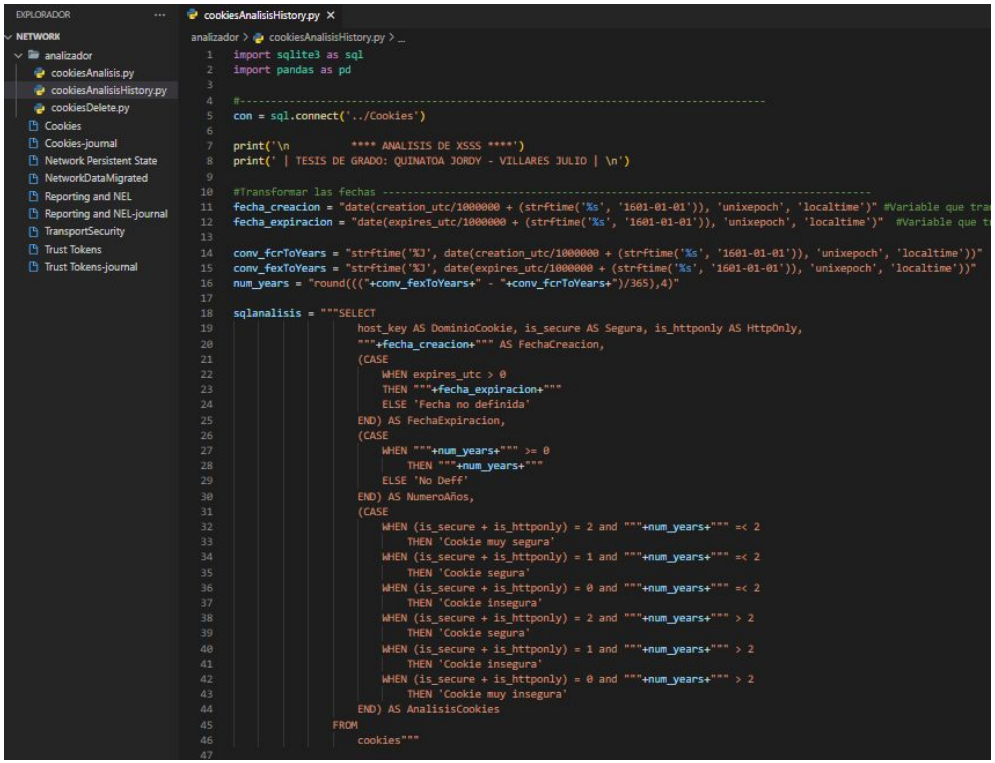






METODOLOGíA

Análisis de las cookies del historial de Chrome de un usuario




```python
#Exportar los resultados a un excel
analisis = pd.read_sql(sql = sqlanalisis , con = con)
analisis.to_excel("C:\\Users\\JORDY\\Desktop\\TesisAnalisisXSS\\AnalisisHistory\\ResultadoAnalisis\\Analisis_History.xlsx", index=False)
#-----------------------------------------------------------------------------------------------------------------------

con.close()

#Mostrar mensaje en cosola
print('||||| Resultados exportados correctamente |||||')
```

METODOLOGíA

historyURLS Save Record

Table View | Source View (JSON)

	Command	Target	Value
1	csvReadArray	History.csv	historyCSV
2	store	0	i
3	XClick	chrome_dpi_96.png	#doubleclick
4	pause	3000	
5	while_v2	${i} <=218	
6	XType	${KEY_CTRL+KEY_L}	
7	XType	${historyCSV[${i}]}	
8	XType	${KEY_ENTER}	
9	pause	6000	
10	executeScript_Sandbox	return Number (${i}) + 1;	i
11	end		
12	XClick	Salir_dpi_96.png	#left
13	pause	2000	
14	XRun	C:\Users\JORDY\Desktop\abrir.vbs	
15	pause	2000	
16	XType	python cookiesAnalisisHistory.py	
17	pause	2000	
18	XType	${KEY_ENTER}	

1

2

Implementación del bot con los procesos para realizar el análisis de las cookies de un usuario en base a su historial de navegación de Chrome

WHILE que contiene las instrucciones o comandos que cumplen la acción de abrir cada una de las páginas web o URLs del archivo CSV que se extrajeron desde la base de datos “History”.

