1ª edición

¿Cómo hacer un informe de seguridad sin ser auditor/a? La guía definitiva.

eBooks
Paradigma

Autores/as: Alberto Perrote y Alba García

Ilustraciones: Marta Ruiz

Maquetación: Candela Nieto

Paradigma

Contenido:

Introducción:

¿Necesitas crear un **informe de seguridad básico** para evaluar las vulnerabilidades de un proyecto y no sabes ni por dónde empezar?

En este ebook evaluaremos algunas de las **herramientas de análisis de seguridad más comunes** a la hora de realizar un análisis básico de un proyecto y ofreceremos una **estructura básica para realizar un informe completo**. Aquí encontrarás información sobre cómo organizar el informe, qué herramientas puedes empezar a usar y cómo interpretar sus resultados.

Este tipo de informe aporta valor a un proyecto, tanto a nivel de seguridad del propio proyecto como a nivel del servicio que ofrecemos al propietario del mismo. ¡Vamos allá!

—01

Qué tener en cuenta antes de empezar.

¿Quién puede realizar un informe de este tipo?

Aún sabiendo que el informe que planteamos no es ni de lejos un informe que presentaría un auditor, sí que **es necesario cierto nivel técnico** para configurar, ejecutar e interpretar los datos que generan las herramientas que proponemos en este ebook.

No obstante, ¡no te asustes! Estas herramientas son de manejo sencillo y, lo más importante, son gratuitas. Debajo de la descripción de cada una de ellas te damos ciertas nociones básicas para ejecutarlas e interpretar los resultados de los informes que genera cada herramienta.

¿Cuál es el mejor momento para elaborar este tipo de informes?

Esta pregunta es importante para plantearnos si es el momento de realizar este tipo de informe, o si nos interesa esperar un poco. Desde nuestro punto de vista, recomendamos realizar el informe sobre un proyecto cuyo montaje o despliegue haya finalizado (aunque no esté en producción). Si es un proyecto que está por desplegar, obviamente vamos a tener información incompleta y tendremos que presentar varias versiones del informe.

El momento más interesante para realizar este tipo de informes es justo al terminar el despliegue del proyecto y antes de ponerlo en producción. Con esto conseguimos haber solventado las vulnerabilidades más críticas de nuestro proyecto antes de exponerlo a los usuarios finales.

¿Qué fases tiene el proceso?

Podríamos decir que la evaluación de seguridad debería tener estas fases:

1. **Disponer de un inventario de los componentes del proyecto a analizar**. Para ello necesitarás, al menos, la siguiente información y accesos:

    - Acceso a los endpoints a evaluar
    - Acceso a los componentes de infraestructura a evaluar
    - Servicios que conforman la plataforma a evaluar

2. **Ejecución de las herramientas de análisis de vulnerabilidades**
3. **Revisión de los informes generados por las herramientas de análisis**
4. **Redacción del informe de seguridad con las conclusiones obtenidas**

El momento más interesante para realizar este tipo de informes es justo al terminar el despliegue del proyecto y antes de ponerlo en producción.

La fase de revisión y análisis de los informes generados por las herramientas es una de las más importantes en este proceso, ya que las herramientas de análisis generan muchísima información y es necesario un análisis y estudio posterior de esta información para filtrar, estudiar y revisar las vulnerabilidades que dichas herramientas reportan. Por este motivo, y como adelantamos al principio, es recomendable que un perfil técnico realice esta labor de análisis y revisión de las vulnerabilidades que se detecten.

A continuación, pasamos a presentarte las herramientas que te proponemos para realizar este análisis.

—02

Herramientas de análisis de vulnerabilidades.

En este apartado vamos a hablar de algunas herramientas gratuitas que puedes usar para elaborar tu propio informe. Vamos a desglosar las herramientas por categorías, según lo que se vaya a evaluar. Hablaremos de herramientas para evaluar vulnerabilidades en los siguientes ámbitos:

- Código de la infraestructura
- Código de la aplicación
- Infraestructura
- Web de la aplicación

Hay que tener en cuenta que estas herramientas reportarán vulnerabilidades cuya corrección no aplique en situaciones concretas. Este tipo de decisiones o matices sobre la corrección o no de las vulnerabilidades detectadas en los informes habrá que comentarlo en la interpretación de los resultados del informe final, con la explicación de por qué se desestima la corrección del hallazgo.

Sin más dilación, pasamos a recomendarte algunas herramientas para realizar tu primer informe de seguridad, con pasos básicos para su ejecución e interpretación de resultados.

Este tipo de decisiones o matices sobre la corrección o no de las vulnerabilidades detectadas en los informes habrá que comentarlo en la interpretación de los resultados del informe final.

—02.01

Herramientas para revisar el código de la infraestructura

En esta entrada hemos seleccionado la herramienta Kics que permite el análisis de código de la infraestructura desplegada usando: Terraform, ficheros de configuración de K8s, Dockerfiles o Docker-compose, ficheros de CloudFormation (AWS) y playbooks o roles de Ansible.

Kics se centra en la búsqueda de vulnerabilidades de seguridad, problemas de conformidad y errores de configuración de la infraestructura del código. Se puede ejecutar de distintas maneras y adaptado a la revisión de distintos parámetros. Aquí te dejamos algunos ejemplos de ejecución:

- Ejecución de Kics mediante una imagen Docker
- Ejecución de Kics a través del código fuente
- Ejecución de Kics usando queries específicas
- Ejecución de Kics para revisión de Passwords y secretos

Ejecución

En este punto vamos a centrarnos en la descripción de la ejecución con Docker.

Kics está disponible como imagen Docker y se puede usar para escanear un fichero o un directorio. Para ello, únicamente hay que montar el fichero o directorio a escanear en el contenedor de Kicks y especificar la ruta en el filesystem del contenedor con el parámetro -p KICS.

Para ello, primero descargamos la imagen docker de kics:

```
docker pull checkmarx/kics:latest
```

Y después ejecutamos el escaneo:

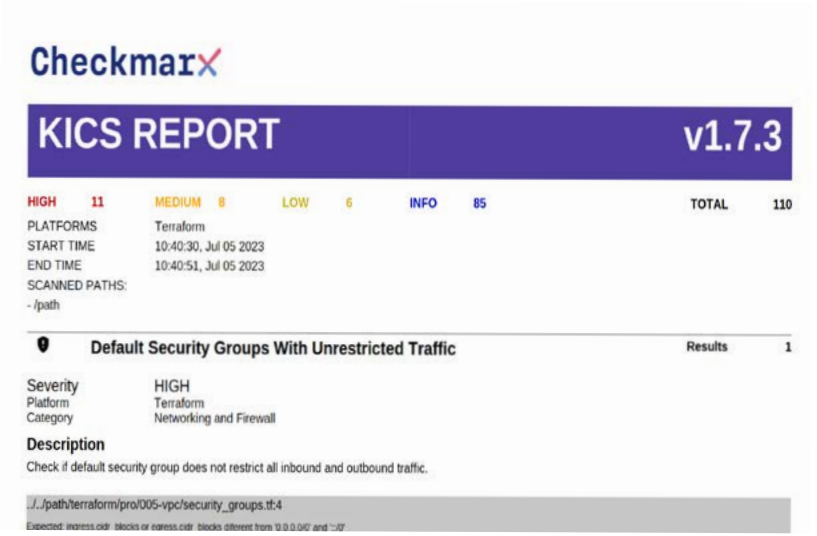
```
sudo docker run -t -v {path_del_repositorio_de_código}:/path
checkmarx/kics:latest scan -p /path -o “/path/” --report-for-
mats “{formato_del_reporte}”
```

Una vez lanzada la ejecución empezará el escaneo y, en pocos minutos, se habrá generado el fichero con las conclusiones del mismo.