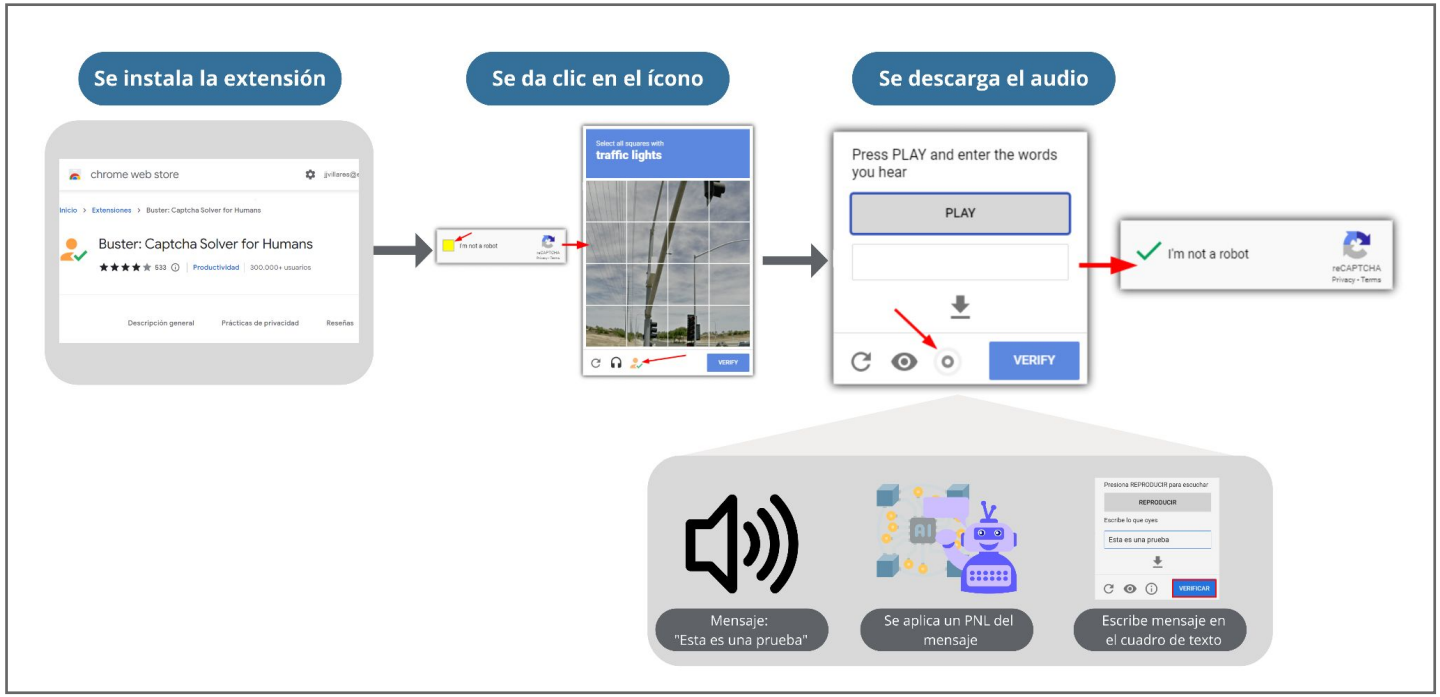
Instrucciones para ejecutar el script “cookiesAnalisisHistory”, para obtener el excel de resultados con el análisis de las cookies.

CAPÍTULO IV
RESULTADOS

RESULTADOS

Resultados de la solución implementada con alternativas existentes

Buster: CAPTCHA Solver for Humans



RESULTADOS

Anti-CAPTCHA Solving

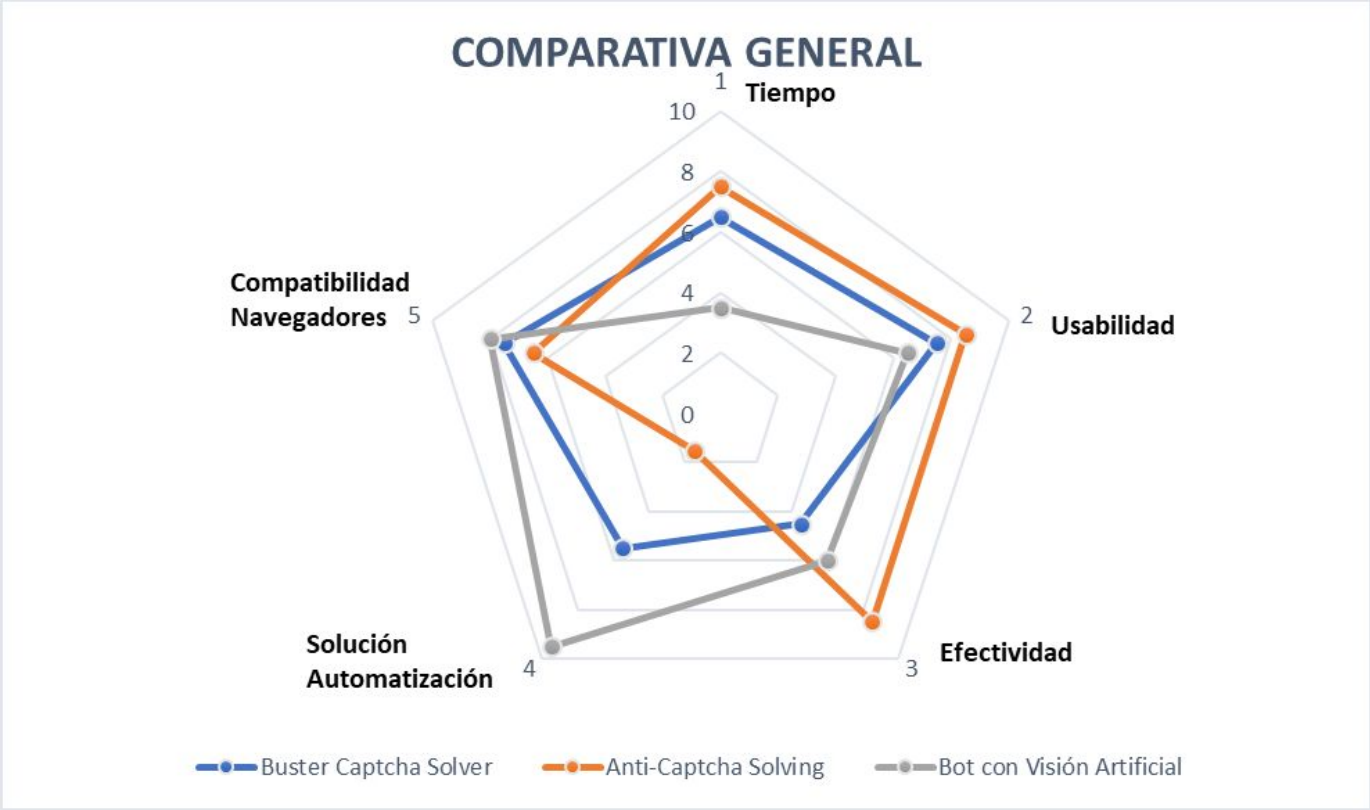
RESULTADOS

Análisis comparativo de las soluciones encontradas para resolver CAPTCHA

Parámetros evaluados	Buster CAPTCHA Solver				Anti-CAPTCHA Solving				Bot con Visión Artificial			
	U1	U2	Prom	Cali. A	U1	U2	Prom	Cali. B	U1	U2	Prom	Cali. C
Tiempo de ejecución	6	7	6,5	MEDIO	8	7	7,5	ALTO	3	4	3,5	BAJO
Usabilidad	8	7	7,5	ALTO	9	8	8,5	ALTO	7	6	6,5	MEDIO
Efectividad	5	4	4,5	MEDIO	8	9	8,5	ALTO	7	5	6	MEDIO
Solución para automatización	6	5	5,5	MEDIO	1	2	1,5	BAJO	9	10	9,5	ALTO
Compatibilidad con navegadores	8	7	7,5	ALTO	6	7	6,5	MEDIO	7	9	8	ALTO
		TOTAL	6,3			TOTAL	6,5			TOTAL	6,7	