Para esta guía se ha elegido el formato PDF para el reporte, ya que es un formato más legible y fácil de analizar.

Interpretación de los resultados

En el fichero de los resultados del escaneo aparecerá, al inicio, un pequeño resumen con el total de las vulnerabilidades detectadas según su severidad.



La severidad de las vulnerabilidades se categoriza de la siguiente manera:

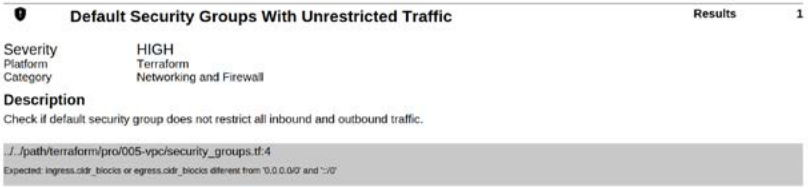
Alta: Para vulnerabilidades que es primordial corregir cuanto antes.

Media: Para vulnerabilidades que se pueden corregir en el corto plazo.

Baja: Para vulnerabilidades que se pueden corregir en el medio plazo.

Info: Vulnerabilidades de tipo informativo que convendría corregir o, al menos, estudiar si es viable su corrección.

A continuación, el resto del informe presentará, una por una, cada una de las vulnerabilidades detectadas en la ejecución de la herramienta. Veamos un ejemplo:



Como se puede observar, en cada vulnerabilidad aparecen los siguientes campos:

Severidad:
Indica la severidad del hallazgo. Como hemos comentado en párrafos anteriores, puede ser de tipo:

Plataforma:
Está referido a la tecnología en la que se encuentra desplegado el código. En el caso del ejemplo, la vulnerabilidad se ha detectado en el código desplegado con Terraform.

Categoría:
Está referido a la categoría en la que se engloba el hallazgo. En el ejemplo, el hallazgo se refiere a una vulnerabilidad que afecta a la red, por lo que lo engloba en la categoría "Networking and Firewall". Aquí puedes consultar todas las categorías.

Descripción:
Este campo incluye una breve descripción de la vulnerabilidad encontrada. En el ejemplo, se hace referencia a que el fichero donde se definen los grupos de seguridad de la red (Security Groups), que son los que actúan permitiendo el acceso de ciertos rangos de IPs a los recursos, no restringen todo el tráfico entrante y/o saliente. Esto es debido a que el fichero tiene reglas donde se permiten las conexiones entrantes y/o salientes a los rangos 0.0.0.0/0 y ::/0.

—02.02

Herramientas para revisar el código de la aplicación

Para la evaluación del código de la aplicación, vamos a mostrar una herramienta de análisis de código de aplicaciones móviles. La herramienta que hemos elegido es MobSF. Esta herramienta permite la evaluación de seguridad, análisis de malware y pruebas de pen-testing de aplicaciones móviles (Android/iOS/Windows) de manera automatizada.

MobSF admite binarios de aplicaciones móviles (APK, XAPK, IPA y APPX) junto con código fuente comprimido y proporciona un API REST para una integración perfecta con los canales que tengas habilitados para CI/CD o DevSecOps.

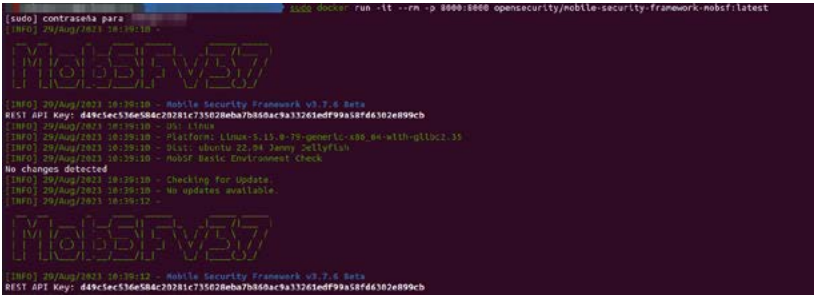
Ejecución

La manera más sencilla de ejecutar la herramienta es mediante el uso de la imagen Docker correspondiente. Para ello, siguiendo la documentación de la herramienta ejecutamos la descarga de la imagen:

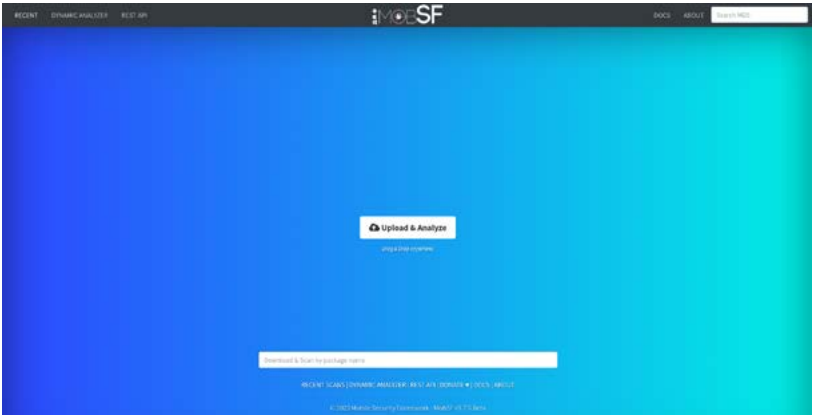
```
docker pull opensecurity/mobile-security-framework-mobsf:latest
```

Y posteriormente la ejecutamos:

```
docker run -it --rm -p 8000:8000 opensecurity/mobile-securi-
ty-framework-mobsf:latest
```

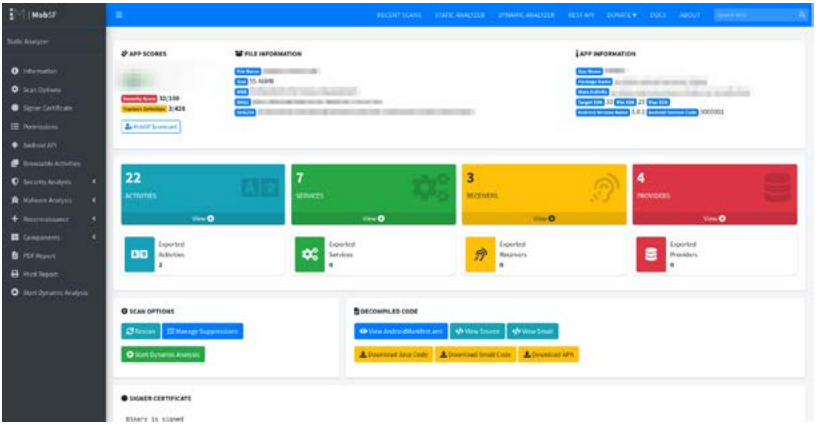


Una vez tengamos ejecutado el contenedor, abriremos en un navegador la URL: 127.0.0.1:8000



En este punto se presentará la interfaz para subir el fichero correspondiente del código de la aplicación que queremos analizar.