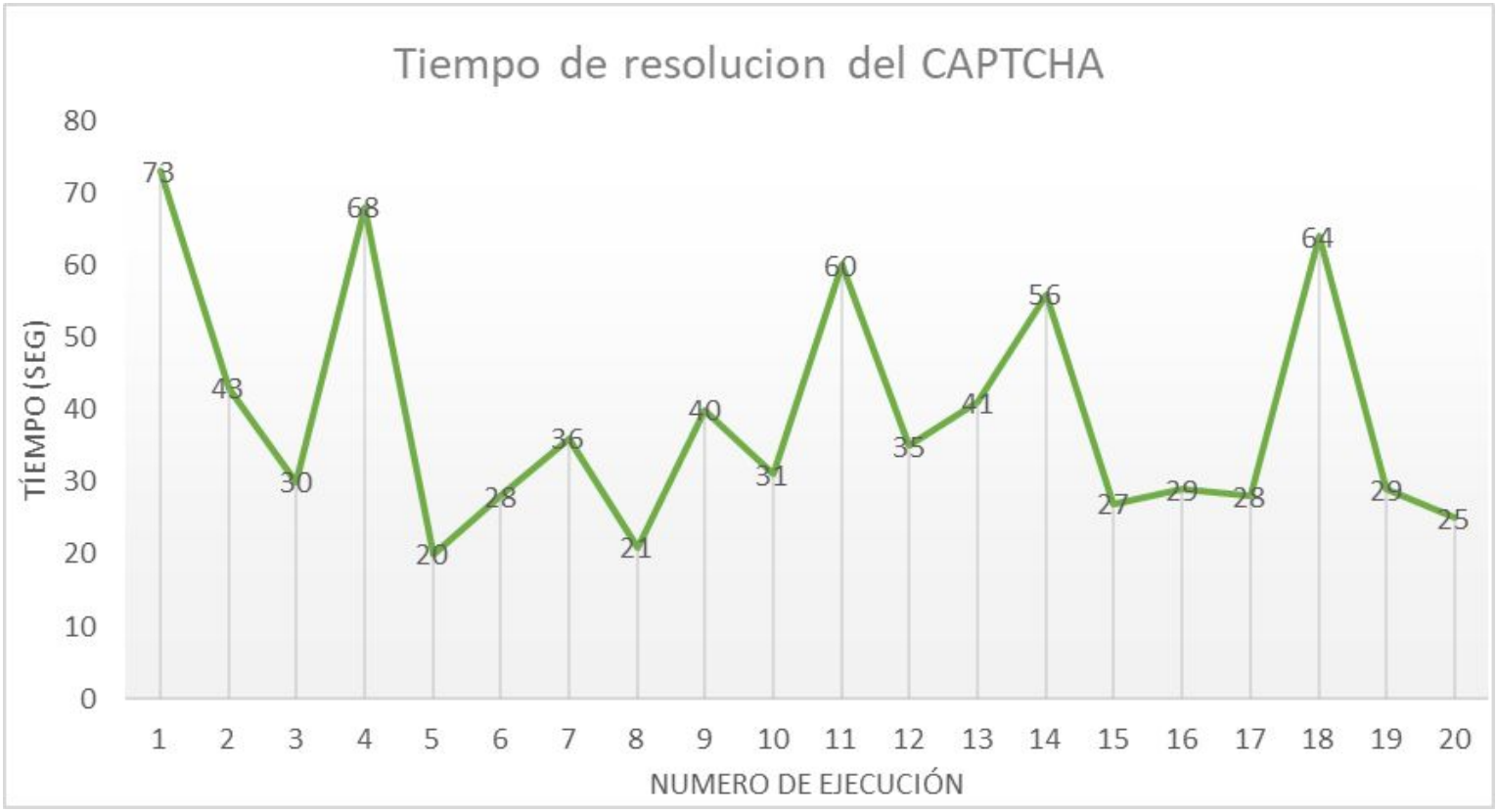
RESULTADOS

Análisis general de las herramientas para resolver los sistemas CAPTCHA



RESULTADOS

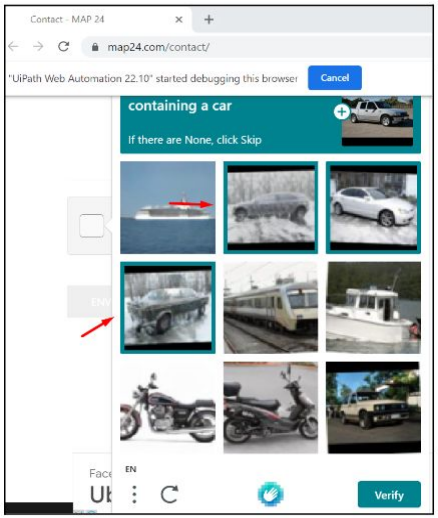
Pruebas de rendimiento y ejecución del Bot

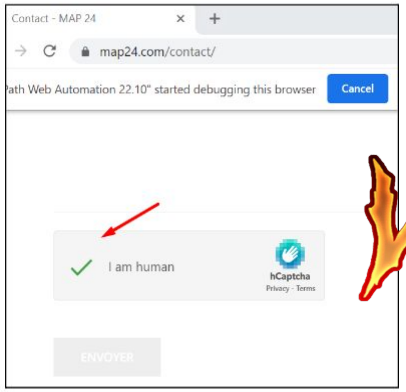


RESULTADOS

Pruebas de rendimiento y ejecución del Bot

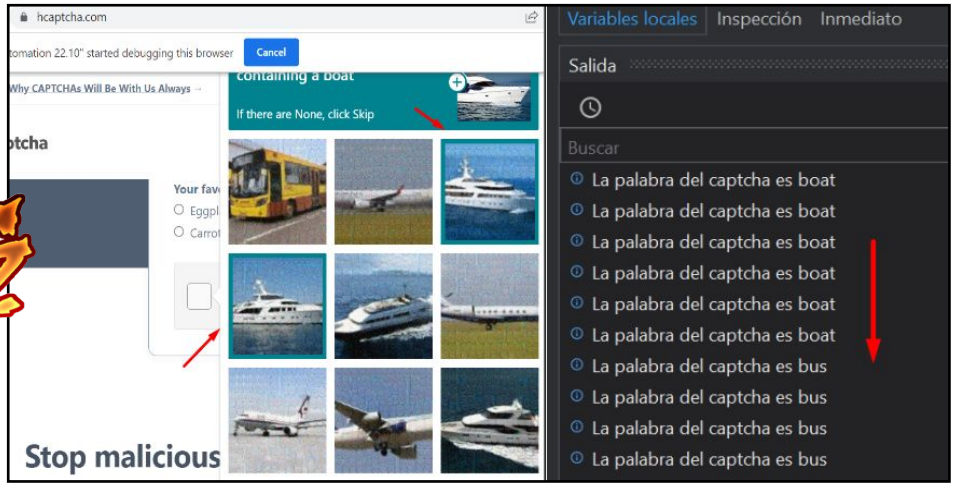
https://www.map24.com/contact/





VS

https://www.hCAPTCHA.com



RESULTADOS

Resultados del bot de análisis de ataques XSS a páginas web

Archivos excel

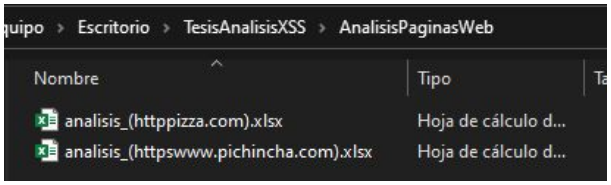
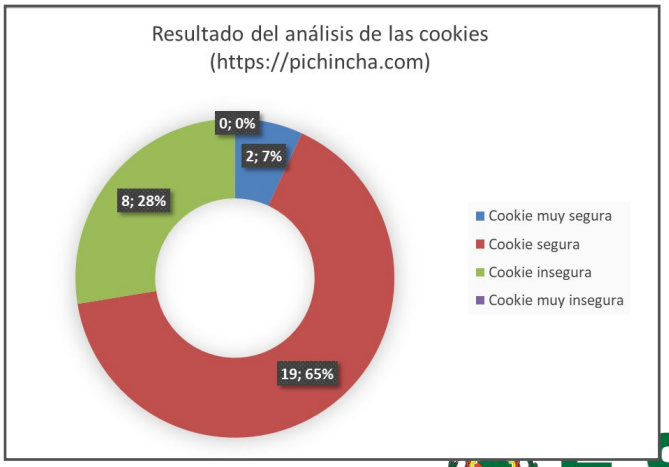


Tabla de resultados

	A	B	C	D	E	F	G
1	DominioCookie	Segura	HttpOnly	FechaCreacion	FechaExpiracion	NumeroAños	AnalisisCookies
2	.doubleclick.net	1	1	2022-07-30	2022-07-30	0	Cookie muy segura
3	.doubleclick.net	1	1	2022-07-30	2024-07-29	2	Cookie muy segura
4	.pizza.com	0	0	2022-07-30	2023-08-24	1.0685	Cookie insegura
5	.pizza.com	0	0	2022-07-30	2023-08-24	1.0685	Cookie insegura
6	.pizza.com	0	0	2022-07-30	2024-07-29	2	Cookie insegura
7	.pizza.com	0	0	2022-07-30	2022-07-30	0	Cookie insegura
8	.pizza.com	0	0	2022-07-30	2022-07-30	0	Cookie insegura
9	.pizza.com	0	0	2022-07-30	2023-01-28	0.4986	Cookie insegura
10	.pizza.com	0	0	2022-07-30	2024-07-29	2	Cookie insegura

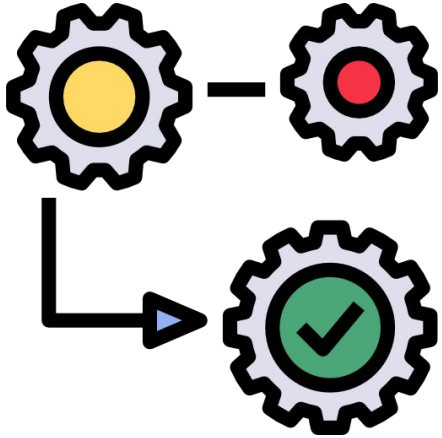
Análisis de las cookies



RESULTADOS

Tiempo de ejecución del bot en base a los dominios analizado

Número de páginas web	Tiempo aproximado
1 página web	38 seg.
2 páginas web	76 seg. (1.26 min.)
10 páginas web	380 seg. (6.33 min)
100 páginas web	3800 seg. (1 hr.)



RESULTADOS

Resultados de la solución para el análisis de las cookies del historial de navegación

Archivo CSV del historial

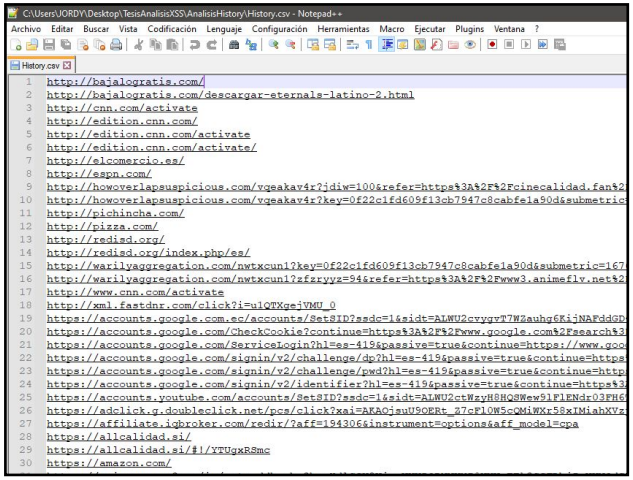
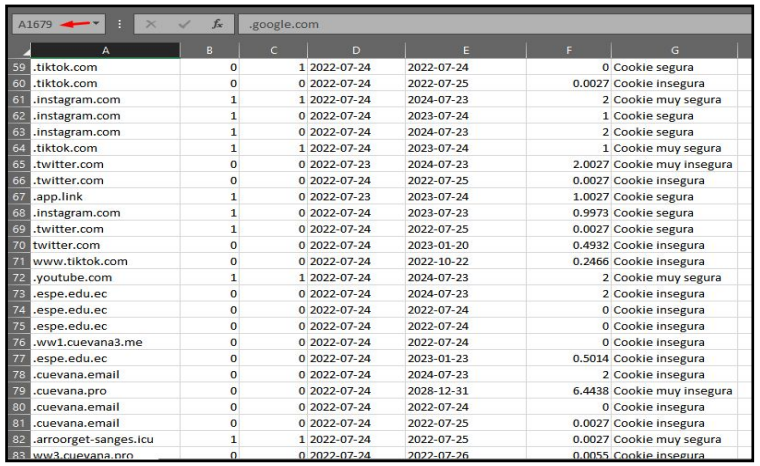
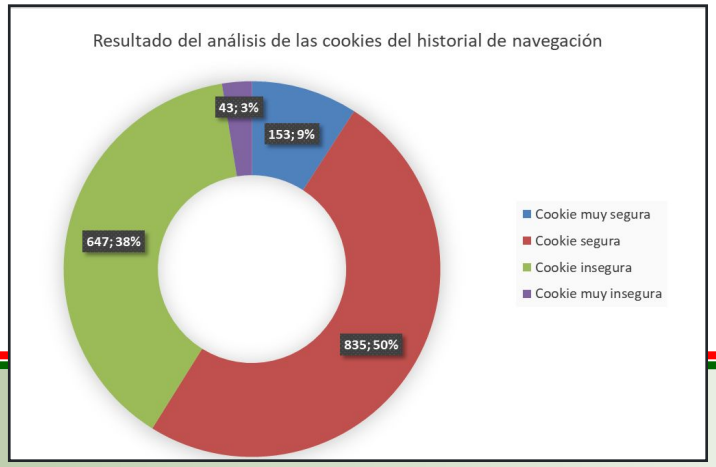