Una vez termine la ejecución, se presentará el siguiente informe de resultados:



El informe se puede presentar en PDF para adjuntarlo posteriormente a los ficheros de evidencias del análisis de seguridad.

Interpretación de los resultados

Para facilitar el análisis de los resultados, es recomendable descargar el fichero del informe en PDF. En dicho informe, aparece al principio un resumen de los hallazgos de seguridad encontrados:

Donde la criticidad de las vulnerabilidades se presenta de la siguiente forma:

High: Para vulnerabilidades que es primordial corregir cuanto antes.

Medium: Para vulnerabilidades que se pueden corregir en el corto plazo.

Low: Para vulnerabilidades que se pueden corregir en el medio plazo.

Secure: Configuraciones de la aplicación que se consideran seguras.

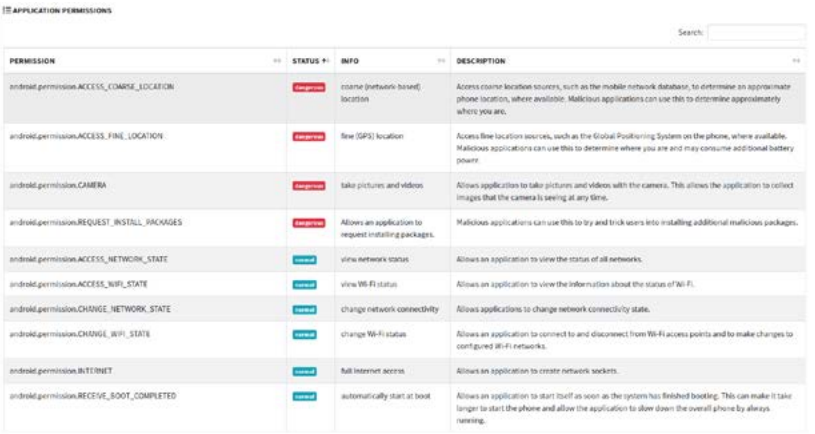
Info: Vulnerabilidades de tipo informativo que convendría corregir o, al menos, estudiar si es viable su corrección.

En el informe aparecen varios apartados referidos a la información de la aplicación y el código desplegado por la misma. A continuación, se citan algunos de los apartados con ejemplos:

Certificate information:
Información sobre el certificado usado por la aplicación.

Application Permissions:

Contiene información sobre los permisos de la aplicación sobre distintos servicios del terminal donde se ejecuta. En el ejemplo aparecen algunas vulnerabilidades detectadas, entre las que destaca una vulnerabilidad que se considera “Peligrosa” debido a que la aplicación requiere el uso de la cámara, entre otros.



APPLICATION PERMISSIONS

Search:

PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.ACCESS_COARSE_LOCATION	dangerous	coarse (network-based) location	Access coarse location sources, such as the mobile network database, to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are.
android.permission.ACCESS_FINE_LOCATION	dangerous	fine (GPS) location	Access fine location sources, such as the Global Positioning System on the phone, where available. Malicious applications can use this to determine where you are and may consume additional battery power.
android.permission.CAMERA	dangerous	take pictures and videos	Allows application to take pictures and videos with the camera. This allows the application to collect images that the camera is seeing at any time.
android.permission.REQUEST_INSTALL_PACKAGES	dangerous	Allows an application to request installing packages.	Malicious applications can use this to try and trick users into installing additional malicious packages.
android.permission.ACCESS_NETWORK_STATE	normal	view network status	Allows an application to view the status of all networks.
android.permission.ACCESS_WIFI_STATE	normal	view Wi-Fi status	Allows an application to view the information about the status of Wi-Fi.
android.permission.CHANGE_NETWORK_STATE	normal	change network connectivity	Allows applications to change network connectivity state.
android.permission.CHANGE_WIFI_STATE	normal	change Wi-Fi status	Allows an application to connect to and disconnect from Wi-Fi access points and to make changes to configured Wi-Fi networks.
android.permission.INTERNET	normal	full Internet access	Allows an application to create network sockets.
android.permission.RECEIVE_BOOT_COMPLETED	normal	automatically start at boot	Allows an application to start itself as soon as the system has finished booting. This can make it take longer to start the phone and allow the application to slow down the overall phone by always running.

APKID Analysis:

Presenta información sobre cómo se ha generado la aplicación (APK en este ejemplo).

ANDROID API

Search:

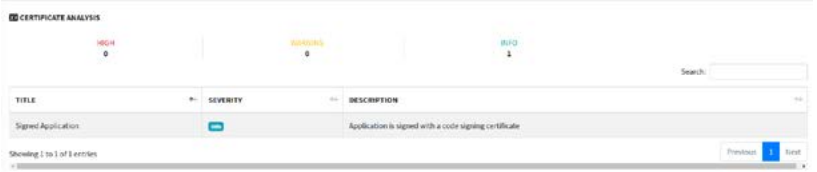
API	FILES
Android Notifications	
Base64 Decode	
Base64 Encode	

Network Security:

Presenta información sobre el dominio de la aplicación.

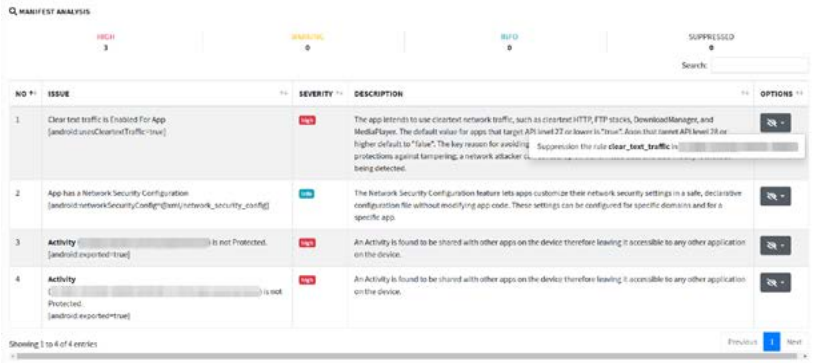
Certificate Analysis:

Presenta información sobre el certificado que usa la aplicación. En el ejemplo, se presenta un hallazgo de tipo informativo indicando que la aplicación está firmada con un certificado de firma de código.