Tabla de resultados



	A	B	C	D	E	F	G
59	.tiktok.com	0	1	2022-07-24	2022-07-24	0	Cookie segura
60	.tiktok.com	0	0	2022-07-24	2022-07-25	0.0027	Cookie insegura
61	.instagram.com	1	1	2022-07-24	2024-07-23	2	Cookie muy segura
62	.instagram.com	1	0	2022-07-24	2023-07-24	1	Cookie segura
63	.instagram.com	1	0	2022-07-24	2024-07-23	2	Cookie segura
64	.tiktok.com	1	1	2022-07-24	2023-07-24	1	Cookie muy segura
65	.twitter.com	0	0	2022-07-23	2024-07-23	2.0027	Cookie muy insegura
66	.twitter.com	0	0	2022-07-24	2022-07-25	0.0027	Cookie insegura
67	.app.link	1	0	2022-07-23	2023-07-24	1.0027	Cookie segura
68	.instagram.com	1	0	2022-07-24	2023-07-23	0.9973	Cookie segura
69	.twitter.com	1	0	2022-07-24	2022-07-25	0.0027	Cookie segura
70	twitter.com	0	0	2022-07-24	2023-01-20	0.4932	Cookie insegura
71	www.tiktok.com	0	0	2022-07-24	2022-10-22	0.2466	Cookie insegura
72	.youtube.com	1	1	2022-07-24	2024-07-23	2	Cookie muy segura
73	.espe.edu.ec	0	0	2022-07-24	2024-07-23	2	Cookie insegura
74	.espe.edu.ec	0	0	2022-07-24	2022-07-24	0	Cookie insegura
75	.espe.edu.ec	0	0	2022-07-24	2022-07-24	0	Cookie insegura
76	.ww1.cuevana3.me	0	0	2022-07-24	2022-07-24	0	Cookie insegura
77	.espe.edu.ec	0	0	2022-07-24	2023-01-23	0.5014	Cookie insegura
78	.cuevana.email	0	0	2022-07-24	2024-07-23	2	Cookie insegura
79	.cuevana.pro	0	0	2022-07-24	2028-12-31	6.4438	Cookie muy insegura
80	.cuevana.email	0	0	2022-07-24	2022-07-24	0	Cookie insegura
81	.cuevana.email	0	0	2022-07-24	2022-07-25	0.0027	Cookie insegura
82	.arroorget-sanges.icu	1	1	2022-07-24	2022-07-25	0.0027	Cookie muy segura
83	ww3.cuevana.pro	0	0	2022-07-24	2022-07-26	0.0055	Cookie insegura

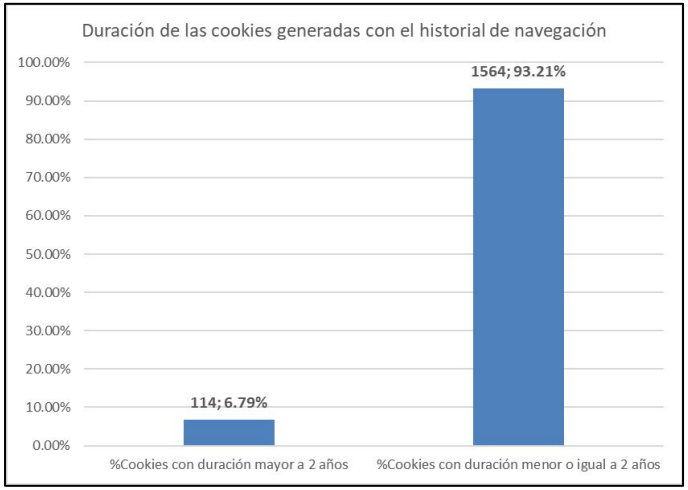
Análisis de las cookies

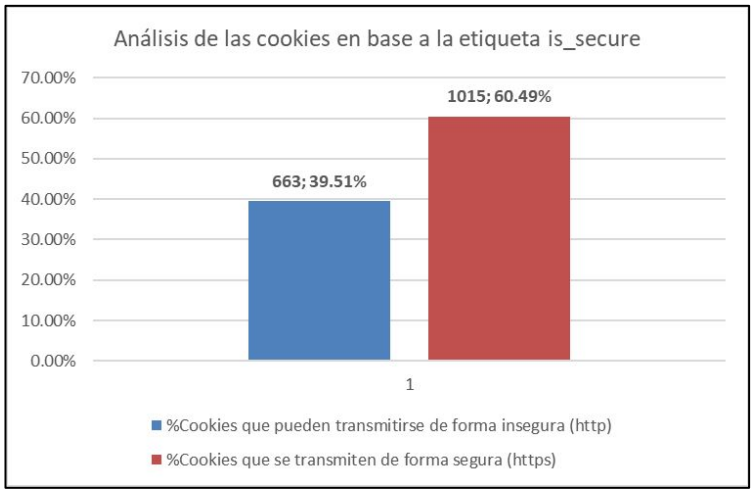


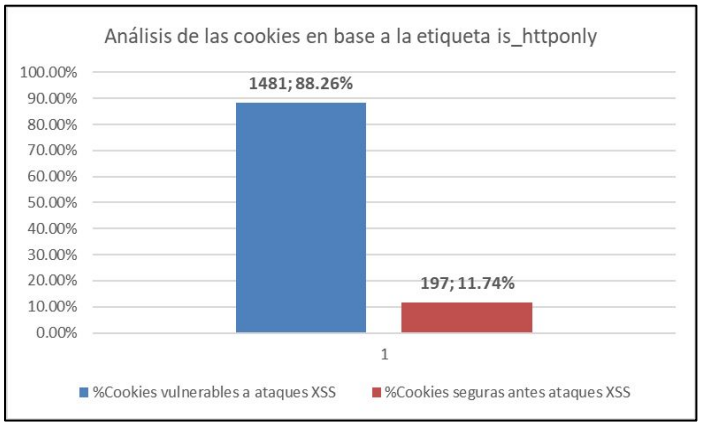
ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