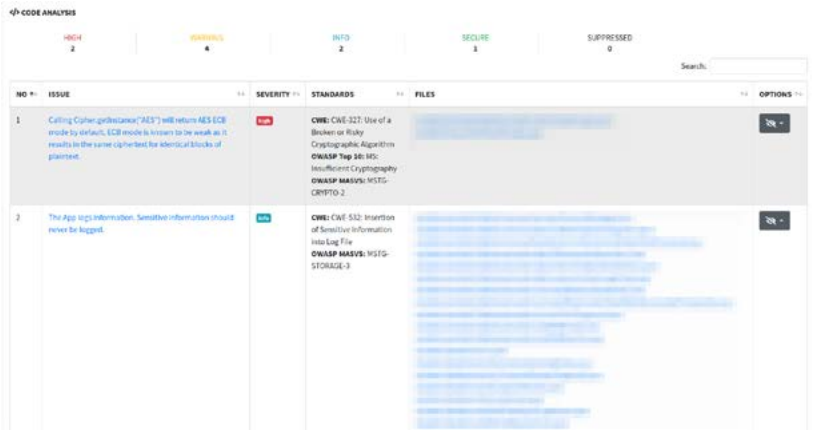
Manifest Analysis:

Contiene información del fichero Manifest usado por la aplicación. El archivo de manifiesto describe información esencial de la aplicación para las herramientas de creación de Android, el sistema operativo Android y Google Play. Como se puede ver en el ejemplo, la aplicación aporta cierta información sobre cómo se puede resolver el hallazgo.



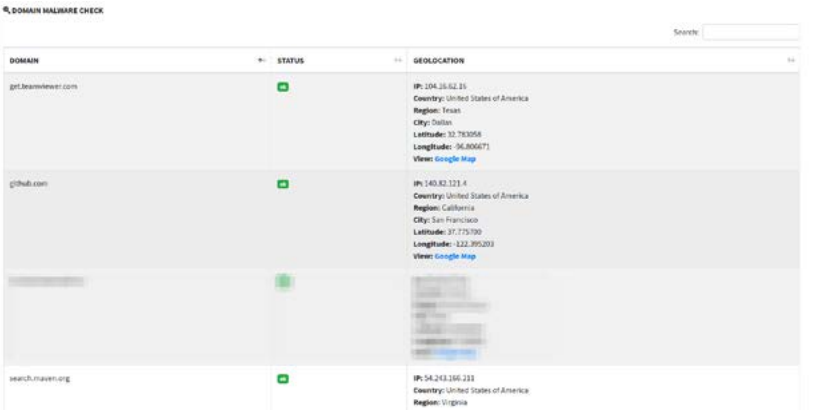
Code Analysis:

Contiene el análisis del código de la aplicación. En el ejemplo aparecen dos vulnerabilidades (una crítica y una de tipo informativo), donde se muestra la vulnerabilidad (Issue), los estándares que lo presentan como vulnerabilidad (OWASP Top 10, CWE, OWASP MASVS) y los ficheros a los que afecta:



Domain Malware Check:

Se revisan los dominios usados por las distintas funcionalidades de la aplicación.



—02.03

Herramientas para revisar la infraestructura

En este punto vamos a centrarnos en herramientas para evaluar la infraestructura desplegada en nuestro proyecto. En este caso, hemos elegido la herramienta OpenVAS. Esta herramienta ofrece un escaneo de vulnerabilidades de la infraestructura de las soluciones evaluadas.

Entre sus capacidades se incluyen: pruebas autenticadas y no autenticadas, uso de protocolos industriales y de Internet de alto y bajo nivel y ajuste de rendimiento para exploraciones a gran escala, entre otros.

Ejecución

En la documentación se indica que existe la posibilidad de ejecutar la herramienta de distintas maneras, pero para hacerlo de una manera más rápida y sencilla, te proponemos descargar el script de la propia página. Este script levanta un cluster de contenedores docker con la herramienta desplegada y accesible desde el navegador web.

Para ejecutar el script debes cumplir los prerrequisitos de instalación. En esta guía te contamos cómo instalarlo en Linux según la documentación oficial de la herramienta:

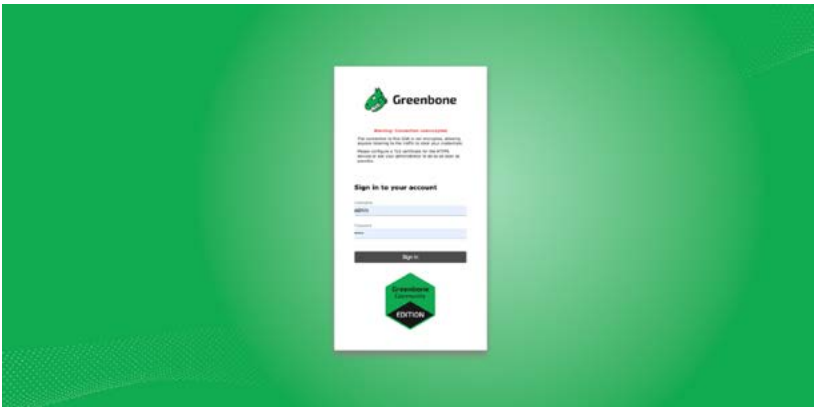
Descargar el script del repositorio oficial:

```
curl -f -O https://greenbone.github.io/docs/latest/_static/
setup-and-start-greenbone-community-edition.sh && chmod u+x
setup-and-start-greenbone-community-edition.sh
```

Otorgar permisos de ejecución al script y ejecutarlo:

```
./setup-and-start-greenbone-community-edition.sh
```

Una vez está la herramienta en ejecución, procederemos a abrir la interfaz desde el navegador:



Las credenciales de acceso, tal y como se indica en la documentación, son:

User: admin

Password: admin

La contraseña puede ser modificada en el momento en el que el despliegue de la herramienta ha finalizado:

Una vez dentro, se pueden observar las siguientes funcionalidades de la herramienta. Aquí te dejamos un ejemplo de visualización:

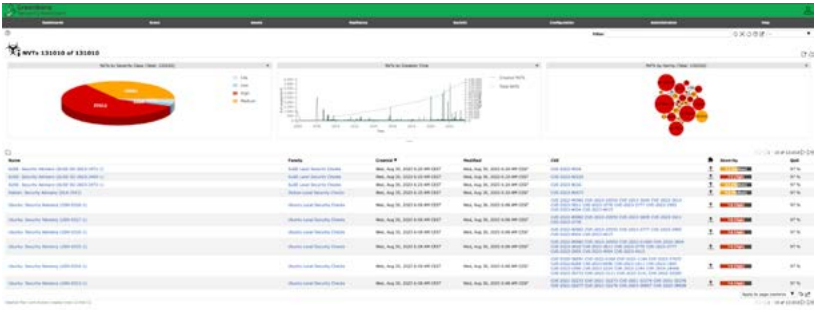
Dentro de los apartados que aparecen en el menú de la herramienta, destacan los siguientes:

Apartado general:
En este apartado se ven los dashboards de los resultados del escaneo realizado.

Listado de escaneos con sus reportes:
Este apartado contiene las distintas vulnerabilidades registradas después del escaneo.

Listado de los recursos a escanear:
Este apartado contiene los recursos que previamente hemos configurado y sobre los que se realiza el escaneo.

Apartado SecInfo:
Este apartado contiene el listado de las vulnerabilidades detectadas y los "CVE" a los que pertenecen. A continuación, puedes ver un ejemplo de este apartado:



Como se puede ver en los ejemplos, esta herramienta proporciona una amplia cantidad de funcionalidades y configuraciones que le permiten generar reportes de la manera que mejor se adapte a las necesidades del proyecto.

Esta herramienta proporciona una amplia cantidad de funcionalidades y configuraciones.

Interpretación de los resultados

Una vez generados los reportes, se obtienen dos apartados:

Revisión general de la infraestructura.

Prueba de autenticación por cada host escaneado.

Revisión por cada host escaneado:
En este apartado se diferencian las vulnerabilidades por:

High: Para vulnerabilidades que es primordial corregir cuanto antes.

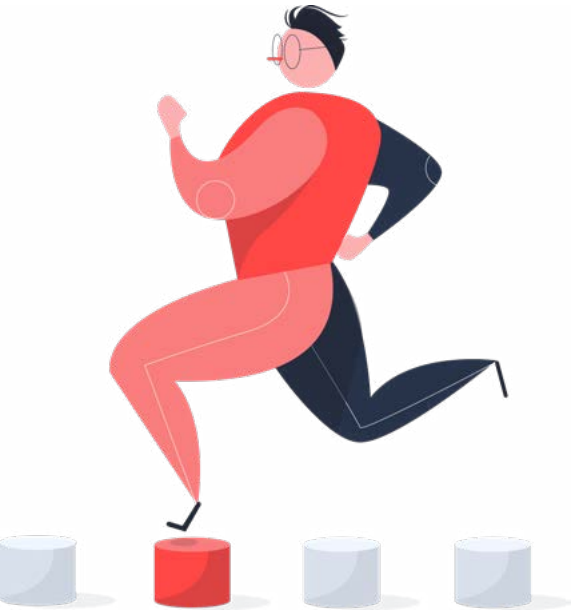
Medium: Para vulnerabilidades que se pueden corregir en el corto plazo.

Low: Para vulnerabilidades que se pueden corregir en el medio plazo.

Dentro de cada clasificación por tipo de vulnerabilidad se generan los siguientes reportes:

- **Product Detection Result**: En este apartado se describe el sis tema operativo detectado.
- **Summary**: Resumen de las vulnerabilidades detectadas en ese sistema operativo.
- **Vulnerability Detection Result**: Resultado de las vulnerabilidades que se han detectado de forma más detallada.

- **Impact**: Impacto que tienen las vulnerabilidades sobre el sistema.
- **Solution**: Posible solución a las vulnerabilidades detectadas.
- **Vulnerability Detection Method**: Método empleado para poder detectar las vulnerabilidades.
- **Product Detection Result**: Resultado de la detección del producto, donde se detalla el sistema y el método utilizado para hacer los escaneos.
- **References**: Referencias recomendables para poder entender la vulnerabilidad y encontrar una solución.



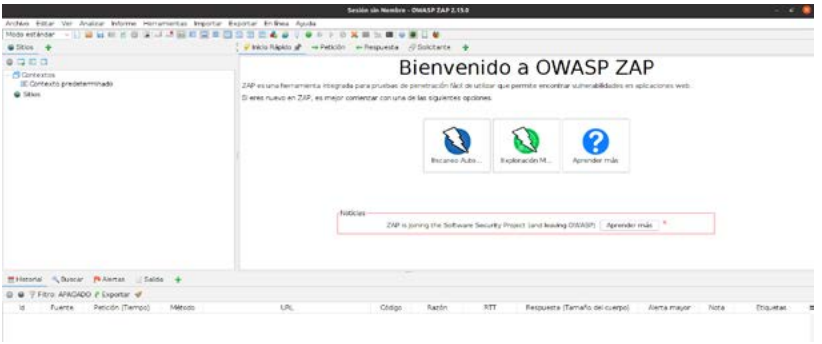
—02.04

Herramientas para revisar la web

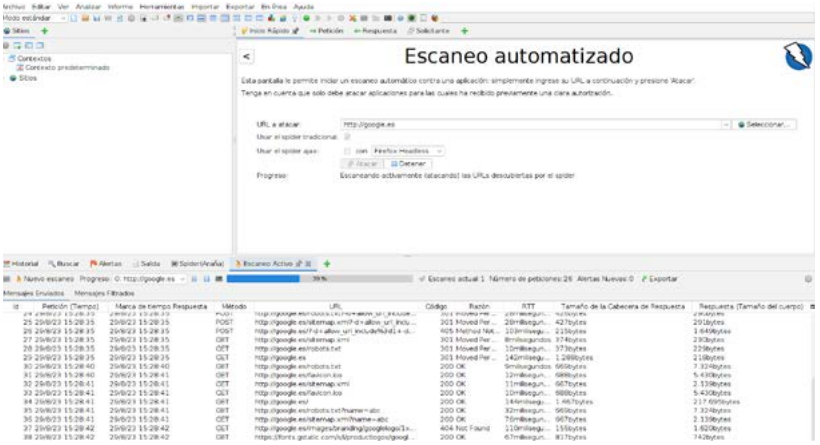
Para la revisión de la web hemos elegido la herramienta ZAP. Esta herramienta ofrece pruebas de pentesting de manera gratuita. Está diseñada para probar aplicaciones web y es flexible y extensible. Actúa como un "proxy man-in-the-middle", situándose entre el navegador de la persona que realiza la prueba y la aplicación web, de modo que puede interceptar e inspeccionar los mensajes enviados entre el navegador y la aplicación web, modificar el contenido si es necesario y reenviar esos paquetes al destino.