VERSIÓN: 1.0

RESULTADOS





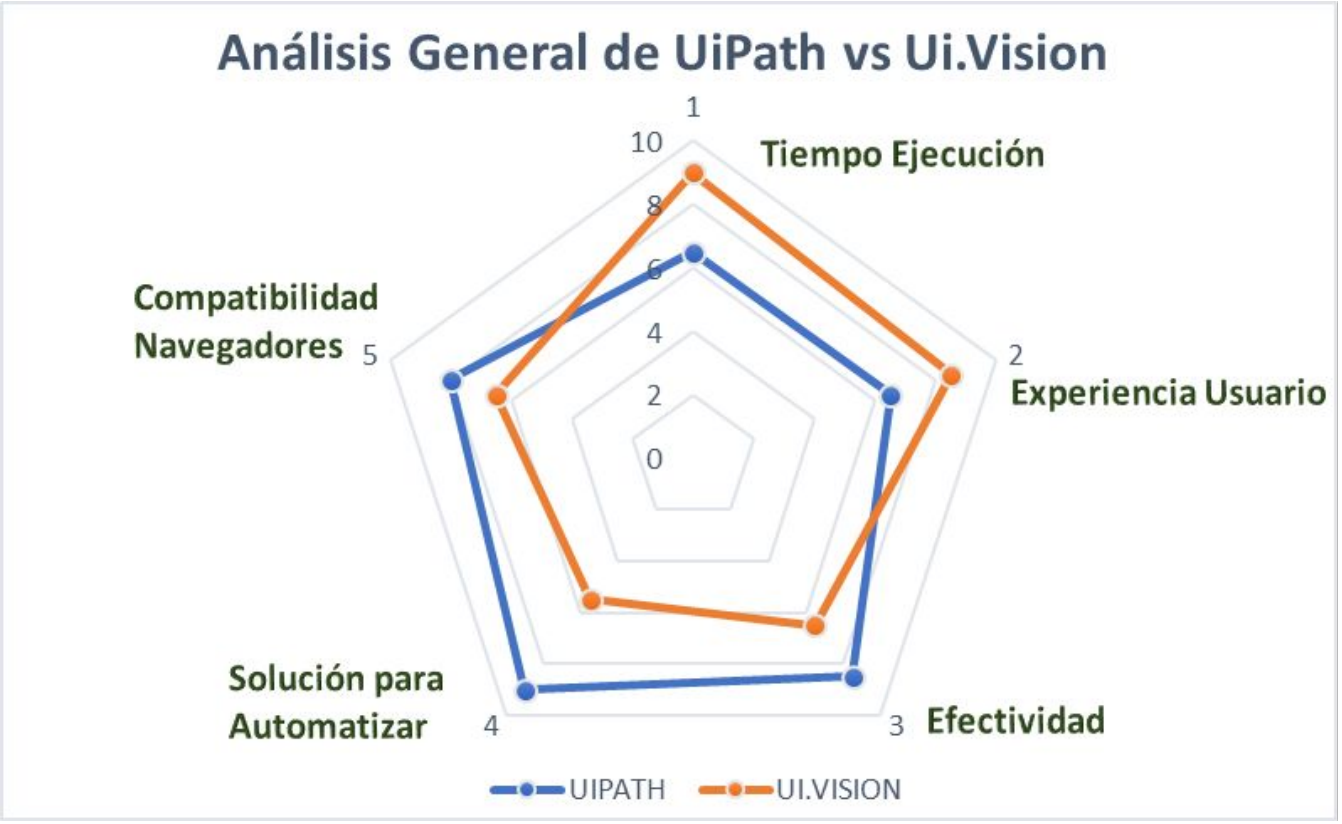


Tiempo de ejecución del bot para análisis del historial

Número de páginas web	Tiempo aproximado
Historial con 1 URL	27 seg.
Historial con 10 URLs	108 seg. (1.8 min)
Historial con 100 URLs	918 seg. (15.3 min)
Historial con 1000 URLs	9018 seg. (150.3 min)

RESULTADOS

Comparativa de las herramientas RPA UiPath y UI.Vision



CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Se concluye que las técnicas más utilizadas para vulnerar este tipo de medidas de seguridad son los sistemas de reconocimiento de caracteres (OCR), los sistemas de reconocimiento de habla y los analizadores sintácticos, cuya función es intentar resolver los desafíos y simular la interacción de un humano.
- En base al análisis comparativo de las herramientas RPA planteado en este estudio, se determinó que UiPath es una de las soluciones más completas y accesibles del mercado. Su versión Community Edition, a pesar de ser limitada, ofrece las opciones suficientes para implementar una automatización funcional.
- Se cumplió con el objetivo de desarrollar un bot para vulnerar los desafíos de los sistemas CAPTCHA, se utilizó como referencia el mostrado en la página web www.map24.com (revista online francesa) que corresponde a un tipo hCapcha, se ha integrado con la API de Google Cloud Vision, proporcionando las técnicas de visión artificial para el análisis de imágenes.
- Cross-site scripting (XSS) se considera como una vulnerabilidad que aprovecha las fallas de seguridad para que los atacantes puedan inyectar scripts ejecutables y maliciosos en aplicaciones o sitios web. Existen tres tipos de ataques, almacenado, reflejado y basado en DOM. Se concluye entonces que el ataque más peligroso es el almacenado porque el atacante inyecta el código malicioso en el servidor web una sola vez y luego afecta a todos los usuarios web que lo visiten.
- Empleando la herramienta UI.Vision se implementó un bot que analiza vulnerabilidades de tipo XSS de una manera rápida y sencilla. A través del estudio de los atributos (nombre, is_secure, is_httponly, fecha de creación y fecha de expiración) de las cookies generadas, se observó que los sitios web que utilizan el protocolo HTTP tienden a generar cookies inseguras a comparación de las que usan HTTPs.

RECOMENDACIONES

- La constante actualización del sistema de visión artificial con las nuevas imágenes que se proporcionan en los desafíos, ya que como se demostró en el presente proyecto, a un bot se le hace más difícil vulnerar imágenes con texturas o patrones incrustados.
- El bot puede llegar a fallar en ciertas ocasiones debido a que las imágenes del CAPTCHA pueden presentar texturas, patrones y/o distorsiones que dificultan el análisis. Para que la detección logre un 100% de efectividad, es recomendable la incorporación de nuevas imágenes con estas características en un dataset propio y actualizado.
- La capacidad de ejecución del bot de análisis de vulnerabilidades XSS realizado en UI.Vision es un punto que se puede mejorar, ya que presenta limitantes relacionadas con el uso de comandos (XClick, XType y XMove) en una misma ejecución. La solución más viable a esta limitante es con la obtención de una licencia para la Edición Personal de UI.Visión RPA XModules.
- Para mejorar la protección de los sistemas de información dentro de la web, se recomienda complementar nuestra propuesta con un verificador de cookies de terceros, ya que estas podrían obtener información confidencial del usuario que visita dicha página.
- Si se va a automatizar un proyecto complejo se recomienda la implementación del software UiPath. Por otro lado, si se desea implementar una automatización sencilla y repetitiva, la utilización de UI.Vision es una muy buena opción, no solo por su facilidad para automatizar, sino también por su portabilidad.

MUCHAS GRACIAS