Ejecución

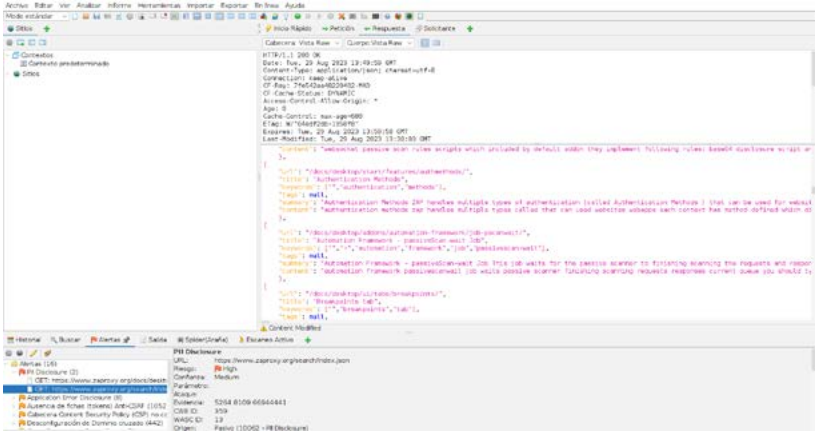
Esta herramienta se puede descargar para diferentes plataformas desde su página web. Una vez instalada la aplicación, procederemos a abrirla:



Y seleccionamos la pestaña “Inicio rápido”. En la ventana que aparece, añadiremos la URL de la aplicación que queremos revisar. En el ejemplo, vamos a escanear la web de la propia aplicación:

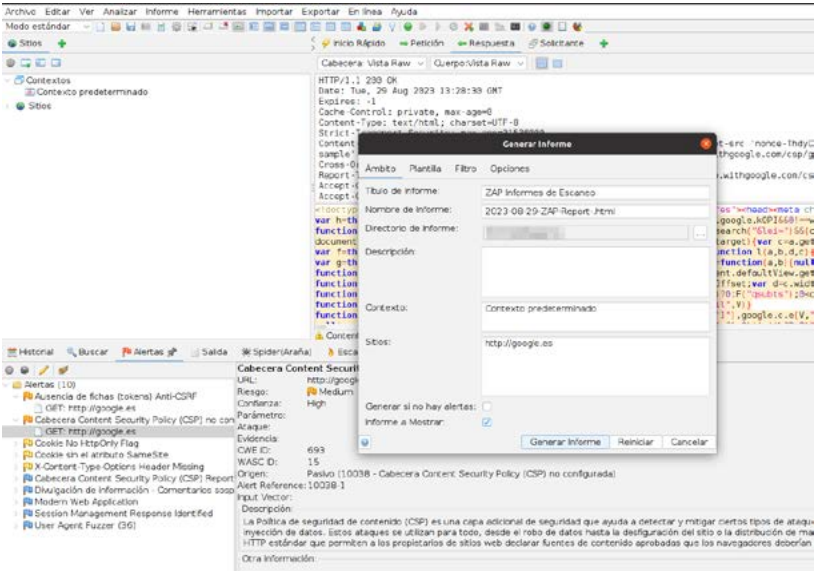


Una vez se termine el escaneo, aparecerán los resultados de las vulnerabilidades detectadas por la aplicación:



Interpretación de los resultados

Una vez terminado el escaneo, podemos descargar el informe desde la opción “Informe”.



En el contenido del informe podemos encontrar, entre otros apartados, información sobre:

- **Volumetría total de vulnerabilidades detectadas**: Las vulnerabilidades se presentan en su totalidad y se clasifican en función de su criticidad. La severidad puede ser de tipo:

Alta: Para vulnerabilidades que es primordial corregir cuanto antes.

Media: Para vulnerabilidades que se pueden corregir en el corto plazo.

Baja: Para vulnerabilidades que se pueden corregir en el medio plazo.

Info: Vulnerabilidades de tipo informativo que convendría corregir o, al menos, estudiar si es viable su corrección.

Los resultados totales de la volumetría de vulnerabilidades se presentan de la siguiente manera en el informe:

Alert counts by risk and confidence

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

		Confidence				
		Confirmado por Usuario	Alta	Media	Baja	Total
Risk	Alto	0 (0,0 %)	0 (0,0 %)	1 (6,2 %)	0 (0,0 %)	1 (6,2 %)
	Medio	0 (0,0 %)	1 (6,2 %)	2 (12,5 %)	1 (6,2 %)	4 (25,0 %)
	Bajo	0 (0,0 %)	1 (6,2 %)	4 (25,0 %)	0 (0,0 %)	5 (31,2 %)
	Informativo	0 (0,0 %)	1 (6,2 %)	2 (12,5 %)	3 (18,8 %)	6 (37,5 %)
	Total	0 (0,0 %)	3 (18,8 %)	9 (56,2 %)	4 (25,0 %)	16 (100%)

Volumetría de vulnerabilidades por sitio web:

En este apartado se reporta el número de vulnerabilidades encontradas por cada sitio web escaneado. A continuación se incluye un ejemplo:

Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

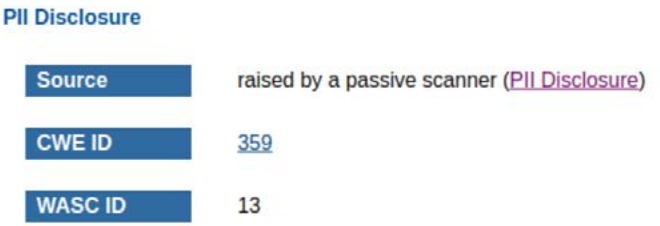
		Risk			
		Alto (= Alto)	Medio (>= Medio)	Bajo (>= Bajo)	Informativo (>= Informativo)
Site	https://www.zaproxy.org	1 (1)	4 (5)	5 (10)	6 (16)

Listado de las vulnerabilidades detectadas:

En el informe se incluye un listado de todas las vulnerabilidades detectadas, haciendo referencia al número de ocurrencias y a la severidad de la vulnerabilidad.

Alert type	Risk	Count
PII Disclosure	Alto	2 (12,5 %)
Application Error Disclosure	Medio	9 (56,2 %)
Ausencia de fichas (tokens) Anti-CSRF	Medio	1152 (7.200,0 %)

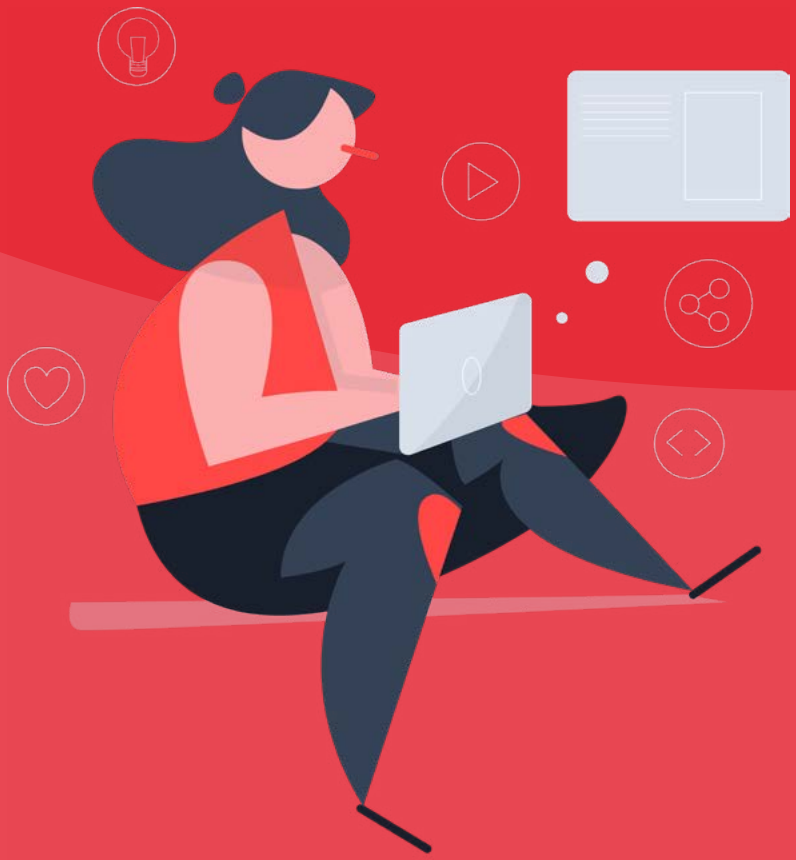
Al seleccionar cada vulnerabilidad, podremos acceder al detalle de la misma:



En el detalle se puede acceder a las propuestas de resolución de la incidencia en el campo “Source”.

—03

Contenido del informe.



¡Ya tienes toda la información relativa a la seguridad de tu proyecto! Ahora toca darle forma y organizar todos los datos. Para ello, te dejamos los principales apartados que debería tener tu informe de seguridad.

—03.01
Resumen ejecutivo

El objetivo de este apartado es tener un resumen de todo el contenido del documento con los puntos más importantes. Entre los apartados que debería incluir este resumen, priorizaremos:

Alcance del informe:
Describir brevemente el proyecto o aplicación y explicar qué tipo de análisis se han realizado. En nuestro caso, añadiremos que se han realizado los siguientes análisis:

- Análisis del código de la infraestructura
- Análisis del código de la aplicación
- Análisis de la infraestructura
- Análisis de la web

Las herramientas usadas para los distintos análisis:
Es importante explicar brevemente qué herramientas se han usado y por qué se han escogido. También es recomendable hacer un breve resumen de cómo se han categorizado o interpretado las vulnerabilidades detectadas. En caso de que este resumen se quede muy largo, se pueden poner referencias a los apartados posteriores del documento donde se explique de manera más extensa el proceso de análisis y revisión de cada vulnerabilidad.

Conclusiones después del análisis:
En el apartado de conclusiones se debería añadir:

El total de vulnerabilidades reportadas:
Deberías tener un apartado donde se indique, al menos, el número de vulnerabilidades reportadas después del análisis. Es conveniente separar las vulnerabilidades según el ámbito al que apliquen. En nuestro caso, según los análisis realizados, tendríamos los siguientes sub-apartados:

- Número de vulnerabilidades del código de la infraestructura
- Número de vulnerabilidades del código de la aplicación
- Número de vulnerabilidades de la infraestructura
- Número de vulnerabilidades de la web

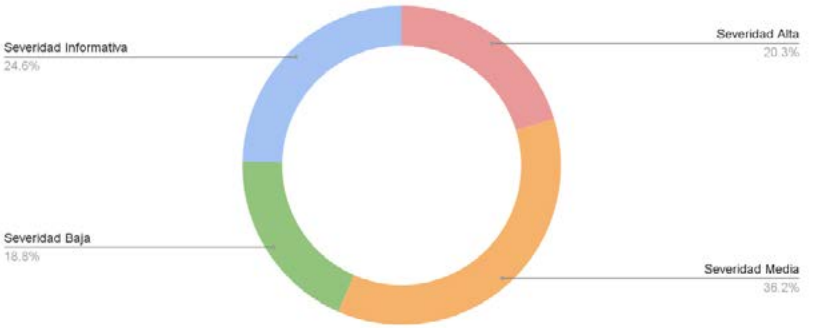
El formato tabla es bastante adecuado para presentar esta información:

Vulnerabilidades tras el análisis					
Vulnerabilidades en total	Severidad crítica	Severidad Alta	Severidad Media	Severidad Baja	Severidad Informativa

Es recomendable segregar los datos por cada tipo de informe que se haya realizado, como en el ejemplo a continuación:

Volumetría de vulnerabilidades por informe				
	Severidad Alta	Severidad Media	Severidad Baja	Severidad Informativa
Informe de la infraestructura				
Informe del código de la infraestructura				
Informe de la Aplicación				
Informe de la web				

Si quieres, puedes apoyarte en gráficas de tipo tarta con representación de los porcentajes de las vulnerabilidades detectadas:



La mitigación de vulnerabilidades:
En caso de que se haya mitigado alguna vulnerabilidad durante el análisis, debería indicarse por qué se han corregido y cómo se han corregido.

El formato más adecuado en este caso vuelve a ser el formato tabla, donde se muestran las vulnerabilidades de un solo vistazo:

La propuesta de mitigación:
Dado que este apartado es el resumen ejecutivo, se debería añadir la propuesta de mitigación de las vulnerabilidades con criticidad alta como mínimo. Es recomendable añadir una alusión al apartado correspondiente del documento donde se detalle la propuesta de mitigación del resto de vulnerabilidades.

Se debería añadir la propuesta de mitigación de las vulnerabilidades con criticidad alta como mínimo.

A continuación te dejamos un ejemplo para poner los datos anteriores en una tabla:

Descripción	Categoría	Estado	Criticidad	Solución propuesta	Comentario de Resolución
Grupos de Seguridad sin tráfico restringido.	Informe del código de la infraestructura	No Aplica	Alta	Eliminar las entradas 0.0.0.0/0 y ::/0 de los security groups definidos en Terraform	
Volumen EFS no cifrado	Informe del código de la infraestructura	Resuelto	Alta	Cifrar en Terraform el disco EFS asociado a las instancias	
EFS sin KMS definido	Informe del código de la infraestructura	Resuelto	Alta	EFS usa KMS CMK en lugar de AWS managed-keys	
IAM Database Auth no habilitado	Informe del código de la infraestructura	No Aplica	Alta	Iam Database Auth Enabled debería estar configurado a true cuando se usen con una versión compatible	

—03.02

Reporte de las vulnerabilidades detectadas

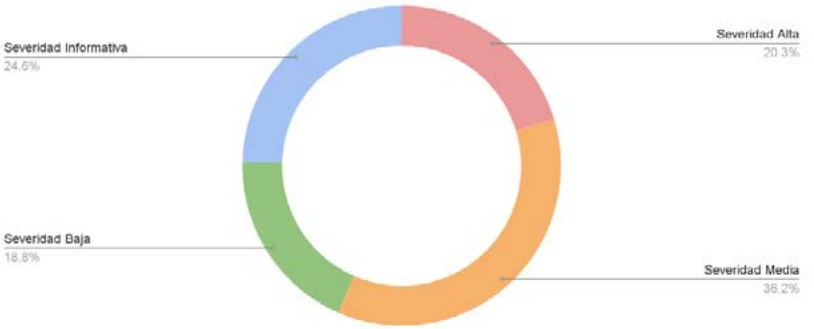
Este es el primer apartado después del resumen ejecutivo. Este apartado y los siguientes están destinados a explicar en detalle el proceso de análisis y estudio de las vulnerabilidades detectadas. Por lo tanto, puedes extenderte tanto como sea necesario.

En este apartado deberían incluirse:

Listado de todas las vulnerabilidades detectadas:
En este apartado deberían incluirse subapartados donde se clasificasen las vulnerabilidades según el tipo de informe al que pertenecen. En el ejemplo de los informes propuestos en esta guía, deberían estar los subapartados:

- Número de vulnerabilidades del código de la infraestructura
- Número de vulnerabilidades de la aplicación
- Número de vulnerabilidades de la infraestructura
- Número de vulnerabilidades de la web

Quedará mucho más claro si lo acompañas de una tabla o de algún diagrama representativo de los porcentajes. Aquí tienes un ejemplo de representación de las vulnerabilidades según su porcentaje:



Vulnerabilidades de cada informe:
Dado que es el momento de extenderse en las explicaciones, deberías incluir un apartado detallado por cada informe que hayas realizado. Siguiendo los ejemplos de informes anteriores, deberías tener los siguientes apartados:

- Informe de seguridad del código de la infraestructura
- Informe de seguridad del código de la aplicación
- Informe de seguridad de la infraestructura
- Informe de seguridad de la web

Cada uno de los apartados debería incluir:

Interpretación de los resultados de los análisis:
En este punto deberías hablar sobre la criticidad de las vulnerabilidades y cómo se interpretan en tu informe. En caso de ser necesario, deberías añadir en este punto cualquier consideración del análisis realizado (por ejemplo, si has decidido omitir en el informe las vulnerabilidades de tipo informativo o alguna consideración que hayas realizado).

Volumetría de las vulnerabilidades: Este apartado debería tener todas las vulnerabilidades reportadas en el informe correspondiente. Te recomendamos poner al principio del apartado una tabla resumen de este tipo:

Vulnerabilidades detectadas en el informe					
Vulnerabilidades en total APP	Severidad crítica	Severidad Alta	Severidad Media	Severidad Baja	Severidad Informativa

Y después, poner una tabla más extensa definiendo todas las vulnerabilidades detectadas y su posible mitigación. A continuación te dejamos un ejemplo:

Tipo de vulnerabilidad	Descripción	Categoría	Estado	Criticidad	Solución propuesta
Cifrado	Cifrado débil para la configuración del certificado	Informe de la web	No resuelto	Media	
Cifrado	Cifrado débil para la configuración del certificado	Informe de la web	No resuelto	Media	
Cifrado	HSTS Preloading no habilitado en Chrome	Informe de la web	No resuelto	Media	

—03.03
Anexo

Te recomendamos que incluyas un apartado de anexo con una de las dos opciones citadas a continuación:

- Poner referencias a cada uno de los informes de seguridad que se han obtenido de las distintas herramientas.
- Adjuntar todos los informes de las distintas herramientas.

Esto último es importante para que el informe cuente con la transparencia y solidez que aporta el tener los distintos informes de los que se nutre.

—04

Conclusiones.

Después de esta guía, ya tienes las bases para hacer un informe de seguridad sobre un proyecto o aplicación.

Y como bonus, te dejamos algunos consejos generales para que tu informe sea de 10:

Que el miedo no te paralice:
Es posible que el primer informe que redactes te cueste un poquito, pero una vez hecho el primero ya vas a tener una base muy importante que te ayude a hacer el resto.

Sé sincero:
No ocultes ni maquilles información. Lo mejor en estos casos es ser transparente con la información. Todo se puede solucionar o poner medios para arreglarlo.

Intenta hacer una plantilla:
Si te estás planteando hacer más de un informe de este tipo, párate a pensar:

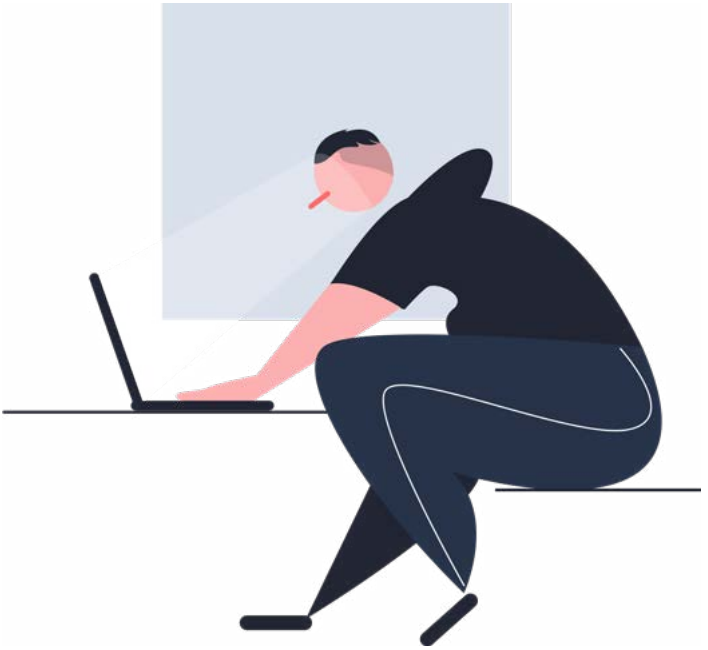
- ¿Qué quiero que contenga mi informe?
- ¿Cómo plantearía el esqueleto de mi informe?
- ¿Me sirven estas herramientas o prefiero usar otras?

Si haces una plantilla de este tipo de informe y la vas evolucionando, tendrás un punto de valor extra para apoyar todos tus proyectos.

Como hemos dicho al principio, este tipo de informes solo aportan valor, así que ¡no tengas miedo de lanzarte a hacer alguno!

Referencias.

- Kics
- Instalación de Kics
- MobSF
- Instalación de MobSF
- APKID
- OpenVAS
- Instalación de OpenVAS - Greenbone
- Pre-requisitos de instalación de OpenVAS - Greenbone
- ZAP



Autores/as:

Alberto Perrote.

Sysadmin Cloud en Paradigma Digital

Curioso, me encanta aprender y escuchar. Apasionado de las nuevas tecnologías y del deporte. He pasado por varias posiciones y trabajos dentro de mi carrera profesional que me han ayudado a comprender muchas de las cosas que me suceden en el día a día. Actualmente trabajo en el equipo de Cloud Gestionada, con numerosas tecnologías pero sobre todo poniendo foco en infraestructuras en la nube, tanto en AWS como en GCP.

Alba García.

Responsable de servicios gestionados en Cloud en Paradigma Digital

Curiosa desde siempre, me encanta aprender y saber cómo funcionan las cosas. He pasado por diversos roles que me han permitido aprender mucho, empezando por desarrollo de software, siguiendo como administradora de sistemas y finalmente trabajando con infraestructuras en cloud, centrándome sobre todo en Amazon Web Services y Google Cloud Platform.

Think Big.

V.1.0 - Febrero de 2024

info@paradigmadigital.com

eBooks
Paradigma